A Review on Hybrid Artificial Intelligence Systems for Integrating Circuit-Level Optimization with Machine Learning Algorithms

Umesh Balkishan Phiske[1], Dr. Gopalkrishna D Dalvi[2]

ALARD School of Doctoral Research, ALARD University, Pune

Mail Id: phiskeumesh@gmail.com

ABSTRACT

The rapid evolution of semiconductor technologies has fueled a significant surge in research on hybrid artificial intelligence (AI) systems, particularly for circuit-level optimization integrated with machine learning (ML). These systems address complex challenges in electronic design by leveraging multiple AI methodologies, such as neural networks, reinforcement learning, and statistical modeling. This review presents a comprehensive analysis of recent advancements in this domain, focusing on trends, methodologies, and global contributions. Between 2015 and 2024, research output grew exponentially, with publications rising from 2 in 2015 to 635 in 2024, underscoring increasing academic and industrial interest. Bibliometric analysis highlights leading sources, with fields like distributed computing and human-centered computing exhibiting the highest citation impacts. Prominent authors and influential publications reflect the intellectual foundations of this field. Countries like China, Saudi Arabia, and the United States dominate contributions, emphasizing global collaboration. Hybrid AI systems excel in optimizing analog and mixed-signal circuits through techniques such as Bayesian optimization, neural networks, and co-simulation with CAD/EDA tools, significantly enhancing efficiency and accuracy. Key applications include automated circuit sizing, fault diagnosis, and RF/microwave circuit modeling. Challenges such as data integration, computational overhead, and variability in manufacturing demand innovative solutions. Future research directions prioritize improving model interpretability, reducing computational costs, and integrating hybrid AI with emerging trends like IoT and edge computing. This review bridges theoretical innovations with practical implementations, contributing to the advancement of AI-driven integrated circuit design and optimization.

I. INTRODUCTION

The application of Artificial Intelligence (AI) in circuit-level optimization has transformed how electronic designs are conceived and developed. With the increasing complexity of integrated circuits (ICs) in the era of advanced semiconductor technologies, traditional methods of optimization have struggled to keep pace with performance and efficiency demands. AI, particularly machine learning (ML), offers the potential to revolutionize this domain by enabling intelligent, data-driven decision-making. Hybrid AI systems, which integrate multiple AI approaches such as ML, deep learning (DL), and expert systems, have emerged as a robust solution for optimizing circuit performance, minimizing power consumption, and reducing design time.

Hybrid AI systems leverage the strengths of various algorithms to address the multifaceted challenges of IC design. These systems combine rule-based methods, statistical models, and ML techniques to achieve optimization goals effectively. For instance, while rule-based systems can quickly identify constraints and boundaries, ML algorithms excel at discovering patterns and predicting optimal conFigureurations. This synergy enables a more comprehensive approach to circuit-level optimization, encompassing aspects such as layout design, timing analysis, and power management. Recent advancements highlight the ability of hybrid systems to balance trade-offs, such as performance versus energy consumption, more effectively than standalone methods.

Machine learning algorithms are central to hybrid AI systems due to their capability to analyze vast datasets generated during IC design processes. Algorithms such as support vector machines (SVMs), decision trees, and neural networks are increasingly used to predict performance metrics, identify bottlenecks, and suggest design improvements. ML techniques also play a pivotal role in automating routine tasks, enabling designers to focus on higher-level optimizations. Moreover, reinforcement learning (RL) has demonstrated exceptional potential in dynamic optimization scenarios, allowing systems to adapt and improve over time based on iterative feedback.

Despite their promise, hybrid AI systems face several challenges in circuit-level optimization. One of the primary hurdles is the integration of diverse data sources and algorithms to ensure seamless operation. Variability in manufacturing processes and the stochastic nature of electronic behavior further complicate optimization efforts. Additionally, the computational overhead of training sophisticated ML models can be significant, requiring careful balancing of accuracy and efficiency. Addressing these challenges requires innovative approaches to hybrid system design, including the development of lightweight algorithms and efficient data preprocessing techniques.

As the demand for high-performance, energy-efficient electronics continues to grow, the role of hybrid AI systems in IC design is set to expand. Emerging trends such as edge computing, Internet of Things (IoT), and autonomous systems underscore the need for scalable and adaptable optimization solutions. The integration of AI with traditional electronic design automation (EDA) tools presents an exciting frontier, promising to redefine industry standards and capabilities. Future research is expected to focus on improving interpretability, enhancing generalization, and reducing the computational costs of hybrid AI systems, ensuring their widespread adoption in the semiconductor industry.

This literature review seeks to explore the advancements, challenges, and future directions of hybrid AI systems in circuit-level optimization, with a focus on their integration with machine learning algorithms. By bridging the gap between theoretical innovation and practical implementation, the study aims to contribute to the evolving landscape of AI-driven IC design.

1.1 Artificial Intelligence & Machine Learning Main Concepts and Techniques

A. General Terminology

Artificial intelligence and Machine Learning (ML) are terms used to describe algorithms that can learn patterns without direct human involvement. Generally, these algorithms are trained, meaning that they learn patterns from some dataset, then they are used for inference. There are several common ways to train a machine learning model e.g. supervised, unsupervised, semi-supervised, and Reinforcement Learning (RL).

1) Supervised Learning: Supervised learning is when the correct corresponding outputs, called labels, are known for every input. This is helpful to train a model to recognize specific useful patterns. Supervised learning allows models to achieve high accuracy in many tasks, such as classification or regression, when enough data is present. The largest drawback of supervised learning is that it either requires the training dataset to already have properly labeled data or it requires a significant amount of work to create and maintain a properly labeled dataset which is both time-consuming and is hard to achieve. Additionally, the designers need to be careful when choosing their training data. If this subset is not representative of the entire set, the model may output inconstant results during inference.

2) Unsupervised Learning: In contrast to supervised learning, unsupervised learning does not require labeled data. This means that less effort is required to obtain proper data, but in exchange, it can be harder to train the model to create the desired output. A powerful approach with unsupervised learning is to cluster the data. Clustering can lead to finding similarities between the features of various inputs to help classification or data extraction.

3) Semi-Supervised Learning: Semi-supervised learning, as the name implies, is a hybrid between the previous two types of learning. Generally, this approach first uses unsupervised learning on a large set of unlabeled data to learn robust patterns. Then, using the labeled data, the model is trained to use those learned patterns to output relevant data. This approach, if implemented correctly, can achieve the best of both worlds. It does not need a vast set of labeled data, yet the training can still be directed. However, additional effort is needed to make sure that both parts of the network properly converge.

4) Reinforcement Learning: Reinforcement learning is a version of machine learning where a software agent, a program with the ability to learn, is rewarded for certain actions. Even though RL models are trained for a specific set of specifications and their reuse is not guaranteed, a major benefit of them over supervised learning is that optimal solutions to problems do not need to be known beforehand. For instance, Wang et al. use RL to optimize the circuit parameters. They “reward” the model when it outputs a circuit with the required specifications and low power consumption and area.

5) AI Model Verification and Performance Evaluation: Designers must exercise caution when training models due to the potential for underfitting and overfitting. Under fitting happens when the model has not recognized all of the general patterns that exist in the training data. To fix this issue, the model can be trained longer on the data in order to reach a local minimum or even the global minimum. However, if the model is too simple to learn all of the patterns, then further training will not be beneficial. In this case, a larger or more complex model is needed in order to get better results. On the other hand, overfitting occurs when the model learns to identify non-generalizable patterns within the training data. For instance, a model that detects human faces would be overfitting if it could only detect faces indoors. This might happen because all of the training data with faces came from pictures taken indoors. Adding more diverse training data can help reduce this problem. Overfitting occurs more readily in larger, more complex models since they can identify and use more features compared to simpler models. To combat this, several tricks have been proposed to reduce this issue in various types of models, but there is no perfect solution to stop overfitting.

B. Common AI & ML Models

1) Support Vector Machines: Many different types of artificial intelligence and machine learning models have been tested over the years. One such model, Support Vector Machine (SVM) is a supervised learning model that is mostly used for classification or regression problems. For classification, SVMs operate by finding an optimal hyperplane to separate the features of two or more classes. An SVM can be seen separating two classes in Figure. 1. SVMs are used for regression by including a distance measure in the loss function. SVMs are widely praised at the start of the twenty-first century for their resilience to overfitting when compared to other ML techniques such as neural networks. Today SVM models can still achieve state-of-the-art performance in certain circuit design applications such as a seizure detection sensor.

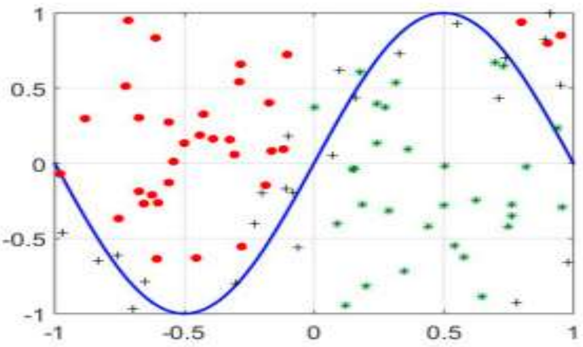

Figure.1 An image showing a Support Vector Machine (SVM) separate two different classes (red dots and green starts) with two different features

2) Artificial Neural Networks: Another type of ML model, artificial neural networks, are designed to imitate how real neurons transmit and interpret data. Groups of neurons, or nodes, that are not connected with one another are called layers. Multiple layers can be used to allow complex non-linear patterns to be learned as shown in Figure. 2. The output or activation, of a single neuron in an ANN can be represented with the following equation.

$$a = f(\sum_{j=1}^{k} w_j x_j + b), \quad \textbf{(1)}$$

where k is the number of activations from the previous layer that are connected with this node. Each of these activations, x, is multiplied by a unique learned weight w, then all of these products are summed together. Next, a learned constant bias b is added to the sum before the result is sent into a non-linear function f. Many non-linear functions can be used including the Rectified Linear Unit (ReLU), the sigmoid function, and the hyperbolic tangent. ANNs have broad applications in NLP, computer architecture and circuit design.

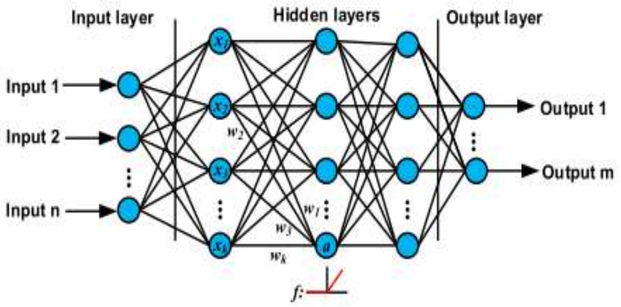

Figure.2 An image showing an Artificial Neural Network (ANN) with several intermediate, hidden, layers

3) Deep Neural Networks: With the advent of more computational power and efficient training methods, neural networks began to get larger and deeper. A new term was coined for networks with tens to hundreds of layers, Deep Neural Networks (DNN). The layers in between the input and output layers are called hidden layers because their values are hidden to the outside world. This leads to DNNs being treated as black boxes where inputs are converted into corresponding outputs. The standard implementation of a DNN has dense connections between the layers, meaning that every node in every layer connects to every node in the next layer. However, other implementations exist with various benefits. One problem with traditional dense DNNs is how quickly the memory requirement increases with the model size. Because of this, the maximum size of a traditional DNN is still limited. A simple method to reduce the memory requirement would be to connect nodes in layer K to only a subset of the nodes in layer K + 1.

4) *Convolutional Neural Networks:* Another method is to instead have a moving window of weights. In this approach, a small group of weights moves across the entire image, often with a stride of 1. Several of these groups, which are called filters, are used in order to identify many different types of features. This approach is known as a Convolutional Neural Network (CNN). CNNs have multiple layers to break down complex patterns into more manageable ones. A pooling layer is often used in conjunction with a CNN. Pooling layers condense a model-defined amount of nearby data into a representation that requires less space. A common type of pooling is max-pooling. In this type, only the maximum value within each subset of data is propagated to later layers. Pooling layers help reduce computation complexity as well as help reduce overfitting. Despite the wide applications of CNNs, there are many problems that deal with data that are in a non-Euclidean structure such as chemical molecules, social networks, and functional networks of the brain. However, CNNs inputs are required to be tensor e.g. images that are modeled as 2-D structure.

II. RESEARCH METHOD

This review explores hybrid artificial intelligence (AI) systems integrating circuit-level optimization with machine learning algorithms, combining a systematic analysis of publication trends, source domains, leading authors, top documents, and global contributions. The methodology includes a bibliometric analysis of scholarly articles published between 2015 and 2024, retrieved from diverse databases focusing on engineering, computer science, and AI. Data Collection: Relevant publications were identified based on keyword searches in prominent academic databases. Data points such as publication year, citations, authorship, document impact, and country contributions were meticulously recorded. Analysis Techniques: Quantitative techniques, including citation analysis and co-authorship mapping, were employed. Temporal trends in publications were visualized to identify growth patterns, and domain-specific impact was assessed by analyzing citation means across key disciplines. Co-authorship and scientific mapping techniques illuminated collaboration networks and research focus areas. Interpretation: Insights from bibliometric and visualization tools were interpreted to uncover research hotspots, influential contributors, and emerging trends. This structured approach provides a comprehensive overview of hybrid AI systems for circuit-level optimization, bridging technical advancements with academic recognition.

Table 1. Year of Publication

Year	Publication
2015	2
2016	5
2017	11
2018	39
2019	83
2020	156
2021	162
2022	184
2023	250
2024	635

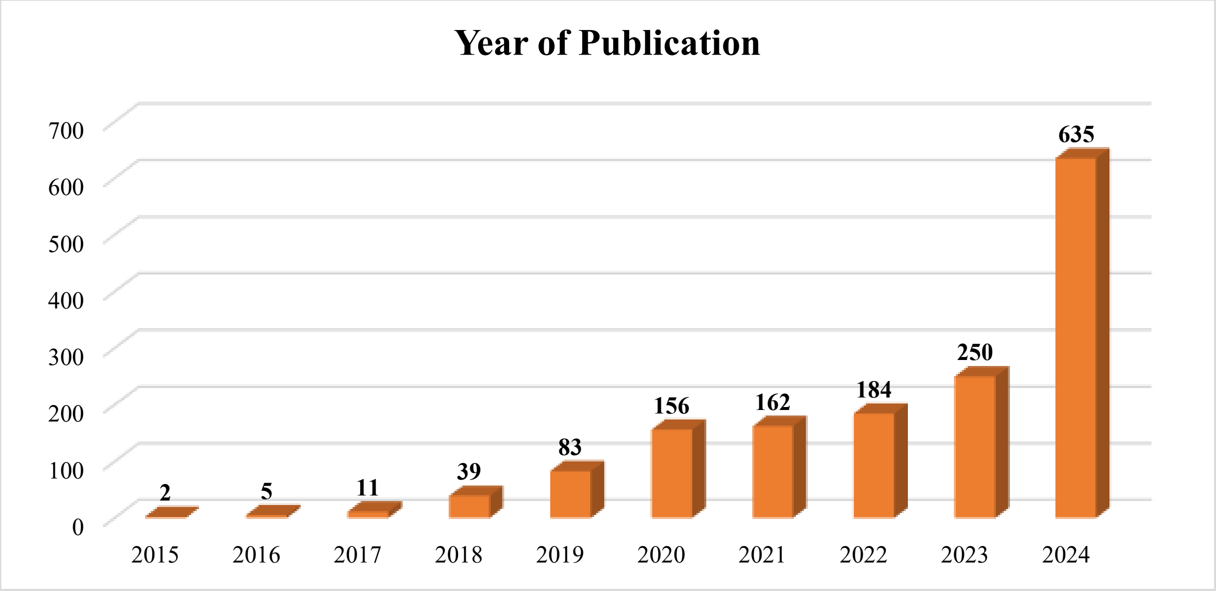

Figure 3. Year of Publication

The publication trend shows a steady and significant increase from 2015 to 2024. In 2015, there were only 2 publications, and this number gradually rose to 5 in 2016, 11 in 2017, and 39 in 2018. The growth accelerated sharply in 2019 with 83 publications, and by 2020, the number surged to 156. The most notable increase occurred from 2020 to 2024, with publications rising from 156 to 635, reflecting a growing interest in this research area. This trend highlights the increasing academic focus on hybrid artificial intelligence systems, particularly for circuit-level optimization. and machine learning applications. The surge in publications underscores the expanding role of these systems in addressing complex engineering problems, particularly in electronics and computational systems. It also reflects the broader adoption of AI-driven optimization methods to tackle evolving challenges in technology. Overall, the data suggests that the integration of machine learning with circuit optimization is becoming a pivotal area of research, with substantial growth in academic contributions in recent years.

Table 2. Source of relevant published articles

Sr. No.	Source	Publications	Citations	Citations mean
1	Information and Computing Sciences	1240	47192	38.06
2	Engineering	806	26018	32.28
3	Data Management and Data Science	418	18221	43.59
4	Electronics, Sensors and Digital Hardware	224	7035	31.41
5	Electrical Engineering	219	6865	31.35
6	Machine Learning	202	7781	38.52

7	Distributed Computing and Systems Software	181	11008	60.82
8	Artificial Intelligence	175	7353	42.02
9	Cybersecurity and Privacy	142	5157	36.32
10	Communications Engineering	121	5679	46.93
11	Computer Vision and Multimedia Computation	108	2520	23.33
12	Human-Centred Computing	85	4934	58.05
13	Theory Of Computation	71	3411	48.04
14	Control Engineering, Mechatronics and Robotics	58	956	16.48
15	Engineering Practice and Education	46	573	12.46

The table provides insights into publications and citations across various research domains, highlighting their academic impact and relevance. Information and Computing Sciences lead with 1,240 publications and 47,192 citations, yielding a mean citation of 38.06, indicating significant scholarly influence. Engineering, with 806 publications and a mean citation of 32.28, reflects steady research activity. Data Management and Data Science stand out with a higher citation mean of 43.59 from 418 publications, showcasing impactful contributions. Distributed Computing and Systems Software demonstrates exceptional influence with a citation mean of 60.82 from 181 publications. Similarly, Human-Centred Computing achieves a notable mean citation of 58.05, despite having only 85 publications. Other fields such as Artificial Intelligence (mean citation: 42.02) and Communications Engineering (46.93) further reflect growing academic attention. Lower citation means are observed in Control Engineering, Mechatronics, and Robotics (16.48) and Engineering Practice and Education (12.46), suggesting emerging areas with room for growth. Overall, the data highlights diverse levels of research activity, with a strong focus on impactful and high-growth domains like computing, data science, and human-centered research.

Table 3. Leading Authors in the Relevant Field

Sr. No.	Name	Publications	Citations	Citations Mean
1	Khursheed Aurangzeb	9	74	8.22
2	Rajamanickam Narayanamoorthi	8	131	16.38
3	Musaed A Alhussein	8	73	9.13
4	Yazeed Yasin Ghadi	6	89	14.83
5	Anand Nayyar	6	336	56
6	Muhammad Shafique	6	192	32
7	Jalil Piran	6	713	118.83
8	Ajith Abraham	6	123	20.5
9	Sudeep Tanwar	6	244	40.67
10	Gulshan Sharma	5	140	28
11	Abdullah Baz	5	83	16.6
12	Mahammad Abdul Hannan	5	9	1.8
13	Naif Al Mudawi	5	101	20.2
14	Farman Ali	5	125	25
15	Sajal Kumar Das	5	301	60.2

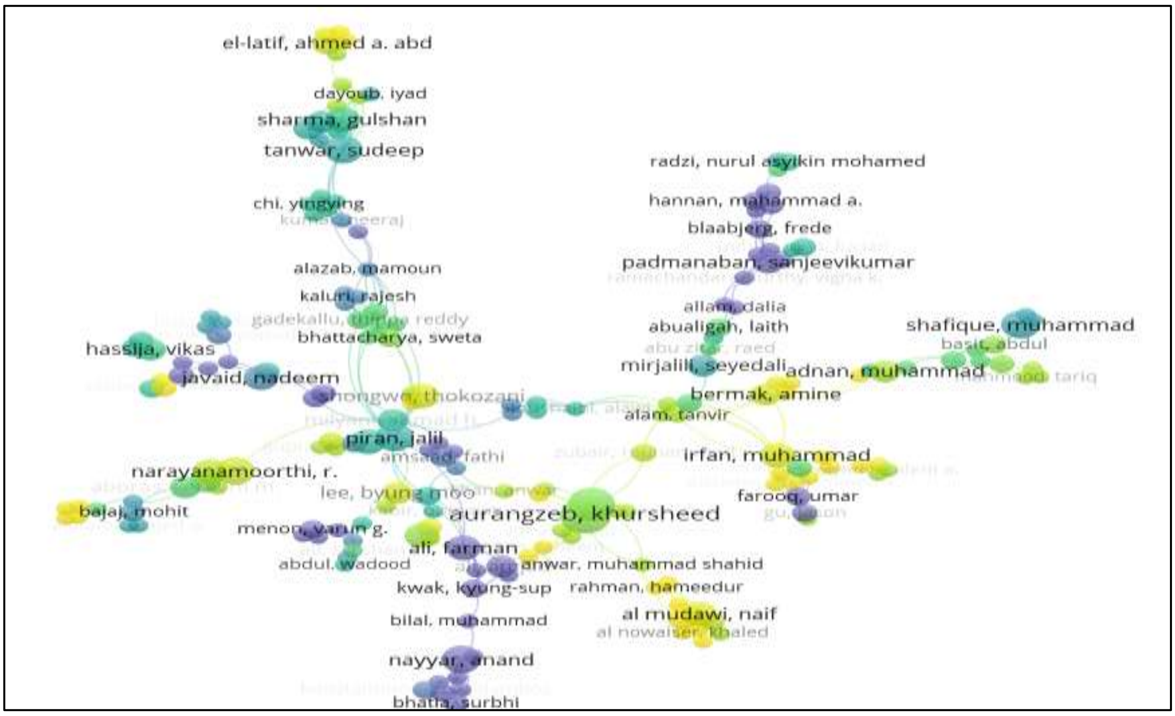

Figure 4. Co-authorship scientific mapping: Overlay Visualization

The table and Figureure highlights the most prolific authors in the field, emphasizing their publication count, total citations, and average citation per publication. Jalil Piran stands out with an impressive citation mean of 118.83 from 6 publications, showcasing exceptional influence. Similarly, Sajal Kumar Das achieves a high citation mean of 60.2 from 5 publications, reflecting impactful research contributions. Anand Nayyar and Sudeep Tanwar also demonstrate significant academic impact with citation means of 56 and 40.67, respectively, from 6 publications each. Muhammad Shafique (32) and Ajith Abraham (20.5) further indicate consistent research activity and influence. Conversely, some authors, such as Mahammad Abdul Hannan (mean citation: 1.8) and Khursheed Aurangzeb (8.22), exhibit lower citation means, indicating potential growth opportunities in their research impact. Overall, the data highlights a diverse range of contributions from leading authors, with significant disparities in citation means reflecting varying levels of influence and recognition in the academic community.

Table 4. Top 15 Mostly Cited Documents in The Relevant Field.

Sr. No.	Documents	Citations	Total link Strength
1	Alsabah (2021)	298	16
2	Nawaz (2019)	447	12
3	Barih (2020)	242	11
4	Pham (2020)	642	10
5	Huang (2019)	394	10
6	Amin (2021)	50	9
7	Ayaz (2019)	798	9
8	Haraz (2024)	0	8
9	Rasheed (2020a)	1091	8
10	Alhaj (2023)	9	8

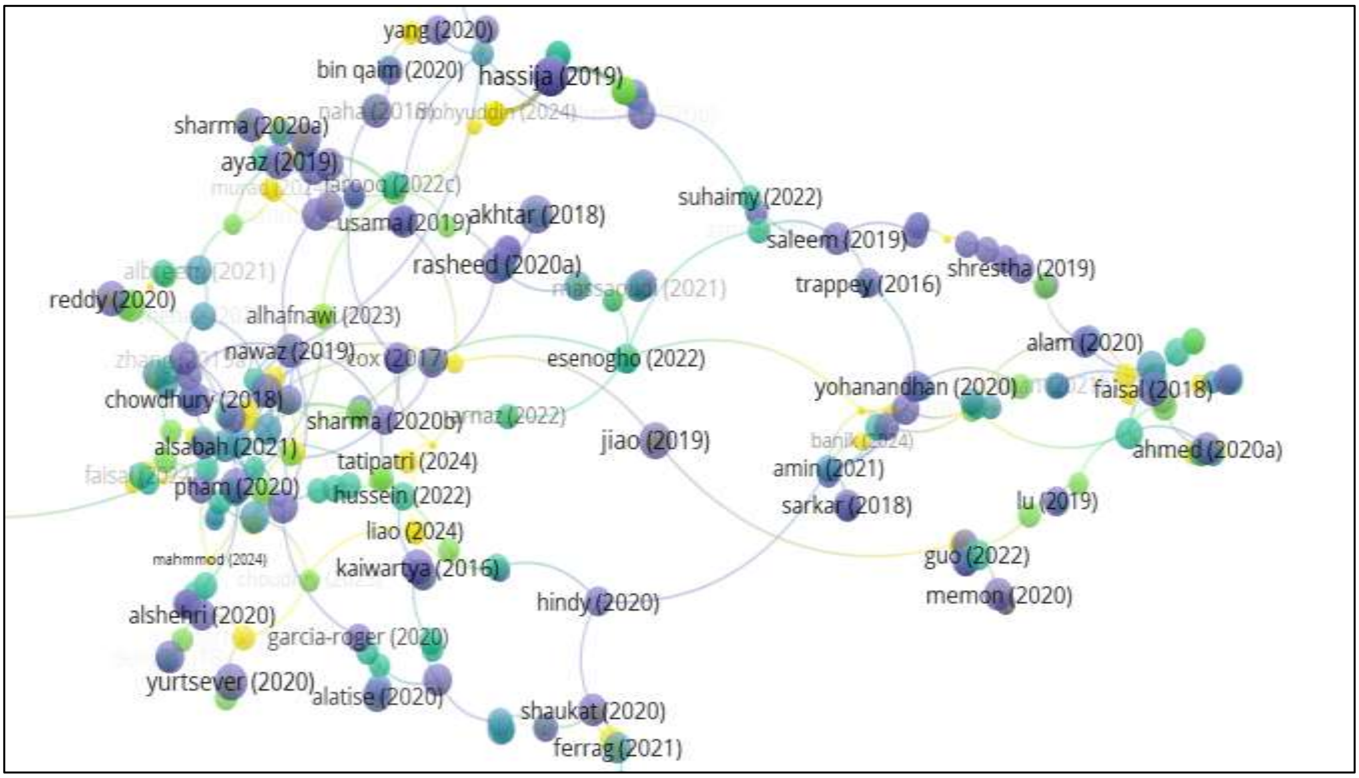

Figure 5. Scientific mapping of the article: Overlay Visualization

The table and Figureure highlights the top 15 documents with the highest citation counts and their corresponding total link strength, indicating their impact and connectedness within the academic community. Rasheed (2020a) leads with an impressive 1,091 citations and a total link strength of 8, showcasing its pivotal role in the field. Ayaz (2019) follows closely with 798 citations and a link strength of 9, reflecting its significant scholarly influence. Similarly, Pham (2020) and Nawaz (2019) have garnered 642 and 447 citations, respectively, indicating their widespread recognition. Documents such as Alsabah (2021) (298 citations, link strength: 16) and **Huang (2019)** (394 citations, link strength: 10) also demonstrate strong academic impact. Interestingly**,** Haraz (2024) has a link strength of 8 but has yet to receive citations, suggesting emerging relevance. Overall, this data underscores the varying degrees of influence among these documents, highlighting their contribution to advancing knowledge and fostering interconnected research in the field

Table 5. Countries Contributing to The Relevant Study Area

Sr. No.	Country	Documents	Citations	Total link Strength
1	Saudi Arabia	244	7702	468
2	Pakistan	144	5894	279
3	United States	203	9086	271
4	India	208	7620	268
5	United Kingdom	132	8503	247
6	China	295	12188	236
7	Malaysia	134	6550	199
8	South Korea	119	4393	172
9	United Arab Emirates	78	3482	159
10	Egypt	79	1836	156
11	Australia	82	7186	121
12	Canada	88	3290	115
13	Spain	51	1631	89
14	Italy	68	1601	83
15	Finland	33	1492	63

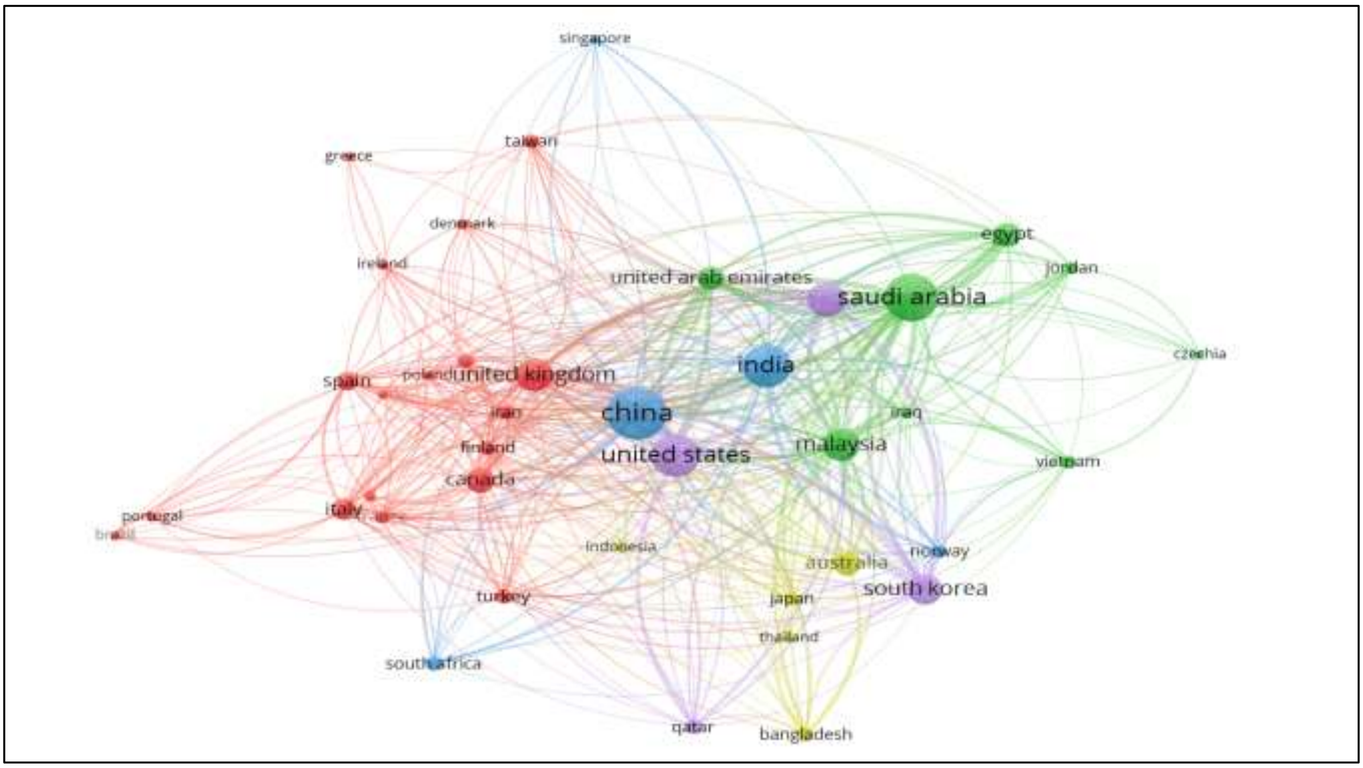

Figure 6. Scientific Mapping of Countries: Network Visualization.

The table and Figure presents the contributions of various countries to the relevant study area, showcasing the number of documents, citations, and total link strength, which indicates the level of collaboration and research impact. China leads with 295 documents and the highest citation count of 12,188, signifying its dominant role in the field. Saudi Arabia follows closely with 244 documents and 7,702 citations, also achieving the highest total link strength of 468, suggesting strong international collaboration. The United States ranks third with 203 documents and 9,086 citations, emphasizing its significant influence in global research. Other countries such as India (208 documents, 7,620 citations) and the United Kingdom (132 documents, 8,503 citations) also make considerable contributions. Pakistan (144 documents, 5,894 citations) and Malaysia (134 documents, 6,550 citations) further strengthen the research landscape. Smaller contributors like Finland (33 documents, 1,492 citations) and Spain (51 documents, 1,631 citations) indicate that while their research output is lower, they still play a role in advancing the field. Overall, the data underscores the global nature of research in this area, with significant contributions from Asia, North America, and Europe, highlighting the importance of international collaboration.

III. RELATED WORK

3.1 Integrated Circuit Design Process Overview

Integrated circuit (IC) design is a highly complex multi-step process involving numerous stages from concept to final product manufacturing. Generally, this process can be roughly divided into three main parts: front-end design (FE), back-end design (BE), and verification (Chong et al. 2018). The front-end design mainly includes specification formulation, logic design and function verification. In the specification development stage, the design team defines the functions, performance indicators and application scenarios of the chip. The logic design phase uses hardware description languages (HDL), such as Verilog or VHDL, to write code to describe the behavior and structure of circuits. The HDL code is then transformed into a gate-level netlist by logical synthesis, and functional verification is performed at this stage to ensure that the design meets expectations (Chovan and Uherek 2018). Back-end design focuses on layout design, physical verification, and manufacturing documentation. Layout is the core link, in which layout involves assigning the spatial location of various components of the circuit on the silicon chip, and routing determines the connection path between these components. Subsequently, physical verification checks whether the design meets criteria such as Electrical Rule Check (ERC), Design Rule Check (DRC), etc. (Bogaerts et al. 2019). Finally, the generated GDSII file is used to guide the actual fabrication of the chip.

3.2 Machine Learning for Circuit Topology Design Automation

Typically, topology design is the first step of analog circuits design, followed by the determination of device sizes and parameters. The process is time-consuming, and unsuitable topology will lead to redesign from the very beginning. Traditionally, topology design relies on the knowledge and experiences of expert designers. As the scale and demand of analog circuits are increasing, CAD tools are urgently needed by engineers. Despite this, automation tools for topology design are still much less explored due to its high degree of freedom. Researchers have attempted to use ML methods to speed up the design process. Some researchers deal with topology selection problem, selecting the most suitable topology from several available candidates. Li et al. focus on extracting well-known building blocks in circuit topology. Recently, Rotman and Wolf use RNN and hypernetwork to generate two-port circuit topology.

Topology Selection: For common-used circuit functional units, like amplifiers, designers may not need to design from the beginning. Instead, it is possible to choose from a fixed set of available alternatives. It is a much simpler problem than designing from scratch. Early in 1996, Orzáez et al., Silgado et al. put forward a fuzzy-logic based topology selection tool called FASY. They use fuzzy logic to describe relationships between specifications (e.g., DC gain) and alternatives and use backpropagation to train the optimizer. More recent research uses CNN as the classifier. They train CNN with circuit specifications as the inputs and the topology indexes as the labels. The main problem with the topology selection methods is that the data collection and the training procedure are time-consuming. Therefore, topology selection is efficient only when repetitive designs are needed such that a trained model can be reused.

Topological Feature Extraction: One challenge of topology design automation is to make algorithms learn the complex relationships between components. To make these relationships more understandable, researchers focus on defining and extracting features from circuit topology. Li et al. present algorithms for both supervised feature extraction and unsupervised learning of new connections between known building blocks. The algorithms are also designed to find hierarchical structures, isolate generic templates (patterns), and recognize overlaps among structures. Symmetry constraint are one of the most essential topological features in circuits. Liu et al. propose a spectral analysis method to detect system symmetry with graph similarity. With a graph representation of circuits, their method is capable of handling passive devices as well. Kunal et al. propose a GNN-based methodology for automated generation of symmetry constraints. It can hierarchically detect symmetry constraints in multiple levels and works well in a variety of circuit designs.

Topology Generation: The previously discussed investigations are not directly generating a topology. A recent study makes the first attempt to generate circuit topology for given specifications. Their focus is limited to two-port circuits. They utilize an RNN and Hypernetwork to solve the topology generation problem and report better performance than the traditional methods when the inductor circuit length $n \geq 4$.

3.3 Analog and Mixed-Signal Circuit Optimization

AMS circuit designers first, decide the circuit topology then optimize the corresponding design parameters e.g. component sizing, and finally generate the layout. A significant amount of effort has been put into optimizing components sizing because of the large effect that it has on a circuit's performance and power usage.

1) Problem formulation: The AMS design circuit sizing optimization problem can be formulated as follows.

$$\text{minimize } f1(x), \ldots, fm(x)$$

$$\text{subject to: } ci(x) < 0, \forall i \in \{1, \ldots, Nc\}, \qquad (2)$$

where $x \in R^d$ denotes d design variables e.g. width and length of MOS transistors, and $fl(x)\ \forall l \in \{1, \ldots, m\}$ are the Figureure of Merit (FOM) of the AMS circuits. Each FOM can be deterministic or noisy depending on the design specification. Nc represents the total number of constraints and $ci(x)$ corresponds to the i-th constraint e.g. $xj \in [p - j, p + j]$. When $m \neq 1$, usually there is no best design as objectives can be conflicting and it is unlikely to optimize all of them simultaneously. The goal then would be concluding the best trade-off between a set of solutions.

2) Classical approaches: The classical AMS circuit optimization approaches can be classified into the model-based (e.g. geometric programming, SVM, ANN, Gaussian Process (GP), etc. and simulation-based methods (e.g. Simulated Annealing (SA), PSO, EA, and gradient-based local search with Multiple Starting Points (MSP) which has a better convergence rate than the others. Analytical manually derived or regression models with simulated data are leveraged to build global models of the FOM in the model-based approaches, while the optimization is driven directly by the circuit simulations for simulation-based methods. The model reusability and low computational cost, especially in the case of using Electro-Magnetic (EM) components, are the main advantages of model-based approaches. However, the accuracy of these models are not usually high as the number of design parameters are usually large and object and constraint functions are highly nonlinear.

Recently, to combat these drawbacks, hybrid methods that combine both models have been proposed for analog circuits as well as mm-wave and Radio Frequency (RF) circuits. These hybrid methods run simulations during the optimization procedure to update online models gradually, instead of using pre-built offline models. Initially, the model is constructed by the data gathered from random sampling and it guides the selection of the next point towards more optimized performance.

A. Bayesian-Based Approaches

The Bayesian optimization method after initial sampling, constructs the probabilistic surrogate model of the objective function, which is refined incrementally based on the new data that optimizes the acquisition function e.g. Expected Improvement (EI), Lower Confidence Bound (LCB), Probability of Improvement (PI). The model uncertainty is evaluated to balance the exploration, i.e. the next point tends to explore the unknown regions with high uncertainty in the surrogate model, and exploitation, i.e. the next point tends to be the optimum with high probability of prediction by the surrogate model, during the optimization [94]. The de facto surrogate model used in BO is the Gaussian process model which reduces the required number of circuit simulations and has closed-forms for both its model prediction and model uncertainty. In a GP, letting $f : X \to R$ be a black-box indicating the performance function, for any finite samples $\{x_1 \dots \dots x_n\} \epsilon\, x^n$, the vector $[f(x_1) \dots \dots f(x_n)\,]^{\mathrm{T}}$ follows joint multivariate Gaussian distribution i.e.

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}$$

$\sim\ N(\mu, K)$, where μ is an n × 1 mean vector and K is an $n \times n$ covariance matrix. The GP can be fully characterized by its mean function, m(x), and its co-variance function $k(x1, x2)$ i.e. $\mu i = m(xi), Kij = k(xi\,, xj\,)\, for\ all\ i, j \in \{1, \dots, n\}. Kij$ denotes the i–th row j–th column covariance matrix element.

Let training set $D = \{X, y\}, where\ X = \{x_1 \dots \dots . . x_n\}. y = \{f(x_1) \dots \dots f(x_n)\}, m = [m(x_1) \dots \dots m(x_N)]^{\mathrm{T}}$, given a new point, $x *$,

$$\begin{cases} k(x^*, X) = [k(x^*, x_1), \dots, k(x^*, x_N)], \\ k(X, x^*) = k(x^*, X)^T. \end{cases}$$

Moreover, the function value, $y\ * = \ f(x\ *)$, and y follow the joint Gaussian distribution. The mean $\mu y * |y$ and the variance $\sigma\ 2\ y * |y$ can be viewed as the prediction, and the confidence of the prediction respectively.

$$\begin{cases} \mu_{y^*|y} = m(x^*) + k(x^*, X)K_N^{-1}(y - m), \\ \sigma^2_{y^*|y} = k(x^*, x^*) - k(x^*, X)K_N^{-1}k(X, x^*). \end{cases} \tag{2}$$

A Gaussian noise $\epsilon \sim \mathrm{N}(0, \sigma_n^2)$ should be added to $f(x)$ in order to avoid overfitting of the simulation model where σ 2 n is the variance of the Gaussian noise. If we take noise into consideration, Equation (2) can be rewritten as:

$$\begin{cases} \mu_{y^*|y} = m(x^*) + k(x^*, X)(K_N + \sigma_n^2 I)^{-1}(y - m), \\ \sigma^2_{y^*|y} = k(x^*, x^*) - k(x^*, X)(K_N + \sigma_n^2 I)^{-1}k(X, x^*). \end{cases}$$

Acquisition functions optimally balance the exploration and exploitation. The improvement of y can be formulated as

$$I(y, \tau) = \begin{cases} \tau - y & y < \tau \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where τ is the minimal value of found $f(x)$. In the GP model, the expectation of improvement can be written as:

$$\begin{aligned} EI(x) &= \mathbb{E}[I(y, \tau)] \\ &= \int_{-\infty}^{+\infty} p(y|D, \theta)dy \\ &= (\tau - \mu(x))\Phi(\tfrac{\tau - \mu(x)}{\sigma(x)}) + \sigma(x)\phi(\tfrac{\tau - \mu(x)}{\sigma(x)}), \end{aligned}$$

where D denotes the given training set, θ is the vector of hyper-parameters, Φ(.) is the Cumulative Distribution Function (CDF) of the standard normal distribution, and φ(.) is the Probability Distribution Function (PDF) of the standard normal distribution.

B. Multiple Starting Point Approach

MSP optimization generally begins by randomly selecting large amounts of starting points that cover the design space. This first step is called the global phase. Afterward, in the local phase, an efficient local search such as Sequential Quadratic Programming (SQP) is applied to each starting point found in the global phase. MSP can successfully approximate the global optimum by choosing the best local optimum. Yang et al. improve the inherent large MSP computation time by proposing smart MSP. A heuristic-biased starting point selection is implemented in the global phase to find the starting points that are likely to be close to optimums instead of using randomly selected points. Furthermore, an intermediate phase is added where sparse regression is applied to predict the circuit performances around the starting points. Moreover, in the local phase, model optimums are used as the starting points for SQP and coupled with Probabilistic-TABU (P-TABU) approach to improve the efficiency of local searches. For evaluating their approach, they optimize a variety of circuits such as an amplifier, a charge pump, a Voltage Control Oscillator (VCO), etc. in different technology nodes.

C. Neural Network Based Approaches

In the conventional circuit sizing approaches, designers or EDA tools try to find the circuit parameters in an iterative loop while using a simulator in each iteration to evaluate the design. In order to avoid time-consuming simulations, several studies have used ANN models to replace and complement the SPICE simulator with an approximated model. They implement the simulator in later stages just to maintain the accuracy. Even though using an ANN instead of a simulator saves time considerably, the training set of such an ANN model should cover the entire design space which consumes a lot of resources. On the other hand, Lourenc¸o et al., as it is illustrated in Figure. 7, train an ANN model to directly size and optimize the circuit for given specifications instead of invoking the simulator many times in an iterative loop to achieve the optimized sizing. Although they use circuit sizing solutions from previous optimizations as the training set, they are able to optimize circuit sizes for specifications outside the ones in the training dataset.

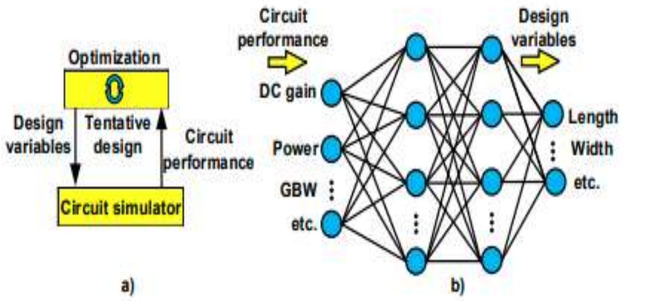

Figure. 7 The different methods of AMS circuit sizing: a) Conventional optimization-based sizing (inverse approach) b) Artificial Neural Networks (ANNs) (direct approach)

To have a better NN model, Lourenc¸o et al. take into account the following considerations for selecting the ANN hyperparameters e.g. number of layers, number of nodes in each layer, etc. First, in order to have a rich encoding, the number of nodes in each layer is increased in the first layers then is decreased toward the output layer. Second, to find a less complicated model, it is better to design it for having a low error model during the training and compensate the overfitting using L2 regularization. Third, for achieving a model which gives the best cross-validation score, it is necessary to explore the hyperparameter space. Two general approaches are grid search which considers all combinations of the specified hyperparameters and random search which takes samples from a hyperparameter space with a specified distribution. In their ANN approach, the circuit specifications are given to input nodes and output nodes determine the circuit sizing.

Although it successfully predicts the analog IC sizing for the target performance, its cost for providing training data is still too high, which makes it impractical. Pan et al. propose a performance exploration method to tackle the major evolutionary-based approaches shortcoming, low accuracy, while significantly speeding up the runtime. For this purpose, they apply Bayesian regression to better model the device variables and use an SVM to increase the model performance space. In other words, they replace the performance evaluation process with SVM predictions. However, they leverage supervised learning which requires a large dataset.

This dataset is difficult to obtain because most analog IPs are not available to the public. In order to evaluate their approach, they compare the specifications of an operational amplifier and a radio-frequency distributed amplifier e.g. voltage gain, DC power, bandwidth, etc. optimized by their method with. Wang et al. leverage RL which learns to efficiently optimize the transistor parameters automatically without any prior knowledge about circuit design rules. In each iteration, after observations, including monitoring DC operating points, AC magnitude, and phase responses, they change the circuit parameters based on the simulator results. Then, a reward would be received to optimize the desired FOM. They compare their approach for optimizing an amplifier and meet the specifications constraints such as the specified bandwidth, gain, power, area, etc. with other methods such as a human expert, etc. The results show that the other compared methods either are not able to meet the constraints at the same runtime or they have a less efficient design in comparison to. Hakhamaneshi et al. find the optimal size of analog circuits by predicting the feasibility of a design using a DNN classifier. They propose a framework that in each iteration utilizes an evolutionary algorithm to create new candidate designs and leverage a DNN classifier to recognize “bad” new offspring by comparing with a reference design that is chosen from the previous “good” population. By passing just these high quality samples to a layout-aware design methodology such as BAG, it achieves more than a 200×runtime improvement in comparison to just an evaluation method without the DNN classifier.

Table 6. Examples of AMS Circuit Sizing Optimization Methods

Reference	AI Method	Surrogate Model & Acquisition Function	Applications & Advantages	Test Results (# Design Variables)	Advantage
http://dx.doi.org/10.1109/TCSI.2017.2768826	BO	Online GP & WEI	Handling multi-objective optimizations and optimization constraints	3 − 36	Higher accuracy, fewer training numbers
https://proceedings.mlr.press/v80/lyu18a/lyu18a.pdf	BO	GP & Multi-functions	Multi-objective acquisition function selection, parallelized BO	10 − 12	Higher accuracy, fastest convergence rate
http://dx.doi.org/10.1109/TCAD.2017.2778061	BO	GP & EI	Yield optimization	6 − 24	Higher accuracy, fewer training numbers

http://dx.doi.org/10.23919/DATE.2019.8714788	BO & DNN	GP & WEI	Low computation complexity, handling optimization constraints	10 − 36	Higher accuracy, fewer training numbers
http://dx.doi.org/10.1145/3316781.3322468	EA & DNN	-	High sample efficiency	21	Much faster runtime
https://doi.org/10.1109/ASP-DAC58780.2024.10473920	MSP	-	Significantly fast	11 − 36	Fewer training numbers, faster runtime
https://doi.org/10.1145/3316781.3322467	BR & SVM	-	Short runtime	-	Higher accuracy, much faster runtime
https://doi.org/10.48550/arXiv.1812.02734	RL	-	High sample efficiency	-	Higher accuracy
https://www.princeton.edu/~nverma/VermaLabSite/Publications/2015/ZhangHuangWangVerma_CICC2015.pdf	SVM	-	Relaxing the circuit performance for an ADC	-	-

Abbreviation list: BO: Bayesian Optimization, GP: Gaussian Process, WEI: Weighted Expected Improvement, EI: Expected Improvement, DNN: Deep Neural Network, EA: Evolutionary Algorithms, MSP: Multiple Starting Points, BR: Bayesian Regression, RL: Reinforcement Learning, SVM: Support Vector Machine, ADC: Analog-to-Digital Converter.

3.4 Optimization techniques for analog CMOS and RF microwave circuits

In recent years, synthesizing and designing high performance analog and RF circuits and systems with advanced optimization tools have received increasing attention. These optimization techniques are performed for improving and meeting design specifications such as wide band frequency, high efficiency, and linear gain performance. This section briefly addresses precise studies of powerful and efficient optimization techniques that suppress experience and trial methods in designing.

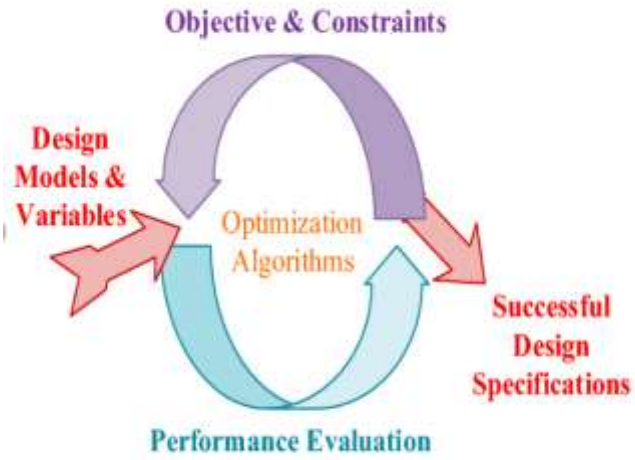

Figure 8. General structure of an iterative optimization loop for analog/ RF circuit designs

3.4.1 Functional Surrogate Modeling

Functional surrogate modeling techniques are functions for predicting the output responses of designs, and are generally applied for modeling the microwave components due to the fast simulation in high-level circuit designs. These techniques aim to solve the problems of multi-objective functions lead to finding the optimal responses. The basic of multi-objective functions are as (4):

$$\left\{ \begin{array}{c} \text{minimize}\, f(x);\ f(x) \in R^m \\ \text{subject to}\, g(x) \leq 0;\ g(x) \in R^k \\ x \in \Omega \end{array} \right\} \quad (4)$$

where $f(x)$ is a vector with m objective functions, g(x) is a vector with k constraints, and x is a vector with n design variables on the search space X. When $m > 1$; in this case multi-objective optimization can be applied.

A surrogate model-aware search mechanism (SMAS) for sizing design variables in mm-wave integrated circuits (ICs) is presentedin https://doi.org/10.1109/TCAD.2013.2284109 and https://doi.org/10.1109/DATE.2009.5090756 . This method is applied as it can provide an automated efficient optimization of IC designs for solving the problems of dependency on designer experiences and consumed designing time. As another example of using surrogate modeling, inverse surrogate model https://doi.org/10.1109/TMTT.2015.2490662 and fast electromagnetic-based (EM-based) parameter tuning approach for redesigning of miniaturized circularly polarized (CP) antennas are presented in. In this method, the geometry parameter scaling in low-cost EM-analysis with the lower operating frequency of the CP antenna are reported that are controlled in terms of both the impedance and the axial ratio bandwidth. EM-based surrogate model for generating the layout of RF IC blocks that is ready-to-fabricate, is reported in https://doi.org/10.1109/TCAD.2018.2834394 . In sizing optimization, it mixes EM-characterized integrated inductors with parasitic extraction from layout, results in reliable device designs with circuit layouts. figure 9 shows the optimization flow for the design of a circuit by using the surrogate model. For modeling microwave components, there are several functional surrogate modeling methods that some of them are: neural network (NN) technique https://doi.org/10.1109/TMTT.2015.2504099 , support vector machine http://dx.doi.org/10.1109/TMAG.2007.892480 , and polynomial-based surrogate modeling http://dx.doi.org/10.1109/TMTT.2016.2623902 . In the following, we will describe the NN method briefly that is applied in various circuit designs.

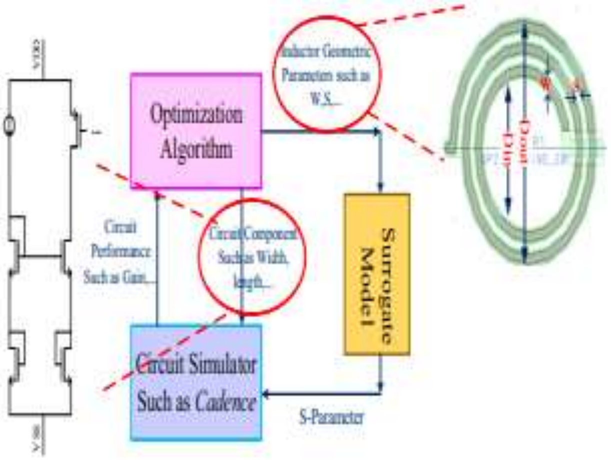
VDD
Optimization Algorithm
Inductor Geometric Parameters such as W,S,...
Circuit Performance Such as Gain,...
Circuit Component Such as Width, length,...
Surrogate Model
Dout
Din
Circuit Simulator Such as Cadence
S-Parameter
VSS

Figure 9. Circuit Sizing Using Functional Surrogate Modeling

1. Neural network technique Recently, automated circuit designs https://doi.org/10.1016/j.mejo.2018.11.015 and https://doi.org/10.1109/ICECS.2016.7841150 have attracted more attention as manual attempts become increasingly hard due to the scaling of IC technology. Machine learning (ML) is a subset of artificial intelligence that learns from experiences and builds a mathematical model through sampling data and provides an automated optimization platform. For getting a good view of ML, in https://doi.org/10.1109/TCAD.2012.2187207 , the memetic machine learning-based differential evolution (MMLDE) method is presented for the design of a millimeter-wave (mm-wave) frequency RF amplifier. The optimization problem of the system is solved using a surrogate model operating online with applying the evolutionary computation to the resulting system https://doi.org/10.1109/TCAD.2007.907284 and https://doi.org/10.1109/TCAD.2011.2162067 that the method is formed based on the present data in the optimization process. This method presents well efficiency optimization for RF IC designs; however, it provides much lower computational cost. As Figure. 10 shows ML divides into three groups of supervised, unsupervised, and reinforcement learning. Each group of learning can be modeled by using the artificial neural network (ANN) http://dx.doi.org/10.1109/TSP49548.2020.9163468 , i.e., a shallow neural network with one or two hidden layers, or the deep neural network (DNN) https://doi.org/10.1109/TCSI.2020.3008947 , i.e., a network with more than two hidden layers.

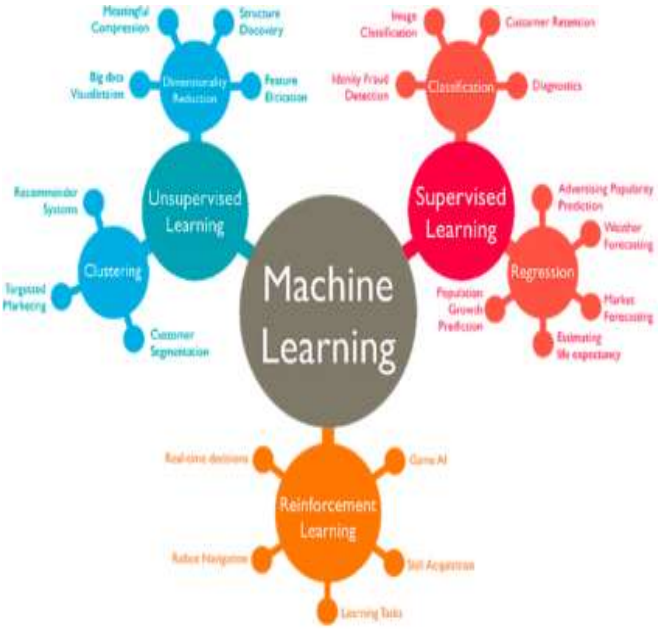

Figure 10. Groups of Machine Learning Algorithms

ANN/DNN networks can model the non-linear behavior of circuits by using input and output data that extracted from designs and provide automated environments for predicting design parameters that meet the design specifications. Figure 4 shows the utilization of neural networks in predicting the required data and estimating targeted performances. The performance of the circuit is predicted by replacing the trained NN instead of a simulated system in the optimization loop, thus reducing the design time substantially.

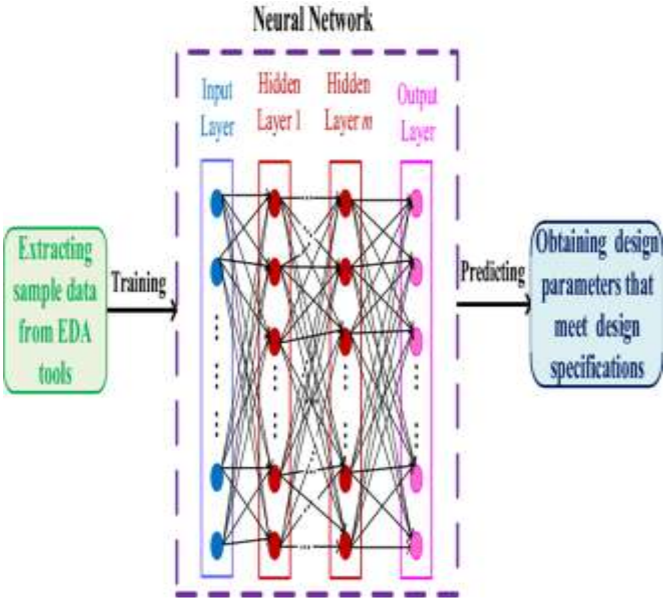
Neural Network
Input Layer
Hidden Layer 1
Hidden Layer m
Output Layer
Extracting sample data from EDA tools
Training
Predicting
Obtaining design parameters that meet design specifications

Figure 11. Neural network method for optimizing and modeling circuits. $m \leq 2$ and $m > 2$ represent ANN and DNN networks

The execution time issue is faster in this case as the new performance and output result are predicted from the previous data without performing simulation https://doi.org/10.1109/SMACD.2019.8795293 . Neural networks are very effective tools used in various design steps of analog and RF circuits https://doi.org/10.1109/TIE.2008.2003319 and https://doi.org/10.1109/IJCNN.2017.7966126 and http://dx.doi.org/10.23919/ICACT48636.2020.9061564 . They are as a 'black-box' modeling and are exerted in modeling of on-chip inductors https://doi.org/10.1109/IJCNN.2005.1556269 , semiconductor devices https://doi.org/10.1007/s10825-017-0984-9 , conventional analog circuit building blocks https://doi.org/10.1007/978-3-642-39162-0 and https://doi.org/10.1109/SOCPAR.2010.5686736 , analog IC sizing https://doi.org/10.1109/SMACD.2019.8795293 and https://doi.org/10.1007/978-3-030-35743-6_4 , and in crucial RF circuit blocks such as power amplifiers https://doi.org/10.1109/MIXDES.2007.4286202 , RF front-end receivers https://doi.org/10.1007/s11277-020-07162-z , low noise amplifiers https://doi.org/10.3103/S1060992X16040111 , voltage-controlled oscillators https://doi.org/10.1109/ACCESS.2019.2905136 and https://doi.org/10.1109/MELCON.2004.1346871 , and multiple-input multiple-output (MIMO) systems https://doi.org/10.1109/LWC.2019.2944179 and https://doi.org/10.1002/jnm.2160 .

3.4.2 Co-simulation method and/or combination of EDA and CAD tools

Any analog/RF circuit consists of three design levels as: (i) system-level (Higher Level), (ii) circuit-level (Middle Level), and (iii) layout-level (Lower Level). Figure 12 demonstrates three design levels for power amplifier design for illustrating the definitions of levels. For simulating and analyzing, software tools such as ADS, NI AWR, Cadence, ANSYS HFSS, MATLAB, CST, and TCAD are frequently used at each design level by the designers. The design optimization algorithms of the software tools could not converge and provide satisfying results when the dimension of processed data is high, especially with multi-objective optimization. Reliability and speed of the optimization algorithms should be considered for large scaled designs, having several active components and using mixed signals.

To tackle these problems and have reliable designs, CAD and EDA tools can be used together that collaboration of tools can compensate the deficits of each other https://doi.org/10.1109/SMACD.2016.7520737 and https://doi.org/10.1109/NEMO.2015.7415067 . Reliability optimization for power amplifier design is done in https://doi.org/10.1109/COMCAS.2017.8244816 based on the electric and thermal co-simulation method. In this method, thermal outputs are returned back to the circuit design and reliability with accuracy results are considered. Following describes in detail the collaboration of other different tools for optimizing various circuits.

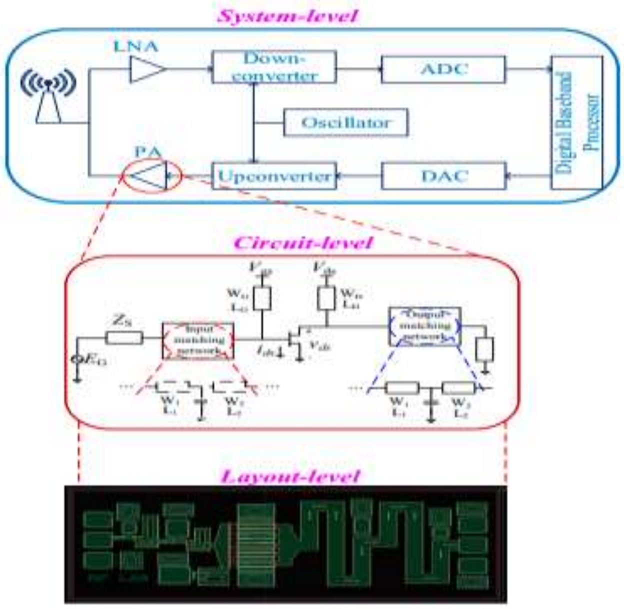

Figure 12. Various design levels include system-level, circuit-level, and layout-level for circuit designs

ADS with HFSS: The electromagnetic-based process for optimizing three-dimensional layout of GaN-HEMT power amplifier IC is presented in https://doi.org/10.1109/TEMC.2018.2820202 . The optimization is performed to reduce the electromagnetic compatibility /electromagnetic emission for the IC. ADS with MATLAB: With a combination of these tools, power amplifiers are optimized automatically lead to time-efficient designs https://doi.org/10.1109/MMS.2018.8611955 and https://doi.org/10.23919/ELECO47770.2019.8990407 . In these works, ADS tool is working in the background, and MATLAB manages automated optimizations. In https://doi.org/10.1109/MMS.2018.8611955 , the values of components are iterated (i.e.,decreasing/increasing) automatically and in https://doi.org/10.23919/ELECO47770.2019.8990407, an automated optimization process is performed for achieving suitable topology for matching networks by exerting the bottom-up optimization process. ADS with TCAD: For extracting effective nonlinear X-parameter model [19], in https://doi.org/10.1109/MWSYM.2019.8700869 a powerful collaboration between TCAD and ADS is presented results in designing amplifiers. ADS with SIMSIDES: MATLAB optimizer is embedded into the SIMSIDES (SIMulink-based SIgma-DElta Simulator) for designing sigma-delta modulators https://doi.org/10.1109/TCSII.2018.2820900 . By getting use of available optimizations in MATLAB, the current version of SIMSIDES http://imse-cnm.csic.es/simsides/ is improved results in a user-friendly interface that is able to choose the appropriate algorithm for maximizing the performance and also setting design variables easily.

IV. ML IN ANALOG CIRCUIT MODELING

SVMs and ANN-based approaches are commonly employed to obtain the functional models of analog circuits. SVMs are usually preferred in analog circuit modeling since they do not get easily stuck at local minima and suffer from the curse of dimensionality when the data points are determined considering the dimensions. In http://dx.doi.org/10.1109/ICECS.2004.1399700 , the authors propose the use of SVMs to model analog circuits. As a kernel, the authors choose Gaussian Radial Basis Functions. The regression method utilized is ²-SV regression. This modeling is applied to a Source Coupled FET Logic (SCFL) buffer, a resistive mixer, and a GaAs ring oscillator. The generated models are validated through SPICE simulations. SVMs are also the preferred method for modeling in http://dx.doi.org/10.1002/widm.1132 ; however, the aim is not to create a full mapping from the input space to the output space, but to identify infeasible regions and prune them. A committee of SVM classifiers is utilized to exclude a large portion of the entire design space, and only the feasibility region and its neighbors are sampled. http://dx.doi.org/10.1007/s12541-012-0096-1 The feasibility design space is defined by the so-called geometry constraints, which include not only device sizing constraints, but also constraints on voltage and current source values, functional constraints which are in terms of node voltages and branch currents, and performance constraints. An active learning approach is employed to train the classifier, where very few samples are taken from the large infeasible space, and most of them are concentrated around the boundaries. This is achieved by checking sample candidates against a committee of classifiers and discarding those candidates rejected by all. The classifier is tested on two examples, an operational trans conductance amplifier (OTA) and a mixer. https://doi.org/10.3390/electronics12183833

Table 7. Summary of modeling of Analog/RF device and systems with ML techniques

Reference	Application-Device	Method(s)	Contributions
http://dx.doi.org/10.1109/ICECS.2004.1399700	Analog Circuits-GaAs transistor	SVMs (²-SV regression)	Robust and accurate modeling of GaAs transistors and circuits
http://dx.doi.org/10.1109/DAC.2003.1219160	Analog Circuits-CMOS	SVMs	Efficient active learning scheme for feasible design space selection
http://dx.doi.org/10.1109/SMACD.2019.8795295	AMS circuits-CMOS	ANN (TDNN)	Robust modeling of power consumption for AMS circuits
http://dx.doi.org/10.1109/TCAD.2010.2043759	Analog-n/d	ANN (Back propagation)	A generic modeling of power consumption for heterogeneous systems
http://dx.doi.org/10.1109/TMTT.2003.809179	RF-microwave components and MESFET	ANN (several)	Review of ANN-based CAD for microwave designs
http://dx.doi.org/10.1002/1099-047X(200101)11:1%3C4::AID-MMCE2%3E3.0.CO;2-I	RF-microwave components, HMT and MESFETs	ANN (several)	Review of model development and nonlinear modeling of microwave devices
http://dx.doi.org/10.1109/TMTT.2003.820897	RF-CPW components	ANN (EM based)	Efficient modeling of CPW components for accurate performance estimations
https://oa.mg/work/2108514954	RF-UC-PBG rectangular waveguide	ANN (RBF-MLP)	Efficient modeling of RF devices for nonlinear microwave applications
http://dx.doi.org/10.1109/TMTT.2022.3197751	RF-MESFET	ANN (WNN-MLP)	Faster design of large signal hard-nonlinear power transistors and circuits

ANN-based modeling approaches have become more pronounced in recent years. ANN can also be used to improve the accuracy of the behavioral models of transistor level design, where some specifications such as, power consumption, area overhead, etc. are not taken into account during the behavioral simulations of the systems. http://dx.doi.org/10.1109/SMACD.2019.8795295 presents a novel methodology for ANN aided inclusion of power consumption information of circuits to their purely functional models of AMS blocks. Due to the nature of the problem, an improved version of the Multilayer Perceptron (MLP) approach, which is called time delay neural network (TDNN) shown in Figure. 13, is utilized in this study. In this approach, the inputs pass through a delay cell and are given as the inputs of the network in order to capture the temporal changes. The flow of the proposed approach is as follows. First, the behavioral model (Verilog-AMS) of the circuit is constructed. Meanwhile, transistor level simulations are performed to extract signal traces for power calculation. Then, the TDNN is trained and the power consumption model is obtained. Once the model is obtained, it is translated into the behavioral model compatible with the circuit simulators. Finally, the first behavioral model is integrated with the power model. As a case study, a low power relaxation oscillator is designed and simulated both at transistor level and with the augmented functional model. According to the reported results, the simulation time decreases to 12 s from 168 s while the estimation error in energy is only 2.7%.

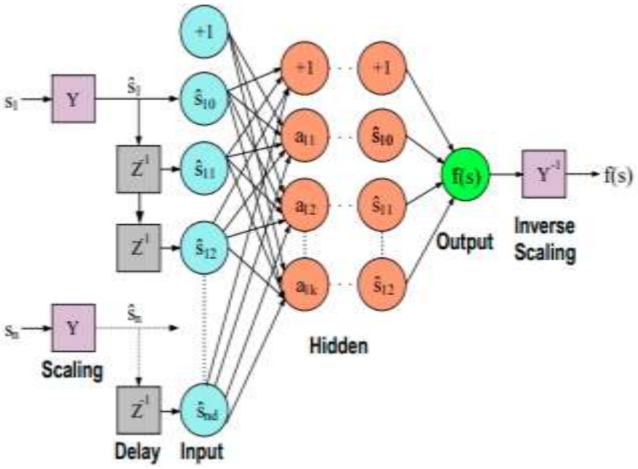

Figure 13. TDNN Delay Neural Network Model

A different application of ANN-based modeling is presented in http://dx.doi.org/10.1109/TCAD.2010.2043759 , where power consumption of analog circuits is modeled and then estimated via empirical-based ANN rather than achieving performances through the input parameters. The idea behind this study is to estimate the mathematical description of the power consumption as a function of varied input parameters of any analog circuit using neural networks. The proposed approach is generic and even suitable for heterogeneous systems. https://doi.org/10.1016/S1389-1286(03)00254-8 Moreover, one can perform online power consumption estimations via the proposed strategy. First, analog circuit power measurements are performed via a measurement set-up including a PC for generating different input patterns and saving the power data. Second, the obtained data is used to train the ANN to obtain a continuous mathematical function of the power consumption. The neural networks include three levels: one input, one hidden, and one output layer. The activation functions for the hidden layer and the output layer are sigmoid and linear, respectively. A backpropagation-based training (Levenberg-Marquardt) is employed. Once the power model is obtained, it is combined with a data flow-based generic functional model of the circuit. https://doi.org/10.1049/wss2.12052 Hence, both circuit performances and the instantaneous power consumption are obtained, which makes possible to estimate circuit performance without performing any empirical measurements. By combining this framework with digital power consumption estimation techniques, the power consumption of heterogeneous systems can be predicted. A wireless sensor system is provided as the case study, where the main focus is to estimate the power consumption of analog parts (a temperature sensor, an amplifier, an analog to 7 digital converters, and a wireless transceiver.) https://doi.org/10.1016/j.heliyon.2024.e40415 According to the results, the maximum and the average estimation errors are 3.06% and 1.53%, respectively.

V. ML IN RF CIRCUIT MODELING

Neural networks have been used for RF and microwave modeling and design, where ANN-based passive/active component/circuit models are then employed at higher design levels. Thus, an accurate response of the whole system can be obtained within shorter durations compared to the expensive conventional approaches. In http://dx.doi.org/10.1109/TMTT.2003.809179 , ANN for RF/microwave modeling and design is discussed from theory to practice. The authors state that neural networks are attractive alternatives to conventional methods such as numerical modeling methods, which could be computationally expensive, or analytical methods which could be difficult to obtain for new devices, or empirical modeling solutions whose range and accuracy may be limited. They provide examples where neural networks are used to model signal propagation delays of a VLSI interconnect network in printed circuit boards (PCBs), coplanar waveguide (CPW) discontinuities, and MOSFETs, all from previous works in the literature. Finally, they illustrate the use of CPW models to optimize microwave circuits. The same authors present a detailed study on modelling issues and ANN-based nonlinear modelling techniques in http://dx.doi.org/10.1002/1099-047X(200101)11:1%3C4::AID-MMCE2%3E3.0.CO;2-I including small/large signal modeling of transistors and dynamic recurrent neural network (RNN) modeling of circuits. Practical microwave examples are used to illustrate the reviewed modeling techniques.

Another method of modeling CPW circuit components by ANN is based on electromagnetic (EM) simulations http://dx.doi.org/10.1109/TMTT.2003.820897 . CPW transmission lines (frequency dependent Zo and ²r e), 900 bends, short-circuit stubs, open-circuit stubs, step-in width discontinuities, and symmetric T-junctions are individually modeled through EM-based ANN. To train the models, a number of EM simulations that exhibit meaningful input/output relationships, which directly affect the model accuracy. A multilayer feedforward ANN consisting of three layers (one input, one hidden, and one output), which utilizes the error-backpropagation learning algorithm, is used. The developed models are then employed to design a CPW folded double-stub filter and a 50- 3-dB power-divider circuit, without performing expensive EM simulations. The proposed framework is also available for the other component of microwave/RF design.

Since EM-based ANN approaches need a relatively long training phase for accurate modeling, the efficiency can be low. https://doi.org/10.1002/9781119847717.ch5 presents a solution for modeling of RF devices with Radial Basis Function(RBF)/MLP modular structure, where the efficient Resilient Backpropagation (Rprop) algorithm is used during the training phase. The authors use a well-known plan, "divide and conquer", where the propose framework is provided in Figure. 6. The complicated design problem is divided in sub-problems, distributed over the neural networks of the modular structure.

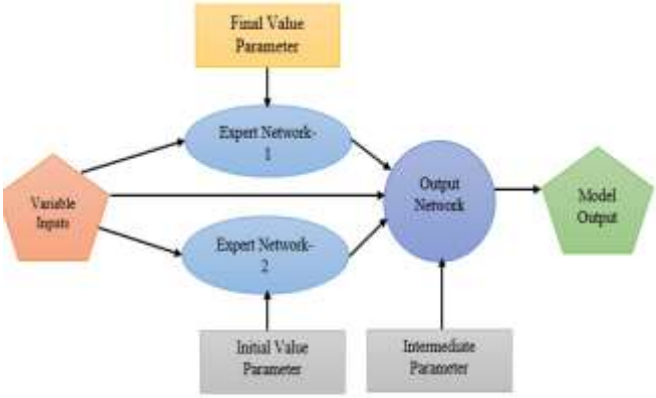

Figure 14. The proposed framework in http://dx.doi.org/10.1080/00207217.2014.989924

ANN. The RBF/MLP structure modules are organized in order to take advantage of the local and global approximation characteristics of the RBF and MLP neural networks, where the RBF network is a local approach while the MLP network is a global approach and acts as an output network, since it improves the generalization capacity of the modular structure. The uniplanar compact-photonic bandgap (UC-PBG) rectangular waveguide and a patch antenna with PBG substrate are used to demonstrate the developed approach. Compared to the single usage of RBF and MLP, the combination of them (modular model) presents a major generalization capacity, which is independent of the number of hidden neurons. Wavelet neural networks are chosen over simple MLP and Gaussian radial basis (GRB) function networks In https://openurl.ebsco.com/contentitem/doi:10.1002/(sici)1099-047x(199905)9:3%3C198::aid-mmce6%3E3.0.co;2-a?sid=ebsco:plink:crawler&id=ebsco:doi:10.1002/(sici)1099-047x(199905)9:3%3C198::aid-mmce6%3E3.0.co;2-a . The first example is a transistor modeling example, where 10 neural networks are used as shown in Figure. 15.

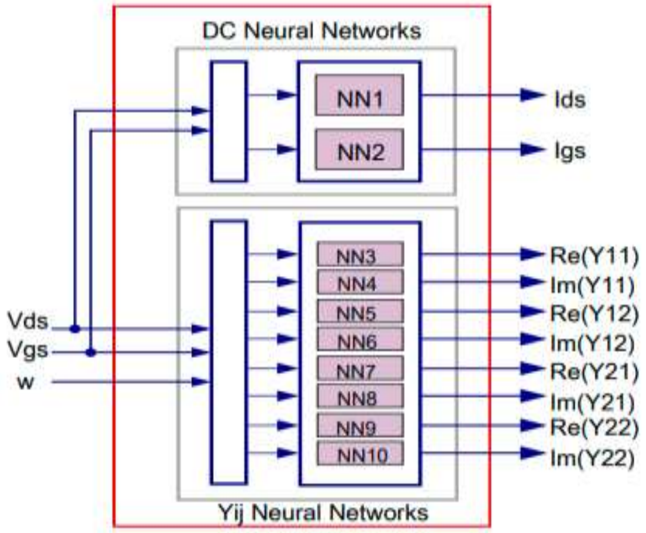

Figure 15. Volterra-ANN device model

Two of them utilize Vd s and Vg s to obtain Ids and Igs. The remaining 8 use Vd s, Vg s, and to yield real and imaginary values for Yij . The total number of parameters is 25 each for the first two and 76 each for the remaining 8. The 10 neural networks are trained separately, on 350 measurement points for DC characteristics and 7000 measurement points for Y-parameters. The results on test points agree perfectly with lumped equivalent circuits. For the circuit modeling example, 4 neural networks were utilized. The 5 inputs are ω and the real and imaginary parts of the input and output voltages, whereas the outputs are the real and imaginary parts of the input and output currents. This type of modeling allows the model to take into account input and output loading. Learning was performed on 2625 measurement points and results on new data were encouraging. The use of more generic neural network-based models could overcome the problems associated with lumped equivalent electrical circuit models, which are the most common models in use. These models offer the advantage of being computationally efficient and accurate, but at the expense of very complex model parameter extraction carried through numerical fitting and optimization as well as the requirement to an accurate circuit structure.

5.1 Machine Learning for IC Circuit Synthesis

Conventionally, circuit synthesis is described as an automatic process in order to determine the dimensions of the devices, such that the resultant circuit meet a given target specification on a given technology node. Considering the type of evaluation, simulation-based approaches is the most prevalent ones in terms of accuracy. However, the cost of SPICE-based circuit synthesis may be expensive in terms of computation time due to the need of running large number simulations (ten and even hundreds of thousands) to achieve the targeted performances. Hereby, ML-based synthesis approaches have become popular to overcome this time efficiency problem. The idea behind employing ML in circuit synthesis is to replacement of the simulations by the functional model(s) generated via ML techniques; thus, the excessive number of simulations can be avoided during the synthesis process. A summary of reviewed papers related to ML-based IC synthesis applications is provided in Table 2.

Table 7. Summary of ML-Based IC Circuit Synthesis Applications

Reference	Application	Method(s)	Contribution
http://dx.doi.org/10.1080/00207217.2014.989924	Analog Circuit Synthesis	ANN (GRP+MLP)	Technology independent sizing of analog building blocks
http://dx.doi.org/10.11591/ijece.v6i1.8700	Analog Circuit Synthesis	ANN (MLP)	Automatic generation of training dataset for analog circuit sizing
http://dx.doi.org/10.1109/SOCPAR.2010.5686736	Analog Circuit Synthesis	ANN	Generation of better FOMs for Op-Amps via ANN based circuit synthesis
https://doi.ieeecomputersociety.org/10.1109/TVLSI.2024.3452032	Analog Circuit Synthesis	DL+RELU	Efficient multiple performance estimation of Op-Amps with DL based models
http://dx.doi.org/10.1109/SMACD.2018.8434896	Analog Circuit Synthesis	ANN	Examining the effect of ANN hyperparameters on analog circuit synthesis
http://dx.doi.org/10.1109/ICECS.2014.7050096	RF Circuit Synthesis	GA+ANN(MLP)	Efficient synthesis of RF circuits via GA assisted ANN
http://dx.doi.org/10.1109/SMACD.2019.8795282	Analog Circuit Synthesis	Polynomial Regression + ANN	Generation of reusable POFs for analog circuit design
https://users.ece.cmu.edu/~xinli/papers/2016_TCAD_pof.pdf	Performance Space Exploration	ANN based text mining +Sparse regression	A global performance space search on the Internet via knowledge harvesting
https://arxiv.org/pdf/2009.13772 and https://dl.acm.org/doi/abs/10.5555/3539845.3540117	Analog Circuit Synthesis	RL (L2DC)	Efficient sizing of analog circuits (25x faster than hand design)
https://doi.org/10.48550/arXiv.2001.01808	Analog Circuit Synthesis	Deep RL	Efficient layout parasitics-aware circuit synthesis (40x faster than GA)

VI. ML IN ANALOG IC FAULT TESTING AND DIAGNOSIS

Specification testing and fault diagnosis are of the utmost importance for robust circuits and systems. Analog circuit testability analysis is significantly more complicated than its digital counterpart. The main culprits are the diversity of analog circuits with both linear and nonlinear characteristics and a multitude of performance metrics that create barriers to a standard definition of fault models. Fault diagnosis for electronics-rich analog systems with industrial-application is usually accomplished by monitoring the deviation of output signals in voltage or current caused by the inevitable degradation of one or more of its components. The degradation arises not only from inherent circuit mechanisms but also from improper technician operation or environmental changes, for example.

Researchers in the area of analog IC testing since long turned to ML algorithms for the automation of analog specifications testing and fault identification http://dx.doi.org/10.1109/TCAD.2015.2504329 . Table 8 summarizes the different ML techniques for IC fault testing and diagnosis that are overview within this section. In http://dx.doi.org/10.1109/DATE.2010.5457099 a fault-model-based diagnosis for analog ICs was proposed. The method is based

Table 8. Summary of the ML applications for Analog IC fault testing, diagnosis and calibration.

Reference	Application	Method(s)	Contributions
http://dx.doi.org/10.1109/DATE.2010.5457099 and https://hal.science/hal-04023648/document	Fault Diagnosis	SVM	A defect filter identifies hard and soft faults, and, for the soft faults, inverse regression is used to locate the fault cause

http://dx.doi.org/10.1109/TIM.2007.904549	Fault Diagnosis	ANN	Wavelet and PCA to reduce dimensionality of features
http://dx.doi.org/10.1109/TIM.2018.2836058	Fault Diagnosis	ANN	Dictionary and PCA reduce dimensionality of features
https://doi.org/10.1109/81.974884	Fault Diagnosis	Fisher DT	LDA to improve class separability while compressing the feature space
http://dx.doi.org/10.1109/ACCESS.2018.2888950	Fault Diagnosis	Naive Bayes	Wavelet followed by kLDA
https://repository.um.edu.mo/handle/10692/108009	Fault Diagnosis	DBN	End-to-end learning simplifies the feature engineering
https://doi.org/10.1016/j.measurement.2018.02.044	Fault Diagnosis	DBN	End-to-end with integrated random sampling for data gathering
http://dx.doi.org/10.1007/s10836-014-5454-8	Remaining Useful Performance	Kernel RVM	PSO is used to train a RVM that predicts the trajectories of the circuits health and predicts the remaining useful performance
https://doi.org/10.1109/tvlsi.2009.2017196	Test Set Compression	ONN	NSGA optimization to select the smallest set of features sufficient to diagnose the CUT, resulting in a cheaper test procedure
http://dx.doi.org/10.1109/TCSI.2016.2598184	One-Shot Calibration	ANN	Post-fabrication calibration to counter performance deviation due to fabrication in a single calibration step
https://scholars.duke.edu/publication/1290533	Post-Layout Modeling	BMF	Uses cheap pre-silicon simulation data, together with a small dataset of fabricated circuits for efficient post silicon modeling

on an ML-based defect filter https://hal.science/hal-04023648/document that distinguishes failing devices due to hard faults, i.e., completely malfunction, or soft faults, i.e., failing due to parametric deviations. Two types of diagnosis are handled based on the decision of the defect filter, and then an SVM-based multi-class ML classifier is used to identify which catastrophic fault has occurred, and, inverse regression functions to localize and identify the soft faults. This approach was demonstrated on an RF LNA. In http://dx.doi.org/10.1007/s10836-014-5454-8 , a sparse relevance vector machine [84] with Gaussian and polynomial kernels is used for fault prognostic and remains useful performance estimation. The approach uses AC voltage values over time as features to estimate the health degree of the circuit. The authors define this health degree as the cosine distance between the measured features and those at nominal value, and its value decreases from 1 for non-fault circuits as the circuit's elements degrade. The sparse kernel coefficients are obtained by minimizing the MSE using particle swarm optimization (PSO). https://doi.org/10.1155/2015/931256 Experiments with a Sallen–Key band pass filter, leapfrog filter, and nonlinear rectifier circuit showed that the methodology was able to accurately estimate the trajectories of the health degrees of the most relevant devices and accurately predict the remaining useful performance of the circuit.

CONCLUSION

1. The integration of hybrid artificial intelligence (AI) systems with machine learning (ML) algorithms for circuit-level optimization has revolutionized electronic design, particularly in analog, RF, and mixed-signal circuits.
2. ML algorithms such as support vector machines (SVMs), artificial neural networks (ANNs), and reinforcement learning (RL) have proven effective in automating and optimizing the design process, significantly expediting workflows.
3. The combination of AI methodologies with conventional circuit design tools addresses the rising complexity and performance demands in modern integrated circuits (ICs).
4. Hybrid AI systems excel in optimizing critical design metrics such as power consumption, area, and performance, offering solutions that outperform traditional methods.
5. Recent advancements in AI-driven tools have improved circuit synthesis, fault diagnosis, and testability analysis, showcasing the versatility of hybrid systems.
6. Despite their potential, these systems face challenges, including high computational demands for model training and the need for effective data integration techniques.
7. The increasing demand for high-performance, energy-efficient circuits underscores the expanding role of AI in circuit-level optimization.
8. The convergence of AI and electronic design automation (EDA) tools opens new avenues for innovation in semiconductor research.
9. Research will likely focus on improving the interpretability, efficiency, and scalability of hybrid AI systems, ensuring broader applicability and adoption across the industry.
10. Hybrid AI systems are poised to become integral to addressing the challenges of modern IC design, driving advancements and shaping the future of the electronics industry.