



**HAL**  
open science

## Review: Machine learning techniques in analog/RF integrated circuit design, synthesis, layout, and test

Engin Afacan, Nuno Lourenço, Ricardo Martins, Günhan Dündar

### ► To cite this version:

Engin Afacan, Nuno Lourenço, Ricardo Martins, Günhan Dündar. Review: Machine learning techniques in analog/RF integrated circuit design, synthesis, layout, and test. *Integration, the VLSI Journal*, 2021, 77, pp.113 - 130. 10.1016/j.vlsi.2020.11.006 . hal-03493756

**HAL Id: hal-03493756**

**<https://hal.science/hal-03493756>**

Submitted on 16 Dec 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

## Review: Machine Learning Techniques in Analog/RF Integrated Circuit Design, Synthesis, Layout, and Test

---

---

---

*Email address:* `engin.afacan@lip6.fr` (Engin Afacan<sup>1</sup>)

*Preprint submitted to Integration, The VLSI journal*

*October 11, 2020*

# Review: Machine Learning Techniques in Analog/RF Integrated Circuit Design, Synthesis, Layout, and Test

Engin Afacan<sup>1\*</sup>, Nuno Lourenço<sup>2</sup>, Ricardo Martins<sup>2</sup>, Günhan Dündar<sup>3</sup>

Laboratoire d'Informatique de Paris 6, Sorbonne University, Paris, France<sup>1</sup>

Instituto de Telecomunicações, Lisbon, Portugal<sup>2</sup>

Department of Electrical and Electronics Engineering, Boğaziçi University, Istanbul, Turkey<sup>3</sup>

## Abstract

Rapid developments in semiconductor technology have substantially increased the computational capability of computers. As a result of this and recent developments in theory, machine learning (ML) techniques have become attractive in many new applications. This trend has also inspired researchers working on integrated circuit (IC) design and optimization. ML-based design approaches have gained importance to challenge/aid conventional design methods since they can be employed at different design levels, from modeling to test, to learn any nonlinear input-output relationship of any analog and radio frequency (RF) device or circuit; thus, providing fast and accurate responses to the task that they have learned. Furthermore, employment of ML techniques in analog/RF electronic design automation (EDA) tools boosts the performance of such tools. In this paper, we summarize the recent research and present a comprehensive review on ML techniques for analog/RF circuit modeling, design, synthesis, layout, and test.

**Keywords:** Artificial Neural Network, Analog and Radio Frequency, Deep Learning, Machine Learning, Artificial Intelligence, Integrated Circuits, Synthesis, Optimization.

## 1. Introduction

Analog and RF devices and circuits are fundamental electronic components in the broadest type of electronic devices. In addition to consumer electronics markets, the IC industry is, more than ever, pressed by the enormous demand of medical, healthcare, automotive, or security electronics, for example. Analog/RF components are already present in more than 50% of the total IC shipments yearly; thus, their design, test, and validation are fundamental tasks to meet the stringent time-to-market constraints and production costs. Computer-aided design (CAD) tools are quintessential in the design of analog IC. In consumer electronics, the massification balances the design effort of the analog/RF circuits. However, the lack of EDA challenges the design of the custom ICs needed to produce state-of-the-art customized equipment and created barriers to innovation. The adoption of automation mechanisms can significantly reduce their development time while simultaneously improving their performance. However, design automation in analog IC design flow is far from being the norm, despite the enormous efforts made by the EDA community over the past few decades. Analog IC design is in sharp contrast to the digital IC design flow, where plenty of EDA tools are available and established. Analog ICs' nonlinear behavior, the increasing complexity observed in nowadays applications, and the challenges in deep

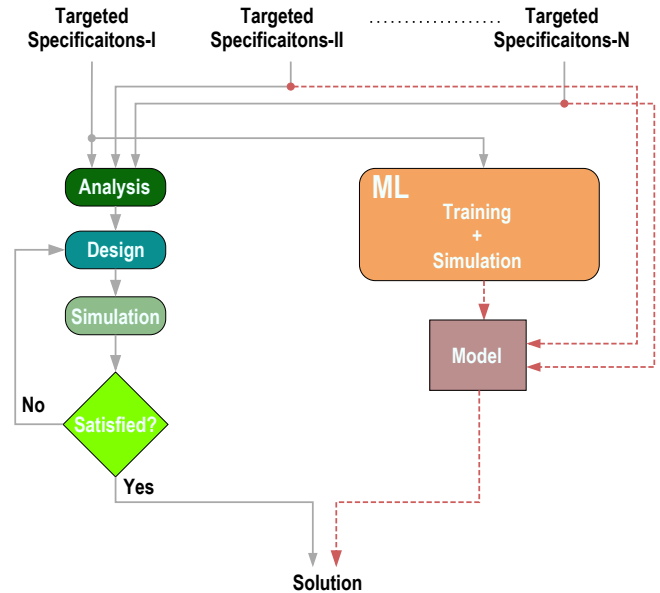


Figure 1: Conventional and ML-based design flows.

nanometer integration technologies, only further increase the difficulties faced on analog/RF IC design and test, placing additional pressure on analog/RF IC designers and EDA development teams. ML has been the subject of intensive research, and it is reshaping society in many different ways. ML also opens new perspectives on how computational intelli-

\* Corresponding author

Email address: engin.afacan@lip6.fr (Engin Afacan<sup>1</sup>)

gent EDA tools for analog and RF IC design can help the IC designers to be more productive.

Fig. 1 illustrates the general flows of the conventional and ML-based design methodologies. When following the conventional flow, the designer repeats the flow for each different targeted specifications, even for the same problem. Designer's own experience, knowledge, and instincts are of the utmost importance, but still, the lack of formalization substantially limits knowledge dissemination and reuse. On the other hand, the ML-based design expeditiously produces solutions. The caveat is how to obtain such successful models. This paper addresses the efforts made by the EDA research community and how the traditional and computational intelligence tools can take advantage of the advances in ML. This paper's organization is as follows. In Section 2, the ML foundations are briefly overviewed, including models with different types of supervision. Section 3 presents the existent techniques for modeling of analog/RF ICs based on ML techniques. In Section 4, the focus is given on the ML synthesis, whereas in Section 5, we outline the most recent ML techniques for layout generation. Studies on fault testing and diagnosis that exploit ML techniques are discussed in Section 6. Finally, in Section 7, the conclusions and future research directions are drawn.

## 2. Background

Machine (or statistical) learning foundations are from artificial intelligence, but while the latter aims at building expert systems, the former focus on the statistical properties of data [1]. Bayes' essay on Probability Theory [2] laid the theoretical foundations for statistical learning and is the base for some early ML techniques, such as Naive Bayes or Markov Chains. In 1951, the first neural network machine was proposed, but was only after Frank Rosenblatt's perceptron [3] and back-propagation [4], in 1958 and 1986, respectively, that artificial neural networks (ANNs) began to receive more attention. In the meanwhile, many other advances have been achieved, and today many techniques to design ML systems for solving classification and regression tasks are available. In a classification problem, the objective is to categorize the data. For example, an email spam filter aims to assign incoming emails to the "spam" or "no-spam" categories. Whereas in regression, the systems try to describe one or more continuous-valued dependent variables as functions of the observations in the data. Critically to all ML systems is their ability to generalize well to new data and avoid overfitting to the training data. Overfitting occurs when an ML system starts to learn the noise in the training data instead of learning the underlying mechanisms that generated the data [1, 5, 6].

Another critical characteristic of ML systems is the amount and type of supervision. In supervised learning, the data used to train the system must include the desired solution, called a label. The label can be categorical (in classification problems) or continuous valued (in regression problems). Some important supervised learning algorithms are linear discriminant analysis, linear regression, logistic regression, poly-

nomial regression, decision trees, support vector machines (SVMs), and ANNs, among others. In unsupervised learning, the data is unlabeled and algorithms group data points based on their features. Clustering, visualization, dimensionality reduction, and anomaly detection are examples of unsupervised learning. Common unsupervised learning algorithms are the k-means and principal component analysis (PCA), and their variants. In Fig. 2a logistic regression (that despite the name is a classifier) illustrates supervised classification, in Fig. 2b polynomial regression is used to model  $Y$  as a function of polynomials of  $X$ , and in Fig. 2c, k-means is used for clustering.

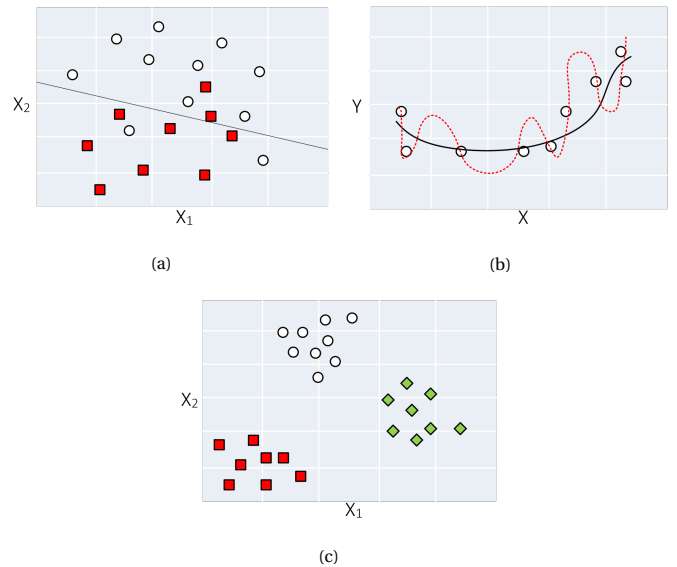


Figure 2: (a) Logistic regression for the classification of two classes; (b) Polynomial regression that describes  $Y$  as a function of  $X$ , solid line shows a good regressor, dashed line shows an overfitted regressor; (c) Group data into 3 different clusters using k-means.

There is also semi-supervised learning, where the data that is used to train the system is partially labeled, and the system is trained with combinations of supervised and unsupervised learning algorithms. For example, deep belief networks (DBN) build upon restricted Boltzmann machines (RBMs) or autoencoders trained in an unsupervised manner, and then the whole system is fine-tuned using supervised learning techniques [7]. A different approach is taken in reinforcement learning (RL). In an RL system, an agent observes and interacts with the environment by selecting and executing actions. The agent is trained to learn a policy that maximizes the expected outcome of the actions over time [8]. These systems can teach robots to learn motor skills [9] or play complex board games [10]. It is also essential to distinguish the application from the algorithm as the same underlying ML algorithms can be applied in several or all of these approaches. ANNs, for example, can be used in all the approaches mentioned above. ANNs, in the form of convolutional neural networks (CNNs), are incredibly efficient image classifiers in a supervised learning setting. On the other hand, autoencoder networks can be trained without supervision to learn latent

space, and deep reinforcement learning has shown impressive results in beating human experts on several games [11]. ML is widespread and horizontally suited for many applications, including EDA. While most algorithms can perform identically on curated large datasets [12], data can be difficult and expensive to acquire, and small to medium-sized datasets are usual. Selecting the most suitable method for the target application is an important design choice. The available options are many, and, in the next sub-sections, some methods found in EDA are briefly described.

### 2.1. Clustering

Clustering algorithms are unsupervised learning algorithms that group unlabeled data into  $K$  predefined clusters, using some distance,  $d(x_i, x_j)$  metric between the data points. The objective of a clustering algorithm is to find the mapping  $C^*(x) = k, k \in 1, 2, \dots, K$  that minimizes (1).

$$W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(x_i)=k} \sum_{C(x_j)=k} d(x_i, x_j). \quad (1)$$

The possible mappings between the input data points and the clusters grow very sharply with the number of data points and number clusters, quickly becoming intractable. Therefore, clustering is usually solved using iterative greedy descent methods, such as K-means. K-means start by assigning centers (randomly or using some spreading criteria) to the clusters, then iterates the two following steps, until no further improvement is possible:

- for each center, identify the training points that are closer to that center than to the other centers;
- update each cluster's center to become the mean of the data points identified as belonging to it.

Clustering methods can be effective solutions to reduce the amount of data to be processed without losing too much information. In [13] clustering is used to reduce up to 10% the data required to train an SVM classifier for analog IC fault diagnostics, whereas in [14], fuzzy c-means groups the elements of the population during analog IC sizing optimization to apply time-expensive Monte Carlo simulations only to a handful of meaningful tentative solutions. While clustering can result in significant savings, determining the number of clusters without losing information can be difficult. Also, clustering is sensitive to the distance metric and scaling between features.

### 2.2. Principal Component Analysis

Another unsupervised learning algorithm is PCA. Like clustering, PCA can be used to reduce the amount of data without losing information. PCA is a linear operation that transforms the feature space in a latent space maximizing the variance. Formally, taking the data's covariance,  $S$ , defined in (2), the

variance of the  $i^{th}$  coordinate in the projected space, where  $u_i x_n$  is given by  $u_1^T S u_1$ .

$$S = \frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})(x_n - \bar{x})^T. \quad (2)$$

Hence to maximize the variance of the projection and constraining the  $\|u\|$  to prevent it from going to infinity, the solution to the Lagrangian is  $Su_1 = \lambda_1 u_1$ , meaning that  $u_i$  is an eigenvector from  $S$  and the corresponding variance is a maximum if the corresponding eigenvalue is the largest. The additional principal components are the eigenvectors corresponding to the higher eigenvalues. By keeping only the components with more variance, data is represented with fewer features. [15] used PCA to reduce the number of design variables in the optimization of an amplifier and a voltage-controlled oscillator (VCO). PCA is a linear operator and does not handle nonlinearity in the data, however, the kernel trick [16] can be used to extend it to nonlinear relations in the data.

### 2.3. Linear Discriminant Analysis

Linear discriminant analysis (LDA) is, like PCA, a linear method to reduce the dimensionality. However, instead of maximizing variability, it aims at maximizing the separation between classes. Fisher's LDA is commonly used and finds the linear combination  $L = a^T X$ , that maximizes the between-class covariance relative to the within-class covariance, as defined in (3).

$$\max_a (a^T S_B a) \text{ subject to } a^T S_W a = 1. \quad (3)$$

The between-class variance of  $L$  is  $a^T S_B a$ , whereas the within-class variance of  $L$  is  $a^T S_W a$ , with  $S_B$  and  $S_W$  being the covariance matrix of the class centroid matrix and the within-class covariance matrix, respectively. The solution to the generalized eigenvalue problem in 2, results in the  $i^{th}$  discriminant variable being given by  $L_i = (S_B^{-1/2} v_i)^T X$ , for the  $i^{th}$  eigenvector of  $S_B^{-1/2} S_W^{-1} S_B^{-1/2}$  with  $i^{th}$  largest eigenvalue. Also, like PCA, LDA can be extended using the kernel trick to learn nonlinear mappings. In [17] kernel PCA is extended to consider separability and used pre-process features on for an ANN-based fault diagnosis method. Also, for fault diagnosis, [18] uses kernel LDA to reduce dimension before training a naïve Bayes classifier.

### 2.4. Decision Tree

Decision trees (DTs) formalize a decision-making process in a directed acyclic graph. There are 2 types of nodes in DTs: the decision nodes and the terminal nodes. The first represents decision criteria, while the latter represents the outcome of the sequence of decisions. DTs offer a clear insight into data, and it is easy to extrapolate conclusions from them. The DTs are often trained with the CART (classification and regression tree) algorithm, which splits the training set in two subsets using a single feature  $k$  and a threshold  $t_k$ , and then, tries to minimize the cost function given by:

$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}, \quad (4)$$

where  $G_{left}$  is the amount of impurity of the left/right subset and  $m_{left}/m_{right}$  is the number of instances in the left/right subset. Nevertheless, complex datasets lead to over-complex trees that end up overfitting. Random Forests improve generalization by creating ensembles of DTs where features and data points are randomly sampled with replacement. [19] uses DTs to automate the selection of a circuit topology given the target specifications, as illustrated in Fig. 3. [20] use the random forest to identify possible rare events during the Monte Carlo simulation.

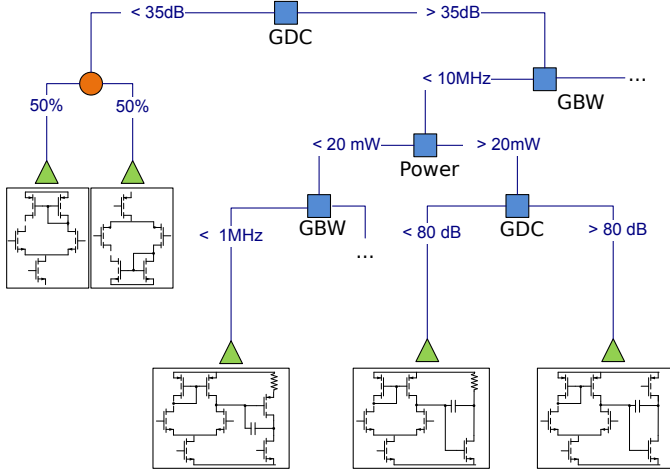


Figure 3: DT for the selection of a circuit topology given the target specifications. Adapted from [19].

## 2.5. Naive Bayes

The Naive Bayes classifier algorithm affords fast, highly scalable training and scoring. The Naive Bayes classifier works by selecting the class  $\hat{G}$  to a new data point represented by features  $x$  from  $R_n$ , according to (5), where  $\mathcal{G}$  is the set of classes, i.e., it chooses the class with maximum a posteriori.

$$\hat{G} = \arg \max_{G \in \mathcal{G}} (p(G/x)) \quad (5)$$

The classifier uses the Bayes rule to compute the posterior, assuming the features' independence, as indicated in (6).

$$p(G/x) = \frac{p(G) \prod_{i=1}^m p(x_i/G)}{\sum_{g \in \mathcal{G}} p(g) p(x/g)} \quad (6)$$

Since the denominator is a constant for a given feature vector  $x$ , the naive Bayes classification decision rule can be formalized only with the prior and the likelihood, as indicated in (7).

$$\hat{G} = \arg \max_{G \in \mathcal{G}} \left( p(G) \prod_{i=1}^m p(x_i/G) \right) \quad (7)$$

These classifiers are relatively easy to understand and build. They are easily trained and do not require large datasets to produce effective results. Despite the assumption of independence of the feature, which is not valid for most real-life situations, naive Bayes is a practical approach in many applications. [18] uses a naive Bayes classifier for fault diagnosis.

## 2.6. Support Vector Machines

SVM is a supervised learning algorithm for data separation, fitting the boundary,  $h(x)^T \beta + \beta_0 = 0$  that maximizes the margin,  $2/\|\beta\|$  between classes as shown in Fig. 4.  $h(x)$  is a transformation of the feature space that enlarges the decision space to improve the performance of the linear classifier, and typically translates to nonlinear boundaries in the original space. In the case of non-separable classes, as shown in Fig. 4 the margin is maximized subject to a total budget  $\sum \xi_i \leq \text{constant}$  as defined in (8).

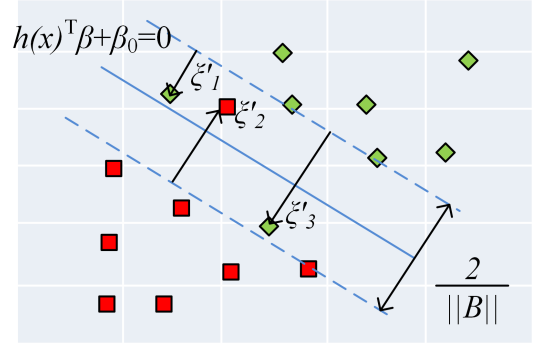


Figure 4: Concept of Margin for non-separable classes. The point on the wrong side of their margin are identified by  $\xi'_j = M \xi_j$  [5].

$$\min \|\beta\| \quad \text{subject to} \begin{cases} y_i (h(x_i)^T \beta) \geq 1 - \xi_i \quad \forall i, \\ \xi_i \geq 0, \quad \sum \xi_i \geq \text{constant}. \end{cases} \quad (8)$$

Where  $\xi_i$  represents how far a point is on the wrong side of the margin. It is zero for points on the proper side of the margin, and  $y_i \in \{1, -1\}$  is the class identifier. The solution to this problem is obtained by maximizing the dual lagrangian, expressed in (9), subject to  $0 \geq \alpha_i \geq C$  and  $\sum \alpha_i y_i = 0$ . The corresponding decision boundary is given by (10).

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{i'=1}^N \alpha_i \alpha_{i'} y_i y_{i'} \langle h(x_i), h(x_{i'}) \rangle. \quad (9)$$

$$f(x) = \sum_{i'=1}^N \alpha_i \alpha_{i'} y_i y_{i'} \langle h(x_i), h(x_{i'}) \rangle + \beta_0. \quad (10)$$

Where  $\alpha_i$  are the Lagrange multipliers, and  $\langle h(x_i), h(x_{i'}) \rangle$  is the inner product in the transformed feature space, or kernel function  $K(x, x')$ . Only those observations  $i$  for which the constraints in are exactly met have nonzero coefficients  $\alpha_i$ . Therefore, the boundary is a linear combination of some data points at the edge of the class, also called the support vectors. SVM quickly identifies the best linear separator if the data is linearly separable, for nonlinear patterns, the kernel trick allows the SVM to do the separation in very high dimension spaces (even infinite). However, grasping insights from the parameters is very hard, making hyper-parameters tuning and selection of the correct kernel difficult challenges. SVMs also underperform if the dimension of the data exceeds the number of points. In [21], SVMs that identify infeasible regions of the solution space avoid unnecessary circuit simulation during sizing optimization.

## 2.7. Artificial Neural Networks and Deep Learning

Deep learning has become quite popular in the last few years in image processing, speech recognition, and other areas where a high volume of data is available. Its basic element is the perceptron, a single layer of linear threshold units (LTUs) which one computing a weighted sum of its inputs,  $z$ , and then applying a nonlinear activation function:

$$\begin{aligned} h_w(x) &= \text{activation}(z) \\ &= \text{activation}(W^T \cdot x) \\ &= \text{activation}(w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n), \end{aligned} \quad (11)$$

where  $x$  is the input vector values, and,  $w$  the vector of weights of the linear threshold unit. This single layer of LTUs, or perceptron, makes a prediction for each instance of  $x$ , and thus, its training can be done by reinforcing the connection weights that contribute to correct the prediction, according to:

$$w_{i,j}^{\text{nextstep}} = w_{i,j}^{\text{currentstep}} + \lambda(y_j - \hat{y}_j)x_i, \quad (12)$$

where  $w_{i,j}$  is the weight between the  $i_{th}$  input and the  $j_{th}$  output,  $x_i$  the  $i_{th}$  input value of the training instance,  $y_j$  the target  $j_{th}$  output for the current training instance,  $\hat{y}_j$  the predicted  $j_{th}$  output for the current training instance, and,  $\lambda$  the learning rate. An ANN is essentially a multi-layer perceptron, i.e., one of more layers of LTUs, which can be efficiently trained using the backpropagation training algorithm, developed in 1986 by D. E. Rumelhart [4]. ANNs can build effective end-to-end ML systems, and they are replacing entire processing pipelines in applications such as computer vision and natural language processing. ANN is an extremely flexible construct. ANNs can also incorporate application-related knowledge both in the model structure and in the cost function. These multi-faceted tools allow the implementation of different tasks in the same network. The price paid when using ANNs is the sheer amount of hyperparameters that can be tweaked. They go from the network's structure and activation functions to the optimizer that finds the best combination of hyperparameter. Unlike SVMs, whose solutions are the optimum of a convex function, ANNs weights' optimization often leads to local optima of the cost function. Therefore initialization is also an essential part of the training. Still, ANNs are widely used in EDA for modeling [22], synthesis [23], layout generation [24], and fault testing [25].

## 3. Modeling of Analog/RF Circuits and Systems with Machine Learning Techniques

Conventional analog IC design is particularly time-consuming due to the complicated non-linear relationship between the design parameters and device/circuit/system specifications. Typically, hand calculations may facilitate this design process, which considerably narrows the design space and provides a good starting point for the designer. Nevertheless, design time still depends on the experience of the designer, who performs a large number of iterative simulations

to achieve the targeted design specifications considering accurate physical properties of the circuit and devices. Furthermore, the approximation errors during modeling and the difficulty of the circuit analysis due to the countless trade-offs dramatically increase the design time. The idea behind using ML techniques in analog/RF circuit design is to generate functional models of devices/circuits/systems that accurately mimic their functional behaviors and exploit them for different contexts. Recently, ANNs have become a viable alternative to numerical modeling methods, analytical methods, and empirical models. These models can immediately generate the solution for a pre-trained problem; hence, the designer can bypass numerous expensive simulations. Over the years, ML-based modeling has been utilized at different levels (from a single device to a complicated system) and for different applications (analog, RF, and heterogeneous). For the reader's convenience, the key properties of the reviewed studies are summarized in Table 1.

### 3.1. 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 [26], 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  $\epsilon$ -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 [21]; 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. 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 transconductance amplifier (OTA) and a mixer.

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. [27] presents a novel methodology for ANN aided inclusion of power consumption

Table 1: Summary of modeling of Analog/RF device and systems with ML techniques.

Reference	Application-Device	Method(s)	Contributions
[26]	Analog Circuits-GaAs transistor	SVMs ( $\epsilon$ -SV regression)	Robust and accurate modeling of GaAs transistors and circuits
[21]	Analog Circuits-CMOS	SVMs	Efficient active learning scheme for feasible design space selection
[27]	AMS circuits-CMOS	ANN (TDNN)	Robust modeling of power consumption for AMS circuits
[22]	Analog-n/d	ANN (Back propagation)	A generic modeling of power consumption for heterogeneous systems
[28]	RF- microwave components and MESFET	ANN (several)	Review of ANN based CAD for microwave designs
[29]	RF-microwave components, HMT and MESFETs	ANN (several)	Review of model development and nonlinear modeling of microwave devices
[30]	RF-CPW components	ANN (EM based)	Efficient modeling of CPW components for accurate performance estimations
[31]	RF-UC-PBG rectangular waveguide	ANN (RBF-MLP)	Efficient modeling of RF devices for nonlinear microwave applications
[32]	RF-MESFET	ANN (WNN-MLP)	Faster design of large signal hard- nonlinear power transistors and circuits

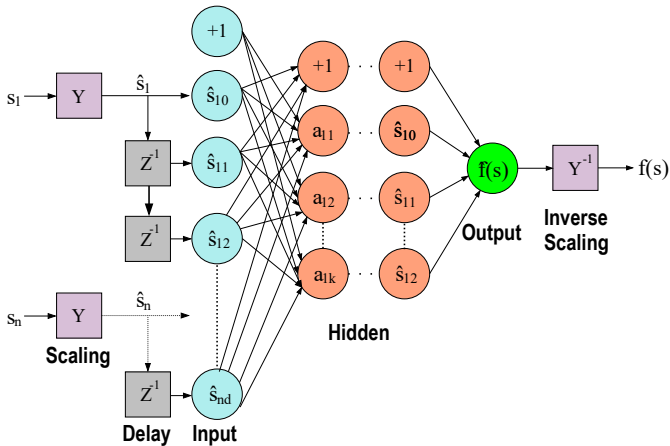


Figure 5: TDNN delay neural network model [27].

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 Fig. 5, 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%.

A different application of ANN-based modeling is presented in [22], 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. Moreover, one can perform on-line 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. 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



digital converter, and a wireless transceiver.) According to the results, the maximum and the average estimation errors are 3.06% and 1.53%, respectively.

### 3.2. 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 [28], 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 MESFETs, 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 [29] 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 [30]. CPW transmission lines (frequency dependent  $Z_o$  and  $\epsilon_{re}$ ), 90° 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. [31] 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 Fig. 6. The complicated design problem is divided in sub-problems, distributed over the neural networks of the modular structure. Their claim is that this type of modular structure can improve the efficiency of EM-based

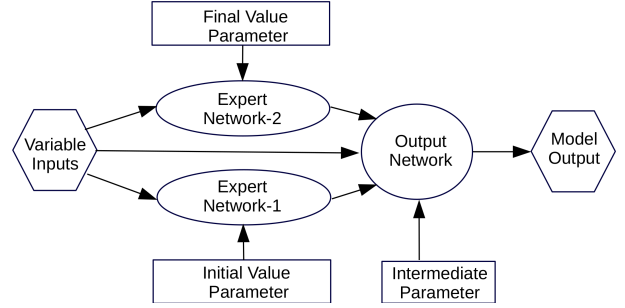


Figure 6: The proposed framework in [31].

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 [32]. The first example is a transistor modeling example, where 10 neural networks are used as shown in Fig. 7.

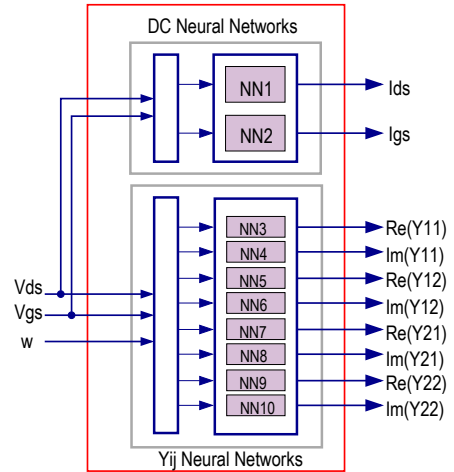


Figure 7: Volterra-ANN device model [32].

Two of them utilize  $V_{ds}$  and  $V_{gs}$  to obtain  $I_{ds}$  and  $I_{gs}$ . The remaining 8 use  $V_{ds}$ ,  $V_{gs}$ , and  $\omega$  to yield real and imaginary values for  $Y_{ij}$ . 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  $\omega$  and

Table 2: Summary of ML-based IC circuit synthesis applications.

Reference	Application	Method(s)	Contribution
[33]	Analog Circuit Optimization	KNN	Large-scale data mining with boosted regressors
[34]	Analog Circuit Optimization	ANN +SPEA2	Efficient optimization via replacing the simulator by ANN based model
[35]	Analog Circuit Optimization	ANN +GA	Fast and accurate layout-aware Op-Amp synthesis
[36], [37]	Analog Circuit Optimization	Bayesian Optimization (GP+LCB+NSGA-II BNN+LCB+MOEA/D)	Fast and accurate optimization of analog circuits to obtain better PFs
[38]	Performance Space Exploration	Bayesian Regression (GA+SVMs)	Accelerated large-scale design space search via multiple ML approaches
[39]	Performance Space Exploration	Polynomial Regression	Automatic generation of POFs for new design context without simulation
[40]	Performance Space Exploration	ANN based text mining +Sparse regression	A global performance space search on the Internet via knowledge harvesting
[41],[23]	Analog Circuit Synthesis	ANN (GRP+MLP)	Technology independent sizing of analog building blocks
[42],[43]	Analog Circuit Synthesis	ANN (MLP)	Automatic generation of training dataset for analog circuit sizing
[44]	Analog Circuit Synthesis	ANN	Generation of better FOMs for Op-Amps via ANN based circuit synthesis
[45]	Analog Circuit Synthesis	DL+RELU	Efficient multiple performance estimation of Op-Amps with DL based models
[46]	Analog Circuit Synthesis	ANN	Examining the effect of ANN hyperparameters on analog circuit synthesis
[47]	RF Circuit Synthesis	GA+ANN(MLP)	Efficient synthesis of RF circuits via GA assisted ANN
[48]	Analog Circuit Synthesis	Polynomial Regression + ANN	Generation of reusable POFs for analog circuit design
[49],[50]	Analog Circuit Synthesis	RL (L2DC)	Efficient sizing of analog circuits (25x faster than hand design)
[51]	Analog Circuit Synthesis	Deep RL	Efficient layout parasitics-aware circuit synthesis (40x faster than GA)

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.

#### 4. 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](#).

##### 4.1. ML for Optimization-based Circuit Synthesis

The most established method to automate the circuit synthesis is optimization-based circuit synthesis that uses an optimization method to explore the design space. Analog/RF circuit optimization tools certainly accelerate the design time, in which several nature-inspired algorithms (evolutionary, particle swarm, reinforcement learning, etc.) are em-

ployed to search the design space and find an optimal solution for a given circuit problem. However, a large design space should be visited via simulations through the iterations, and, more dramatically a few of them are only used at the end of optimization process, which means that a large portion of the simulation data is wasted during the optimization process. Integration of ML techniques into the conventional optimization loop is highly promising to mitigate this computational cost by reuse of the simulation data in order to learn circuit behavior and to develop a model that will replace the circuit simulator once the model is obtained, shown Fig. 8. Since the model is generated by using real and filtered (satisfying all constraints and biasing conditions) data, the accuracy of the optimization does not degrade after the replacement of the simulator by the model.

In general, circuit optimization tools optimize the circuit design parameters (device dimensions, values of passive components, and bias voltages and currents) considering the design objectives and constraints. Conventionally, the employed ML typically emulates the circuit behavior thorough a model as a function of design variables. On the other hand, it is also possible to change the roles of objectives and design variables during modeling, where the circuit design variables are modeled as a function of design objectives. There are several attempts that integrate the ML approaches into the conventional optimization flow in the literature.

Exploiting optimization tools for dataset generation is one application of ML-based optimization techniques [33]. The optimizer manipulates many parameters during its course, creating data points suitable for data mining. Then, they try to select a model and fit it through regression. For such a large population of high-dimensional data points, it is difficult to find a suitable functional for the regressor that can adequately fit the data, while remaining simple enough to allow the solution of the fitting parameters. Some authors have suggested radial basis functions in and posynomial-approximated signomials to this end. In this paper, a committee of regressors is built, each of which fits perfectly in some portions of the design space, and with high error in others, rather than struggling to build a single regressor capable of superior fitting across a very large sample space. To combine the results of the regressors, they use a k-nearest neighbors (KNN) algorithm to select the best fitting K regressors, and apply weighted averaging to combine the outputs of the regressors, where the weights are determined by the distance from the point to be projected. The regressors themselves are two hidden layer feedforward networks with 10 neurons in each hidden layer. This approach is illustrated on two examples, an RF low noise amplifier (LNA) circuit and a more complex LNA circuit with about 50 devices. One peculiarity of this study is that the models are developed over a single objective synthesizer so that the synthesizer evolves towards a small region in space. Hence, the models developed during the last 20% of the synthesis cluster more closely about the final solution, whereas those developed in the 20% try to cover the whole design space, but have more error because they have more outlier points. An ANN is embed-

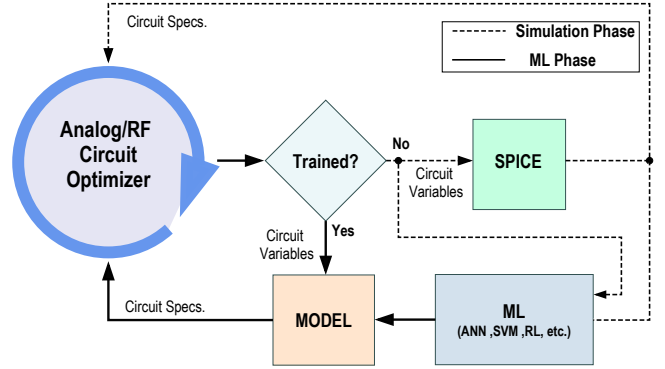


Figure 8: A general flow of the ML-based analog/RF circuit optimization.

ded into a multi-objective analog sizing tool in order to increase the efficiency of the optimization process in [34]. The flow of the proposed approach is very similar to the flow illustrated in Fig. 8. Strength Pareto Evolutionary Algorithm-2 (SPEA), which is a well-known evolutionary algorithm for multi-objective evolutionary algorithms, is employed as an optimization engine, where SPICE is used as the performance estimator. The optimization starts with SPICE simulations and goes on for several tens of generations. Meanwhile, the produced simulation data is used to train an ANN. Thanks to the present mechanisms in the optimization tool (constraint violation check, operating region elimination, performance selection), a filtration is automatically applied to avoid infeasible solutions; thus, an accurate model can be efficiently developed. Once the ANN model is successfully trained, the optimization moves to a second phase, where the simulator is replaced by the ANN model; thereafter the performance estimation is performed without running any simulation. Conventional feed-forward neural networks are used to construct the ANN, which has 4 layers: one input, one output, and two hidden layers. To demonstrate the proposed approach two different types of amplifiers, a single-stage amplifier and a folded cascode amplifier, are optimized with both the conventional and proposed tools. According to the results, the proposed tool can reduce the execution time by up to 64.8%.

Another ANN-based methodology is proposed in [35] for creating fast and efficient models for estimating the performance parameters of CMOS operational amplifier topologies. The flow of the algorithm is very similar to the flow shown in Fig. 8. A uniform sampling of the parameter set was performed to create 3125 different sizings of the Op-Amps to be used as training samples. Seven neural networks were set up for seven performance parameters. These neural networks were of feedforward type with one hidden layer. Then, the neural network models were used in a synthesis flow inside a combinatorial optimizer, namely genetic algorithms. The approach was demonstrated on several Op-Amps and was found to yield reasonably good results.

Bayesian optimization based approaches are commonly used for expensive black-box functions. The approach has two important compartments: the probabilistic surrogate

modeling and the acquisition function. The surrogate models are used for performance prediction while the acquisition function is used to explore the space considering the surrogate model optimally. In [36], a Bayesian based multi-objective algorithm is proposed for automatic synthesis of analog/RF circuits, in which Gaussian processes (GP) are used as the online surrogate models for multiple objectives and the lower confidence bound (LCB) functions are employed as the acquisition functions to select data points. First, a GP model is trained using the existing simulation data. The GP models are only updated when a new data point is selected from the PF. Hereby, the circuit simulation is called to obtain the performances. Then, the LCB functions are constructed and optimized by using a modified version of the NSGA-II algorithm. The optimum of the acquisition functions are selected as the next data points to be evaluated. As the case study, a three-stage low power amplifier, a 60 GHz transformer, and a power amplifier, are optimized with proposed algorithm. According to the results, the proposed tool achieves better PFs than the state-of-the-art algorithms with considerably lower simulation effort.

In [37], a similar multi-objective optimization approach that uses Bayesian optimization is proposed. To model multiple performance of interest (PoI) of any analog circuit, instead of GP, a single Bayesian Neural Network (BNN) is used. To train the BNN efficiently, automatic differential variational inference (ADVI) method [52] is employed. The BNN is then combined with a Bayesian optimization framework, in which a modified MOEA/D algorithm is used as optimization engine. The BNN model is built with a training data set and the acquisition function, namely LCB, is defined based on the BNN model, which is minimized through optimization. The proposed approach is initialized with generation of "pseudo" Pareto points with BNN. Then, transistor-level simulations are performed for each point to obtain the actual result. The BNN model is recalibrated using the new transistor-level simulation results. The procedure is repeated until the convergence is achieved. A charge pump and a three stage amplifier circuits are used to demonstrate the proposed approach. According to synthesis results, the proposed BNNBO method can achieve accurate POFs with almost 0.5× reduction in computation cost.

#### 4.2. ML for Design Space Exploration

Design space exploration is another cumbersome problem, in which the whole design space, which is theoretically infinite, should be scanned in order to determine the design boundaries for a given problem. Furthermore, this process must be repeated for new contexts, i.e., supply voltage, technology node, etc., although the problem is the same. Several automatic sizing approaches have been developed to facilitate the search of such infinite design spaces; however, they suffer from curse of dimensionality due to the excessively increased simulation workload. The use of ML techniques for design space exploration is based on extracting regression models of the circuit that is being optimized. Similar to the other approaches, the data for fitting the model is

collected via the simulations at the initial phase of the optimization process. Then, a circuit model is developed through fitting the obtained data to a suitable function; thus, performance evaluations can be performed by using the circuit model without SPICE simulations and the design space can be efficiently explored.

A learning-based performance space exploration for analog/RF circuit approach is presented in [38]. The developed methodology has a hierarchical structure, which consists of three major steps: device/circuit model fitting, evaluation, and design boundary determination and adjustment. First, all geometry and biasing variables are explored and simulated to obtain the circuit-level design variables for different technology nodes. Numerous simulations are performed and the obtained data are used to fit the circuit behavior into a model, where Bayesian regression is employed. Second, the evaluation takes place, in which SVMs is employed in parallel with a genetic algorithm to reduce the runtime. Finally, the sample boundaries are dynamically adjusted considering the density of feasible samples. The core of the proposed software is developed with C++, where MATLAB convex optimization tool and SPICE simulator are also integrated for searching and performance estimation, respectively. To demonstrate the proposed framework, a folded-cascode operational amplifier and an RF distributed amplifier are optimized and the results are compared with the results of two different circuit synthesis approaches. According to the comparison results, the proposed tool successfully generates solutions for given design specifications within considerably shorter runtimes.

Design space exploration is part of a larger framework, which is used as a design assistant tool for analog intellectual property (IP)-(DATA-IP) [39]. The idea behind this study is to generate the Pareto-optimal fronts (POFs) for different design contexts (different load, power, etc.) for a particular circuit without performing any simulations. The proposed framework presents a number of different options to users: generation of POFs with the embedded optimization algorithm, using an existing POF to determine the design parameters for a given circuit problem, generation of POFs for new contexts using the model, topology selection, and verification with existing SPICE-based evaluation. The proposed framework uses a Strength Pareto Evolutionary Algorithm-2 (SPEA2) as the multi-objective optimization engine. First, a number of POFs are generated for either different loading or power consumption constraints via the optimization algorithm. Then, the obtained POFs are fitted into a model by using polynomial regression. Once a circuit is successfully modelled, the POF of a new circuit context can be readily generated without any optimization run. The framework is also capable of verification of the solution points on the POFs generated through the circuit model. Furthermore, all those options are integrated and a user-friendly interface is developed. A folded cascade amplifier is selected to demonstrate the proposed approach. According to the results, the proposed design framework successfully generates the POFs for new design context. The analog library can also be extended

by adding new topologies and circuits.

In [40], design space exploration for large-scale analog circuits is examined via a pretty marginal way. The proposed approach is based on harvesting the huge design knowledge from published papers and datasheets on the Internet and encoding the knowledge as PFs rather than using an optimization based framework. Furthermore, the obtained high dimensional PFs for large-scale analog systems also include layout parasitics and process nonidealities since only silicon verified results are used. The approach has two major functions: harvesting the design knowledge from the Internet and modeling of PFs by using the collected data. For data collection, an ad hoc text mining technique is adopted such that it provides high-quality information from different sources by analyzing the patterns based on statistical learning. The collected data is then preprocessed to fit the POF for a given circuit, since many of them may not be Pareto optimal. To select the Pareto optimal points from enormous data, an efficient algorithm is proposed, which is based on filtration of feasible points for each performance metric and determination of Pareto dominated ones. Then, the basis function selection takes place. Since using a fixed set of basis functions are not applicable to all cases, an adaptive selection mechanism is proposed, which uses an adapted version of sparse regression with grid discretization. The algorithm basically uses a brute-force approach by iteratively selecting of important basis function from a huge candidate pool. By using the basis function and the model coefficients, the nonlinear function of each performance metric is constructed and Pareto front of interest is defined. Demonstrated examples indicate that the proposed tool can accurately model POFs for complex and high-level analog systems.

#### 4.3. ML-based Circuit Synthesis

Conventionally, as the first step of the design process, a designer usually selects an appropriate topology among a number of different topologies and sizes the circuit of that particular topology. The designer should re-design the circuit for a different technology or for different specifications even if there is no change in the topology. Typically, it is supposed that the technology parameters are the inputs of the circuit as well as the device dimensions. As a result, once a topology is accurately trained via ML, the model can generate solutions for different technologies without running any simulations.

In [41], ANN assisted technology independent design of current steering PMOS only digital-to analog converter (DAC) is presented. The motivation behind the study is to obtain design parameters of a pre-trained circuit for the newer technologies without any circuit simulation effort. For that purpose, a large database for the current steering DAC is constructed by numerous simulations for different technologies; 1.5  $\mu\text{m}$ , 0.5  $\mu\text{m}$ , and 0.35  $\mu\text{m}$ . Static specification parameters (SSP), Differential Nonlinearity (DNL) error, Integral Nonlinearity (INL) error, monotonicity, and gain error) are defined as the inputs of the ANN while the transistor dimensions are the outputs of the network. General Regression

Neural Network (GRNN) is used as the ANN approach. According to the results, the ANN-based design approach can design the circuit for a newer technology. Furthermore, the proposed methodology can achieve better specifications (improved monotonicity and reduction in DNL, INL and gain error.)

Similarly, an ANN assisted technology independent sizing of building blocks (basic current mirrors and differential amplifiers) for analog integrated circuits is studied in [23]. The models are trained using different technologies; 1.5  $\mu\text{m}$ , 0.5  $\mu\text{m}$ , 0.35  $\mu\text{m}$ , and 0.25  $\mu\text{m}$  while the test data was obtained for only 0.18  $\mu\text{m}$  technology to demonstrate the technology independency of the approach. The ANN-based models provide the corresponding circuit design parameters for a new technology without any circuit simulation. GRNN and MLP utilizing the Rprop algorithm are used as ANN. The proposed approach is based on developing a relatively larger database for different technologies, where properly sized circuits are simulated and the results and the corresponding transistor dimensions are recorded. Basic, cascode, Wilson, and regulated Wilson current mirrors are selected for current mirror examples while the conventional differential amplifier is selected as case study circuit. For both circuit topologies, the ANN provides the width of the transistors for the targeted specifications. To make the approach technology independent, as a straightforward method, the minimum channel length is defined as input parameter as well as the performances (i.e. reference current for mirror circuits, gain, gain-bandwidth product, slew-rate etc.). According to the reported results, GRNN estimates the transistor sizes for current mirror circuits with 94% accuracy while MLP can estimate the sizes for the differential amplifier circuit with 90% accuracy, in which a 10% tolerance was determined for circuit performances.

Generation of large training dataset is generally a problem for ANN-based circuit optimization. The proposed approach [42] addresses this problem for a current to voltage converter circuit. Two levels are utilized to data generation for testing and training and application of this data to developed ANN. An MLP is employed as the ANN structure since it can implement the arbitrary mappings between inputs, i.e., current and gate to source voltage, and the outputs, channel length and width. To generate the training data, SPICE simulations are performed by varying transistor dimensions and the input current. Then, the circuit is modeled through the developed ANN and the results are validated by SPICE simulations. According to the presented results, the developed models can estimate output voltage of the converter with 99.69% accuracy. The same methodology is applied for modeling and design of inverter threshold quantization-based current comparator in [43]. The comparator is decomposed into two stages: current to voltage comparator and inverter stages, in which a particular MLP-based ANN is constructed for each stage. The input current and gate to source voltage are the inputs while transistor width and length are the outputs of the first ANN. Considering the inverter stages, the transistor lengths and input-output voltages are determined as the in-

put of the system while transistor widths are explored. According to the post-layout simulation results, the maximum errors were measured as 0.31% and 0.65% for stages 1 and 2, respectively.

ANN-based circuit synthesis approach has been applied for more complicated circuits such as a three stage Op-Amp circuit [44]. 200 samples were generated through SPICE simulations, 150 of which were used to train the model and the remaining 50 were used for testing. The ANN consists of four layers: input layer, two hidden layers, and an output layer, where dc gain, bandwidth, phase margin, slew-rate, power consumption, etc. are defined as the input parameters, while the transistor dimensions are defined as the output parameters of the network. According to the training results, after 134 epochs, the error decreased to the desired level (less than 1%). Then, the obtained results are validated through SPICE simulations to ensure that they still satisfy the targeted specifications. Even though considerable differences exist between the requested specifications and the simulated performances of the predicted sizing, all targeted design specifications are satisfied. Furthermore, the authors also demonstrate that they achieve high figures of merit (FOMs) for both large and small signal operations.

Another deep learning-based circuit sizing prediction methodology to archive the targeted specifications of Op-Amp is presented in [45]. The methodology is based on learning the correlation between circuit specifications and circuit elements and determining the particular sizing that satisfies the targeted circuit performances. The flow starts with determination of the prediction target. Once a circuit is determined, the prediction of circuit element values from performances is obtained via a regression analysis. In the second step, the data for learning phase is collected through three sub-steps: data classification, data collection, and normalization. 13 specifications, such as gain, bandwidth, power, etc., and transistor widths are selected as the outputs and the inputs of the network, respectively. To generate the learning data, an initial input set consisting of 100 different elements is randomly generated and simulated. The generated solutions are evaluated against a pre-defined FOM and the outperforming solution is selected as the initial point for the next generation, which is also randomly generated in the range of  $\pm 30\%$  of the initial solution. This flow is repeated until 13500 data points are obtained. Then, the obtained performance values are normalized to prevent any error due to the differences in the units. In the third step, a feedforward learning scheme is constructed to implement the regression models, in which transistors sizes and performances are the inputs and the outputs of the proposed network. In the fourth step, the network is trained with 13400 data collected in the second step. The remaining 10 sets are used to validate the prediction accuracy. According to the validation results, 13 circuit performances are predicted with an average accuracy of 93.3%.

The effect of ANN hyperparameters (dataset, number of epochs, data augmentation, etc.) on the performance circuit synthesis approaches is explored in [46]. The ANN models have fully connected layers without weight sharing. To eval-

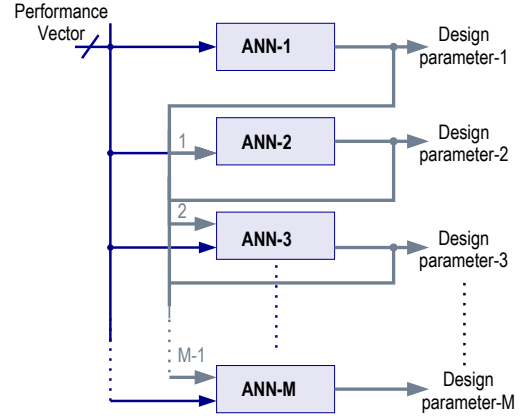


Figure 9: The block diagram of the ANN array methodology [47].

uate the ANN model, 80%-90% of the datasets are allocated for training. The remaining data is kept for model validation; meanwhile, a small portion is used to verify the practical application of the model. The number of nodes is kept high at initial layers and decreased through the further layers to have high performance in the training data, at the cost of overfitting, which is then addressed using L2 weight regularization. Then, a grid search is applied over the hyperparameters (number of layers, number of nodes per layer, non-linearity and regularization factor) to fine tune the model. During the ANN data sampling for  $p$  predictions, an acceptance coefficient of  $\gamma=0.15$  is used to expand the model validity beyond the dataset limits. The selection of solutions from these  $P$  predictions is performed by simulating the predicted circuit sizing, and, either using of the FOMs or using the Pareto dominance sorting. To demonstrate the proposed approach, a single stage amplifier using voltage combiners for gain enhancement is selected. A dataset consists of 16,600 different design points (before data augmentation) is used. DC Gain, bias current (IDD), gain-bandwidth product (GBW), and phase margin (PM) are determined as performance metrics. There different ANNs were trained in this study: ANN-1 is trained by original dataset with 5000 epochs, ANN-2 is trained by augmented (40 times) datasets with 5000, and ANN-3 is trained by the augmented dataset with 500 epochs, whose weights are initialized with the ANN-1. According to the experimental results, ANN-1 is able to find solutions for new specifications, however, it suffers from variability and produces worse designs. On the other hand, ANN-2 and ANN-3 generate better designs when sampled inside the training data. On the other hand, ANN-2 shows more limitations when trying to explore new specifications. ANN-3, because it has transferred information from ANN-1, is more flexible to new specifications, but still lags when compared to ANN-1.

Neural networks can also be used in sequence during mapping circuit performances to circuit sizing [47]. The block diagram of the proposed synthesis methodology is shown in Fig. 9. Compared to the previous approaches, the inputs of the ANNs are the performances while a particular ANN is

used to obtain each design parameter at the output. The first neural network takes the set of desired performances as inputs and has only the chosen design parameter as output. A genetic algorithm (GA) controls the learning process of this ANN. That is, the GA selects which architecture to use (MLP or RBF), and determines its size and which design parameter should be output. Once this ANN is ready, the output becomes the input to a second ANN, who has the task of specifying a second design parameter as a function of the performance criteria and of the first design parameter. This process continues until all design parameters are covered. This procedure was applied to a classical cascode LNA circuit. 235 valid LNA designs were randomly generated for training. 10 design parameters were targeted and 6 variables were used. The models were observed to correctly predict the behavior of the LNA to within 5% error.

A framework for reusable POFs for multi-objective optimization is presented in [48]. The proposed approach has two-levels: context independent performance estimator (CIPE) and circuit sizing predictor (CSP). The flow chart of the approach is shown in Fig. 10.

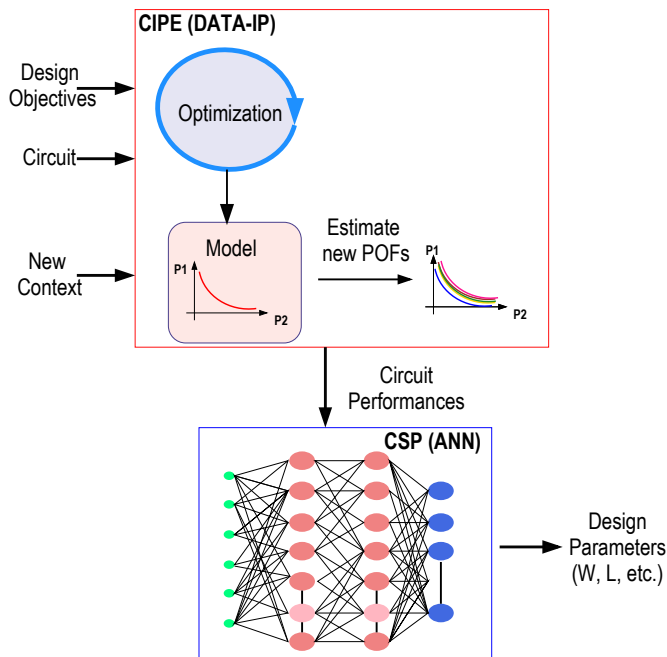


Figure 10: The flow chart of the proposed tool in [48].

In the first step, DATA-IP [39] is assigned as the performance predictor, provides circuit performances and device sizing for new design context. Moreover, the CIPE is extended to predict other performance metrics as well as the design objectives. The predicted outputs are then fed into an ANN, which eventually predicts the device sizing to achieve the corresponding design specifications. Since the input of the CSP given by the CIPE that always follows the optimal performance trade-off, there is no need for filtering the data, resulting in faster training of the model. The efficient training of the model enables the user to use it iteratively with stan-

dard automatic optimization-based tools. To demonstrate the proposed approach, a folded cascode amplifier is used. First, the circuit is optimized for different loading conditions (100fF, 250fF, 350fF, and 450fF); hence, a POF set is obtained to train the model at the CIPE level. The fronts for 150fF, 400fF, and 500fF are used to validate the model. The training of the model takes less than 10ms and predicting 200 samples for a new load takes around 1ms at this level. Considering the CSP, an ANN with 20 input variables, one hidden layer with 100 nodes, and 19 output layers for device parameters is employed. ANN operation takes 15ms to obtain the 100 sizing solutions for the three loads took less than 15ms. According to the validation results the proposed tool exhibits quite good accuracy compared to the SPICE simulations. Authors claim that these performance trade-offs are obtained by using only 300 circuit simulations, whereas a conventional optimization would need almost 120K simulations to perform the same task.

#### 4.4. Reinforcement Learning-based Circuit Synthesis

RL is used to solve complex problems in several systems/applications. RL techniques are inspired by human learning mechanisms, where an agent, working as a human brain cortex, is assigned for learning process based on iterative trial and error process. RL is based on learning from positive and negative assigned rewards. The learning loop of the RL approach is shown in Fig. 11. An agent is a function that transforms the current ( $S_t$ ) state and reward ( $R_t$ ) into an action; environment is a function that converts an action taken in the current ( $A_t$ ) state into the next state ( $S_{t+1}$ ) and reward ( $R_{t+1}$ ). This loop yields a sequence of states, actions and rewards.

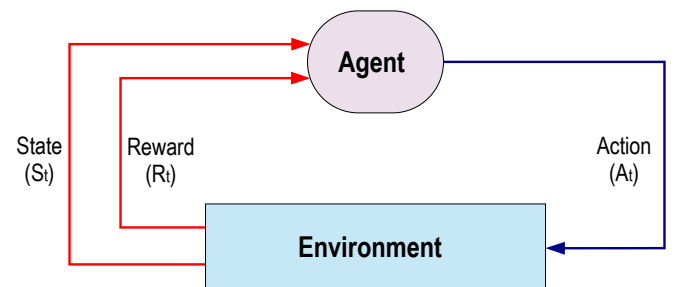


Figure 11: The agent-environment interaction in reinforcement learning loop [49].

As discussed in previous section, generation of training dataset for supervised learning to model circuits is difficult. This is due to the fact that circuit simulation is slow, thus rendering generation of a large-scale dataset as time-consuming and that most circuit designs are proprietary IPs within individual IC companies, making it expensive to collect large-scale datasets. As a result, RL engine that know nothing about analog design is proposed in [49]. The RL agent first learns to meet hard constraints, and then learns to optimize the targets. Compared with grid search-aided human design, L2DC

can achieve  $250\times$  higher sample efficiency with comparable performance. The RL agent generates circuits' data by itself and learns from the data to search for best parameters. The RL agents were trained from scratch without giving it any rules about circuits. In each iteration, the agent obtains observations from the environment, produces an action (a set of parameters) to the circuit simulator environment, and then receives a reward to optimize the desired FOMs composed of several performance metrics. By maximizing the reward, RL agent can optimize the circuit parameters. The system was demonstrated on several Op-Amp examples successfully.

A ML analog circuit sizing framework that uses deep reinforcement learning approach is presented in [50]. Policy gradient neural network (PGNN), is built to predict the changes of circuit parameter values, which yields the probability distribution over all valid actions. The objective of the agent is to learn sequences of actions that will maximize its expected cumulative rewards. Considering the circuit sizing problem, circuit parameters (device dimensions, voltages, capacitances, etc.) are encoded as the states. Thereby, the actions (increment and decrement) are defined as the change of those circuit variables. To adjust the amount of change, two other parameters are also defined: change rate and change capacity. Once an action is taken, it should be evaluated to ensure whether it satisfies the design constraints. Actions that violate the design constraints are directly eliminated without running any simulation, resulting in significantly reducing in the execution time. Then, the design objectives are encoded into the rewards in order to manage the learning. To classify the objectives, positive and negative weights are assigned for each objective, where a positive coefficient intends to maximize the objective while the negative one aims to minimize it. Before the simulator, a rough pre-evaluation is performed by a symbolic filter, where the small signal parameters of devices are extrapolated by mapping of circuit variables, which then turns into performance estimations. If the candidate is verified by the symbolic analysis, it is simulated through SPICE and rewarded by considering with the design objectives and constraints. Otherwise, the reward is set to zero. To demonstrate the proposed approach, a folded cascode amplifier is optimized, where dc gain, the bandwidth, the phase and the gain margins are chosen as the design specifications. According to the reported results, the proposed tool is able to optimize the circuit satisfying all the design specifications.

Post-layout circuit parameters are found for a given target specification using deep RL [51]. The approach can be classified under two steps: training and deployment. In the first step of training, the performance trajectories are obtained for a given problem, where the objective specifications are obtained via SPICE simulations and given to the RL. Then, the RL agent observes the state of the environment and operates according to its knowledge at each step. The neural network uses the observed and targeted specifications as well as design parameters to decide the action whether to increment, decrement, or retain the same value for each circuit parameter. The environment returns a new state for calculation of the reward. The agent iteratively operates through a trajec-

tory of multiple environment steps, accumulates the rewards at each step, and updates the NN weights until the objective criterion is met or the maximum iteration count is reached. During the rewarding process, hard design constraints and objectives that are being minimized are taken into account. The reward increases as the RL agent's observed performance gets closer to the target specification. The training terminates once all targeted specifications are satisfied. During deployment, the trained agent is used to generate trajectories for new specifications. Moreover, the proposed approach is combined with a layout generator tool to perform the post-layout extracted simulations. Once learning is performed at schematic level; it is directly transferred to a different environment. Here, the layout generator is employed and the parasitic extracted netlist is given to the trained to deploy the agent. To demonstrate the proposed approach, a transimpedance amplifier, a two stage OTA, and a two stage OTA with negative  $g_m$  load are optimized. The results indicates that the approach is almost  $40\times$  more sample efficient than a typical genetic algorithm. Also, the proposed post-layout simulation framework is  $9.6\times$  more sample efficient than the state-of-the-art thanks to transfer of learning at the schematic level.

## 5. Machine Learning in Analog/RF Layout Synthesis

The widespread application of ML to different areas, including analog/RF IC layout automation, opens new perspectives for developing push-button solutions that simultaneously incorporate legacy data or expert design insights in a manner that was not possible in the previous generations of EDA tools. These recent ML applications for layout automation range from placement tools to routing drafters, but also, pre- and post-placement processing. Table 3 summarizes the different ML techniques for layout automation that are overview within this section.

### 5.1. Pre-Placement Processing

An expert IC designer can examine a circuit schematic and instinctively recognize several building blocks (e.g., differential pairs, level-shifters, current mirrors, etc.) formed by various basic primitives (blocks), based solely on his prior experience. These primitives and larger building blocks define more complex structures (e.g., operational amplifiers or voltage-controlled oscillators), and ultimately, are built up in the hierarchy to form complete systems (e.g., an analog-to-digital converters or RF transceivers). Existing methodologies able to recognize such structures are usually based on graph representations of the netlist [58], and, take advantage of its sub circuits defined explicitly. However, while subgraph isomorphism operations are somewhat possible at building-block level, the number of combinations becomes impractical at higher levels, as a countless number of circuit/system variations can be implemented for similar functionalities. ML are opening new possibilities in this recognition, as proposed in ALIGN (Analog Layout, Intelligently Generated from Netlists) [53]. This framework receives as input an unannotated netlist, and, identifies hierarchies to recognize the



Table 3: Summary of the ML applications for layout automation.

Reference	Design Step	Model	Training	Contributions
[53]	Building block identification	n/d	n/d	Alternative to subgraph isomorphism
[54]	Placement	ANN with $n \times W \times H$ neurons	Weights assigned by hill-climbing	ANN used as discrete $W \times H$ layout plane
[55]	Placement	ANN with 3 hidden layers	Supervised	Reproduces legacy data patterns
[56]	Placement	ANN with 4 hidden layers	Unsupervised	Trained with sizing data only
[57]	Well definition	GAN	Supervised	Reproduces legacy data patterns
[24]	Routing	ANN used as VAE	Semi-supervised	Acquired knowledge used to assist A* search

building blocks of the design so that they may be appropriately optimized. The primitives at the lowest levels are set, and then, ML handles the ambiguities in the way these primitives are assembled, attempting to mimic the expert IC designer.

### 5.2. Placement by ANNs

Analog/RF IC layout design is usually split into placement and routing. In the placement task, many requirements must be considered to produce a robust floorplan solution against parasitic structures or process variations, e.g., minimizing the layout area while satisfying several topological constraints that span from symmetry, proximity, or boundary, among others without hindering its potential to be routed effectively. Analog/RF IC placement automation has been intensively studied in the last few decades, and the works proposed usually follow a descriptive approach or an optimization-based approach. Descriptive procedural [59] or template-based [60] approaches are applied with a moderate level of success on migration of legacy layouts [61] or layout-aware sizing methodologies [62], where fast generation techniques in-the-loop are required to be executed in-the-loop. Optimization mechanisms, mostly based on simulated annealing kernels that either change the absolute coordinates of the cells on a 2-dimensional plane [63] or perturb a topological representation that encodes the floorplan [64]. While presenting a reduced setup time, its execution can be time-consuming. This trade-off between setup time and computational efficiency marked the previous generations of automatic placement tools, a reality that ML promises to change by pursuing, for the first-time, flexible push-button solutions. An initial approach based on an ANN architecture was proposed in [54], whose goal was to place the cells within a segmented plane of fixed size  $W \times H$ . A mean-field neural network with  $n \times W \times H$  neurons, where  $n$  are the number of cells to be placed, is used. Each neuron is assigned with a binary output value, whose '1' corresponds to the assignment of that cell to a respective panel of the  $W \times H$  plane. While an ANN structure is used to represent the problem, the hill-climbing algorithm is still used to solve its gradients as a new set of cells' dimensions are requested. The energy function being

minimized weights simultaneously several factors, including interconnection estimates between pairs of cells, overlap, symmetry, proximity and boundary, the later used to keep cells inside the  $W \times H$  plane. Recently, ANNs were used to pursue the knowledge mining route on placement automation [55]. A model with 3 hidden layers with 250 to 1000 neurons each was used to learn the design patterns (including the inherent topological constraints) of more than 10.000 different placement solutions with conflicting guidelines among them (validated symmetry and current-flow constraints) of the same circuit topology. The output layer is used to provide the exact placement coordinates of each cell of the circuit for any given sizing in the 2-D plane, as illustrated in Fig. 12. The model training is made by minimizing the mean squared error (MSE) between the predicted floorplan and its corresponding solution from the training set. Unlike previous deterministic knowledge mining approaches [65], this end-to-end approach does not require to define any kind of tie breaker manually, with the trained model embedding reusable design patterns that generalize beyond the training data, and, provide different placement alternatives (e.g., different aspect-ratios) for the same circuit sizing at push-button speed.

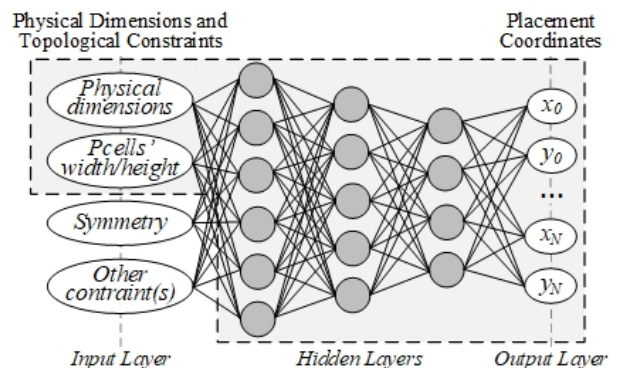


Figure 12: ANN architecture used to solve the map from the physical and effective Pcells' dimensions to the placement coordinates, where topological constraints are implicit (shaded box) [55]. Topological constraints can be added in the input layer for topological loss function training [56].

In [56], a nonlinear ANN model is also applied but used to

train topological loss function on legacy sizing data only, that learns how to fulfill the topological constraints. It promotes the application of the acquired "knowledge" instead of penalizing it with high MSE errors as in [55]. Additionally, the work took one step further towards the prediction of floorplan solutions for circuit topologies, which the model has never been trained before, by supporting different circuit topology encodings (with different number of devices) on the input layer of the same ANN, reusing knowledge among topologies.

### 5.3. Post-Placement Processing

When designing a floorplan, experienced IC designers often have the locations of the well regions in mind, i.e., areas where the doping is uniformly shared among a group of devices. Although abutment techniques help [60], embedding this information during automatic placement methodologies is not always straightforward. In WellGAN [57], n-type well definition is left for post-placement placement, where a generative adversarial network (GAN) is used to mimic the behavior of experienced designers, by reusing the knowledge embedded on previous manually-crafted layouts. To extract the information from legacy data, the oxide diffusion (OD) layer of layouts is used as input pattern, and, an RGB channel encoding is used to differentiate the ODs, i.e., OD inside n-type wells (red) and OD outside n-type wells (green), while wells are assigned to the blue channel. Thus, after training, the model receives as input images with patterns R and G, and outputs images with RGB channels. To convert this information into a floorplan, a post-refinement stage is used to rectilinearize and legalize the wells based on these guiding regions, fulfilling design rules (e.g., minimum spacing, enclosure, width, and area design rules). This approach was incorporated on MAGICAL framework [66].

### 5.4. Routing

Routing has a determinant impact on the post-layout performance of analog/RC ICs, especially at deep nanometer integration nodes, where the increasing congestion causes disproportionate growth of the interwire capacitances. Different types of automatic analog/RF IC routers were proposed in the last few years, based on: (1) procedures [67] or template descriptions [68]; (2) heuristics that encode different routing techniques as constraints (e.g., wiring symmetry), and then, path-finding algorithms (e.g., maze search [69]) are applied to draw a wire that connects two different terminals of a net in the presence of obstacles; (3) integer linear programming (ILP) [70], by constructing a priori high quality routes for individual nets, and then, using ILP to commit each net to only one of its candidate routes; and, optimization [71], where an evolutionary algorithm performs structural and layer changes in the physical representation of a population of independent routing solutions, allowing to optimize all wires of all nets simultaneously. Still, due to its high setup configuration and customization, only procedural or template-based approaches are usually capable of reproducing the IC designer preferences. Thus, automatic routing

methodologies have not been popular among industrial IC design environments. GeniusRoute [24] attempts to extract routing strategies of legacy layouts and apply the acquired knowledge in guiding a routing algorithm. Similar to WellGAN, in the pre-processing of the training data, placement and routing are represented as 2-D images, where routing-relevant information is extracted. For each data point, the pins of the entire design and pins for the given net are mapped into two separate  $64 \times 64$  channels. These channels are then used on a semi-supervised model training, where, first, the ANN used as variational autoencoder (VAE) is initialized in an unsupervised fashion, and only after, supervised decoder training. GeniusRoute then uses a classical A\* path-finding algorithm assisted by the model's inference, which generates the routing probability map to guide search. Traditional rip-up and reroute techniques are still used to ensure that a successful solution is attained. However, the legacy design patterns will be present on the automatically generated routing solutions.

## 6. 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 [83]. Table 4 summarizes the different ML techniques for IC fault testing and diagnosis that are overview within this section. In [72] a fault-model-based diagnosis for analog ICs was proposed. The method is based on an ML-based defect filter [73] 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 [80], 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

Table 4: Summary of the ML applications for Analog IC fault testing, diagnosis and calibration.

Reference	Application	Method(s)	Contributions
[17]	Fault Dianosis	MLP	Haar wavelet followed by kPCA to reduce dimensionality of features
[72],[73]	Fault Dianosis	SVM	A defect filter identifies hard and soft faults, and, for the soft faults inverse regression is used to locate the fault cause
[74]	Fault Dianosis	ANN	Wavelet and PCA to reduce dimensionality of features
[75]	Fault Dianosis	ANN	Dictionary and PCA reduce dimensionality of features
[76]	Fault Dianosis	Fisher DT	LDA to improve class separability while compressing the feature space
[77]	Fault Dianosis	Naive Bayes	Wavelet followed by kLDA
[78]	Fault Dianosis	DBN	End-to-end learning simplifies the feature engineering
[79]	Fault Dianosis	DBN	End-to-end with integrated random sampling for data gathering
[80]	Remaining Useful Performance	Kernel RVM	PSO is used to train a RVM that predict the trajectories of the circuits health and predict the remaining useful performance
[81]	Test Set Compression	ONN	NSGA optimization to select the smallest set of features sufficient to diagnose the CUT, resulting in a cheaper test procedure
[25]	One-Shot Calibration	ANN	Post-fabrication calibration to counter performance deviation due to fabrication in a single calibration step
[82]	Post-Layout Modeling	BMF	Uses cheap pre-silicon simulation data, together with a small dataset of fabricated circuits for efficient post silicon modeling

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). Experiments with a Sallen–Key bandpass 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.

### 6.1. Pre-Processing with Dimensionality Reduction

As the dimensionality of feature space increases, fault diagnosis methods started to use longer data processing pipelines with a structure similar to that shown in Fig. 13. These methods start by collecting the raw data that is then pre-processed and transformed, e.g., wavelet transformations are a common approach to compress the raw data into smaller but significant coefficients [74]. The next step is dimensionality reduction, which is done with PCA or LDA and their kernel extensions, and finally, the classifier is trained.

In [75], the authors define the fault dictionary exercising each analog structure with different input signals. This fault dictionary is essentially a table that contains all the fault characteristics of the circuit-under-test (CUT), and, used as a lookup table. PCA is then used to perform an orthogonal transformation of the fault space to a lower dimension, and a quantitative measure of distance separation, designated Bhattacharyya coefficient, is used to selection of the relevant features. The ANN is trained in this reduced space, and then, used to classify the applied input signal in the CUT, as faulty or fault-free output. A distinguishing factor in this work is the use of the rider optimization algorithm to train the ANN. This algorithm follows the analogy that a group of riders race towards a target objective. The approach was tested in different analog structures, including a triangular wave generator, a low noise bipolar transistor amplifier, a differentiator, and

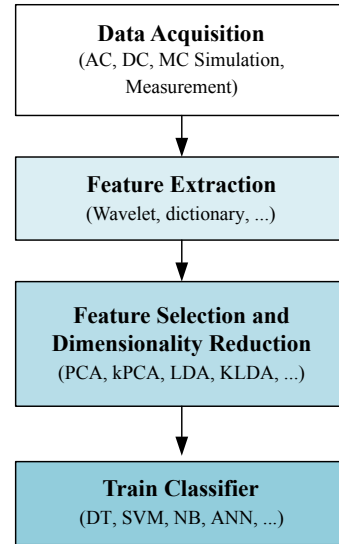


Figure 13: Common pipeline used on recent fault diagnosis systems.

a solar power converter. The experiments showed promising accuracy levels above 95%. [17] applied Haar wavelet transform to obtain the coefficients for low- and high-frequency components of the time-response. Then, a modified kernel PCA that selects the kernels to maximize class separability following a criterion similar to LDA. The authors use an MLP multi-class classifier with a one-hot encoding at the output. The cost function optimized during training is not specified. The method is applied for a single and double fault model on a Sallen-Key bandpass filter, a biquad high-pass filter, and a nonlinear rectifier circuit, achieving a 100% accuracy rate on the two first with just 5 principal components. In [85] sinusoidal excitation [76] is used to gather voltage amplitude features for the CUT. Then LDA is used to increase class separability, and an oblique Fisher decision tree [86] is induced.

The authors also use fuzzification to soften the hard decision criteria of the DT and increase the performance of the classifier. The method was applied to an active filter and an audio amplifier circuit. In [77], the authors further extend the data processing pipeline that starts from raw time signals that describe the faults. The signals are then subject to cross wavelet transform to obtain time-frequency coefficients that produce time-frequency matrix representations. These matrices may present redundant patterns. A variation of the local binary pattern that considers the 8-Kirsch masks identifies repeating patterns regardless of rotations. Then, the extracted features are selected. First features are selected using the Hilbert-Schmidt independence criterion. Then kernel LDA produces lower dimensionality features that are used to train a Naive Bayes classifier. Despite the simplicity of classifier Naive Bayes, the involved pre-processing and feature selection scheme shows promising results on the two analog circuits tested. As a downside, the feature pre-processing is computationally demanding, which limits its use in real-time monitoring systems. The authors also identify some limitations in finding the most discriminative features of nonlinear circuits.

### 6.2. Shorten the ML pipeline with DBNs

Unlike the previously described fault diagnose approaches that measure, analyze, and collapse the high-dimensional raw systems' output signals (obtained as time-domain, frequency-domain, or time-frequency simulation) into a lower-dimensional feature set to help to isolate the fault. In [78], DL is applied to identify a hierarchical structure that captures the different levels of semantic representations of the raw output signals. As only the response of the circuit under test is monitored, it does not suffer from the accessibility problems to internal nodes of analog ICs, common to equation-based approaches. A Gaussian-Bernoulli (GB)-DBN classifier is trained in a semi-supervised learning approach. This framework, shown in Fig. 14, is composed of two training phases on raw data, pre-training, performed in an unsupervised fashion and independently on the different layers of a stacked RBM, and then, fine-tuning, performed in a supervised fashion, where all the RBM layers are fine-tuned with respect to the classification errors. The objective is to obtain a latent space that enlarges the interclass distance for different fault classes and reduces the intraclass distance among each fault, improving the classifier's ability to identify them. This approach was experimentally validated on two typical analog filter circuits, learning more discriminative features than traditional feature extraction, and thus, producing better diagnostic results with a smaller dataset. [79] adaptively extracts features from Monte Carlo sampling and then uses them to train a general-purpose DBN classifier, while simultaneously embedding dimensionality reduction into it. The sampling is embedded into the methodology flow, discarding most of the human involvement in feature preparation. The complete flow is composed by the following steps: (1) firstly, for the CUT are identified the potential fault modes,

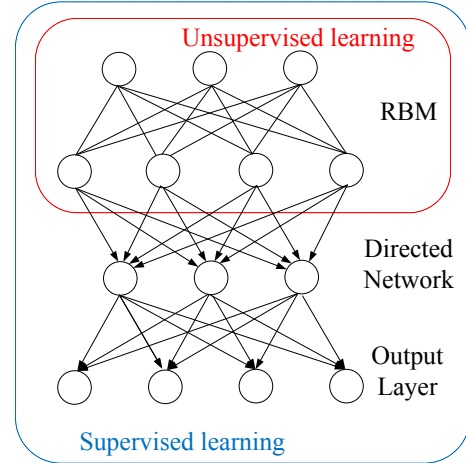


Figure 14: DBN structure, showing hidden RBM that are trained in unsupervised learning.

nominal parameters, and their tolerances; (2) for each potential fault mode obtain via simulation the raw time-series signals, from time-domain transient, and Monte Carlo analysis; (3) build the dataset using the raw time-series signals, where each time-series signal of each Monte Carlo sample corresponds to one input instance; (4) data engineering for ML model construction and separation of the dataset into training and test sets; (5) used the data to train the DBN, which is accomplished by a layer-by-layer unsupervised pre-training stage and a fine-tuning stage using backpropagation; and (6) the DBN is then used to for fault classification. The experimental results conducted over a Sallen-Key bandpass filter and four-opamp biquad high-pass filter, show higher classification accuracy when compared with existing data-driven methods, with lower dependency on the data. In this method, the use of raw time-series signals directly enables the detection of faults whose effects are only reflected in segments of the time series output signals.

### 6.3. Post-Fabrication Automatic Calibration

In other works, more than fault identification and diagnosis, ML is used for specification testing and calibration of analog circuits against process variation. These methods ensure quality electronic devices and systems while reducing test costs. The latter also increases production yield. [81] uses ML to identify a subset of tests that is sufficient for performance testing. A circuit that fails any test in the subset is immediately discarded. Those that pass all tests in the subset are then classified, using KNN and ontogenic neural networks trained using the tests on the subset. The selection of the subset of tests is made by NSGA-II multi-objective optimization that explores the trade-off between test cost and accuracy. The method was applied on an RF device and showed that testing the RF circuit without RF specific testing equipment was able to detect most of the failing devices. Including some RF testing allowed the method to detect all failing devices with less than 10% of the cost. In [25], the authors propose a One-Shot calibration mechanism, similar to the one in Fig. 15 that

relies on an ANN predictor trained to estimate the performance given the test measurements and the settings of the tuning knobs. The dataset is built of 67 fixed combinations of turning knobs for each measured circuit. Statistical sampling mechanisms [87] can also be used to build the training set. Once the system is in place, the CUT is measured, if performance is not within specifications, it used the trained regressor to search for the best settings of the tuning knobs. The method was applied to an RF power amplifier and was able to recover 96% of devices failing specifications. How-

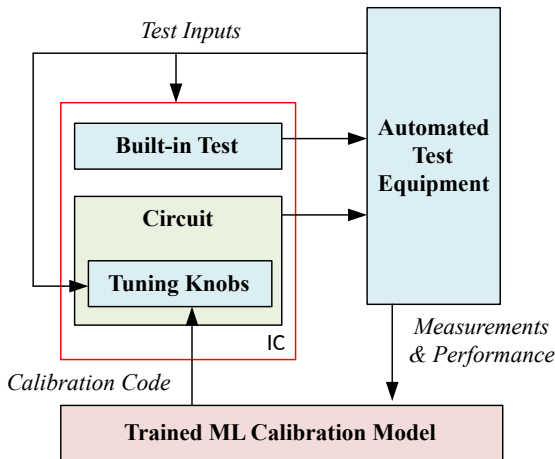


Figure 15: One-shot calibration of fabricated circuits.

ever, in practice, the number of silicon measurements that can be made to create the training data is limited. Still, accurate modeling of parasitic and losses related to all the testing setup is intricate and prevents the direct use of simulation-based data to train the model. Nonetheless, there is valuable information that can be gathered from simulations. Heeding to this, in [82], the authors used Bayesian model fusion, as indicated in Fig. 16. Simulation-based data defines a prior that reduces the amount of post-fabrication measurements needed to build an accurate performance regressor.

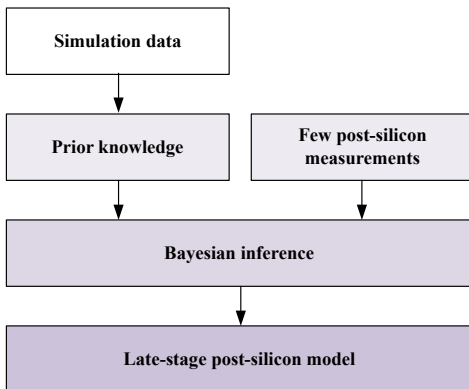


Figure 16: BMF for post-layout modeling.

## 7. Conclusion and Future Directions

Recently, ML-based techniques have been efficiently utilized in several applications, where enhanced learning capability makes them unique to solve any complex/nonlinear problem. IC design has also benefited from ML techniques at different design levels, from device modeling to test of manufactured chips.

The attempts in device/circuit/system modeling have aimed to generate accurate models at different levels of abstraction and replace the simulator, especially in RF applications, by these models; hence, the human effort and the design time can be mitigated. Furthermore, creating technology-independent models can enable their use in further technologies for a given problem. Considering the ascending popularity of ANNs, researchers may revisit their application to analog/RF IC modeling, where variability and reliability problems have not been fully addressed in this manner. Furthermore, the capability of technology-free modeling of CMOS devices via ANNs will undoubtedly contribute to appearance of analog/RF IPs in EDA tools.

ML-based modeling of analog/RF devices and circuits has paved the way for ML-based circuit sizing. The main problem with the reported sizing approaches is the trade-off between the accuracy and the efficiency of the synthesis process. Symbolic/analytical model-based approaches are quite efficient, but, they suffer from poor accuracy of those models, while simulation-based approaches are quite precise. However, they are computationally expensive. Incorporation of ML modeling with conventional sizing algorithms has become a remedy for this bottleneck, in which the circuit performances are accurately modeled via ML techniques, and the generated models are employed during the evaluation of circuits. Furthermore, putting this mechanism into the optimization loop may move the efficiency further, where a portion of the initial solutions are used as the filtered data (satisfying all constraints) to train the model; thus, the learning phase is automatically carried-out without any external effort. ML-based circuit synthesis also enables fast and accurate search of the vast analog/RF design space for multi-/many-objective optimization tools. Recent progress in this area shows that the proposed approaches are still immature, and several developments are needed, i.e., integration process, selection of NN, and intermediate model verification, etc. Moreover, the efficiency problems with variability- and reliability-aware circuit synthesis can be solved via ML-based synthesis, in which the generated models may be reused during these analyses. Hierarchical synthesis of analog/RF circuits will be an important part of the future directions. The feasibility of the conventional bottom-up approaches may become much more efficient, since the transitions from different levels of hierarchy can be facilitated through the developed accurate ML-based models.

Automatic layout generation for analog/RF ICs suffers from similar problems with sizing approaches. Several ML-based methodologies, mostly on floorplan design, have been proposed. While the previous generation of au-

automatic placement tools was marked by the high setup (descriptive/template-based approaches) versus high computational effort (optimization-based approaches) trade-off, ML-based methodologies, mostly via ANNs, are attempting to change this reality by pursuing flexible push-button solutions. On top of it, the models can also be based on previous legacy knowledge. Future research directions in this field will most likely be focused on how to reuse those design patterns learned and generalize them well beyond the training data. These generalization capabilities are also expected to be achieved for newer circuit topologies. Still, an actual problem is how to generate a dataset robust enough for that purpose. While acquiring robust sizing data for several topologies is still quite feasible, acquiring validated legacy layouts is not straightforward. The approach taken recently was to use other EDA tools to generate synthetic data. Nonetheless, some (error-prone) mechanism or human inspection is still necessary to consider them "legacy-proved". For automatic routing, the application of ML is still taking its first steps, yet promising. Before working on the proper deep models, the most urgent matter to be researched is data engineering, how to concisely and accurately represent routing data on a dataset, which data is relevant to be fed to the model, and, how far it could be generalized. The problems of acquiring robust legacy data previously found on placement, are only further aggravated here. While a handful of robust manually routed designs can still be acquired and used, the necessity of tens of thousands will ultimately rely on previous EDA tools, e.g., template-based, heuristic-based or ILP routing procedures.

The use of ML for fault diagnosis is well established in the research community. Several classification methods appear in the literature. There is no particular method standing out, but supervised learning is the most common approach. Fault diagnosis ML systems often show a pipeline that includes feature transformation and pre-processing, such as wavelet transform, discriminative feature selection, such as LDA or Kernel LDA, and the classification algorithm, such as Naive Bayes, SVM, or ANNs. A few works explore semi-supervised learning, using DBNs that embed the feature selection in the classifier. However, the latent variables are more obscure and harder to understand their physical meaning. Future development for ML-based fault diagnosis should strive further to address fault localization, aging, and time-dependent performance effects, and identify the right strategy for hyperparameter tuning. The hyperparameter tuning and training strategies have a significant impact on the performance of the classifiers, but there are not many criteria given in the literature. Another major challenge is to decrease the computation requirements, as the current state-of-the-art approaches are too sophisticated for real-time or embedded applications.

ML techniques can also be employed to ease the transitions between design levels, e.g., incorporating layout information into sizing, ultimately, leading to widespread use of analog/IPs. Besides, the use of ML in analog/RF IC design is still in its infancy and there is room for further developments. The aforementioned future directions are not exclusive and

many new developments can be expected in the near future.

## Acknowledgement

This work is funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the project UIDB/50008/2020, including internal research projects HAICAS (X-0009-LX-20) and LAY(RF)<sup>2</sup> (X-0002-LX-20).

## References

- [1] K. P. Murphy, *Machine learning: a probabilistic perspective (adaptive computation and machine learning series)*. MIT Press., 2012.
- [2] M. Bayes and M. Price, "An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton," *Philosophical Transactions (1683-1775)*, vol. 53, pp. 370–418, 1763.
- [3] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
- [4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [5] J. Hastie, Trevor, Tibshirani, Robert, Friedman, *The Elements of Statistical Learning The Elements of Statistical Learning Data Mining, Inference, and Prediction, Second Edition*, 2009.
- [6] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information science and statistics. Springer, 2007, vol. 16, no. 4.
- [7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [8] R. S. Sutton and A. G. Barto, *Reinforcement Learning, Second Edition*, 2018.
- [9] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Networks*, vol. 21, no. 4, pp. 682–697, 2008.
- [10] D. Silver and et. al., "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.
- [11] V. Mnih and et. al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [12] Z. H. Zhou and et. al., "Big data opportunities and challenges: Discussions from data analytics perspectives [Discussion Forum]," *IEEE Computational Intelligence Magazine*, vol. 9, no. 4, pp. 62–74, 2014.
- [13] A. Zhang, C. Chen, and B. Jiang, "Analog circuit fault diagnosis based UCISVM," *Neurocomputing*, vol. 173, pp. 1752–1760, 2016.
- [14] A. Canelas and et. al., "FUZY: A Fuzzy C-Means Analog IC Yield Optimization using Evolutionary-based Algorithms," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 0070, no. 3, 2018.
- [15] T. Pessoa and et. al., "Enhanced analog and RF IC sizing methodology using PCA and NSGA-II optimization kernel," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2018, pp. 660–665.
- [16] B. Schölkopf and A. Smola, *Learning with Kernels | The MIT Press*, 2001.
- [17] Y. Xiao and Y. He, "A novel approach for analog fault diagnosis based on neural networks and improved kernel PCA," *Neurocomputing*, vol. 74, no. 7, pp. 1102–1115, 2011.
- [18] W. He, Y. He, B. Li, and C. Zhang, "A Naive-Bayes-Based Fault Diagnosis Approach for Analog Circuit by Using Image-Oriented Feature Extraction and Selection Technique," *IEEE Access*, vol. 8, pp. 5065–5079, 2020.
- [19] T. McConaghy and et. al., "Automated extraction of expert knowledge in analog topology selection and sizing," in *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD, 2008*, pp. 392–395.
- [20] R. El-Adawi and M. Dessouky, "Monte Carlo general sample classification for rare circuit events using Random Forest," in *14th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, 2017.

- [21] M. Ding and R. Vemuri, "An active learning scheme using support vector machines for analog circuit feasibility classification," in *18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design*. IEEE, 2005, pp. 528–534.
- [22] A. Suissa and et. al., "Empirical method based on neural networks for analog power modeling," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 5, pp. 839–844, 2010.
- [23] N. Kahraman and T. Yildirim, "Technology independent circuit sizing for fundamental analog circuits using artificial neural networks," in *Ph. D. Research in Microelectronics and Electronics (PRIME)*. IEEE, 2008, pp. 1–4.
- [24] K. Zhu and et. al., "Geniusroute: A new analog routing paradigm using generative neural network guidance," in *Proceedings of International Conference on Computer Aided Design (ICCAD)*, 2019.
- [25] M. Andraud, H. Stratigopoulos, and E. Simeu, "One-Shot Non-Intrusive Calibration Against Process Variations for Analog/RF Circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 11, pp. 2022–2035, 2016.
- [26] V. Ceperic and A. Baric, "Modeling of analog circuits by using support vector regression machines," in *Proceedings of the 2004 11th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, 2004, pp. 391–394.
- [27] M. Grabmann, F. Feldhoff, and G. Gläser, "Power to the model: Generating energy-aware mixed-signal models using machine learning," in *16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE, 2019, pp. 5–8.
- [28] Q.-J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design—from theory to practice," *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, no. 4, pp. 1339–1350, 2003.
- [29] V. K. Devabhaktuni and et. al., "Neural networks for microwave modeling: Model development issues and nonlinear modeling techniques," *International Journal of RF and Microwave Computer-Aided Engineering: Co-sponsored by the Center for Advanced Manufacturing and Packaging of Microwave, Optical, and Digital Electronics (CAMPmode) at the University of Colorado at Boulder*, vol. 11, no. 1, pp. 4–21, 2001.
- [30] P. M. Watson and K. C. Gupta, "Design and optimization of CPW circuits using EM-ANN models for CPW components," *IEEE Transactions on Microwave Theory and Techniques*, vol. 45, no. 12, pp. 2515–2523, 1997.
- [31] M. G. Passos, P. d. F. Silva, and H. C. Fernandes, "A RBF/MLP modular neural network for microwave device modeling," *International Journal of computer science and network security*, vol. 6, no. 5A, pp. 81–86, 2006.
- [32] Y. Harkouss and et. al., "The use of artificial neural networks in nonlinear microwave devices and circuits modeling: An application to telecommunication system design (invited article)," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 9, no. 3, pp. 198–215, 1999.
- [33] H. Liu and et. al., "Remembrance of circuits past: macromodeling by data mining in large analog design spaces," in *Proceedings of the 39th Annual Design Automation Conference*, 2002, pp. 437–442.
- [34] G. Islamoglu and et. al., "Artificial Neural Network Assisted Analog IC Sizing Tool," in *16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE, 2019, pp. 9–12.
- [35] G. Wolfe and R. Vemuri, "Extraction and use of neural network models in automated synthesis of operational amplifiers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 198–212, 2003.
- [36] W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Multi-objective bayesian optimization for analog/rf circuit synthesis," in *Proceedings of the 55th Annual Design Automation Conference*, 2018, pp. 1–6.
- [37] Z. Gao, J. Tao, F. Yang, Y. Su, D. Zhou, and X. Zeng, "Efficient performance trade-off modeling for analog circuit based on bayesian neural network," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–8.
- [38] P.-C. Pan, C.-C. Huang, and H.-M. Chen, "Late Breaking Results: An Efficient Learning-based Approach for Performance Exploration on Analog and RF Circuit Synthesis," in *56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–2.
- [39] E. Kaya, E. Afacan, and G. Dundar, "An Analog/RF Circuit Synthesis and Design Assistant Tool for Analog IP: DATA-IP," in *15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE, 2018, pp. 1–9.
- [40] J. Tao, C. Liao, X. Zeng, and X. Li, "Harvesting design knowledge from the internet: High-dimensional performance tradeoff modeling for large-scale analog circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 1, pp. 23–36, 2015.
- [41] R. Vural and et. al., "Process independent automated sizing methodology for current steering dac," *International Journal of Electronics*, vol. 102, no. 10, pp. 1713–1734, 2015.
- [42] V. Bhatia and et. al., "Modelling a simple current to voltage converter using ANN," in *IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*. IEEE, 2016, pp. 1–4.
- [43] V. Bhatia, N. Pandey, and A. Bhattacharyya, "Modelling and design of inverter threshold quantization based current comparator using artificial neural networks," *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 6, no. 1, 2016.
- [44] A. Jafari, S. Sadri, and M. Zekri, "Design optimization of analog integrated circuits by using artificial neural networks," in *2010 International Conference of Soft Computing and Pattern Recognition*. IEEE, 2010, pp. 385–388.
- [45] N. Takai and M. Fukuda, "Prediction of element values of OpAmp for required specifications utilizing deep learning," in *International Symposium on Electronics and Smart Devices (ISESD)*. IEEE, 2017, pp. 300–303.
- [46] N. Lourenço and et. al., "On the exploration of promising analog ic designs via artificial neural networks," in *15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE, 2018, pp. 133–136.
- [47] E. Dumesnil, F. Nabki, and M. Boukadoum, "RF-LNA circuit synthesis using an array of artificial neural networks with constrained inputs," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 573–576.
- [48] N. Lourenço and et. al., "Using Polynomial Regression and Artificial Neural Networks for Reusable Analog IC Sizing," in *16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE, 2019, pp. 13–16.
- [49] H. Wang and et. al., "Learning to design circuits," *arXiv preprint arXiv:1812.02734*, 2018.
- [50] Z. Zhao and L. Zhang, "Deep reinforcement learning for analog circuit sizing," in *IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2020, pp. 1–5.
- [51] K. Settaluri and et. al., "AutoCkt: Deep Reinforcement Learning of Analog Circuit Designs."
- [52] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei, "Automatic differentiation variational inference," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 430–474, 2017.
- [53] K. Kunal and et. al., "Align: Open-source analog layout automation from the ground up," in *Proceedings of the 56th Annual Design Automation Conference (DAC)*, 2019, pp. 1–4.
- [54] R. He and L. Zhang, "Artificial neural network application in analog layout placement design," in *Canadian Conference on Electrical and Computer Engineering*. IEEE, 2009, pp. 954–957.
- [55] D. Guerra and et. al., "Artificial Neural Networks as an Alternative for Automatic Analog IC Placement," in *16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. IEEE, 2019, pp. 1–4.
- [56] A. Gusmao and et. al., "Semi-Supervised Artificial Neural Networks towards Analog IC Placement Recommender," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5.
- [57] B. Xu and et. al., "Wellgan: Generative-adversarial-network-guided well generation for analog/mixed-signal circuit layout," in *56th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2019, pp. 1–6.
- [58] M. Eick and et. al., "Comprehensive generation of hierarchical placement rules for analog integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 2, pp. 180–193, 2011.
- [59] A. Bhaduri and et. al., "Parasitic-aware synthesis of RF LNA circuits considering quasi-static extraction of inductors and interconnects," in *The 47th Midwest Symposium on Circuits and Systems. MWSCAS'04*, vol. 1.

- IEEE, 2004, pp. 1–477.
- [60] R. M. Martins, N. C. Lourenço, and N. C. Horta, *Generating analog IC layouts with LAYGEN-2*. Springer Science & Business Media, 2012.
- [61] S. Bhattacharya, N. Jangkrasarn, and C.-J. Shi, “Multilevel symmetry-constraint generation for retargeting large analog layouts,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 945–960, 2006.
- [62] R. Martins and et. al., “Two-step RF IC block synthesis with preoptimized inductors and full layout generation in-the-loop,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 6, pp. 989–1002, 2018.
- [63] —, “Current-flow and current-density-aware multi-objective optimization of analog IC placement,” *Integration*, vol. 55, pp. 295–306, 2016.
- [64] A. Patyal and et. al., “Analog placement with current flow and symmetry constraints using PCP-SP,” in *55th ACM/ESDA/IEEE Design Automation Conference (DAC)*. IEEE, 2018, pp. 1–6.
- [65] P.-H. Wu and et. al., “A novel analog physical synthesis methodology integrating existent design expertise,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 2, pp. 199–212, 2014.
- [66] B. Xu and et. al., “MAGICAL: Toward Fully Automated Analog IC Layout Leveraging Human and Machine Intelligence,” in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019, pp. 1–8.
- [67] E. Chang and et. al., “BAG2: A process-portable framework for generator-based AMS circuit design,” in *IEEE Custom Integrated Circuits Conference (CICC)*, 2018, pp. 1–8.
- [68] A. Unutulmaz, G. Dünder, and F. V. Fernández, “A template router,” in *20th European Conference on Circuit Theory and Design (ECCTD)*. IEEE, 2011, pp. 334–337.
- [69] E. Yilmaz and G. Dundar, “Analog layout generator for CMOS circuits,” *IEEE Transactions on computer-aided design of integrated circuits and systems*, vol. 28, no. 1, pp. 32–45, 2008.
- [70] C.-Y. Wu, H. Graeb, and J. Hu, “A pre-search assisted ILP approach to analog integrated circuit routing,” in *33rd IEEE International Conference on Computer Design (ICCD)*, 2015, pp. 244–250.
- [71] R. Martins, N. Lourenco, and N. Horta, “Routing analog ICs using a multi-objective multi-constraint evolutionary approach,” *Analog Integrated Circuits and Signal Processing*, vol. 78, no. 1, pp. 123–135, 2014.
- [72] K. Huang, H. Stratigopoulos, and S. Mir, “Fault diagnosis of analog circuits based on machine learning,” in *Design, Automation Test in Europe Conference Exhibition (DATE 2010)*, 2010, pp. 1761–1766.
- [73] H. Stratigopoulos and et. al., “Defect filter for alternate rf test,” in *14th IEEE European Test Symposium*, 2009, pp. 101–106.
- [74] M. Aminian and F. Aminian, “A modular fault-diagnostic system for analog electronic circuits using neural networks with wavelet transform as a preprocessor,” *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 5, pp. 1546–1554, 2007.
- [75] D. Binu and B. S. Kariyappa, “RideNN: A New Rider Optimization Algorithm-Based Neural Network for Fault Diagnosis in Analog Circuits,” *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 1, pp. 2–26, 2019.
- [76] Feng Li and Peng-Yung Woo, “Fault detection for linear analog IC-the method of short-circuit admittance parameters,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 1, pp. 105–108, 2002.
- [77] W. He and et. al., “A Naive-Bayes-Based Fault Diagnosis Approach for Analog Circuit by Using Image-Oriented Feature Extraction and Selection Technique,” *IEEE Access*, vol. 8, pp. 5065–5079, 2020.
- [78] Z. Liu and et. al., “Capturing high-discriminative fault features for electronics-rich analog system via deep learning,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1213–1226, 2017.
- [79] G. Zhao and et. al., “A novel approach for analog circuit fault diagnosis based on Deep Belief Network,” *Measurement*, vol. 121, pp. 170–178, 2018.
- [80] C. Zhang and et. al., “A multiple heterogeneous kernel RVM approach for analog circuit fault prognostic,” *Cluster Computing*, vol. 22, no. 2, pp. 3849–3861, 2019.
- [81] H. Stratigopoulos and et. al., “RF Specification Test Compaction Using Learning Machines,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 6, pp. 998–1002, 2010.
- [82] F. Wang and et. al., “Bayesian Model Fusion: Large-Scale Performance Modeling of Analog and Mixed-Signal Circuits by Reusing Early-Stage Data,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 8, pp. 1255–1268, 2016.
- [83] V. Rajan, Jie Yang, S. Chakrabarty, and K. Pattipati, “Machine learning algorithms for fault diagnosis in analog circuits,” in *SMC’98 Conference Proceedings. IEEE International Conference on Systems, Man, and Cybernetics*, 1998, pp. 1874–1879.
- [84] M. E. Tipping, “Sparse bayesian learning and the relevance vector machine,” *J. Mach. Learn. Res.*, vol. 1, p. 211–244, 2001.
- [85] Y. Cui, J. Shi, and Z. Wang, “Analog circuits fault diagnosis using multivalued Fisher’s fuzzy decision tree (MFFDT),” *International Journal of Circuit Theory and Applications*, vol. 44, no. 1, pp. 240–260, 2016.
- [86] A. López-Chau and et. al., “Fisher’s decision tree,” *Expert Systems with Applications*, vol. 40, no. 16, pp. 6283–6291, 2013.
- [87] F. Cilici and et. al., “Efficient generation of data sets for one-shot statistical calibration of RF/mm-wave circuits,” in *16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, 2019, pp. 17–20.