Data-Driven Modeling of a Wideband Power Amplifier for Performance Prediction and Feature Analysis

Nishat Anjumane Salsabila
Electrical and Electronic Engineering (EEE)
Chittagong University of Engineering and Technology (CUET)
Chattogram, Bangladesh
nishatanjumane@gmail.com

***Abstract*—This paper presents a comprehensive study on modeling the gain and bandwidth of a power amplifier (PA) using machine learning (ML) techniques based on key circuit parameters. The input dataset comprises inductor values, MOSFET aspect ratios, and bias voltage (V_b), while the corresponding gain and bandwidth measurements are obtained from simulated results of the PA. Three different ML algorithms, Random Forest (RF), XGBoost, and Artificial Neural Network (ANN) are trained and rigorously evaluated to predict amplifier's performance. Comparative evaluation reveals that all models accurately capture the nonlinear input-output relationships, with the XGBoost model demonstrating superior predictive accuracy. Furthermore, permutation feature importance (PFI) analysis identifies the key design parameters that impact PA performance. This work highlights the potential of ML methods as effective and efficient tools for circuit performance prediction and design optimization, offering a robust alternative to traditional simulation-based approaches in wireless communication system design.**

***Index Terms*—Power Amplifier, Random Forest, Artificial Neural Network, XGBoost, Permutation Feature Importance.**

I. Introduction

The development of integrated PAs optimized for gain and bandwidth performance has propelled significant progress in wireless communication systems. Modern RF transmitters rely on PAs to ensure strong, high-quality signals that directly impact overall system efficiency and reliability. As device miniaturization and low-power operation become increasingly significant, accurately modeling PA characteristics from critical design parameters becomes vital for effective amplifier optimization [1].

Designing RF PAs with precise gain and bandwidth performance is challenging due to nonlinear device behaviors. Traditional tools such as Cadence, ADS, and HFSS require extensive simulation time and computational resources [2]. To address these challenges, ML methods, especially ANNs, offer a promising data-driven alternative way to model nonlinear circuit behavior, and reduce reliance on repeated simulations. Prior studies have utilized the Levenberg–Marquardt (LM) algorithm for amplifier modeling without matching networks and explored ANN-based models integrated with metaheuristic algorithms, validated using 3D simulations. These studies highlight the capability of ANNs to deliver efficient and accurate modeling solutions for complex amplifier designs [3] [4].

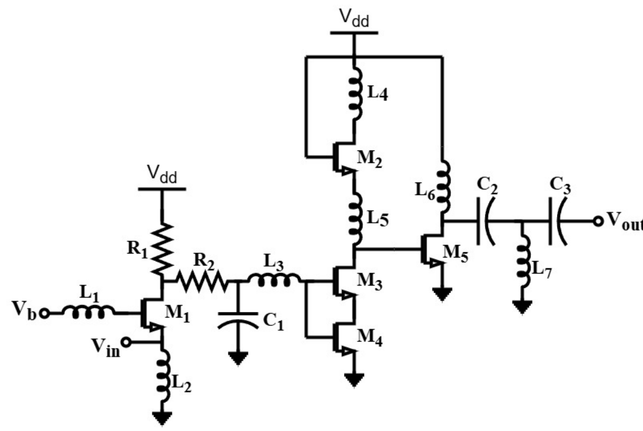

Fig. 1: Schematic of the wide band PA [10]

The amplifier design process is inherently resource-intensive and complex, underscoring the need for efficient techniques such as ANNs that can learn from data and model nonlinear behaviors effectively [5]. A different approach presented in [6] employed a set of ANNs integrated with a Genetic Algorithm (GA) for amplifier circuit synthesis. Although the GA-based method required a relatively high number of generations to achieve convergence, it still represented a significant step. In [7], CMOS Low Noise Amplifier (LNA) modeling with Adaptive Neuro-Fuzzy Inference System (ANFIS) demonstrated significantly lower prediction errors compared to Multilayer Perceptron (MLP) and Radial Basis Function (RBF) models. Recent studies, such as [8], have demonstrated efficient PFI-based feature selection techniques in real-world applications, while the Scikit-learn library in [9] provides a robust and accessible implementation for practical model interpretation. Collectively, these advancements position PFI as a valuable tool for understanding feature contributions and guiding optimization in ML-based circuit modeling frameworks.

This work tackles the challenge of modeling the PA's gain and bandwidth using ML, leveraging measured data from a previously designed PA to achieve realistic performance characterization, as shown in Fig-1 [10]. The input parameters considered include inductor values (L_1, L_5, L_6), MOSFET aspect ratios (M_4, M_5), and bias voltage (V_b) whereas gain and bandwidth are defined as output variables, as illustrated in Fig-2. This study builds predictive models using three different ML algorithm: Random Forest, XGBoost and ANN. The study aims to find the most accurate and efficient modeling approach for predicting PA performance by comparing these algorithms. The proposed methodology provides a data-driven alternative to traditional circuit simulations, significantly reducing design cycle time and computational costs. Moreover, it enables rapid design space exploration and facilitates the optimization of RF PAs for diverse wireless applications, ultimately enhancing system performance and lowering development effort.

This paper is organized as follows: Section II details the Methodology, covering the establishment and configuration of the three ML models used for prediction. Section III presents the Results and Discussion, offering a concise overview of the models' performance metrics, comparative analysis, and engineering insights. Finally, Section IV provides the Conclusion.

II. Methodology

Our previous work involved the design of a two-stage PA using 90 nm CMOS technology in Cadence Virtuoso, which delivered a gain of 21.90 dB over 350 MHz bandwidth, consumed 153.1 mW of power, and maintained stability [10]. Building on this, the present study focuses on developing an intelligent prediction system based on ML to replace traditional simulation methods, thereby significantly speeding up the design and optimization of high-frequency integrated circuits.

Incorporating body biasing and capacitor coupling, the PA was designed to deliver optimized gain, low power consumption and enhanced stability. Key performance metrics, such as gain and bandwidth, vary proportionally with the inductor values, MOSFET widths, and bias voltage over their respective ranges. Therefore, the selected inductor values, MOSFET widths, and bias voltage are treated as input variables, while gain and bandwidth serve as the output variables in the PA model, as illustrated in Fig-2.

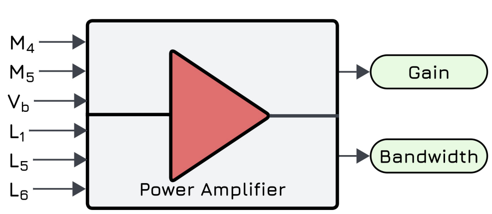

Fig. 2: Parameters considered for Machine Learning modeling of the PA

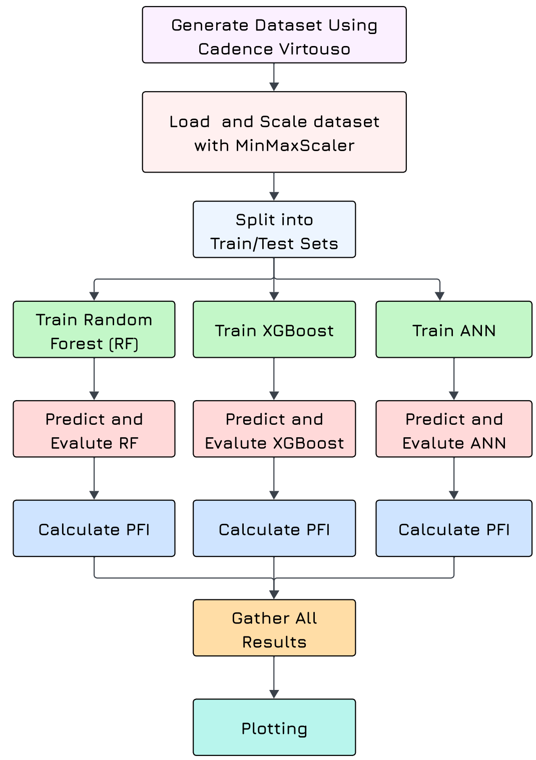

Fig. 3: Flowchart of Machine Learning work flow

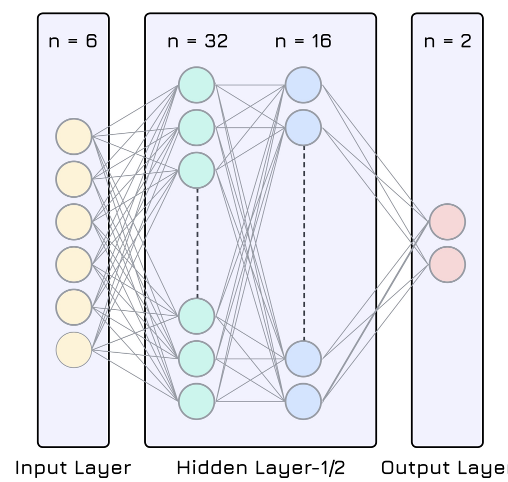

Fig. 4: Neural Network model developed for the PA.

TABLE I: Definition of Input Parameter Ranges and Step Sizes

Parameter	Unit	Lower Bound	Upper Bound	Step Size
V_b	V	0.20	1.0	0.1
M_4	μm	80	200	30
M_5	μm	120	150	10
L_1	nm	1	125	25
L_5	nm	30	100	10
L_6	nm	100	150	10

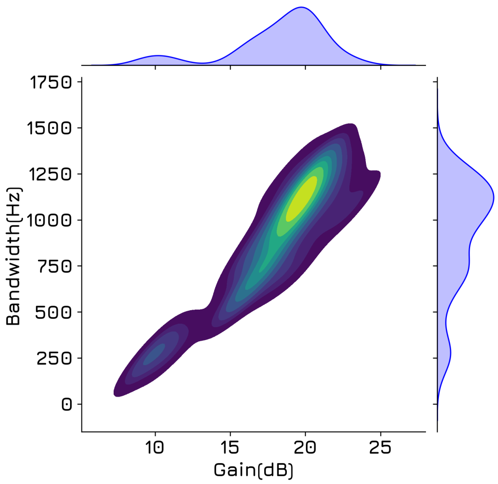

Fig. 5: Density Distribution of Circuit Performance Metrics: Gain (dB) versus Bandwidth (Hz)

A. Data Acquisition

We leverage a custom dataset of PA design parameters and their associated performance metrics to build ML models. Initially, we systematically generated the input features of the dataset by exploring design space. Tab-I shows the operating range and step size of each input feature. This exhaustive approach produced 432000 unique combinations of features. To create a computationally viable dataset for model training and evaluation, 300 samples were randomly selected. This method guarantees that the final dataset is an unbiased and representative subset of the entire possible design space as mentioned in Tab-I. Finally, the corresponding gain and bandwidth for each feature set were calculated from the Cadence Virtuoso simulation. In Fig-5, the joint distributions of gain and bandwidth reveal complex multi-modal distributions for both targets. This characteristic motivates this study to select the advanced ML models that can effectively learn complex relationships. Fig-3 elucidates the entire life cycle of this study. As shown in the flowchart, before feeding the dataset to ML models, values are normalized to the [0,1] range to eliminate disproportionate influence of features with large values using *MinMax* scaler implemented in *scikit-learn*. The dataset was partitioned into a training set (80%) and a test set (20%). To ensure the reproducibility of the experimental results, a fixed random state (seed = 42) was used for all training procedures.

B. Development of Random Forest model

This study employs the Random Forest (RF) regression technique to predict the outcomes by utilizing the well-established *scikit-learn* python library. We specifically chose this model due to its exceptional ability to capture complex, non-linear behavior in data while minimizing the risk of overfitting. Typically, this method builds multiple decision trees trained on random subsets of the entire dataset and it averages their predictions from all trees [11]. This process of creating diverse trees is key to improving prediction accuracy and generalization. This ensures the predictive model remains both powerful and well-suited for our circuit design scenarios. For this study, the RF model was configured to build a forest of 100 individual decision trees. Prediction performance was assessed using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2) ensuring a reliable and interpretable framework for modeling multiple circuit design outputs.

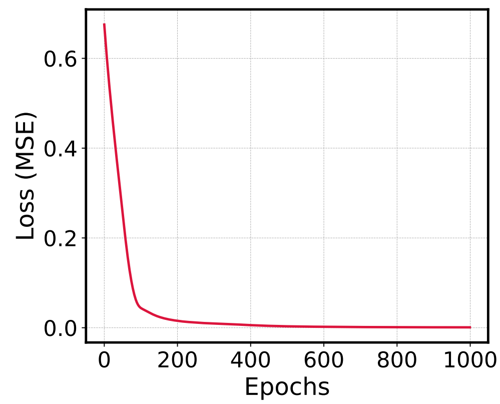

Fig. 6: ANN Training Loss

C. Development of XGBoost model

To assess the effectiveness of ensemble-based method, we employed XGBoost (Extreme Gradient Boosting) in addition to Random Forest. In this method, a series of decision trees are built where each tree was trained to correct the errors of the prior one [12]. Even though both Random Forest and XGBoost are ensemble-based methods, the key difference is that RF builds independent parallel trees, whereas XGBoost builds them sequentially. This progressive improvement makes XGBoost an extremely powerful method to capture complex nonlinear behavior, as required in this study.

An XGBoost regression model is employed due to its robustness in capturing complex nonlinear relationships in the current dataset. Multi-target gradient boosting regression is implemented using the *xgboost* and *scikit-learn* libraries. Original implementation of regressor from *xgboost* library doesn't support multi-target natively. *MultiOutputRegressor* wrapper from *scikit-learn* was used to train simultaneous targets. The regressor is configured with 100 estimators and was set to minimize the squared error which commonly used in regression. The model's performance was evaluated by comparing the predicted and actual test values using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2). This method demonstrates the effectiveness in precisely modeling multi-output circuit parameters, underscoring XGBoost's suitability for efficient and accurate circuit design prediction.

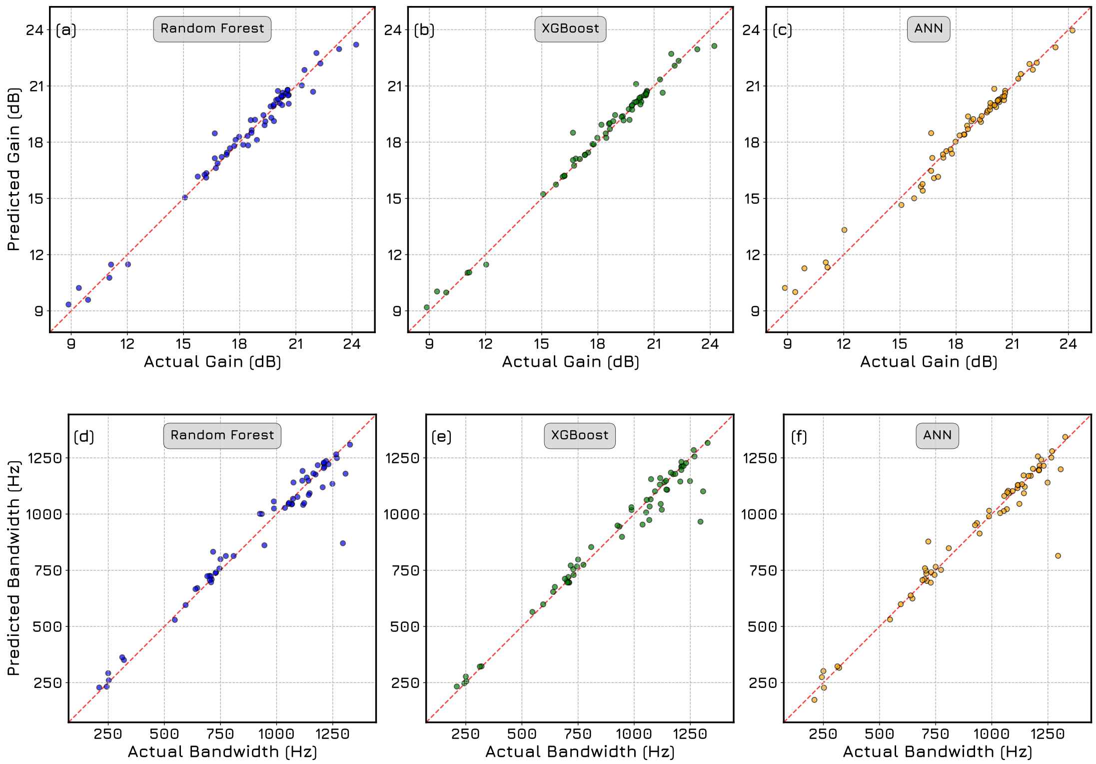

Fig. 7: Gain (dB) and Bandwidth (Hz) Prediction Comparison: Actual vs. ML Predicted Values.

D. Development of ANN model

To broaden the analysis beyond the single class of ensemble models we used, it was crucial to introduce a fundamentally different ML approach: the Artificial Neural Network (ANN). Relying solely on one model type, even a powerful one, risks overlooking important features within the data. Its unique architecture of interconnected layers allows it to learn deep, hierarchical features via activation function like the Rectified Linear Unit (ReLU). This change ensures a more robust and comprehensive assessment of the modeling task, offering distinct mechanism for pattern identification unlike the divide-and-conquer approach used by ensemble methods.

In this study, the ANN architecture used 6 circuit design variables as input, processes them through two hidden layers with 32 and 16 neurons, and outputs two parameters: gain and bandwidth, as depicted in Fig-4. ReLU activation function was applied in the hidden layers to introduce non-linearity and data was preprocessed for better convergence. The model was implemented in PyTorch, trained for 1000 epochs using MSE loss and the Adam optimizer. The training loss is monitored to ensure stability with MSE at a final loss of 0.000887 and consistent learning, as shown in Fig-6. This steady decline without big fluctuation validates that the chosen architecture and learning rate are well-suited for the VLSI regression task and indicates near-optimal convergence. The model's predictive performance, assessed on the test set using metrics like RMSE, MAE, and R^2, demonstrates its capability as a reliable and efficient tool, allowing for a fair comparison against the ensemble methods.

TABLE II: Comparison of Machine Learning Model Performance

Model	**Gain Target**			**Bandwidth Target**		
	RMSE	**MAE**	$\mathbf{R^2}$	**RMSE**	**MAE**	$\mathbf{R^2}$
Rand. Forest	0.46	0.34	0.979	70.97	39.53	0.946
XGBoost	0.41	0.25	0.984	62.08	34.28	0.959
ANN	0.50	0.32	0.976	73.32	35.16	0.943

III. Result and Discussion

A. Exploring the design space of gains and bandwidth

In Fig-5, we have shown the distribution of gains and bandwidths generated by Cadence Virtuoso simulation. Here, central contour plot describes the combined distribution of both targets, and the marginal density plots above and to the right depict the distributions of Gain and Bandwidth separately. The brighter regions in the contour plot indicate where most measurements occur. The contour plot mostly extends toward the diagonal, indicating a positive correlation between gains and bandwidths. The probability distribution of gains clearly exhibits a bimodal distribution, where the primary mode is centered at 20 dB and a relatively smaller secondary mode is located at near 10 dB. Interestingly, the bandwidth's density plot shows more than two modes or a multi-modal distribution. The dominant mode is located near 1100 Hz, where weaker modes are located at approximately 750 Hz and 250 Hz, respectively. This multi-modal behavior strongly indicates that the input design parameters drive the VLSI circuit into separate stable operating regimes with discontinuous performance characteristics. Consequently, predicting bandwidth is more challenging for the regression models, as they must capture the abrupt, non-linear transitions between distinct operational modes.

B. Comparative Accuracy of ML Models

We have presented performance metrics (RMSE, MAE, and R^2) of all three ML models in Tab-II. These numbers indicate that all ML models are extremely capable of establishing the relationship between key circuit parameters and the targeted outputs, namely gains and bandwidth, yet there are clear distinctions in their predictive accuracies.

XGBoost has demonstrated overall the best performance in predicting both gain and bandwidth. For gain prediction, XGBoost achieved an impressive coefficient of determination (R^2) value of 0.984, while minimizing errors, as evidenced by the lowest RMSE (0.41 dB) and MAE (0.25 dB). Similarly, it has also scored the best prediction accuracy for bandwidth, registering an R^2 value of 0.959, an RMSE value of 62.08 Hz, and an MAE value of 34.28 Hz. In the performance hierarchy, the RF model secured a competitive second-best position, performing only slightly worse than XGBoost. Conversely, ANN was the least accurate model out of all, with R^2 value of 0.976 and 0.943 for gain and bandwidth, respectively.

A key observation is that all models exhibited a drop in performance when predicting bandwidth compared to gain. For instance, XGBoost's R^2 value decreased from 0.984 for Gain to 0.959 for Bandwidth. This can be primarily attributed to, as discussed earlier, the fact that bandwidth exhibited complex multi-modal distribution compared to gain, making it more difficult for regression models to capture the abrupt, non-linear transitions in the design space.

In addition to quantitative values, the actual vs. predicted plots (Fig-7) also visually corroborated the performance of the ML models. Specifically, tightly clustered data points around ideal $y = x$ line indicate the high fidelity of prediction.

In sum, the tree-based ensembles (XGBoost and Random Forest) have performed better compared to the neural network based approach. This can be attributed to their inherent ability to learn complex relationship from structured, smaller dataset. Typically, ANNs are prone to overfitting, leading to poor performance for a dataset consisting of 300 samples. XGBoost's sequentially learning and self-correction is proven to be extremely effective at capturing the underlying physics of the VLSI circuit.

C. Explainable AI

The Permutation Feature Importance (PFI) analysis is significant for design engineers. It offers a simple approach to measure the impact of each feature on prediction by shuffling feature values and observing the change in performance. It measures the effect of each design parameter (like V_b, M_4, L_1) on gain and bandwidth. By quantifying the influence of each input parameter, the PFI plots for gain prediction directly identify the dominant performance factors: the V_b (Bias Voltage) is the most influential parameter, followed by inductor value L_5 (nm), and then the transistor width M_4 (µm). Conversely, the effects of M_5, L_1 and L_6 are almost negligible, as shown in Fig-8. For bandwidth prediction, a similar trend is observed, with V_b, M_4 and M_5 are identified as the dominant parameters influencing model output, while the effects of inductors (L_1, L_5, L_6) remain minimal. The PFI analysis identifies which circuit components are the dominant performance bottlenecks. This insight is vital for optimizing future circuit designs.

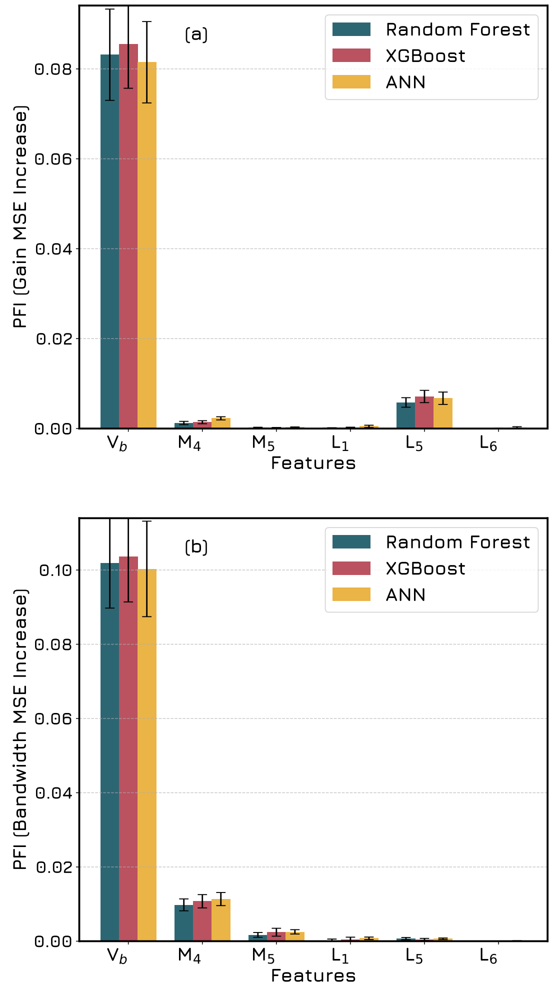

Fig. 8: Permutation Feature Importance (PFI) of Gain and Bandwidth.

Conclusion

This study represented the comparative performance of three ML models, Random Forest, XGBoost, and Artificial Neural Network, in predicting the gain and bandwidth of wide bandgap power amplifier. We found that the tree-based ensemble regression methods consistently outperformed the ANN. Specially, XGBoost outshone the others by delivering the best R^2 values and the least error metrics. Furthermore, explainable AI technique, Permutation Feature Importance (PFI), was employed to identify the key design parameters affecting the power amplifier's gain and bandwidth. It revealed that, biasing voltage (V_b) is the most dominant design parameter, while the inductor value (L_6) appears to be the least dominant factor and relatively insensitive to minor tolerances.

References

[1] B. Subburaman, V. Thangaraj, V. Balu, U. M. Pandyan, and J. Kulkarni, "Artificial neural network modeling of a cmos differential low-noise amplifier using the bayesian regularization algorithm," *Sensors*, vol. 23, no. 21, p. 8790, 2023.

[2] S. Sorkhabi, M. Mosavi, and M. Rafei, "Low noise amplifier synthesis using multidimensional mlp neural network," *IETE Journal of Research*, vol. 64, no. 3, pp. 374–386, 2018.

[3] S. Singh and P. Chopra, "Artificial neural network approach for lna design of gps receiver," *Optical Memory and Neural Networks*, vol. 25, no. 4, pp. 236–242, 2016.

[4] F. Güneş, S. Demirel, and S. Nesil, "Design optimization of lnas and reflectarray antennas using the full-wave simulation-based artificial intelligence models with the novel metaheuristic algorithms," in *Simulation-Driven Modeling and Optimization: ASDOM, Reykjavik, August 2014*. Springer, 2016, pp. 261–298.

[5] S. Kumar and S. Kumari, "Design of low power, high gain lna for wcdma range and parameters extraction using artificial neural network (ann)," in *2015 IEEE Power, Communication and Information Technology Conference (PCITC)*. IEEE, 2015, pp. 436–441.

[6] E. Dumesnil, F. Nabki, and M. Boukadoum, "Rf-lna circuit synthesis using an array of artificial neural networks with constrained inputs," in *2015 IEEE international symposium on circuits and systems (ISCAS)*. IEEE, 2015, pp. 573–576.

[7] G. Karimi, S. B. Sedaghat, and R. Banitalebi, "Designing and modeling of ultra low voltage and ultra low power lna using ann and anfis for bluetooth applications," *Neurocomputing*, vol. 120, pp. 504–508, 2013.

[8] A. Khan, A. Ali, J. Khan, F. Ullah, and M. Faheem, "Using permutation-based feature importance for improved machine learning model performance at reduced costs," *IEEE Access*, 2025.

[9] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[10] N. A. Salsabila, "Design of a two-stage broadband stable power amplifier for low power wireless telemetry systems using 90 nm cmos technology," in *2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN)*. IEEE, 2025, pp. 1–6.

[11] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[12] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.