Performance Comparison of Ensemble Algorithms with Neural Networks based Methods for Small-Signal Modeling of GaN HEMTs

Bagylan Kadirbay, BTech in Electrical and Computer Engineering

Submitted in fulfilment of the requirements
for the degree of Master of Science
in Electrical and Computer Engineering

NAZARBAYEV
UNIVERSITY

School of Engineering and Digital Sciences Department of Electrical and Computer Engineering Nazarbayev University

53 Kabanbay Batyr Avenue,
Nur-Sultan, Kazakhstan, 010000

Supervisors: Dr. Mohammad Hashmi and Dr. Prashant Jamwal

March 2024

Declaration

I hereby, declare that this manuscript, entitled *”Performance Comparison of Ensemble Algorithms with Neural Networks based Methods for Small-Signal Modeling of GaN HEMTs”*, is the results of my own work except for quotations and citations which have been duly acknowledged.

I also declare that, to the best of my knowledge and belief, it has not been previously or concurrently submitted, in whole or in part, for any other degree or diploma at Nazarbayev University or any other national or international institution.

Signature: ____________

Bagylan Kadirbay
April 2024

Abstract

This thesis investigates and compares the ensemble modeling methods and neural networks approaches for small-signal modeling of Gallium Nitride High Electron Mobility Transistors (GaN HEMTs). Specifically, ensemble methods are represented by Random Forests and eXtreme Gradient Boosting (XGBoost) algorithms, while neural networks techniques consist of Adaptive Neuro-Fuzzy Inference Systems (ANFIS) and Feed-forward Artificial Neural Networks (FFANN). To carry out the research to a higher standard, this work utilizes two distinct GaN HEMT devices. The first, a GaN HEMT grown on Diamond, is characterized by a smaller dataset and fewer modeling parameters. Conversely, the second device, a GaN HEMT on Silicon, possesses a larger dataset and a greater number of training parameters.

Furthermore, the model performance is meticulously evaluated using Mean Square Error (MSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). Findings suggest that ensemble models exhibit enhanced stability and greater robustness against overfitting. While the neural networks-based models demonstrate superior accuracy and a more streamlined development process. This research provides critical guidance for researchers and engineers in selecting the most suitable modeling approach for certain GaN HEMT devices. The choice hinges on a careful balance between prioritizing accuracy, mitigating overfitting, and managing the complexities inherent in model development.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, **Dr. Mohammad Hashmi**, and **Ph.D. candidate Saddam Husain**, for their invaluable guidance, insightful feedback, and unwavering support throughout my thesis journey.

My sincere thanks also extend to my co-supervisor, **Dr. Prashant Jamwal** and committee member, **Dr. Mehdi Bagheri**, for their constructive suggestions and encouragement.

I am grateful to have worked alongside the exceptional team under the supervision of **Dr. Mohammad Hashmi**. Thank you for creating a collaborative and friendly environment.

Also, I will express my deep gratitude to my family and friends, especially, my wife, thank you for your endless patience, understanding, and belief in me. I could not have done this without you.

Table of Contents

List of Abbreviations & Symbols

1. 2DEG	Two-Dimensional Electron Gas
2. ANFIS	Adaptive Neuro-Fuzzy Inference Systems
3. ANN	Artificial Neural Networks
4. DT	Decision Tree
5. ECMs	Equivalent Circuit Models
6. F	Frequency
7. FFANN	Feed-Forward Artificial Neural Networks
8. GaN	Gallium Nitrite
9. HEMTs	High Electron Mobility Transistors
10. IoT	Internet of Things
11. LSM	Large Signal Modeling
12. MPA	Marine Predators Algorithm
13. ML	Machine Learning
14. MAE	Mean Absolute Error
15. MSE	Mean Squared Error
16. OAs	Optimization Algorithms
17. POA	Pelican Optimization Algorithm
18. R^2	Correlation Coefficient
19. ReLu	Rectified Linear Unit
20. RF	Radio Frequency
21. RFs	Random Forests
22. SSM	Small Signal Modeling
23. SVR	Support Vector Regression
24. T	Temperature
25. TSA	Tunicate Swarm Algorithm

List of Tables

List of Figures

1 Chapter: Introduction

1.1 Background

Radio Frequency (RF) technology plays a pivotal role in modern communication technologies. Its widespread use, coupled with the ever-growing demand for increased and faster connectivity, drives the demand for continuous innovation and advancement within the field of RF technology [1, 2, 3]. For example, RF technology is fundamental to a vast array of modern communication systems, including cellular networks, satellite communications, radar, and wireless data transmission. Its ability to enable seamless connectivity also lays the foundation for cutting-edge innovations like the Internet of Things (IoT) and the rapid deployment of 5G networks. [2, 4, 5]. In the case of 5G, the demand for higher throughput is met by only building the antenna stations bigger and concentrated for a small area to facilitate unprecedented data speeds, low latency, and the capacity to support a massive number of connected devices simultaneously [6, 7]. This will lead to the higher installation cost of the 5G technology and is also not friendly to the environment [3, 7]. Therefore, substantial improvement is needed to eliminate these drawbacks.

Gallium Nitride High Electron Mobility Transistors (GaN HEMTs) have garnered popularity in RF applications owing to their distinctive attributes, including a wide band gap (3.4 eV), high electron mobility (1500 $cm^2\ V^{-1}S^{-1}$), robust break-down voltage, and elevated operating frequency [8, 9]. The wide band gap and high electron mobility nature of the material enable the current Silicon (S) based antenna to operate at a next-level operating speed, and the high bread-down voltage allows the device to provide more stable performance under different demands and conditions [8, 9].

To implement the material into the industry, we have to have a complete model of the device for extensive voltage inputs. At present, there are two major modeling phases in the current development process of the device. The first is the Small Signal Modeling (SSM), and the second one is the High Signal Modeling (LSM) of GaN HEMTs []. The imperative for conducting both small and large signal modeling of Gallium Nitride High Electron Mobility Transistors (GaN HEMTs) lies in the comprehensive understanding of the device's operational characteristics. These modeling approaches serve distinct yet interrelated purposes and collectively contribute to the optimization and advancement of GaN HEMT-based RF technologies.

1.2 Small-Signal Modeling of GaN HEMTs

This work focuses on the initial phase of analysis, where the SSM plays a crucial role. The SSM enables us to investigate how the device behaves under small, incremental changes. This allows us to analyze the device's linear response and assess its stability. Particularly, by characterizing the small signal parameters, such as transconductance and capacitances, researchers gain insights into the linear response of GaN HEMTs [9, 10]. This information is paramount for designing and fine-tuning RF circuits, providing a foundational understanding of how the device responds to small variations in input signals. SSM thus serves as a fundamental roadmap for the subsequent stages of design and optimization [9, 10].

The Equivalent Circuit Models (ECMs) are the common step in SSM of GaN HEMT because the ECMs provide a tangible understanding of the device characteristics and allow the researcher to design the device accordingly [9, 11]. There are generally ”Cold” and hybrid parameter extraction methods available in the current research [9, 12]. However, the effectiveness of each approach is highly dependent on the initial measurement, correctness of the assumption, and extraction procedure. Small errors in any single part of the process will lead to cumulative errors that have severe impacts on the final result, and even malfunctioning of the device. Therefore, a deep understanding of the physics of the material, sophisticated statistical analysis, and the formulation of empirical relationships that encapsulate the device's response to small signals are needed during the extraction of ECMs of the GaN HEMT using these methods.

Machine learning (ML) approaches become popular in the SSMs of GaN HEMT as they entail the utilization of algorithms to autonomously learn patterns from data, enabling the prediction of device responses to small signals without explicit mathematical modeling [11, 13]. This technique can adapt to intricate non-linearities and dependencies within the data, potentially outperforming traditional analytical or empirical methods in capturing the intricate behaviors of GaN HEMTs under small signal conditions.

Among various ML methods, there are different types of learning algorithms that promise to exhibit an aptitude for discerning intricate patterns within complex datasets. In this context, we will compare two distinct algorithms in the field. The first one is the ensemble method, XGBoost, and Random Forests (RFs) chosen as the representative for the method. The second type of approach is based on the neural networks, and to show the capability of the algorithm two techniques developed on different platforms are selected, one is Adaptive Neuro-Fuzzy Inference Systems (ANFIS) and the second one is Feed-Forward Neural Networks (FFANN).

Each algorithm has its advantages and disadvantages. Therefore, depending on the data specification researchers have ended up with various results which give suggestions about the utilization of the various algorithms [11]. For example, XGBoost and RFs, in the SSM of GaN HEMT provide a sophisticated framework for aggregating the predictive prowess of the large amount weak learners. XGBoost is characterized by its gradient-boosting framework while the RFs is a robust ensemble technique based on decision trees. On the other hand, the neural networks based ANFIS and FFANN techniques are popular for their high accuracy and high performance.

1.3 Aims & Objectives

As we showed above the SSM of GaN HEMT is critical for the accurate characterization of their high-frequency performance, linear response, and stability, therefore, plays a pivotal role in modern high-frequency electronic devices and communication systems. One promising approach is the use of ensemble methods that combine the strengths of aggregating weak learners' techniques. The second neural networks techniques also give high promise in modeling the device. This work investigates and develops models for RFs, XGBoost, ANFIS, and FFANN on different GaN HEMT devices. Lastly, compare the effectiveness of each algorithm and suggest the possible improvement from each model. Mean absolute error (MAE), mean squared error (MSE), and the coefficient of determination (R^2) are used to give a thorough analysis of the accuracy of each model.

The most optimal modeling method should:

- Provide accurate SSM for GaN HEMTs across a range of biasing conditions and frequencies.

- Address the limitations of existing modeling techniques, such as over-fitting and low accuracy in high-frequency areas.

- Create efficient and reliable models for extracting the device parameters.

1.4 State of the Art

[14] states the importance of SSM for GaN HEMTS before entering the LSM and further designing the power amplifier and offers an effective method for extracting intrinsic parameter values of the GaN on Silicon-Carbon (SiC) substrate HEMT. This study aims to develop an accurate parameter extraction algorithm for the SSM of GaN HEMT. This implies the importance of accuracy in modeling the device, as the GaN HEMT is prone to small errors, and the error cumulative nature of the traditional de-embedded method strengthens the need, in order to prevent unexpected behavior. This research implements MatLab to program the algorithm, which improves the effectiveness of the overall process. The algorithm consists of two major branches. The first is a parasitic parameter extraction procedure that aims to iteratively enhance the accuracy of each extrinsic parameter. Along with every iteration, there is parameter dynamic scanning takes place to fasten the iteration to converge. This paper also pointed out the use of machine-learning techniques in the field and the impact of ambient temperature involved in the extraction process.

The next research uses Artificial Neural Networks (ANN) in the parameter extraction process along with consideration of ambient temperature (20 °C to 80 °C) and resulted in a more accurate model [15]. The author builds different types of ANN models for imaginary and real Scattering parameters (S-parameters) to meet the difference. The comparison between simulated and measured data achieved a high correspondence. Additionally, the test is also implied to the range of temperature that was not included in the training cycle. The final result continuously performs well in those kinds of out-of-range cases.

When considering the impact of the temperature on the operation accuracy of the device, the diamond possesses the highest thermal conductivity of any known material (around 2000

W/mK). This is substantially higher than silicon's thermal conductivity (roughly 150 W/mK). Furthermore, GaN HEMTs generate significant heat during operation, especially in high-power applications. Diamond substrates allow for much more efficient heat dissipation away from the active device area. This will reduce the possibility of the occurrence of the self-heating issue compared to the silicon-based material. [16] introduced a hybrid modeling approach for the SSM of GaN-on-diamond HEMT, which evaluates the scanning base systematic parametric extraction method with the combination of direct extraction methods with Optimization Algorithms (OAs) such as Marine Predators Algorithm (MPA), Pelican Optimization Algorithm (POA) and Tunicate Swarm Algorithm (TSA). The result shows that the scanning method performs well in the whole frequency range with less execution time compared to the hybrid method, but a good initial assumption is the key to success. While the hybrid technique is also highly accurate and reliable, however, the model-building time is the main drawback. This will pose the question of the efficiency of the ensemble method in this work but indirectly asserts that the ensemble model may have more capability in modeling the device than the single approach.

The base of the ensemble methods is the Decision Tree (DT), which is faster than the average modeling technique. [17] have developed DT based SSM of GaN HEMT and compared the accuracy and efficiency of the method with Support Vector Regression (SVR) and ANN. The main purpose of this paper was to implement the novel tree structure-based algorithm for modeling the device. The root node is split into two groups according to the splitting points, and the new nodes will experience the same division until reaching the leaf node or the final response. Correspondence between simulated and measured scattering (S) parameter values serves as the main indicator for conducting the research, and they vary according to the input voltages and the frequency. From the research process, it can be concluded that this learning algorithm is relatively simple compared to SVR and ANN, and unlike SVR little or even no data preprocessing is needed. This paper clearly shows the generalization capabilities of the DT algorithm and with several mathematical computations explains the working principle of the algorithm. Additionally, the accuracy of the DT also outperformed the other learning algorithms. Regarding the efficiency of the learning model, the researchers only compared modeling complexity within each scattering parameter, not with SVR and ANN as a whole. Therefore, it is not sufficient to claim the efficiency of the model compared to others.

The second research assessed the performance of the DT modeling technique for GaN HEMT over a larger temperature range [18]. As the temperature rises the electron mobility capacity will degrade nonlinearly due to the impurity scattering. The researchers show that some modeling processes underestimate the impact of the temperature and then lead to large errors as the temperature departs from the ideal condition. The DT algorithm has high generalization ability, and can conduct multidimensional data analysis; therefore, the project decided to use this technique for modeling the device. The involvement of the temperature parameter for modeling the GaN HEMT is important and practical because it is common for the device to work under different environmental conditions. However, in this paper, the result from the ML technique overlapped with the result from the simulation approach. Generally, the approach taken by the researchers is correct, but the robustness of the model should be improved. The overfitting issue that occurred in the result.

[13] implemented a novel ML technique for modeling the Gallium Arsenide High Electron Mobility Transistor (GaAs pHEMT) with a multi-finger layout. The main purpose was to model the behavior of GaAs pHEMT more efficiently compared to the conventional parameter extraction method. The researchers used the XGBoost regression with the K-folds cross-validation

machine learning method for developing the RF characteristic of the device. This hybrid technique is well-known for its efficiency and accuracy among other modeling competitions, and in this experiment, it has the highest score of 99.88%. This result was achieved in the condition with several inevitable deviations such as uncertainty in measuring the microwave and the device degradation. Another important point in the research project was the use of the number fingers N_f and their width W_0 for finding the hidden layer of the autoencoder.

At the base, there are two types of ensemble methods according to the model's development. The first one iteratively improves the result obtained from the previous algorithm, and the second one combines the result from all the algorithms at the beginning and then optimizes it [19]. [19] uses RFs as the base for modeling the GaN HEMT device to show the comprehensiveness and versatility of the algorithm. This research project implements the Bayesian optimization technique for further improvement. In the end, the result proves the initial prediction with low error and high stability over the entire frequency range. The next comprehensive research related to the ensemble methods implies that RFs has lesser efficiency as it requires a bigger memory size to store and this led to the longer training and tuning time expenses [9].

In the next model fusion state, the weight-voting and meta-learning approaches are generally in use. The proposed hybrid ensemble method uses all the approaches simultaneously [20]. This approach outperformed the ANN and other learning techniques in terms of robustness because the risk is divided among different types of algorithms, the failure of one algorithm can subsequently be filled by the second algorithm. The optimal result will be obtained from the resulting pool [20]. On the other hand, this approach reduces the over-fitting rate [9]. [21] used the ensemble method in default prediction and obtained significant improvement over other techniques. The next research used an ensemble stacking model in classifying the gender according to the voice and scored 99.64% accuracy [22]. In terms of the generalization capability, this algorithm also showed high performance over the other machine learning methods such as Recurrent Neural Network (RNN), Support Vector Machine (SVM), XGBoost, and LightGBM [23, 24, 25]. There is less research done on the comparison of ensemble methods with neural networks based algorithm in modeling the RF signal, especially, the SSM of GaN HEMT. This research will try to fill the gap by extensively studying the performance of both techniques in modeling the GaN HEMTS.

2 Chapter: Device Physics, Characterization and Data Processing

2.1 Device Physics Characterstics

2.1.1 GaN-on-Di

The first GaN HEMT device uses a 500 μm diamond substrate because the diamond's exceptional thermal conductivity allows for improved heat management compared to traditional Silicon based substrate devices. The device is initially constructed on a SiC substrate, then transferred and bonded to the diamond substrate [26, 27, 28]. More specifically, the epitaxial layers include a 1 nm AlN nucleation layer, 2 μm Fe-doped GaN buffer, 20 nm AlGaN barrier layer (0.3 nm Al-mole), and a 2 nm GaN cap. Precision fabrication techniques (E-beam lithography) are used to create the gate with the following specifications:

- space between gate and drain: L_{gd} = 2 μm
- space between gate and source: L_{gs} = 1 μm
- gate-finger width: W_g = 125 μm
- gate length: L = 0.25 μm
- gate number: N_g = 4

N5245 Vector Network Analyzer was used to characterize the behavior of the device in complex S-parameters consisting of real and imaginary parts. S-parameters are used to record the behavior of both devices because they represent the ratio of the outgoing (scattered) wave to the incoming wave at each port of the device [29, 30]. In other words, they show how effectively the GaN HEMT device is transmitting the signal. The combination of S-parameters taken for the behavior of both devices is listed below:

- Input Reflection Coefficient (S_{11}): It shows how much of the input signal is reflected to the input port.
- Reverse Transmission Coefficient (S_{12}): This is the amount of the input signal that leaks through the device and appears at the output port.
- Forward Transmission Coefficient (S_{21}): It shows the ratio of the input signal transmitted to the output port.
- Output Reflection Coefficient (S_{22}): Just inverse to the S_{11}, this parameter represents how much of the output signal is reflected towards the output port.

The output dataset for GaN on Diamond HEMT consists of 36400 data samples. Specifically, the voltage between gate-to-source (V_{GS}) is in a range of -3 V to 0 V, drain-to-source V_{DS} starts from 0 V to 30 V, and lastly, the frequency of the device up to 40 GHz.

2.1.2 GaN-on-Si

The device has 10 fingers with 200 μm for each and is grown on the Silicon substrate as shown in Figure 1 [12, 31]. The fabrication of this device includes several advanced technologies such as an improved passivization procedure that was introduced to enhance RF responses and reduce the surface trapping effect, the addition of the nucleation layer aims to prevent low buffer trapping due to lattice incompatibility. The dataset assembly incorporates the short-open-load-thru (SOLT) technique to cope with possible systematic errors. N5245A VNA is used to characterize the behavior of the device in terms of real and imaginary S-parameters. Compared to the previous device, the dataset gathered from the GaN-on-Si is more complex as it has 103057 data samples and it has an additional thermal parameter from $25\,^{\circ}\mathrm{C}$ to $175\,^{\circ}\mathrm{C}$ with a step size of 25. The predictors consist of V_{GS} -2 to 2 V, V_{DS} 7 to 48 V, and the frequency up to 26Gz.

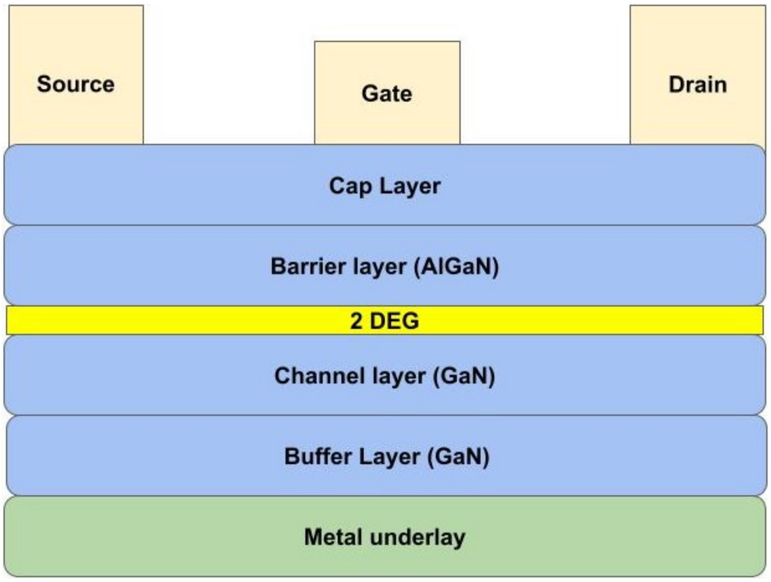


Figure 1: Simple GaN HEMT structure.

2.2 Problem Statement

The accurate SSM of GaN HEMTs is essential for optimizing their high-frequency behavior in modern power electronics and communication systems [9, 32]. Neural networks have proven effective in this modeling domain, and ensemble methods have also shown high performance. To fully exploit the capabilities of these data-driven approaches, a comprehensive comparison is needed to evaluate the efficiency and performance of ensemble methods against neural networks based techniques.

The ideal modeling method should:

- First and foremost is to achieve high accuracy. In other words, precisely predict GaN

HEMT behavior across various operational conditions.

- Secondly, address the limitations of existing modeling techniques, such as over-fitting and low accuracy in high-frequency areas.
- Lastly, the model should be computationally efficient. Because in real-time design optimization, the time for the modeling and simulation is less than the research development period.

2.3 Methodology

Python programming language is a well-known tool in the data science and ML field. Therefore, it is used for data cleaning and preprocessing before the individual simulation. Additionally, RFs, XGBoost, and FFANN models are trained with the help of the Python ML libraries. While the ANFIS models are built under the MatLab development environment. Specifically, the Fuzzy Logic Designer tool set has a well-developed user interface that is shown in Fig. 2 which eliminates the need to write codes as the previous methods do.

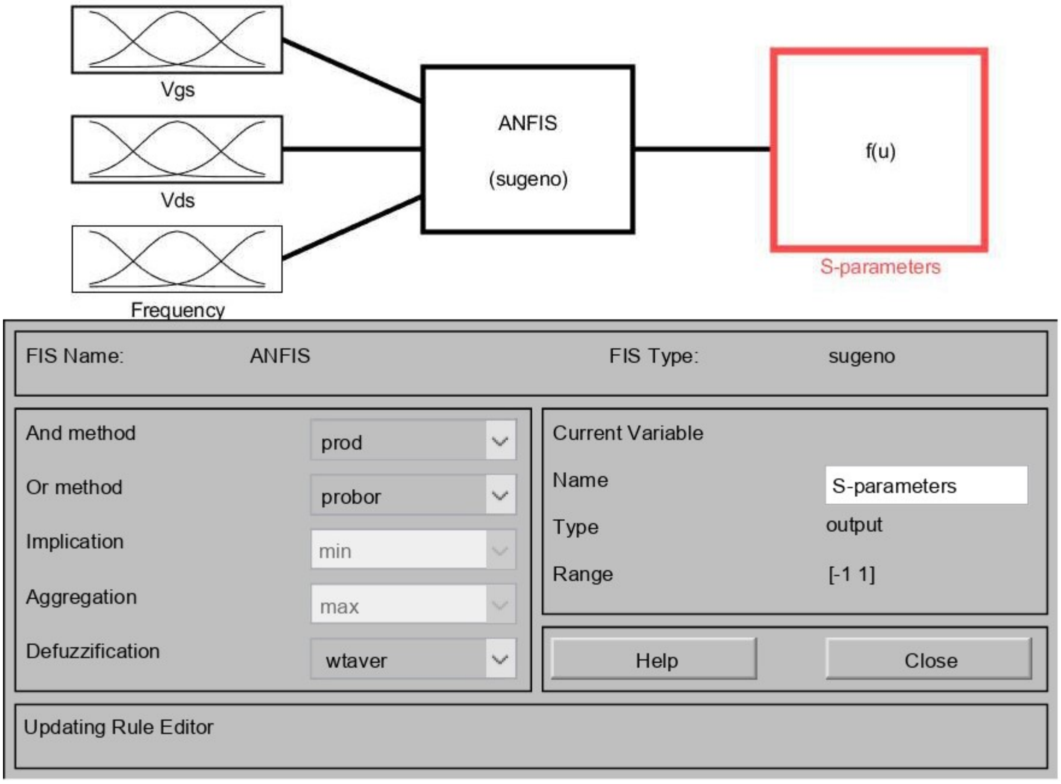
In the end, every model is evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), and proportion of the variance ($\%R^2$) as shown in the below formulas.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

$$\%R^2 = (1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}) * 100 \tag{3}$$

Until then we need to prepare the dataset and train the model to achieve the most optimal



FIS Name: ANFIS

FIS Type: sugeno

And method: prod

Or method: probor

Implication: min

Aggregation: max

Defuzzification: wtaver

Current Variable

Name: S-parameters

Type: output

Range: [-1 1]

Updating Rule Editor

Figure 2: ANFIS user interface in MatLab for the GaN on Diamond HEMT.

results.

2.4 Data Prepossessing

Prior to the model selection and development the most important stage is the data preparation part. Because the outliers or null data among the dataset heavily impact the overall learning of the model and cause lower performance or misleading generalization. Another purpose of the data preprocessing section is to feature scale the value of parameters to a common range. By bringing features to a similar scale, feature scaling aims to prevent any single feature from dominating the learning process solely due to its magnitude, thus enhancing model stability and convergence. In our case, the magnitude of the frequency is multiple times bigger than other parameters, and this will mislead the model in the training section. After the dataset division into training and testing sets both device datasets are scaled to a range between -1 and 1 with the help of Min-Max scaling in Python. The individual dataset splitting into training and testing sets is shown in the following subsections.

2.4.1 GaN on Diamond HEMT

The quality of the dataset is relatively high as it does not have missing values and the number of outliers is also low enough that do not impact the later training process. The whole dataset consists of 36400 data samples, and it is divided based on the value of V_{DS}. Sections where V_{DS} equals 5 V, 15 V, and 25 V with 8400 samples are allocated for the testing part while the remaining 28000 data samples are used for building the model. The input-output relationship is shown in Fig. 3.

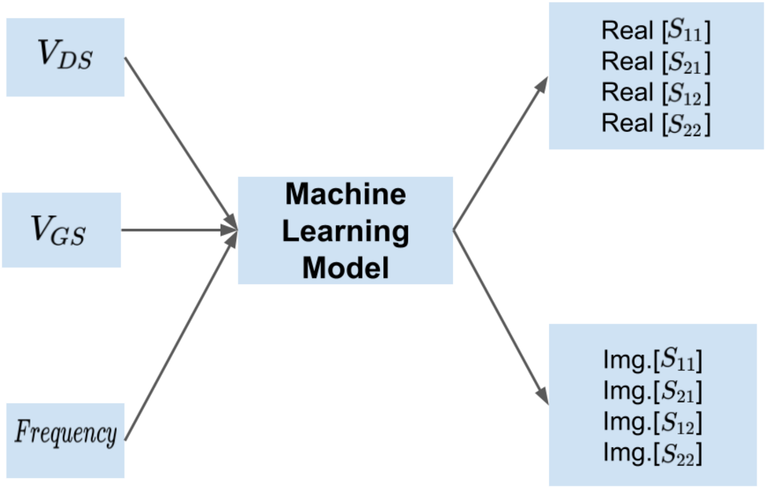


Figure 3: Input-output relationship for GaN on Di HEMT.

2.4.2 GaN on Si HEMT

GaN HEMT grown on silicon (Si) substrate has different structures and sizes in terms of the dataset. The dataset consists of 103057 samples, with voltages applied across gate and source (V_{GS}), drain and source (V_{DS}), frequency starts from 0.1 GHz to 26 GHz, and temperature between $25\,^{\circ}\mathrm{C}$ to $175\,^{\circ}\mathrm{C}$ as input parameters while real and imaginary S-parameters serve as the output results [33]. The data division is made according to the temperature. 100 °C and 175 °C are selected for the testing dataset, and one thing to note is that the second temperature parameter serves as the unknown samples for the training range. The remaining temperature samples are allocated for the training dataset. As a result, the training dataset has 78997 samples while the testing one takes 24060 examples. The relationship between the data processing and model building process is shown in Fig 4.

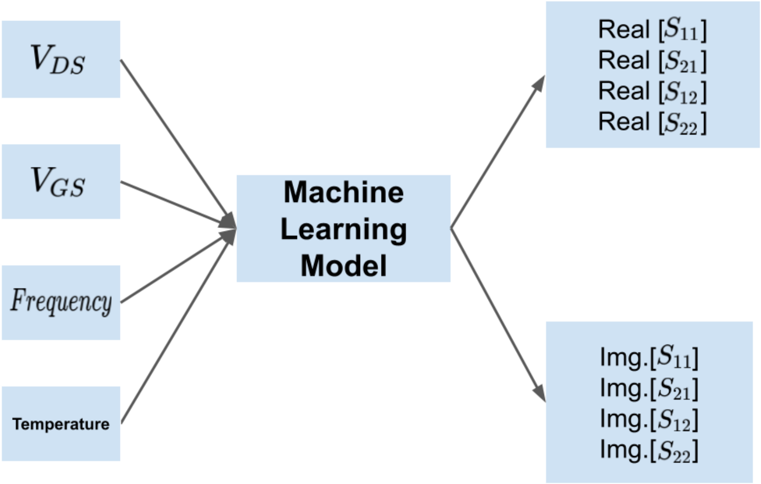


Figure 4: Input-output relationship for GaN on Si HEMT.

2.5 Hyperparameter tuning

A formatted dataset with the model library itself will not give the final optimal result. In real practice, finding the optimal set of hyperparameters is essential to unlocking the full potential of a model. specifically, the right combination of hyperparameters enables the model to:

- Avoid overfitting and underfitting. The right hyperparameters help strike the critical balance between a model that excels at the training data but not on the testing one or not performing well for both scenarios.

- Increase the accuracy, precision, recall, and overall performance of the model.

- Enhance the efficiency.

- Stabilizes the model by having higher generalization capability. In other words, leads models to perform well on unseen data, not just the training set.

The Hyperparameter tuning techniques used for the thesis include *Manual Search, Grid Search, and Random Search* as shown in the Fig. 5.

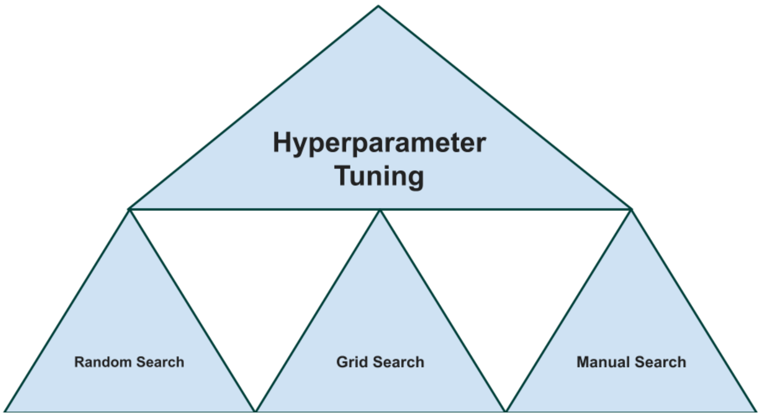


Figure 5: Types of hyperparameter tuning.

2.5.1 Manual Search

This process is an iterative process where the researcher adjusts a model's hyperparameters manually in search of optimal performance. This often involves intuition, domain expertise, and trial and error to find the configuration that best generalizes to unseen data. Due to its time-consuming and potentially resource-intensive nature, manual tuning might be less suitable for computationally demanding ensemble models and FFANNs. While the ANFIS is developed through Matlab provided UI. The tunning process for the ANFIS follows the iterative selection of the Membership Function along with adjusting the layer structure as represented in Fig. 6.

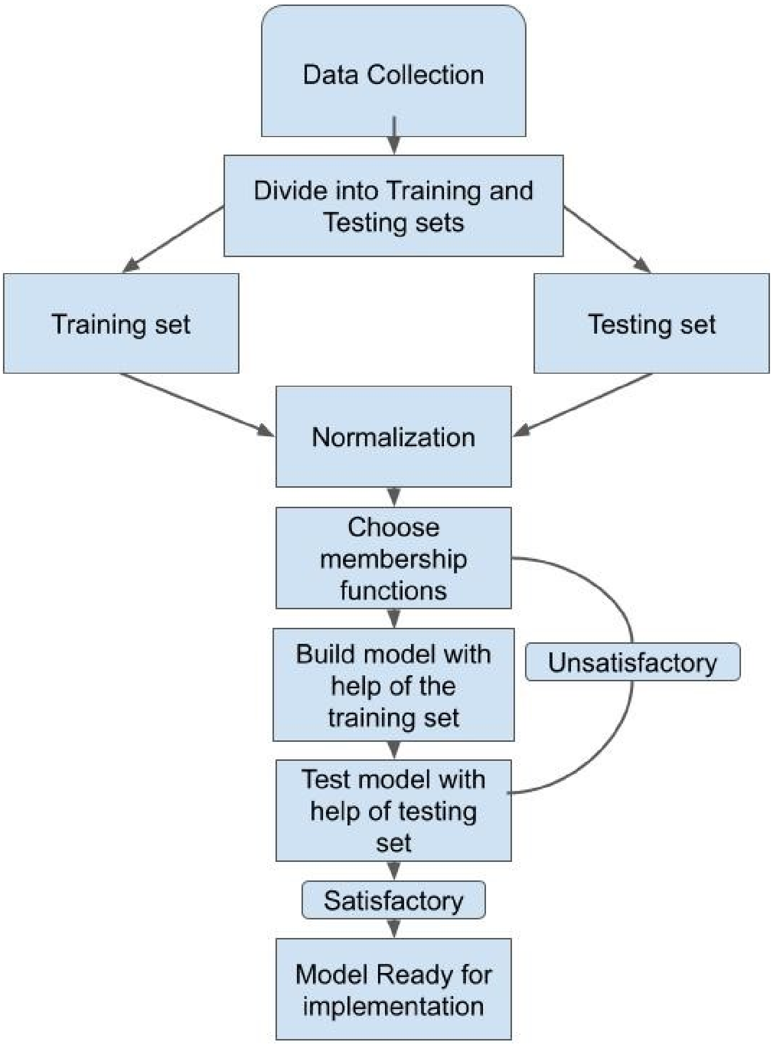

Data Collection
Divide into Training and Testing sets
Training set
Testing set
Normalization
Choose membership functions
Build model with help of the training set
Test model with help of testing set
Unsatisfactory
Satisfactory
Model Ready for implementation

Figure 6: ANFIS model development procedure.

2.5.2 Grid Search

This technique does not need intensive human involvement as the previous method. The tuning parameters with the value range for each one of them are defined at the beginning. Then with the help of built-in modules such as GridSearchCV from Python which is used for this work, we will eventually obtain the best combination of parameters for the given scopes. Since it tests every possible combination from the parametric pool.

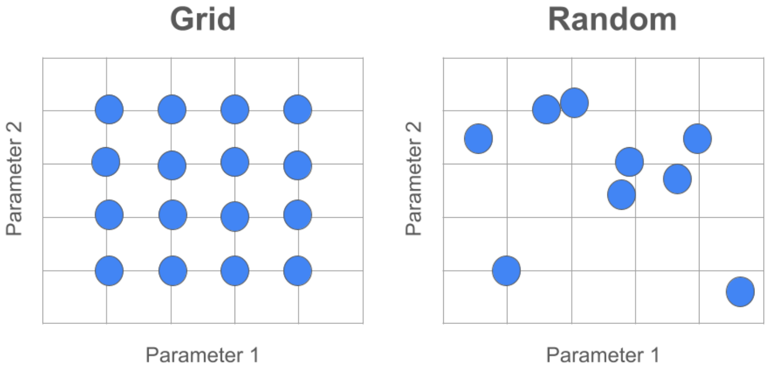


Figure 7: Grid vs. Random Search.

2.5.3 Random Search

From Fig. 7 the main difference between Grid and Random tuning processes can be obtained. The random search arbitrarily evaluates the possible combination of several or all parameters in the given range. On the other hand, this approach is not as computationally expensive as the initial methods. Best practice in the field suggests using the Random search to find the high performance range with the most influential parameters at first, then using Grid search to further examine the reduced extent []. This approach is used for finding the best combination of parameters along with their values for the RFs, XGBoost, and FFANN.

3 Chapter: Model Development, Validation and Discussion

3.1 Ensemble Models

The core idea of the ensemble algorithm is that instead of relying on a single model, which might have its own biases or limitations, harness the collective wisdom of a diverse set of models. In practice, it aggregates predictions through techniques like voting, averaging, or more sophisticated methods [8]. In the end, ensemble models often produce more accurate, robust, and stable solutions compared to any individual model on its own. There are two types of ensemble methods used for the research. Random Forests (RFs) represents the parallel learning type while XGBoost from the second group which sequentially adds up each learning model. This kind of selection is made to fully utilize the potential of the ensemble algorithm, as both learning models have above-average performance in their division. Random Forests has higher interpretability compared to the neural networks based algorithm, while the XGBoost offers several powerful techniques that give high regularization with above-average accuracy.

3.2 Random Forests

Random Forests is a popular type of bagging ensemble learning technique where numerous decision tree models are constructed for different subsets of the dataset independently during the training process and then combined at the end [8]. The individual models within Random Forests are Decision Trees (DTs). Specifically, DTs are simple models that ask a series of yes/no questions about a data point's features to arrive at a prediction as shown in Fig. 8 as orange dots.

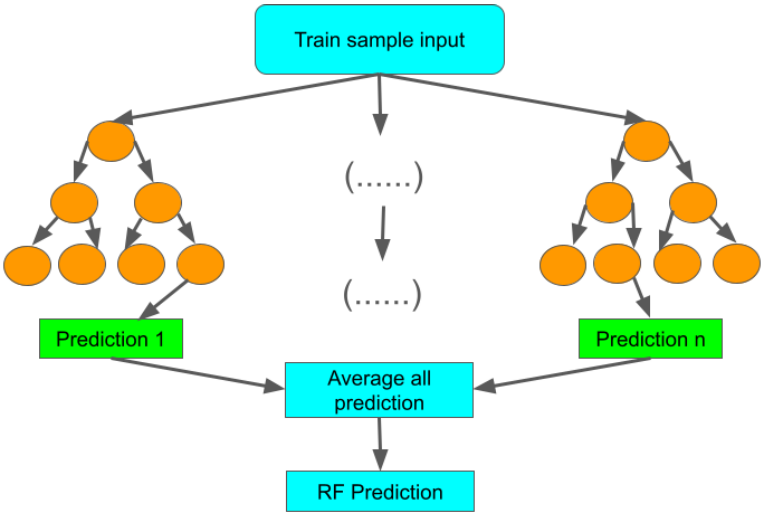


Figure 8: Model building process of Decision Tree.

3.2.1 Bootstrapping

Bagging, short for bootstrap aggregating, is a powerful technique that serves as the base of Random Forests Regression. As shown in Fig. 9 the bootstrapping involves creating multiple diverse datasets by randomly sampling the original data with replacement while the boosting algorithm builds the model iteratively by improving the previous model without modifying the dataset. In RFs, the randomness has two sources:

- Data Randomness: Each DT is trained on a unique, randomly sampled subset of the data.
- Feature Randomness: Only a random subset of features is considered when splitting nodes within a tree.

3.2.2 Background Mathematics

As we have pointed out the foundation of RF Regression lies in DT, which is constructed based on features that split the dataset into subsets, recursively forming a tree structure in the last. Since the tree grows the difference of members inside every subset will decrease. In

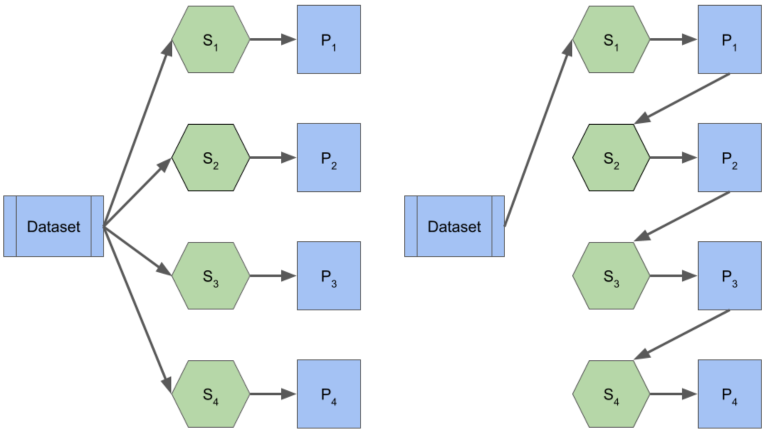


Figure 9: Bagging vs Boosting.

mathematical representation:

$$\Delta I_G = I_G(\text{parent}) - \sum_{i=1}^{k} \frac{N_i}{N} \cdot I_G(\text{child}_i) \tag{4}$$

ΔI_G measures how much the impurity has decreased after a split. $I_G(\text{parent})$ is the error of the parent node until the split. This value will gradually decrease with the help of the fraction representing the ratio of the number of instances (N_i) in the i_{th} child node to the total number of instances (N) in the parent node multiplied by the Gini impurity of the i_{th} child node after the split. A DT can also be represented as a set of rules that guide the traversal from the root to the leaves, where each leaf corresponds to a predicted value. The decision at each node is based on a specific feature and a threshold value. Mathematically, a DT can be defined as a function $f(x)$, where x is the input feature vector.

$$f(x) = \begin{cases} y_1 & \text{if } x_i \leq \text{threshold} \\ \\ y_2 & \text{if } x_i > \text{threshold} \end{cases} \tag{5}$$

Here, y_1 and y_2 are the predicted values at the corresponding branches.

As we explained above the RFs randomly selects the sample dataset with replacement from the original dataset, to ensure that each tree is trained on a unique subset. In the end, this diversity management policy contributes to the overall robustness of the model. In mathematical representation, let D_i be the bootstrapped subset for the i-th tree. The size of each subset is typically the same as the original dataset.

$$D_i = \{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), ..., (x_{iN}, y_{iN})\} \tag{6}$$

In addition to bootstrapping, RFs introduces feature randomness by considering only a random subset of features for each split in a decision tree. Also from the math perspective let's consider m to be the total number of features, and m' be the number of features considered at each split. The features used for each weak learner will be:

$$m' = \sqrt{m} \tag{7}$$

This further diversifies the individual trees, preventing them from becoming overly correlated and enhancing the model's generalization capabilities. By aggregating predictions from numerous DTs, RF regression achieves higher accuracy and improved resistance to outliers. This mitigates the risk of overfitting, enhances model stability, and results in robust predictions. Overall, the RFs regression algorithm can be summarized as follows:

1. For B iterations (where B is the number of trees):

   (a) Sample a bootstrapped dataset D_i.

   (b) Train a decision tree T_i on D_i, considering only a random subset of features at each split.

2. Predict the target variable Y by averaging predictions from all trees:

$$\hat{Y} = \frac{1}{B} \sum_{i=1}^{B} T_i(x) \tag{8}$$

Even though the algorithm has many advantages compared to the other algorithms. Recent advancements in RF regression focus on addressing its limitations and improving its performance [34]. Some notable developments include:

- Interpretability: The first one is to enhance the interpretability of RandomForest models when subdividing the sample features. There are techniques such as feature importance analysis and partial dependence plots that contribute to a better understanding of the model's decision-making process. Accordingly, we can modify the model parameters for a dataset to take the most out of it.
- Scalability: The second one is the relatively low scalability of Random Forests Regression. More efforts are required to allow the model to handle larger datasets efficiently. Some research suggested that distributed computing and parallelization techniques could accelerate model training and prediction speed [35, 36].

3.2.3 Model Development

With the help of random and grid hyperparameter tuning approaches discussed in the previous section, we have found the Random Forest Regression algorithm has three main parameters that influence the final result most. They are:

1. Number of Trees (n_estimators): Intuitively, more trees generally lead to better performance, by reducing overfitting and improving accuracy overall. However, after a certain

threshold, the improvement made on the overall performance of the model will become less, and excessive trees will increase computation time.

2. Tree Depth (max_depth): This parameter controls the complexity of individual trees. Specifically, the lower depths model will be less prone to overfitting, while deeper trees can capture more intricate patterns in the data but may lead to overfitting. Therefore, the optimal depth is essential for balancing underfitting and overfitting.

3. Number of Features at Each Split (max_features): By using the parameter we can decide the max limit of features that are randomly sampled when determining the best split for a node.

The main parameters for the final model architecture for each S-parameter are shown in the table.

Table 1: RFs Model Architecture for GaN-on-Diamond HEMT devices.

		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
n_estimators	All bias	380	420	350	510	370	420	390	450
max_depth	All bias	7	8	8	7	8	7	7	9
max_features	All bias	3	3	4	3	3	4	3	3

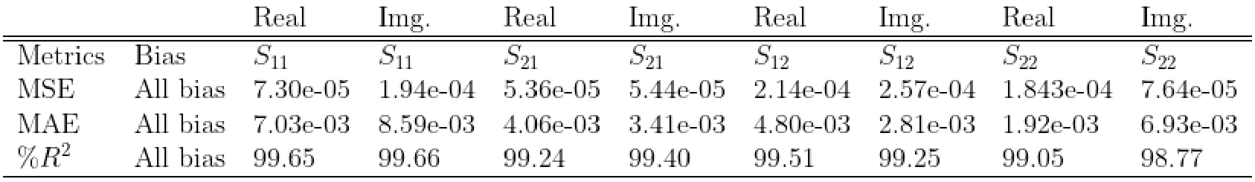
3.2.4 Random Forests Performance

Table 10 and 11 list evaluation metric result of the model. Except for the S_{22} the model has above 99% correlation score. The high correspondence of the simulation result towards

Table 2: RFs Model Architecture for GaN-on-Si HEMT devices.

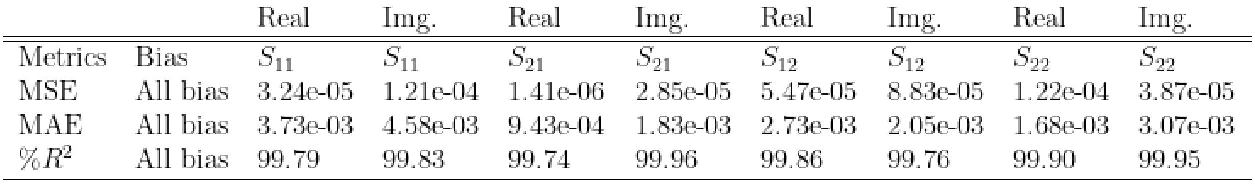
		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
n_estimators	All bias	460	470	395	460	450	580	480	550
max_depth	All bias	10	9	12	11	10	12	10	11
max_features	All bias	3	3	4	3	3	4	3	3

the measured data can also be noticed in Fig 14. Additionally, this algorithm is less sensitive to hyper-parameters compared to the XGBoost. Table 10 and 11 shows the performance of the method in terms of Mean Square Error (MSE), Mean Absolute Error (MAE), and the coefficient of regression (R^2).



		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	7.30e-05	1.94e-04	5.36e-05	5.44e-05	2.14e-04	2.57e-04	1.843e-04	7.64e-05
MAE	All bias	7.03e-03	8.59e-03	4.06e-03	3.41e-03	4.80e-03	2.81e-03	1.92e-03	6.93e-03
$\%R^2$	All bias	99.65	99.66	99.24	99.40	99.51	99.25	99.05	98.77

Figure 10: Performance of RF models on test samples for GaN-on-Diamond HEMT devices.

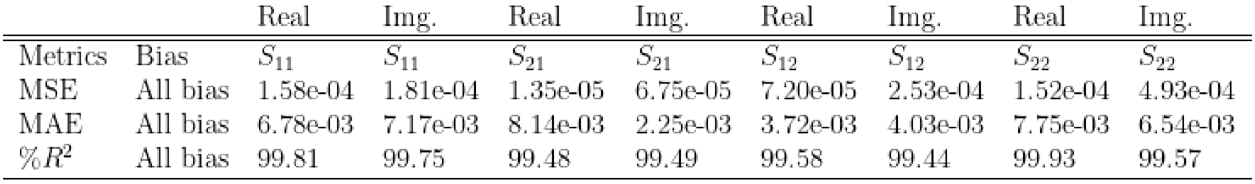


		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	3.24e-05	1.21e-04	1.41e-06	2.85e-05	5.47e-05	8.83e-05	1.22e-04	3.87e-05
MAE	All bias	3.73e-03	4.58e-03	9.43e-04	1.83e-03	2.73e-03	2.05e-03	1.68e-03	3.07e-03
$\%R^2$	All bias	99.79	99.83	99.74	99.96	99.86	99.76	99.90	99.95

Figure 11: Performance of RF models on train samples for GaN-on-Diamond HEMT devices.

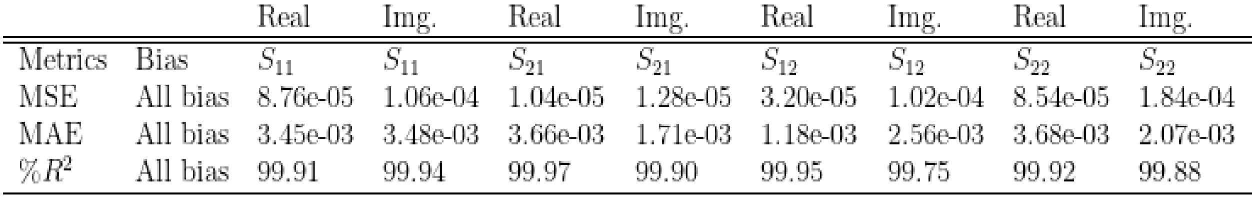
3.3 XGBoost

XGBoost (eXtreme Gradient Boosting) is a highly scalable and efficient machine-learning algorithm renowned for its exceptional performance in regression tasks. Initially proposed



		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	1.58e-04	1.81e-04	1.35e-05	6.75e-05	7.20e-05	2.53e-04	1.52e-04	4.93e-04
MAE	All bias	6.78e-03	7.17e-03	8.14e-03	2.25e-03	3.72e-03	4.03e-03	7.75e-03	6.54e-03
$\%R^2$	All bias	99.81	99.75	99.48	99.49	99.58	99.44	99.93	99.57

Figure 12: Performance of RF models on test samples for GaN-on-Si HEMT devices.



		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	8.76e-05	1.06e-04	1.04e-05	1.28e-05	3.20e-05	1.02e-04	8.54e-05	1.84e-04
MAE	All bias	3.45e-03	3.48e-03	3.66e-03	1.71e-03	1.18e-03	2.56e-03	3.68e-03	2.07e-03
$\%R^2$	All bias	99.91	99.94	99.97	99.90	99.95	99.75	99.92	99.88

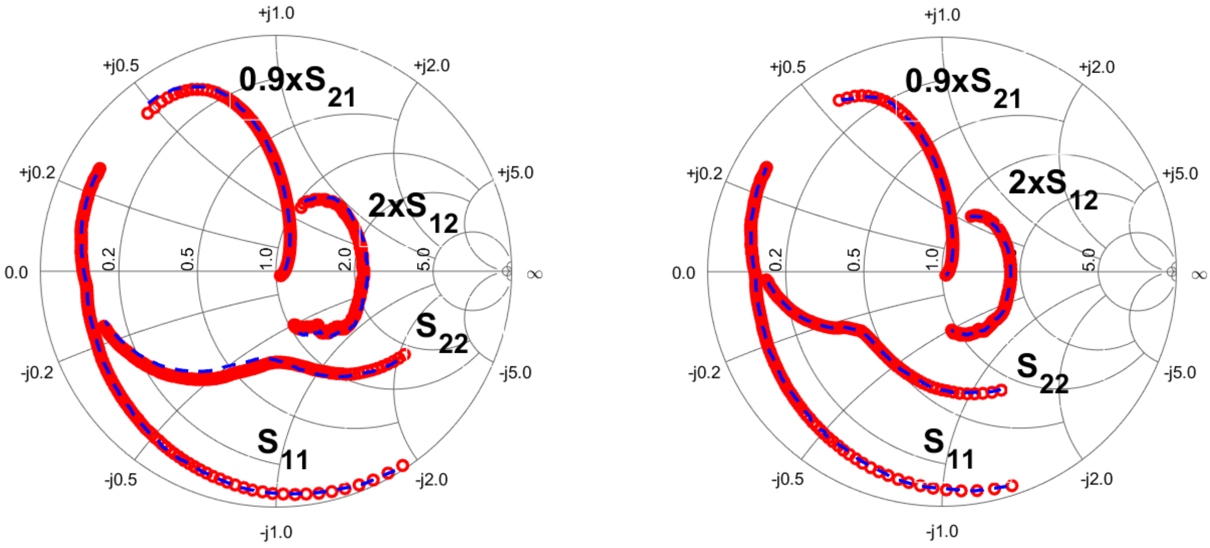
Figure 13: Performance of RF models on train samples for GaN-on-Si HEMT devices.

by Tianqi Chen, its widespread adoption in research and industry stems from its success in both competitive settings and real-world applications [37]. XGBoost's key strengths include a novel tree-learning algorithm for sparse data, a theoretically sound weighted quantile sketch for handling instance weights, and parallelized computation for accelerated model exploration. Additionally, utilizing out-of-core computation, XGBoost enables the processing of massive datasets even on typical desktop computers. These techniques synergize to create a powerful end-to-end system capable of handling vast datasets with exceptional resource efficiency [38]. In the following paragraphs, we will show the working principles and background math of the algorithm.

3.3.1 Background Mathematics

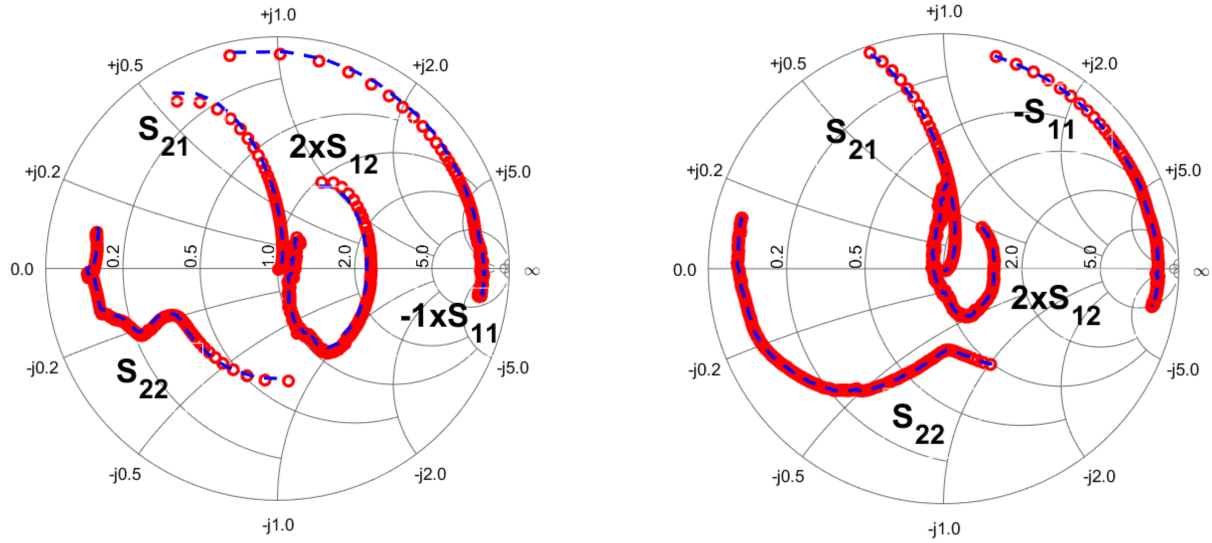
Gradient boosting builds an ensemble of weak learners and iteratively improves their predictions by focusing on the errors of previous models in the ensemble. Mathematically, it can be represented as:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{9}$$



(a) (b)

Figure 14: Smith chart plot comparison of the performance of RandomForest on the measured (red) and model predicted (blue) S-parameters for GaN-on-Diamond HEMT at (a) V_{GS} = -1 V and V_{DS} = 25 V and (b) V_{GS} = -2 V and V_{DS} = 2.5 V



(a) (b)

Figure 15: Smith chart plot comparison of the performance of RandomForest on the measured (red) and model predicted (blue) S-parameters for GaN-on-Si HEMT at (a) V_{GS} = -0.6 V, V_{DS} = 25 V and temperature = 175°C and (b) V_{GS} = -0.8 V, V_{DS} = 28 V and temperature = 25°C

where $\hat{y}_i$ is the predicted value for instance i, K is the number of trees in the ensemble and $f_k(x_i)$ represents the prediction of the k_th tree for instance i.

At each iteration, a new tree is added to the ensemble by minimizing a loss function that measures the discrepancy between the predicted and actual target values. Common loss functions in regression include mean squared error (MSE) and absolute loss, while classification tasks often employ logistic loss. XGBoost goes beyond standard gradient boosting by incorporating a regularization term into the objective function. This term penalizes overly complex trees, preventing overfitting and promoting models that generalize well to unseen data. The XGBoost objective function can be expressed as:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{10}$$

where $\mathcal{L}(\theta)$ represents the objective function to be minimized θ denotes the model parameters (including tree structures and weights) $L(y_i, \hat{y}_i)$ is the loss function for the i_th instance $\Omega(f_k)$ represents the regularization term for the k_th tree. The loss function $L(y_i, \hat{y}_i)$ captures the discrepancy between the predicted value and the true target value. XGBoost supports various loss functions tailored to specific problem types.

The regularization term, denoted by $\Omega(f_k)$, typically involves a combination of L1 and L2 penalties on the complexity of individual trees. L1 regularization encourages sparsity by penalizing the sum of absolute values of tree weights, promoting models with fewer influential splits. L2 regularization penalizes the squared values of weights, favoring simpler trees with smoother decision boundaries.

3.3.2 Gradient Descent and Tree Learning

XGBoost employs gradient descent optimization to minimize the objective function. In each iteration, the negative gradient of the objective function concerning the model parameters is computed. This gradient indicates the direction of the steepest descent in the loss landscape, guiding the update of the model parameters to reduce the loss.

However, directly minimizing the objective function for tree structures is computationally expensive. XGBoost addresses this challenge by adopting an approximate approach based on the second-order Taylor expansion of the loss function. This approximation leads to an efficient strategy for updating tree weights during the learning process.

3.3.3 Handling Sparse Data

XGBoost incorporates a novel tree learning algorithm specifically designed for sparse data, where many features have zero values for a significant number of instances. In the meantime, traditional tree learning algorithms can struggle with sparsity, as splitting on features with many missing values may not be informative.

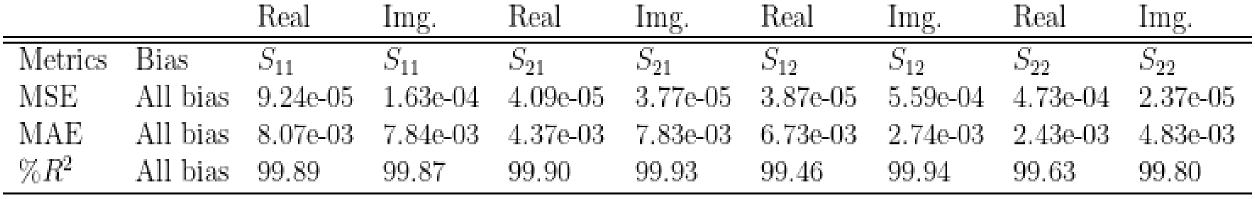
XGBoost addresses this by utilizing a technique called ”feature pre-selection.” Before growing the tree, it selects a subset of the most informative features for each split based on their potential gain in reducing the loss function. This pre-selection process significantly improves efficiency, especially when dealing with high-dimensional sparse datasets.

3.3.4 Parallel Computing for Scalability

XGBoost leverages parallel and distributed computing techniques to accelerate the training process, especially on large datasets. This allows for efficient training on clusters with multiple machines. The tree learning algorithm is parallelized, enabling the construction of trees in a distributed fashion across multiple cores or machines.

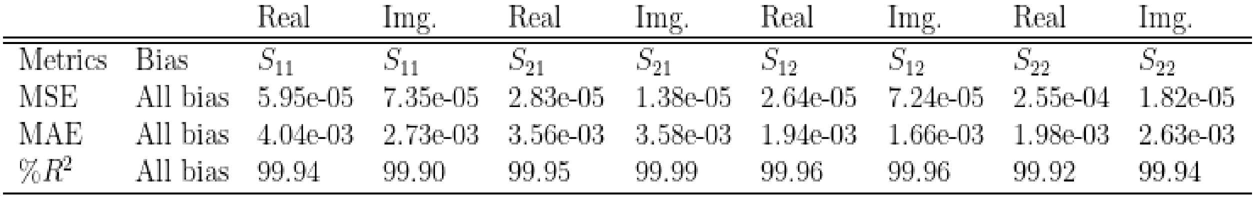
Additionally, XGBoost employs a technique called ”out-of-core” computation. This strategy allows for processing massive datasets that may not entirely fit into memory by loading data in smaller chunks as needed during the training process.

Table 16 and 17 show the stable and high accuracy of the technique. The hyperparameter tuning part of the method requires a deep understanding of the technique and the dataset. It consists of many parameters that can influence the model convergence rate and some parameters have a huge impact on the overfitting issue that may occur in the future.



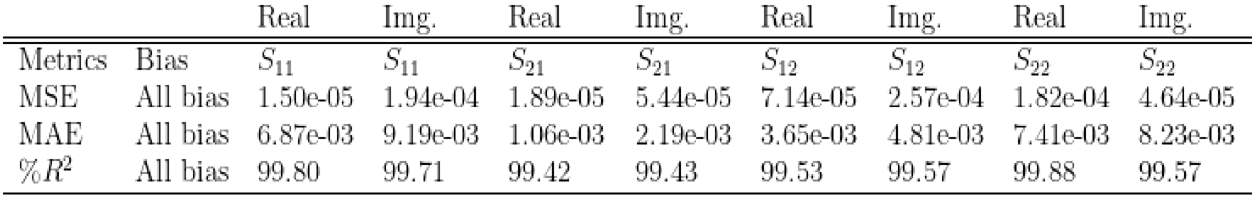
		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	9.24e-05	1.63e-04	4.09e-05	3.77e-05	3.87e-05	5.59e-04	4.73e-04	2.37e-05
MAE	All bias	8.07e-03	7.84e-03	4.37e-03	7.83e-03	6.73e-03	2.74e-03	2.43e-03	4.83e-03
$\%R^2$	All bias	99.89	99.87	99.90	99.93	99.46	99.94	99.63	99.80

Figure 16: Performance of XGBoost models on test samples for GaN-on-Diamond HEMT devices.



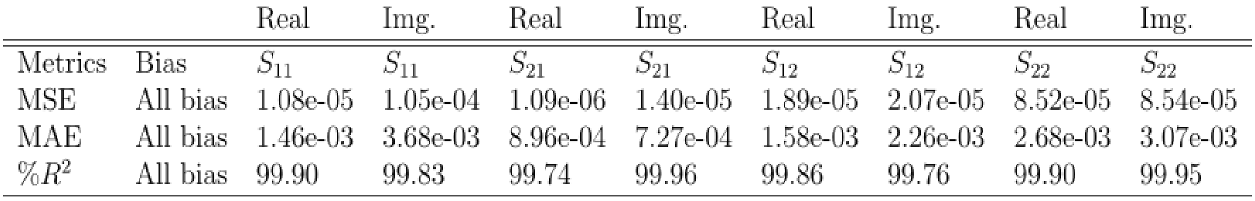
		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	5.95e-05	7.35e-05	2.83e-05	1.38e-05	2.64e-05	7.24e-05	2.55e-04	1.82e-05
MAE	All bias	4.04e-03	2.73e-03	3.56e-03	3.58e-03	1.94e-03	1.66e-03	1.98e-03	2.63e-03
$\%R^2$	All bias	99.94	99.90	99.95	99.99	99.96	99.96	99.92	99.94

Figure 17: Performance of XGBoost models on train samples for GaN-on-Diamond HEMT devices.



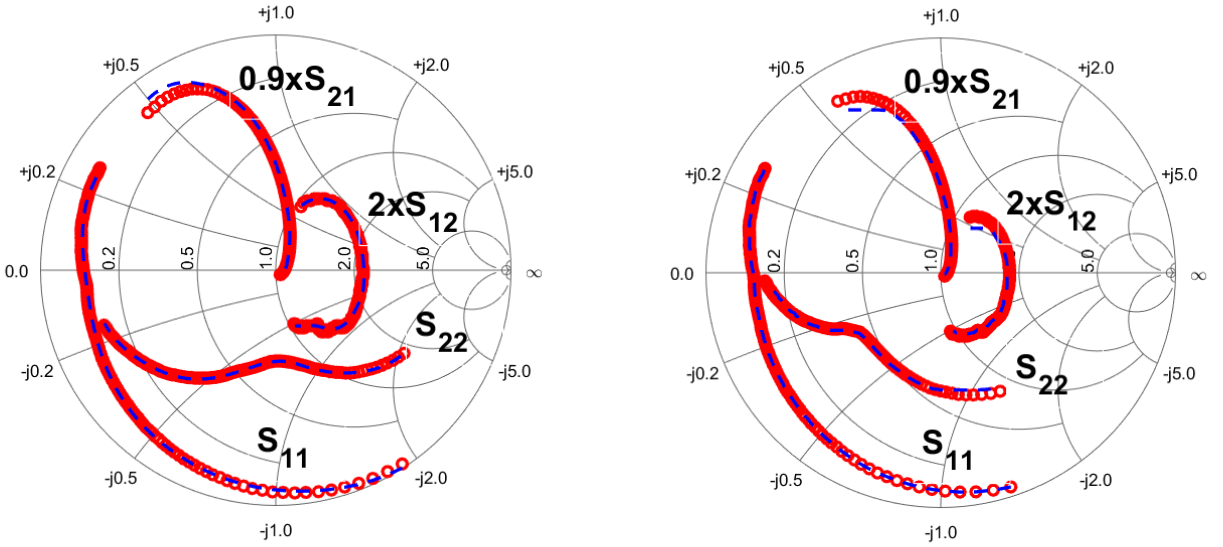
		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	1.50e-05	1.94e-04	1.89e-05	5.44e-05	7.14e-05	2.57e-04	1.82e-04	4.64e-05
MAE	All bias	6.87e-03	9.19e-03	1.06e-03	2.19e-03	3.65e-03	4.81e-03	7.41e-03	8.23e-03
$\%R^2$	All bias	99.80	99.71	99.42	99.43	99.53	99.57	99.88	99.57

Figure 18: Performance of XGBoost models on test samples for GaN-on-Si HEMT devices.



		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	1.08e-05	1.05e-04	1.09e-06	1.40e-05	1.89e-05	2.07e-05	8.52e-05	8.54e-05
MAE	All bias	1.46e-03	3.68e-03	8.96e-04	7.27e-04	1.58e-03	2.26e-03	2.68e-03	3.07e-03
$\%R^2$	All bias	99.90	99.83	99.74	99.96	99.86	99.76	99.90	99.95

Figure 19: Performance of XGBoost models on train samples for GaN-on-Si HEMT devices.



(a)

(b)

Figure 20: Smith chart plot comparison of the performance of XGBoost on the measured (red) and model predicted (blue) S-parameters for GaN-on-Diamond HEMT at (a) V_{GS} = -1 V and V_{DS} = 25 V and (b) V_{GS} = -2 V and V_{DS} = 2.5 V

3.4 ANFIS

Adaptive Neuro-Fuzzy Inference Systems (ANFIS) represent a hybrid modeling approach that merges the strengths of fuzzy logic and neural networks [39]. Fuzzy logic, inspired by the imprecision inherent in human reasoning, handles uncertainty through the concept of partial membership. This allows for the representation of linguistic variables (e.g., ”warm,” ”cool”) rather than solely crisp numerical values. In contrast, neural networks excel at learning complex patterns and relationships from data. ANFIS leverages the interpretability of fuzzy systems, where rules are expressed in an ’if-then’ format, alongside the adaptive learning capabilities of neural networks. This integration facilitates the fine-tuning of fuzzy membership functions

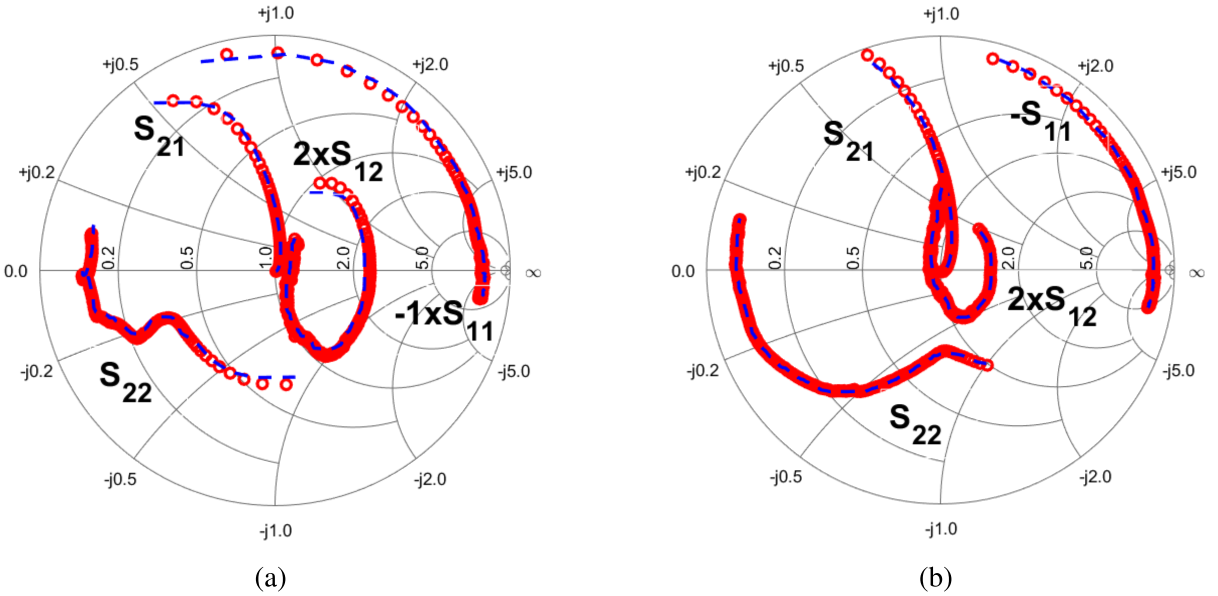


Figure 21: Smith chart plot comparison of the performance of XGBoost on the measured (red) and model predicted (blue) S-parameters for GaN-on-Si HEMT at (a) V_{GS} = -0.6 V, V_{DS} = 25 V and temperature = 175°C and (b) V_{GS} = -0.8 V, V_{DS} = 28 V and temperature = 25°C

and rules, enhancing the model's accuracy and its ability to explain its decision-making process with greater clarity.

3.4.1 ANFIS structure

ANFIS is structurally represented as a five-layered feed-forward network as shown in Fig. 22:

- Input layer: Each node in this layer represents the input parameter of the model. In the case of the GaN-on-Dimond HEMT, the input parameters are V_{GS}, V_{DS}, and the frequency [10, 39].

- Fuzzification layer: Membership functions are chosen in the stage to map the input value from 0 to 1. There are different types of membership functions available, for example, the Generalized bell, triangular, or Gaussian. After extensive research on the behavior and comparing the performance of all MFs, Gaussian MFs are chosen as it has the most accurate result.

- Rule Nodes: These nodes represent fuzzy rules derived from the combination of input features. Each rule node calculates the firing strength of the rule by multiplying the membership degrees from Layer 2, indicating the degree to which the corresponding rule is applicable.

- Normalization layer: Each node in this layer normalizes the firing strength from the previous layer by dividing it by the sum of all firing strengths. This provides a relative weight to each rule within the rule set.

- Output layer: This single node calculates the final weighted output. It combines the normalized firing strengths from Layer 4 with a corresponding linear function (consequent parameters), producing the final prediction.

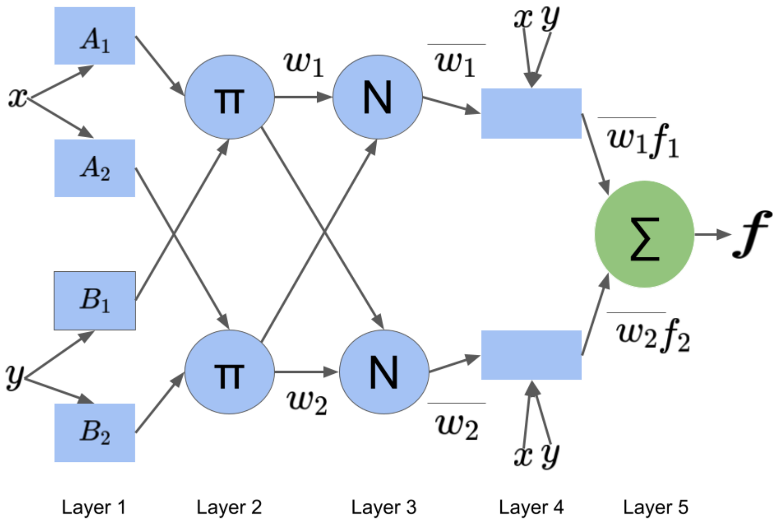


Figure 22: ANFIS layers.

3.4.2 ANFIS Model building

Matlab was used to program the ANFIS model. The model-building process for the ANFIS involves three steps. The first stage is the selection of the most suitable membership function for the dataset. Among the all available membership functions in the software, the Gaussian function performed stable and accurately for all biasing conditions. This is because the behavior of both devices has linear and non-linear parts and with other types of Membership functions only the half range is satisfied. With the Gaussian function, the smooth transaction between the linear to non-linear parts has been accomplished.

The next part of the model training part involves searching for the most appropriate layer structure. In this step, the built model should find the number of layers and the number of nodes in each layer that have high efficiency and avoid both underfitting and overfitting issues.

Considering the size and structure of each dataset differ significantly, the final structure of the models also has relative shifts. The optimal layer structure for the GaN on Diamond HEMT is 5-5-5, 3 layers with 5 nodes for each. For the GaN on Si HEMT, the complexity has increased a little bit, and it ends up as 4-4-5-4. There is one major barrier that hindered the development of the more complex model that has higher accuracy is the high resource consumption of the algorithm, this becomes even more obvious when training the second dataset that has more than double the size of the first one. The average training time for the GaN on Dimond HEMT is 12.54 minutes, and for the GaN on Si HEMT, the turnover time increased to 36.92 minutes. The addition of another layer or node will make the process even slower and eventually cause memory limit issues in the software. This is the major drawback of the algorithm compared to the other methods used in this paper.

3.4.3 ANFIS performance

The model built by the ANFIS has the most outperforming accuracy when the dataset is small. As the dataset becomes more complex and larger then the ANFIS becomes efficient compared to the other techniques. This is due to that the ANFIS has a high demand for computational resources compared to others and the Matlab tool for building the ANFIS model has a limit on memory use. This result is also obtained in another research that focuses only on comparison between ANFIS and ANN [40].

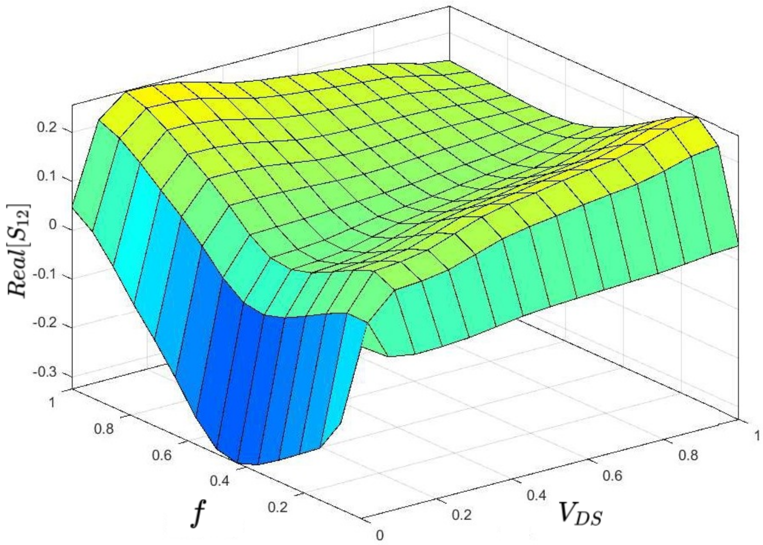
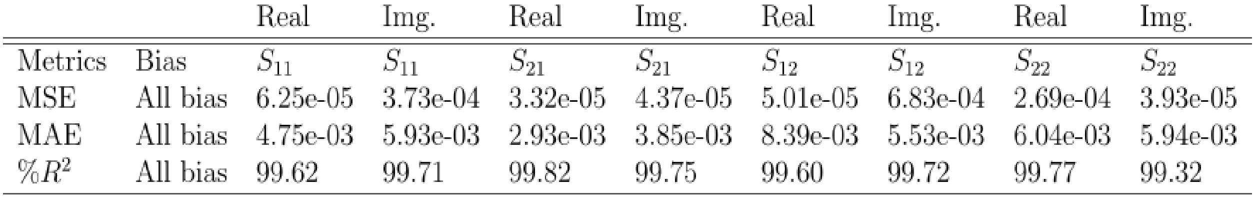
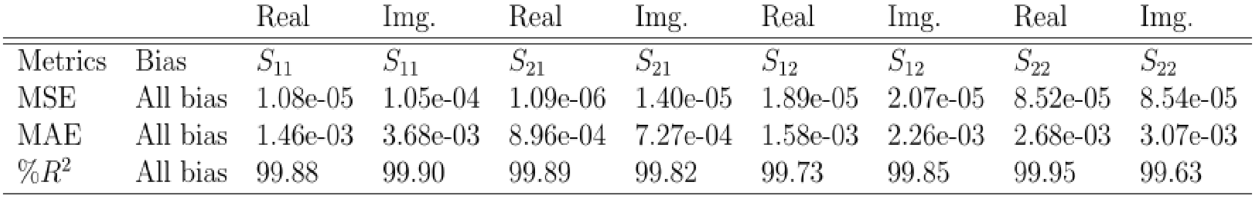


Figure 23: Surface representation of the ANFIS model.



		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	6.25e-05	3.73e-04	3.32e-05	4.37e-05	5.01e-05	6.83e-04	2.69e-04	3.93e-05
MAE	All bias	4.75e-03	5.93e-03	2.93e-03	3.85e-03	8.39e-03	5.53e-03	6.04e-03	5.94e-03
$\%R^2$	All bias	99.62	99.71	99.82	99.75	99.60	99.72	99.77	99.32

Figure 24: Performance of ANFIS models on test samples for GaN-on-Diamond HEMT devices.



		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	1.08e-05	1.05e-04	1.09e-06	1.40e-05	1.89e-05	2.07e-05	8.52e-05	8.54e-05
MAE	All bias	1.46e-03	3.68e-03	8.96e-04	7.27e-04	1.58e-03	2.26e-03	2.68e-03	3.07e-03
$\%R^2$	All bias	99.88	99.90	99.89	99.82	99.73	99.85	99.95	99.63

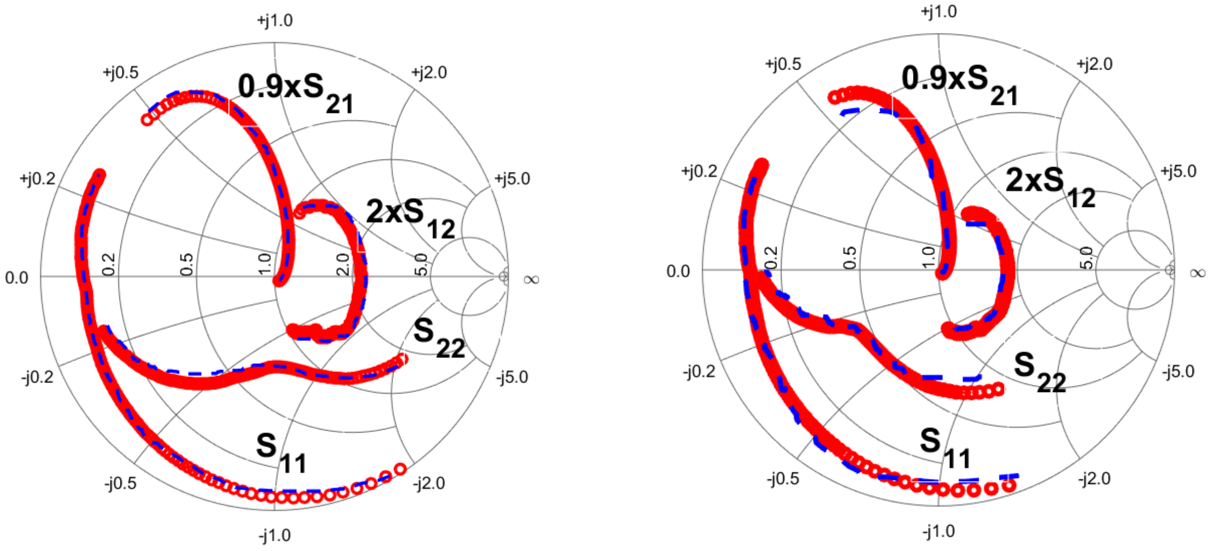
Figure 25: Performance of ANFIS models on train samples for GaN-on-Diamond HEMT devices.

		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	3.00e-04	2.83e-04	4.09e-05	9.40e-05	8.18e-05	2.60e-04	2.22e-04	4.74e-05
MAE	All bias	7.79e-03	9.48e-03	1.09e-03	2.38e-03	3.71e-03	7.82e-03	7.80e-03	6.43e-03
$\%R^2$	All bias	99.68	99.63	98.92	99.26	99.59	99.48	99.63	99.58

Figure 26: Performance of ANFIS models on test samples for GaN-on-Si HEMT devices.

		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	2.27e-04	1.26e-04	3.89e-05	7.35e-05	1.87e-05	1.09e-04	8.62e-05	3.28e-05
MAE	All bias	2.74e-03	2.70e-03	1.00e-03	1.19e-03	1.45e-03	2.63e-03	2.62e-03	2.73e-03
$\%R^2$	All bias	99.74	99.81	98.96	99.52	99.91	99.91	99.97	99.98

Figure 27: Performance of ANFIS models on train samples for GaN-on-Si HEMT devices.



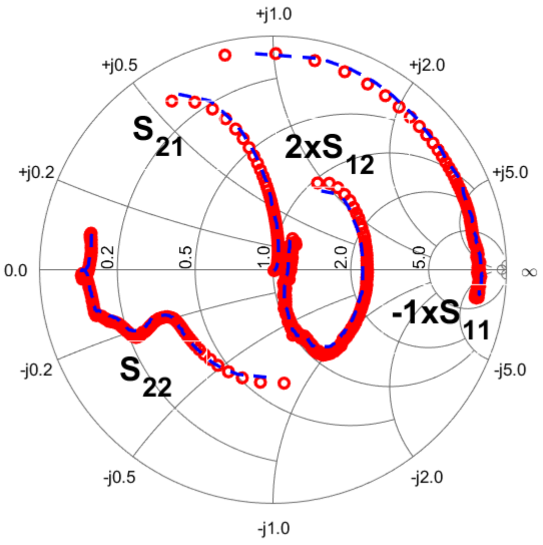
(a)

(b)

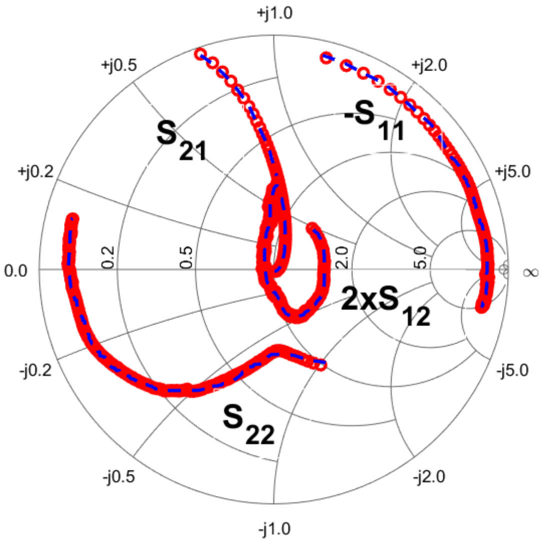
Figure 28: Smith chart plot comparison of the performance of ANFIS on the measured (red) and model predicted (blue) S-parameters for GaN-on-Diamond HEMT at (a) V_{GS} = -1 V and V_{DS} = 25 V and (b) V_{GS} = -2 V and V_{DS} = 2.5 V

3.5 Feed Forward ANN

Feed Forward artificial neural networks (ANNs) offer a powerful and versatile framework for regression problems [11, 41, 42]. Unlike traditional linear regression models, they excel at capturing complex, nonlinear relationships within datasets, and this is crucial when modeling the small signal mode of the GaN HEMT as usually has non-leaner behavior as the input voltages



(a)



(b)

Figure 29: Smith chart plot comparison of the performance of ANFIS on the measured (red) and model predicted (blue) S-parameters for GaN-on-Si HEMT at (a) V_{GS} = -0.6 V, V_{DS} = 25 V and temperature = 175°C and (b) V_{GS} = -0.8 V, V_{DS} = 28 V and temperature = 25°C

increase.

3.5.1 Working principles of Feed-forward ANN

Generally, the structure of FFNN comprises layers that are interconnected computational units, known as neurons [11, 43]. Input features are propagated through these layers, transforming weights and activation functions. The output layer generates a continuous numerical prediction, which is compared to the target value during training. Through the iterative process of backpropagation, the network's weights are optimized to minimize the error between the predicted and actual values.

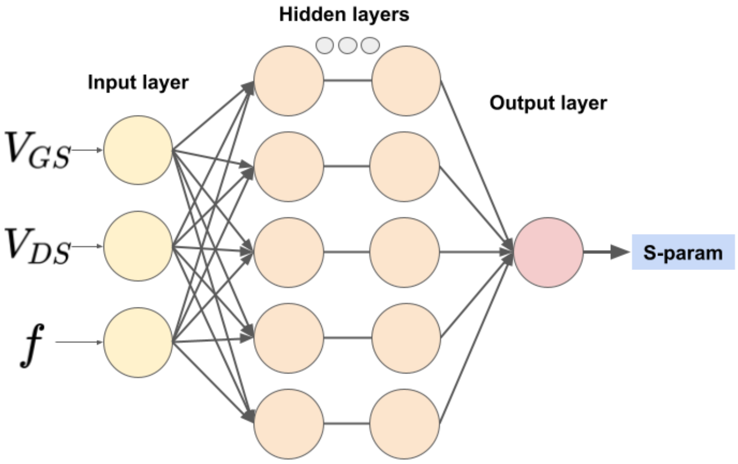


Figure 30: Structure of the Feed-forward ANN model for the GaN on Diamond HEMT.

3.5.2 Background Mathematics

Generally, each neuron in an FFANN computes an output y based on its input values, weights, and biases [11]. The computation model of a neuron can be simply expressed as:

$$y = \sigma \left(\sum_{i=1}^{n} w_i x_i + b \right) \tag{11}$$

where:

- $x_1, x_2, \ldots, x_n$ are the inputs to the neuron.
- $w_1, w_2, \ldots, w_n$ are the weights assigned to the inputs.
- b is the bias term.
- σ represents the activation function, which will be discussed in the next section explicitly.

3.5.3 Model building process of Feed-forward ANN

The model architecture design for Feed-forward ANN involves a similar process as the ANFIS model but with its features. Both algorithms have the step that needs to determine the number of hidden layers and the number of neurons within each layer. As for the Feed-forward ANN, there is an additional step that selects activation functions for neurons. The method has common activation functions as shown by [44]:

- Rectified Linear Unit (ReLU): $ReLU(x) = max(0, x)$. It's simple, computationally efficient, and often helps mitigate the vanishing gradient problem in deep networks. However, ReLU can suffer from the "dying ReLU" issue where neurons become inactive for negative inputs.

- Sigmoid: The sigmoid function, denoted by $S(x)$, is defined as:

  $$S(x) = \frac{1}{1 + e^{-x}} \tag{12}$$

  Sigmoid smoothly maps inputs to the range (0, 1). It's a popular activation function at first, but prone to vanishing gradients, especially in the saturation regions (near 0 or 1), which can hinder learning lastly.

- Hyperbolic Tangent (Tanh):

  $$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \tag{13}$$

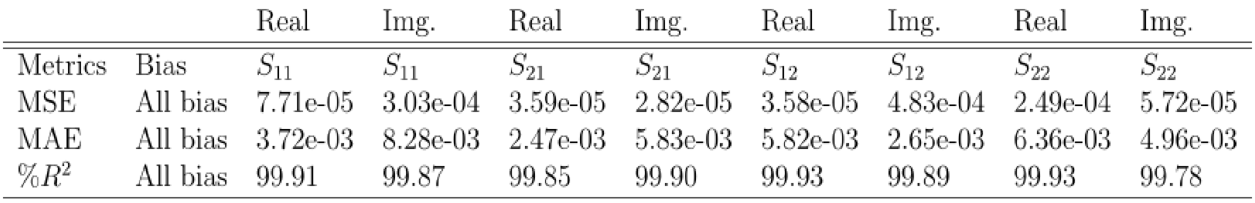
  Tanh is similar to sigmoid but maps to the range (-1, 1), making it zero-centered. This property can lead to faster convergence during training. Like sigmoid, it can still suffer from vanishing gradients in certain regions.

Tanh is chosen for both device modeling tasks after the consideration that datasets reprocessed to uniformly distributed between the range (-1, 1). Additionally, the zero-centered nature of the Tanh function results in stronger gradients around zero, potentially leading to faster training compared to sigmoid.

As we discussed above the training process starts by feeding the training data into the ANN, calculating errors, and using backpropagation to update the weights of the network iteratively. For the error calculation, we have used Mean Squared Absolute Error as the loss function. L1/L2 regularization is also included to reduce overfitting.

3.5.4 Performance of Feed-forward ANN

The architecture of the FFANN for GaN on Diamond HEMT is 5-6-6-4, and for the GaN on Si HEMT, the complexity increased to 6-5-5-6. Overall accuracy of the models is high as shown in Table 31 to Table 34. The Smith chart also gives high correspondence between the measured values and simulated results. However, there is a noticeable divergence at the high-frequency range. This might be that the model has a little inclination to overfitting, and could not have high generalization. Compared to other ML algorithms used in this work, the FFANN has the highest model build-up speed. This advantage is significantly noticeable when compared to the same neural networks base ANFIS techniques. With an average turnover time of 1.2 minutes for GaN on Diamond HEMT and 3.5 minutes for GaN on Si HEMT.



		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	7.71e-05	3.03e-04	3.59e-05	2.82e-05	3.58e-05	4.83e-04	2.49e-04	5.72e-05
MAE	All bias	3.72e-03	8.28e-03	2.47e-03	5.83e-03	5.82e-03	2.65e-03	6.36e-03	4.96e-03
$\%R^2$	All bias	99.91	99.87	99.85	99.90	99.93	99.89	99.93	99.78

Figure 31: Performance of FFANN models on test samples for GaN-on-Diamond HEMT devices.

		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	5.73e-05	2.93e-04	1.85e-05	1.28e-05	2.03e-05	3.02e-05	4.94e-05	4.58e-05
MAE	All bias	2.62e-03	3.84e-03	2.02e-03	3.74e-04	1.03e-03	1.85e-03	3.75e-03	2.85e-03
$\%R^2$	All bias	99.92	99.92	99.89	99.97	99.95	99.92	99.94	99.89

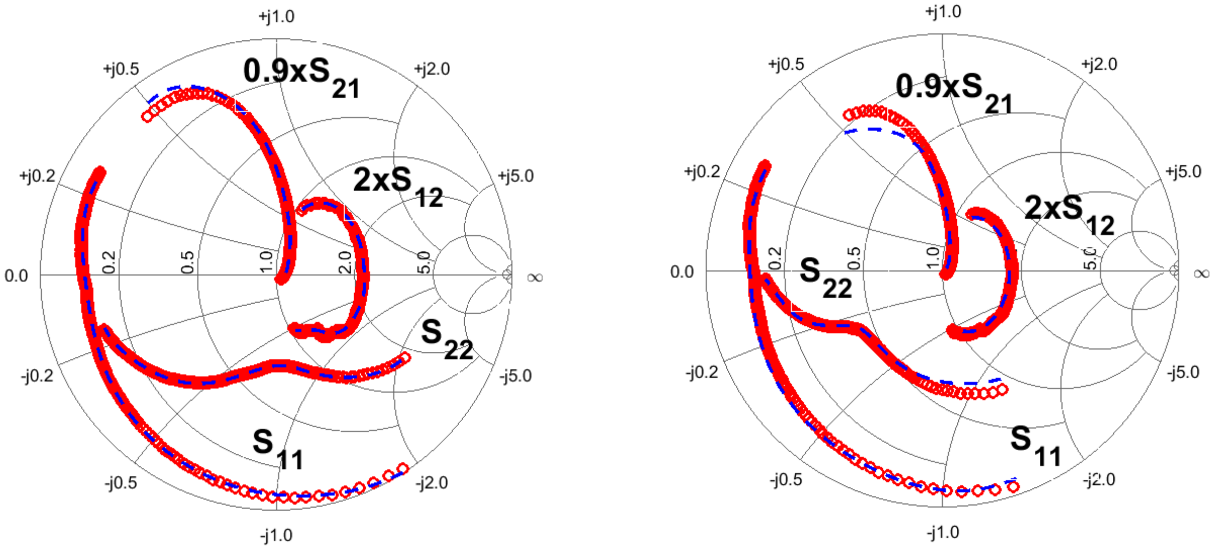
Figure 32: Performance of FFANN models on train samples for GaN-on-Diamond HEMT devices.

		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	7.37e-05	8.36e-05	3.67e-05	6.38e-05	7.39e-05	4.75e-04	1.11e-04	2.47e-05
MAE	All bias	5.73e-03	4.47e-03	5.82e-03	4.85e-03	4.29e-03	5.94e-03	5.68e-03	2.83e-03
$\%R^2$	All bias	99.85	99.79	99.72	99.80	99.66	99.59	99.84	99.87

Figure 33: Performance of FFANN models on test samples for GaN-on-Si HEMT devices.

		Real	Img.	Real	Img.	Real	Img.	Real	Img.
Metrics	Bias	S_{11}	S_{11}	S_{21}	S_{21}	S_{12}	S_{12}	S_{22}	S_{22}
MSE	All bias	5.64e-04	5.64e-05	2.94e-05	5.86e-05	4.96e-05	3.65e-04	9.65e-05	5.78e-05
MAE	All bias	4.78e-03	2.59e-03	5.02e-03	3.54e-03	2.40e-03	2.94e-03	1.58e-03	1.72e-03
$\%R^2$	All bias	99.88	99.86	99.76	99.90	99.81	99.77	99.90	99.94

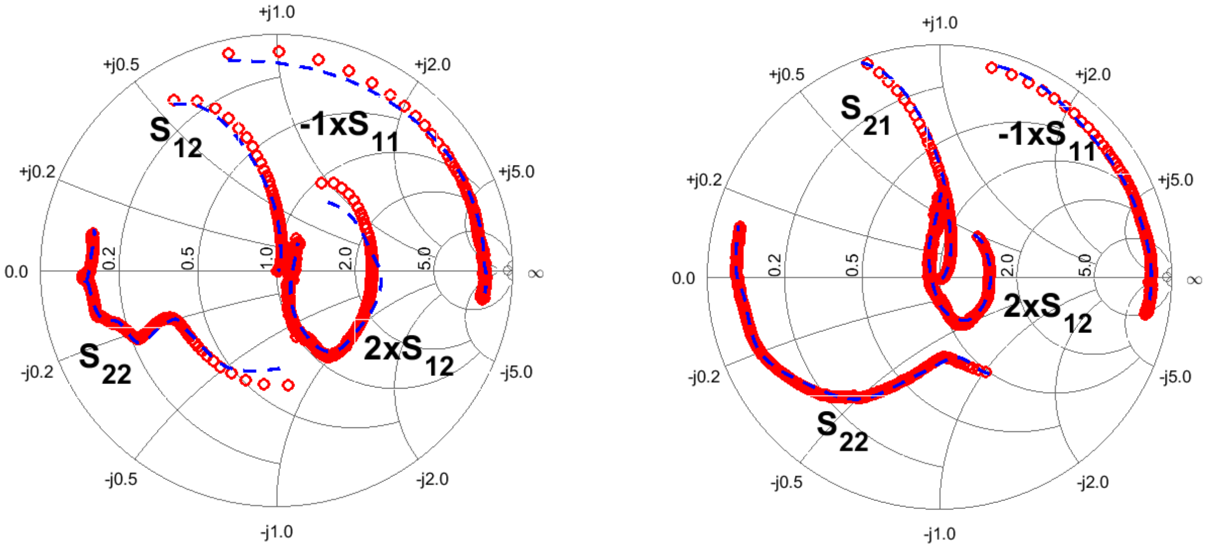
Figure 34: Performance of FFANN models on train samples for GaN-on-Si HEMT devices.



(a)

(b)

Figure 35: Smith chart plot comparison of the performance of FFNN on the measured (red) and model predicted (blue) S-parameters for GaN-on-Diamond HEMT at (a) V_{GS} = -1 V and V_{DS} = 25 V and (b) V_{GS} = -2 V and V_{DS} = 2.5 V



(a)

(b)

Figure 36: Smith chart plot comparison of the performance of FFNN on the measured (red) and model predicted (blue) S-parameters for GaN-on-Si HEMT at (a) V_{GS} = -0.6 V, V_{DS} = 25 V and temperature = 175°C and (b) V_{GS} = -0.8 V, V_{DS} = 28 V and temperature = 25°C

4 Chapter: Conclusion

After the comprehensive comparison between the ensemble and neural networks we have demonstrated the advantages and disadvantages of the modeling approach for Small Signal Modeling of Gallium Nitride High Electron Mobility Transistors. Individual evaluations revealed XGBoost's exceptional accuracy and RandomForest's robust resistance to overfitting. While ANFIS has high accuracy when the dataset is small, and FFANN has stable high performance in all scenarios, Crucially, the Ensemble models outperformed Neural Networks models in terms of robustness, and are not prone to overfitting. MSE, MAE, R^2 indicates that the Obtained SSMs for GaN HEMT have high accuracy. Additionally, the different modeling instruments and approaches give future researchers a better introduction to the challenges. This work will also serve as a good foundation for the LLM of the device as it has accurately demonstrated the linear responses and stability of devices. Even though the models developed for GaN on Si HEMT included the test on a temperature that is not in the range of the training phase. The models still have precise results which unveiled the high standard of the work.

References

[1] P. M. Tomé, F. M. Barradas, L. C. Nunes, J. L. Gomes, T. R. Cunha, and J. C. Pedro, "Characterization, modeling, and compensation of the dynamic self-biasing behavior of gan hemt-based power amplifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 69, no. 1, pp. 529–540, 2020.

[2] G. Lv, W. Chen, X. Liu, F. M. Ghannouchi, and Z. Feng, "A fully integrated c-band gan mmic doherty power amplifier with high efficiency and compact size for 5g application," *IEEE Access*, vol. 7, pp. 71665–71674, 2019.

[3] A. U. H. Pampori, S. A. Ahsan, and Y. S. Chauhan, "Modeling the impact of dynamic fin-width on the i–v, c–v and rf characteristics of gan fin–hemts," *IEEE Transactions on Electron Devices*, vol. 69, no. 5, pp. 2275–2281, 2022.

[4] N. Islam, M. F. P. Mohamed, M. F. A. J. Khan, S. Falina, H. Kawarada, and M. Syamsul, "Reliability, applications and challenges of gan hemt technology for modern power devices: A review," *Crystals*, vol. 12, no. 11, p. 1581, 2022.

[5] A. Helali, M. Gassoumi, M. Gassoumi, and H. Maaref, "Design and optimization of lna amplifier based on hemt gan for x-band wireless-communication and iot applications," *Silicon*, vol. 13, no. 8, pp. 2645–2653, 2021.

[6] D. Kuchta, D. Gryglewski, and W. Wojtasiak, "A gan hemt amplifier design for phased array radars and 5g new radios," *Micromachines*, vol. 11, no. 4, p. 398, 2020.

[7] S. Ganguly, K. M. Bothe, A. Niyonzima, T. Smith, Y. Liu, J. Fisher, F. Radulescu, D. A. Gajewski, S. T. Sheppard, J. W. Milligan, *et al.*, "Dc and rf reliability assessment of 5g-mmw capable gan hemt process," in *2022 IEEE International Reliability Physics Symposium (IRPS)*, pp. 11B–5, IEEE, 2022.

[8] S. J. Rigatti, "Random forest," *Journal of Insurance Medicine*, vol. 47, no. 1, pp. 31–39, 2017.

[9] S. Husain, M. Hashmi, and F. M. Ghannouchi, "Comprehensive investigation and comparative analysis of machine learning-based small-signal modelling techniques for gan hemts," *IEEE Journal of the Electron Devices Society*, vol. 10, pp. 1015–1032, 2022.

[10] B. Kadirbay, S. Husain, and M. Hashmi, "Small-signal modeling of gan-on-diamond hemt using anfis method," in *2023 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–6, IEEE, 2023.

[11] S. Husain, B. Kadirbay, A. Jarndal, and M. Hashmi, "Comprehensive investigation of ann algorithms implemented in matlab, python and r for small-signal behavioral modeling of gan hemts," *IEEE Journal of the Electron Devices Society*, 2023.

[12] A. Edwards, B. Geller, and I. C. Kizilyalli, "Extracting a nonlinear electro-thermal model for a gan hfet.," *Microwave Journal*, vol. 51, no. 2, 2008.

[13] G. Bhargava, V. Vadalà, S. Majumdar, and G. Crupi, "Auto-encoder based hybrid machine learning model for microwave scaled gaas phemt devices," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 32, no. 11, p. e23339, 2022.

[14] Z. Wen, Y. Xu, C. Wang, X. Zhao, and R. Xu, "An efficient parameter extraction method for gan hemt small-signal equivalent circuit model," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 30, no. 1, p. e2127, 2017.

[15] Z. Marinković, G. Crupi, A. Caddemi, G. Avolio, A. Raffo, V. Marković, G. Vannini, and D. M.-P. Schreurs, "Neural approach for temperature-dependent modeling of gan hemts," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*,

vol. 28, no. 4, pp. 359–370, 2015.

[16] S. Husain, A. Jarndal, M. Hashmi, and F. M. Ghannouchi, "Accurate, efficient and reliable small-signal modelling approaches for gan hemts," *IEEE Access*, 2023.

[17] A. Khusro, M. S. Hashmi, A. Q. Ansari, and M. Auyenur, "A new and reliable decision tree based small-signal behavioral modeling of gan hemt," in *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 303–306, IEEE, 2019.

[18] N. Sharma, Y. Gupta, A. Sharma, and H. Sharma, "Thermal modeling of the gan hemt device using decision tree machine learning technique," in *Communication and Intelligent Systems: Proceedings of ICCIS 2020*, pp. 13–20, Springer, 2021.

[19] S. Husain, K. Begaliyeva, A. Aitbayev, M. A. Chaudhary, and M. Hashmi, "Decision tree based small-signal modelling of gan hemt and cad implementation," pp. 1–6, 2022.

[20] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," *Journal of King Saud University-Computer and Information Sciences*, 2023.

[21] H. He and Y. Fan, "A novel hybrid ensemble model based on tree-based method and deep learning method for default prediction," *Expert Systems with Applications*, vol. 176, p. 114899, 2021.

[22] E. H. Alkhammash, M. Hadjouni, and A. M. Elshewey, "A hybrid ensemble stacking model for gender voice recognition approach," *Electronics*, vol. 11, no. 11, p. 1750, 2022.

[23] Z. Ma, S. Guo, G. Xu, and S. Aziz, "Meta learning-based hybrid ensemble approach for short-term wind speed forecasting," *IEEE Access*, vol. 8, pp. 172859–172868, 2020.

[24] Q. Xu, M. Wu, E. Khoo, Z. Chen, and X. Li, "A hybrid ensemble deep learning approach for early prediction of battery remaining useful life," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 1, pp. 177–187, 2023.

[25] P. Rathore, D. Kumar, J. C. Bezdek, S. Rajasegarar, and M. Palaniswami, "A rapid hybrid clustering algorithm for large volumes of high dimensional data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 641–654, 2018.

[26] Q. Wu, Y. Xu, J. Zhou, Y. Kong, T. Chen, Y. Wang, F. Lin, Y. Fu, Y. Jia, X. Zhao, *et al.*, "Performance comparison of gan hemts on diamond and sic substrates based on surface potential model," *ECS Journal of Solid State Science and Technology*, vol. 6, no. 12, p. Q171, 2017.

[27] Q. Wu, Y. Xu, Y. Chen, Y. Wang, W. Fu, B. Yan, and R. Xu, "A scalable multiharmonic surface-potential model of algan/gan hemts," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 3, pp. 1192–1200, 2017.

[28] Y. Chen, Y. Xu, J. Zhou, Y. Kong, T. Chen, Y. Zhang, B. Yan, and R. Xu, "Temperature-dependent small signal performance of gan-on-diamond hemts," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 33, no. 3, p. e2620, 2020.

[29] G. Gonzalez, *Microwave transistor amplifiers: analysis and design*, vol. 2. Prentice hall New Jersey, 1997.

[30] L. Pace, N. Defrance, A. Videt, N. Idir, J.-C. De Jaeger, and V. Avramovic, "Extraction of packaged gan power transistors parasitics using s-parameters," *IEEE Transactions on Electron Devices*, vol. 66, no. 6, pp. 2583–2588, 2019.

[31] B. A. Bagdad and F. Gamiz, "Characterization of semiconductor structures using scanning microwave microscopy technique," in *2017 Joint International EUROSOI Workshop and International Conference on Ultimate Integration on Silicon (EUROSOI-ULIS)*, pp. 200–

203, IEEE, 2017.

[32] P. M. Tomé, F. M. Barradas, L. C. Nunes, J. L. Gomes, T. R. Cunha, and J. C. Pedro, "Characterization, modeling, and compensation of the dynamic self-biasing behavior of gan hemt-based power amplifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 69, no. 1, pp. 529–540, 2020.

[33] S. Husain, B. Kadirbay, A. Jarndal, and M. Hashmi, "Comprehensive investigation of ann algorithms implemented in matlab, python, and r for small-signal behavioral modeling of gan hemts," *IEEE Journal of the Electron Devices Society*, vol. 11, pp. 559–572, 2023.

[34] C. Molnar, T. Freiesleben, G. König, J. Herbinger, T. Reisinger, G. Casalicchio, M. N. Wright, and B. Bischl, "Relating the partial dependence plot and permutation feature importance to the data generating process," in *World Conference on Explainable Artificial Intelligence*, pp. 456–479, Springer, 2023.

[35] F. Vázquez-Novoa, J. Conejero, C. Tatu, and R. M. Badia, "Scalable random forest with data-parallel computing," in *European Conference on Parallel Processing*, pp. 397–410, Springer, 2023.

[36] K. Senagi and N. Jouandeau, "Parallel construction of random forest on gpu," *The Journal of Supercomputing*, vol. 78, no. 8, pp. 10480–10500, 2022.

[37] 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*, pp. 785–794, 2016.

[38] S. Ramraj, N. Uzir, R. Sunil, and S. Banerjee, "Experimenting xgboost algorithm for prediction and classification of different datasets," *International Journal of Control Theory and Applications*, vol. 9, no. 40, pp. 651–662, 2016.

[39] J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[40] B. Kadirbay, S. Husain, A. Jarndal, and M. Hashmi, "Comparison of anfis and ann for small-signal modelling of gan hemt up to 40 ghz," in *2023 International Conference on Microelectronics (ICM)*, pp. 125–130, IEEE, 2023.

[41] M. H. Sazli, "A brief review of feed-forward neural networks," *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, vol. 50, no. 01, 2006.

[42] M. Talaat, M. Farahat, N. Mansour, and A. Hatata, "Load forecasting based on grasshopper optimization and a multilayer feed-forward neural network using regressive approach," *Energy*, vol. 196, p. 117087, 2020.

[43] S. Cheng, Y. Wu, Y. Li, F. Yao, and F. Min, "Twd-sfnn: Three-way decisions with a single hidden layer feedforward neural network," *Information Sciences*, vol. 579, pp. 15–32, 2021.

[44] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, "A review of activation function for artificial neural network," in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pp. 281–286, 2020.

Back Cover