Development and Deployment of CAD Integrable Machine Learning Based Modelling Techniques for Advanced RF Devices

Kashif Khan, Master in Electrical and Computer Engineering

Submitted in fulfilment of the

requirements for the degree of

Master of Science in Electrical and

Computer Engineering

School of Engineering and Digital Sciences

Department of Electrical and Computer Engineering

Nazarbayev University

53 Kabanbay Batyr,

Avenue, Astana,

Kazakhstan, 010000

Supervisors: Dr. Mohammad Hashmi and Dr. Galymzhan Nauryzbayev

April 2022

DECLARATION

I hereby, declare that this manuscript, entitled “Development and Deployment of CAD Integrable Machine Learning Based Modelling Techniques for Advanced RF Devices”, is the result 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 of author)

Name: Kashif Khan

Date: 01.05.2023

Abstract

This research investigates and advances effective machine learning (ML) methodologies for analyzing the performance of gallium nitride (GaN) high electron mobility transistors (HEMTs). A multilayer perceptron (MLP) framework within an artificial neural network (ANN) is utilized for the behavioral representation of a 2-mm GaN-on-silicon device. It is recognized that ANNs exhibit a dependence on initial weight and bias values. To address this limitation, multiple global optimization algorithms, including grey wolf optimization-particle swarm optimization (GWO-PSO), ant colony optimization (ACO), whale optimization algorithm (WOA), and ant lion optimization (ALO), are integrated into the MLP structure. The models are trained on an extensive dataset encompassing various operating conditions, such as ambient temperatures and bias voltages, within a frequency range of 0.1 to 26 GHz. Subsequently, these models are subjected to temperature interpolation and extrapolation tests, with the aim of determining their level of precision, robustness and stability. A strong correlation between the measured and modeled S-parameters across the entire frequency spectrum attests to the efficacy and resilience of the proposed methodologies. Moreover, the GWO-PSO-assisted ANN outperforms other models, as evidenced by mean squared error (MSE), mean absolute error (MAE), and coefficient of determination (%R^2) metrics.

Acknowledgment

I would like to express my deepest gratitude and appreciation to everyone who has supported and guided me throughout the course of my Master's thesis in Electrical and Computer Engineering.

First and foremost, I would like to extend my heartfelt thanks to my esteemed supervisors, Dr. Muhammad Hashmi, and Dr. Galymzhan Nauryzbayev for their invaluable guidance, unwavering support, and constant encouragement. Their expertise, patience, and constructive feedback have been instrumental in shaping my research and ensuring its success.

My sincere appreciation goes out to my fellow graduate students and friends, especially Saddam Husain, for their camaraderie, technical assistance, and countless stimulating discussions that have enriched my understanding of the subject matter. I would also like to acknowledge the Nazarbayev University for providing me with the necessary resources, facilities, funding and conducive environment to conduct my research.

Lastly, I owe my deepest gratitude to my family—my parents, and my siblings, for their love, encouragement, and belief in my abilities. Their support has been my source of strength and motivation during my academic journey.

This thesis would not have been possible without the collective efforts and support of all these individuals, and I am eternally grateful for their contributions.

List of Abbreviations

ACO	Ant Colony Optimization
ALO	Ant Lion Optimization
ANN	Artificial Neural Network
BP	Backpropagation
CAD	Computer-Aided Design
FET	Field Effect Transistor
GA	Genetic Algorithm
GaN	Gallium Nitride
GD	Gradient Descent
GO	Global Optimization
GRNN	General Regression Neural Network
GWO-PSO	Grey Wolf Optimization-Particle Swarm Optimization
HEMT	High Electron Mobility Transistors
LM	Levenberg-Marquardt
LSM	Large Signal Modelling
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NARX	Nonlinear Autoregressive with Exogenous Inputs
PA	Power Amplifier
PSO	Particle Swarm Optimization
SRM	Structural Risk Minimization
SSM	Small Signal Modelling
SVR	Support Vector Regression
VNA	Vector Network Analyzer
WOA	Whale Optimization Algorithm

List of Tables

List of Figures

Table of Contents

Chapter- 1 Introduction

1.1. General

As RF and microwave frequencies increasingly pervade various domains, including space, defense systems, and commercial wireless technologies, the study of these frequencies becomes progressively more complex and intricate. The demand for wireless device technology and the expectations of their performance levels have escalated exponentially. The amplifier circuits, with transistors as the crucial components, have become the very essence of wireless communication devices and have thus dominated the overall device performance in wireless systems. As a result, the behavioral modeling of transistors has become an essential aspect of time and cost-efficient design of amplifier circuits [1], with the combination of advanced wireless technologies and microwave frequencies making the modeling of transistors increasingly perplexing and challenging. This calls for the development of innovative solutions to improve model performance, as the complexity of transistor modeling becomes an ever more broaden area of Computer-Aided Design (CAD), driven by the rapid technological developments in semiconductor circuit technology.

The GaN High Electron Mobility Transistor (HEMT) is a type of semiconductor device that is becoming increasingly popular in communication and broadcasting systems. One of the key advantages of GaN HEMTs is the high power capability at high frequencies with significantly increased efficiency [2]. This is due to their unique electronic properties, which allow them to operate at higher voltages and temperatures than other semiconductor materials. In addition to their high power capabilities, GaN HEMTs also offer higher gain and better noise characteristics compared to other types of transistors. This makes them an ideal candidate to become an essential

building block in the design of integrated transceivers and low noise amplifiers. In general, the use of GaN HEMTs in the wide spectrum of broadcasting and communication applications offers several predominant benefits, including improved efficiency, high output power, and improved noise characteristics [3]. As such, they are likely to continue to be an important technology for these applications in the future.

The accuracy of the employed large-signal models of GaN HEMT devices affects the dependability of power amplifiers in broadcasting and wireless transmitters. When creating these models, it is crucial to factor in parasitic effects at high frequencies, self-heating under high-power drive, surface and buffer trapping, and memory effects, which are frequency-dependent phenomena that significantly impact the device's performance [4]. The accurate modeling of these effects is crucial in designing and optimizing power amplifiers (PAs) for wireless and broadcasting applications, ensuring their reliability and efficiency at high power levels while maintaining their performance over time. This is essential for guaranteeing the quality and dependability of wireless and broadcasting systems, which are pivotal for communication and other applications.

Number of modelling techniques have been reported in the past; Physics-based modeling, analytical modeling, Table-based, and Machine learning (ML) based modeling. Physics-based modeling involves deriving mathematical equations based on fundamental physical principles. This approach is often used in engineering and physical sciences, where the behavior of a system can be described by mathematical equations derived from basic physical laws. The advantage of physics-based modeling is its ability to capture the fundamental behavior of a system, which can provide insights into its underlying physical mechanisms. However, this approach can be challenging and time-consuming, as it often requires a deep understanding of the system and its physical principles [5]. A cost-effective option for implementation and development, table-based

modeling's discrete nature leads to lower convergence rates and speed, resulting in inadequate outcomes at higher frequencies of operation [6]. In contrast, analytical modeling relies on closed-form formulations and exhibits faster convergence rates and better prediction capabilities due to its continuous nature [7]. However, this approach requires substantial efforts and longer time for optimizing the fitting parameters of the model, which is a major drawback. On the other hand, ML based modeling of GaN HEMT is becoming popular now a days [8]-[30]. Specifically in ML based modeling, artificial neural network (ANN) based modeling is a popular approach, which involves training a neural network to learn the specific patterns and relationships in a dataset. ANN-based models are particularly useful in cases where the dependence of the output on the input variables is complex and non-linear. The advantage of ANN-based modeling is its ability to learn from data without requiring explicit knowledge of the physical principles. So this thesis is focused on ML based modeling of GaN HEMTs to check their behavior.

1.2. Literature Review

Lately, there has been a significant upswing in interest in using learning-based models that employ artificial neural networks (ANNs). This technique holds great promise due to its simplicity of development and its ability to achieve an excellent trade-off between precision and computational time. The impenetrability of ANNs minimizes the search cost of the appropriate algorithm for estimating model parameters. ANNs offer a higher convergence rate than analytical modeling, and an appropriate choice of model topology and activation function can lead to improved predictive accuracy [9]. The primary advantage of using ANNs is their ability to learn the interdependency of input and output data, which makes them efficient in predicting output values for any input. During training, ANNs calculate the resulting output at specific input values, compare it with measured data, and calculate the error. The error is propagated in a backward

direction through the system, leading to the adjustment of weights and biases for improved fitting. This depicts the typical method for the widely-utilized backpropagation (BP) artificial neural network (ANN).

Backpropagation as a gradient method is primarily limited by its elevated sensitivity to the initial guess, which can cause the algorithm to converge to local minima instead of the global minimum. This problem becomes more pronounced in larger scale ANN models with non-linear problems such as behavioral modeling [10]. There are several ways to overcome this limitation, such as finding a proper initial guess, optimizing the model's topology through tuning, objective function modification, or changing the activation function. However, these methods may not always be effective, and the process of training may need to be re-initialized continuously to achieve optimal fitting. One alternative approach to training neural networks is to use global optimization techniques such as particle swarm optimization (PSO) [11] and genetic algorithms (GA). These methods can effectively search the entire solution space and find the global minimum, avoiding the problem of getting stuck in local minima.

In the study [12]-[13], the authors conducted a comparison between the equivalent circuit and ANN based modeling. The paper emphasized that the representation of S22 behavior is arduous mainly because of the Kirk effect. The researchers in [14], presented a straightforward ANN based behavioral modeling, which employs the popular Multilayer Perceptron (MLP) architecture. In order to mitigate the intricacy of models identified in prior research [15]-[16], several artificial neural network (ANN) structures have been suggested. The operation of transistors in high-power applications is substantially influenced by temperature and frequency, necessitating the proposal and evaluation of models at elevated temperatures, with subsequent presentation of electro-thermal modeling in [17].

In [18], a hybrid small-signal model parameter extraction method for GaN HEMT on Si and SiC substrates was developed using global optimization techniques. The aim of the study was to create an efficient small signal model (SSM) for efficient modeling of low-noise amplifiers and large signal modeling (LSM). An assessment of the efficacy of parameter extraction and the resultant values extracted for both substrates was conducted, while taking into account the cost-benefit analysis of the fabricated device. Si is a more widely available and established substrate, prompting a comparative investigation of functional parameters between SiC and Si based GaN HEMTs. The authors employed a more effective method of extracting intrinsic element values directly and optimizing extrinsic parameters, as opposed to directly extrapolating extrinsic elements, which may result in impracticality of the model. The study utilized global optimization method for parameter tuning, with validation of results through plotting of S-parameters fitting. The study demonstrated the effectiveness of this method.

In [19], the authors conducted comparative analysis between Neural Network-based modeling and parameter extraction methods. The authors were motivated by the ANN's ability to generalize and learn dynamic non-linear relationships quickly and in real-time. The study aimed to provide insights into accuracy, computational efficiency, model complexity, and model utility. The ANN model was developed using inputs such as temperature, gate to source voltage V_{gs}, frequency, drain to source voltage V_{ds}, and Scattering (S-parameters) as outputs. The capacity of both models to accurately reflect S-parameters was ascertained, but the ANN model demonstrated higher efficiency due to its consideration of deep intrinsic phenomena related with the device. Additionally, the ANN demonstrated better generalization capability and constructive generalizing ability, indicating its potential for practical applications.

In their study, [20] delved deeper into the high-periphery modelling with the maximum frequency of 26 GHz and 200°C temperature. They aimed to thoroughly investigate the behavior of the device under extreme operating conditions. The small-signal modeling approach was used to fully specify the parameters, thereby enabling a thorough examination, in accordance with the methodology outlined in reference [21]. Temperature was found to have a significant impact on the performance of the transistor due to the accumulation of enough electron-hole pairs, resulting in a gradual deterioration of the device's performance as temperature increased. Additionally, they mathematically and systematically examined the effects of thermal-based parameters.

In [22], the authors conducted a detailed study on microwave field effect transistors (FETs) as well as GaN HEMT, using an ANN approach. They included the same inputs as the previous studies, but also added the device geometry as an input. Separate models were developed for all eight S-parameters, which is represented by real and imaginary parts. Real and imaginary forms have the advantage of in the range of 0 to 1, which helps the training algorithms to find the optimal path while searching for minimums, except for S22, which is a magnitude parameter that extends beyond this range. To address this problem, the paper proposed a preprocessing method based on logarithms. The investigation revealed that the utilization of this modelling technique allows for the simulation of even the problematic kink effect of the S-parameters.

[23]-[24] carried out a more extensive investigation on the small-signal modelling of GaN HEMT using ANN-based modelling, with a particular focus on comparing the commonly used Multilayer Perceptron (MLP) architecture and various practiced ANN architectures for modeling. To this end, the authors introduced the Nonlinear Autoregressive with Exogenous Inputs (NARX) structure in both parallel and series-parallel architecture [25]. From a theoretical standpoint, the series-parallel configuration was found to have an advantage over the parallel configuration due to

its capability of feeding back the actual outputs to the input end, whereas the parallel architecture feeds back estimated outputs. Utilizing a combination of Mean Squared Error (MSE) analysis and parameter fitting techniques, the present study provides empirical evidence supporting the theoretical finding. Two different architectures were proposed and evaluated, with a detailed description of the experimental methodology and results. The authors have established, through rigorous empirical analysis that implementing the NARX architecture in a series-parallel configuration can lead to improved performance in the areas of generalization and simulation. The study contributes to the existing body of knowledge in the field, and offers potential avenues for further research. However, the main disadvantage of this approach is its complexity due to the increased number of hardware, such as delay elements, which reduces computational efficiency.

The limitation identified in reference [25] motivated the authors to seek alternative architectures. Consequently, they proposed a novel Cascade MLP architecture for small-signal GaN HEMT simulation, which can effectively predict s-parameters. In the study [26], the authors investigated various preprocessing techniques, initialization methods, training algorithms, and compared the performance of Cascade MLP and simple MLP in terms of % MRE, fitting curves, and MSE for all four S-parameters. The authors also thoroughly analyzed the bias dependencies and frequency of S-parameter reflecting the output of GaN HEMT device. Based on their findings, the authors concluded that the Cascade MLP architecture outperforms the MLP architecture slightly due to its greater number of connections.

In [27], the authors extended their research on small-signal modelling by expanding the temperature range under consideration. In contrast to the preceding study's characterization of temperature variation as falling between 40°C and 80°C, the present authors have broadened their examination to encompass a more expansive thermal gradient extending from 35°C to 200°C. This

broadened the scope of their investigation, enabling a more comprehensive analysis of the device at higher temperatures. The authors employed the same ANN-MLP architecture to build their model. Their findings indicate that the ANN is capable of simulating device operation over a wide range of frequencies and temperatures.

In [28]-[29], the authors expanded upon previous research by developing an ANN-based learning technique for both SSM and LLM of active devices. A thorough investigation of the devices' dependency on frequency, temperature, and bias was carried out by the researchers, who subsequently suggested a feedback-based ANN structure. This architecture floods neurons with additional information to enhance recognition of the relationship between datasets. Unlike prior works, the authors employed s-parameters represented in magnitude and phase, rather than real and imaginary. A comprehensive evaluation of fitting parameters was undertaken in the study, and the authors generated s-parameter plots across multiple temperatures and bias conditions. To sum up, the authors verified the accuracy of their results through the utilization of a CAD tool. Additionally, their study was expanded in [30].

The authors introduced an electro-thermal modelling methodology that employed the use of ANN techniques [31]. Previous research highlighted the degradation of device performance [27] at higher temperatures giving rise to self-heating effects and trapping impacts. Self-heating occurs due to power dissipation, which results in reduced electron speed and mobility, subsequently lowering the speed of the drain current and diminishing device output power, gain and efficiency. Trapping effects, including surface and buffer trapping, further impede device performance, with surface trapping which is the most concerning challenge. The authors generated multiple models to replicate diverse effects by breaking down device nonlinearities into partial nonlinearities. An initial model was put forth for the isothermal case, in which the device was operated in an un-

biased state to prevent the occurrence of trapping. Next, the authors incorporated the effect of temperature and proposed a genetic algorithm-assisted ANN model to check temperature dependencies. The ultimate stage involved the implementation of a model utilizing CAD.

All of the ANN models mentioned above primarily utilized BP based algorithms that can lead to less effective and unique final solutions because the algorithm randomly generates initial weights and biases during each updating process. To address this problem, several solutions could be implemented. One potential solution is to integrate global optimization techniques into the standard BP-based algorithms. For example, Jarndal in [32] investigated various global optimization techniques, such as particle swarm optimization (PSO), genetic algorithms (GA), and grey wolf optimization (GWO), in combination with ANN. The study evaluated the performance of each model, with GA providing more diverse search space for better initializations, PSO providing better computational efficiency, and GWO showing overall performance comparable to a combination of both techniques. The authors also conducted parameter tuning of both ANN optimization techniques to identify the optimal behavioral models. Ultimately, the study concluded by and global presenting the fitting results and improved initializations achieved by each model.

In the field of power amplifier fabrication, Silicon is considered the most favorable substrate material due to its low-cost of fabrication. However, researchers are exploring alternative substrates for the fabrication of GaN HEMT devices. In a recent study by Jarndal [33], three different substrate used for GaN HEMT devices were investigated such as Silicon (Si), Silicon Carbide (SiC), and Diamond. The study focused on the effects of threading dislocations resulting from using substrates other than the original material, and analyzed the changes in thermal and electrical performance of the devices. The development of TCAD-based models enabled the researchers to devise strategies for the efficient implementation of these devices in terms of design.

In comparison to the other materials assessed in the study, Diamond demonstrated superior thermal conductivity, as concluded by the researchers [15]-[30]. This property can help reduce self-heating caused by power dissipation at higher temperatures. The researchers also discovered that the collapsing of current is minimal for GaN HEMTs on SiC substrate in contrast to the other devices with different substrates. However, trapping effects were found to significantly impact the electro-thermal behavior of these devices. To mitigate the dislocation of elements and augment the device's performance, the researchers propose the necessity of optimizing GaN on Diamond substrate.

The literature review revealed that most of the papers in this field relied on Artificial Neural Network (ANN) based approaches for the behavioral modeling specifically for GaN HEMTS. However, one of the issues with this approach is that it can suffer from the problem of converging at local minimums. Some works have suggested the implementation of globally optimized augmentation techniques to address this issue, while alternative methods have sought to use different activation functions and preprocessing steps for dataset. Saddam in [34] suggested to use Support Vector Regression (SVR) to model a GaN HEMT device on a Silicon Carbide (SiC) substrate. The Structural Risk Minimization (SRM) principle, embedded within the Support Vector Regression (SVR) algorithm, facilitates global minimum attainment by leveraging supervised learning techniques, unlike traditional SRM techniques in ANN-based approaches. In the paper, the authors analyzed temperature, frequency, V_{DS}, V_{GS}, and the device's geometry as inputs, while the S-parameters' magnitude and phase were the outputs. Before training the model, they used highly efficient outlier detection based preprocessing steps. Additionally, they provided a comprehensive analysis of noise detection in the paper. Additionally, the authors conducted a thorough analysis of noise detection in their paper. The optimization of model parameters for enhanced performance necessitates parameter tuning. In this regard, the authors have leveraged the

global Particle Swarm Optimization (PSO) technique to fine-tune the hyperparameters. Finally, the model was deployed into a Computer-Aided Design (CAD) tool to produce a power amplifier prototype. However, the computational demands and dataset magnitude associated with this modeling approach pose a notable challenge. In situations where the dataset size exceeds a certain threshold, the model's effectiveness may be diminished, leading to suboptimal performance or elongated training time. Non-parametric approaches can be investigated to address this issue.

In [35], the authors explored General Regression Neural Network (GRNN) and Multi-Layer Perceptron Neural Network (MLPNN) methods to check the linear behavior of transistor. It was found that MLPNN showed significantly improved results due to complex architecture as compared to GRNN but took more time to train and simulate. They used very small dataset of 150 samples for S-parameters and 75 samples for N parameters, which led to restricted capacity of handling novel inputs for ML based modeling of transistors.

These techniques exhibit promising results in emulating complex behaviors, providing better prediction capabilities, and good computational efficiency. On the other hand, they require large sample sizes of measurements. Despite these advantages, there are still unresolved mysteries about these approaches due to the lack of exploration in several aspects. Generally, there is no single Machine Learning (ML) based modeling method which can be considered ideal and excellent choice for a wide variety of applications. The effectiveness of ML algorithms relies on various factors such as the specific problem, data quality, feature extraction, data nature, input and output distributions, number of samples, data division, preprocessing techniques, performance metrics, error functions, parameter tuning, and algorithms to tune the parameters, and type of ML algorithm. The final design application exerts a significant influence on the selection of machine learning algorithms for constructing the model. For example, some applications may require a

trade-off between accuracy, and computational efficiency, and thus the selection of suitable ML algorithms must be made in accordance with the requirements. Moreover, when a distinctive model is necessitated by the intended application, ML algorithms that generate exclusive solutions take precedence.

While various modelling techniques are available for GaN HEMTs, each of these methods is typically suitable for specific operational parameters of the device and may perform poorly under different conditions. Unfortunately, the generalization potential of machine learning (ML) based modeling for GaN HEMTs remains largely unexplored due to limited information. This lack of information often leaves researchers unaware of the potential of ML-based modelling techniques. Keeping all the drawbacks and research gap in mind, this research focuses on global optimization based ANN to explore the behavioral modelling of GaN HEMTs. The inputs to the ML models are frequency, biasing of GaN HEMT, temperature while the outputs are four Scattering parameters with real and imaginary parts; S_{11},$S_{12,}$$S_{21}$ and S_{22}.

1.3. Aims and Objectives

The design of an effective machine learning model for advance RF devices is the primary goal of this research. This research demonstrates that which global optimization algorithm performs better for modelling GaN HEMTs. The modelling is done in MATLAB. The problem of MLP getting stuck in local minima is solved by applying global optimization i.e. grey wolf optimization-particle swarm optimization (GWO-PSO), ant colony optimization (ACO), whale optimization algorithm (WOA), and ant lion optimization (ALO). The weights and biases are optimized. The small signal based models exploited S-Parameters dependency on the frequency, temperature, and biasing. Furthermore, a detailed comparison and analysis is performed using coefficient of determination (R^2), relative mean square error, mean squared error (MSE), mean

absolute error (MAE), simulation, and hyperparameters optimization time. Since there are many global optimization algorithms, finding the right global technique for the given problem is crucial. Therefore, a comparative analysis is paramount and necessary. So, a thorough study of global optimization assisted ML algorithms and compare them in terms of complexity, computational efficiency, simulation speed and CAD compatibility for the modelling of high-frequency devices, is presented. To enhance the credibility of the models, simulation curves also provided. In a nutshell, the aims can be stated as follows:

- Evaluate ML methods for constructing small-signal models
- To develop Global optimization assisted ANN models
- To find the optimal architecture of all the developed models
- To quantify model performance through comparison of R^2, MAE, MSE, and simulated curves
- Ultimately, analyze and examine the comprehensive comparison
- Select suitable global optimization for small signal modelling of GaN HEMTs

Chapter-2 Methods and Device Physic

2.1 Device Description

This study investigates a device comprising of 10 fingers, each with a gate width of 200 μm, resulting in a total 2 mm of gate width. The device is grown on Si substrates using the 0.5 μm NRF1 process. The fabrication and composition of the device is done through metalorganic chemical vapor deposition approach. To mitigate mismatch-induced defects and buffer trapping, an interlayer for nucleation is interposed between the substrate and the buffer. Optimization of the passivation technique has led to a reduction in surface trapping and improved RF characteristics of the device. Interconnection of the source pads is accomplished through the use of via-hole technology. The device performance is improved by implementing the source field plate technique and modeled using Angelov2 equivalent-circuit modelling approach [36]. Figure 2.1 displays a die photograph of the device, along with its general structure. To evaluate the performance of the device, the real and imaginary parts of its S-parameters are measured through the utilization of a vector network analyzer (VNA). S-parameter measurements at varying external temperatures are obtained by fixing the device onto a thermal chuck with temperature control. When under active bias, the temperature inside the device is determined by adding the temperature of the thermal chuck to the self-heating temperature of the device. Number of measurements are taken under the different biasing conditions, with the temperature varied in increments of 25°C from 25°C to 175°C. Measurements are carried out within the 100 MHz to 26 GHz frequency range.

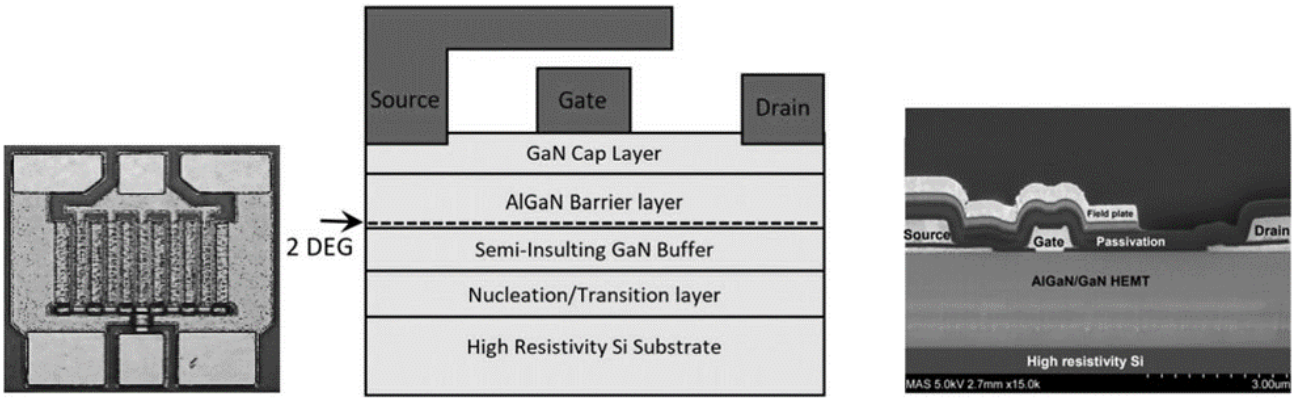

Figure 2.1: A photograph of the 2-mm (10 × 200 µm) AlGaN/GaN on Si substrate, along with its crystalline layer structure. Additionally, an implanted source field plated technology cross-sectional scanning electron microscope (SEM) image is also provided [36].

2.2 Multi-Layer Perceptron (MLP) Model

The feed-forward artificial neural network known as MLP establishes input-output relationships through dataset processing. Three layers make up the MLP model: input, hidden, and output. The input layer has the same number of neurons as the features present in the dataset. Depending on the output features, the number of neurons in the output layer can be single or multiple. Meanwhile, the hidden layer can include several sub-layers with varying neuron number. Control parameters such as neurons and hidden layers, are adjusted by the user to suit the problem type. Figure 2.2 illustrates the MLP overall structure.

In the MLP, the interconnection between each neuron and neurons in the subsequent layer is established through weight values. To obtain the value of a neuron, the weights of the connected neurons from the preceding layers are multiplied by their output values, resulting in a NET sum [37]. This sum is then fed to an activation function f (NET) to determine the neuron's output value. Figure 2.3 demonstrates the procedure for calculating neuron output values.

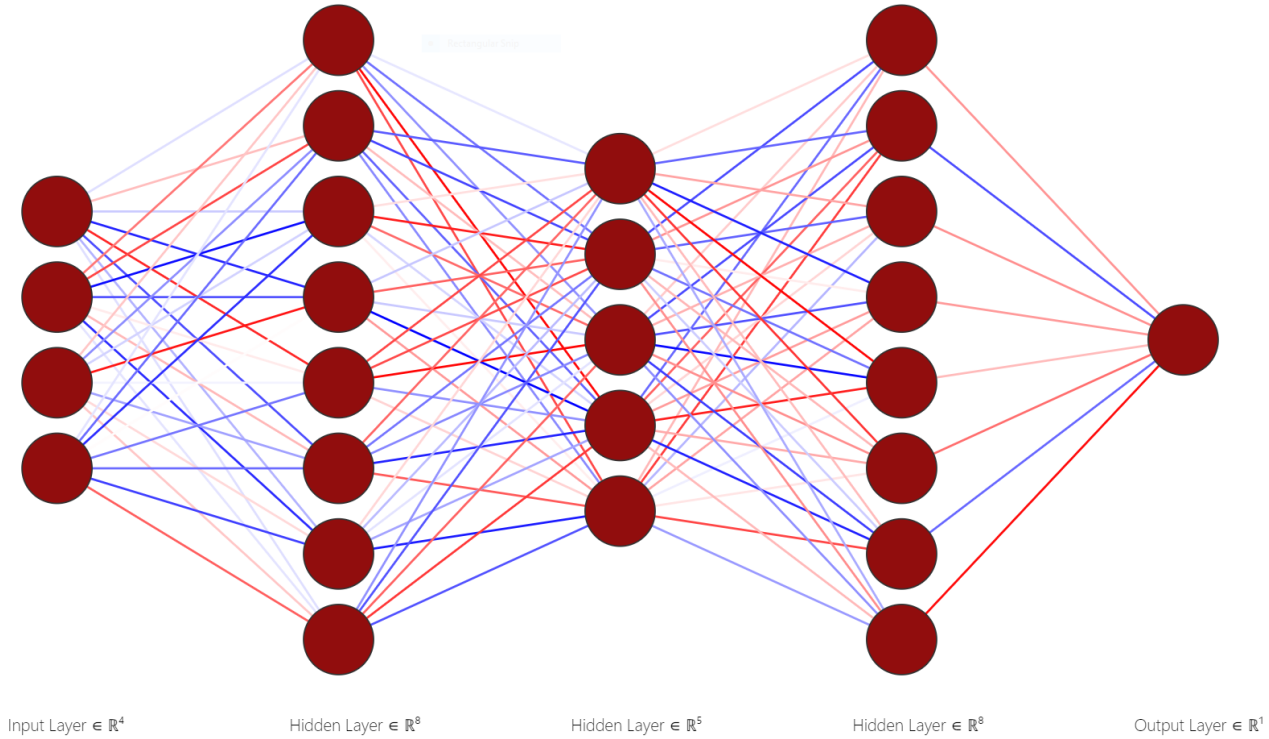

Figure 2.2: Generalized Architecture of MLP

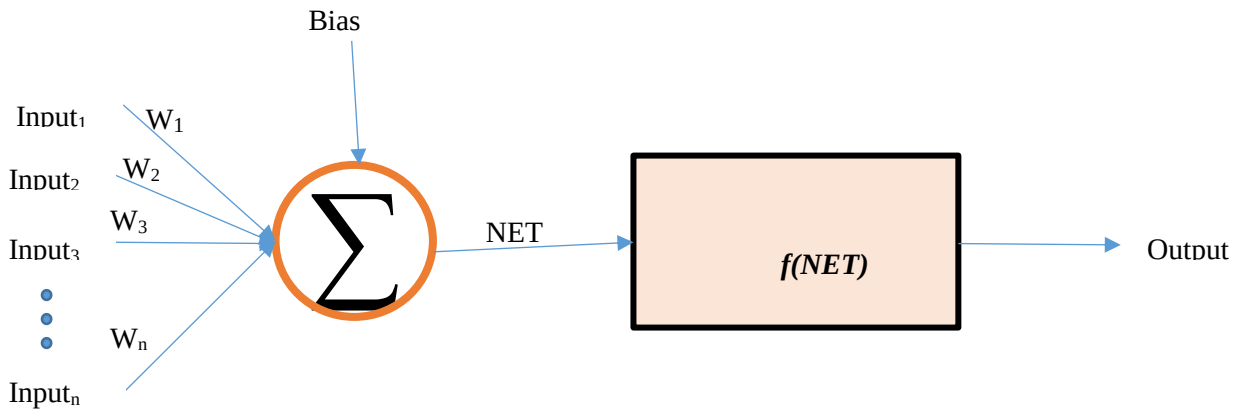

Figure 2.3: Derivation of output from inputs considering weights and bias

In the f(NET) system, activation functions are employed to cover a wide range of operations, including the linear activation function (purelin), the log-sigmoid activation function (logsig), and the hyperbolic tangent sigmoid activation function (tansig), with the selection of the function dependent on the specific problem. In linear regression scenarios, linear activation

functions are typically the preferred choice, whereas nonlinear activation functions are better suited for nonlinear regression tasks. Despite this, specific nonlinear regression problems may still demand the application of linear activation functions.

Ertuğrul acknowledged the challenge in choosing the ideal activation function, asserting that no analytical technique could ascertain it. Consequently, Ertuğrul proposed an innovative method for defining activation functions, capable of producing linear activation functions for nonlinear issues, thus enabling their use in nonlinear problems as needed [38].

The performance of MLP can be improved by utilizing distinct activation functions in separate layers. Therefore, a comprehensive system that takes all possibilities into account is essential to achieve favorable outcomes. Activation functions play a crucial role in adjusting input values to fit within a specific range. The most prevalent activation functions are illustrated in Figure 2.4.

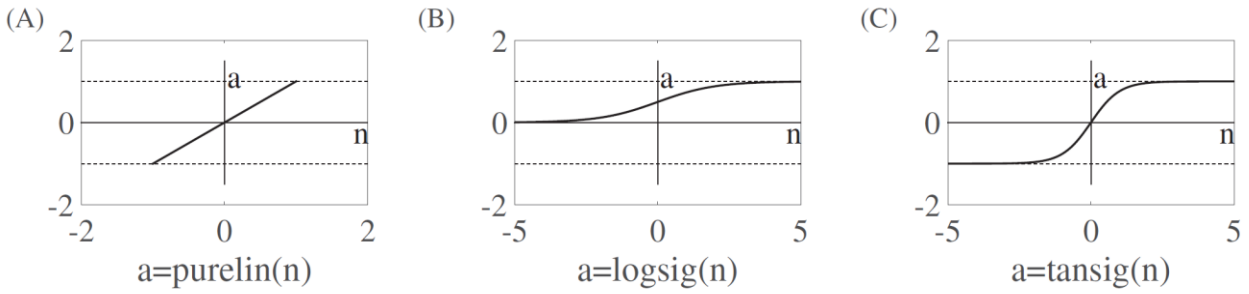

Figure 2.4: Various Activation Functions

The completion of the feed-forward stage in MLP is demonstrated in Figure 2.2, whereby the output values of all neurons in each layer are calculated, leading to the final layer. Subsequently, network performance is evaluated by computing the deviation between the predicted and actual output utilizing the MSE technique, as illustrated in Equation (2.1).

$$e_i = target_i - output_i \quad (2.1)$$

Consider e_i to be the error value pertaining to the ith neuron situated in the output layer, Equation (2.2) provides the MSE value of the ith neuron.

$$\varepsilon_i = \frac{1}{2} e^2{}_i \quad (2.2)$$

To determine the total Mean Squared Error (MSE) value of the network, one must aggregate the individual MSE values of all neurons present in the output layer, as demonstrated in equation (2.3).

$$E = \frac{1}{2} \sum_{i=1}^{n} e^2{}_i \quad (2.3)$$

Following the computation of total ε, where n represents neurons present in the output layer, the back propagation step is initiated to determine the optimal MLP structure, weights are chosen to minimize the output error effectively. The weight updating process involves adjusting connection weights of each neuron using the Gradient Descent (GD) algorithm, which utilizes the first derivative to locate local minima.

Equation (2.4) shows the ith value of neuron associated with the output layer.

$$output_i = f\left(\sum_j^m w_{ij} * output_j + bias\right) \quad (2.4)$$

The weight value, w_{ij}, corresponds to the connection between the ith neuron in the output layer and the jth neuron in the preceding layer, where m signifies the neurons associated with the ith neuron in the pre-output layer. Designate the output value for the jth neuron in the layer prior to the output layer as $output_j$. Equation (2.5) outlines the relationship between the weight value of a relevant neuron and the total error.

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial e_i} * \frac{\partial e_i}{\partial output_i} * \frac{\partial output_i}{\partial w_{ij}} \quad (2.5)$$

The chain rule is applied to calculate the derivative of Equation (2.5). Once $\frac{\partial E}{\partial w_{ij}}$ is computed, the final operation involves weight updating. The weight update procedure for w_{ij} is executed by employing the expression outlined in Equation (2.6).

$$w_{ij}^{new} = w_{ij} - \alpha * \frac{\partial E}{\partial w_{ij}} \quad (2.6)$$

Where, α denotes the learning coefficient. It is essential to adjust the user-defined parameter α in accordance with the characteristics of the given problem. Equations (2.1)-(2.6) are executed for in the completion of one iteration after updating all weights. The training process the total neurons present in the output layer, resulting continues until the error is optimized and comparable with the user-defined threshold or the specified number of iterations is reached.

2.3 Ant Lion Optimization (ALO)

Seyedali Mirjalili introduced the ALO meta-heuristic optimization algorithm, modeled after the hunting methods of ant lion larvae [39]. The larvae excavate a cone-shaped depression in the sand by rotating their mandibles in a circular pattern, concealing themselves at the base of the pit. Their trap's steep walls make it simple for prey to fall in, while sand is utilized to cause the prey's feet to slip, resulting in the prey being ensnared at the bottom of the depression.

The stochastic movement pattern of the ants serve as the initial point for the ALO algorithm, with Equations (2.7) and (2.8) providing the mathematical representation.

$$Y(x) = [0, c(2r(x_1) - 1), c(2r(x_2) - 1) \dots c(2r(x_n) - 1)] \quad (2.7)$$

$$r(x) = \begin{cases} 1, & rand(0,1) > 0.5 \\ 0, & rand(0,1) \leq 0.5 \end{cases} \quad (2.8)$$

To prevent the ants' random walks from going beyond the limits of specified search space, Equation (2.7) is employed, with function *c* representing the total sum, and n representing the iterations, *x* representing the incremental advancements of random walks, and $r(x)$ representing the random walks.

$$Y_j^x = \frac{(Y_j^x - a_j) \times (d_j^x - c_j^x)}{(b_j - a_j)} + c_j^x \quad (2.9)$$

Where, a_j and b_j are employed to indicate the minimum and maximum values corresponding to the Y_j^x position vector, respectively, while, c_j^x and d_j^x serve as representations of the minimum and maximum values of the jth dimension during the xth iteration, respectively in equation (2.9).

Figure 2.5: The foraging behavior of an ant lion

Under the influence of the ant lion, trapped ants are propelled towards the bottom of the pit, as demonstrated by Equations (2.10)-(2.13), which mathematically model the ant lion's behavior of sliding ants to the pit bottom at a particular sliding velocity.

$$c_j^x = Antlion^x + c^t \tag{2.10}$$

$$d_j^x = Antlion^x + d^t \tag{2.11}$$

$$c^x = \frac{c^x}{Z} \tag{2.12}$$

$$d^x = \frac{d^x}{Z} \tag{2.13}$$

$Antlion^x$ is used to represent the position of the selected ant lion during the *x*th iteration, with Equation (28) providing a means of determining the sliding ratio, represented by *Z*.

$$Z = \begin{cases} 1 + 10^2 \frac{x}{X_{max}}, & 0.1X_{max} < x < 0.5X_{max} \\ 1 + 10^3 \frac{x}{X_{max}}, & 0.5X_{max} < x < 0.75X_{max} \\ 1 + 10^4 \frac{x}{X_{max}}, & 0.75X_{max} < x < 0.9X_{max} \\ 1 + 10^5 \frac{x}{X_{max}}, & 0.9X_{max} < x < 0.95X_{max} \\ 1 + 10^6 \frac{x}{X_{max}}, & 0.95 < x < X_{max} \\ 1, & otherwise \end{cases} \tag{2.14}$$

The ants' population position is ascertained using Equation (2.15), while Equation (2.16) prescribes how the ant lion modifies its position after consuming Ant_j.

$$Ant_j^x = \frac{R_A + R_E}{2} \tag{2.15}$$

$$Antlion^x = \begin{cases} Ant_j^x, & f\left(Ant_j^x\right) < f(Antlion^x) \\ Antlion^x, & otherwise \end{cases} \tag{2.16}$$

2.4 Whale optimization algorithm (WOA)

The WOA, a meta-heuristic optimization algorithm, is modeled after the effective hunting behavior of humpback whales. The use of a bubble-net is a clever and successful technique that enables the whales to trap and catch their prey while remaining hidden. Figure 2.6 shows the visual representation of the hunting behavior of humpback whales, which is a testament to the efficacy of the WOA algorithm [40].

To optimize the search for the best solution, the WOA algorithm utilizes three stages - encircling, bubble-net attacking, and searching for prey.

2.4.1 Encircling prey

The encircling behavior of humpback whales is a highly effective technique that has inspired the WOA algorithm. The algorithm uses the fitness value of the fittest whale to update the positions of other whales during each iteration, as mathematically expressed in equations (2.17) and (2.18).

$$\vec{D} = \left|\vec{C}.\vec{X}^{*}(x) - \vec{X}(x)\right| \quad (2.17)$$

$$\vec{X}(x+1) = \vec{X}^{*}(x) - \vec{A}.\vec{D} \quad (2.18)$$

To execute the WOA algorithm effectively, it is essential to comprehend the role of variables such as x ,$\vec{X}^{*}$, $\vec{A}$, and C. The position vector of the best whale is represented by $\vec{X}^{*}$, and $\vec{A}$ and $\vec{C}$ serve as coefficient vectors that can be calculated using equations (2.19) and (2.20) during each iteration.

$$\vec{A} = 2.\vec{a}.\vec{r} - \vec{a} \quad (2.19)$$

$$\vec{C} = 2.\vec{r} \tag{2.20}$$

In this context, $\vec{r}$ represents a random vector, and the coefficient $\vec{a}$ gradually decreases from 2 to 0 over the course of iterations.

2.4.2 Bubble-net predation technique

To optimize the search for solutions, the WOA algorithm utilizes two whale attack methods, the Spiral updating position and shrinking encircling approach. The Shrinking encircling technique is attained by reducing the $\vec{a}$ value in the expression (2.21), while Equation (2.22) represents the process of Spiral updating position.

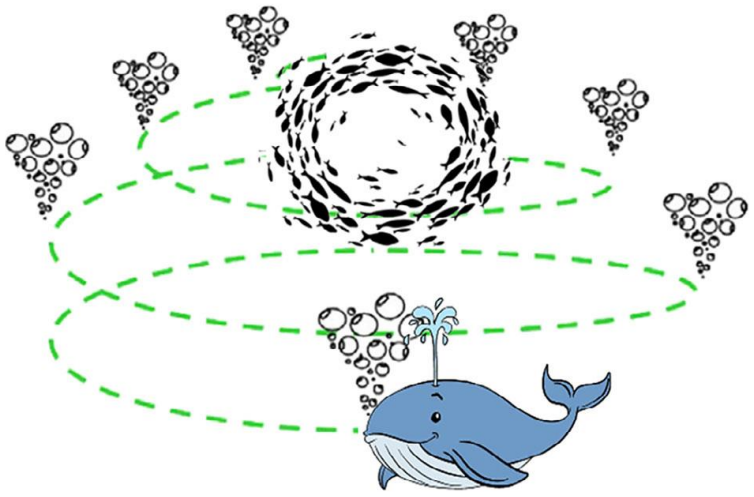

Figure 2.6: Predatory patterns exhibited by humpback whales

$$\vec{X}(x+1) = \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{X^*}(x) \tag{2.21}$$

Where, b acts as the constant component throughout the process of generating the spiral form, $\overrightarrow{D'} = \left|\overrightarrow{X^*} - \vec{X}(x)\right|$, and l are random numbers ranging from [-1, 1]. The selection process for either the Shrinking Encircling Mechanism or the Spiral Updating Position methods is random,

with a 0.5 probability assigned to each technique. The updated positions of the whales are computed utilizing Equation (16).

$$\vec{X}(x+1) = \begin{cases} \vec{X}^*(x) - \vec{A}.\vec{D}, & p < 0.5 \\ \overrightarrow{D'}.e^{bl}.cos(2\pi l) + \overrightarrow{X^*}(x), & p \geq 0.5 \end{cases} \quad (2.22)$$

Where, p ranges from 0 to 1.

2.5 GWO-PSO

GWO-PSO is a hybrid optimization algorithm that combines the Gray Wolf Optimization (GWO) and Particle Swarm Optimization (PSO) algorithms. GWO is a nature-inspired optimization algorithm that imitates the hunting behavior of gray wolves in the wild life [41]. The algorithm works by dividing the search space into a set of territories that are explored by the wolf pack. In the pack, every wolf corresponds to a possible solution for the optimization problem, and the pack as a whole collaborates to find the optimal solution.

Imitating the food-searching behavior of bird flocks, PSO serves as another nature-based optimization algorithm. Each potential solution in PSO is depicted by a particle that explores the search space with the aim of discovering the optimal solution. The particles communicate with each other and adjust their movement relying on the optimal solutions identified thus far.

The GWO-PSO algorithm combines the advantages of both GWO and PSO. The algorithm starts with the GWO phase, where the wolf pack explores the search space and finds a set of potential solutions. These solutions are then passed on to the PSO phase, where particles are used to further

refine the search and find the optimal solution. This algorithm is founded on a hybridization approach that leverages the exploration capabilities of the GWO algorithm and the exploitation capabilities of the PSO algorithm, thus exploiting the advantages of both approaches. To accomplish this, the optimal positions of the top three search agents (α, β, and δ) are updated within the search space using the equations detailed in (2.23). Specifically, an inertia constant (w) modulates the exploitation and exploration activities of the wolves within the specific search space, as depicted by the following equations:

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1} * \overrightarrow{X_\alpha} - w * \vec{X}\right|, \overrightarrow{D_\beta} = \left|\overrightarrow{C_2} * \overrightarrow{X_\beta} - w * \vec{X}\right|,$$

$$\overrightarrow{D_\delta} = \left|\overrightarrow{C_3} * \overrightarrow{X_\delta} - w * \vec{X}\right| \quad (2.23)$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1} * \overrightarrow{D_\alpha}, \overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2} * \overrightarrow{D_\beta}, \overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3} * \overrightarrow{D_\delta} \quad (2.24)$$

Utilizing the aforementioned equations, the PSO and GWO variants are integrated by updating the position and velocity equations as given below:

$$v_j^{i+1} = w * \left(v_j^i + c_1 r_1 \left(X_1 - x_j^i\right) + c_2 r_2 \left(X_2 - x_j^i\right) + c_3 r_3 \left(X_3 - x_j^i\right)\right)$$

(2.25)

$$x_j^{i+1} = x_j^i + v_j^{i+1} \quad (2.26)$$

2.6 Ant Colony Optimization (ACO)

In [42], Ant Colony Optimization (ACO) has been utilized as a multi-agent approach to handle optimization problems, notably the Travelling Salesperson Problem (TSP). This algorithm emulates the behavior of natural ants searching for food sources. ACO relies on a network of

artificial ants that collaborate to identify optimal solutions via the indirect information exchange using ants' artificial pheromone. ACO algorithm executes parallel exploration through several computational threads, drawing on local problem data and including a dynamic memory structure that captures the problem's characteristics.

In the context of Ant Colony Optimization (ACO), pheromone serves as a critical component for guiding artificial ants in building new solutions. Throughout the search procedure, a learning mechanism accumulates artificial pheromone at runtime, rewarding high-quality solutions. Paths with more pheromone are more likely to be followed by the ants. Additionally, an evaporation rule is tied to the pheromone, which reduces the probability of poor quality solutions. The strength of the ACO algorithm relative to other algorithms stems primarily from the quality of the solutions generated by the colonies [43].

Given that q0 ∈ (0, 1], the path with the highest pheromone concentration and the shortest distance is selected with a probability of q0. On the other hand, the path with a probability of 1 - q0 is presented in Equation (2.27):

$$P_{ij}^{k} = \begin{cases} \dfrac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{m \in N_i^k}(\tau_{im})^{\alpha}(\eta_{im})^{\beta}}, & j \in N_i^k \\ 0 & j \notin N_i^k \end{cases} \tag{2.27}$$

The ACO algorithm uses the pheromone value τ_{ij} associated with a component, as well as the heuristic information value η_{ij} provided by the weighting function, to calculate the probability of selecting a particular path. The positive factors alpha and beta are used to assign weight to the pheromone and heuristic information, respectively. If alpha is set to 0, the algorithm favors

selecting closer cities, effectively converting ACO into a stochastic search algorithm. On the other hand, if beta is set to 0, only the pheromone is used, leading to faster convergence. Achieving an optimal balance between pheromone and heuristic information is essential, utilizing alpha and beta parameters to avoid getting trapped in local minima, where all ants converge in an inappropriate direction.

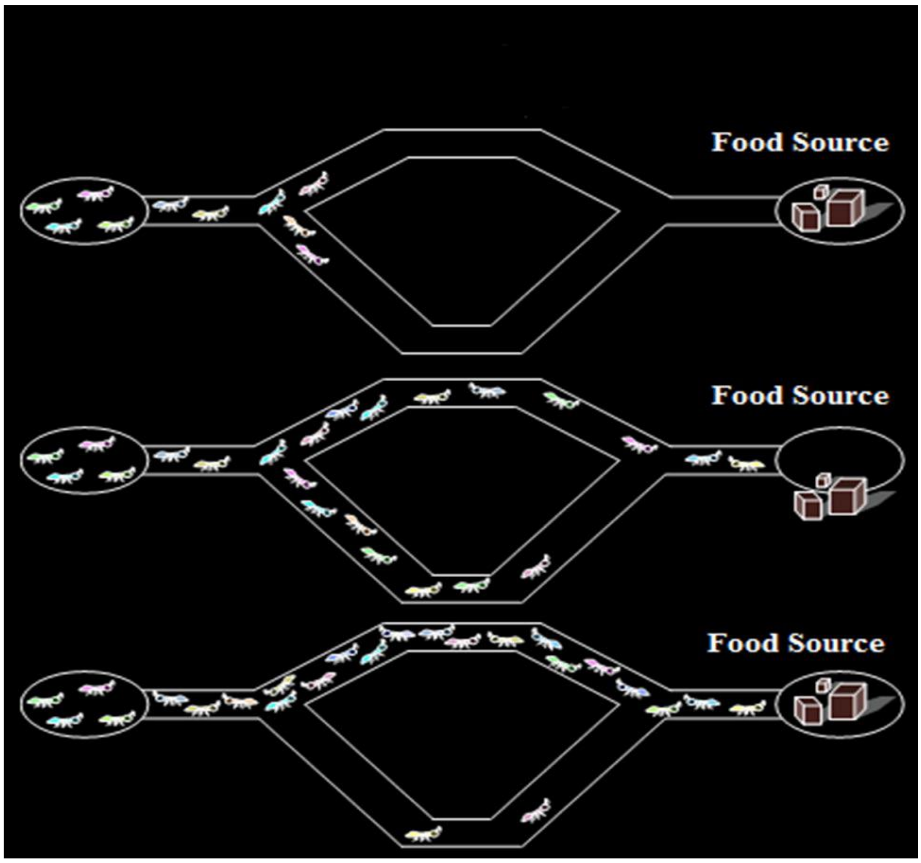

Figure 2.7: collective foraging behavior

Chapter-3 Model Validation and Discussion

In this section, a comprehensive validation and evaluation of the models examined in this study are provided. Notably, MATLAB serves as the primary tool for model development, validation, and evaluation.

3.1 Dataset

In machine learning based modelling, a dataset is a collection of data points or records used to train, and test machine learning models. Datasets consist of features (independent variables) and target variables (dependent variables). Machine learning algorithms use datasets to learn patterns, relationships, or structures within the data and make predictions based on the input features.

It should be noted that the acquired measurement data originates from a gallium nitride (GaN) high-electron-mobility transistor (HEMT) device featuring a 10x2 mm geometry, as illustrated in Figure 3.1. Employing the conventional biasing methodology delineated in [44], the device is biased accordingly. Subsequently, its corresponding response under varying bias conditions is documented as scattering parameters (S-parameters), incorporating both real and imaginary components, through the utilization of a vector network analyser (VNA). The scattering parameters (S-parameters) measurements involve the application of gate and drain bias voltages, which are crucial in characterizing the performance of the GaN HEMT device under various operational conditions. These bias voltages, specifically gate-source voltage (V_{GS}) and drain-source voltage (V_{DS}), influence the device's electrical characteristics and contribute to the overall analysis of the device's behaviour within the tested frequency range. Where, gate-to-source voltage (V_{GS}) is ranging from -2 V to 2 V with the step sizes of 0.1 V or 0.2 V. The voltage ranges for the drain-to-source voltage (V_{DS}) are 7 V, 28 V, and 48 V, and each voltage range is swept using a different

step size for V_{GS}. Specifically, when V_{DS} is 7 V, the step size for V_{GS} is 0.2 V, while for V_{DS} equal to 28 V and 48 V, the step size for V_{GS} is 0.1 V as depicted in the Table 3.1. The comprehensive dataset encompasses measurements taken at seven distinct temperatures, ranging from 25°C to 175°C in increments of 25°C. Characterization of the device is conducted over a frequency (f) range of 0.1 to 26 GHz. Comprising 103,057 samples, the dataset incorporates four predictor variables: gate-source voltage (V_{GS}), drain-source voltage (V_{DS}), frequency (f), and temperature. Furthermore, eight predicted variables are included, representing the real and imaginary components of the scattering parameters S_{11}, S_{21}, S_{12}, and S_{22}.

Table 3.1: Gate and Drain Voltages

S.No	Range of Vgs (V)	Step Size of Vgs	Vds (V)
1	[-2,2]	0.2	7
2	[-2,-0.4]	0.1	28
3	[-2,-0.8]	0.1	48

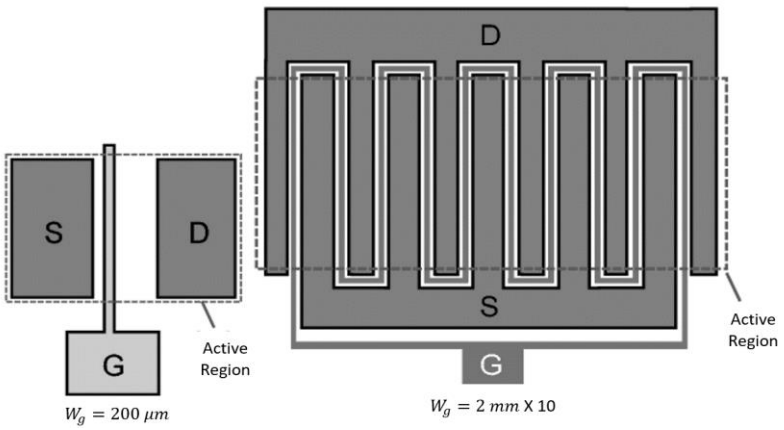

Figure 3.1: Layout of 2 mm ×10 GaN HEMT Device

3.2 Pre-processing of the Dataset

Pre-processing is an indispensable element in data science operations, often serving as a reliable predictor of model effectiveness [45]. Thorough data pre-processing and well-crafted feature engineering can significantly elevate model performance. The data utilized in this research is based on a Silicon substrate, with inputs, V_{GS}, V_{DS}, temperature (T), and a frequency domain spanning 0.1 to 26 GHz. The dataset includes 103,057 data examples.

Understanding the dataset distribution is crucial. It is noticeable that the ranges of most parameters differ. Additionally, output values in some samples are discontinuous and do not align with the overall distribution of the respective S-parameters. The presence of outliers and uneven peak values is also apparent, which can be ascribed to measurement anomalies. One significant issue with having distinct ranges for predictors and predicted variables is the potential distortion or skewing of the error function's contour, which can result in an elliptical shape. This can cause incorrect outcomes, as the algorithm may struggle to identify the global minimum and instead converge at local minima in the worst cases.

Furthermore, outliers directly influence the decision boundary's positioning, which can lead to a higher error rate on the testing set. Different algorithms respond differently to datasets containing outliers. For example, the mean absolute error function is less affected by outliers compared to the mean squared error function. Thus, it is vital to eliminate outliers and smooth out peak values as much as possible. However, removing some outliers may not be feasible, as it could lead to information loss.

The subsequent section details further pre-processing stages employed in this study:

- Identifying and handling missing data points

- Partitioning the dataset into train and test subsets
- Implementing feature Scaling

3.2.1 Identifying and Handling Missing Data Points

If the dataset is populated with numerous invalid (NaN) and erroneous values, the model is likely to give error. Therefore, addressing such missing data is crucial. A straightforward MATLAB command was employed to check missing data.

3.2.2 Dividing the dataset into train and test subsets

One of the most critical aspects of machine learning tasks is data division, which often necessitates trial-and-error techniques. The model's optimal performance hinges on the optimal data segmentation. In this research, temperature is employed as the dividing criterion. Given that the dataset includes seven different temperatures, five are randomly assigned to the training set and two to the testing set. The allocation of temperatures to each set is based on a trial-and-error method, ultimately leading to the best distribution of training set (25°C, 50°C, 75°C, 100°C, and 150°C) and testing set (125°C and 175°C).

3.2.3 Implementing feature Scaling

The standardization of a range of diverse independent data elements is achieved through feature scaling [46]. Widely known as data normalization in data processing, this method is often utilized during the data preparation process. Standardization/normalization is consistently founded on a core principle. Variables evaluated on various scales might not contribute equally to the learning function and model fitting, giving rise to bias. The proposed model employs Min-Max Scaling normalization to mitigate these concerns. The Min-Max scaler can be employed to

standardize input features. Consequently, elements are transformed into the range [0, 1], signifying that the minimum and maximum values of the variable will be 0 and 1, respectively.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3.1)$$

All preprocessing steps are illustrated in the flowchart presented in Figure 3.2.

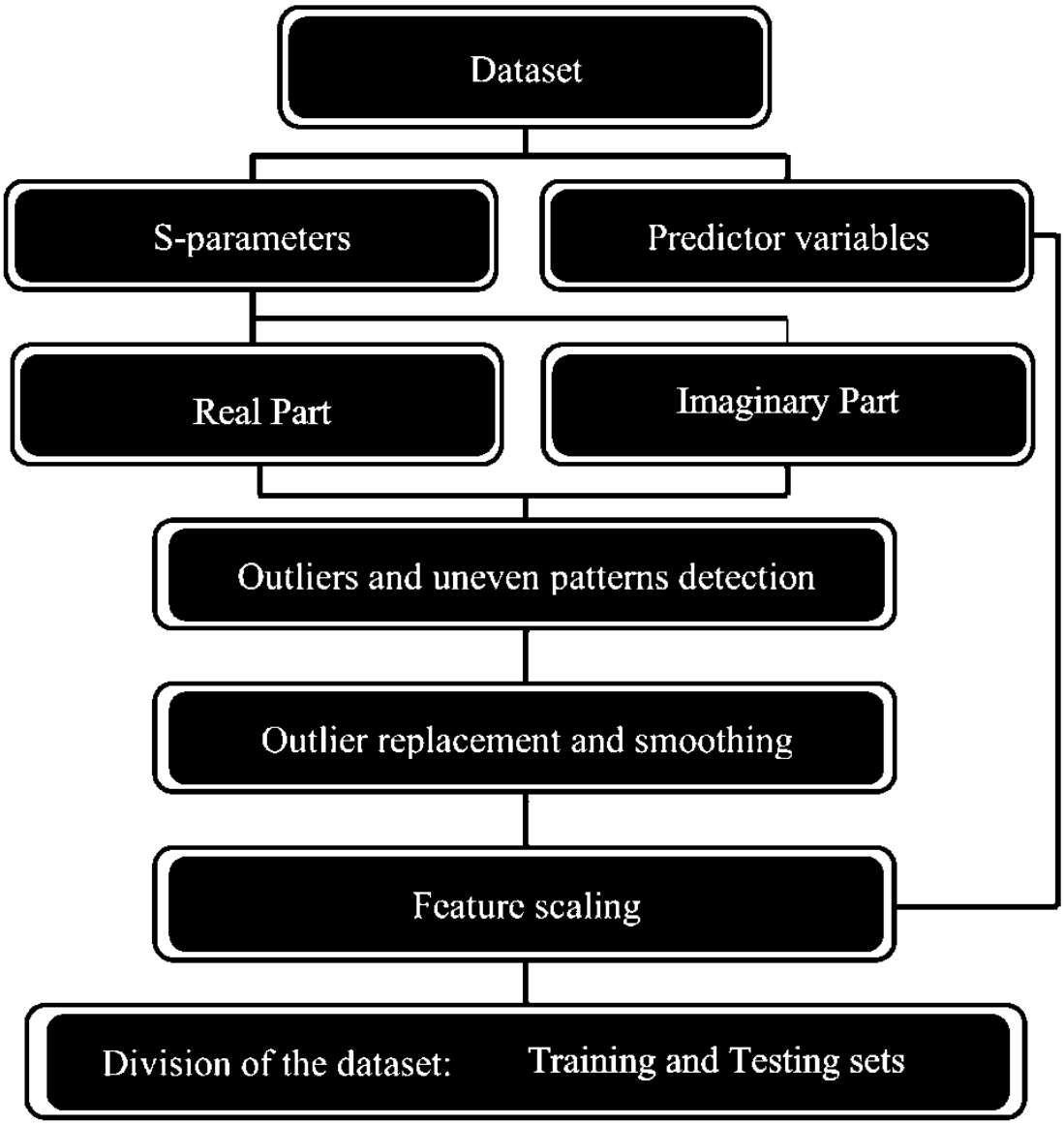

Figure 3.2: Pre-processing Steps for Training the Model

3.3 Validation of ANN

The development of ANN models in this work employs MATLAB and the Levenberg-Marquardt (LM) backpropagation algorithm for training. Other backpropagation-based training approaches, including gradient descent, gradient descent with momentum, variable learning rate gradient descent, and BFGS Quasi-Newton, do not deliver satisfactory performance [47]. While both the scaled Conjugate Gradient and Resilient Backpropagation algorithms produce comparable results, the LM algorithm exhibits marginally superior performance across all metrics. Furthermore, although Bayesian regularization backpropagation achieves the highest accuracy, it demands a longer simulation time due to its embedded automatic regularization, facilitating the algorithm's operation in overfitting situations [48]. Therefore, the LM backpropagation algorithm is selected, considering the balance between simulation time and accuracy. The proposed ANN model is shown in the figure 3.3.

In this research, an iterative approach is employed to fine-tune the model size. Starting with a smaller model size, the efficiency, based on MSE, is assessed by comparing measured and simulated plots. The model size is gradually increased, and training is iteratively performed until an acceptable MSE is reached. The ideal configuration, as presented in this work, consists of a 4-10-20-10-1 model size (inputs, first hidden layer, second hidden layer, third hidden layer, and output, respectively). Furthermore, the *logsig* activation function is selected after examining a range of activation functions. For training dataset, the Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (%R^2) metrics are computed in the table 3.2. To ensure a succinct yet thorough analysis, only the generalization metrics (MSE, MAE, and R^2) for the independent and unseen data, which reflect the errors incurred on the testing set, are incorporated in Table 3.3 and

table 3.4. Importantly, the error profiles sufficiently illustrate the models' effectiveness and efficiency.

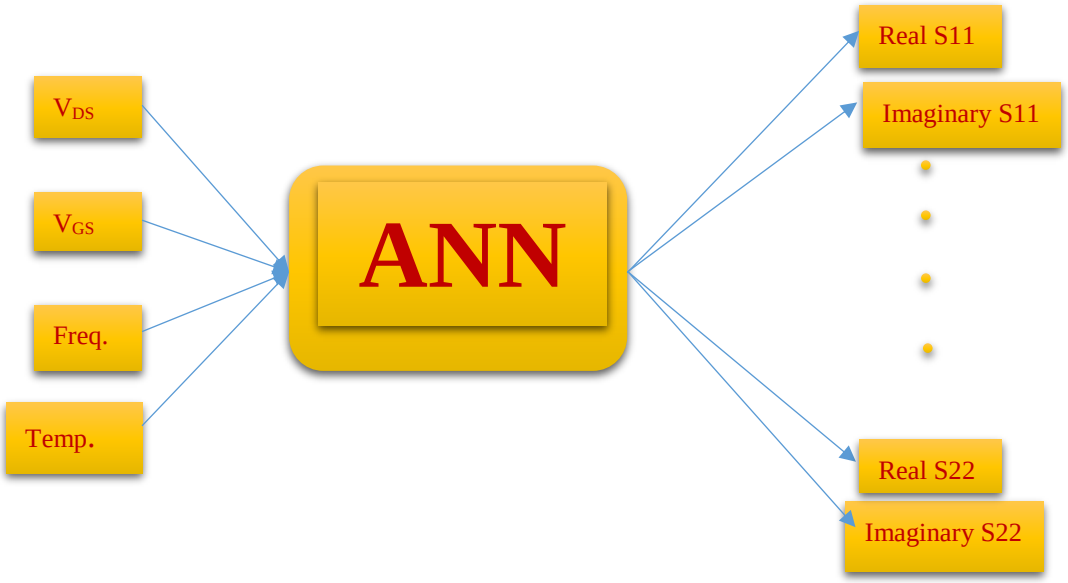

Figure 3.3: The proposed artificial neural network (ANN) model schematic

Table 3.2: Calculation of Cost functions for ANN on the independent training set

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	9.61e-06	9.97e-05	9.11e-04	7.47e-04	3.73e-06	2.78e-05	9.66e-05	7.62e-05
MAE	All bias	2.90e-03	3.48e-03	8.07e-03	7.89e-03	4.90e-04	4.86e-04	2.97e-03	3.67e-03
% R-Squared	All bias	99.91	99.95	99.70	99.99	99.95	99.94	99.93	99.91

Table 3.3: Calculation of Cost functions for ANN on the independent testing set (interpolation case)

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	8.85e-05	9.32e-04	2.26e-03	6.13e-03	3.93e-05	1.57e-04	7.25e-04	8.32e-04
MAE	All bias	2.25e-02	1.96e-02	2.13e-02	2.02e-02	2.99e-03	5.32e-03	5.08e-03	7.10e-03
% R-Squared	All bias	99.88	99.18	99.57	99.80	97.84	87.23	99.52	99.78

Table 3.4: Calculation of Cost functions for ANN on the independent Extrapolation set (Extrapolation case)

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	9.85e-04	9.56e-04	9.90e-03	9.80e-03	8.98e-04	9.96e-04	7.96e-04	8.90e-04
MAE	All bias	6.56e-02	1.98e-02	2.56e-02	7.88e-02	4.31e-03	7.75e-03	1.67e-02	8.74e-03
% R-Squared	All bias	99.81	98.47	99.50	99.52	95.76	86.92	99.30	99.60

3.4 Global Optimization Assisted ANN Model

The models previously described face a substantial challenge concerning the initialization of weights and biases. In any backpropagation (BP) algorithm, weights and biases are initialized either randomly or via a specific probability distribution function. The model's convergence is contingent upon the initial values, making it susceptible to converging at local minima. It was noted during the investigation that the models occasionally exhibit significant overfitting [49]. A typical solution involves running the algorithm repeatedly and selecting the most favourable results; however, this iterative process is both laborious and unsuitable for practical applications. Another issue arises from the narrow and elliptical shapes of the contour in the absence of normalization and proper initial values. All architectures fundamentally function similarly and rely on the initial values of weights and biases, leading to convergence problems [50]. To tackle this issue, this study suggests initializing the ANN model using Global Optimization. Figure 3.4 presents a general flow diagram.

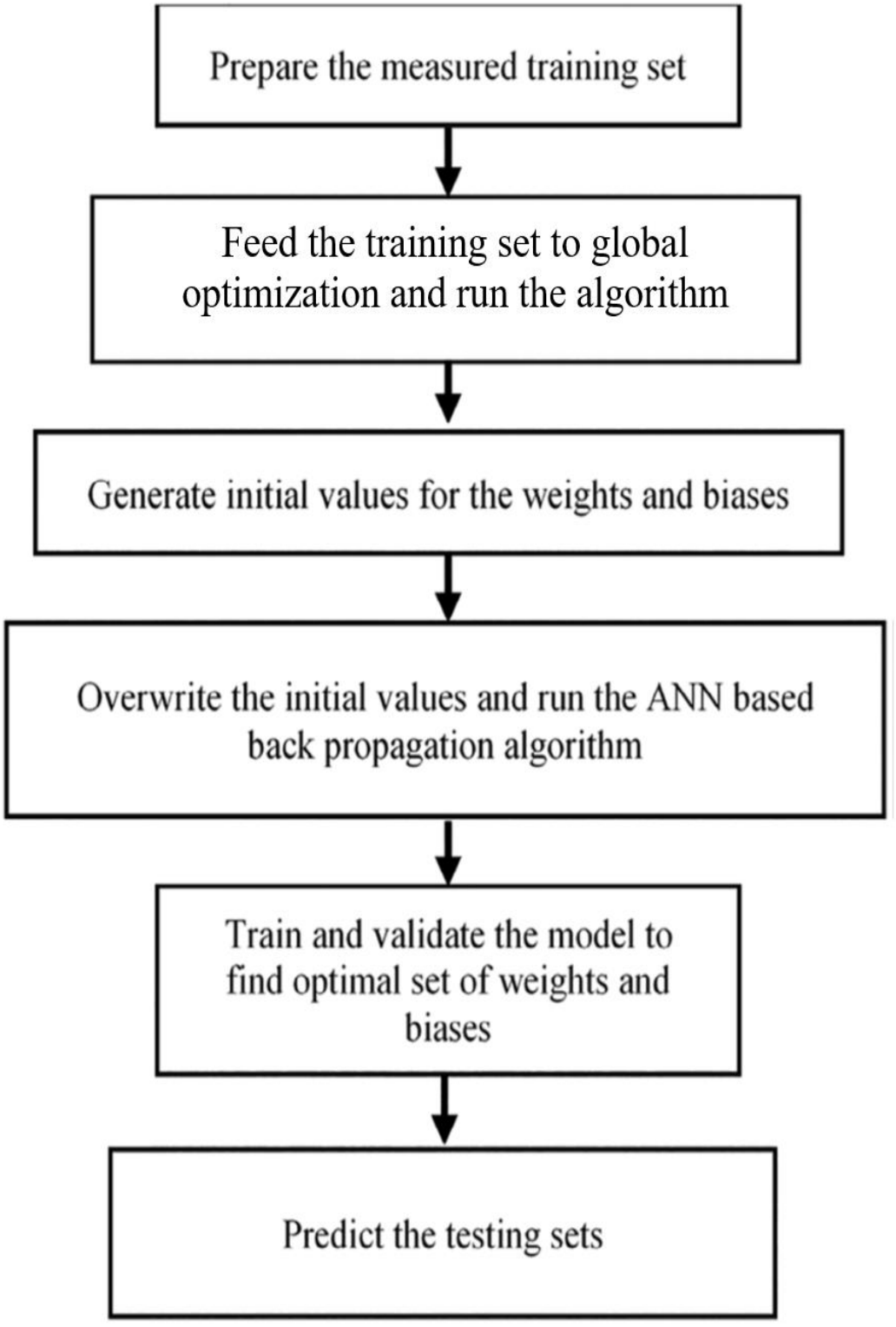

Figure 3.4: A general working flowchart of Global optimization Assisted ANN

3.4.1 GWO-PSO Assisted ANN

Grey Wolf Optimization-Particle Swarm Optimization (GWO-PSO) is a hybrid optimization algorithm that combines the strengths of both GWO and PSO algorithms. In the context of artificial neural networks (ANNs), the weights and biases play a critical role in determining the network's performance and overall functionality. The optimization of these parameters is essential for achieving accurate and efficient learning. The proposed method incorporates the utilization of a hybrid GWO-PSO algorithm, which combines the strengths of Grey Wolf Optimization (GWO) and Particle Swarm Optimization (PSO), to optimize the ANN's weights and biases. This hybrid approach enhances the search capabilities, convergence, and overall performance of the optimization process. First, the overall architecture of the neural network, including the number of input, hidden, and output layers, as well as the number of neurons in each layer has been selected through trial and error approach.

The ANN's weights and biases are represented as a vector, which is then subjected to the GWO-PSO algorithm for optimization. Each particle within the GWO-PSO population embodies a candidate solution, comprising a distinct set of weights and biases for the ANN. The particles' position and velocity in the search space represent the values of these parameters. The initial values of GWO-PSO algorithm such as; population size, number of iterations, and other algorithm-specific parameters were selected based on trial and error. After multiple time running the algorithm for different set of population size, iteration and other parameters, the one with the best results were selected; 500 population size, 150 iterations, upper and lower bound were 1 and -1 respectively.

The GWO-PSO algorithm commences with the initialization of a population of particles, wherein each particle is assigned a random position in the search space. As the algorithm iterates,

the particles traverse the search space, influenced by their own best-known position and the global best-known position in the swarm. The GWO component contributes by incorporating a social hierarchy, which guides the exploration and exploitation phases, further enhancing the algorithm's search capabilities. Throughout the optimization process, the particles' positions are continually updated based on their individual and social experiences. The objective is to minimize the error function associated with the ANN's output, effectively adjusting the weights and biases to improve the network's overall performance. Once the termination criteria are met, the best particle's position is extracted as the optimized set of weights and biases for the ANN, ultimately enhancing its learning capabilities and performance as shown by the flowchart in the figure 3.5.

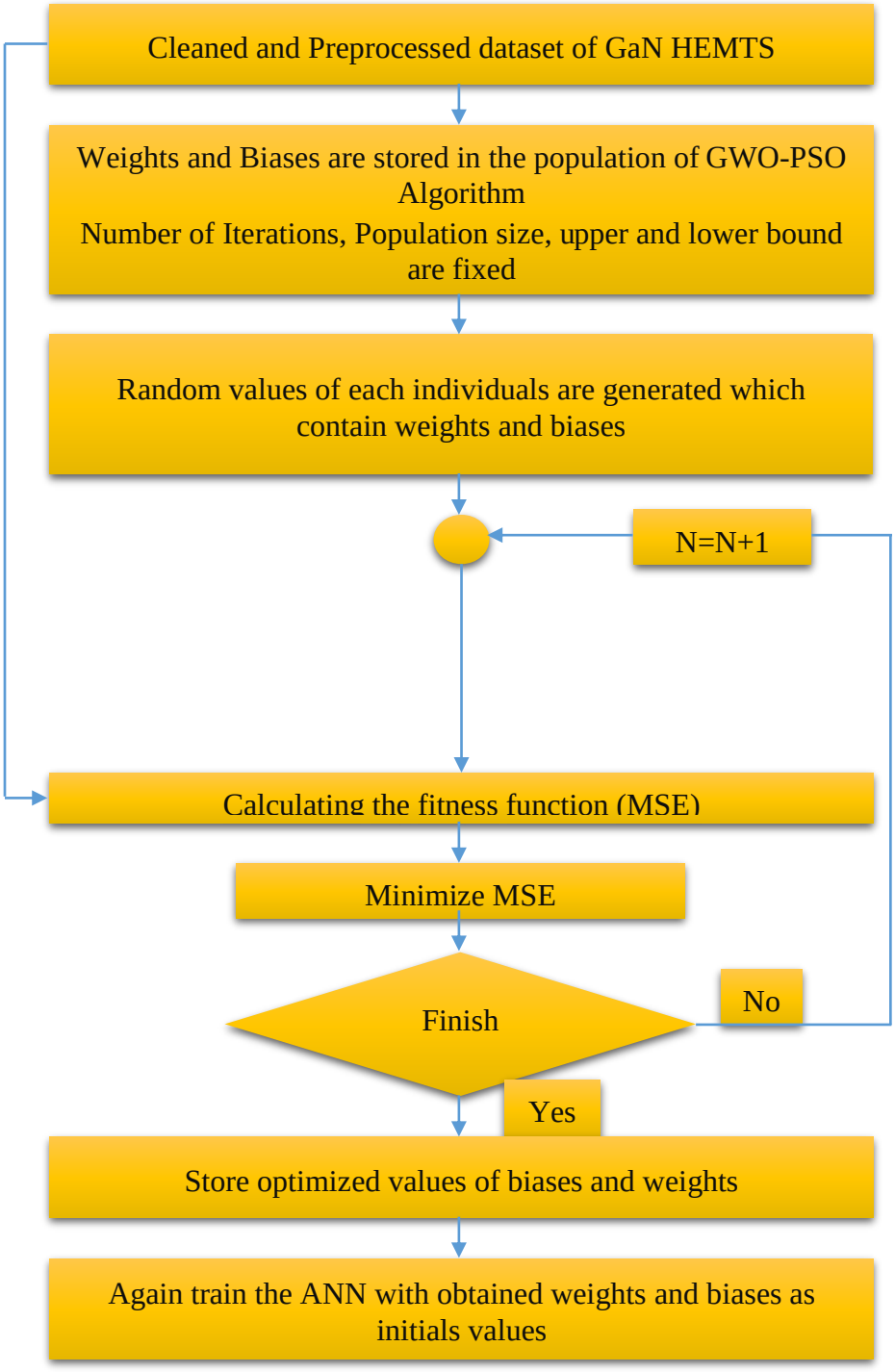

Figure 3.5: Implementation of Proposed GWO-PSO Assisted ANN

GWO-PSO algorithm was run for the specified number of iterations. In each iteration, the position and velocity of each particle according to the GWO-PSO rules were updated. The fitness function, MSE was evaluated of each particle and the global best solution was updated accordingly. After the GWO-PSO optimization has converged, the best set of weights and biases are fed to the update the ANN. The performance of the optimized ANN on training, testing and extrapolation datasets were evaluated in terms of MSE, MAE and R^2 to ensure that it generalizes well to the data as shown in the table 3.5, 3.6 and 3.7.

By following this approach, GWO-PSO can be effectively used for the validation of ANNs and optimization of their weights and biases, leading to improved performance and generalization.

Table 3.5: Calculation of Cost functions for ANN on the training set

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	9.05e-06	3.28e-06	3.04e-04	1.20e-04	2.705e-07	3.10e-07	1.16e-05	3.03e-05
MAE	All bias	4.05-03	2.13e-03	1.02e-03	4.33e-03	3.86e-04	2.82e-04	2.23e-03	2.38e-03
% R-Squared	All bias	99.98	99.98	99.99	99.99	99.98	99.97	99.99	99.95

Table 3.6: Calculation of Cost functions for ANN on the independent testing set (interpolation case)

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	2.28e-05	2.03e-05	5.01e-04	6.65e-04	1.51e-06	2.19e-06	3.50e-05	4.66e-05
MAE	All bias	4.55-03	4.11e-03	1.38e-02	1.95e-02	4.06e-04	3.19e-04	4.77e-03	4.78e-03
% R-Squared	All bias	99.97	99.92	99.93	99.87	99.87	99.76	99.94	99.91

Table 3.7: Calculation of Cost functions for ANN on the independent Extrapolation set (Extrapolation case)

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	5.65e-05	4.15e-05	3.27e-03	7.25e-04	2.98e-06	2.67e-06	5.98e-05	5.13e-05
MAE	All bias	5.38e-03	5.33e-03	1.59e-02	3.55e-02	5.21e-04	3.20e-04	5.88e-03	3.39e-03
% R-Squared	All bias	99.95	99.86	99.89	99.81	99.78	99.53	99.95	99.90

3.4.2 ALO Assisted ANN

Ant Lion Optimization (ALO) is a nature-inspired optimization algorithm based on the hunting behaviour of antlions in their interaction with ants. In this work, ALO is used for validation and optimization of Artificial Neural Networks (ANNs) by finding the best weights and biases for the network.

To optimize the weights and biases of an ANN using ALO, the first step is to define the objective function which is ANN, and loss function, MSE needs to be optimized. The second step is to initialize the antlion population, which consists of two types of entities: ants and antlion traps. The ants represent the solutions, and the antlion traps represent the best solutions found so far. The population of ants is initialized randomly, and the antlion trap is initialized to be the best solution found so far. The same number of iterations, population size, upper bound, and lower bound were chosen as in the case of GWO-PSO.

After the population is initialized, the next step is to evaluate the fitness of the solutions. The fitness of each ant is determined by using the objective function. The fitness of the ant is the value of the objective function at the solution represented by the ant. Once the fitness of each ant is calculated, the ants are moved towards the antlion traps. The position of each ant is updated by adding a random value multiplied by the difference between the position of the antlion trap and the current position of the ant using the following mathematical equation 3.6.

$$X_{new} = X_{old} + rand * (X_{antlion} - X_{old})$$

Where X_{old} is the current position of the ant, $X_{antlion}$ is the position of the antlion trap, and $rand$ is a random number between 0 and 1.

After the new positions of the ants are determined, the fitness of the new solutions is evaluated using the objective function. If the fitness of any ant is better than the fitness of the antlion trap, the antlion trap is replaced with the new ant. The process of moving the ants towards the antlion traps, evaluating the fitness of the solutions, and updating the antlion trap is repeated until the algorithm converges. The convergence criterion can be a maximum number of iterations and minimum value of the objective function as shown in the figure 3.6.

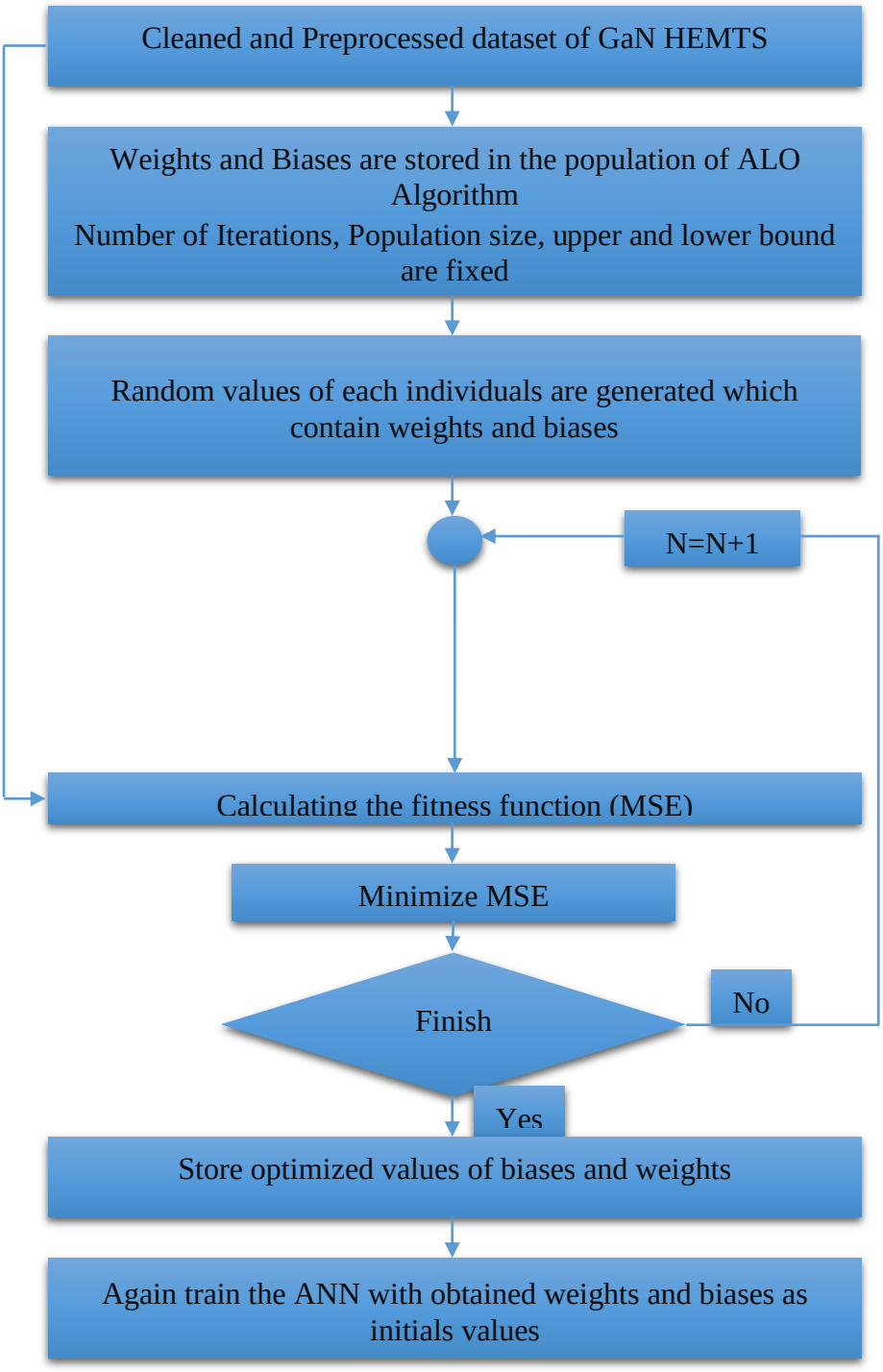

Figure 3.6: Implementation of Proposed ALO Assisted ANN

Once the algorithm converges, the optimized weights and biases can be extracted from the best solution found so far, which is represented by the antlion trap. At last, the optimized weights and biases are fed to ANN and trained again. The performance is evaluated through MSE, MAE and R^2 based on training, testing and extrapolation datasets as shown in the tables 3.8, 3.9 and 3.10 The performance of ALO can be sensitive to the choice of algorithm parameters, such as the number of ants, the mutation rate, and the crossover rate. ALO is a powerful optimization algorithm that can be used to optimize the weights and biases of ANNs, and it has the potential to improve the performance of ANNs in various applications.

Table 3.8: Calculation of Cost functions for ANN on the training set

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	8.68e-06	5.68e-06	6.55e-04	5.91e-04	8.50e-07	7.50e-07	5.50e-06	4.50e-05
MAE	All bias	7.18e-03	3.68e-03	7.90e-03	6.90e-03	3.50e-04	4.50e-04	2.50e-03	6.10e-03
% R-Squared	All bias	99.95	99.98	99.99	99.99	99.96	99.95	99.96	99.91

Table 3.9: Calculation of Cost functions for ANN on the independent testing set (interpolation case)

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	7.86e-05	6.90e-05	1.85e-03	6.91e-03	7.40e-06	8.57e-06	8.65e-05	8.20e-05
MAE	All bias	8.00e-03	7.68e-03	2.03e-02	6.03e-02	7.50e-04	6.50e-03	6.50e-03	6.50e-03
% R-Squared	All bias	99.92	99.90	99.72	99.81	99.80	97.54	99.60	99.88

Table 3.10: Calculation of Cost functions for ANN on the independent Extrapolation set (Extrapolation case)

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	9.27e-05	7.69e-04	7.53e-03	7.53e-03	7.53e-06	9.60E-06	9.19E-05	8.89E-05
MAE	All bias	8.01e-03	7.48e-03	2.53e-02	6.53e-02	8.71e-04	6.53e-03	7.50e-03	8.50e-03
% R-Squared	All bias	99.88	98.89	99.65	99.68	97.70	96.95	99.45	99.70

3.4.3 Whale Optimization Algorithm (WOA) Assisted ANN

Whale Optimization Algorithm (WOA) is a cutting-edge nature-inspired optimization technique, derived from the humpback whale's hunting behavior known as bubble-net feeding. The WOA has demonstrated remarkable efficacy in optimizing complex problems, including the optimization of weights and biases in Artificial Neural Networks (ANNs). By leveraging the WOA's exploration and exploitation capabilities, ANNs can achieve improved convergence rates and a reduction in the likelihood of being trapped in local optima.

In the context of ANNs, optimizing weights and biases is crucial for minimizing the loss function, thereby improving the network's accuracy and generalization capabilities. The WOA-based optimization process involves encoding the weights and biases into a vector called the 'search agent.' Each search agent represents a possible solution to the problem, and the algorithm iteratively updates these agents to move towards the global optimum.

The Whale Optimization Algorithm (WOA) comprises three primary behaviors: encircling prey, shrinking encircling, and search for prey. These behaviors are mathematically represented to guide the search agents in the solution space. Firstly, the encircling prey behavior exploits the best-so-far solution by adjusting the search agents' positions towards it. Secondly, the shrinking encircling behavior involves a spiral updating mechanism, which enables the search agents to move in a spiral path towards the prey. This behavior ensures a thorough exploration of the solution space while narrowing down the search area. Finally, the search for prey behavior is responsible for the algorithm's exploration capabilities, in which random search agents are generated to explore new regions in the solution space.

To optimize ANNs using WOA, first the search agents, iterations, upper and lower bound were selected same as for GWO-PSO and ALO. Afterward, the search agents are initialized with random weights and biases. During each iteration, the agents are updated according to the mathematical models of the whale's hunting behaviors, ultimately converging towards the global optimum. The fitness of each search agent is evaluated using the ANN's loss function, such as mean squared error (MSE). The algorithm terminates when a predefined stopping criterion is met, such as a maximum number of iterations or an acceptable level of fitness. The overall process is shown in the figure 3.7.

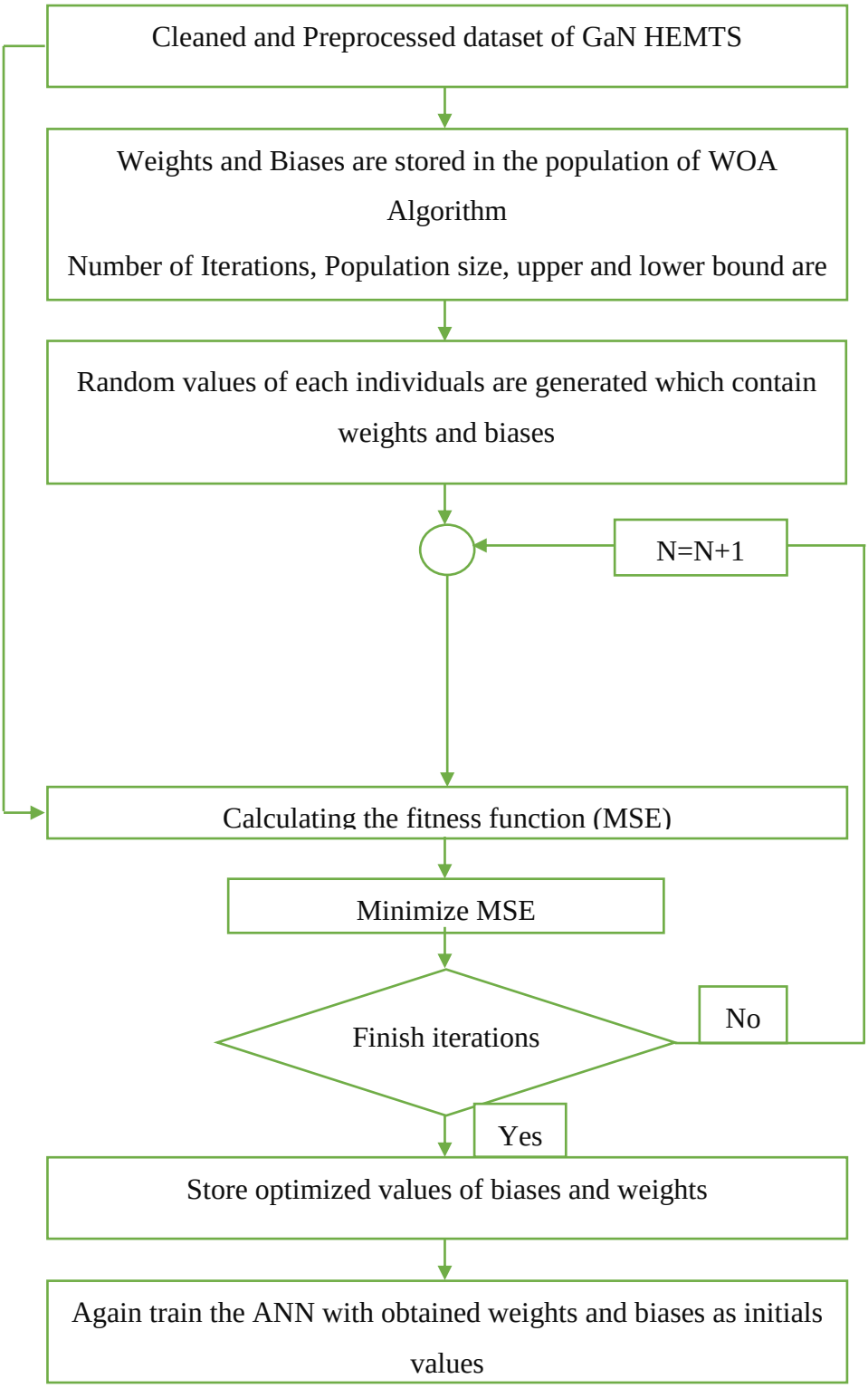

Figure 3.7: Implementation of Proposed WOA Assisted ANN

After getting the optimized weights and biases, ANN is trained again to evaluate the generalization capability based on MRE, MAE and R^2 as shown in the table 3.11, 3.12, and 3.13.

Table 3.11: Calculation of Cost functions for ANN on the training set

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	1.00e-05	4.57e-06	5.48e-04	5.44e-04	4.51e-07	6.84e-07	4.74e-06	4.33e-05
MAE	All bias	5.19e-03	2.43e-03	6.45e-03	5.65e-03	5.60e-04	3.89e-04	2.39e-03	5.35e-03
% R-Squared	All bias	99.98	99.98	99.99	99.98	99.97	99.96	99.99	99.94

Table 3.12: Calculation of Cost functions for ANN on the independent testing set (interpolation case)

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	6.67e-05	5.34e-05	1.55e-03	5.65e-03	6.37e-06	7.20e-06	6.78e-05	7.05e-05
MAE	All bias	6.57e-03	6.20e-03	1.96e-02	5.01e-02	6.30e-04	4.13e-03	5.23e-03	6.13e-03
% R-Squared	All bias	99.95	99.91	99.85	99.84	99.48	99.18	99.88	99.87

Table 3.13: Calculation of Cost functions for ANN on the independent Extrapolation set (Extrapolation case)

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	8.61e-05	6.32e-05	5.55e-03	6.55e-03	6.54e-06	8.01e-06	8.01e-05	6.13e-05
MAE	All bias	7.36e-03	6.36e-03	2.30e-02	5.30e-02	6.92e-04	5.13e-03	7.13e-03	7.13e-03
% R-Squared	All bias	99.89	99.60	99.80	99.70	99.63	98.50	99.60	99.84

3.4.4 Ant Colony Optimization (ACO) Assisted ANN

Ant Colony Optimization (ACO) is a metaheuristic optimization algorithm inspired by the foraging behavior of ants in nature. In this work, this powerful algorithm has been successfully applied to the optimization of weights and biases in Artificial Neural Networks (ANNs). By mimicking the ants' ability to find the shortest path between their nest and food source using pheromone trails, ACO can efficiently explore the solution space and converge towards the global optimum while avoiding local optima.

In ANNs, optimizing the weights and biases is essential for minimizing the loss function specifically MSE in this research, which in turn leads to enhanced accuracy and generalization capabilities. The ACO-based optimization process entails encoding the weights and biases into a solution vector called the 'ant'. Each ant

represents a candidate solution, and the algorithm iteratively updates these ants to move towards the optimal solution based on the pheromone trails and heuristic information.

The Ant Colony Optimization (ACO) algorithm is characterized by its core components: construction, pheromone update, and daemon actions. During the construction phase, ants construct a solution by traversing the solution space and selecting the next component (i.e., weights and biases) based on a probabilistic decision rule. This rule incorporates both the pheromone trails and heuristic information, guiding the ants towards promising regions in the solution space. The pheromone update phase involves the evaporation of pheromone trails and the deposition of new pheromones by the ants, based on the quality of their solutions. This adaptive mechanism promotes the exploration of new regions while reinforcing good solutions. Since ACO has many hyperparameters, such as population size, number of iterations, intensification factor q (selection pressure), deviation-distance ratio ζ (zeta), upper bound, lower bound, alpha, and beta, it is necessary to select the values of these parameters through trial and error. Those variables that are the same as in other optimization cases were assigned the same values. To optimize ANNs using ACO, ants are initialized with random weights and biases. During each iteration, ants construct their solutions, and the fitness of each solution is evaluated using the ANN's loss function, such as mean squared error (MSE). The pheromone trails are updated accordingly to favor better solutions, and the process repeats until a predefined stopping criterion is met, such as a maximum number of iterations or an acceptable level of fitness shown in the figure 3.8.

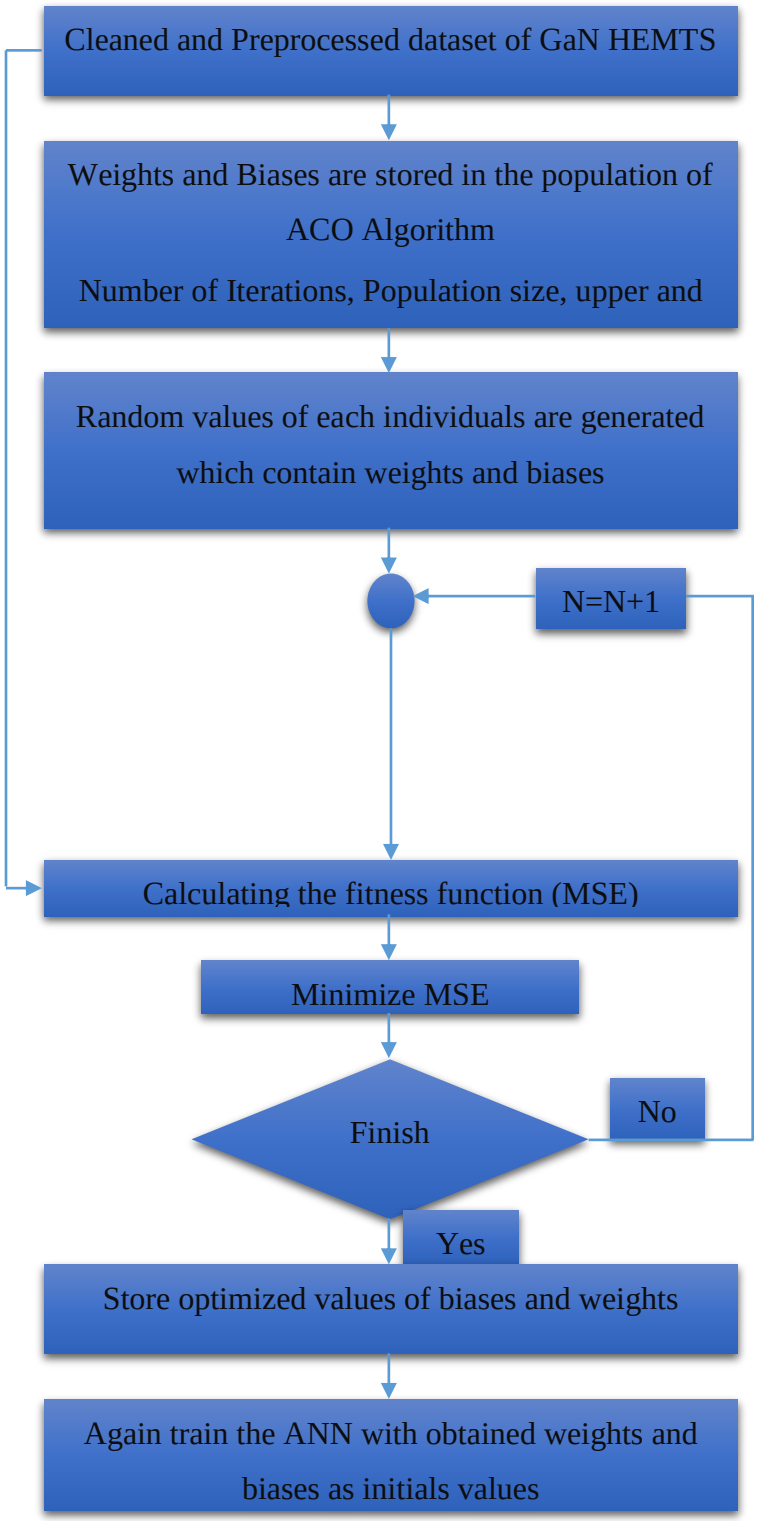

Figure 3.8: Implementation of Proposed ACO Assisted ANN

Finally, the ANN is trained again with the optimized weights and biases obtained from ACO. The proposed model was tested using training, testing, and extrapolation datasets in order to assess the generalization capability of the ANN, as shown in Tables 3.14, 3.15, and 3.16.

Table 3.14: Calculation of Cost functions for ANN on the training set

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	9.48e-05	7.40e-05	8.85e-04	7.43e-04	8.90e-07	8.83e-07	5.98e-05	5.58e-05
MAE	All bias	2.85e-03	3.40e-03	8.02e-03	7.70e-03	4.73e-04	4.85e-04	2.95e-03	3.58e-03
% R-Squared	All bias	99.92	99.96	99.75	99.99	99.96	99.95	99.70	99.92

Table 3.15: Calculation of Cost functions for ANN on the independent testing set (interpolation case)

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	8.41e-05	7.95e-05	2.22e-03	8.40e-03	9.89e-06	8.84e-06	6.25e-05	6.32e-04
MAE	All bias	8.43e-03	8.40e-03	2.20e-02	6.70e-02	9.62e-04	5.11e-03	5.01e-03	6.58e-03
% R-Squared	All bias	99.89	99.87	99.62	99.85	99.76	96.23	99.53	99.86

Table 3.16: Calculation of Cost functions for ANN on the independent Extrapolation set (Extrapolation case)

Metrics	Bias	Real (S_{11})	Img. (S_{11})	Real (S_{21})	Img. (S_{21})	Real (S_{12})	Img. (S_{12})	Real (S_{22})	Img. (S_{22})
MSE	All bias	8.45e-04	8.20e-04	8.22e-03	8.50e-03	8.50e-06	8.50e-05	6.80e-04	7.90e-04
MAE	All bias	8.53e-03	8.50e-03	2.30e-02	6.80e-02	3.30e-03	6.30e-03	8.5e-03	8.76e-03
% R-Squared	All bias	99.82	98.85	99.55	99.60	96.68	95.94	99.40	99.65

3.5 Discussion

In this research, MLP along with four optimization have been implemented; GWO-PSO, ACO, WOA, and ALO as well as ANN with MLP architecture has been implemented separately. The global optimization (GO) assisted to avoid convergence at local minima of ANN.

Global optimization (GO) effectively identifies the best weights and biases associated with global minimums. To assess and validate the performance of GO-enhanced models, the same measurement data from Table 3.1 is used throughout the entire frequency range and various

temperature ranges. The outcomes are evaluated using MSE, MAE, and $\%R^2$ metrics. It is observed that these results significantly outperform the simple MLP-based model results found in Tables 3.2 to 3.4. The enhanced performance can be attributed to the refined initial values supplied by GO. Moreover, the modeled and measured results exhibit a higher degree of agreement compared to the simple MLP model. This improvement is achieved through the accurate initialization of biases and weights, facilitating convergence to the global optimum. The calculated MSE values for training, testing (interpolation), and testing (extrapolation) scenarios are provided for all the developed models.

Regarding the comparison of the four implemented global optimization techniques, it can be concluded that the GWO-PSO augmented MLP model surpasses the other models proposed in this study in terms of MSE, $\%R^2$, MAE, and overall accuracy on the testing dataset. Additionally, the performance on the extrapolation dataset also improves. Additionally, the GWO-PSO hyper-parameter optimization time is shorter compared to the other three algorithms, indicating superior time complexity.

In contrast, ACO has more hyperparameters, such as population size, number of iterations, intensification factor q (selection pressure), deviation-distance ratio ζ (zeta), upper bound, lower bound, alpha α, and beta β, which require additional time to select the optimal values through trial and error techniques. On the other hand, GWO-PSO, ALO, and WOA have population size, iterations, upper bound, and lower bound as hyperparameters. Based on the results, GWO-PSO is the most computationally efficient and fastest algorithm among those presented in this work. The overall comparison of the implemented algorithms with MLP is provided in Table 3.17.

In summary, the results suggest that the GO-enhanced MLP model can serve as an effective alternative to the standard MLP. This hybrid approach increases the robustness of the training

process, delivering a unique solution and accurately simulating device behavior under various operating conditions and ambient temperatures. The simulation performance of the proposed GO model has also been compared with the models reported in References 16, 30, and 34, demonstrating improved outcomes. This work can further be integrated into ADS.

Table 3.17: Comparison of GWO-PSO, WOA, ALO, ACO and MLP in terms of Various Parameters

Models	Computational efficiency	Average Pr.tunning time (Minutes)	Average simulation time	Parameters /complexity	Average (MSE)	Average (%R^2)	Average (MEA)
GWO-PSO	Best	375.86	7.53	S.Agents = 500 Dimensions = 491 Iteration = 150 Up. Bound = 1 L.Bound = -1 N.Epocs = 1000	5.26e-04	99.83	9.03e-03
WOA	Best	378.31	7.63	S.Agents = 500 Dimensions = 491 Iteration = 150 Up. Bound = 1 L.Bound = -1 N.Epocs = 1000	1.55e-03	99.57	1.37e-02
ALO	Better	384.82	7.70	S.Agents = 500 Dimensions = 491 Iteration = 150 Up. Bound = 1 L.Bound = -1 N.Epocs = 1000	2.01e-03	98.98	1.62e-02
ACO	Good	389.69	7.78	S.Agents = 500 Dimensions = 491 Iteration = 150 Up. Bound = 1	2.49e-03	98.68	1.69e-02

				L.Bound = -1 N.Epocs = 1000			
MLP		-	7.10	-	3.15e-03	97.36	2.84e-02

To better comprehend the error profile of the GOs discussed in this study, the average Mean Squared Error (MSE), percentage coefficient of determination (R^2), and Mean Absolute Error (MAE) have been illustrated in Figures 3.9, 3.10, and 3.11, respectively.

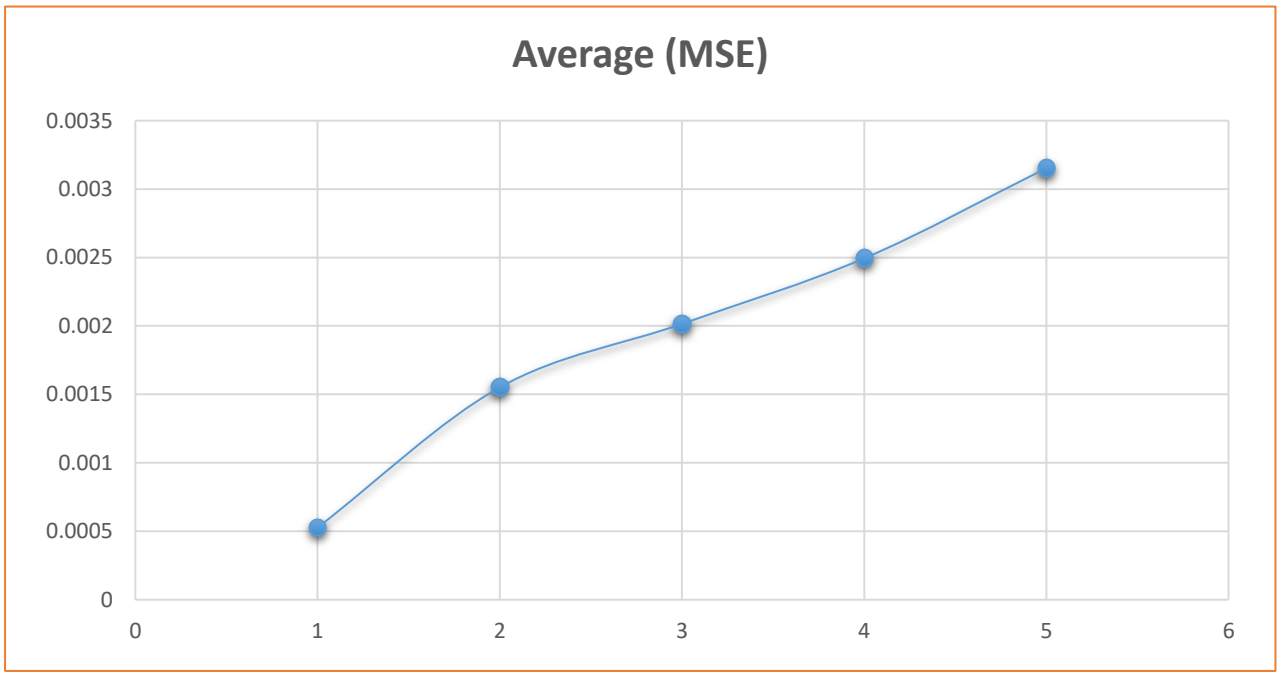

Figure 3.9: Average MSE of GWO-PSO, WOA, ALO, ACO and MLP

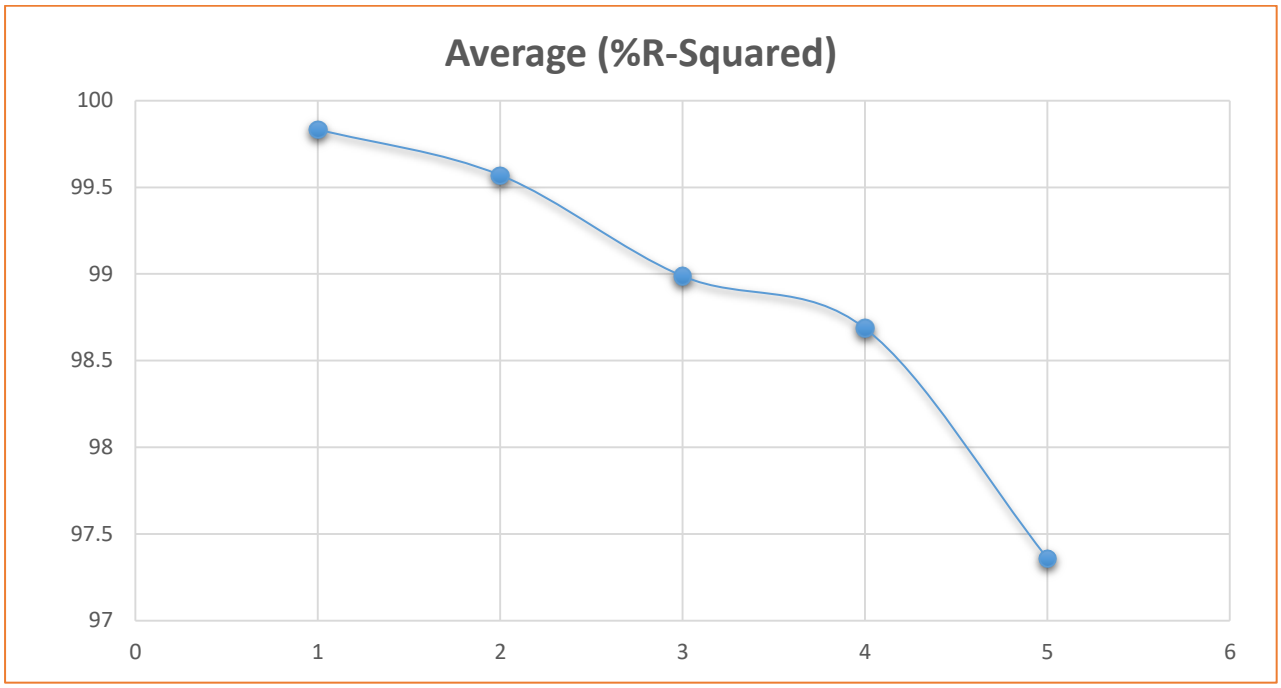

Figure 3.10: Average % R2 of GWO-PSO, WOA, ALO, ACO and MLP

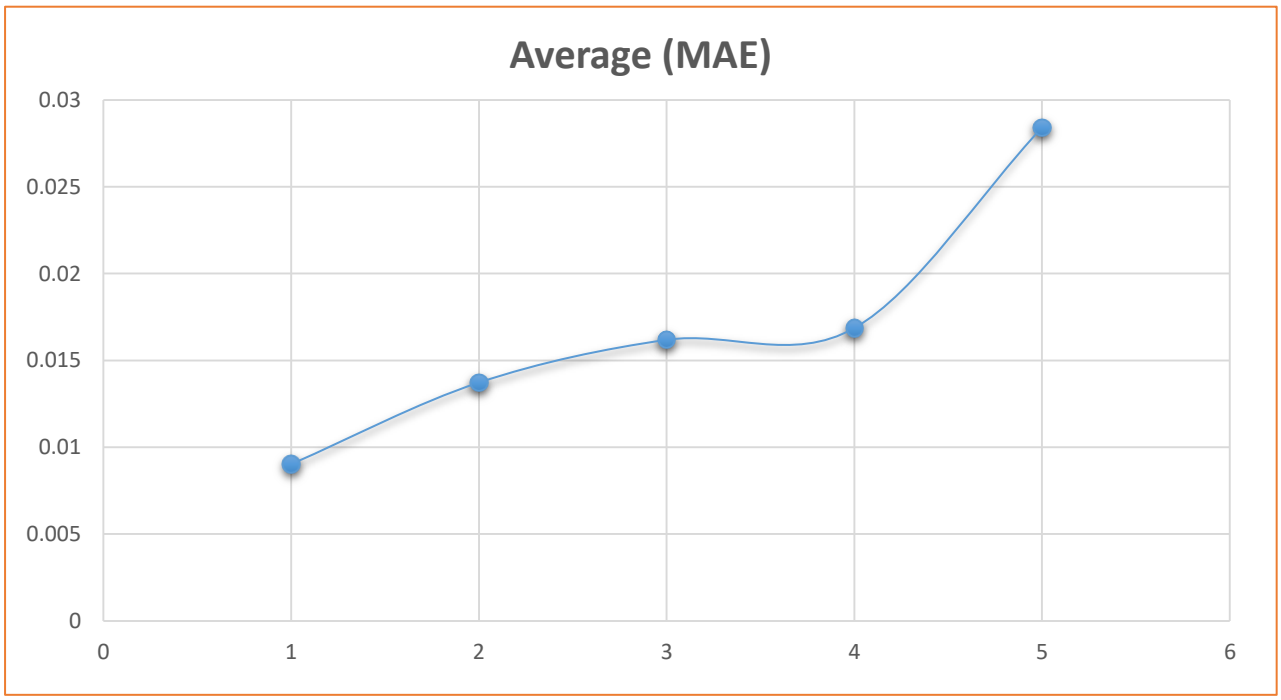

Figure 3.11: Average MAE of GWO-PSO, WOA, ALO, ACO and MLP

Finally, the convergence curves of all four algorithms are plotted in Figure 3.12.

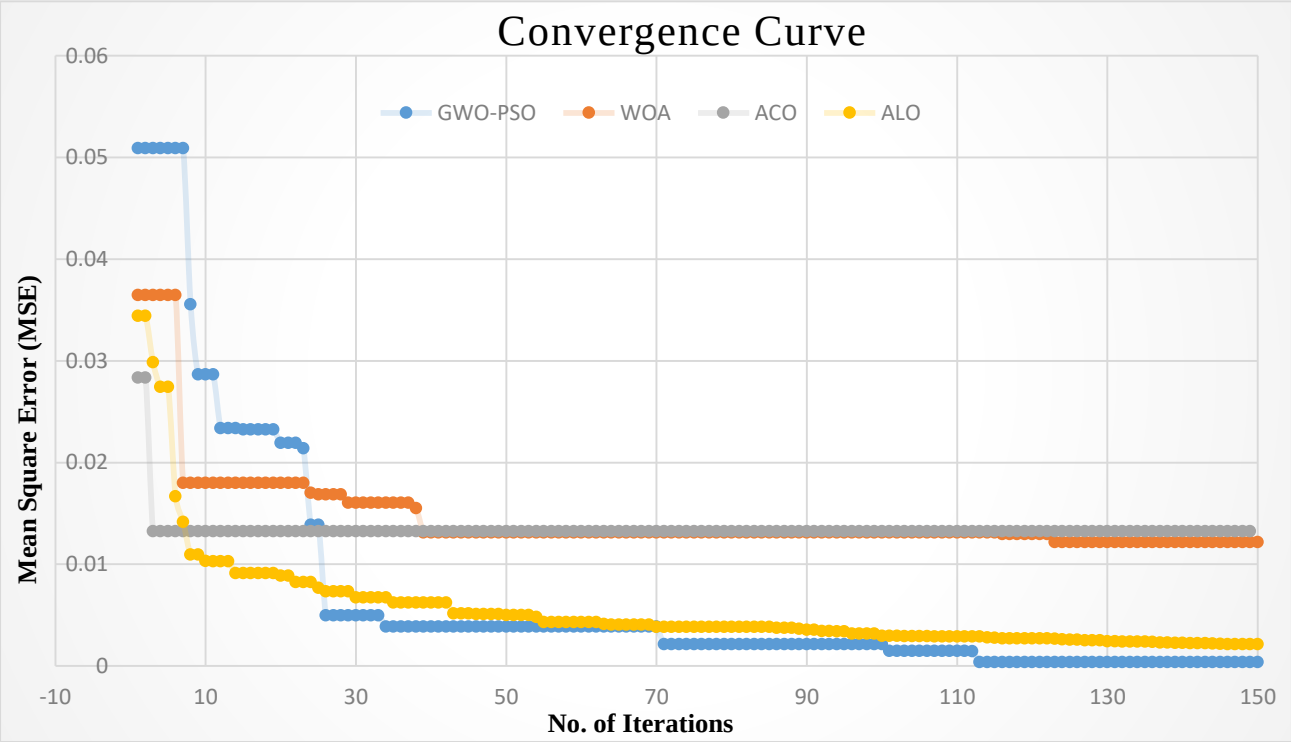

Figure 3.12: Convergence Curve of GWO-PSO, WOA, ALO, ACO

Chapter- 4 Conclusion

This study developed and examined an efficient behavioral modeling approach for GaN-on-Si HEMT device. It was determined that an MLP featuring a 4-10-20-10-1 topology effectively models the device's behavior under varying voltage, frequency, and temperature conditions. The outstanding agreement between temperature-dependent data and modeled results highlights the model's robustness in managing temperature-dependent S-parameter modeling. Prior research suggested that ANN performance depends on initial weight and bias values, with inaccurate initialization potentially leading to overfitting. This study addressed these issues by determining optimal initialization values using global optimization (GO) algorithms. Validation confirms that the GWO-PSO-enhanced MLP offers the most robust performance across a range of measurement data acquired under diverse operating conditions and requires less time to optimize hyper-parameters. WOA and ALO shows relative results while ACO has many hyper-parameters to be optimized first which in turn takes more time and is computationally inefficient. In this work, single-objective global optimization has been employed, as it was exclusively required for optimizing weights and biases. However, for future work, multi-objective global optimization can be utilized to determine the optimal architecture of an Artificial Neural Network (ANN), including number of hidden layers, the number of neurons, the activation function, and number of epochs.

References

[1] J. C. Rautio, "Transistor-level behavioral modeling of microwave circuits," IEEE Transactions on Microwave Theory and Techniques, vol. 67, no. 9, pp. 3495-3506, 2019.

[2] Y.-F. Wu, A. Saxler, M. Moore, R. Smith, S. Sheppard, P. M. Chavarkar, S. Keller, S. P. DenBaars, and U. K. Mishra, "30-W/mm GaN HEMTs by field plate optimization," IEEE Electron Device Lett., vol. 25, no. 3, pp. 117-119, Mar. 2004.

[3] Y. Cai, Y. Zhou, Q. Chen, K. J. Chen, and X. Zhang, "Review of technology for low-noise GaN HEMT-based microwave monolithic integrated circuits," IEEE Access, vol. 5, pp. 17460-17471, 2017.

[4] A. Maaloum, F. Aniel, and E. Morvan, "Noise performance of AlGaN/GaN HEMTs for low-noise applications," IEEE Trans. Electron Devices, vol. 55, no. 12, pp. 3479-3485, Dec. 2008.

[5] X. Cheng, M. Li, and Y. Wang, "Physics-based compact model for AlGaN/GaN MODFETs with close-formed I–V and C–V characteristics," IEEE Trans. Electron Devices, vol. 56, no. 12, pp. 2881-2887, Dec. 2009.

[6] A. Jarndal et al., "Large-signal model for AlGaN/GaN HEMTs suitable for RF switching-mode power amplifiers design," Solid-State Electron., vol. 54, no. 7, pp. 696–700, Jul. 2010.

[7] A. Jarndal, A. Z. Markos, and G. Kompa, "Improved modeling of GaN HEMTs on Si substrate for design of RF power amplifiers," IEEE Trans. Microw. Theory Techn., vol. 59, no. 3, pp. 644-651, Mar. 2011.

[8] A. Khusro, S. Husain, M. S. Hashmi, and A. Q. Ansari, "Small signal behavioral modeling technique of GaN high electron mobility transistor using artificial neural network: An accurate, fast, and reliable approach," in Int. J. RF Microw. Comput.-Aided Eng., vol. 30, no. 4, Apr. 2020, Art. no. e22112.

[9] A. Jarndal, "On neural networks based electrothermal modeling of GaN devices," in IEEE Access, vol. 7, pp. 94205-94214, 2019.

[10] J. B. King and T. J. Brazil, "Nonlinear electrothermal GaN HEMT model applied to high-efficiency power amplifier design," IEEE Trans. Microw. Theory Techn., vol. 61, no. 1, pp. 444-454, Jan. 2013.

[11] A. H. Jarndal and S. Muhaureq, "A particle swarm neural networks electrothermal modeling approach applied to GaN HEMTs," in J. Comput. Electron., vol. 18, pp. 1272-1279, 2019.

[12] Z. Marinković, G. Crupi, A. Caddemi and V. Marković, "GaN HEMT small-signal modelling: Neural networks versus equivalent circuit," 2017 IEEE 30th International Conference on Microelectronics (MIEL), pp. 153-156, 2017.

[13] A.-D. Huang, Z. Zhong, W. Wu, and Y.-X. Guo, "An artificial neural network-based electrothermal model for GaN HEMTs with dynamic trapping effects consideration," IEEE Trans. Microw. Theory Techn., vol. 64, no. 8, pp. 2519-2528, Aug. 2016.

[14] S. Husain, M. Hashmi, A. Jarndal, M. Chaudhary and G. Nauryzbayev, "Comparative Analysis of ANN Architectures for the Development of GaN HEMT Small-Signal Model," 2021 IEEE MTT-S International Microwave and RF Conference (IMARC), KANPUR, India, 2021

[15] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," J. Mach. Learn. Res., vol. 13, no. 2, pp. 281–305, 2012.

[16] S. Husain, A. Khusro, M. Hashmi, G. Nauryzbayev, and M. A. Chaudhary, "Gaussian process regression for small-signal modelling of GaN HEMTs," in Proc. IEEE Int. Conf. Consum. Electron., Las Vegas, NV, USA, Jan. 2021, pp. 1–4.

[17] S. Husain, M. Hashmi, A. Jarndal, M. Chaudhary and G. Nauryzbayev, "Comparative Analysis of ANN Architectures for the Development of GaN HEMT Small-Signal Model," 2021 IEEE MTT-S International Microwave and RF Conference (IMARC), KANPUR, India, 2021, pp. 1-4

[18] A. Khusro, M. S. Hashmi, A. Q. Ansari, A. Mishra, and M. Tarique, “An accurate and simplified small signal parameter extraction method for GaN HEMT,” Int. J. Circuit Theory Appl., vol. 37, no. 6, pp. 941–953, Jun. 2019

[19] A. Mishra, A. Khusro, M. S. Hashmi and A. Q. Ansari, "Modeling and parameter extraction method for AlGaN/GaN HEMT," 2017 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT), Aligarh, India, 2017, pp. 214-217, doi: 10.1109/MSPCT.2017.8364007.

[20] G. Crupi, A. Raffo, V. Vadalà, G. Vannini and A. Caddemi, "High-periphery GaN HEMT modeling up to 65 GHz and 200 °C", Solid-State Electronics, vol. 152, pp. 11-16, 2019.

[21] X. Du et al., “ANN-Based Large-Signal Model of AlGaN/GaN HEMTs With Accurate Buffer-Related Trapping Effects Characterization,” IEEE Trans. Microw. Theory Tech., vol. 68, no. 7, pp. 3090-3099, July 2020.

[22] Z. Marinković, G. Crupi, A. Caddemi, V. Marković and D. Schreurs, "A review on the artificial neural network applications for small-signal modeling of microwave FETs", International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, vol. 33, no. 3, 2019.

[23] A. Jarndal, "Neural network electrothermal modeling approach for microwave active devices", International Journal of RF and Microwave Computer-Aided Engineering, vol. 29, no. 9, 2019.

[24] A. Majumdar, S. Chatterjee, S. Chatterjee, S. Chaudhari and D. Poddar, "An ambient temperature dependent small signal model of GaN HEMTusing method of curve fitting", International Journal of RF and Microwave Computer-Aided Engineering, vol. 30, no. 12, 2020.

[25] A. Jarndal et al., "Large-signal model for AlGaN/GaN HEMTs suitable for RF switching-mode power amplifiers design," Solid-State Electron., vol. 54, no. 7, pp. 696-700, Jul. 2010.

[26] A. Jarndal, "On Neural Networks Based Electrothermal Modeling of GaN Devices", IEEE Access, vol. 7, pp. 94205-94214, 2019.

[27] Z. Marinković et al., "Temperature Dependent Small-Signal Neural Modeling of High-Periphery GaN HEMTs," 2019 14th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), pp. 33-36, 2019.

[28] A. Jarndal and A. Hussein, "Hybrid small-signal model parameter extraction of GaN HEMTs on Si and SiC substrates based on global optimization", International Journal of RF and Microwave Computer-Aided Engineering, vol. 29, no. 10, 2018.

[29] S. Agnihotri, S. Ghosh, A. Dasgupta, S. A. Ahsan, S. Khandelwal and Y. S. Chauhan, "Modeling of trapping effects in GaN HEMTs," 2015 Annual IEEE India Conference (INDICON), New Delhi, India, 2019, pp. 1-4, doi: 10.1109/INDICON.2015.7443658.

[30] A. Khusro, S. Husain, M. S. Hashmi, A. Q. Ansari, and S. Arzykulov, "A Generic and Efficient Globalized Kernel Mapping-Based Small-Signal Behavioral Modeling for GaN HEMT," IEEE Access, vol. 8, pp. 195046–195061, 2020.

[31] Jarndal A, Hussein A. Hybrid small-signal model parameter extraction of GaN HEMTs on Si and SiC substrates based on global optimization. Int J RF Microwave Comput-Aided Eng. 2018;29(10)

[32] A. Jarndal, S. Hamdan and M. Bettayeb, "On Neural Networks Modeling Based on GA, PSO and GW Optimization Techniques," 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO), pp. 1-5, 2019.

[33] A. Jarndal, L. Arivazhagan and D. Nirmal, "On the performance of GaN-on-Silicon, Silicon-Carbide, and Diamond substrates", International Journal of RF and Microwave Computer-Aided Engineering, vol. 30, no. 6, 2020.

[34] S. Husain, A. Hamadeh, O. Zaatar, T. A. Gumaei, and A. Jarndal, "Temperature Dependent SVR and ANN based I-V Models for GaN HEMTs," in Proc. 2020 Int. Conf. Commun., Comput., Cybersecur., and Inform. (CCCI), Sharjah, United Arab Emirates, 2020, pp. 1-4, doi: 10.1109/CCCI49893.2020.9256407.

[35] B. S¸enel and F. A. S¸enel, "Novel neural network optimization approach for modeling scattering and noise parameters of microwave transistor," in Int. J. Numer. Model., vol. 35, no. 1, Jan. 2022, Art. no. e2930.

[36] Nitronex, "GaN essentials TM AN-011: Substrates for GaN RF devices," Nitronex Application Notes, 2008.

[37] S. Husain, K. Khan, A. Jarndal, G. Nauryzbayev, and M. Hashmi, "Temperature Dependent I-V Models for Microwave Transistor Using Radial Basis NNs, Generalized Regression NNs and Feedforward NN," in Proc. 2022 5th Int. Conf. Multimedia, Signal Process. Commun. Technol. (IMPACT), Aligarh, India, 2022, pp. 1-5, doi: 10.1109/IMPACT55510.2022.10029074.

[38] Ö. F. Ertuğrul, "A novel type of activation function in artificial neural networks: Trained activation function," in Neural Netw., vol. 99, pp. 148-157, Mar. 2018, doi: 10.1016/j.neunet.2018.01.007.

[39] S. Mirjalili, "The ant lion optimizer," in Adv. Eng. Softw., vol. 83, pp. 80-98, 2015.

[40] S. Mirjalili and A. Lewis, "The whale optimization algorithm," in Adv. Eng. Softw., vol. 95, pp. 51-67, 2016.

[41] M. A. Shaheen, H. M. Hasanien, and A. Alkuhayli, "A novel hybrid GWO-PSO optimization technique for optimal reactive power dispatch problem solution," in Ain Shams Eng. J., vol. 12, no. 1, pp. 621-630, 2021.

[42] S. Mirjalili and S. Mirjalili, "Ant colony optimisation," in Evolutionary Algorithms and Neural Networks: Theory and Applications, pp. 33-42, 2019.

[43] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," in Proc. 1999 Congr. Evol. Comput. (CEC), 1999, doi: 10.1109/CEC.1999.782657.

[44] M. A. González-Sentís, P. Tounsi, A. Bensoussan, and A. Dufour, "Degradation indicators of power-GaN-HEMT under switching power-cycling," in Microelectron. Reliab., vol. 100, Art. no. 113412, 2019.

[45] X.-H. Zhou, C. Zhou, D. Lui, and X. Ding, Applied Missing Data Analysis in the Health Sciences. John Wiley & Sons, 2014.

[46] A. Cahsai, C. Anagnostopoulos, and P. Triantafillou, "Scalable data quality for big data: the pythia framework for handling missing values," in Big Data, vol. 3, no. 3, pp. 159-172, 2015.

[47] M. A. Wani Saduf, "Comparative study of backpropagation learning algorithms for neural networks," in Int. J. Adv. Res. Comput. Sci. Softw. Eng. (IJARCSSE), vol. 3, no. 12, pp. 1151-1156, Dec. 2013.

[48] D. Wu, H. Huang, S. Qiu, Y. Liu, Y. Wu, Y. Ren, and J. Mou, "Application of Bayesian regularization back propagation neural network in sensorless measurement of pump operational state," in Energy Rep., vol. 8, pp. 3041-3050, 2022.

[49] Z. Tian, W. Gan, X. Zou, Y. Zhang, and W. Gao, "Performance prediction of a cryogenic organic Rankine cycle based on back propagation neural network optimized by genetic algorithm," in Energy, vol. 254, Art. no. 124027, 2022.

[50] A. Jarndal, S. Husain, and M. Hashmi, "Genetic algorithm initialized artificial neural network based temperature dependent small-signal modeling technique for GaN high electron mobility transistors," in Int. J. RF Microw. Comput.-Aided Eng., vol. 31, no. 3, 2021.

Back Cover