MICROSTRIP PATCH ANTENNA MODELING WITH KNOWLEDGE-BASED NEURAL NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EKİN SU SAÇIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONIC ENGINEERING

AUGUST 2023

Approval of the thesis:

MICROSTRIP PATCH ANTENNA MODELING WITH KNOWLEDGE-BASED NEURAL NETWORKS

submitted by **EKİN SU SAÇIN** in partial fulfillment of the requirements for the degree of **Master of Science** i**n Electrical and Electronic Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of the Department, **Electrical and Electronics Eng.** _____

Assist. Prof. Dr. Ahmet Cemal Durgun
Supervisor, **Electrical and Electronics Eng., METU** _____

Examining Committee Members:

Prof. Dr. Sencer Koç
Electrical and Electronics Eng., METU _____

Assist. Prof. Dr. Ahmet Cemal Durgun
Electrical and Electronics Eng., METU _____

Assoc. Prof. Dr. Lale Alatan
Electrical and Electronics Eng., METU _____

Prof. Dr. Özgür Ergül
Electrical and Electronics Eng., METU _____

Prof. Dr. Özlem Özgün
Electrical and Electronics Eng., Hacettepe Uni. _____

Date: 01.08.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name : Ekin Su Saçın

Signature :

ABSTRACT

MICROSTRIP PATCH ANTENNA MODELING WITH KNOWLEDGE-BASED NEURAL NETWORKS

Saçın, Ekin Su
Master of Science, Electrical and Electronic Engineering
Supervisor: Assist. Prof. Dr. Ahmet Cemal Durgun

August 2023, 105 pages

Designing and optimizing microwave devices require significant computational resources, particularly when dealing with intricate stack-up structures. Utilizing machine learning techniques based on neural networks as surrogate models for microwave device design and optimization offers the advantage of reducing computational resource usage. In this thesis, a spectral transposed convolutional neural network was developed for the design of a microstrip patch antenna on a package. The neural network was trained to predict the S_{11} parameters and gain of the antenna as a function of its geometrical parameters and material properties. The neural network provides scalable frequency resolution by upsampling the output dimension through transposed convolution layers. The antenna was intentionally designed without a predefined feeding network to allow the integration of versatile feeding techniques based on specific criteria. Weights of the neural network are updated using the Huber loss function, while the relative squared error (RSE) was adopted as the evaluation metric. To further enhance the performance of the neural network with a smaller dataset, knowledge-based regularization methods were implemented during the training process. These methods include integrating derivatives and spectral domain representation of the signal into the loss function,

and magnitude regularization to minimize passivity violation. Moreover, the loss function of the neural network was improved by incorporating the cavity model of the patch antenna, allowing accurate prediction of the antenna performance while considering resonant frequencies of the relevant modes. This work demonstrated the remarkable capability of neural networks to accurately predict patch antenna performance.

Keywords: Neural Network, Spectral Transposed Convolutional Neural Network, Microstrip Patch Antenna, Knowledge-Based Regularization, Cavity Model

ÖZ

BİLGİYE DAYALI YAPAY SİNİR AĞLARI KULLANILARAK MİKROŞERİT YAMA ANTENİN MODELLENMESİ

Saçın, Ekin Su
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği
Tez Yöneticisi: Assist. Prof. Dr. Ahmet Cemal Durgun

Ağustos 2023, 105 sayfa

Mikrodalga cihazların tasarımı ve optimize edilmesi, özellikle fazla sayıda katman içeren paket yapılarıyla uğraşırken önemli miktarda kaynak tüketimi gerektirir. Mikrodalga cihaz tasarımı ve optimizasyonu için vekil modeller olarak yapay sinir ağlarını baz alan makine öğrenimi tekniklerinin kullanılması, kaynak kullanımını azaltmayı sağlar. Bu tezde, bir paket üzerine yerleştirilecek mikroşerit yama antenin tasarımı için bir spektral transpoze konvolüsyonel sinir ağı geliştirilmiştir. Sinir ağı, S_{11} parametrelerini ve antenin kazancını, geometrik parametrelerinin ve materyal özelliklerinin bir fonksiyonu olarak tahmin etmek için eğitildi. Sinir ağı, transpoze konvolüsyonel katmanlar aracılığıyla, çıktının boyutunu artırarak ölçeklenebilir frekans çözünürlüğü sağlar. Anten, belirli kriterlere dayalı farklı besleme tekniklerinin entegrasyonuna izin verebilmek için, önceden tanımlanmış bir besleme yapısı olmadan tasarlandı. Sinir ağının ağırlıkları, Huber kayıp fonksiyonu kullanılarak güncellenirken, değerlendirme ölçütü olarak göreli hata karesi (RSE) belirlendi. Daha küçük bir veri seti ile sinir ağının performansını artırmak için, eğitilme sürecinde bilgiye dayalı regülarizasyon yöntemleri uygulandı. Bu yöntemler, sinyalin türevlerini ve spektral alan temsilini kayıp fonksiyonuna entegre etmeyi ve pasiflik ihlalini en aza indirmek için büyüklük regülarizasyonunu içerir.

Ayrıca sinir ağının kayıp fonksiyonu, ilgili modların rezonans frekanslarını dikkate alarak anten performansını doğru bir şekilde tahmin edilmesini sağlayacak yama anteninin boşluk modeli dahil edilerek iyileştirildi. Bu çalışma, yama anten performansını doğru bir şekilde tahmin etmek için sinir ağlarının kayda değer yeteneğini gösterdi.

Anahtar Kelimeler: Sinir Ağı, Spektral Transpoze Konvolüsyonel Sinir Ağı, Mikroşerit Yama Anten, Bilgiye Dayalı Regülarizasyon, Oyuk Model

To my family

ACKNOWLEDGMENTS

I wish to express my deepest gratidude to my supervisor Assist. Prof. Dr. Ahmet Cemal Durgun for his advice, guidance, constructive criticism and encouragements thorough the research.

I would also thank to Assist. Prof. Dr. Mustafa Gökçe Baydoğan for his suggestions and comments.

I am deeply grateful to the esteemed jury members of my thesis defense, Prof. Dr. Sencer Koç, Assoc. Prof. Dr. Lale Alatan, Prof. Dr. Özgür Ergül, and Prof. Dr. Özlem Özgün, for their comments, constructive criticism, and valuable guidance.

The technical assistance of Hasan Said Ünal are gratefully acknowledged.

I would also appreciate to Furkan Şık for his invaluable support.

This thesis has been partially supported by TÜBİTAK Project No 118C198.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

CNN	Convolutional Neural Network
EM	Electromagnetic
FCNN	Fully Connected Neural Network
MAE	Mean Absolute Error
MmWave	Millimeter Wave
MSE	Mean Squared Error
NN	Neural Network
RAE	Relative Absolute Error
RMSE	Root Mean Square Error
RRMSE	Relative Root Mean Square
RSE	Relative Squared Error
STCNN	Spectral Transposed Convolutional Neural Network

CHAPTER 1

INTRODUCTION

The complexity of the design process for high frequency and high-level applications is intensified by the increasing demand, necessitating rigorous electromagnetic (EM) analysis and extensive computational resources. To mitigate the computational burden, Neural Networks (NN) can be employed to create surrogate models for EM analysis, thereby reducing computational time. This thesis presents the development of a spectral transposed convolutional neural network (STCNN) specifically designed for microstrip antennas printed on packages. By leveraging the geometrical and material properties of the antenna, this model accurately estimates the S_{11} parameter and gain of the antenna within the frequency band of 23-33 GHz. This chapter provides a concise overview of the historical development of NN surrogate modeling, highlighting the key innovations that have been discovered, and the original contributions of this work to the field.

1.1 A Brief History of Neural Network (NN) Surrogate Models

The human brain inspires NNs to process information. They can be trained to realize any arbitrary nonlinear input/output relationships by employing data. An NN consists of processing units mimicking neurons and their interconnections. These connections are weighted, allowing information to be processed, and outputted based on the activation threshold of each neuron. These characteristics enable NNs to be trained to represent an input/output mapping depending on the data content, even for complex behaviors.

The universal approximation theorem states that any nonlinear continuous function can be approximated by piecewise functions. These piecewise functions could be considered as constant steps with enough numbers, which can approximate the function by adjusting the weights, biases, and thresholds. In this way, a relation between the input and output data can be created by using simple linear operations instead of solving complex mathematical expressions. In this regard, it is possible to approximate RF/microwave characteristics as a function of the system's physical or geometrical parameters [1].

Machine learning (ML) has become popular and more effective in recent years because of the abundance of data. RF/microwave devices have also been designed and optimized with this unconventional method [2]. For instance in [3], the field effect transistor (FET) model's physical/geometrical parameters were optimized by NN considering gate, drain, and source currents and total charges. NN optimization increases the speed of the design process by avoiding repeated simulations and solving physical expressions after some minor modifications. Another implementation of NN focused on creating a relation to understand the effect of coplanar waveguide (CPW) discontinuities like chamfered bends, T-junctions, steps-in-width, open and short circuit stubs on S-parameters [4]. In another research, using NN, S-parameters of microstrip square spiral inductors were modeled to observe the parasitic and coupling effect of the components [5]. Modeling and optimization of the geometrical parameters of vias and their interconnections were constructed by NN mapping them to the return loss [6].

Although the mathematical nature of ML makes it a valuable tool for overcoming complex models, the solutions based on input/output relationships cannot always be physical since it does not contain any physical knowledge of the structure. To tackle this challenge, implementing knowledge-based algorithms can enhance the understanding of the underlying physics of the structure. These knowledge-based algorithms were applied to the automatic generation of radio frequency integrated circuit (RFIC) components like laterally-diffused metal-oxide semiconductor

(LDMOS) power transistors, for which accurate modeling is critical, and the computational cost is too high to solve all physical equations for each model [7].

ML based algorithms have also been applied to surrogate models to design and optimization of different types of antennas [8-11]. NNs have been used to predict the reflectarray design and optimization [12-14]. For a planar inverted F antenna, the properties of the ferromagnetic material substrate were adjusted by ML to predict the required permeability and permittivity for the desired antenna performance [15]. The double T-shaped monopole antenna's geometrical parameters were optimized by ML [16]. The profiled corrugated circular horn, being a contemporary antenna type characterized by its compact size, necessitates precise modeling and extensive computational time for a practical design, which was accomplished by employing an NN to obtain its geometrical parameters while considering both its co-polar and cross-polar patterns. [17].

The increasing demand for miniaturization has led to a greater utilization of patch antennas. The advantage of using NNs for patch antenna design and optimization is their ability to handle the nonlinear and highly complex relationships between the antenna's geometrical parameters and its performance characteristics. NNs can capture intricate patterns and nuances that may be challenging to model analytically. Moreover, the optimization process can be accelerated by exploring a vast design space more efficiently through the NN's predictive capabilities. In the literature, some optimization studies have been completed by NNs for antennas on package [18 – 21].

In conclusion, implementing an NN based surrogate model, which predicts the performance of patch antennas at high frequency band, will decrease the computational cost. The use of NNs in patch antenna design also enables researchers and engineers to explore a broader range of design possibilities and quickly converge to optimal solutions, ultimately leading to improved antenna performance in terms of gain, bandwidth, efficiency, and radiation pattern control. The literature contains an extensive research on patch antenna implementations of NNs. The design and

optimization of patch antenna problems has been solved accurately by two NN-based methods, such as “forward side and reverse side” [22 - 25]. In the forward side, the dimensions of the patch were obtained as a function of thickness, permittivity of substrate, and resonant frequency. In the reverse side, the resonant frequency was obtained by mapping with the dimensions of the patch, thickness and dielectric constant of subsrate. Also, the resonant frequency was calculated for different shapes of patch antennas, such as circular and triangle [26 – 27]. In another research [28], the geometrical parameters of inset fed, affecting the impedance matching, were optimized in addition to the dimensions of the patch. NN-based optimized patch antennas were measured in [29] to indicate how the antennas generated by NN deviate from the measurements. In [30], NN was applied to the design of the truncated circular patch antennas wihch are used in radar applications. In a different research [31], upper and lower resonance were obtained for coaxial-fed stacked-up patch antenna. All these studies have accomplished to obtain accurate models. However, the NNs were trained at resonant frequencies, and a relatively few number of design parameters were considered to correlate them with the antenna performance. As a result, these solutions can be viewed as constrained and limited in their scope.

In [32 – 33], the authors propose novel frameworks for modeling antenna structures using data-driven surrogate modeling approaches. They introduce the concept of domain confinement, which reduces the initial setup cost by defining the model domain using random observables and antenna-relevant knowledge. Both frameworks demonstrate remarkable predictive power, surpassing existing methods in terms of relative rms error. They also provide broader parameter ranges compared to previous studies.

To sum up, NN-based surrogate models are valuable tools for antenna design and optimization. However, training these models to provide a general solution, encompassing the antenna's performance across a wide frequency range and accounting for numerous design parameters, presents significant challenges due to the immense number of profiles and nonlinear relationships involved.

1.2 Motivation

Artificial NNs have shown a high potential for designing and optimizing antennas by utilizing various design parameters such as antenna dimensions, and substrate properties to determine performance parameters. Several research studies have demonstrated the success of this unconventional approach. However, it is important to note that NNs operate by mapping inputs to outputs rather than comprehending the physical principles underlying the antenna structure. This characteristic poses challenges, leading to limited generalizability of the models. Previous studies have successfully applied NNs to specific antenna modeling problems, but they often had constraints on the number of input parameters, their ranges, or the frequency band of the output response. These limitations restrict the applicability of the developed models and simplify the learning process. In this paper, a more general problem by employing a NN modeling approach with a wider range of input parameters is adressed. Specifically, a patch antenna-on-package for 5G applications operating at 28 GHz was determined to design. 11 design parameters of the antenna, which determines the performance of antenna, are selected as input parameters. This is a relatively large range if the past research mentioned before [22-33] is considered, and leads to a generalized model which can predict a lot of different antennas. The NN is aimed to be trained to predict S_{11} parameters of the models as the function of design parameters within the 23-33 GHz frequency band. Since the frequency range at the output is also relatively large, a spectral transposed convolutional neural network (STCNN) was decided to implement. Finally, a knowledge-based methods were implemented into the NN to reach a high accuracy levels for a generalized problem.

1.3 Thesis Outline

The initial chapter of this study delves into the historical background and the motivation behind the rising demand for surrogate models in antenna design.

Chapter 2 thoroughly examines the construction of an STCNN with various aspects including the identification of input and output parameters for the model, generation of the dataset, construction and optimization of the STCNN, and the numerical results.

In the subsequent chapter, Chapter 3, different regularization methods are elucidated in terms of their types and their contributions towards enhancing the performance of the NN.

Lastly, in Chapter 4, the loss functions are augmented with the inclusion of the cavity model of the microstrip antenna, considering the fundamental modes. A knowledge-based NN are generated to predict the S_{11}, the resonant frequency and the fractional bandwidth for the fundamental mode.

CHAPTER 2

BLACK BOX NN MODELING

Black box modeling is a method where the internal workings or mechanisms of a system are not explicitly known or understood. In this modeling technique, the system takes inputs and applies a series of mathematical or computational operations to generate an output, without any knowledge about the internal processes or logic that occur within the box.

The black box model accurately predicts the behavior of a complex system instead of understanding the underlying processes or relationships involved. Therefore, they can be considered as fast decision-maker devices in the existence of adequate data. This property makes them promising in diverse fields, including ML, finance, statistics, and engineering. NNs have emerged as one of the most popular algorithms for constructing black box models.

The demand for stacked-up technologies in the field of electromagnetics is on the rise, driven by the need to combine and integrate different applications. However, optimizing the system for all parts at high frequencies and meeting specific requirements can be challenging. Additionally, even small modifications often require repeated system simulations, which can be computationally intensive. To address this, black box modeling serves as an excellent alternative for creating surrogate models in the mmWave frequency band.

Microstrip antennas, known for their low profiles, are a prominent type of antenna used in stacked-up structures. This research specifically focuses on developing an accurate surrogate model for microstrip patch antennas on package. The chapter provides a detailed explanation of the working principles of NNs, the fundamentals of patch antennas, and presents the specifics and outcomes of the surrogate model.

2.1 Neural Networks

NNs are a type of ML method and serve as the information processing units that identify patterns within the provided information. This feature enables them to acquire human skills such as memorization, learning, and observation. In today's world, they are used in various areas requiring modeling complex behaviors such as facial recognition, medical diagnosis, targeted marketing, financial predictions, and energy demand forecasting. Roughly, they represent a digital model of biological human brain. Therefore, just like human learning is a continuous process that evolves with experiments, the amount and quality of data presented is the most critical part of the functionality of an NN.

2.1.1 Fully Connected Neural Networks

The structure of fully connected NNs (FCNNs) contains an input layer, an output layer, and hidden layers consisting of nodes to represent neurons, as shown in Figure 2.1. In this network, the information flows in a unidirectional manner. Nodes on each layer are connected to the nodes on the next layer with multiplicative weight ($\mathcal{W}$) and additive bias (b) terms. An activation function (σ) introduces nonlinearity to determine the output of each node based on a defined threshold. One of the most common activation functions, which is also used in this thesis, is hyperbolic tangent (*tanh()*), which is plotted in Figure 2.2. Equation (2.1) represents the process of creating predictions from the available input in the forward propagation. Weights and biases are the unknowns of function, and are calculated according to the feedback of the predictions during the training process.

$$a^{k+1} = \sigma(\mathcal{W}a^k + b^k) \tag{2.1}$$

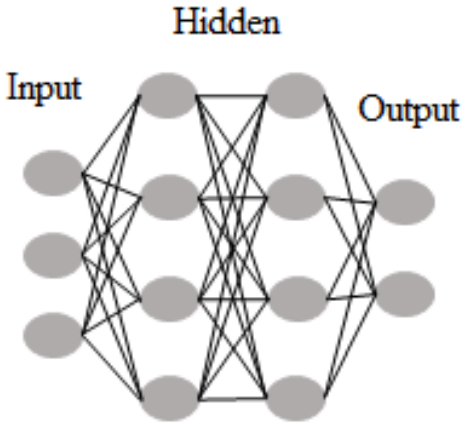


Figure 2.1 Schematic drawing of a fully connected NN.

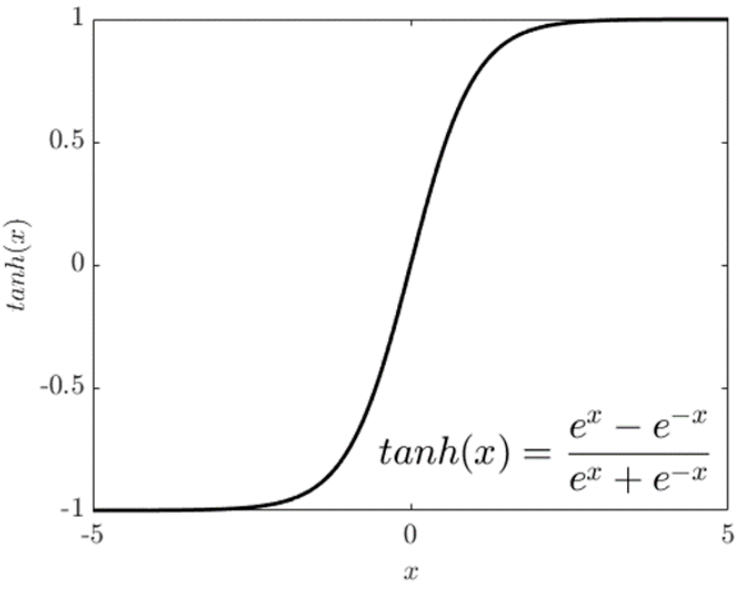


Figure 2.2 The hyperbolic tangent.

Universal approximation theory states that any output can be derived by using linear manipulations and including nonlinear connections from any input in the existence of enough hidden layers, no matter how complex the mathematical relations are between them [34]. Also, it is not necessary to define particular assumptions for

inputs of FCNNs. These concepts make FCNNs broadly applicable to processing information, such as natural language processing and regression problems.

In order to obtain the most precise prediction for the actual output, it is necessary to update the weights and biases. This involves calculating a loss function, which measures the distance between the prediction and the actual output, and striving to minimize this function during the training process. There are a lot of types of loss functions. For instance, they can be represented as the absolute difference or the squared absolute difference between the predicted and actual outputs. For a two dimensional output, the average distance between the predictions and true outputs is obtained by averaging the distance between every output entries over the two dimensions. Subsequently, the average distance is used to update the unknown parameters of the NN by feeding the gradient information to the NN in a backward direction. There are several methods to optimize the learnable parameters of the NN, and the most common one is the gradient descent. The gradient represents the slope of the function in a given direction. The gradient descent algorithm locates the local minima and maxima as being scaled with the learning rate (α), and it can guide to reach the global minimum by making necesary adjustments, as illustrated in Figure 2.3. Consequently, the gradient of the loss function can be used to update the weights and biases in the direction of minimizing loss function ($\mathcal{L}$), as in (2.2) and (2.3). Here, w_{ij}^{k} represents the entries of the weight matrix which connects i^{th} node on the k^{th} layer and j^{th} node on the $(k+1)^{th}$ layer. b_j^k represents the output of j^{th} node on the k^{th} layer for the bias matrix.

$$w_{ij}^{k+1} = w_{ij}^{k} - \alpha \frac{\partial \mathcal{L}}{\partial w_{ij}^{k}} \tag{2.2}$$

$$b_j^{k+1} = b_j^k - \alpha \frac{\partial \mathcal{L}}{\partial b_j^k} \tag{2.3}$$

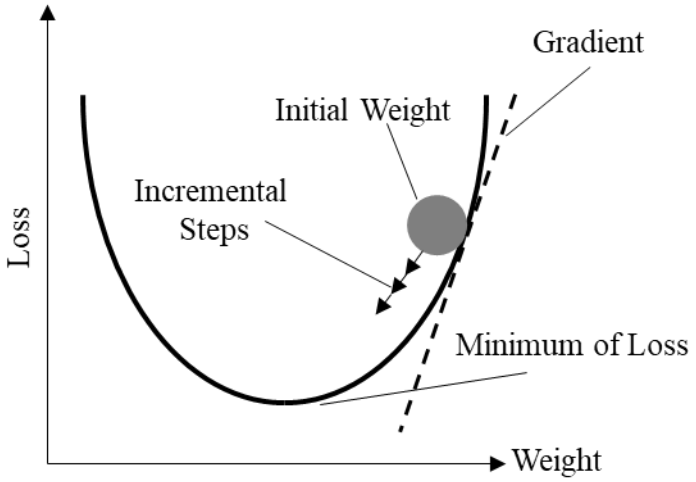


Figure 2.3 Illustration of the gradient descent algorithm.

In this thesis, a modified version of the gradient descent algorithm, which is known as the Adam Optimizer [35], has been utilized. Adam means “adaptive moment estimation”, and it adapts the learning rate of each unknown parameters by using the information of the gradients and momentum. The Adam optimizer uses both the first and second moment which are the mean of the gradients and the uncentered variance of the gradients, respectively. The mean of the gradients helps the optimizer to understand the overall trend of how to update the wieghts and biases to minimize the loss function. On the other hand, the uncentered variance of gradients have the information of how much gradients vary from their mean. It gives a representation of the flatness or curviness of the optimization landscape. For the flat landscape, gradients will be small and close to the mean. For the curvy landscape, gradients will be larger and more spread out. Since the Adam optimizer uses the adaptive learning rate, it can adjust the size of the steps according to the type of the optimization landscape. For example, in areas where the landscape is flat, the optimizer can take larger steps with larger learning rates to reach the minima faster. On the other hand, in the curvy areas, the optimizer uses smaller learning rates, which leads to take smaller steps, to prevent overshooting the minima. As a result, this dynamic technique helps to find the optimal solutions faster.

Another situation, known as overfitting, indicates the performance of the NNs and the generalization of the predictions. A high amount of neurons and layers can cause high computational costs and be prone to overfitting, which is a highly undesirable behavior in ML. To explain the issue of overfitting, two critical terms should be introduced, which are bias and variance. Bias can be defined as the measure of oversimplifying the model. A high bias means that the NN is too simple to learn the input/output relations. On the other hand, the variance can be defined as the sensivity of the model to small fluctuations in the data. When the variance is high, the model overfits, and learns the unnecessary relations or noise intead of capturing the meaningful information. Therefore, if overfitting occurs, the training process fails to generalize well to make accurate predictions. Then, the test loss is much larger than the training loss. An overfitted NN simply memorizes the training data instead of learning the patterns. Consequently, when it encounters with a separate test set, it struggles to predict the accurate output.

2.1.2 Convolutional Neural Networks

Convolutional NNs (CNNs) are a special type of NNs for processing large input features, such as images, videos, and audio spectrograms. This approach has remarkable success for image processing problems like high resolution image classification, object detection, or image segmentation [36-37].

The fundamental components of a CNN are multiple learnable filters, also called as kernels, constructing the convolutional layers. Each filter represents a compact weight matrix that is convolved (element wise multiplication and summation) with local receptive fields of the input data. This operation slides the filter with specified strides across the entire input to look for a similar pattern and downscales the input feature size in accordance with Equation (2.4) for a one-dimensional input. The strides control the step size of the filter, determining the spatial sampling rate of the

input. The computation of the output is illustrated in Figure 2.4. In this case, since the number of strides is equal to 1, the filter, which has the size of 3, is slided by 1 through the input.

$$Output\ Size = \frac{(Input\ Size - Kernel\ size)}{(Number\ of\ Strides)} + 1 \tag{2.4}$$

The convolutional operation exploits filter effectively to capture local spatial patterns in the input data, such as edges, corners, and textures. Processing local spatial patterns instead of processing each input feature enables them to make accurate predictions and classifications for complex and large inputs.

The convolutional layers are generally combined with FCNN layers which aggregate the high-level features and perform the final classification or regression tasks.

The unknown parameters, including the weights and biases of the convolutional layers, are adjusted to minimize the difference between the predicted output and the actual labels by using backpropagation and gradient descent like FCNNs.

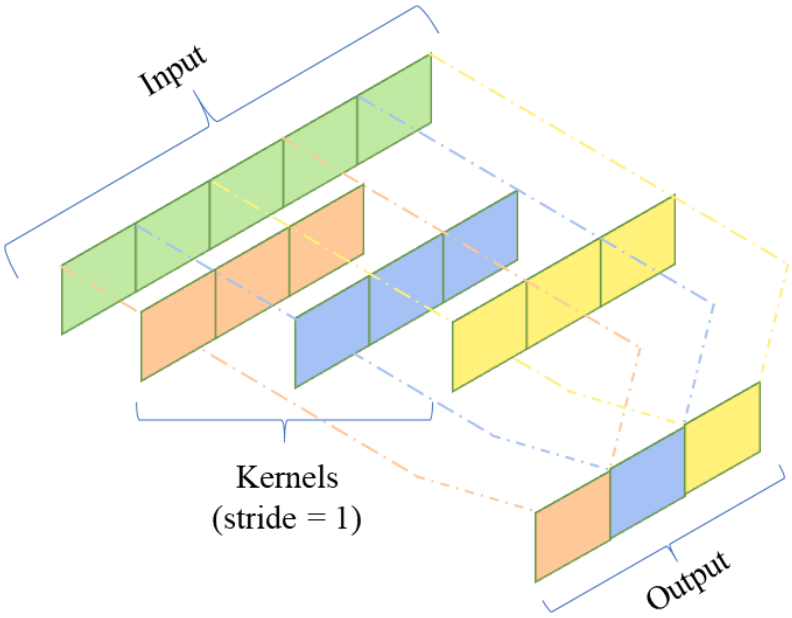


Figure 2.4 The illustration of production of output for CNN.

2.1.3 Transposed Convolutional NNs

Transposed CNNs (TCNNs) are known as deconvolutional networks, and they are a powerful class of deep learning models widely used for tasks such as image segmentation, image super-resolution, generating high-resolution images, and up-sampling [38-39]. They work in the opposite direction of convolutional layers. They are specifically designed to generate high-resolution outputs from lower-resolution inputs.

The key mechanism employed by TCNNs is the transposed convolution, which utilizes filters to expand the spatial dimensions of the input. This expansion is achieved by inserting zeros between the elements and then convolving the modified input with a set of learnable filters. This process facilitates up-sampling, allowing the network to reconstruct higher-dimensional outputs from lower-dimensional inputs. On the other hand, STCNNs as a special type of TCNNs operate up-sampling in the frequency or spectral domain instead of spatial domain. These networks can be employed to expand the spectral information of the input function.

During the training phase, the network learns the unknown parameters by optimizing them with backpropagation and gradient descent of the loss function as in the CNNs and FCNNs.

Transposed convolutional operations play a crucial role in reconstructing high-dimensional outputs by employing Equation (2.5) for one dimensional input. Figure 2.5 shows how to generate an upsampled output.

$$Output\ Size = (Input\ Size - 1)\ x\ (Number\ of\ Strides) + (Kernel\ size) \tag{2.5}$$

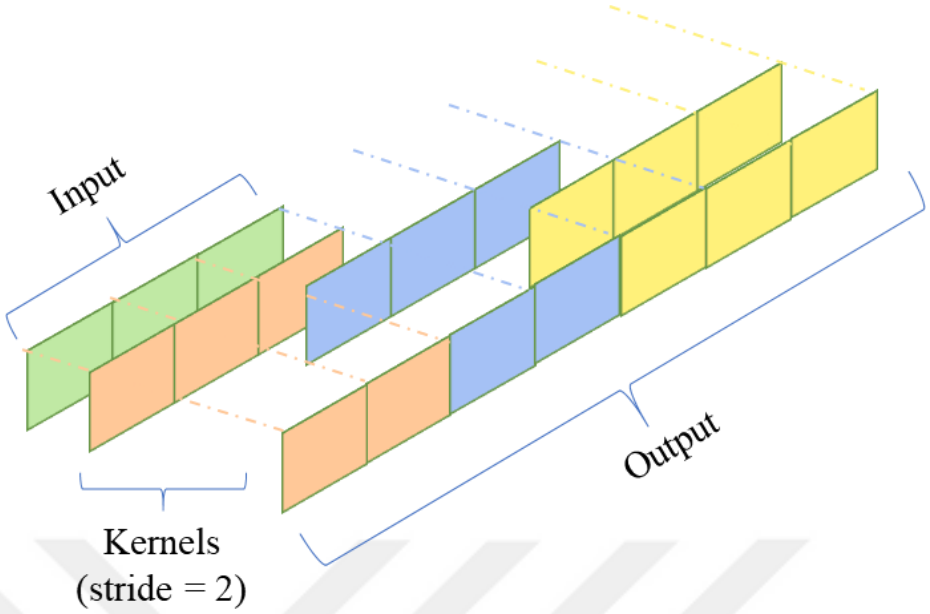


Figure 2.5 The illustration of production of output for TCNN.

2.2 Antenna Modeling

Antennas play a crucial role in the performance of modern radio frequency (RF) communication systems, particularly in the realm of 5G applications. The advent of 5G has necessitated the utilization of millimeter waves (mmWave) to meet the demand for higher data rates, larger channel bandwidths, and reduced latency [40]. MmWave frequencies operate in the range of 3 to 300 GHz, and cover a vast amount of spectrum [41].

Antennas serve as the vital link between the system and the surrounding environment, which acts as the wireless communication channel. In the ever-growing wireless technology landscape, the endeavor of antenna design has become progressively more challenging. Wireless communication systems demand antennas to be compact, lightweight, high gain, efficiencient, low profile, and reconfigurable. Morevover, the propagation of mmWave signals can be easily affected by atmospheric attenuation and urban obstacles like buildings [42]. Thus, reducing loss

is critical to optimize the signal tranmission. In this regard, packaging technologies help to provide effective signal transmission for mmWave applications, and facilitate the integration of essential components, such as antennas, transceivers, and sensors on the same platform. Additionally, packaging technologies contribute to the miniaturization of structures [43]. Antenna-in-package (AiP) and Antenna-on-package (AoP) technologies involve integrating one or more antennas with a transceiver die and other peripherals in a single device.

The choice of antenna in AiP/AoP technologies is typically based on popular designs that are easily adaptable for specific applications. Package selection is governed by considerations such as automatic assembly feasibility. Material and process choices involve tradeoffs to meet requirements such as electrical performance, thermal-mechanical reliability, compactness, manufacturability, and cost. In this regard, a lot of research has been completed to design and implement antennas in/on package [44 – 49]. Microstrip patch antennas are the most prominent types for packaging applications due to their low profiles, light weights, and low costs.

Microstrip antennas are commonly utilized in stacked-up packaging configurations. Patch antennas, a type of microstrip antenna, are composed of a metallic patch or a plate mounted on a dielectric substrate. Patch antennas can be designed in various shapes, including rectangular, circular, triangle, or equilateral, catering to different design requirements. They feature ease of installation, and integration into different structures, such as printed circuit boards, or integrated circuits [50]. Furthermore, patch antennas can be operated over a wide frequency range, and this makes them applicable to a range of wireless communication systems, within the mmWave spectrum (3–300 GHz). 28 GHz band is particularly favorable due to its lower atmospheric absorption [42] and compatibility with multiband operation [51]. In the light of all mentioned, a rectangular patch antenna geometry, operating in the 28 GHz band, for packaging applications was used in this thesis.

2.2.1 Problem Geometry

The top and cross section views of modeled antenna are illustrated in Figure 2.6. The patch is printed on the top conductive layer of the package. The substrate, where the patch is mounted on, is composed of a dielectric material, and the choice of material affects antenna perfomance such as the antenna's radiation pattern, impedance matching, bandwidth, and efficiency. Additionally, a solder resist layer is applied on the top metal layer, as commonly seen in packaging applications. A metal probe is used for excitation, and is positioned on the second conductive layer, between the probe and the ground plane. The model also incorporates a pad, which serves as the conductor connecting the excitation port to the probe, and an anti-pad, representing the gap between the pad and the ground plane. All conductive components are assigned as copper. The model does not include a feed network. This provides to enhance the flexibility in choosing the feeding method. The model just contains a metalic probe which is excited with a coaxial port located on the ground plane. This allows propercascadeding with different types of feeding networks, such as a coaxial probe or a stripline, according to design criteria. This flexibility of feeding technique is advantegeous for specially stacked-up structures because there is no need to design and optimize the parameters of the model again when the feeding technique is modified. The feeding part can be added separately and indepedent of the antenna model.

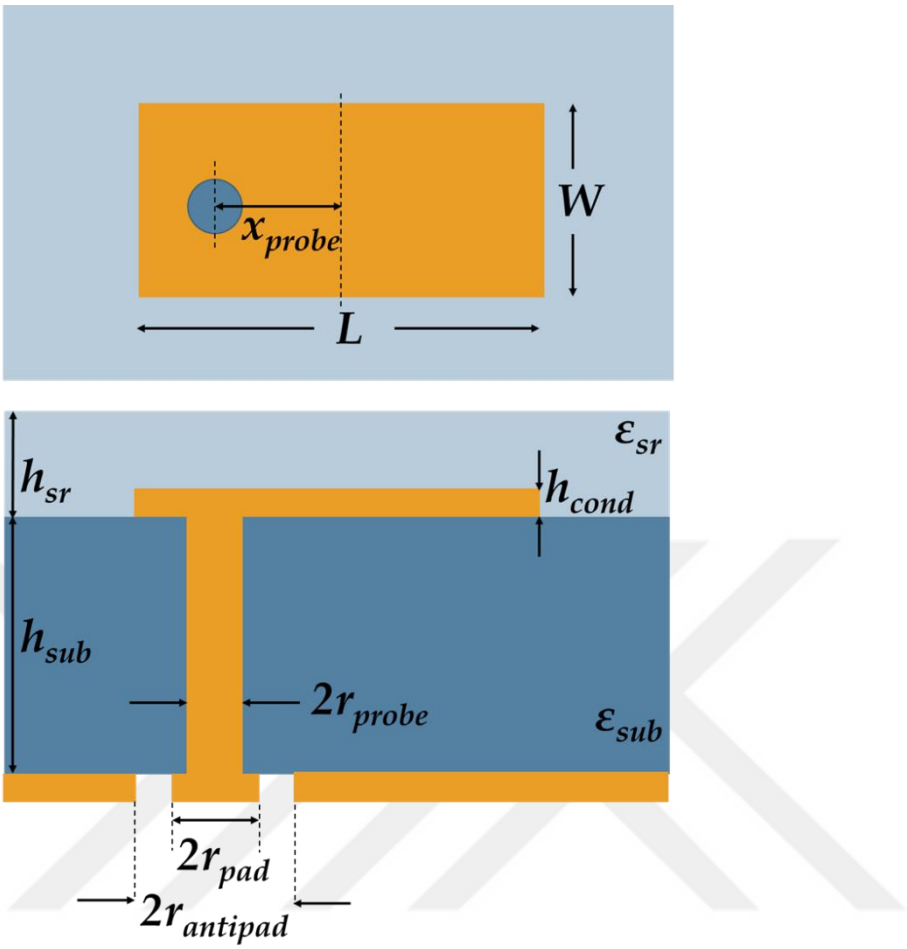


Figure 2.6 Top and cross section views of the modeled patch antenna.

This thesis utilizes an NN-based surrogate model to represent the performance of antennas as a function of design parameters. The S_{11} parameters are identified as the performance metrics and are the network's output, while 11 independent design parameters are defined as inputs. These design parameters encompass the geometric and material properties of the patch, probe, and stack-up. The broad ranges of input parameters, as presented in Table 2.1, allow reasonable antenna designs within a relatively extensive design space. The boundaries of the ranges were determined by considering the limits of the intended operating frequency band (23- 33 GHz).

Table 2.1 The range of design parameters.

Input Parameter	Range
L (mm)	1.800 – 5.200
W (mm)	1.800 – 5.200
h_{cond} (mm)	0.010 – 0.040
h_{sub} (mm)	0.100 – 0.800
h_{sr} (mm)	0.020 – 0.100
r_{probe} (mm)	0.015 – 0.050
r_{pad} - r_{probe} (mm)	0.000 – 0.025
$r_{antipad}$ - r_{pad} (mm)	0.025 – 0.100
x_{probe} / L	0.050 – 0.450
ε_{sub}	2 - 5
ε_{sr}	2 - 5

2.2.2 Data Generation

To obtain the dataset, a total of 5000 distinct antennas were generated using Ansys HFSS [52]. The selection of these antennas was carried out by utilizing Latin Hypercube Sampling (LHS) [53], an advanced statistical technique. This method enables the generation of samples that are randomly distributed within specified parameter ranges, as outlined in Table 2.1. LHS ensures that the samples cover the entire range of parameter values comprehensively. The histogram of some input parameters, as illustrated in Figure 2.7, confirms the almost uniform distribution of samples within their designated ranges. The input parameters that have a significant influence on antenna performance, such as L, W, h_{sub}, x_{probe} / L and ε_{sub}, have been selected to illustrate the uniform random distribution in Figure 2.7, but other parameters also exhibit the similar distribution. Moreover, Table 2.2 indicates the maximum and minimum values of the generated input parameters, rounded to the

third decimal digit, to verify that the regions close to the boundaries of the hyperrectangle are also sampled. The ensemble of input parameters, featuring their corresponding distributions, collectively forms the input dataset.

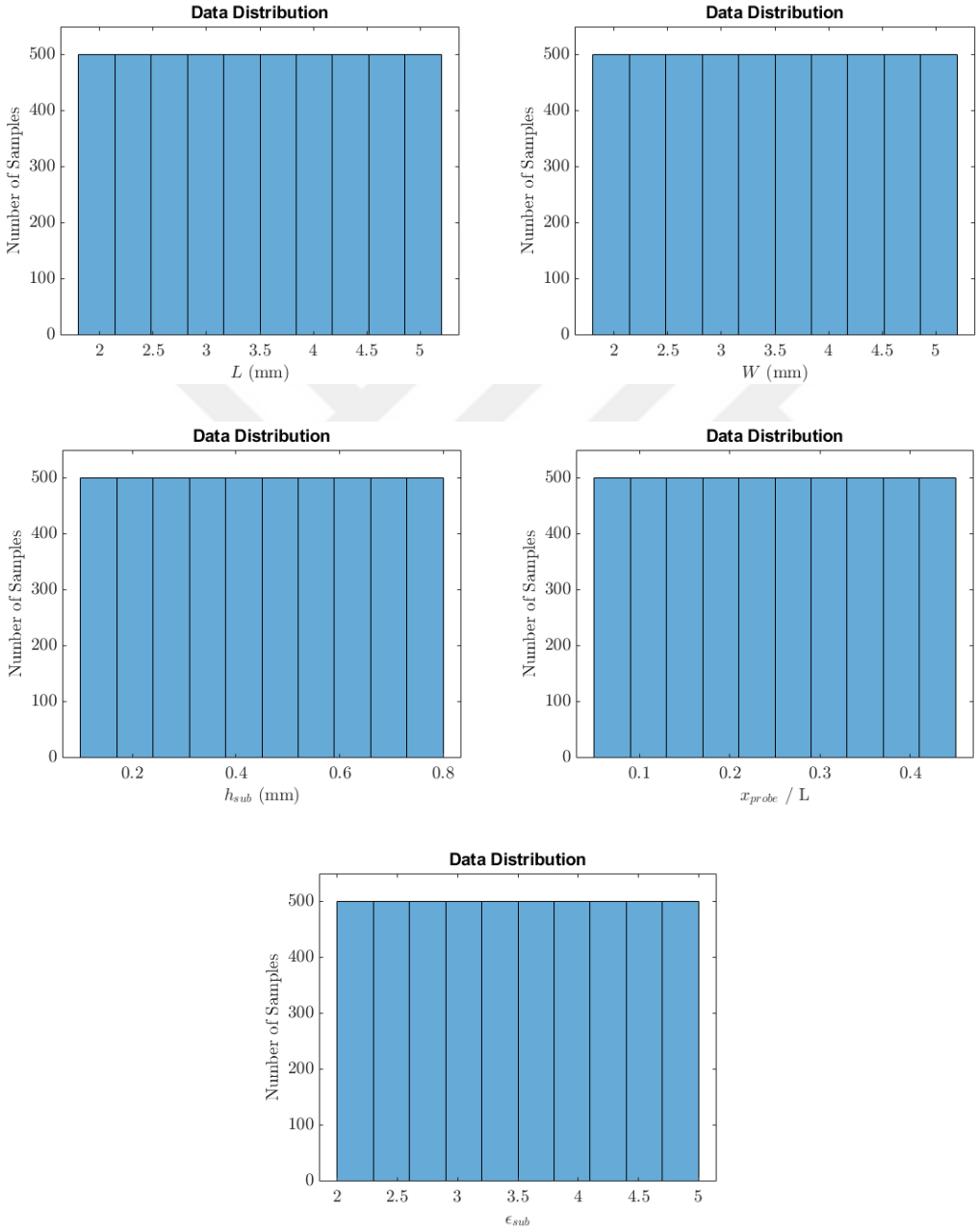


Figure 2.7 The illustration of data distribution of the most critical input parameters with the histogram.

Table 2.2 Minimum and maximum values of input parameters.

Input Parameter	Min	Max
L (mm)	1.800	5.200
W (mm)	1.801	5.200
h_{cond} (mm)	0.010	0.040
h_{sub} (mm)	0.100	0.800
h_{sr} (mm)	0.020	0.100
r_{probe} (mm)	0.015	0.050
r_{pad} - r_{probe} (mm)	0.000	0.025
$r_{antipad}$ - r_{pad} (mm)	0.025	0.100
x_{probe} / L	0.050	0.450
ε_{sub}	2	5
ε_{sr}	2	5

After the generation of the sample points, these antennas were simulated within the frequency band of 23-33 GHz using Ansys HFSS. Complex S_{11} values, which were collected at 201 frequency points to comprehensively evaluate antenna performance, become the output dataset.

2.3 NN Architecture

The NN structure comprises three subparts: an FCNN, an STCNN, and a learnable smoothing filter. Figure 2.8 illustrates the cascaded arrangement of these parts. The hyperparameters were carefully optimized to determine the most appropriate weight and bias matrices.

The FCNN component consists of three hidden layers, each containing 60 nodes. These nodes map the input parameters to a latent space with 60 dimensions. The output of the FCNN is flattened and then upscaled to 201 points using STCNN layers. The STCNN layers employ kernel sizes and strides of 21, 7, 3, and 1, 3, 3,

respectively. To achieve a smoother output, the STCNN output is further processed using a Gaussian filter. The Gaussian filter adds another convolutional layer to the output of tranposed convolutional layers. Opposed to conventional convolutional layers, in this case, the shape of the filter is fixed to be a Gaussian function and only the size and the standard deviation of the function is modified. The output tensor is padded symmetrically according to the filter size. Figure 2.9 is a representation of the gaussian filter. The size of filter The standard deviation is a learnable parameter through the training process, and the filter size is a hyperparameter which is optimized as 51. The NN was trained during the 3200 epoch. An epoch refers to a complete pass through the entire training dataset during the training process. During each epoch, the NN forces to minimize the loss function by updating the unknown parameters. To perform the search for minima, Adam optimizer is utilized with an initial learning rate of 0.007. The learning rate is reduced with a decay rate of 0.7 applied at every 1000 epochs.

Since the output was defined as the linear scaled S_{11} parameters of the antenna, the data was distributed between -1 and 1. Therefore, the input parameters were normalized between -1 and 1 before entering the NN. To introduce nonlinear relations, the hyperbolic tangent (*tanh()*) function is chosen as the activation function according to the range of dataset.

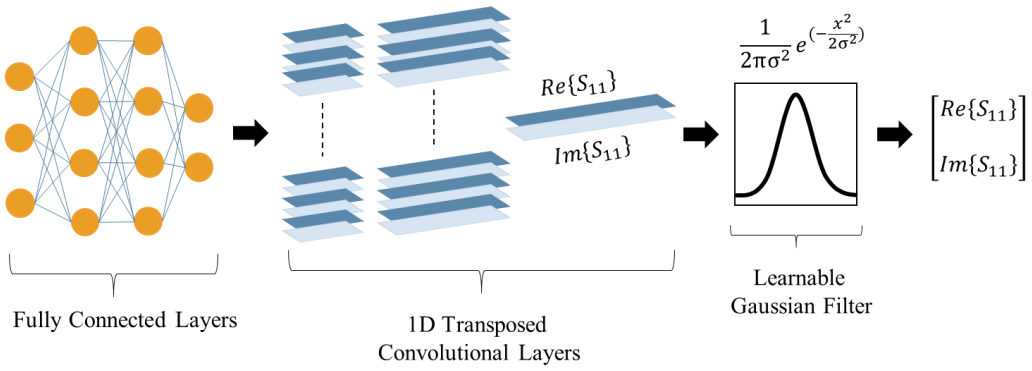


Figure 2.8 The architecture of cascaded NN.

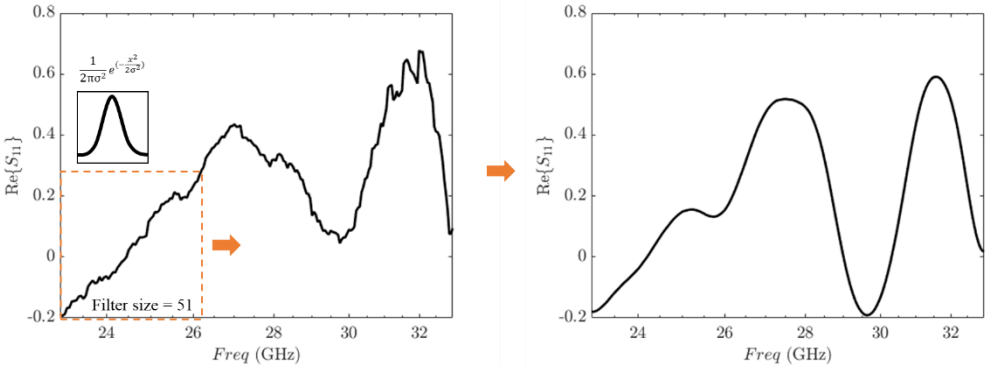


Figure 2.9 The illustration of gaussian filter.

2.3.1 The Determination of Loss Function

Loss functions are critical in NNs as they act as the objective or optimization function that directs the learning process. They quantify the disparity between the predicted and actual outputs, and the NN adjusts its parameters to minimize the loss function.

There are various types of loss functions available, and selecting the appropriate one is crucial to establish a suitable guide for optimizing predictions. In this thesis, seven distinct loss functions have been examined: Mean Absolute Error (MAE), Mean Squared Error (MSE), Huber loss, Relative Absolute Error (RAE), Relative Squared Error (RSE), Root Mean Square Error (RMSE), and Relative Root Mean Square (RRMSE). Equations (2.6) – (2.12) outline the calculation of these specific loss functions [54] where y and $\hat{y}$ represents the actual and predicted output, respectively. N_s and N_f denote the number of rows and columns in the output matrix, respectively. In this context, the rows correspond to the number of antenna samples, and the columns signify the frequency points spanning the entire band.

$$\text{Mean Absolute Error (MAE)} = \frac{1}{N_s}\frac{1}{N_f}\sum_{s=1}^{N_s}\sum_{f=1}^{N_f}|y-\hat{y}| \qquad (2.6)$$

$$\text{Mean Squared Error (MSE)} = \frac{1}{N_s}\frac{1}{N_f}\sum_{s=1}^{N_s}\sum_{f=1}^{N_f}(y-\hat{y})^2 \tag{2.7}$$

$$\text{Huber Loss } = \begin{cases} \frac{1}{N_s}\frac{1}{N_f}\sum_{s=1}^{N_s}\sum_{f=1}^{N_f}\frac{0.5(y-\hat{y})^2}{\beta} \; if \; |y-\hat{y}| < \beta \\ \frac{1}{N_s}\frac{1}{N_f}\sum_{s=1}^{N_s}\sum_{f=1}^{N_f}|y-\hat{y}| - 0.5 \times \beta \; otherwise \end{cases} \tag{2.8}$$

$$\text{Relative Absolute Error (RAE)} = \frac{1}{N_s}\sum_{s=1}^{N_s}\frac{\sum_{f=1}^{N_f}|y-\hat{y}|}{\sum_{f=1}^{N_f}\left|y-\frac{1}{N_f}\sum_{f=1}^{N_f}y\right|} \tag{2.9}$$

$$\text{Relative Squared Error (RSE)} = \frac{1}{N_s}\sum_{s=1}^{N_s}\frac{\sum_{f=1}^{N_f}(y-\hat{y})^2}{\sum_{f=1}^{N_f}(y-\frac{1}{N_f}\sum_{f=1}^{N_f}y)^2} \tag{2.10}$$

$$\text{Root Mean Squared Error (RMSE)} = \sqrt{\frac{1}{N_s}\frac{1}{N_f}\sum_{s=1}^{N_s}\sum_{f=1}^{N_f}(y-\hat{y})^2} \tag{2.11}$$

$$\text{Relative Root Mean Square (RRMSE)} = \frac{1}{N_s}\sum_{s=1}^{N_s}\sqrt{\frac{\sum_{f=1}^{N_f}(y-\hat{y})^2}{\sum_{f=1}^{N_f}(y)^2}} \tag{2.12}$$

These loss functions can be categorized as L1-based functions, which measure the absolute distance between the predicted and actual output, such as MAE and RAE,

and L2-based functions, that measures the squared distance between the predicted and actual output, such as MSE, RSE, RMSE and RRMSE. Also, as an alternative, Huber loss is defined as a combined form of L1- and L2-based loss functions.

MAE provides a straightforward measure of distance, and it gives more accurate predictions for the regression problems [54]. Meanwhile, the MSE scales the error using squared values, resulting in smoother gradients near zero, compared to MAE. Huber loss combines MSE for small errors and MAE for larger errors, offering a balanced approach. Relative errors, on the other hand, involve normalization. In particular, RAE and RSE, are measures of how better the predictions are than the mean of the actual output.

The loss function and the evaluation metric to assess the "goodness of fit" of the predictions can be chosen separately. The evaluation metric measures how good the predictions are, essentially gauging the resemblance between the predictions and the true output. The loss function is used to update the gradients and optimizer. It is crucial to select a loss function that ensures reaching the highest possible accuracy at the fastest rate. In this part, the loss functions were implemented on the NN and the selection of the evaluation metric and the loss function was carried out. During the training process, 3200 epochs, the NN was initialized with the identical weight matrices for all trials, in order to reduce the randomness.

Figure 2.10 helps to determine the evaluation metric by comparing two predictions, Prediction I and Prediction II, with the HFSS results. These predictions are generated by updating with two different loss functions. Among them, Prediction II (depicted in green) exhibits a better agreement with the true output within the circled regions compared to Prediction I. These regions contain the points around the resonance, where $\mathrm{Re}\{S_{11}\}$ approaches zero. Table 2.3 presents the performance Prediction I and Prediction II in terms of different evaluation metrics. The RSE loss is more sensitive to the differences between the predicitons. Given that RSE includes normalization with respect to the variance of the true output, it magnifies the shifts along the

frequency axis. Therefore, as seen in the Table 2.3, it can measure the proximity between the predicted and actual data, specially around the resonance. According to these facts, the RSE loss was chosen as the evaluation metric.

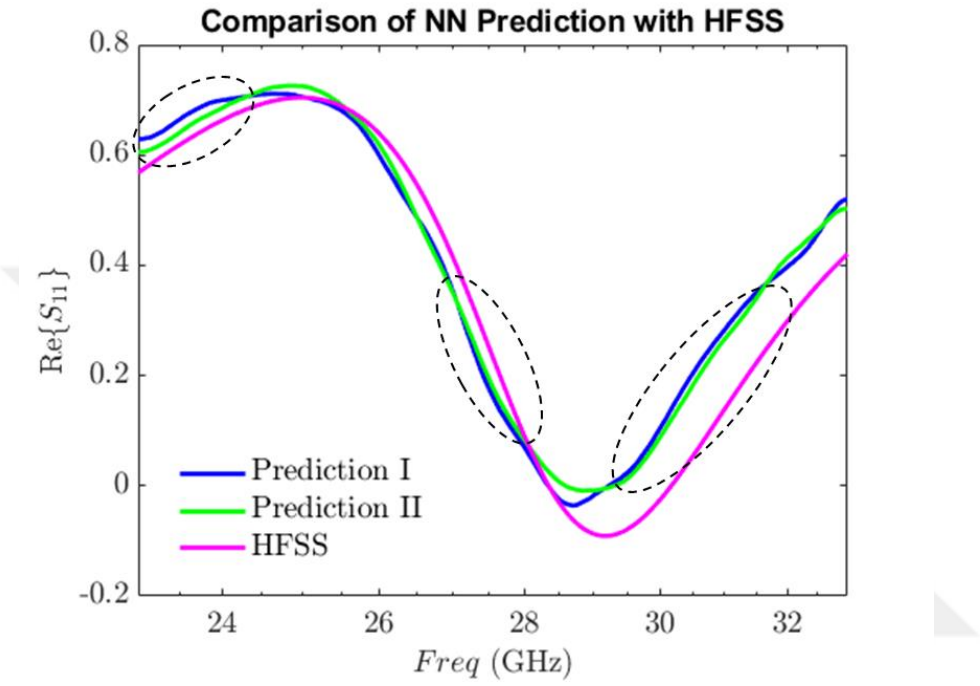


Figure 2.10 The demonstration of evaluation performance of RSE for two predictions which are updated with MAE and MSE.

Table 2.3 Comparison of Prediction I and II using different evaluation metrics.

Loss Update	RSE	MSE	MAE
Prediction I	0.08214	0.00642	0.06774
Prediction II	0.08697	0.00680	0.06843

After determining the evaluation metric, the update metric was defined according to the implementations of different loss functions. Figure 2.11 demonsrates the RSE values of different loss functions for each epoch. Huber loss has the fastest decay

and reaches the lowest value at the last epoch compared to the others. Moreover, Table 2.4 gives the quantiles of the loss functions in terms of RSE. Quantile denotes the percentage of data points that fall below a given value. For example, 0.9 quantile represents the line where the loss of 90% of the data is lower. According to Table 2.4, the lowest 0.9 and 0.75 quantile belong to the Huber loss. Hence, Huber loss improves the performance of relatively worse predictions resulting in a lower mean compared to the other loss functions. Therefore, Huber loss was chosen as the loss functionto update the weights. The value of β was optimized like other hyperparameters of the NN, and was chosen as 0.1.

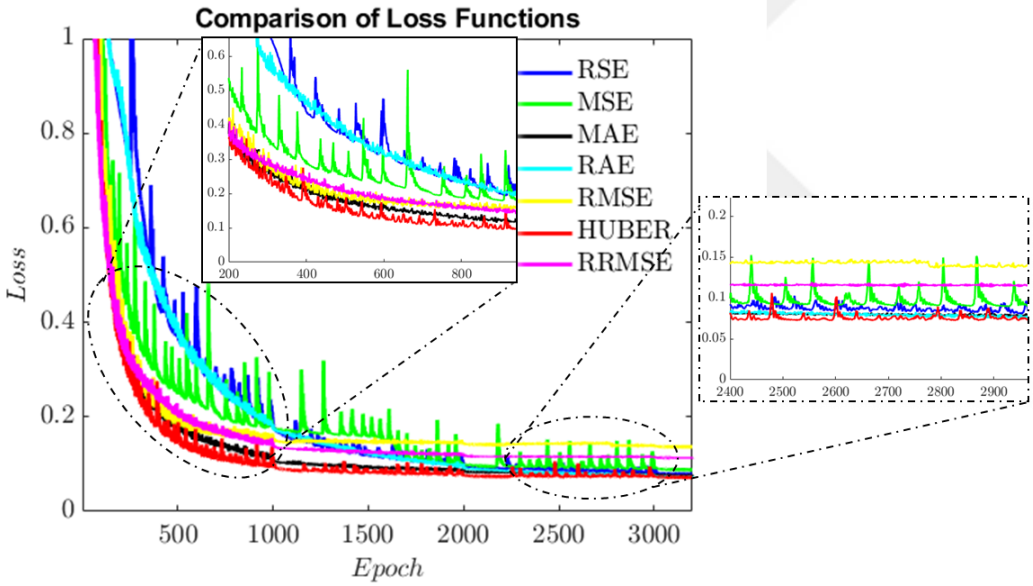


Figure 2.11 The comparison of different loss functions.

Table 2.4 The comparison of quantiles of different loss functions in terms of RSE.

	Q-0.9	**Q-0.75**	**Q-0.25**	**Q-0.1**
MAE	0.16745	0.06789	0.01125	0.00494
MSE	0.16255	0.07087	0.01063	0.00533
HUBER	0.14441	0.06665	0.01205	0.00510
RAE	0.16285	0.06975	0.01083	0.00532
RSE	0.17693	0.07732	0.01575	0.00824
RMSE	0.20894	0.08547	0.01416	0.00666
RRMSE	0.19467	0.08064	0.01352	0.00567

2.4 Numerical Results

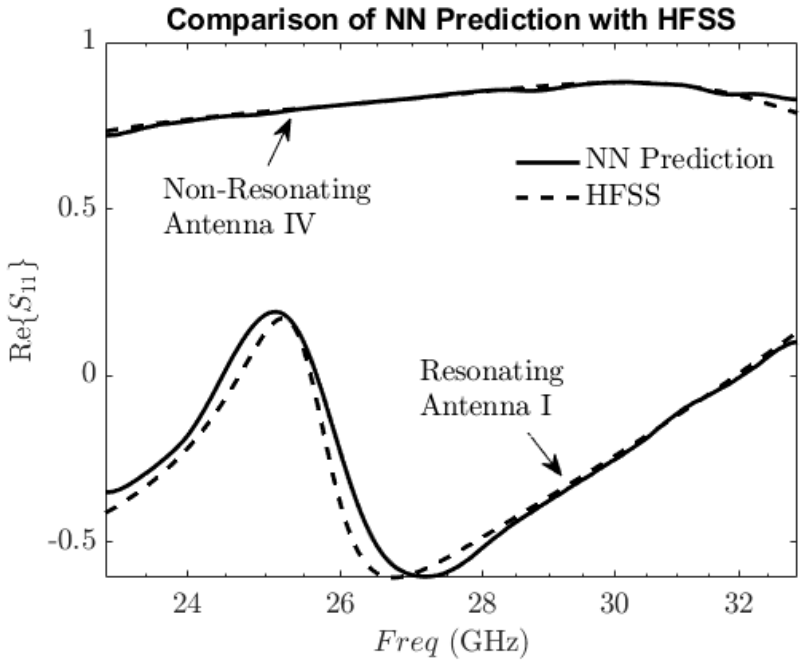
Owing to the broad ranges of the input parameters and output frequency spectra, along with distinct differences in S_{11} profiles between resonating and non-resonating geometries, the learning process becomes exceptionally challenging. Achieving a comprehensive understanding of the system behavior across the extensive solution space necessitates a large number of samples. Hence, 5000 samples were generated to create the dataset. 4000 samples were leveraged as the train set, and 1000 samples were used for the test. Since LHS creates random distribution between top and bottom limits for each input parameter, the samples exhibit both resonating and non-resonating geometries. The data collection took around 4 days on an Intel(R) Xeon(R) E-2136 CPU machine with 32 GB RAM

The training process was conducted over 3200 epochs, taking approximately 5 minutes to complete with the usage of NVIDIA Quadro P2000 GPU acceleration. The RSE of the test set reached its minimum average value of 0.052 at the last epoch. Table 2.5 indicates the best, average and worst losses in terms of RSE for linear scaled S_{11} parameters. To illustrate these numerical losses, Figure 2.12 demonsrates the NN predictions with average losses for the S_{11} parameters of the resonating and non-resonating profiles. In this figure, the real part (a), imaginary part (b), and magnitude (in dB scale) (c) of S_{11} are demonsrated for the predictions which have

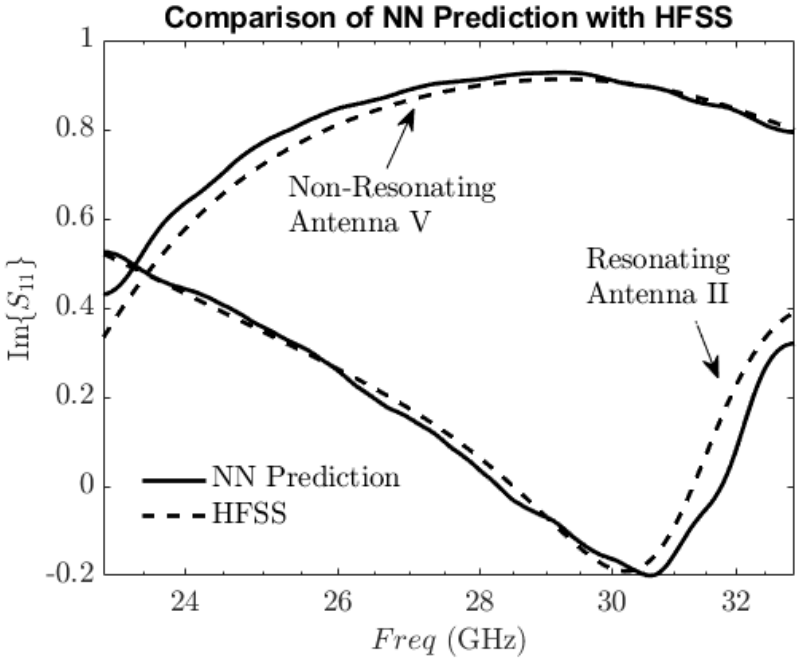
around 0.05 RSE losses. In these comparions, Antennas I-III, and Antennas IV-VI, exhibit resonating and non-resonating profiles, respectively. Moreover, the worst predictions for real and imaginary parts of S_{11}, which belong to Antennas VII and VIII, are also shown in Figure 2.13. The input parameters of each antenna sample are listed in Tables 2.6-2.8.

Table 2.5 Return loss prediction error in terms of RSE for the test set.

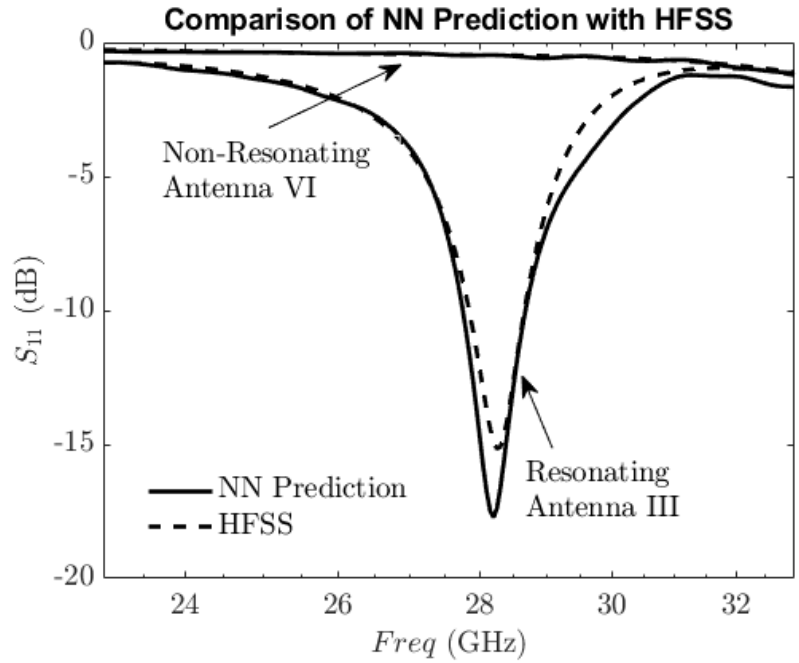
	Best	**Average**	**Worst**
Re{S11}	0.00047	0.05108	1.39900
Im{S11}	0.00034	0.05251	1.35991



(a) Real part predictions with average loss.

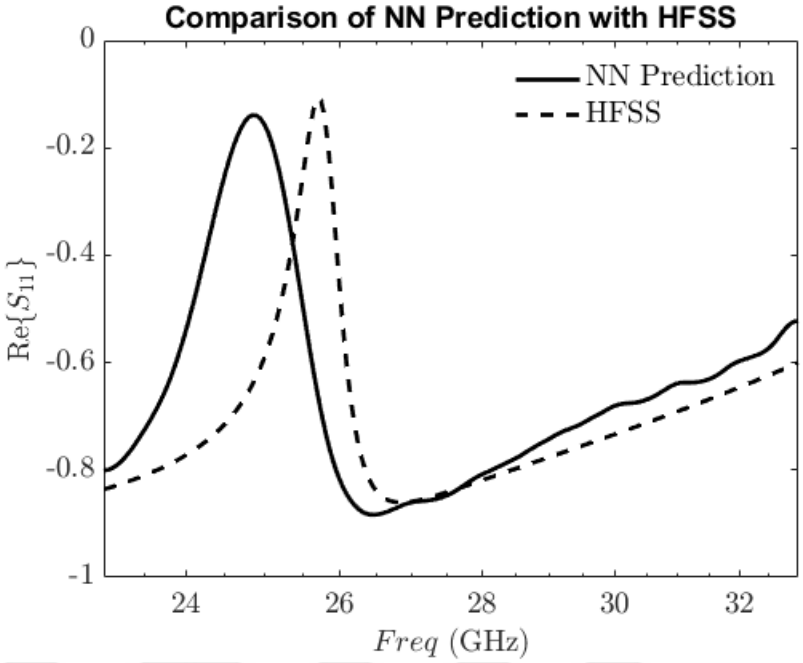


(b) Imaginary part predictions with average loss.

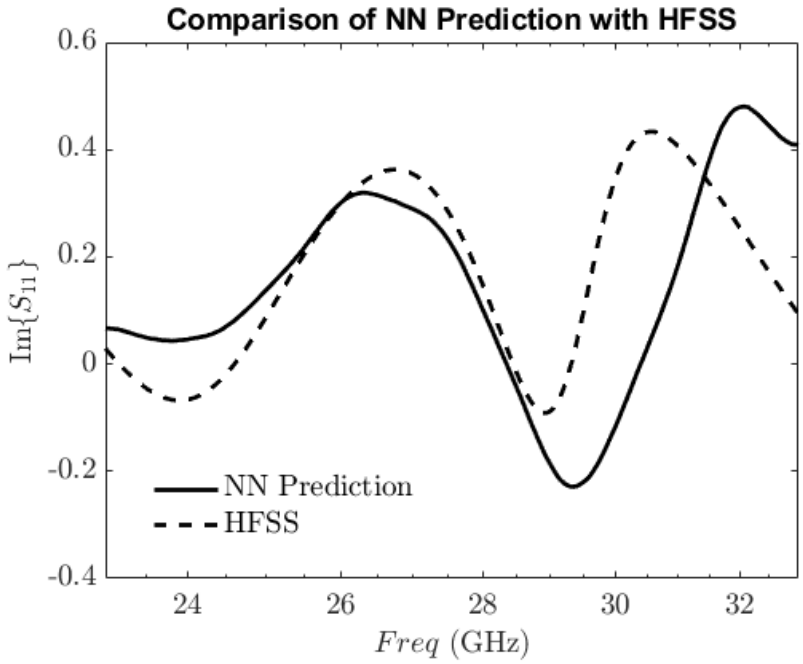


(c) S_{11} predictions with average loss.

Figure 2.12 Comparison of resonating and non-resonating profiles.



(a) Real part predictions of Antenna VII with the worst loss, RSE = 1.39900.



(b) Imaginary part predictions of Antenna VIII with worst loss, RSE = 1.35991.

Figure 2.13 The illustration of the predictions with the worst loss.

Table 2.6 The input parameters for the predictions with average losses for the resonating profiles.

Input Parameter	Antenna I	Antenna II	Antenna III
L (mm)	4.538	2.318	3.606
W (mm)	4.569	3.007	4.719
h_{cond} (mm)	0.019	0.028	0.015
h_{sub} (mm)	0.688	0.295	0.603
h_{sr} (mm)	0.047	0.091	0.053
r_{probe} (mm)	0.022	0.048	0.025
r_{pad} - r_{probe} (mm)	0.000	0.021	0.022
$r_{antipad}$ - r_{pad} (mm)	0.088	0.038	0.084
x_{probe} / L	0.065	0.145	0.345
ε_{sub}	3.556	4.806	4.172
ε_{sr}	2.440	2.705	4.241

Table 2.7 The input parameters for the predictions with average losses for the non-resonating profiles.

Input Parameter	Antenna IV	Antenna V	Antenna VI
L (mm)	4.463	3.960	1.926
W (mm)	2.069	4.455	2.691
h_{cond} (mm)	0.031	0.039	0.013
h_{sub} (mm)	0.542	0.758	0.516
h_{sr} (mm)	0.087	0.090	0.061
r_{probe} (mm)	0.046	0.016	0.0313
r_{pad} - r_{probe} (mm)	0.004	0.018	0.003
$r_{antipad}$ - r_{pad} (mm)	0.050	0.069	0.058
x_{probe} / L	0.275	0.075	0.202
ε_{sub}	2.300	2.289	2.515
ε_{sr}	4.812	3.951	2.682

Table 2.8 The input parameters for the predictions with the worst losses.

Input Parameter	**Antenna VII**	**Antenna VIII**
L (mm)	3.485	4.857
W (mm)	1.840	4.923
h_{cond} (mm)	0.030	0.015
h_{sub} (mm)	0.194	0.705
h_{sr} (mm)	0.052	0.076
r_{probe} (mm)	0.037	0.026
r_{pad} - r_{probe} (mm)	0.023	0.019
$r_{antipad}$ - r_{pad} (mm)	0.034	0.036
x_{probe} / L	0.068	0.205
ε_{sub}	2.284	4.458
ε_{sr}	3.263	4.945

Although most of the predictions achieve a good performance, some less accurate predictions may also occur. Most of such cases have rare profiles which lead to a decrease in the performance of the NN. Figure 2.14 illustrates such a profile, which has two very close peaks with noticably narrow widths, for Antenna IX. The RSE loss for this sample is 0.59795 and its input parameters are presented in Table 2.9.

Moreover, the NN achieved to predict the antennas with multiple resonances. As seen in Figures 2.15 and 2.16, the S_{11} profiles of antennas with double or triple resonances were predicted with, RSE losses of 0.01532 and 0.30251. These samples are labeled as Antenna X and Antenna XI, and Table 2.10 presents their input parameters.

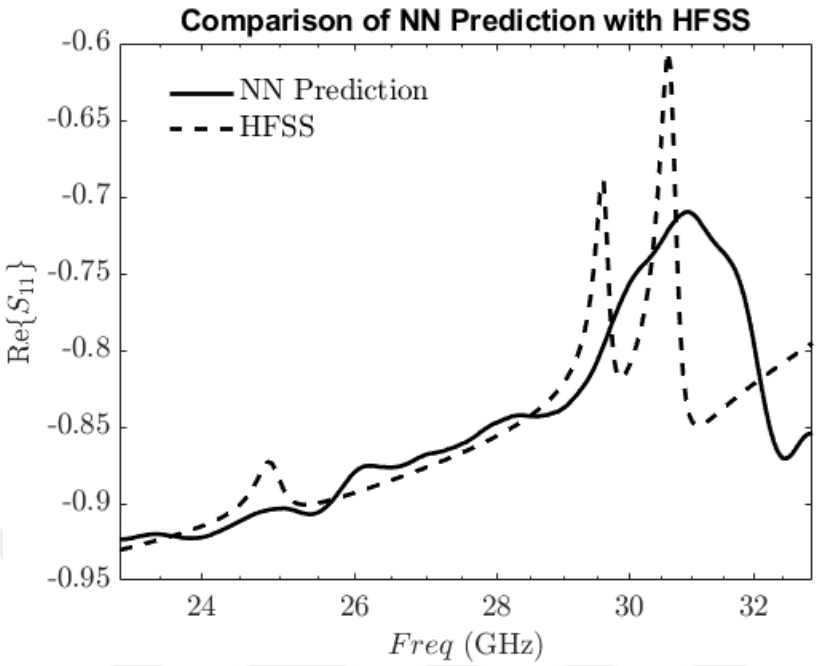


Figure 2.14 The illustration of Antenna IX with low performance, RSE = 0.59795.

Table 2.9 The input parameters of Antenna IX.

Input Parameter	**Antenna IX**
L (mm)	4.523
W (mm)	2.618
h_{cond} (mm)	0.021
h_{sub} (mm)	0.102
h_{sr} (mm)	0.082
r_{probe} (mm)	0.017
r_{pad} - r_{probe} (mm)	0.019
$r_{antipad}$ - r_{pad} (mm)	0.092
x_{probe} / L	0.235
ε_{sub}	4.113
ε_{sr}	2.364

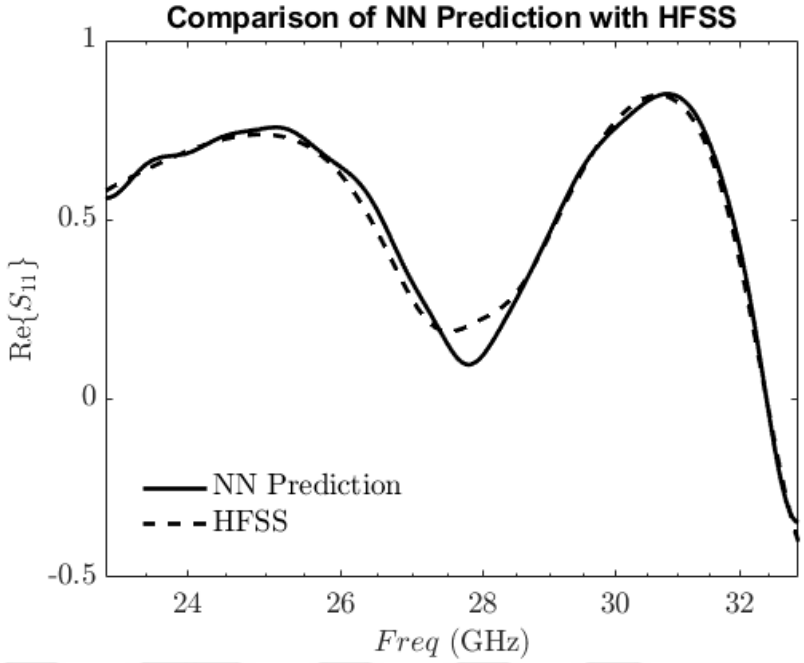


Figure 2.15 The NN prediction of Antenna X which is double resonated antenna, RSE = 0.01532.

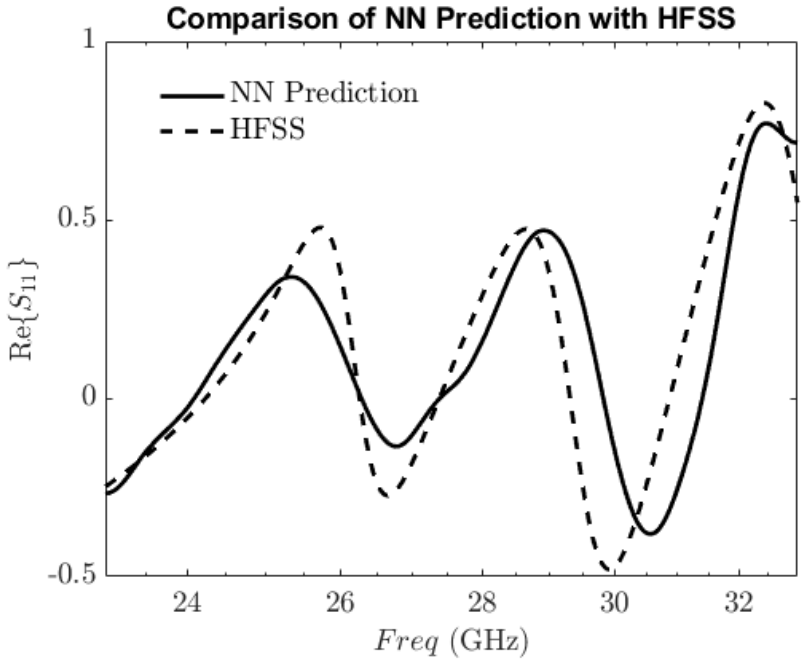


Figure 2.16 The NN prediction of Antenna XI which is triple resonated antenna, RSE = 0.30251.

Table 2.10 The input parameters of the antennas with multiple resonances.

Input Parameter	**Antenna X**	**Antenna XI**
L (mm)	4.427	5.151
W (mm)	4.781	4.567
h_{cond} (mm)	0.028	0.021
h_{sub} (mm)	0.539	0.260
h_{sr} (mm)	0.035	0.049
r_{probe} (mm)	0.033	0.028
r_{pad} - r_{probe} (mm)	0.007	0.015
$r_{antipad}$ - r_{pad} (mm)	0.061	0.074
x_{probe} / L	0.344	0.412
ε_{sub}	4.484	4.481
ε_{sr}	2.667	2.686

The existence of resonating and non-resonating data causes that the reported performance values encompass all types of geometries. However, this can be misleading because non-resonating geometries often exhibit a flat S_{11} response, making them relatively easier to predict. The test set gives an idea how predictions are good for unseen data but it is important to note that the test data is not so different from the training data. Therefore, it is difficult to forecast the performance for an ill-conditioned problem. To understand the generalization of the NN completely, the uncertainty qualification should be applied by evaluating the confidenece levels. However, it is out of hte scope for this work. To specifically evaluate the performance of the NN on resonating geometries, a smaller test set consisting of 20 resonating antennas was manually generated.

These antennas were designed to resonate at 0.5 GHz intervals from 23 GHz to 32.5 GHz, within the predefined parameter ranges. Table 2.11 provides the mean values of the best, average, and worst errors for the predicted real and imaginary parts.

Figure 2.17 shows the comparison of the predicted and actual S_{11} in dB scale with average loss for Antenna XII, and the input parameters of this sample are presented in Table 2.12.

Table 2.11 Return loss prediction error in terms of RSE for the smaller test set.

	Best	**Average**	**Worst**
Re{S11}	0.00150	0.03378	0.13441
Im{S11}	0.00100	0.03214	0.13511

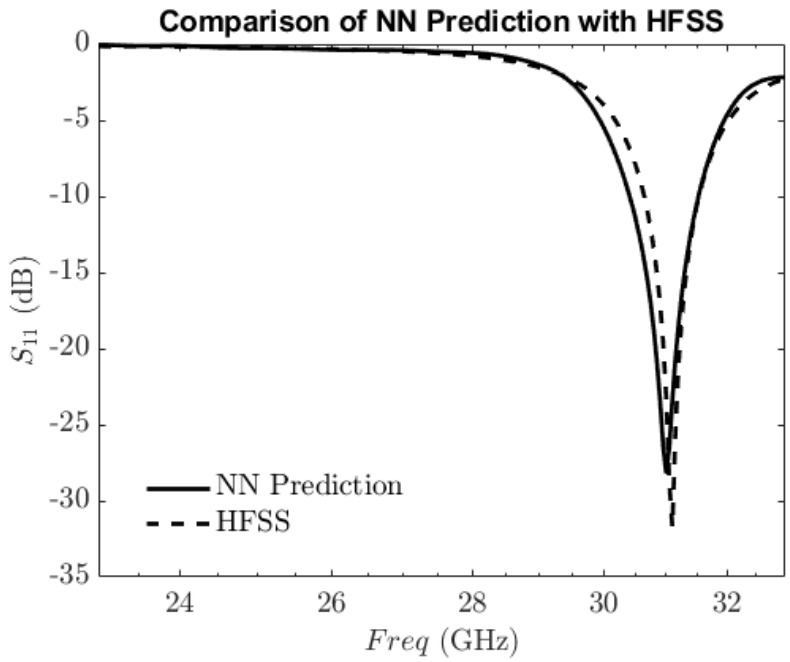


Figure 2.17 Separate test set predictions of Antenna XII with average loss.

Table 2.12 The input parameters of the prediction with average loss for the separated test set.

Input Parameter	Antenna XII
L (mm)	2.740
W (mm)	4.980
h_{cond} (mm)	0.012
h_{sub} (mm)	0.188
h_{sr} (mm)	0.020
r_{probe} (mm)	0.0159
r_{pad} - r_{probe} (mm)	0.025
$r_{antipad}$ - r_{pad} (mm)	0.045
x_{probe} / L	0.300
ε_{sub}	2.575
ε_{sr}	2.850

2.5 Flexible Feeding

As mentioned in the Section 2.2.1, the feed network was excluded from the model structure in order to gain flexibility. In this way, any feeding network according to design criteria can be cascaded with the model as in Figure 2.18.

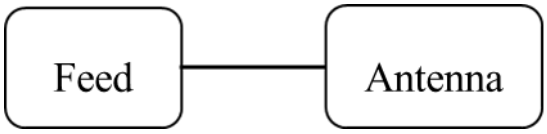


Figure 2.18 The illustration of cascaded structure.

Figure 2.19 illustrates an antenna model with stripline feeding. The feeding network was placed at the bottom of the second conductive layer of the patch antenna model. In Figure 2.20, two comparisons are represented. In (a), the NN predictions and

HFSS results are illustrated for the patch antenna model without the feeding network. In (b), the S-parameters of the stripline-feeding network were cascaded with both NN predictions (Cascade NN) and HFSS results of the patch antenna model (Cascade HFSS), and these cascaded S_{11} parameters are compared to the HFSS results of the model including the feding network (Stripline HFSS). Figure 2.20 shows a good correlation between the results. This indicates that the feeding network can be added into the surrogate model independently, depending on the specific design requirements. Moreover, Table 2.13 indicates that the resonant frequencies and the bandwidths of the three models are close to each other.

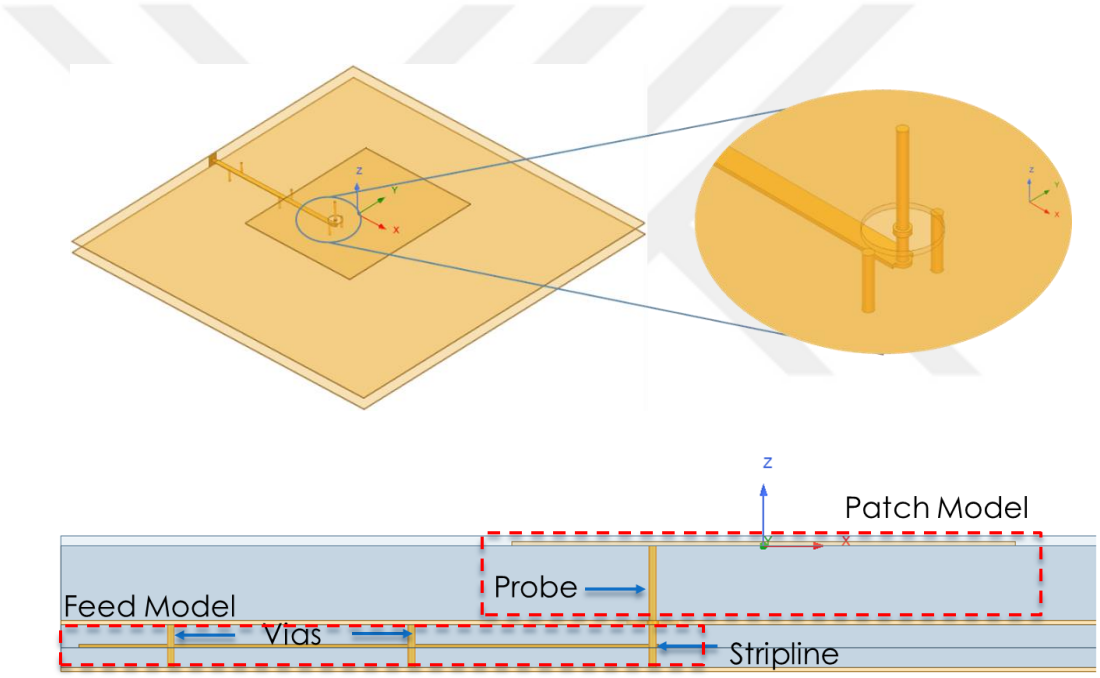


Figure 2.19 The illustration of cascaded stripline feeding network.

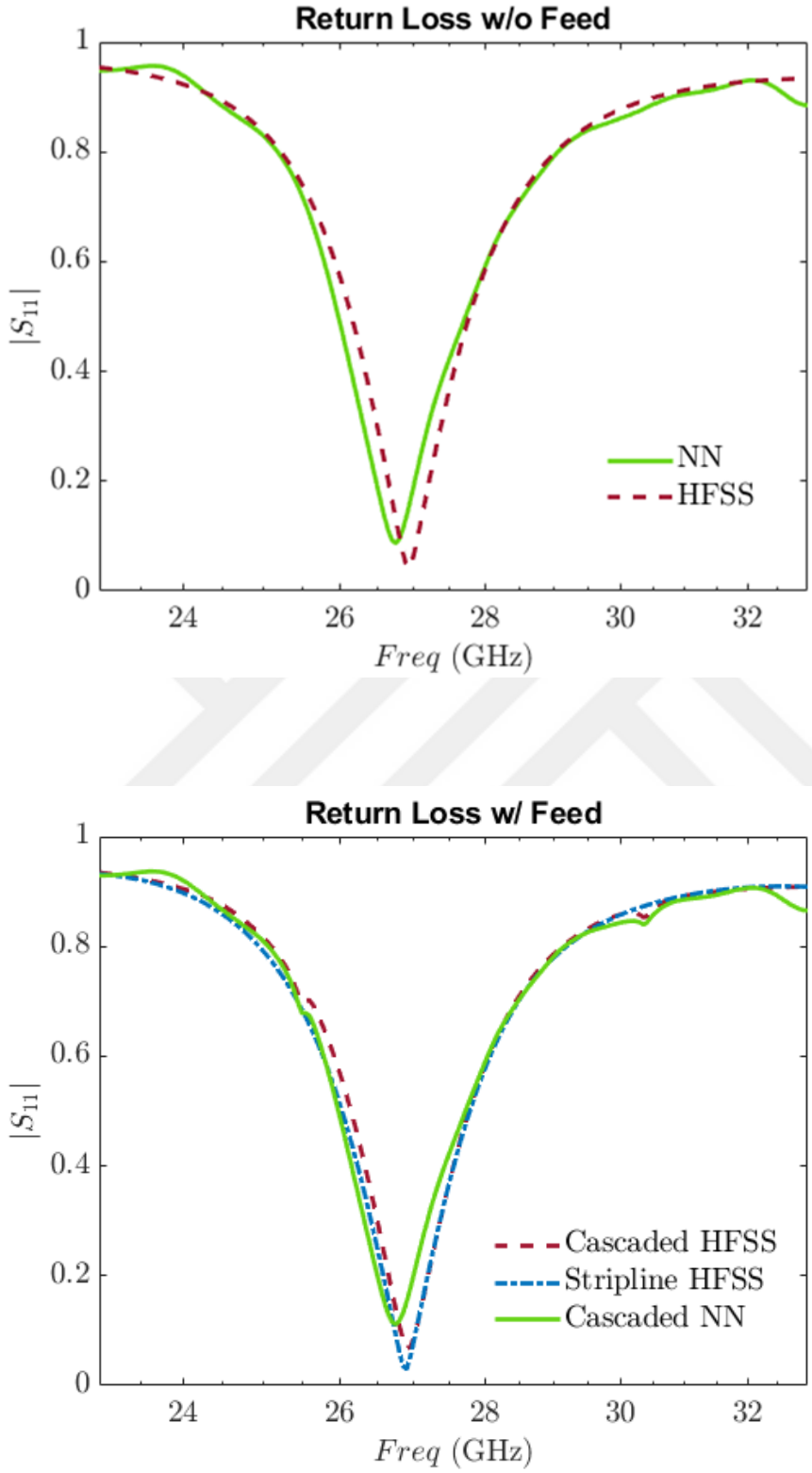


Figure 2.20 The comparison of model without feeding network and model with feeding network cascaded.

Table 2.13 Comparison of resonant frequency and bandwidth for the cascaded NN prediction, cascaded HFSS and stripline-fed HFSS result.

	Resonant Frequency (GHz)	**Bandwidth (GHz)**
Cascaded HFSS	26.95	1.85
Stripline HFSS	26.90	1.97
Cascaded NN	26.75	1.98

2.6 Gain Predictions

Another NN has been implemented to predict the gain of the antenna. Therefore, a separate dataset was generated by computing the gains of 2000 antennas at 11 discrete frequency points within 23-33 GHz. The gain values were collected at 0° azimuth and 0° elevation, which is the typical maximum radiation direction for patch antennas. To create smoother predictions, the dataset were linearly interpolated upto 201 points, and thus an STCNN was implemented. The optimized structure of FCNN contains five fully connected layers with the nodes of 46, 56, 64, 72, 82 respectively. The output of FCNN is applied to three STCNN layers with the kernel sizes of 21, 7, 3 and the strides of 1, 3, 3 respectively. A gaussian filter is also added to the output of the last spectral transposed convolutional layer with the size of 51. Figure 2.8 illustrates the structure of this network that is emloyed to predict the gain of the antenna.

Huber loss, as defined in (2.8), was used to minimize the distance between the actual and predicted gaina with a β of 0.05. Learning rate was optimized as 0.007 with the decay rate of 0.7 each 1000 step. The RSE loss was minimized at 0.26388, and MSE was minimized at 0.03928. Even if, the RSE result is higher than the RSE of S_{11} predictions numerically, the results actually have a good correlation between the predictions and actual outputs for the high gain samples. Figure 2.21 represents the comparison of NN predictions with the HFSS results. The RSE values of these predictions are 0.00323 and 0.00208, respectively.

In some non-resonating cases as shown in Figure 2.22, the gain values are extremely small, and these samples distort the performance of NN. The RSE value of this prediction is 0.2. This is the main reason behind relatively higher RSE values. Nevertheless, for samples with higher gains, the model can achieve a good accuracy.

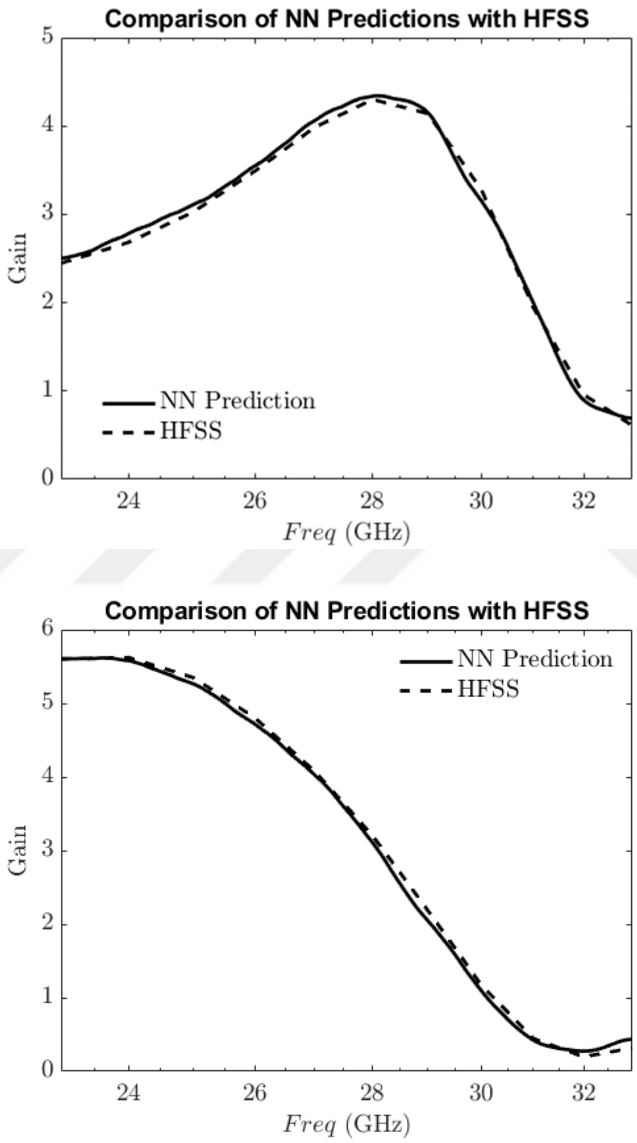


Figure 2.21 The comparison of two NN predictions with HFSS for the gain values of the antenna, RSE = 0.00323 and 0.00208, respectively.

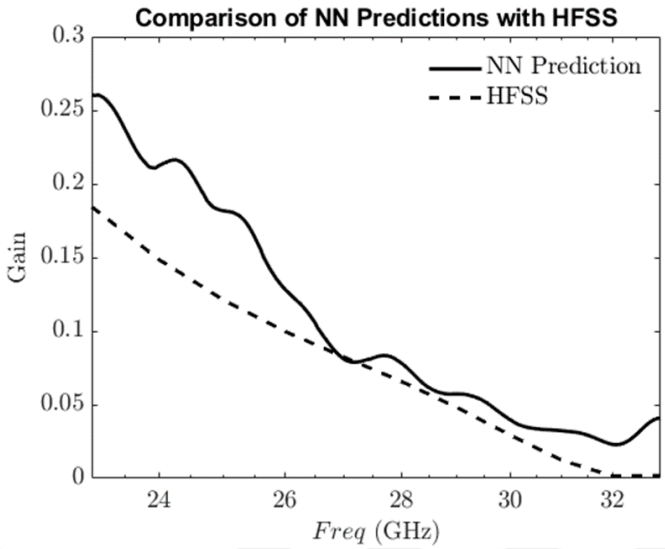


Figure 2.22 The comparison of NN prediction with HFSS for less accurate sample, RSE = 0.2.

CHAPTER 3

KNOWLEDGE-BASED REGULARIZATION

Regularization is a technique used in the machine learning domain to improve the performance of NNs, which prevents overfitting without oversimplifying the NN. Overfitting occurs when the NN learns any detail present in the given data, including noise, which prevents it from capturing the meaningful patterns between input and output effectively. These models have high variance, namely, they have too complex fit into the training data. Hence, the NN cannot generalize well for the unseen data, resulting in much higher validation loss compared to the training loss [55-56].

On the other hand, underfitting occurs when the NN has high bias, resulting in oversimplified models that fail to capture the patterns in the training data. In this case, both the validation and training losses are high [56].

A trade-off exists between bias and variance. An overly simple model leads to a high bias and a low variance, causing underfitting, while an excessively complex model results in a high variance and a low bias, leading to overfitting. Both scenarios are undesirable for practitioners of NNs. To avoid overfitting, the complexity of the model should be reduced to decrease the variance, but without increasing the bias. To prevent the underfitting, the complexity of the model must be increased to decrease the bias, but without increasing the variance. Regularization can help optimize this bias-variance tradeoff.

Regularization methods involve the addition of penalties into the loss functions to regulate the influence of unknown coefficients in the NN such as bias or variances. By applying these penalties, the complexity of the network can be adjusted to find the optimal model that strikes a balance of bias and variance [56]. In this chapter, novel regularization methods were implemented on the NN to improve the performance. These methods are considered knowledge-based regularization as they

are derived using the prior information about the system of interest and the structure of the output data. They include derivative, spectral domain, and magnitude regularizations, which will be explained in detail in this chapter. A comprehensive comparison of these methods will be discussed, along with their contributions to enhancing the performance of NNs.

3.1 Lasso (L1) and Ridge (L2) Regularization

Lasso (L1) and Ridge (L2) regularization are the most common regularization techniques to penalize the unknown coefficients of the NN. The main difference between these methods are the penalty terms. L1 regularizer penalizes the loss function with (3.1). By adjusting the value of the hyperparameter λ, the model aims to minimize the sum of the magnitudes of the coefficients. As λ increases, some coefficients are forced to shrink towards zero, and if λ is strong enough, some coefficients may become exactly zero. Consequently, L1 regularizer leads feature selection by eliminating less meaningful features, thus solving overfitting by reducing the number of features and simplifying the model.

On the other hand, L2 regularizer employs the squared of the magnitudes of the unknown coefficients to penalize the loss function (3.2). This regularization method discourages large weights to avoid overfitting similar to L1 regularization. However, unlike L1, it does not eliminate features entirely; instead, it shrinks the coefficients towards zero without making them exactly zero. This encourages the impact of non-zero features to be spread across the entire model, ensuring that all features contribute to some extent to the model's predictions.

$$Loss\ w/L1 = Error(y, \hat{y}) + \lambda \sum_{i=1}^{m} \sum_{j=1}^{n} \left| w_{ij}^{k} \right| \tag{3.1}$$

$$Loss\ w/L2 = Error(y, \hat{y}) + \lambda \sum_{i=1}^{m} \sum_{j=1}^{n} (w_{ij}^{k})^2 \tag{3.2}$$

3.2 Derivative Regularization

The black box method just focuses on minimizing the distance between the predicted and true output. This is fairly enough to solve a regression problem with a single and scalar output, but in this problem, the predicted output is a function of frequency. Therefore, the NN should not only minimize the distance between predicted and true outpus; should also minimize the distance between their derivatives. Hence, adding derivatives into the loss function allows to enhance the performance of NN, and decreases the non-physical predictions.

Morever, the derivatives of the output can provide valuable information about the sensitivity of the network's predictions to small changes in the input. By integrating the derivatives of output into the loss function, the network can be encouraged to generate more robust outputs that are less affected by noise or minor perturbations in the input data. Additionally, the regularization terms, which penalize large derivatives, help the optimization to find smoother solutions minimizing the abrupt changes in the function. Therefore, the derivatives of the NN output can be utilized to regularize the training process effectively.

Since S_{11} parameters are continuous and differentiable, the first derivatives of S_{11} are also continuous [57]. Thus, the first derivative of the output can be employed as an additional term in the loss function. As for higher derivatives, such as second or third derivatives, their differentiability needs to be investigated, and if their gradients exist, they can also be integrated into the loss function.

To penalize the NN with the derivatives of the output, a new loss function, which is obtained by (3.3), was defined in terms of Huber loss.

$$h(y,\hat{y},\beta) = \begin{cases} \frac{1}{N_s}\frac{1}{N_f}\sum_{s=1}^{N_s}\sum_{f=1}^{N_f}\frac{0.5(y-\hat{y})^2}{\beta} & if\ |y-\hat{y}| < \beta \\ \frac{1}{N_s}\frac{1}{N_f}\sum_{s=1}^{N_s}\sum_{f=1}^{N_f}|y-\hat{y}| - 0.5\times\beta & otherwise \end{cases} \quad (3.3.a)$$

$$Loss = h(y,\hat{y},\beta) + \sum_{n=1}^{N}\lambda_n\, h\left(\frac{\partial^n y}{\partial f^n},\frac{\partial^n \hat{y}}{\partial f^n},\alpha\right) \quad (3.3.b)$$

In this new loss function, Huber loss, which is expressed by $h(y,\hat{y},\beta)$, with a β of 0.1 was employed to update the weights, biases, and the optimizer. The derivatives of the true and predicted outputs were integrated into $h(y,\hat{y},\beta)$ as the function of $h\left(\frac{\partial^n y}{\partial f^n},\frac{\partial^n \hat{y}}{\partial f^n},\alpha\right)$ with an α of 0.001. These β and α values are hyperparameters, and were determined by optimizing the accuracy of NN. Since the optimized α value is very small, the loss is very similar to MAE. The studies showed that including the derivatives with orders higher than 2 does not provide a significant improvement in the accuracy, particularly when the computational cost is considered. The derivative parts were weighted during the hyperparameter optimization with λ_1 of 6 and λ_2 of 0.05.

The performance improvement of the NN with derivative regularization is indicated in Figures 3.1, 3.2 and 3.3. These results were obtained for one of the test samples, namely Antenna XIII, with input parameters listed in Table 3.2. Figure 3.1 shows that the integration of the first derivative significantly improves the smoothness of the output for Re{S_{11}}. Zooming in on Figure 3.1, Figure 3.2 further demonstrates that including the second derivative enables the NN to capture the actual profile more effectively, particularly around where the antenna is resonating. Moreover, Figure 3.3 shows the NN predictions for Im{S_{11}}. Although the predictions including derivative regularization are distorted around local minima and maxima, the regularized predictions of Im{S_{11}} have a better fit to the actual data for the rest of of profile as seen in Figure 3.3. Table 3.1 shows RSE losses for Antenna XIII to

represent these improvements. The overall performance increases with the impact of the derivative regularization. Although applying Huber loss without regularization generates good fits and smooth profiles for most predictions, there are instances, like Antenna XIII, where less accurate predictions occur. The purpose of derivative regularization is to increase the overall accuracy, specifically for such antennas.

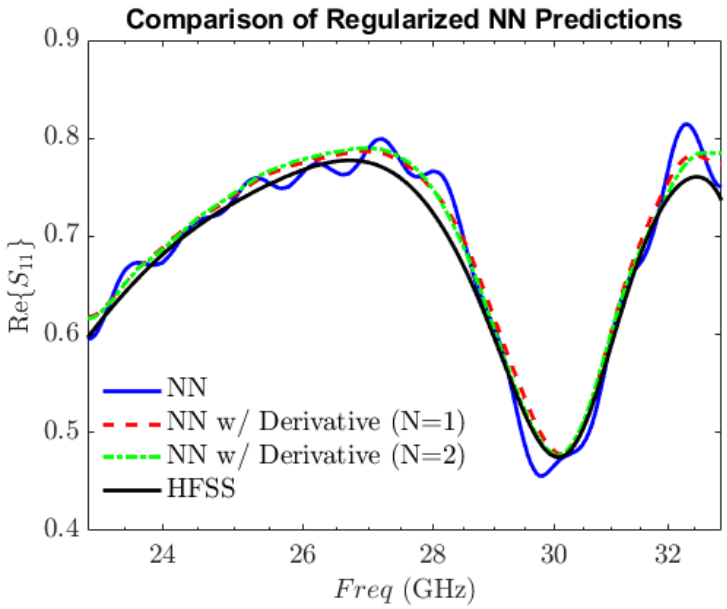


Figure 3.1 The demonstration of the performances for Re{S_{11}} with derivative (N=1) and (N=2) regularization for Antenna XIII.

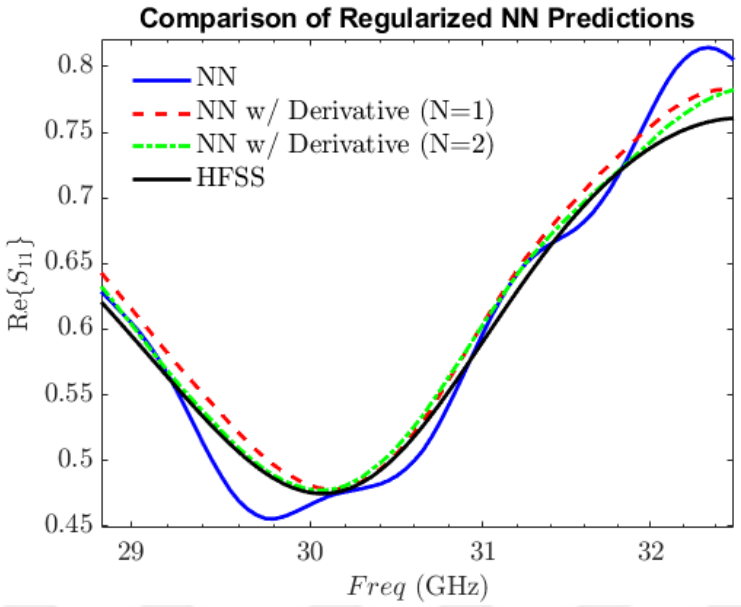


Figure 3.2 The zoomed-in version of Figure 3.1.

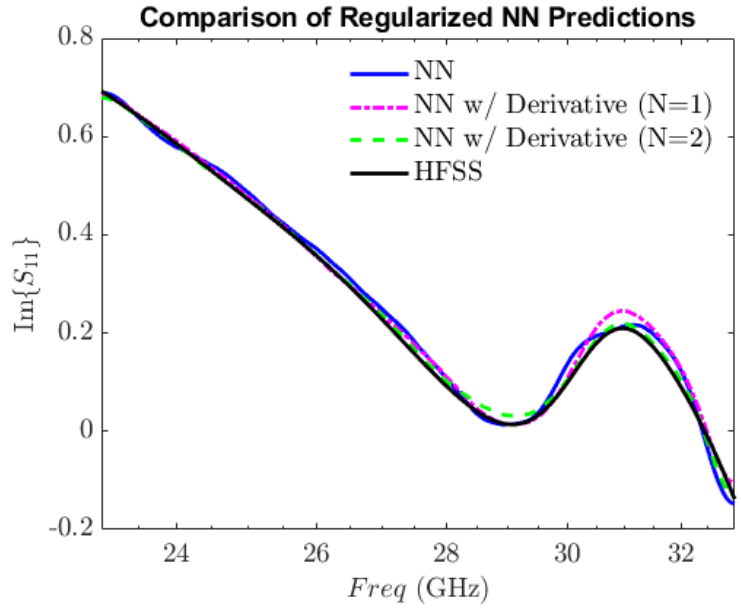


Figure 3.3 The demonstration of the performances for Im{S_{11}} with derivative (N=1) and (N=2) regularization for Antenna XIII.

Table 3.1 The comparison of RSE losses which are generated by derivative (N=1) and (N=2) regularization for Antenna XIII.

	RSE	
Regularization	**Re{S_{11}}**	**Im{S_{11}}**
NA	0.05032	0.00754
Derivative (N=1)	0.02595	0.00680
Derivative (N=2)	0.02328	0.00199

Table 3.2 The input parameters of Antenna XIII.

Input Parameter	**Antenna XIII**
L (mm)	4.864
W (mm)	4.672
h_{cond} (mm)	0.013
h_{sub} (mm)	0.718
h_{sr} (mm)	0.072
r_{probe} (mm)	0.045
r_{pad} - r_{probe} (mm)	0.013
$r_{antipad}$ - r_{pad} (mm)	0.085
x_{probe} / L	0.337
ε_{sub}	3.355
ε_{sr}	4.757

3.3 Spectral Domain Regularization

Following a similar motivation with the derivative regularization, one can also conclude that the predicted output function should have a similar spectrum with the true output. Therefore, the addition of the spectral domain representation of the output into the loss function can lead to improve the performance of the NN and the smoothness of the response. By penalizing the NN with higher order spectral

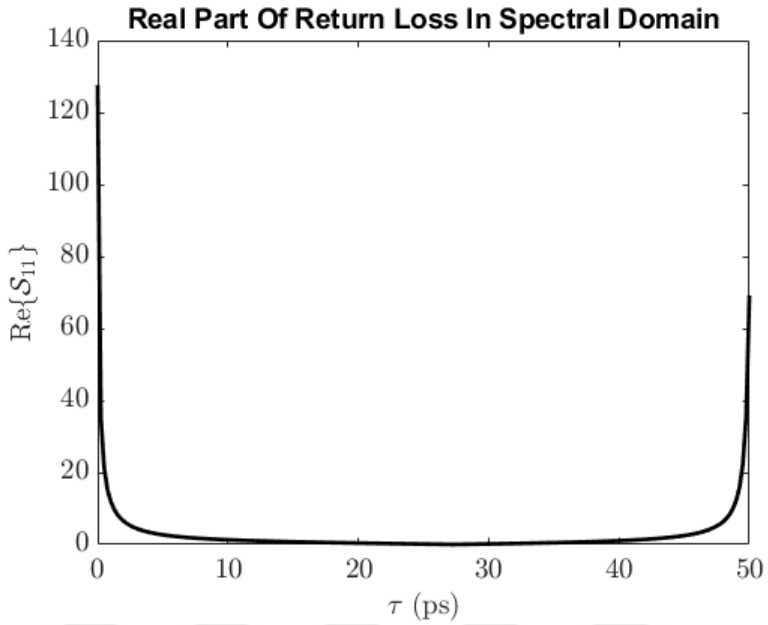
components, the NN focuses on the dominant components of the output and minimize the influence of irrelevant or noisy spectral components.

The predictions of S_{11} parameters may exhibit decreased accuracy for higher spectral components. This is because each higher spectral component represents a larger spectral increment relative to the fundamental component, and their magnitudes change faster compared to the low spectral components, which have a smoother and more gradual variation. Hence, penalizing the NN with higher spectral components, which introduce faster changes of the predictions, can be effective to improve the smoothness of the output by minimizing these rapid changes.

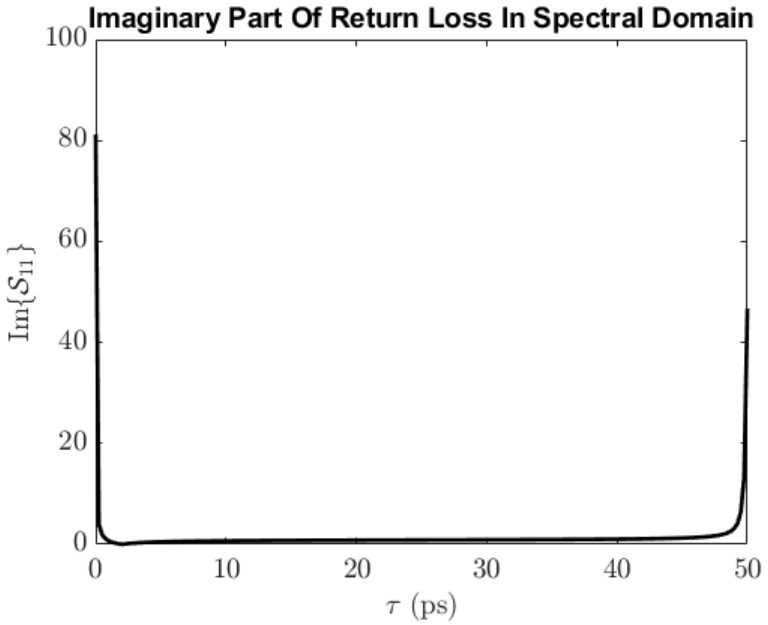
Therefore, the fast fourier transform (FFT) of the output which is represented with $\mathcal{y}$ can be incorporated into the loss function to further improve the accuracy of predictions. (3.4) shows the loss function with FFT regularization. In this case, the Huber loss, as previously defined, employed a β value of 0.1 that is equal to α, and with a weighting factor, λ, of 0.1.

$$Loss = h(y, \hat{y}, \beta) + \lambda\, h(\mathcal{y}, \hat{\mathcal{y}}, \alpha) \tag{3.4}$$

To further improve the performance of the NN, FFT of the output was filtered to include only the spectral components with relatively higher magnitudes. As shown in Figure 3.4, for most of the S_{11} samples, the components in the middle range of the spectrum have much lower magnitudes and they were filtered out using a band stop filter. Specifically, the filter was optimized to select 40 spectral points at the beginning and 40 spectral points at the end of the signal. By applying this band stop filter, the network can emphasize on the selected significant spectral components and reduce the impact of the less meaningful ones in the middle of the spectrum to improve the performance of the NN and smoothness of the predictions. The filter-applied version of the FFT regularization loss is called as the filtered FFT (F-FFT) regularization loss. The F-FFT regularization loss gives optimum results when the λ is equal to 0.05, β is 0.1 and α is 0.001.



(a) The spectral domain representation of real part of S_{11} parameters.



(b) The spectral domain representation of imaginary part of S_{11} parameters.

Figure 3.4 The spectral domain representations of real and imaginary parts of S_{11} parameter for a random antenna sample to show the more significant impact of the components at the beginning and end of the time range.

Figures 3.5, 3.6 and 3.7 demonsrate the impact of FFT and F-FFT regularization for the test sample, Antenna XIII. As seen in Figure 3.5, the F-FFT regularization forces to the NN to generate closer predictions to the actual data, particularly around the minimum of Re{S_{11}}. Moreover, both FFT and F-FFT increase the smoothness of the output. Figure 3.6 shows the comparison of the predictions of Im{S_{11}}, and it is observed that the predictions are distorted with FFT regularizations. The regularization can improve one part while distorting other part for some particular cases. Table 3.3 shows the RSE losses for the mentioned methods which proves that the overall performance of real part is improved by regularization while the overall perfromance of imaginary part decreases. Moreover, Figure 3.7 represents the comparison of derivative (N=2) and F-FFT regularization.

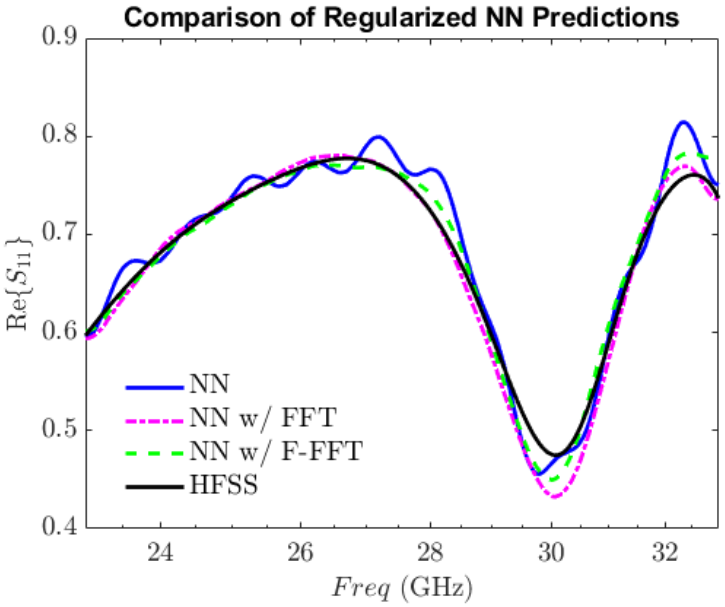


Figure 3.5 The comparison of the NN performances for Re{S_{11}} with FFT and filtered F-FFT regularization for Antenna XIII.

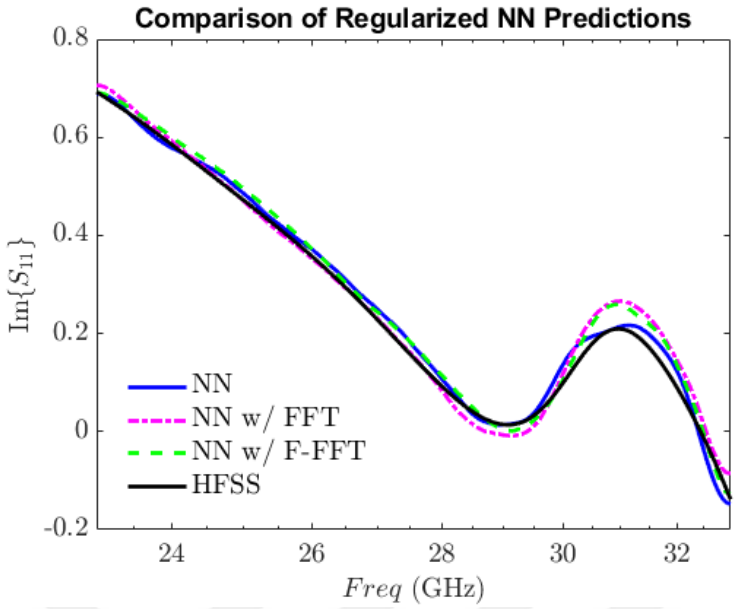


Figure 3.6 The comparison of the NN performances for Im{S_{11}} with FFT and F-FFT regularization for Antenna XIII.

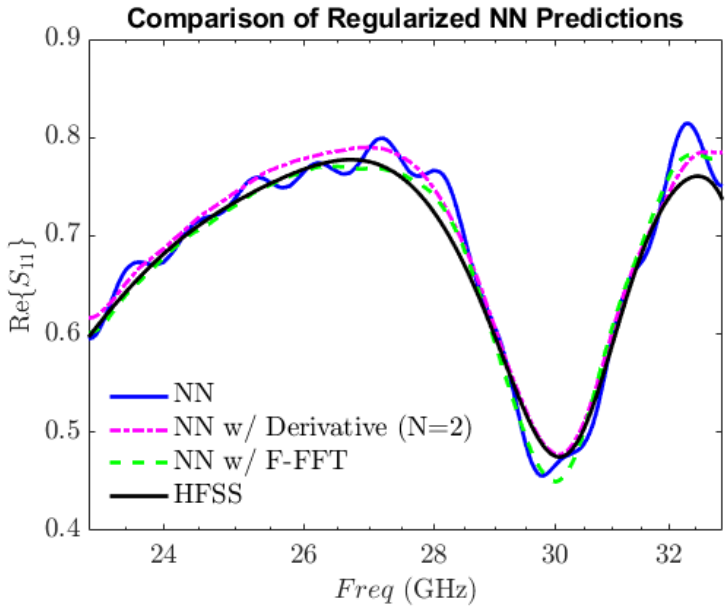


Figure 3.7 The comparison of the performance improvements of the NN with derivative (N=2) and F-FFT regularization for Antenna XIII.

Table 3.3 The comparison of RSE losses which are generated by FFT and F-FFT regularization for Antenna XIII.

	RSE	
Regularization	**Re{S_{11}}**	**Im{S_{11}}**
NA	0.05032	0.00754
FFT	0.03115	0.01831
F-FFT	0.02194	0.01368

3.4 Magnitude Regularization

Since the antenna is a passive device, the magnitude of the return loss of an antenna should be less than unity. However, some predictions can violate this passivity rule, which results in an S_{11} magnitude higher than unity, as the NN does not contain any physical information about the antenna. This may result in potentially non-physical predictions, even if the NN effectively minimizes the distance between the predictions and the actual outputs. The NN prediction updated by Huber loss without any regularization for Antenna XIV indicates this behavior in Figure 3.8. To enforce the passivity, magnitude regularization has been implemented by including magnitude-related terms in the loss function. Predictions with passivity violations are forced to fit in the physically meaningful range by adding extra penalties that include the information of their magnitudes. The magnitude integrated Huber loss function can be seen in (3.5). The improvement by this regularization is demonsrated in Figure 3.9 where the magnitudes, that violate the passivity, were supressed for Antenna XIV. The input parameters of Antenna XIV are listed in Table 3.4. Noteworthy, the passivity violations for all cases are less than 0.1 dB and occur in the regions far from resonance.

$$Loss = h(y, \hat{y}, \beta) + \lambda\, h(|y|, |\hat{y}|, \alpha) \tag{3.5}$$

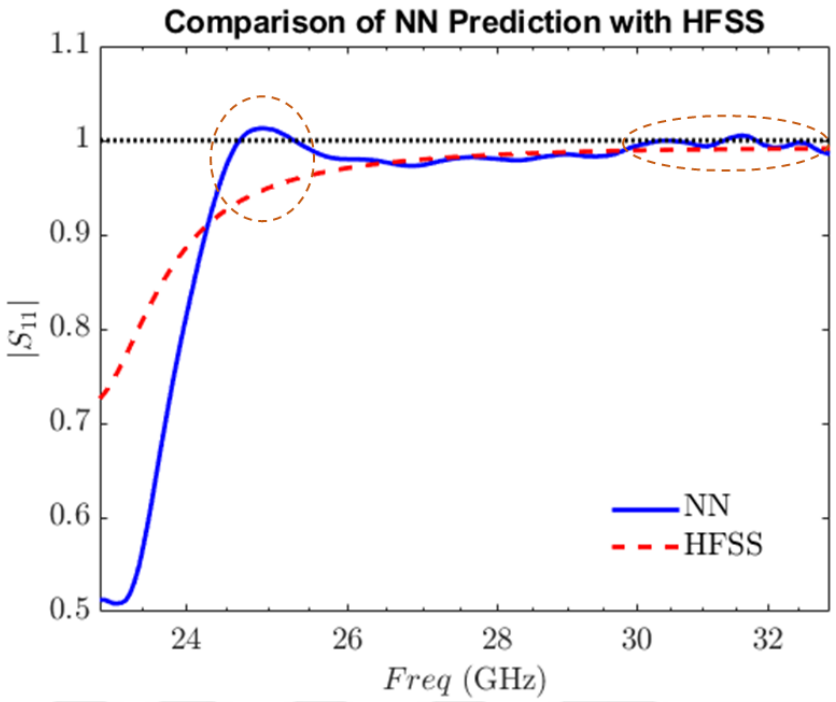


Figure 3.8 The illustration of passivity violation for Antenna XIV.

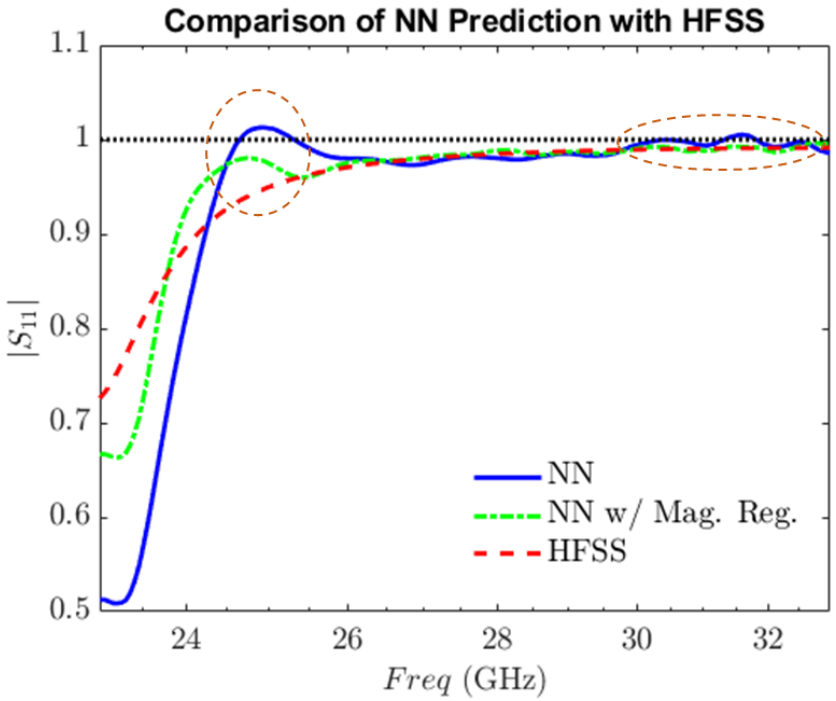


Figure 3.9 The illustration of supressed passivity violation for Antenna XIV.

Table 3.4 The input parameters of Antenna XIV.

Input Parameter	Antenna XIV
L (mm)	3.099
W (mm)	2.877
h_{cond} (mm)	0.019
h_{sub} (mm)	0.114
h_{sr} (mm)	0.052
r_{probe} (mm)	0.046
r_{pad} - r_{probe} (mm)	0.016
$r_{antipad}$ - r_{pad} (mm)	0.047
x_{probe} / L	0.408
ε_{sub}	3.991
ε_{sr}	3.631

To better illustrate this influence, Table 3.5 displays the decrease in the number of samples with passivity violations for different NNs. Although the magnitude regularization could not entirely prevent the generation of non-physical samples, it significantly reduced their occurrence.

Table 3.5 The number of passivity violated samples for different types of losses.

	Number of Samples with Passivity Violation	
Regularization	**w/o Magnitude Regularization**	**w/ Magnitude Regularization**
NA	91	28
Derivative (N=2)	52	11
F-FFT	66	17

Moreover, the magnitude regularization helps the NN to further improve the accuracy of predictions. To show this improvement, the influence of magnitude regularization on Antenna XIII, a representative test sample chosen to showcase the effects of previous regularization methods, was also examined. Figures 3.10 and 3.11 are the versions of Figures 3.1 and 3.5 including the magnitude regularization. To represent the impact of magnitude regularization, the imaginary parts are also illustrated in Figures 3.12 and 3.13. Lastly, Table 3.6 indicates the RSE losses for the examined regularization methods.

For the derivative regularized NN, the magnitude regularization helps to improve the results of real part as seen in Figure 3.10. It helps to increase the accuracy for the maxima of the output as emphasized in the circled area. But, it slightly distorts the predictions of the imaginary part, as seen in Figure 3.12.

On the other hand, for the FFT regularized NN, the magnitude regularizaton leads to improve the predictions of the imaginary part. As seen in the circled area of Figure 3.13, the predictions around local maxima are improved. Moreover, according to Table 3.6, the magnitude regularized predictions have higher accuracy than the predictions without regularization which is indicated in Table 3.3. The integration of magnitude regularization improves the results of FFT regularization. However, it decreases the performance of the real part predictions as illustrated in Figure 3.11. Since the real and imaginary part are applied to the NN together, the improvement of the performance is tested for both real and imaginary part.

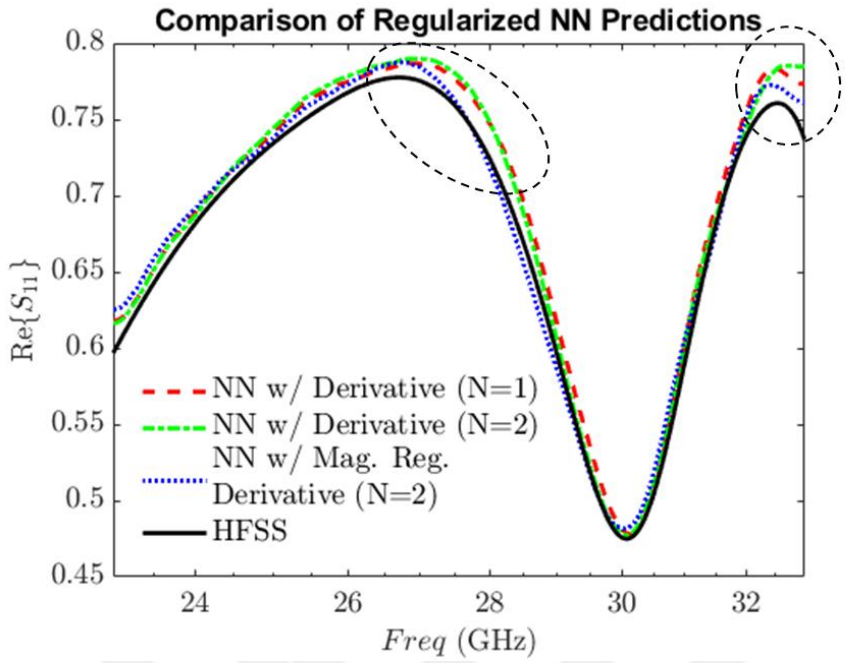


Figure 3.10 The comparison of the magnitude regularized NN predictions with the derivative regularized NN predictions for Re{S_{11}} of Antenna XIII.

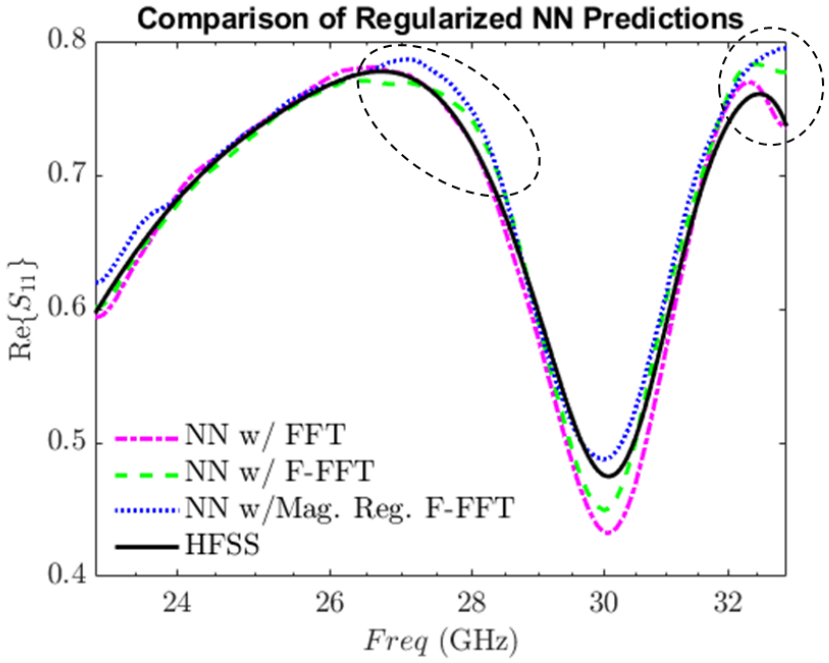


Figure 3.11 The comparison of the magnitude regularized NN predictions with the FFT regularized NN predictions for Re{S_{11}} of Antenna XIII.

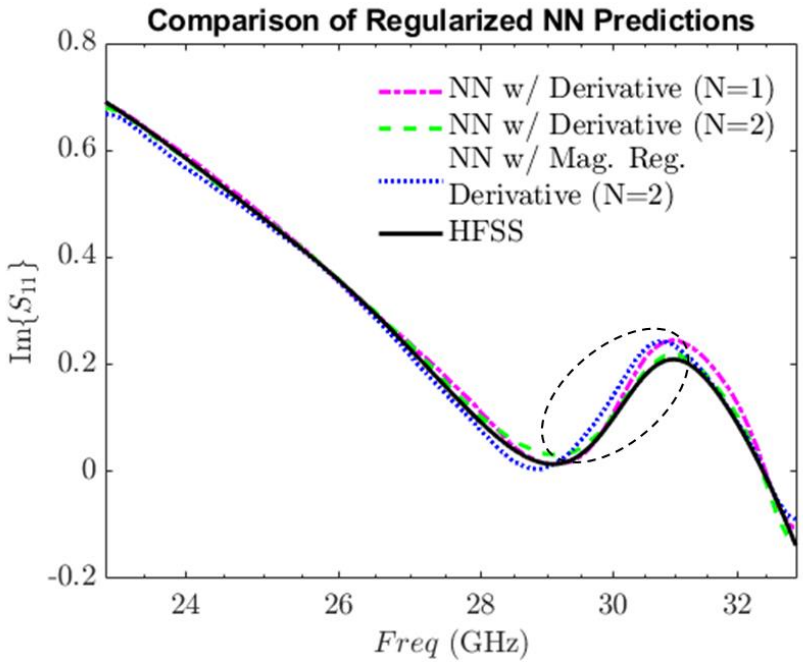


Figure 3.12 The comparison of the magnitude regularized NN predictions with the derivative regularized NN predictions for Im{S_{11}} of Antenna XIII.

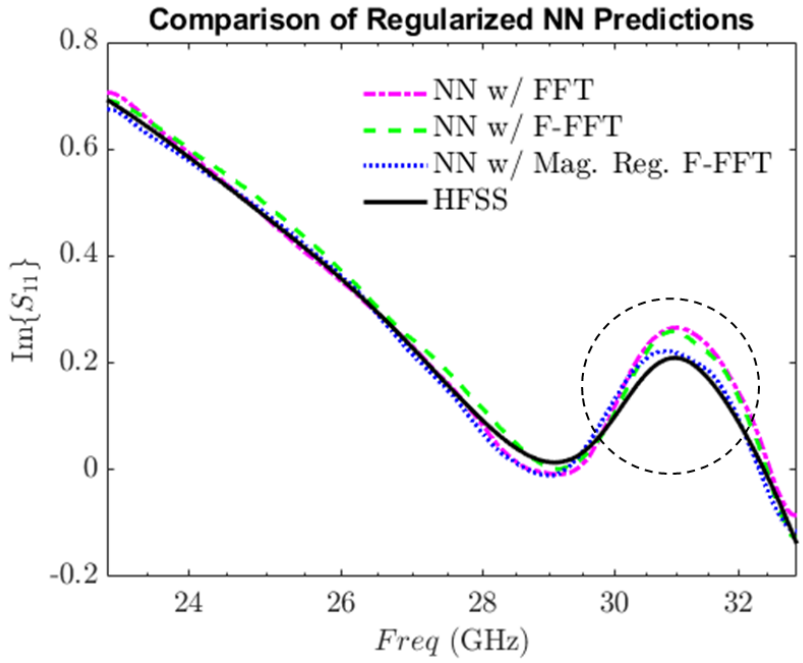


Figure 3.13 The comparison of the magnitude regularized NN predictions with the FFT regularized NN predictions for Im{S_{11}} of Antenna XIII.

Table 3.6 The RSE losses of the different regularizations for Antenna XIII.

		RSE	
	Regularization	**Re{S_{11}}**	**Im{S_{11}}**
	Derivative (N=1)	0.02595	0.00680
	Derivative (N=2)	0.02328	0.00199
w/ Magnitude	**Derivative (N=2)**	0.01229	0.00858
	FFT	0.03115	0.01831
	F-FFT	0.02194	0.01368
w/ Magnitude	**F-FFT**	0.03221	0.00595

3.5 Comparison of the Implemented Regularization Techniques

To reemphasize, the main motivation behind using knowledge-based regularization is to enhance the accuracy of the NN, reduce the passivity violations, and improve smoothness of the output. In this regard, the NN was trained with the same dataset containing 5000 samples, and the weights were updated by different regularization methods that are integrated into the loss function. Moreover, to minimize the randomness in the predictions, the weights were initialized with the same values for all cases.

Table 3.7 indicates the decrease of RSE loss of the test set for different regularization methods compared to the unregulated NN performance given in the first row. The addition of the second derivative into the loss function enhances accuracy to some extent, although its impact is not as significant as that of the first derivative. This aligns with the expectations, as the first derivative typically has a more pronounced effect. Additionally, the FFT brings about a similar improvement as the first and second derivatives combined. However, the F-FFT yields the best performance by filtering out the spectral components with relatively lower coefficients.

Magnitude regularization was also included in the loss function, together with other regularization methods, to decrease the number of non-physical predictions as mentioned earlier. In the optimized NN with magnitude regularization, the weight factor, λ, in (3.5) was determined as 0.4 for the Huber loss, 0.5 for the derivative regularization loss and 1.2 for the F-FFT regularization loss. The impact of magnitude regularization on the whole test set was demonstrated in Table 3.7.

To quantify the smoothness of the output, its arc length can be used because, the arc length gives a numerical metric for the "curviness" or the total length of a function in a given interval. The arc length is calculated as in (3.6), where a and b are the intervals of frequency range.

$$arc\ length = \int_a^b \sqrt{1 + \left(\frac{dy}{df}\right)^2}\, df \tag{3.6}$$

This integration represents the cumulative length of the curve traced by the function. In this study, the calculation of arc length was modified to obtain a better smoothness evaluation metric. Instead of calculating the arc length of the entire function, the arc lengths of smaller sections of 10 frequency points were computed by utilizing a sliding window. This methodology provides a more detailed understanding of the function's smoothness by considering local segments rather than relying solely on a single global measure. As a result, the values of a and b are altered throughout the frequency range of 23-33 GHz. Then, the absolute difference of arc lengths for smaller sections between the predicted and actual output was calculated, and normalized with the arc length of actual data. Lastly, the mean values of the relative absolute difference for the entire test set are represented in Table 3.7 as the smoothness metric. Functions with relative absolute differences are typically smoother, as they exhibit less fluctuation around the true value over the specified interval. All regularization methods improve the smoothness of the NN, as indicated in Table 3.7. The addition of the derivative (N=1) and FFT of the output into the loss function significantly increase the smoothness of the predictions. The contribution of the derivative (N=2) and F-FFT also develop the smoothness, but their impact on

smoothness are not as significant as their impact on the accuracy. Similar to the accuracy results, F-FFT with magnitude regularization exhibits the smoothest outputs.

Table 3.7 The illustration of the performance improvement for different regularization techniques.

		RSE		
	Regularization	**Re{S_{11}}**	**Im{S_{11}}**	**Smoothness Metric**
	NA	0.05100	0.05300	1.799
w Magnitude	**NA**	0.04900	0.04700	1.771
	Derivative (N=1)	0.04617	0.04850	1.272
	Derivative (N=2)	0.04600	0.04650	1.257
w Magnitude	**Derivative (N=2)**	0.04470	0.04670	1.259
	FFT	0.04684	0.04690	1.299
	F-FFT	0.04500	0.04400	1.269
w Magnitude	**F-FFT**	0.04200	0.04280	1.255

As a result, the knowledge-based regularization methods achieved to increase the accuracy and smoothness of the predictions, and decrease the number of non-physical predictions. According to these improvements, it is also expected that the regularized NN can reach to similar accuracy levels, compared to an unregularized NN, with number of samples less than 5000. Table 3.8 indicates RSE values of the training and test sets for incremental number of training data. The loss value of 0.052 in Table 3.8 is obtained with Huber loss without regularization (NA) when 4000 samples are used for training. However, the NN, that is trained with derivative (N=2) and F-FFT regularization loss, can reach at this accuracy level with 500 less training data. If the magnitude regularization is also included, the unregularized NN reaches a similar accuracy level with 3500 training data, whereas the regularization techniques achieves it with 3000 training data. This means that the F-FFT or

derivative regularization combined with magnitude regularization can achieve the performance of an unregularized NN with 1000 (25%) less training data. Table 3.8 also indicates that the overffitting decreases with the integration of regularization. In the table, the test losses are much closer to the train losses when regularization is applied. This reduction in overfitting is a clear indication of the improved performance. Ideally, a well-performing NN should achieve test performance that is as close as possible to the train performance, and the presence of regularization helps to achieve this goal.

Finally, Table 3.9 provides a comparison of the computation times per epoch for the different regularization methods which are implemented with the usage of NVIDIA Quadro P2000 GPU acceleration. As expected, the improved accuracy of the regularized NNs comes with an expense of increased computation time for the same number of data samples. In comparison, the reduction of samples by 1000 saves approximately one day of data generation. Comparing the computation times, smoothness, passivity violations, and accuracy, derivative and F-FFT regularizations have trade-offs. While F-FFT provides the most accurate and smoothest results, derivative regularization provides a faster training time and less passivity violations, with slightly inferior accuracy.

Table 3.8 The comparison of train and test RSE losses for non-regularized, derivative (N=2) and F-FFT regularized methods with/without magnitude regularization for incremental dataset.

		NA		Derivative (N=2)		F-FFT	
	Amount of Data	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
w/o Magnitude	**2000**	0.018	0.100	0.017	0.093	0.015	0.095
	2500	0.020	0.074	0.019	0.069	0.016	0.062
	3000	0.022	0.059	0.021	0.058	0.024	0.056
	3500	0.026	0.057	0.021	0.052	0.027	0.051
	4000	0.027	0.052	0.023	0.046	0.021	0.045
w/ Magnitude	**2000**	0.017	0.090	0.017	0.085	0.014	0.083
	2500	0.020	0.068	0.018	0.063	0.015	0.061
	3000	0.022	0.055	0.021	0.054	0.018	0.054
	3500	0.023	0.052	0.020	0.049	0.018	0.046
	4000	0.025	0.048	0.023	0.046	0.020	0.043

Table 3.9 The comparison of computation times of the different regularization methods in terms of times (sec) / epoch for 3200 epochs.

	Times (sec) / epoch	
Regularization	**w/o Magnitude Regularization**	**w/ Magnitude Regularization**
NA	0.070	0.073
Derivative (N=2)	0.075	0.086
F-FFT	0.090	0.096

CHAPTER 4

THE CAVITY MODEL BASED NEURAL NETWORK

In this section, a novel knowledge-based approach was applied to the NN. The analytical formulations of the cavity model of the patch antenna was employed to guide the predictions of the NN. This led to the development of an NN that includes the equations of the physical structure, thereby creating a physics-aware model.

4.1 The Cavity Model

The cavity model of a patch antenna is a representation which gives the behavior of the antenna using a simplified concept, known as the resonant cavity. The region between the patch and the ground plane in the basic microstrip patch antenna geometry resembles an electromagnetic resonator or cavity. This model contains several simplifying assumptions, which are valid for thin substrates, to deduce the characteristics of the antenna. According to these assumptions, the fields between the patch and the ground plane are calculated analytically.

In the case of a patch antenna, the fields inside the cavity can leak out to the outer medium and radiate. These radiated fields can be obtained by finding the fields inside the cavity, imposed by the boundary conditions on the side walls of the antenna. The cavity model assumes that the substrate is thin satisfying the condition $h_{sub} << \lambda$, where λ is the wavelength within the dielectric [58]. Therefore, the fields are assumed to have no variation along the normal direction to the patch. Figure 4.1 illustrates the assumed boundary conditions of the cavity model. The boundary side walls can be considered as Perfect Magnetic Conductor (PMC) walls, where the tangential component of the magnetic field vanishes ($H_t = 0$), and similarly, the top

and bottom walls are considered as Perfect Electric Conductor (PEC), where the tangential component of the electric field vanishes ($E_t = 0$). Since ($E_t = 0$) on the conductive walls, $\vec{E} = \hat{z}E_z$ and $\vec{J}$, the current density of the feed, is assumed to be directed along *z*.

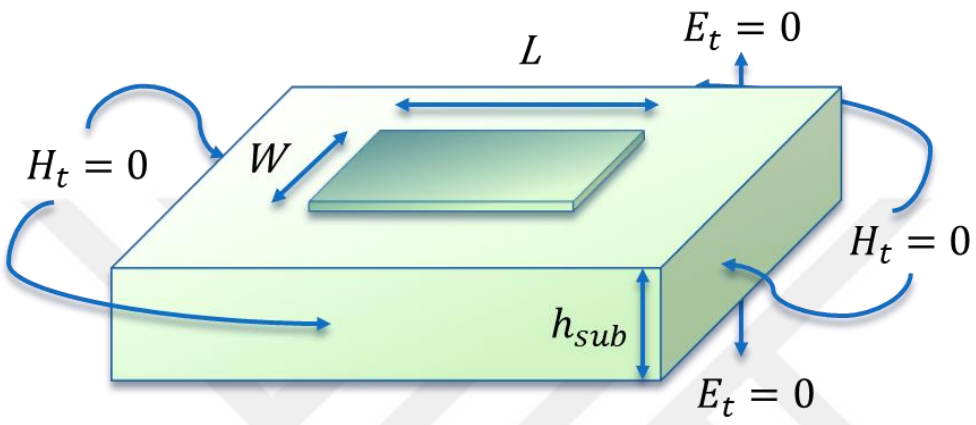


Figure 4.1 The boundary conditions of the cavity model of patch antenna.

According to these assumptions, the inhomogenous wave equation of $\vec{E}$ can be obtained as in (4.1) subject to the Neuman boundary conditions.

$$(\nabla^2 + k_d^2)E_z = j\omega\mu_0 J_z \tag{4.1}$$

The eigenfunction expansion of (4.1) can be described as in (4.2) for the model illustrated in Figure 4.1.

$$E_z = \sum_m \sum_n A_{mn}\psi_{mn} \tag{4.2}$$

where

$$\psi_{mn} = \cos\left(\frac{m\pi}{L}\right)\cos\left(\frac{n\pi}{W}\right) \; m, n = integers$$

Hence, k_d values that satisfy (4.1) are in the form of resonance wavenumber, k_{mn}, and the resonant frequency, f_r can be determined from (4.3), assuming no radiation.

$$f_r = \frac{k_{mn}}{2\pi\sqrt{\mu_0 \varepsilon_{sub}}} \tag{4.3}$$

where

$$k_{mn} = \sqrt{\left(\frac{m\pi}{L}\right)^2 + \left(\frac{n\pi}{W}\right)^2} \quad m, n = integers$$

The losses are not considered in (4.3), and at the resonant frequency, k_d is purely real and the real part of the input impedance goes to infinity. In a more realistic representation of the cavity model, the losses, including the conductor, dielectric, and radiation losses, should be taken into account. When radiation is considered, the wall admitances of the boundaries are complex, leading to a complex k_d to satisfy the boundary conditions and Rmax assumes large but finite values at the resonant frequencies.

The radiation of the patch antenna gives rise to fringing fields, as depicted in Figure 4.2. These fields are radiated from the edges of the patch, causing an extension of the effective length and width. The distance where fringing fields extend to and the resulting effective length are represented by Δl and L_{eff} respectively. Additionally, the radiated fields emerging from the substrate, requires modification in the permittivity because the radiated fields are also observed out of the subsrate, not just inside of it. The effective permittivity (ε_{eff}) is used to represent the nonuniform field distribution around the antenna. The expressions, obtained by curve-fitting assuming only air above the antenna, of these effective parameters are given in (4.4) [58-59].

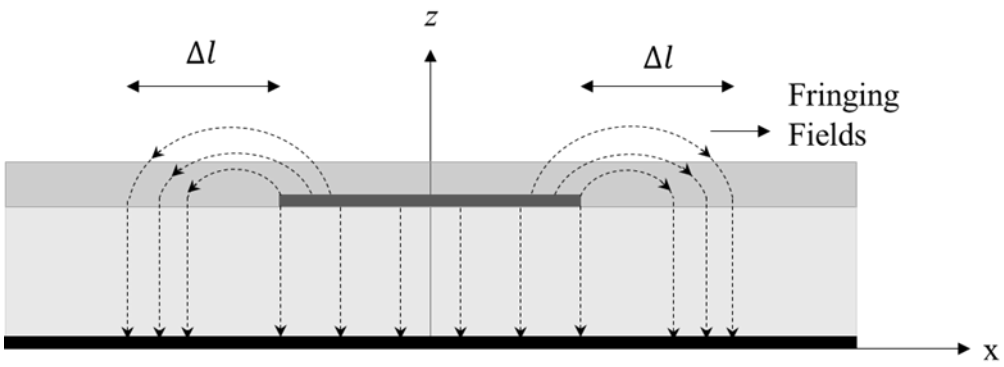


Figure 4.2 The illustration of radiated patch antenna.

$$\varepsilon_{eff} = \frac{\varepsilon_{sub} + 1}{2} + \frac{\varepsilon_{sub} - 1}{2}\left(1 + \frac{12h_{sub}}{W}\right)^{-\frac{1}{2}} \qquad (4.4.a)$$

$$\frac{\Delta l}{h_{sub}} = 0.412 \times \left(\frac{\varepsilon_{eff} + 0.300}{\varepsilon_{eff} - 0.258}\right) \times \left(\frac{\frac{W}{h_{sub}} + 0.264}{\frac{W}{h_{sub}} + 0.800}\right) \qquad (4.4.b)$$

$$L_{eff} = L + 2\Delta l \qquad (4.4.c)$$

The effective permitivity is a complex value for more realistic cases and can be obtained by (4.5) [62]. Here, [62] employed the effective loss tangent ($tan\delta_{eff}$) to represent the impact of all losses including the dielectric, conductor, radiation, surface wave, and leaky wave losses. Thus, ${\varepsilon_{eff}}^*$ is defined as the complex permivity which takes into account both radiated and disspated power.

$${\varepsilon_{eff}}^* = \varepsilon_{eff} - j\varepsilon_{eff}\, tan\delta_{eff} \qquad (4.5)$$

For a radiating antenna, the resonant frequencies of different modes are expressed as in (4.6), where *m* and *n* are the mode numbers of the cavity. These formulas have a limited accuracy even though they provide an analytical solution.

$$f_r = \frac{c\sqrt{\left(\frac{m}{L_{eff}}\right)^2 + \left(\frac{n}{W_{eff}}\right)^2}}{2\sqrt{\varepsilon_{eff}{}^*}} \qquad \begin{matrix} m = 1, 2, 3, \ldots \\ n = 0, 1, 2, \ldots \end{matrix} \tag{4.6}$$

For antennas with superstrates, like the model used in this thesis, more complex analytical solutions are required. The superstrate is a structure that contains one or more dielectric layers on the conductive patch. To represent the behavior of the superstrate better, the model is mapped onto a complex g-plane [60]. The filling fractions, q_i, of each dielectric layer, which is defined as the ratio of the area of the dielectric layers, are calculated to obtain the ε_{eff}. [63] provides the derivation of ε_{eff} as given in (A.9), based on the q_i of each dielectric layer. In this approach, L_{eff} is derived with (4.4.c) where Δl is calculated by (A.12). Finally, f_r is obtained by (4.6), but the solution of [63] could be applied to the fundamental mode only, where $m = 1, n = 0$. Moreover, the model of [63] does not consider the surface wave losses. [64] modified the calculation of L_{eff} to take the surface wave losses into account as given in Appendix B. Lastly, [65] extended the formulation of [64] to get more accurate L_{eff} in the presence of surface waves, and modified the calculation of ε_{eff} by considering leaky waves as stated in Appendix C. These analytical calculations give an approximate result for f_r of the fundamental mode, but the accuracy of these results are limited. In this chapter, these derivations have been adapted to the patch antenna model with three dielectric layers which are the substrate, the solder resist and air, and their solutions were compared to the cavity model-based NN predictions.

In addition to the resonant frequency, the cavity model gives an expression for the input impedance. The input impedance can be modelled as in Figure 4.3. This representation contains a series combination of parallel RLC circuit and the impedance of the probe. Here, R represents the radiation and dissipative losses, L represents inductive energy storage, and C represents the capacitive energy storage due to the interaction of metallic patch and ground plane. $R_L + jX_L$ introduces the impedance of the probe. In general, the probe feed has an inductive effect but, in the model used in this thesis, it could be capacitive due to the presence of the pad. At the fundamental mode, the input impedance is obtained by the formula (4.7) [67]. Here, R_{max} is the maximum of the real part of the input impedance, and the quality factor, Q_T, is used to introduce the impact of the losses. Therefore, it is derived from $tan\delta_{eff}$ and depends on the fractional bandwidth.

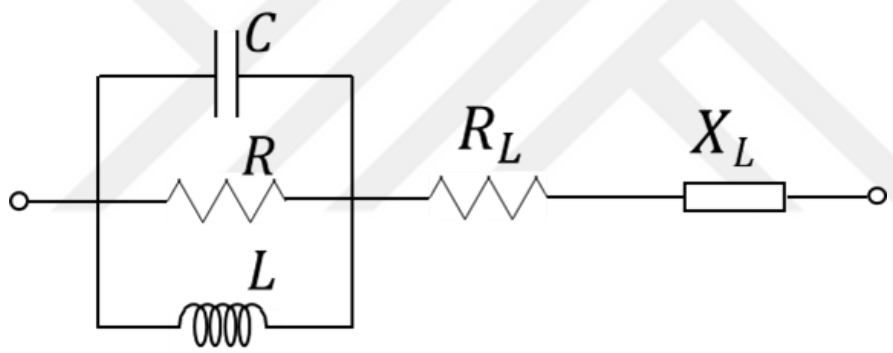


Figure 4.3 The input impedance model of the patch antenna.

$$Z_{in}(f) = \frac{R_{max}}{1 + {Q_T}^2\left(\frac{f}{f_r} - \frac{f_r}{f}\right)^2} + j\left[X_L - \frac{R_{max}Q_T\left(\frac{f}{f_r} - \frac{f_r}{f}\right)}{1 + {Q_T}^2\left(\frac{f}{f_r} - \frac{f_r}{f}\right)^2}\right] \tag{4.7}$$

where

$$Q_T = \frac{1}{tan\delta_{eff}} = \frac{1}{BW(\%)}$$

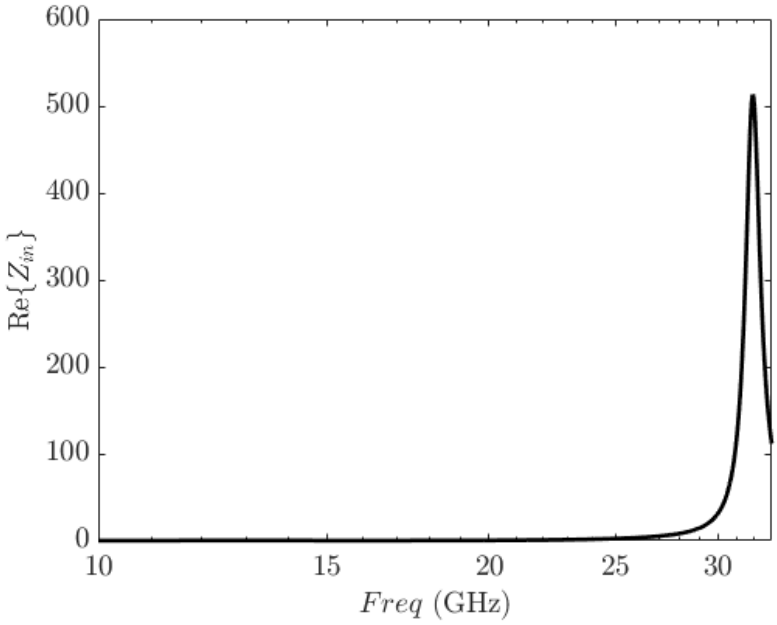
4.2 The Cavity Model-Based NN

This part explains the development of a novel NN by integrating these analytical calculations to create more accurate predictions of the performance of the patch antenna, such as the resonant frequency, fractional bandwidth and the input impedance for the fundamental mode. In this approach, NN merges the analytical formulation of the cavity model with the ability of generating input/output relationships. Since the quality of the data is very critical, to build the correct mapping between the input and output, the previous dataset, which was used to predict S_{11} parameters in the previous chapters, needs some modifications before implementing the cavity model-based NN. As mentioned before, (4.7) is a solution of the input impedance for the fundamental mode. S_{11} parameters can be derived from $Z_{in}(f)$ using (4.8). However, one needs the mode information to correctly implement the NN.

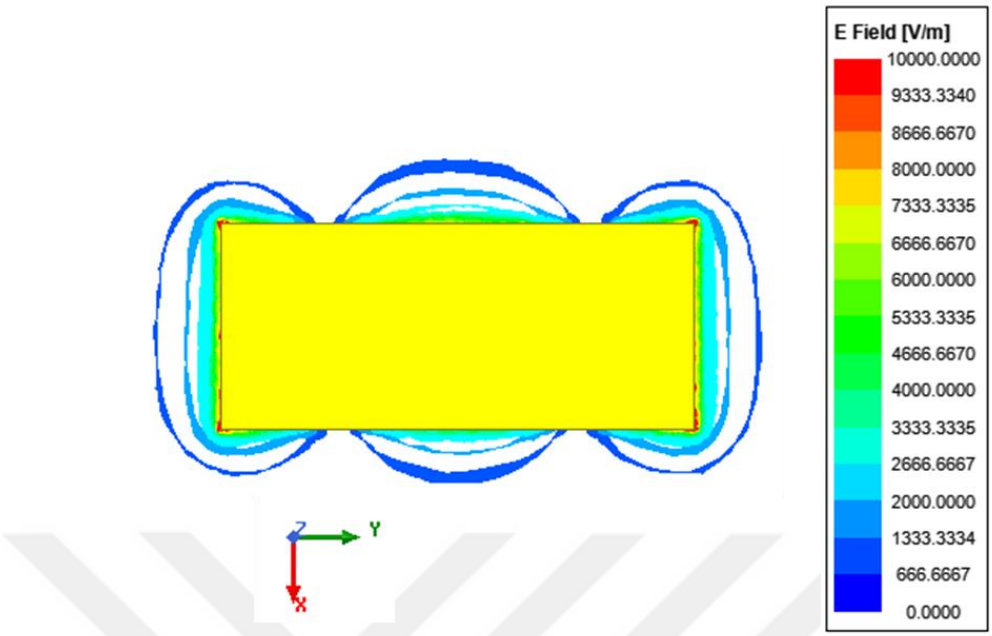
$$S_{11} = \frac{Z_{in} - Z_0}{Z_{in} + Z_0} \tag{4.8}$$

Therefore, a new dataset with 1465 data was generated by also considering the mode knowledge. To ensure the first resonance takes place at the first mode, the frequency range was expanded to 10-33 GHz. In this range, the samples with higher length than width ($L > W$) have f_r of the fundamental mode (10) at the first R_{max}. In an effort to create a more comprehensive dataset, the data with ($L < W$) were also considered. However, predicting the fundamental modes for this subset proved challenging for two reasons. Firstly, the excitation was consistently directed along the *x* coordinate, and hence, the (01) modes were not present in most of the cases. Secondly, for certain instances with dimensions ($2L < W$), the (02) modes resonanted before the (10) modes. For example, Figure 4.4 illustrates an antenna sample with the first resonant frequency occurring at 31.9 GHz but the fields exhibit a mode (02) distribution. To

obtain the first resonance information of $(L < W)$ data, the only viable method was to simulate these models at the frequency where first R_{max} is observed and visualize the E-field. By separating (10) modes, the new dataset was generated for the fundamental set, and it contains the information of f_r, $BW(\%)$ and the input impedance.



(a) The demonstration of $Re\{Z_{in}\}$ where its maxima occur at 31.90 GHz.



(b) The demonstration of fringing fields at 31.90 GHz.

Figure 4.4 The illustration of 02 modes for the sample with $(2L < W)$.

After collecting the information of the (10) modes, an NN was implemented to predict the input impedance and S parameters of the antenna. Firstly, an FCNN was created to give the fundamental resonant frequency and the relevant fractional bandwidth. The outputs of the FCNN were defined as the extension parameters of the length of the patch, Δl and permittivity of the substrate, $\Delta\varepsilon$, and the fractional bandwidth, $BW(\%)$. $BW(\%)$ was defined as the width between the frequency points where the input resistance is $1/\sqrt{2}$ of its maximum value. Based on the analytical calculations of the effective parameters, the outputs of the NN were used to obtain the predicted effective parameters, $\hat{L}_{eff}$ and $\hat{\varepsilon}_{eff}$, using the equations (4.9) and (4.10). These new equations were derived by considering a physical range which the predicted output needs to be fitted. For example, since the output of activation function is between 0 and 1, the effective permivity will be between the average of the permittivities of the substrate and air, and the average of the permittivities of the

substrate and solder resist. Finally, the fundamental resonant frequency was calculated by (4.6) for the fundamental mode where $m = 1,\ n = 0$.

$$\hat{L}_{eff} = L + 2\Delta l \tag{4.9}$$

$$\hat{\varepsilon}_{eff} = \frac{\varepsilon_{sub} + 1}{2} + \frac{\varepsilon_{sr} - 1}{2}\Delta\varepsilon \tag{4.10}$$

The FCNN, that predicts the mode-10 resonant frequencies and fractional bandwidth, has 3 hidden layers with 25 nodes each. The activation function is *tanh()* except for the last layer. At the output of the last layer, *sigmoid()* is used as the activation function, which is a natural choice to satisfy the physical requirements. Learning rate was optimized as 0.007, and it is reduced with the decay rate of 0.7 at every 250 epochs.

Moreover, another NN was implemented to predict the input impedance, which are used to obtain S_{11} parameters, and R_{max}. Since R_{max} is a single-valued variable, this part was predicted by an FCNN which was optimized with 2 hidden layers containing 25 nodes each. To predict the input impedance, the probe impedance, $(R_L + jX_L)$, were obtained as the output of STCNN. This network is an STCNN because the impedance of the probe is a multivalued vector as it changes with the frequency. Here, R_L does not only represent the dissipative loss, but also the impact of higher order modes and an additive term to improve the solutions of the cavity model. For the same reason, another parameter, Δf is predicted to improve the resonant frequency predictions when the cavity model is not sufficient to approximate the resonant frequency of the fundamental mode. The updated resonant frequency is expressed as $\tilde{f}_r = \hat{f}_r + \Delta f$. The STCNN has three fully connected layers with 25 nodes, and three convolutional layers with kernels of 21, 10, 3 and strides of 1, 2, 2, respectively. The learning rate was optimized as 0.009. Δf is obtained from the

output of the last FCNN layer before STCNN because it does not depend on the frequency and it should be adjusted with the input impedance prediction.

As a result, there are two networks, which are trained and optimized separately but updated with the common loss function (4.11) where λ_1 of 5, λ_2 of 15, λ_3 of 5 and λ_4 of 2, as illustrated in Figure 4.5. The obtained frequency with the predicted effective parameters, fractional bandwidth, R_{max} and the real and imaginary of S_{11} parameters were compared to the true output by employing Huber Loss as in (2.8) with the β of 0.001. Finally, these four losses were summed up and used to update the parameters.

$$Loss = \lambda_1 h(f_r, \hat{f}_r, \beta) + \lambda_2 h(BW, \widehat{BW}, \beta) + \lambda_3 h(S_{11}, \hat{S}_{11}, \beta) + \lambda_4 h(R_{max}, \hat{R}_{max}, \beta) \quad (4.11)$$

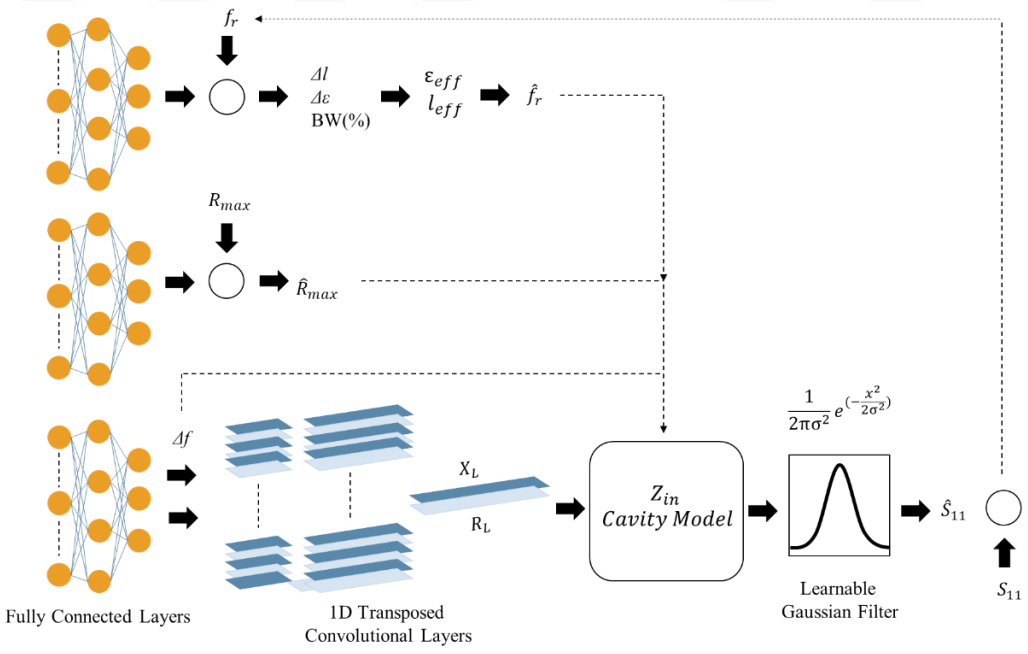


Figure 4.5 The illustration of the cavity model-based NN architecture.

Although only the fundamental mode is considered in the dataset, some of data exhibit the impact of the higher order modes. These samples distort the performance of the NN. Figure 4.6 illustrates one such sample, where the impact of the higher modes has altered the expected profile. Due to the rarity of these samples, they have been excluded from the dataset.

Table 4.1 shows the accuracy for the performance parameters of the antenna. The losses are listed in terms of MAE (2.6), which presents the absolute distance between the predicted and actual outputs. The resonant frequency can be predicted with an mean absolute percentage deviation of 0.57%, which is noticably good, when the broad range of the frequency is considered.

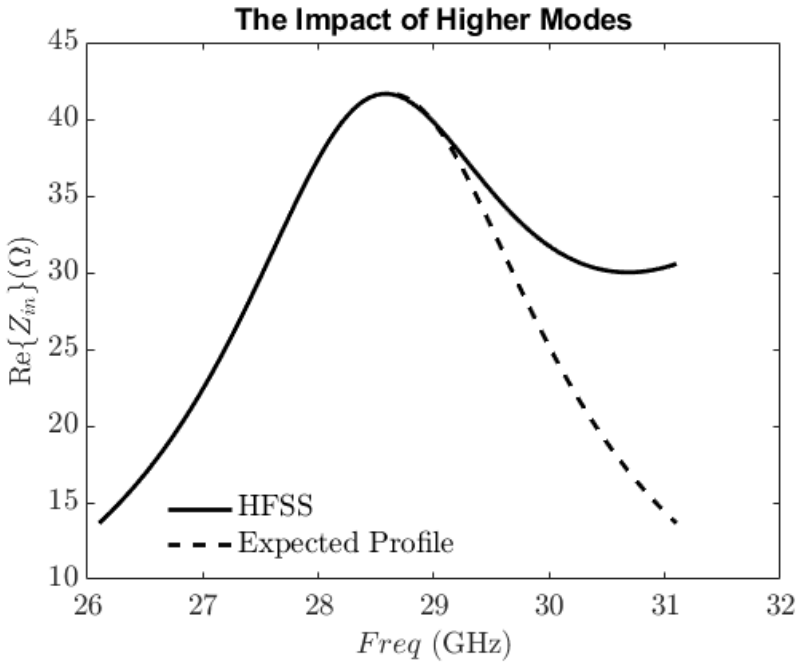


Figure 4.6 The illustration of the impact of higher modes.

Table 4.1 The MAE values of each performance parameters.

	MAE
f_r **(GHz)**	0.10510
$BW(\%)$	0.00270
R_{max}	0.03820
$Re\{Z_{in}(f)\}$	0.01630
$Im\{Z_{in}(f)\}$	0.01740

Figure 4.7 and 4.8 present the results for three antenna samples, Antennas XV-XVII, to illustrate the accuracy of the NN. The predictions have good correlations with the HFSS results. Figures 4.9 and 4.10 also demonsrate the S_{11} profiles of the same samples. The relevant RSE losses and the input parameters of the antennas are listed in Tables 4.2 and 4.3, respectively.

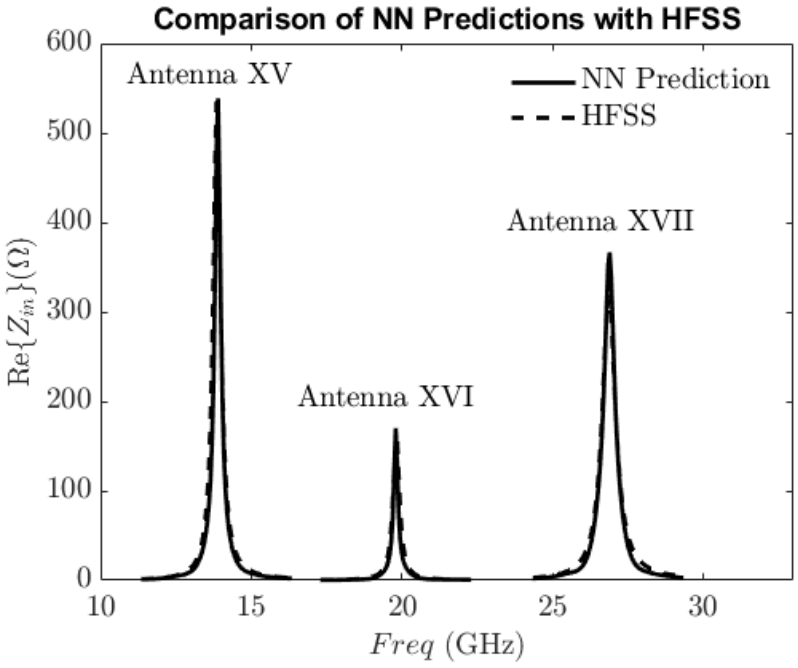


Figure 4.7 The comparison of NN predictions with HFSS for $\mathrm{Re}\{Z_{in}\}$ with RSEs of 0.04807, 0.06292, and 0.01910, respectively.

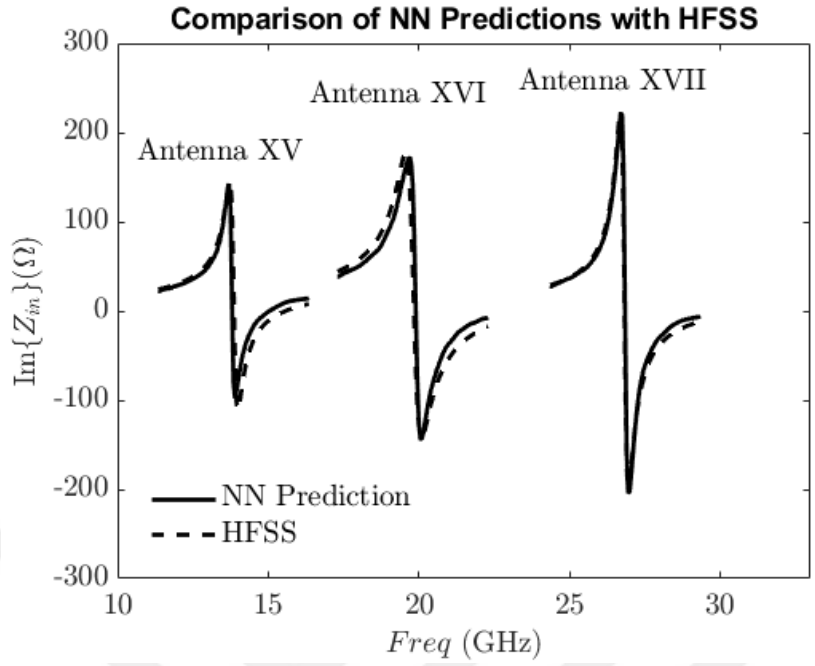


Figure 4.8 The comparison of NN predictions with HFSS for Im{Z_{in}} with RSEs of 0.02945, 0.05063, and 0.03762, respectively.

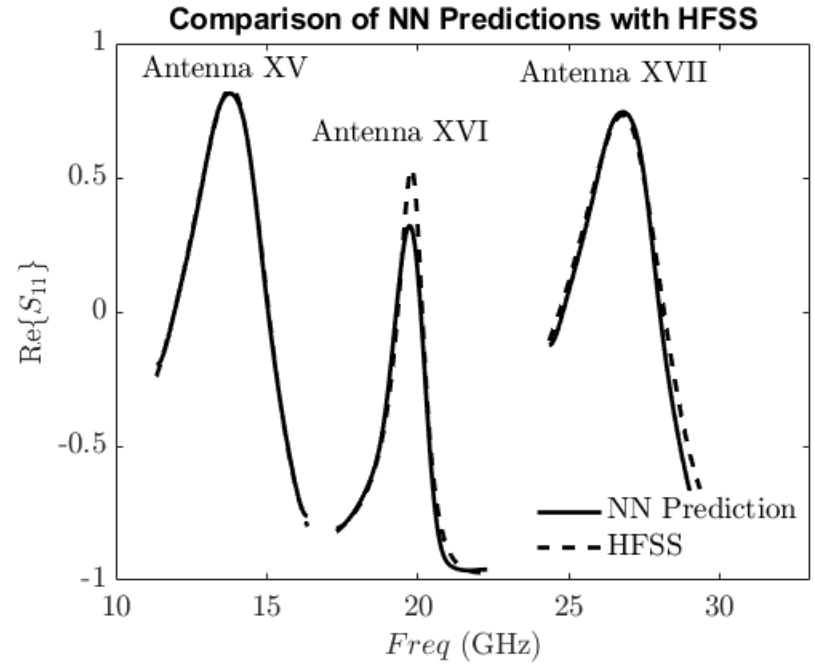


Figure 4.9 The comparison of NN predictions with HFSS for Re{S_{11}} with RSEs of 0.00054, 0.02346, and 0.03317, respectively.

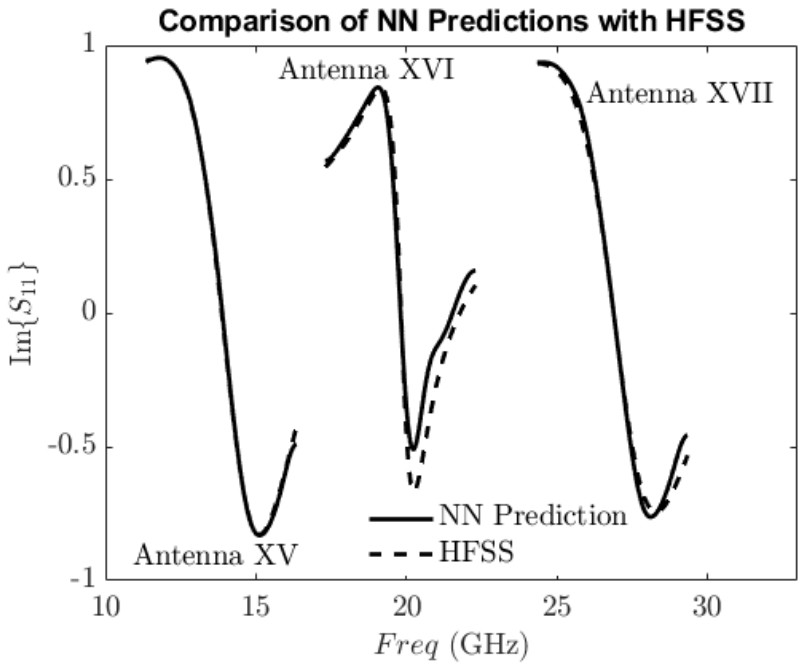


Figure 4.10 The comparison of NN predictions with HFSS for Im{S_{11}} with RSEs of 0.00030, 0.03582, and 0.00367, respectively.

Table 4.2 The RSE losses of the illustrated samples Antennas XV-XVII.

	Re{Z_{in}}	**Im{Z_{in}}**	**Re{S_{11}}**	**Im{S_{11}}**
Antenna XV	0.04807	0.02945	0.00054	0.00030
Antenna XVI	0.06292	0.05063	0.02346	0.03582
Antenna XVII	0.01910	0.03762	0.03317	0.00367

Table 4.3 The input parameters of the Antennas XV-XVII.

Input Parameter	**Antenna XV**	**Antenna XVI**	**Antenna XVII**
L (mm)	4.699	3.566	2.875
W (mm)	2.695	4.445	2.405
h_{cond} (mm)	0.020	0.011	0.023
h_{sub} (mm)	0.413	0.101	0.166
h_{sr} (mm)	0.063	0.025	0.090
r_{probe} (mm)	0.041	0.027	0.030
r_{pad} - r_{probe} (mm)	0.001	0.016	0.007
$r_{antipad}$ - r_{pad} (mm)	0.062	0.062	0.075
x_{probe} / L	0.386	0.368	0.359
ε_{sub}	4.511	4.148	3.136
ε_{sr}	4.373	3.288	2.800

Some predictions can be less accurate. Figures 4.11 and 4.12 show the worst predictions for Re{S_{11}} and Im{S_{11}}, and Tables 4.4 and 4.5 presents their input parameters.

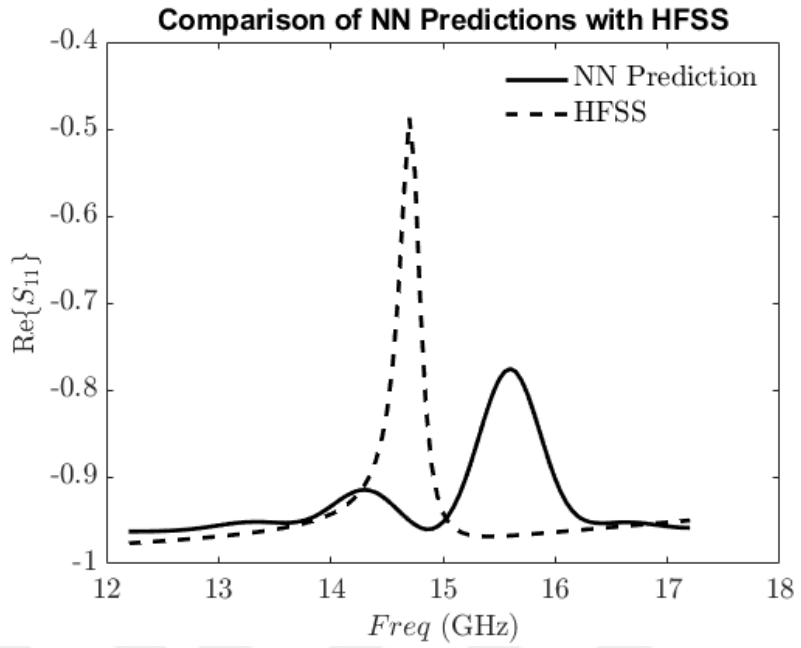


Figure 4.11 The comparison of the worst NN prediction with HFSS for Re{S_{11}} with an RSE of 1.47345.

Table 4.4 The input parameters of the worst NN prediction for Re{S_{11}}.

Input Parameter	**The Worst Prediction Of Re{S_{11}}**
L (mm)	4.469
W (mm)	4.227
h_{cond} (mm)	0.016
h_{sub} (mm)	0.111
h_{sr} (mm)	0.068
r_{probe} (mm)	0.044
r_{pad} - r_{probe} (mm)	0.022
$r_{antipad}$ - r_{pad} (mm)	0.043
x_{probe} / L	0.094
ε_{sub}	4.901
ε_{sr}	2.541

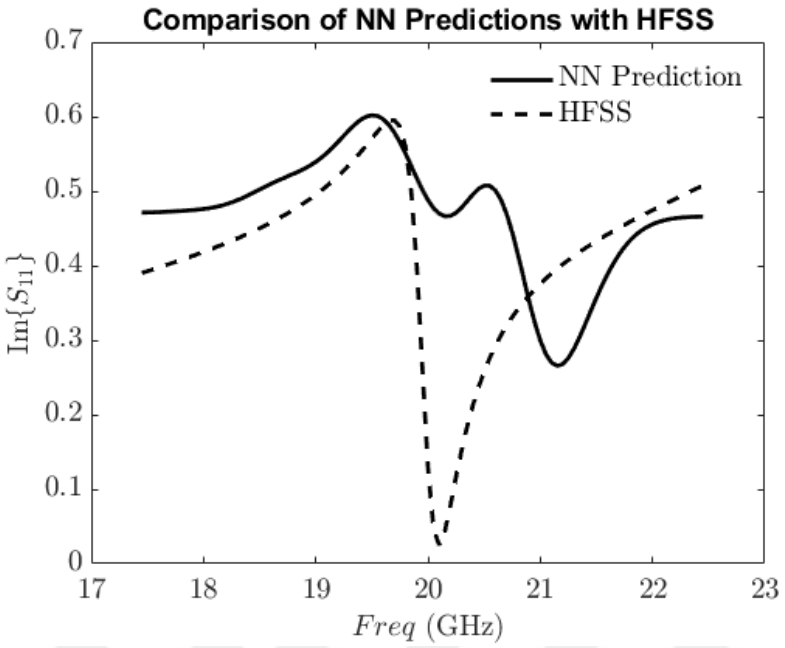


Figure 4.12 The comparison of the worst NN prediction with HFSS for Im{S_{11}} with an RSE of 1.29421.

Table 4.5 The input parameters of the worst NN prediction for Im{S_{11}}.

Input Parameter	**The Worst Prediction Of Im{S_{11}}**
L (mm)	3.220
W (mm)	4.833
h_{cond} (mm)	0.028
h_{sub} (mm)	0.108
h_{sr} (mm)	0.057
r_{probe} (mm)	0.032
r_{pad} - r_{probe} (mm)	0.022
$r_{antipad}$ - r_{pad} (mm)	0.043
x_{probe} / L	0.117
ε_{sub}	4.853
ε_{sr}	2.902

The RSE of S_{11} of the cavity model-based NN is compared with the RSE of S_{11} of the regularized NNs discussed in Chapter-3. The dataset of the cavity implementation has 1180 traning samples. As seen in Table 3.8, the best predictions were obtained by F-FFT with an RSE of 0.095, when the NN was trained with 2000 samples. Cavity model can achive a better accuracy with less data by including the physical behavior of the patch antenna into the picture. Lastly, the cavity model-based NN prevents passivity violations since it is built upon a physical model. The maximum magnitude of S_{11} parameters is less than unity for the whole test data.

4.3 Comparison of the NN Predictions with the Analytical Solutions

As mentioned before, the analytical expression for f_r can be obtained from the cavity model. The derivation of f_r as the solution of (4.6) is presented in Verma's formula [62]. This solution is based on a model that includes only the substrate and accounts for all losses. A more accurate approach for the patch antenna model is presented in Bernhard's formula [63], where f_r is determined by considering the presence of the superstrate. Additionally, Li's formula [64] introduce the effect of surface waves, while the improved Li's formula [65] provides a more correct calculation for L_{eff} and ε_{eff} by considering the leaky wave losses. In Table 4.6, the accuracies of these methods are summarized by comparing each other and also by comparing them with the cavity model-based NN predictions. The table also shows the worst resonant frequency predictions. As expected, with an increase in the considered mechanisms and model complexity, the accuracy also improves. Notably, the NN predictions demonstrate the best performance overall. For this dataset, the NN also exhibits the smallest maximum error. Although the MAE of [65] is relatively small, it's worth noting that the maximum loss value is still high. This underscores the superiority of the NN predictions in this particular scenario.

Table 4.6 The comparison of analytical formulations with NN predictions.

	f_r **(GHz)**	
	$\mathbf{mean}(\lvert f_r - \widehat{f_r} \rvert)$	$\mathbf{max}(\lvert f_r - \widehat{f_r} \rvert)$
Verma' Formula	2.08	6.87
Bernhard's Formula	1.47	2.84
Li's Formula	1.28	3.70
Improved Li's Formula	0.42	5.36
NN Prediction	0.11	1.45

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, the primary objective has been to predict the performance of the microstrip patch antenna by using its geometrical and material parameters. To achieve this, first a surrogate model in the form of a black box NN was developed to predict S_{11} parameters and the gain of the antenna. An STCNN architecture was developed to predict the return loss and gain of the patch antenna, which vary with frequency. The dataset used for training was generated by accounting high number of design parameters and a relatively wide frequency band to create a generalized model. STCNN allows the scalable frequency resolution at the output by adjusting the kernel sizes and strides of transposed convolutional layers. Additionally, the feeding network was also excluded from the model to be adjustable for different feeding networks.

The predictions of the black box NN have a good accuracy in the test dataset. Test data gives an idea how the predictions are accurate for unseen data. However the test data was also generated with the training data and its profile is not so different than the training data. This immediately puts the accuracy of the NN into question for an ill-conditioned problem. The complete evaluation of the confidence level of the NN and how well it can generalize for a completely new data, requires an uncertainty quantification, which can be performed via quantile regression approaches. Although this was beyond the scope of this thesis, the black box NN was tested with a separately generated smaller, to provide an insight about its generalization capabilities. .

Throughout the investigation, the significance of the loss function was given paramount importance, leading to an in-depth exploration and comparison of various implementations. Among the different options evaluated, the Huber loss function

emerged as the most effective one in updating the unknown parameters of the NN, ensuring more accurate predictions. Moreover, the RSE was identified as the optimal evaluation metric, enabling a comprehensive assessment of the model's performance.

To further enhance the NN's performance, a diverse set knowledge-based regularization methods were applied. Through observations, it was discovered that integrating the derivative or spectral domain information of the output into the loss function effectively eliminates noise and irrelevant perturbations on the predictions, contributing to the model's overall accuracy and reliability. Also, these regularizations increase the smoothness of the output. Moreover, magnitude regularization was also integrated into the loss function to prevent passivity violations. This approach helps to decrease the amount non-physical predictions.

As a result, a powerful and robust surrogate model for predicting the performance of microstrip patch antennas was successfully implemented. The comprehensive exploration of the loss functions and the novel utilization of the knowledge-based regularization techniques paved the way for more accurate and reliable predictions, opening new possibilities for optimizing antenna designs and applications.

Furhermore, the cavity model of the patch antenna was seamlessly integrated into the NN architecture. The incorporation of information from this simplified model proved to be instrumental in achieving remarkably accurate predictions for the performance of the patch antenna, all while significantly reducing the size of the required dataset. Moreover, the integration of the cavity model completely prevents the non-physical predictions with passivity violations.

However, the utilization of the analytical equations associated with the cavity model requires a more constrained dataset. As elucidated in Chapter 4, the accurate determination of the fundamental resonant frequency heavily relies on the knowledge of the specific modes of the antenna. When dealing with data where the width exceeds the length $L < W$, the prediction of the point where R_{max} occurs becomes rather intricate. As a remedy to this challenge, modes were aptly defined for each sample falling under the category of $L < W$. Crafting this modified dataset

posed the most formidable and demanding aspect of this approach, requiring keen attention to detail and precision.

Overall, this thesis presents a comprehensive approach to predict microstrip patch antenna performance, leveraging an STCNN, knowlegde-based regularization techniques, and physical awareness through the integration of the cavity model. By utilizing these surrogate model alternatives, accurate predictions are achieved while maintaining a high level of generality. This means a wide spectrum of design parameters are considered within a relatively broad range, providing extensive alternatives for antenna designs. The proposed surrogate models can be used as a fast and accurate predictive tool at the early stages of the design process.

As a future work, the generalization of the problem and the performance of the NN can be improved further. For instance, the wave equations can be direclty integrated into the loss function to develop a physics informed NN which can generate more accurate predictions while requiring less data. It can also lead to predict outside of the specified dataset. With the help of pyhsical structure, the shapes of antennas can be expanded instead of using just rectangular patch. The NN predictions can be forced to capture the performance for arbitrarily shape of antennas. For this case, the radiation pattern of the predictions can also be predicted for the different shapes of antennas. Furthermore, the optimization algorithms can be incorporated into the output of the NN to enhance the selection of optimal solutions. In light of all these improvements, the research can evolve into patch array design by incorporating phase shift predictions.

As another future work, the cavity model-based NN can be forced to predict the performance for higher modes. This can be achieved by generating a new dataset containing higher modes. On the other hand, a pyhsics informed NN can be applied with the cavity model to capture higher modes without generating a new dataset.

Finally, quantile regression can be implemented to the NN to understand the variations of the predicted variables in the different parts of data distribution. Quantile regression is interested in the quantiles of the data while the conventional

NN is just considering the mean of the data during the training process. Therefore, it can represent the relation between input/output more effectively. By integrating quantile regression with the uncertainty qualification, which was introduced before, creates a powerful framework for understanding the distribution of the data. This way, the measure of uncertainty is incorporated around the quantiles, and the quantile values are modelled with quantifying the uncertainty associated with those values.

REFERENCES

[1] S. Koziel, A. Pietrenko-Dabrowska, Performance-Driven Surrogate Modeling of High-Frequency Structures, Springer, doi: 10.1007/978-3-030-38926-0

[2] Qi-Jun Zhang, K. C. Gupta and V. K. Devabhaktuni, "Artificial NNs for RF and microwave design - from theory to practice," in IEEE Transactions on Microwave Theory and Techniques, vol. 51, no. 4, pp. 1339-1350, April 2003.

[3] A. H. Zaabab, Qi-Jun Zhang and M. Nakhla, "A NN modeling approach to circuit optimization and statistical design," in IEEE Transactions on Microwave Theory and Techniques, vol. 43, no. 6, pp. 1349-1358, June 1995.

[4] P. M. Watson and K. C. Gupta, “Design and optimization of CPW circuits using EM-ANN models for CPW components,” IEEE Transactions on Microwave Theory and Techniques, vol. 45, no. 12, pp. 2515-2523, Dec. 1997.

[5] G. L. Creech, B. J. Paul, C. D. Lesniak, T. J. Jenkins and M. C. Calcatera, "Artificial NNs for fast and accurate EM-CAD of microwave circuits," in IEEE Transactions on Microwave Theory and Techniques, vol. 45, no. 5, pp. 794-802, May 1997.

[6] P. M. Watson and K. C. Gupta, "EM-ANN models for microstrip vias and interconnects in dataset circuits," in IEEE Transactions on Microwave Theory and Techniques, vol. 44, no. 12, pp. 2495-2503, Dec. 1996.

[7] L. Zhang, H. Kabir, R. Sweeney and K. Kim, “Enabling Automatic Model Generation of RF Components: A Practical Application of NNs,” IEEE Microwave Magazine, vol. 22, no. 10, pp. 51-61, Oct. 2021.

[8] H. El Misilmani, T. Naous, S. Al Khatib, “A Review on the Design and Optimization of Antennas Using Machine Learning Algorithms and

Technique,” International Journal of RF and Microwave Computer-Aided Engineering, 2020.

[9] D. Shi, C. Lian, K. Cui, Y. Chen and X. Liu, "An Intelligent Antenna Synthesis Method Based on Machine Learning," in IEEE Transactions on Antennas and Propagation, vol. 70, no. 7, pp. 4965-4976, July 2022.

[10] W. Chen, Q. Wu, C. Yu, H. Wang and W. Hong, "Multibranch Machine Learning-Assisted Optimization and Its Application to Antenna Design," in IEEE Transactions on Antennas and Propagation, vol. 70, no. 7, pp. 4985-4996, July 2022.

[11] Q. Wu, W. Chen, C. Yu, H. Wang and W. Hong, "Multilayer Machine Learning-Assisted Optimization-Based Robust Design and Its Applications to Antennas and Array," in IEEE Transactions on Antennas and Propagation, vol. 69, no. 9, pp. 6052-6057, Sept. 2021.

[12] D. R. Prado, J. A. López-Fernández, M. Arrebola and G. Goussetis, "Support Vector Regression to Accelerate Design and Crosspolar Optimization of Shaped-Beam Reflectarray Antennas for Space Applications," in IEEE Transactions on Antennas and Propagation, vol. 67, no. 3, pp. 1659-1668, March 2019.

[13] D. R. Prado, J. A. López-Fernández, M. Arrebola, M. R. Pino and G. Goussetis, "Wideband Shaped-Beam Reflectarray Design Using Support Vector Regression Analysis," in IEEE Antennas and Wireless Propagation Letters, vol. 18, no. 11, pp. 2287-2291, Nov. 2019.

[14] G. Oliveri, M. Salucci and A. Massa, "Towards Efficient Reflectarray Digital Twins - An EM-Driven Machine Learning Perspective," in IEEE Transactions on Antennas and Propagation, vol. 70, no. 7, pp. 5078-5093, July 2022.

[15] C. Gianfagna, M. Swaminathan, P. M. Raj, R. Tummala and G. Antonini, "Enabling antenna design with nano-magnetic materials using machine

learning," 2015 IEEE Nanotechnology Materials and Devices Conference (NMDC), Anchorage, AK, USA, 2015, pp. 1-5.

[16] Y. Sharma, H. H. Zhang and H. Xin, "Machine Learning Techniques for Optimizing Design of Double T-Shaped Monopole Antenna," in IEEE Transactions on Antennas and Propagation, vol. 68, no. 7, pp. 5658-5663, July 2020.

[17] G. Fedi, S. Manetti, G. Pelosi and S. Selleri, "Profiled corrugated circular horns analysis and synthesis via an artificial NN," in IEEE Transactions on Antennas and Propagation, vol. 49, no. 11, pp. 1597-1602, Nov. 2001.

[18] S. K. Jain, "Bandwidth Enhancement Of Patch Antennas Using NN Dependent Modified Optimizer," International Journal of Microwave and Wireless Technologies, vol. 8, no. 7, pp. 1111–1119, 2016.

[19] F. -C. Chu, S. -C. Hsieh and C. -C. Wang, "Simulation and Measurement Work for 28GHz AiP Design of 5G Communication Application," in 2021 16th International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT), 2021, pp. 135-138.

[20] W. Hong, A. Goudelev, K. -h. Baek, V. Arkhipenkov and J. Lee, "24-Element Antenna-in-Package for Stationary 60-GHz Communication Scenarios," IEEE Antennas and Wireless Propagation Letters, vol. 10, pp. 738-741, 2011.

[21] Y. -W. Lu, B. -S. Fang, H. -H. Mi and K. -T. Chen, "Mm-Wave Antenna in Package (AiP) Design Applied to 5th Generation (5G) Cellular User Equipment Using Unbalanced Substrate," in 2018 IEEE 68th Electronic Components and Technology Conference (ECTC), 2018, pp. 208-213.

[22] Kaur, Sukhdeep & Khanna, Rajesh & SAHNI, Dr & Kumar, Naveen. (2019). Design and Optimization of Microstrip Patch Antenna using Artificial NNs. International Journal of Innovative Technology and Exploring Engineering.

[23] V. S. Kushwah and G. S. Tomar, "Design of Microstrip Patch Antennas Using NN," 2009 Third Asia International Conference on Modelling & Simulation, Bundang, Indonesia, 2009, pp. 720-724.

[24] Thakare, V.V. and Singhal, P. (2010), Microstrip antenna design using artificial NNs. Int J RF and Microwave Comp Aid Eng, 20: 76-86.

[25] Türker, Nurhan & Güneş, Filiz & Yildirim, Tülay. (2006). Artificial neural design of microstrip antennas. Turkish Journal of Electrical Engineering and Computer Sciences. 14. 445-453.

[26] Sağiroğlu, Ş., Güney, K. and Erler, M. (1998), Resonant frequency calculation for circular microstrip antennas using artificial NNs. Int J RF and Microwave Comp Aid Eng, 8: 270-277.

[27] Sagiroglu, S. and Güney, K. (1997), Calculation of resonant frequency for an equilateral triangular microstrip antenna with the use of artificial NNs. Microw. Opt. Technol. Lett., 14: 89-93.

[28] H. Kumawat and P. Agarwal, "Artificial NN Model to Predict the Design Parameters of Inset-fed Microstrip Patch Antenna," 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2021, pp. 943-948.

[29] I. Vilović, N. Burum and M. Brailo, "Microstrip antenna design using NNs optimized by PSO," ICECom 2013, Dubrovnik, Croatia, 2013, pp. 1-4.

[30] T. M. Neebha and M. Nesasudha, "Artificial NN based design of a microstrip patch antenna for RADAR applications," 2017 First International Conference on Recent Advances in Aerospace Engineering (ICRAAE), Coimbatore, India, 2017, pp. 1-6.

[31] S. K. Jain, S. N. Sinha and A. Patnaik, "Analysis of coaxial fed dual patch multilayer x/ku band antenna using artificial NNs," 2009 World Congress on

Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 2009, pp. 1111-1114.

[32] S. Koziel, N. Çalık, P. Mahouti and M. A. Belen, "Low-Cost and Highly Accurate Behavioral Modeling of Antenna Structures by Means of Knowledge-Based Domain-Constrained Deep Learning Surrogates," in IEEE Transactions on Antennas and Propagation, vol. 71, no. 1, pp. 105-118, Jan. 2023.

[33] S. Koziel, N. Çalık, P. Mahouti and M. A. Belen, "Accurate Modeling of Antenna Structures by Means of Domain Confinement and Pyramidal Deep NNs," in IEEE Transactions on Antennas and Propagation, vol. 70, no. 3, pp. 2174-2188, March 2022.

[34] K. Hornik, M.B. Stinchcombe, H.L. White, “Multilayer feedforward networks are universal approximators,” NNs, vol. 2, pp. 359-366.

[35] [Online]. Available: https://keras.io/api/optimizers/adam/

[36] S. Haykin, B. Kosko, "GradientBased Learning Applied to Document Recognition," in Intelligent Signal Processing, IEEE, 2001, pp.306-351.

[37] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” Jan. 2018, arXiv:1603.0728. [Online]. Available: http://arxiv.org/abs/1603.07285

[38] Y. Sugawara, S. Shiota and H. Kiya, “Super-Resolution Using Convolutional NNs Without Any Checkerboard Artifacts,” in 2018 25th IEEE International Conference on Image Processing (ICIP), 2018, pp. 66-70,

[39] F. Sahito, P. Zhiwen, et al., “Transpose convolution based model for super-resolution image reconstruction,” Appl Intell 53, 2023.

[40] J. G. Andrews et al., "What Will 5G Be?," in IEEE Journal on Selected Areas in Communications, vol. 32, no. 6, pp. 1065-1082, June 2014.

[41] Z. Pi and F. Khan, ‘‘An introduction to millimeter-wave mobile broadband systems,’’ IEEE Commun. Mag., vol. 49, no. 6, pp. 101–107, Jun. 2011.

[42] T. S. Rappaport et al., “Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!,” IEEE Access, vol. 1, pp. 335-349, 2013.

[43] J. Laskar et al., "Review of RF packaging research at Georgia Tech's PRC, "Proceedings. 4th International Symposium on Advanced Packaging Materials Processes, Properties and Interfaces (Cat. No.98EX153), Braselton, GA, USA, 1998, pp. 139-150.

[44] Y. Zhang and J. Mao, "An Overview of the Development of Antenna-in-Package Technology for Highly Integrated Wireless Devices," in Proceedings of the IEEE, vol. 107, no. 11, pp. 2265-2280, Nov. 2019.

[45] C. Zhu, Y. Wang, Z. Duan and Y. Dai, "Design of Patch Antenna in Embedded Glass Fan Out Package for 77-GHz Automotive Radar," 2020 IEEE Asia-Pacific Microwave Conference (APMC), Hong Kong, Hong Kong, 2020, pp. 1066-1068.

[46] Kyutae Lim, A. Obatoyinbo, M. Davis, J. Laskar and R. Tummala, "Development of planar antennas in multi-layer packages for RF-system-on-a-package applications," IEEE 10th Topical Meeting on Electrical Performance of Electronic Packaging (Cat. No. 01TH8565), Cambridge, MA, USA, 2001, pp. 101-104.

[47] S. -H. Wi et al., "Package-Level Integrated LTCC Antenna for RF Package Application," in IEEE Transactions on Advanced Packaging, vol. 30, no. 1, pp. 132-141, Feb. 2007.

[48] C. Zhu and Z. Duan, "Design of 94-GHz Wideband Waveguide-Feed Patch Antenna and Array in eWLB Package," 2020 50th European Microwave Conference (EuMC), Utrecht, Netherlands, 2021, pp. 824-827.

[49] G. Huang, Y. Dai, N. Liu, Z. Duan, Y. Wu and T. Lin, "A low cost 60GHz antenna in Fan-Out Panel Level Package for millimeter-wave radar application," 2020 21st International Conference on Electronic Packaging Technology (ICEPT), Guangzhou, China, 2020, pp. 1-4.

[50] A. Raeesi et al., "A Low-Profile 2D Passive Phased-Array Antenna-in-Package for Emerging Millimeter-Wave Applications," in IEEE Transactions on Antennas and Propagation, vol. 71, no. 1, pp. 1093-1098, Jan. 2023.

[51] T. Deckmyn, M. Cauwe, D. Vande Ginste, H. Rogier and S. Agneessens, "Dual-Band (28,38) GHz Coupled Quarter-Mode Substrate-Integrated Waveguide Antenna Array for Next-Generation Wireless Systems," IEEE Transactions on Antennas and Propagation, vol. 67, no. 4, pp. 2405-2412, April 2019.

[52] [Online]. Available: http://www.ansoft.com/products/hf/hfss/

[53] [Online]. Available: https://www.mathworks.com/help/stats/lhsdesign.html

[54] Jadon, A., Patil, A., & Jadon, S. (2022). A Comprehensive Survey of Regression Based Loss Functions for Time Series Forecasting. arXiv preprint arXiv:2211.02989.

[55] E. Phaisangittisagul, "An Analysis of the Regularization Between L2 and Dropout in Single Hidden Layer Neural Network," 2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS), Bangkok, Thailand, 2016, pp. 174-179.

[56] Q. Li, M. Yan and J. Xu, "Optimizing Convolutional Neural Network Performance by Mitigating Underfitting and Overfitting," 2021 IEEE/ACIS 19th International Conference on Computer and Information Science (ICIS), Shanghai, China, 2021, pp. 126-131.

[57] D. M. Pozar, Microwave Engineering, New York, AddisonWesley, 1990.

[58] C. A. Balanis, Antenna Theory Analysis and Design. Hoboken, NJ: Wiley, 2016.

[59] K.F. Lee, K.M. Luk, Microstrip Patch Antennas. Imperial College Press, 2011.

[60] H. A. Wheeler, “Transmission-line properties of parallel strips separated by a dielectric sheet,” IEEE Trans. Microwave Theory Tech., vol. 13, pp. 172–185, Mar. 1965.

[61] M. Kirschning, R. H. Jansen, and N. H. L. Koster, “Accurate model for open end effect of microstrip lines,” Electron. Lett., vol. 17, pp. 123–125, Feb. 1981.

[62] A. K. Verma and Nasimuddin, "Resonance frequency and bandwidth of rectangular microstrip antenna on thick substrate," in IEEE Microwave and Wireless Components Letters, vol. 12, no. 2, pp. 60-62, Feb. 2002.

[63] J. T. Bernhard and C. J. Tousignant, “Resonant frequencies of rectangular microstrip antennas with flush and spaced dielectric superstrates,” IEEE Trans. Antennas Propag., vol. 47, pp. 302–308, Feb. 1999.

[64] Li, Y. and N. Bowler, “Resonant frequency of a rectangular patch sensor covered with multilayered dielectric structures,” IEEE Trans. Antennas Propagat., Vol. 58, 1883–1889, 2010.

[65] Manotosh Biswas Sourav Banik and Mihir Dam, "Li's Formula Extended to Determine Accurate Resonant Frequency of a Rectangular Patch Antenna in Multi-Dielectric Layers," Progress In Electromagnetics Research M, Vol. 99, 177-189, 2021.

[66] D. Guha, “Resonant frequency of circular microstrip antennas with and without air gaps,” IEEE Trans. Antennas Propag., vol. 49, pp. 55–59, Jan. 2001.

[67] Nasimuddin, Nasimuddin & Verma, A.K. (2004). Multilayer Cavity Model for Microstrip Rectangular and Circular Patch Antenna. Electromagnetics.

[68] Chattopadhyay, S., B. Biswas, J. Y. Siddiqui, and D. Guha, “Rectangular microstrips with variable air gap and varying aspect ratio: Improved formulations and experiments,” Microw Opt. Tech. Lett., Vol. 51, 169–173, 2009.

APPENDICES

A. Bernhard's Formula [63]

$$W_{eff} = \sqrt{\frac{\varepsilon_r'}{\varepsilon_{eff}}}\left[W + 0.882h_{sub} + 0.164\frac{h_{sub}(\varepsilon_r' - 1)}{(\varepsilon_r')^2}\right] + \sqrt{\frac{\varepsilon_r'}{\varepsilon_{eff}}}\frac{h_{sub}(\varepsilon_r' + 1)}{\pi\varepsilon_r'}\left[\ln\left(\frac{W}{2h_{sub}} + 0.94\right) + 1.451\right] \quad (A.1)$$

$$V_e = \frac{2h_{sub}}{\pi}\arctan\left(\frac{\pi}{\frac{\pi}{2}\frac{W_{eff}}{h_{sub}} - 2}\left(\frac{h_{sr}}{h_{sub}} - 1\right)\right) \quad (A.2)$$

To calculate W_{eff}, two step iteration is applied. At the first step, ε_r' is approximated to ε_{sub} and ε_{eff} is approximated to ε_r'. Then, these values are used to calculate W_{eff}.

This value of W_{eff} is used to calculate ε_{eff} by (A.9). Then, with this ε_{eff}, the value of ε_r' are obtained by (A.10). Lastly, W_{eff} is secondly obtained with the new values of ε_r' and ε_{eff}. Then, ε_{eff} is calculated by using the last value of W_{eff}.

$$q_1 = 1 - \frac{h_{sub}}{2W_{eff}}\ln\left(\frac{\pi}{h_{sub}}W_{eff} - 1\right) \quad (A.3)$$

$$q_2 = 1 - q_1 - \frac{h_{sub} - V_e}{2W_{eff}}$$

$$x \ln\left(\pi \frac{W_{eff}}{h_{sub}} \frac{\cos\left(\frac{V_e \pi}{2h_{sub}}\right)}{\pi\left(\frac{h_{sr}}{h_{sub}} - \frac{1}{2}\right) + \frac{V_e \pi}{2h_{sub}}} + \sin\left(\frac{V_e \pi}{2h_{sub}}\right) \right) \qquad (A.4)$$

$$q_3 = \frac{h_{sub} - V_e}{2W_{eff}} x \ln(\pi \frac{W_{eff}}{h_{sub}} \frac{\cos\left(\frac{V_e \pi}{2h_{sub}}\right)}{\pi\left(\frac{h_{sr}}{h_{sub}} - \frac{1}{2}\right) + \frac{V_e \pi}{2h_{sub}}} + \sin\left(\frac{V_e \pi}{2h_{sub}}\right) \qquad (A.5)$$

$$q_4 = \frac{h_{sub}}{2W_{eff}} \ln\left(\frac{\pi}{2} - \frac{h_{sub}}{2W_{eff}}\right) \qquad (A.6)$$

$$q_1' = q_1 - q_4 \qquad (A.7)$$

$$q_2' = 1 - q_1' - q_3 - 2q_4 \qquad (A.8)$$

$$\varepsilon_{eff} = \varepsilon_{sub} q_1' + \varepsilon_{sub}(1 - q_1')^2 \times N/D \qquad (A.9)$$

where

$$N = {\varepsilon_{sr}}^2 q_2' q_3 + \varepsilon_{sr}\varepsilon_{air}(q_2' q_4 + (q_3 + q_4)^2)$$

$$D = {\varepsilon_{sr}}^2 q_2' q_3 q_4 + \varepsilon_{sub}(\varepsilon_{sr} q_3 + \varepsilon_{air} q_4)(1 - q_1 + q_4)^2 + \varepsilon_{sr}\varepsilon_{air} q_4(q_2' q_4 + (q_3 + q_4)^2)$$

$$\varepsilon_r' = \frac{2\varepsilon_{eff} - 1 + A}{1 + A} \qquad (A.10)$$

where

$$A = \left(1 + \frac{h_{sub}}{W_{eff}}\right)^{-\frac{1}{2}}$$

The L_{eff} is calculated by (A.11) by using (A.12) from [61].

$$L_{eff} = L + 2\Delta l \qquad (A.11)$$

$$\frac{\Delta l}{h_{sub}} = \frac{\xi_1 \xi_3 \xi_5}{\xi_4} \qquad (A.12)$$

with

$$\xi_1 = 0.434907 \frac{{\varepsilon_{eff}}^{0.81} + 0.26}{{\varepsilon_{eff}}^{0.81} - 0.189} \frac{\left(\frac{W}{h_{sub}}\right)^{0.8544} + 0.26}{\left(\frac{W}{h_{sub}}\right)^{0.8544} + 0.87}$$

$$\xi_2 = 1 + \frac{\left(\frac{W}{h_{sub}}\right)^{0.371}}{2.358\varepsilon_{sub} + 1}$$

$$\xi_3 = 1 + \frac{0.5274 \arctan\left(0.084 \left(\frac{W}{h_{sub}}\right)^{\frac{1.9413}{\xi_2}}\right)}{{\varepsilon_{eff}}^{0.9236}}$$

$$\xi_4 = 1 + 0.0377 \arctan\left(0.067 \left(\frac{W}{h_{sub}}\right)^{1.456}\right)\left(6 - 5e^{(0.036(1-\varepsilon_{sub}))}\right)$$

$$\xi_5 = 1 - 0.218 e^{\left(-7.5\frac{W}{h_{sub}}\right)}$$

B. Li's Formula [64]

In this method, the L_{eff} are obtained differently as in [66].

$$L_{eff} = L\sqrt{(1+q)} \tag{B.1}$$

$$\alpha = \frac{L}{\pi - 2} \tag{B.2}$$

$$\varepsilon_{re} = \frac{\varepsilon_{sub}}{\varepsilon_{eff}} \tag{B.3}$$

$$q = u + v + uv \tag{B.4}$$

with

$$u = \frac{1+\varepsilon_{re}}{\varepsilon_{re}} \frac{4}{\frac{\pi\alpha}{h_{sub}}}$$

$$v = \frac{2}{3t} \frac{\ln(p)}{\frac{\pi\alpha}{h_{sub}}} + \frac{\frac{1}{t} - 1}{g}$$

$$t = 0.37 + 0.63\varepsilon_{re}$$

$$p = \frac{1 + 0.8(\alpha/h_{sub})^2 + (0.31\alpha/h_{sub})^4}{1 + \frac{0.9\alpha}{h_{sub}}}$$

$$g = 4 + \frac{2.6\alpha}{h_{sub}} + \frac{2.9h_{sub}}{\alpha}$$

C. The extended Li's Formula [65]

In this method, L_{eff} is defined as in (A.11). But, the calculation of Δl is different, which is obtained from [68].

$$\Delta l = \frac{\pi\alpha\left(\sqrt{(1+q)}-1\right)}{2\left(2.5-\frac{0.5W}{L}\right)} \qquad (C.1)$$

$$q = q_1 + (1+q_1)(q_2 + q_3) \qquad (C.2)$$

with

$$q_1 = u$$

$$q_2 = \frac{2}{3}\left(\frac{1}{t}\right)\left(\frac{1}{\left(8+\frac{\pi\alpha}{h_{sub}}\right)} x \ln\left(\frac{1}{p}\right)\right)$$

$$q_3 = \left(\frac{1}{g}\right)\left(\frac{1}{t}-1\right)$$