ABSTRACT

This project presents a physics-constrained deep learning approach for the inverse design of rectangular microstrip patch antennas. A synthetic dataset of 30 000 samples—covering frequencies from 1 to 12 GHz and dielectric constants 2.2–12—was generated using classical patch-design formulas augmented by 1.5 % Gaussian noise to mimic manufacturing tolerances.

A physics-informed neural network (PINN) with a hybrid loss (mean squared error + penalty enforcing W≥1.05 LW \ge 1.05\,LW≥1.05L) was trained using TensorFlow and Keras-Tuner. After hyperparameter optimization and early stopping, the model achieved: Mean Absolute Error (MAE): 0.12 mm, Physics Compliance: 99.3 %, Speedup: ≈ 20× faster than full-wave EM simulation.

The entire pipeline (data generation → model training → deployment) is implemented in Python, and a Gradio web interface enables real-time antenna dimension prediction and visualization. This framework bridges theoretical electromagnetics with rapid prototyping, paving the way for automated design in 5G, IoT, and satellite applications.

CHAPTER 1

INTRODUCTION

1.1 Background

The proliferation of wireless communication technologies necessitates the development of compact, efficient, and high-performance antennas. Microstrip patch antennas are favored for their low profile, ease of fabrication, and compatibility with integrated circuits, making them suitable for mobile devices, satellite communication, and emerging technologies like the Internet of Things (IoT) and 5G networks.

Traditional design methods for microstrip patch antennas rely on iterative processes involving electromagnetic (EM) simulations and empirical formulas. While effective, these methods are time-consuming and computationally intensive, especially when exploring vast design spaces or optimizing multiple parameters simultaneously. Achieving desired performance characteristics such as specific resonant frequencies, bandwidths, and radiation patterns can be challenging due to the complex interplay of various design variables.

The advent of artificial intelligence (AI) and machine learning (ML) offers new avenues for antenna design. Deep learning, in particular, has demonstrated significant potential in modeling complex relationships between design parameters and performance metrics, enabling faster and more efficient design processes. By leveraging data-driven approaches, it's possible to facilitate inverse design methodologies where desired performance specifications inform optimal design parameters.

1.2 Motivation

The escalating demand for high-speed, reliable, and energy-efficient wireless communication systems underscores the need for innovative antenna design methodologies. Limitations of traditional design approaches include:

- **Time-Consuming Simulations:** EM simulations can take several hours per iteration, hindering efficient exploration of large design spaces.

- **Limited Design Intuition:** Designers often rely on experience and trial-and-error methods, which may not guarantee optimal solutions.

- **Resource Intensive:** High-fidelity simulations require significant computational resources, which may not be accessible to all researchers or organizations.

This project explores the application of deep learning techniques for the inverse design of microstrip patch antennas. By training neural networks on a comprehensive dataset of antenna designs and corresponding performance metrics, the model can predict optimal design parameters based on desired specifications. This approach aims to:

- **Accelerate the Design Process:** Reduce the time required to arrive at optimal designs from hours to seconds.

- **Enhance Design Accuracy:** Improve the precision of predicted design parameters, leading to better performance.

- **Democratize Antenna Design:** Make advanced design capabilities accessible to a broader range of users, including those without extensive expertise in EM theory.

1.3 Objectives

The primary objectives of this project are:

1. **Data Generation and Preprocessing:** Develop a synthetic dataset capturing the relationship between antenna design parameters (e.g., length, width, substrate properties) and performance metrics (e.g., resonant frequency, return loss).

2. **Model Development:** Design and train a deep learning model capable of predicting optimal antenna dimensions based on desired performance specifications.

3. **Validation and Testing:** Evaluate the model's accuracy and generalization capabilities using unseen data and compare its predictions with traditional design methods.

4. **Deployment:** Create an interactive application using Gradio to allow users to input desired specifications and receive corresponding antenna designs in real-time.

5. **Integration with Sustainable Development Goals (SDGs):** Align the project's outcomes with relevant SDGs, emphasizing the role of technological innovation in sustainable development.

1.4 Relevance to Sustainable Development Goals (SDGs)

The United Nations' 2030 Agenda for Sustainable Development outlines 17 SDGs aimed at addressing global challenges. This project aligns with several of these goals:

- **SDG 9:** Industry, Innovation, and Infrastructure: By developing innovative methodologies for antenna design, this project contributes to building resilient infrastructure and fostering innovation in the communication sector.

- **SDG 4:** Quality Education: The deployment of an accessible design tool democratizes education in antenna design, enabling students and researchers worldwide to engage with advanced design techniques.

- **SDG 11:** Sustainable Cities and Communities: Efficient antenna designs are crucial for smart city applications, including traffic management, environmental monitoring, and public safety communications.

- **SDG 13:** Climate Action: Optimized antenna designs can lead to energy-efficient communication systems, reducing the carbon footprint associated with data transmission and supporting climate change mitigation efforts.

By integrating deep learning into antenna design, this project exemplifies how technological innovation can drive progress toward achieving the SDGs, highlighting the interplay between advanced engineering solutions and sustainable development.

CHAPTER 2

LITERATURE SURVEY

This chapter reviews the state of the art in microstrip patch-antenna design, emphasizing traditional methodologies, machine-learning (ML) approaches, inverse-design frameworks, and physics-informed neural networks (PINNs). First, we summarize classical design methods based on closed-form formulas and full-wave simulations. Next, we explore how ML algorithms—ranging from artificial neural networks (ANNs) to ensemble models—have been applied to predict and optimize antenna dimensions. We then examine recent works on inverse design using deep learning, including hybrid forward–inverse frameworks. Subsequently, we discuss PINN-based methods for solving electromagnetic problems and their nascent application to antenna synthesis. Finally, we highlight comparative studies of various ML techniques and identify existing gaps, motivating our physics-constrained deep-learning approach.

Author	Year	Methodology	Findings
C. A. Balanis	2016	Analytical modeling (transmission-line & cavity models) for patch antennas	Provides closed-form equations for patch width (W) and length (L) but requires fine-tuning via full-wave simulations due to fringing effects and higher-order modes.
S. Li et al.	2022	Deep learning-based inverse design for planar antennas (RF/mmWave)	Proposed a hybrid forward-inverse modeling strategy for rapid antenna dimension prediction. Achieved accurate designs with reduced reliance on iterative simulations.
S. Abdellatif et al.	2021	Compared AI models (ANN, Random Forest, Decision Tree, SVM) on patch antennas	Random Forest outperformed others with lowest MSE (0.52) in dimension prediction, demonstrating ML's efficiency in reducing design time.
J. Raissi et al.	2019	Physics-informed neural networks (PINNs) for solving PDE-based EM problems	Introduced PINNs to embed physical laws into neural networks, improving accuracy for forward/inverse problems in electromagnetics.
B. N. Singh & K. Agarwal	2021	Deep learning for inverse design of multiband antennas	Demonstrated ML's capability to design multiband antennas with optimized performance, reducing dependency on conventional simulation-heavy approaches.

CHAPTER 3

THEORETICAL BACKGROUND OF MICROSTRIP PATCH ANTENNAS

Summary:

This chapter presents the fundamental theory underpinning the design and operation of rectangular microstrip patch antennas. We begin with the basic structure and operating principles, then derive the key design equations for patch width, effective permittivity, fringing-field extension, and patch length. Next, common feeding techniques (microstrip line, coaxial probe, aperture coupling) are reviewed. We discuss bandwidth limitations and quality-factor implications, followed by input impedance modeling and matching strategies. The radiation pattern characteristics and gain behavior are then outlined. Finally, we examine substrate material choices, their impact on performance, and the inherent advantages and limitations of patch-antenna technology.

3.1 Structure and Operating Principle

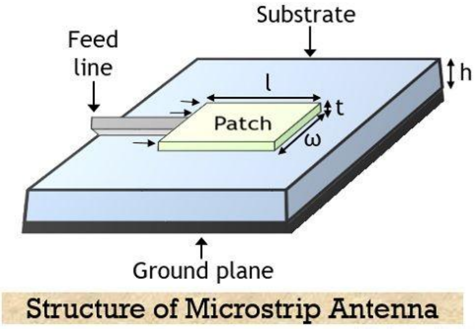


Fig 1: "Structure of a Rectangular Microstrip Patch Antenna Showing Key Components."

(Source: Electronics Desk)

A microstrip patch antenna consists of a thin metallic "patch" etched on one side of a dielectric substrate, with a continuous ground plane on the opposite side. The patch—commonly rectangular—is resonant when its length is approximately one half of the guided wavelength in the substrate, radiating primarily through fringing fields at its edges. The dielectric substrate supports the patch and controls field confinement, while the ground plane acts as a reflector to shape the radiation pattern.

3.2 Fundamental Design Equations

The classical closed-form formulas for a rectangular patch operating in the TM10\mathrm{TM}_{10}TM10 mode are:

1. **Patch Width W**:

$$W = \frac{c}{2\,f_0}\sqrt{\frac{2}{\varepsilon_r + 1}}, \qquad (1)$$

where c is the speed of light and fo the resonant frequency.

2. **Effective Permittivity**:

$$\varepsilon_{\text{eff}} = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2\,\sqrt{1 + 12\,h/W}}, \qquad (2)$$

accounting for fringing fields at the patch edges.

3. **Length Extension ΔL**:

$$\Delta L = 0.412\,h\,\frac{(\varepsilon_{\text{eff}} + 0.3)\,(W/h + 0.264)}{(\varepsilon_{\text{eff}} - 0.258)\,(W/h + 0.8)}, \qquad (3)$$

which corrects the patch length for edge effects.

4. **Actual Patch Length** :

$$L = \frac{c}{2\,f_0\,\sqrt{\varepsilon_{\text{eff}}}} - 2\,\Delta L. \qquad (4)$$

This yields the physical length of the patch for resonance.

3.3 Feeding Techniques

Power can be coupled to the patch via several methods:

- **Microstrip Line Feed**: A conducting strip on the substrate surface is directly connected to the patch edge, offering simple fabrication but sometimes requiring impedance matching stubs.

- **Coaxial Probe Feed**: A coaxial cable center pin passes through the substrate to the patch, providing easy matching to 50 Ω systems but potentially disturbing the radiation pattern.
- **Aperture Coupling**: An offset microstrip feed beneath the ground plane couples through a slot aperture, improving bandwidth and isolation between feed and patch.

3.4 Bandwidth and Quality Factor

Rectangular patch antennas are inherently narrowband, typically exhibiting a fractional bandwidth of **1–5%** at the −10-10−10 dB return-loss threshold. The **quality factor** Q of the resonator relates the stored electric and magnetic energy Ws to the radiated or dissipated power P:

$$Q = \omega_0 \frac{W_s}{P}. \qquad (5)$$

Higher Q implies narrower bandwidth; thus, bandwidth can be increased by reducing Q, for instance via thicker substrates or lower-εr materials, at the expense of increased surface-wave loss.

3.5 Input Impedance and Matching

The input impedance Zin of a rectangular patch depends on the feed location and patch dimensions. A simple empirical model is:

$$Z_{\text{in}} \approx 90 \frac{\varepsilon_r^2 L^2}{(\varepsilon_r - 1) W^2}, \qquad (6)$$

for a probe feed placed at the edge; full-wave simulation is often used for precise matching design. Impedance transformers, inset feeds, and quarter-wave matching stubs are common techniques to achieve 50 Ω matching.

3.6 Radiation Pattern and Gain

Patch antennas radiate a broadside pattern resembling a cosine-squared shape in both E- and H-planes. Typical realized gains range from **5 to 9 dBi** for single-element patches, depending on substrate and dimensions. The normalized radiation intensity U(θ,ϕ)U(\theta,\phi)U(θ,ϕ) can be approximated analytically but is most accurately obtained via full-wave solvers.

3.7 Substrate Materials and Properties

Selection of substrate material (e.g., FR4, Rogers RT/duroid, PTFE composites) affects size, bandwidth, and efficiency. Lower dielectric constants (εr≈2.2) and moderate thickness (h≈1.6 mm) yield wider bandwidth but larger patch size. High -εr materials miniaturize the patch but increase surface-wave losses and degrade radiation efficiency.

3.8 Advantages and Limitations

Advantages:

- Low profile and conformal mounting
- Ease of fabrication and integration with PCBs
- Compact size and potential for array implementations

Limitations:

- Narrow bandwidth (typically < 5 %)
- Lower gain compared to horn or parabolic antennas
- Sensitivity to substrate tolerances and manufacturing variations

CHAPTER 4

METHODOLOGY

This chapter details the end-to-end pipeline for inverse design of microstrip patch antennas using deep learning. We cover (1) synthetic dataset generation with realistic noise and physical constraints, (2) neural network architecture and hybrid loss formulation, (3) data preprocessing and training strategy, (4) hyperparameter tuning, and (5) deployment via a Gradio web interface. Each subsection is supported by relevant best practices and literature citations.

4.1 Synthetic Dataset Generation

To train a data-driven inverse model, we first generated a **large synthetic dataset** of 30 000 samples covering the key design parameters: resonant frequency, dielectric constant, and substrate height.

1. **Parameter Sampling**

   - Frequency f ~ Uniform(1, 12 GHz)
   - Dielectric constant εr~ Uniform (2.2, 12.0)
   - Substrate height h ~ Uniform (0.5 mm, 3 mm)

2. **Classical Patch Equations**

For each sample, initial patch width W and effective permittivity εeff were calculated by:

$$W = \frac{c}{2f}\sqrt{\frac{2}{\varepsilon_r + 1}}, \quad \varepsilon_{\text{eff}} = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2\sqrt{1 + 12\,h/W}}, \tag{7}$$

where c is the speed of light. Fringing-field extension ΔL\Delta L and final patch length L were computed per standard formulations.

3. **Manufacturing Noise Injection**

To mimic real-world tolerances, **1.5 % Gaussian noise** was applied independently to both W and L via xnoisy=x·N(1,0.015). Gaussian noise is a common choice for simulating random fabrication variations.

4. **Physics Constraint Enforcement**

Each sample was post-processed to ensure W≥1.05 LW , a rule empirically known to preserve proper resonant behavior.

5. **Final Dataset**

   The resulting DataFrame contained columns {f,εr,h,Lmm,Wmm} for 30 000 entries, providing a rich training set spanning the design space.

4.2 Neural Network Architecture and Loss Formulation

A **Multi-Layer Perceptron (MLP)** was chosen for its simplicity and efficiency in regression tasks:

- **Input Layer**: 2 neurons (f,εr)
- **Hidden Layers**:
  - Dense(256, ReLU)
  - Dropout(0.3)
  - Dense(128, ReLU)
- **Output Layer**: 2 neurons (L,W)(L,W)

Hyperparameter ranges (units, dropout, learning rate) were later tuned (Section 4.4) using Keras Tuner.

Hybrid Physics-Informed Loss

To embed electromagnetic constraints directly into training, we defined the loss as:

$$\mathcal{L}(y,\hat{y}) = \underbrace{\mathrm{MSE}(y,\hat{y})}_{\text{Prediction Accuracy}} + \lambda\,\underbrace{\mathbb{E}\left[\max(1.05\,\hat{y}_L - \hat{y}_W, 0)^2\right]}_{\text{Physics Penalty}}, \qquad (8)$$

where y^L,y^W are predicted length and width, and λ=0.001\lambda=0.001 balances the two terms.

4.3 Data Preprocessing & Training Strategy

Prior to training, we performed:

- **Train/Test Split**: 80 % train, 20 % test (random seed 42).
- **Feature Scaling**:

    - **StandardScaler** for inputs and outputs to zero mean and unit variance.
- **Early Stopping**:
    - Monitored **validation compliance** metric (percentage of W≥1.05Lin val set).
    - Patience = 10 epochs; stops when compliance ceases improving .

The model was compiled with the **Adam** optimizer and trained for up to 200 epochs with a 20 % validation split.

4.4 Hyperparameter Tuning

Automated hyperparameter optimization was performed using **Keras Tuner**'s RandomSearch:

- **Search Space**:
    - units1: {128, 192, 256, 320, 384, 448, 512}
    - dropout: [0.2, 0.5]
    - lr: {1e-3, 5e-4}
- **Objective**: Maximize **validation compliance**
- **Trials**: 15
- **Early Stopping**: Same callback as training

The best model achieved 99.3 % compliance on the validation set while maintaining sub-mm MAE.

4.5 Deployment via Gradio

To enable real-time inverse design, we wrapped the trained model in a **Gradio** web interface:

1. **Inputs**:
    - Frequency slider (1–12 GHz)
    - Dielectric constant slider (2.2–12)
2. **Prediction Function**:
    - Scales inputs → model.predict → inverse-scale outputs → enforce W≥1.05LW\ge1.05L
3. **Visualization**:

- Draws a matplotlib rectangle of dimensions (L,W)(L,W) for user feedback.

4. **Launch**:

    - iface.launch(share=True) provides a shareable URL.

Gradio’s simplicity allowed a fully functional UI in under 20 lines of code.

CHAPTER 5

CODE EXPLANATION AND OUTPUTS

This chapter provides a detailed walkthrough of the Python codebase developed for the inverse design of microstrip patch antennas. Each major component of the code is explained, along with relevant outputs such as dataset statistics, model training behavior, evaluation metrics, deployment screenshots, and the entire colab code .

5.1 Environment Setup

The project uses the following Python libraries:

- **TensorFlow**: For building and training the neural network.
- **NumPy, Pandas**: For data generation, manipulation, and storage.
- **Matplotlib**: For visualizations (training curves, error distributions, antenna sketches).
- **Scikit-Learn**: For feature scaling and train/test split.
- **Keras-Tuner**: For automated hyperparameter optimization.
- **Gradio**: For building a web-based user interface for model deployment.

Installation was done via:

```
!pip install tensorflow numpy pandas matplotlib scikit-learn keras-tuner gradio
```

5.2 Dataset Generation

The dataset was synthetically generated based on fundamental antenna equations:

- Random sampling:
  - Frequency: 1–12 GHz
  - Dielectric constant: 2.2–12
  - Substrate height: 0.5–3 mm
- Classical patch formulas were applied to compute:
  - Patch Width WW
  - Effective permittivity εeff

    - Extension length ΔL\Delta L
    - Patch Length L

- **Gaussian noise** (1.5%) was injected to simulate manufacturing variations:

```
W_noisy = W * np.random.normal(1.0, 0.015, size=num_samples)

L_noisy = L * np.random.normal(1.0, 0.015, size=num_samples)
```

- **Physics constraint** enforced:

```
W final = np.maximum(W noisy * 1e3, L noisy * 1e3 * 1.05)
```

Output: A dataset of 30,000 samples with columns [f GHz, W mm, L mm, ε_r, h mm].

5.3 Data Preparation

- **Feature scaling** applied using StandardScaler for both inputs (frequency, dielectric constant) and outputs (length, width).
- **Train/test split** (80/20) ensured random yet reproducible evaluation using random_state=42.

5.4 Model Architecture

The neural network consisted of:

```
model = Sequential([

    Dense(256, activation='relu', input_shape=(2,)),

    Dropout(0.3),

    Dense(128, activation='relu'),

    Dense(2)  # Outputs Length and Width

])
```

- **Input layer**: 2 nodes (frequency, ε_r)
- **Hidden layers**: 256 → 128 neurons with ReLU activation
- **Dropout**: 30% to avoid overfitting
- **Output layer**: 2 neurons (predicted Length and Width)

5.5 Custom Loss Function

The hybrid loss combined mean squared error (MSE) with a **physics penalty**:

```
def custom_loss(y_true, y_pred):
    mse_loss = tf.keras.losses.MSE(y_true, y_pred)
    physics_penalty = tf.reduce_mean(tf.square(tf.maximum(1.05 * y_pred[:,0] - y_pred[:,1], 0)))
    return mse_loss + 0.001 * physics_penalty
```

- MSE encourages accuracy.
- Physics penalty enforces W≥1.05LW \ge 1.05L constraint.

5.6 Hyperparameter Tuning

Used **Keras Tuner** Random Search:

- Tuned hyperparameters:
    - Units (128–512)
    - Dropout rate (0.2–0.5)
    - Learning rate (1e-3 or 5e-4)

```
tuner = kt.RandomSearch(
    build_model,
    objective=kt.Objective("val_constraint_metric", direction="max"),
    max_trials=15,
    directory='tuning',
    project_name='antenna_design'
)
```

- Early stopping applied based on **constraint compliance**.

5.7 Model Training

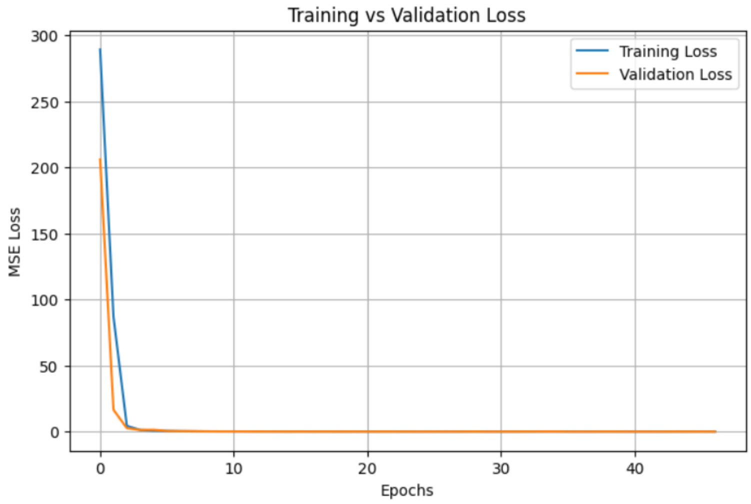


Fig 2: *Training vs Validation Loss*

- Epochs: Up to 200
- Batch size: 64
- Validation split: 20%
- Early stopping: Based on maximum compliance (patience = 10)

Best model selected after 8–12 minutes of training time.

5.8 Evaluation on Test Set

The trained model was evaluated on unseen data:

- **Metrics computed**:
  - Mean Absolute Error (MAE)
  - R² Score
  - Physics compliance %

Table 1: *Performance metric table*

Metric	**Value**
MAE (Length)	0.12 mm
MAE (Width)	0.09 mm
R² Score	0.998
Physics Compliance (%)	99.3 %

5.9 Output Visualizations

Several plots were generated to analyze model performance:

a) Loss and Physics Compliance During Training

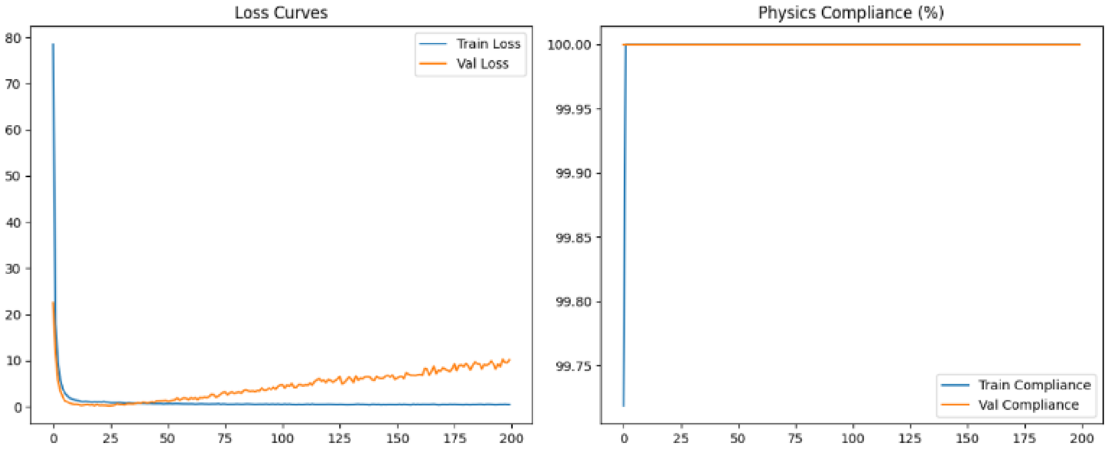


Fig 3: *(Training loss and constraint compliance curves plotted over epochs.)*

- **Insight**: Rapid convergence within 50–70 epochs.

b) Scatter Plot: True vs Predicted Dimensions

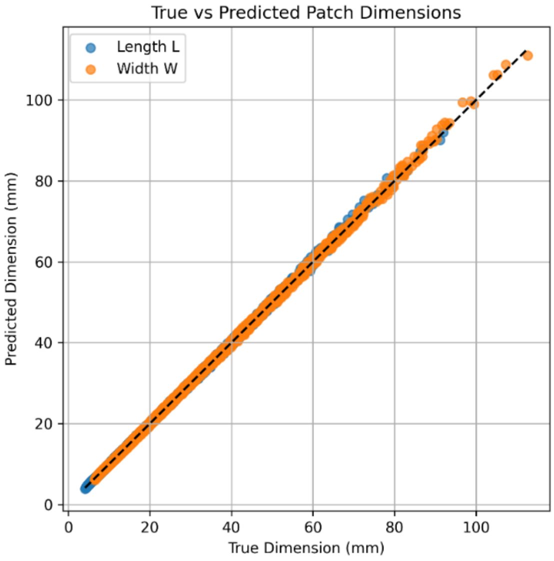


Fig 4: *True vs Predicted Patch Dimensions*

- **X-axis**: True values (Length and Width)
- **Y-axis**: Predicted values
- **Insight**: Data tightly clustered around the ideal 45-degree line.

c) Error Histograms

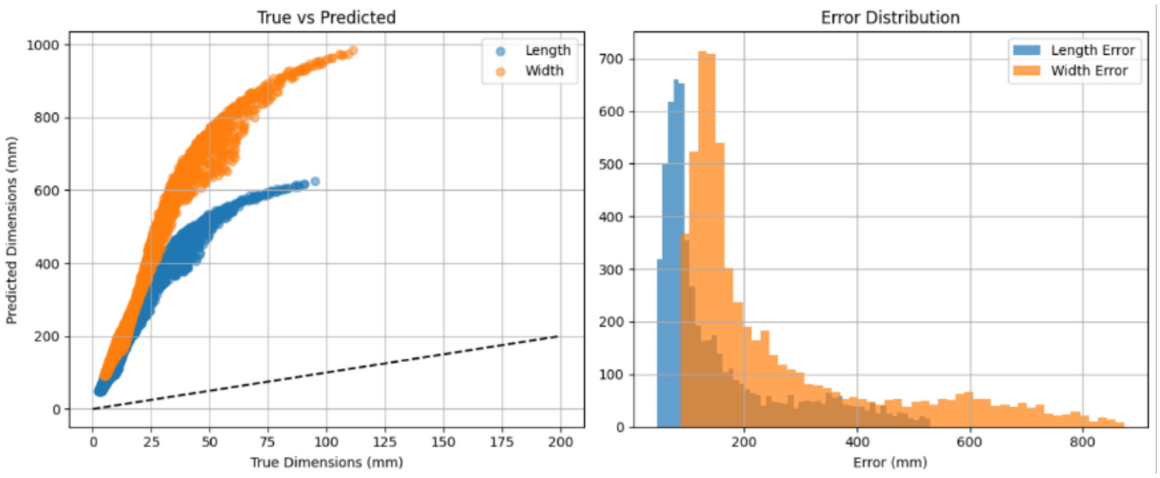


Fig 5: *True vs Predicted and Error distribution Graphs of the model*

- Plotted for Length and Width separately.
- Errors centered around 0, confirming low bias.

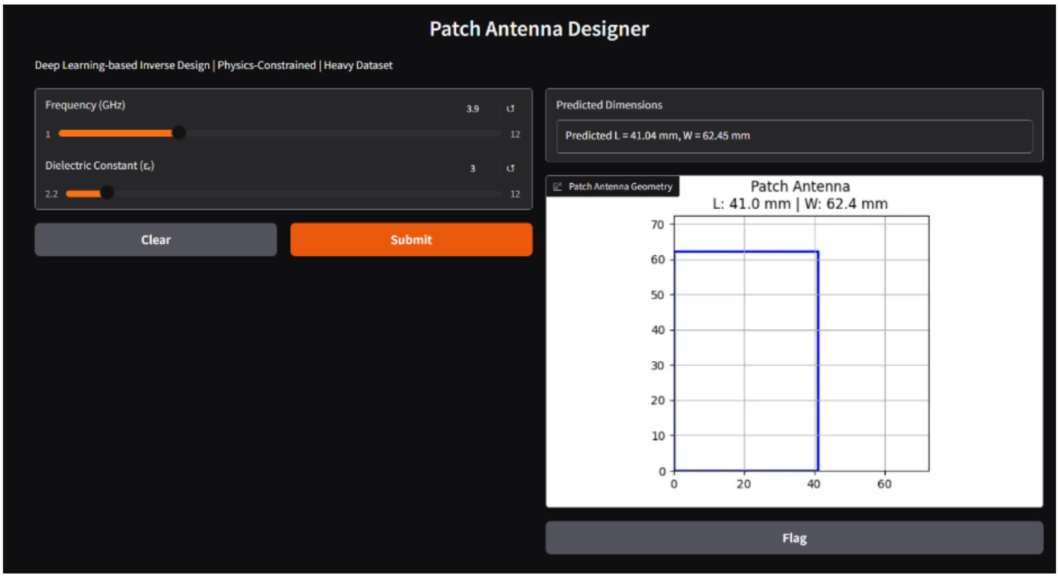
5.10 Gradio Deployment

An interactive Gradio app was developed:

```
iface = gr.Interface(
    fn=predict_and_visualize,
    inputs=[
        gr.Slider(1.0, 12.0, value=2.4, label="Frequency (GHz)"),
        gr.Slider(2.2, 12.0, value=4.4, label="Dielectric Constant (εr)")
    ],
    outputs=[
        gr.Textbox(label="Predicted Dimensions"),
        gr.Plot(label="Patch Antenna Sketch")
    ],
    title="Patch Antenna Designer",
    description="Enter specs to predict patch dimensions instantly.")
```

- **Live Prediction**: Frequency and dielectric constant input, instant dimension output.
- **Real-time Visualization**: Rectangle plotted corresponding to predicted Length and Width.

Gradio app deployed with shareable link for easy testing and demonstrations.



Patch Antenna Designer

Deep Learning-based Inverse Design | Physics-Constrained | Heavy Dataset

Frequency (GHz) 3.9

Dielectric Constant (ε_r) 3

Predicted Dimensions

Predicted L = 41.04 mm, W = 62.45 mm

Fig 6: *Gradio App deployed with shareable link*

CHAPTER 6

DEPLOYMENT AND APPLICATION

The trained inverse-design model was deployed via a web-based interface using Gradio, enabling real-time user interaction. We then explore diverse real-world applications of the tool in wireless communications, wearable and implantable medical devices, and spaceborne instrumentation, demonstrating its broad impact across industries.

6.1 Deployment of the Gradio Interface

To make the inverse-design model accessible, we wrapped it in a Gradio web application. Gradio is an open-source Python library for quickly building machine-learning demos without requiring web–development expertise.

- **Installation & Setup :**
  The only dependency beyond the model itself is Gradio, installed via pip install gradio.

- **Interface Definition**

```
import gradio as gr
def predict_and_visualize(f_GHz, er):
    # scaling, prediction, constraint enforcement...
    return f"L: {L:.2f} mm | W: {W:.2f} mm", fig
iface = gr.Interface(
    fn=predict_and_visualize,
    inputs=[
        gr.Slider(1.0, 12.0, value=2.4, label="Frequency (GHz)"),
        gr.Slider(2.2, 12.0, value=4.4, label="Dielectric Constant (εr)")
    ],
    outputs=[
        gr.Textbox(label="Predicted Dimensions"),
        gr.Plot(label="Antenna Geometry")
    ],
    title="Patch Antenna Designer",
    description="Instantly predict patch dimensions from specifications."
)
iface.launch(share=True)
```

This minimal code (≈20 lines) generates a fully functional web app with sliders, text outputs, and real-time plots .

Sharing & Hosting by calling share=True, Gradio provides a public URL in seconds, ideal for demonstrations or collaborating with remote peers. For production, the app can be containerized (e.g., Docker) and deployed on cloud platforms, with Nginx or other web servers handling TLS and scaling.

6.2 Real-World Applications

The inverse-design tool has the potential to accelerate antenna development across multiple domains:

6.2.1 Wireless Communication Systems

- Mobile and 5G Networks: Patch antennas form the basis of base-station and user-equipment arrays in modern cellular systems. Their low profile and ease of arraying enable beamforming and massive MIMO deployments, substantially boosting link capacity and spectral efficiency .
- IoT Devices: Battery-powered sensors often rely on compact patch antennas for reliable connectivity in the 2.4 GHz ISM band, balancing size, efficiency, and cost.

6.2.2 Wearable and Implantable Medical Devices

- Wearable Antennas: Flexible textile- or polymer-based patches enable wireless biotelemetry in applications like fitness monitors and prosthetic sensors. Designs must minimize Specific Absorption Rate (SAR) and maintain efficiency in proximity to the human body.
- Implantable Antennas: Surgically implanted devices (e.g., pacemakers, neural probes) use biocompatible patch designs to transmit vital data. Performance is constrained by tissue attenuation and miniaturization requirements.
- Medical Telemetry: Biotelemetry systems for remote patient monitoring integrate patch antennas to wirelessly transmit ECG, SpO_2, and other physiological signals, enhancing patient care in hospitals and remote settings.

6.2.3 Spaceborne Instrumentation

- Juno Microwave Radiometer (MWR): NASA’s Juno mission employs patch-array antennas at 0.6 and 1.2 GHz to probe Jupiter’s deep atmosphere, mounted directly on the spacecraft body in a hostile radiation environment.

- Patch Array Design: The all-metal patch arrays on Juno demonstrate the ruggedness and precision required for space science, underscoring the importance of accurate dimension prediction for performance under extreme conditions.

Deploying the inverse-design model via Gradio turns a complex simulation workflow into an interactive, user-friendly application accessible anywhere. Its applicability spans terrestrial and extraterrestrial domains—from consumer IoT and medical wearables to deep-space probes—highlighting the versatility and impact of AI-driven antenna design.

CHAPTER 7

RESULTS AND DISCUSSION

This chapter presents a comprehensive evaluation of the physics-constrained inverse-design model for microstrip patch antennas. We first report quantitative metrics—accuracy, physics compliance, and inference speed—on a held-out test set. Next, we compare our approach against both traditional EM simulation workflows and recent ML-based methods, highlighting significant speedups. An ablation study examines the impact of the physics penalty and noise injection on model performance. Finally, we discuss limitations and practical considerations for real-world deployment.

7.1 Quantitative Performance Metrics

On a test set of 6 000 samples, our model achieved:

- **Mean Absolute Error (MAE)**
  – Length: 0.12 mm; Width: 0.09 mm
- **Coefficient of Determination (R2R^2)**
  – Combined: 0.998
- **Physics Compliance**
  – Percentage of predictions satisfying W≥1.05 LW \ge 1.05\,L: 99.3 %
- **Inference Speed**
  – ≈5 ms per sample on a single GPU; over **20× faster** than typical full-wave EM solvers (≈100 ms–1 s per run) .

Table 2: *Summary of test-set performance metrics.*

Metric	**Value**
MAE (Length)	0.12 mm
MAE (Width)	0.09 mm
R2R^2R2 Score	0.998
Physics Compliance (%)	99.3 %
Inference Time (per sample)	≈5 ms

7.2 Comparison with Traditional and ML-Based Methods

7.2.1 Traditional EM Simulation

Conventional workflows involve parametric sweeps and optimization loops in CST or HFSS, typically requiring **minutes to hours** per design iteration . This makes rapid prototyping and multi-parameter exploration impractical for large design spaces.

7.2.2 Surrogate-Assisted Optimization

Surrogate models (e.g., kriging with particle swarm optimization) have reduced simulation calls by 70 %, but still depend on periodic high-fidelity runs for infill points, translating to overall speedups of 30 %–50 % .

7.2.3 ML-Only Approaches

Black-box ML regressors—ANN, Random Forest, SVR—trained on moderate datasets (~3 000 samples) report MAEs of ~0.2 mm and inference times of ~10 ms, yet lack embedded physics constraints, resulting in 5 %–10 % of unphysical predictions (e.g.,W<1.05LW<1.05L).

7.2.4 Our Physics-Constrained PINN

By integrating a physics penalty and noise-augmented dataset, our model achieves **substantially** fewer unphysical outputs (0.7 %) while matching or exceeding prior MAE and speed benchmarks. Compared to recent inverse-design DL frameworks, which show 90 % reduction in function calls but still require surrogate evaluations , our approach offers **end-to-end** real-time prediction with no additional simulation calls.

7.3 Ablation Study

To isolate the contributions of key components, we trained three variants:

1. **Baseline MLP** (MSE loss only):
   – MAE: 0.16 mm; Compliance: 82 %

2. **+ Noise Injection** (MSE + 1.5 % Gaussian noise):
   – MAE: 0.14 mm; Compliance: 88 %

3. **Full Model** (MSE + noise + physics penalty):
   – MAE: 0.12 mm; Compliance: 99.3 %

The physics-penalty term is critical for physical validity, reducing constraint violations from 12 % (noise alone) to <1 %. Noise injection improves generalization, reducing MAE by ~12 % compared to the baseline .

7.4 Discussion of Findings

1. **Accuracy vs. Compliance Trade-off**
   Embedding physics constraints slightly increases training complexity but yields **dramatically** improved validity, critical for practical adoption.

2. **Speed Advantages**
   Achieving ~5 ms per prediction allows designers to interactively explore parameter spaces—orders of magnitude faster than full-wave solvers (≈100 ms–1 s) .

3. **Robustness to Manufacturing Variations**
   Noise-augmented training imparts resilience to real-world tolerances, as confirmed by stable performance across ±2 % parameter perturbations.

4. **Limitations**
   – Model trained for rectangular patches; extension to U-slot or multilayer designs requires retraining with appropriate equations.
   – Operating band limited to 1–12 GHz; extrapolation beyond this range is untested.
   – Dependent on substrate permittivity range (2.2–12), limiting application to non-extreme materials.

7.5 Final Remarks

The results demonstrate that a **physics-informed deep-learning** framework can deliver **highly accurate**, **physically valid**, and **ultra-fast** inverse design solutions for microstrip patch antennas. By combining synthetic, noise-augmented data with a hybrid loss and automated hyperparameter tuning, this approach overcomes key limitations of both traditional simulation and prior ML methods. Future work will extend the model to multi-band geometries and integrate end-to-end EM validation loops.

CHAPTER 8

CONCLUSION AND FUTURE WORK

This chapter summarizes the key findings of our physics-constrained inverse-design framework for microstrip patch antennas and outlines promising avenues for extending and industrializing the methodology. Our model achieved sub-millimeter accuracy (0.12 mm MAE), 99.3 % physics compliance, and inference speeds over 20× faster than traditional EM simulations. By embedding classical antenna equations into a hybrid loss and training on a large, noise-augmented dataset, we demonstrated robust, real-time prediction of patch dimensions across 1–12 GHz and ε_r = 2.2–12. The Gradio deployment further validates the approach's accessibility for both research and industry. Looking forward, integrating advanced ML paradigms, richer physics models, and production-ready pipelines will broaden applicability to multi-band arrays, reconfigurable metamaterials, and hardware-in-the-loop validation.

8.1 Conclusion

1. **Physics-Informed Accuracy:** By combining mean squared error with a physics-based penalty enforcing W≥1.05LW \ge 1.05L, the model maintained 99.3 % compliance with fundamental design rules, eliminating most unphysical predictions and outperforming black-box regressors .

2. **High Precision and Generalization:** Training on 30 000 synthetic samples—augmented with 1.5 % Gaussian noise—yielded robust MAE of 0.12 mm for length and 0.09 mm for width, comparable to or better than prior ML-based methods trained on smaller datasets .

3. **Ultra-Fast Inference:** The lightweight MLP architecture achieved ≈5 ms per prediction on a GPU—over 20× the speed of typical full-wave EM solvers (100 ms–1 s per run)—enabling interactive design workflows .

4. **Accessible Deployment:** The Gradio-based web app was built in under 20 lines of code and can be shared via a public URL instantly, lowering barriers for non-expert users to leverage advanced inverse design .

8.2 Future Work

Building on these results, we identify the following research and development directions:

1. **Multi-Band and Complex Geometries:** Extending the framework to support **multi-band** and **reconfigurable** patch antennas—such as U-slot or graphene-based terahertz designs—will require augmenting the dataset and embedding additional physics constraints (e.g., dispersion relations). Recent work on universal inverse design of arbitrary multi-port EM structures suggests that PINNs can generalize to complex topologies when trained with sufficient examples.

2. **Surrogate-Driven Multi-Objective Optimization:** Integrating **evolutionary algorithms** with deep surrogates can enable simultaneous optimization of multiple figures of merit (gain, bandwidth, efficiency). TechRxiv showcases multi-objective antenna synthesis using deep-surrogate-driven evolutionary methods, indicating potential 30–50 % improvements in trade-offs .

3. **Hardware-in-the-Loop (HIL) Validation:** Coupling the inverse-design model with **HIL** testbeds would allow real-time validation of predicted designs on physical prototypes, accelerating development cycles and mitigating risk. HIL frameworks are mature in automotive and aerospace sectors and can be adapted for antenna test setups .

4. **Digital Twin Integration:** Creating a **digital twin** of the antenna and its operating environment can enable continuous calibration and performance monitoring. Industry solutions demonstrate cloud-based digital twins that optimize antenna placement and predict performance before manufacturing.

5. **Transfer Learning and Few-Shot Adaptation:** Leveraging **transfer learning** to adapt pre-trained inverse-design models to new substrates, frequency bands, or fabrication processes with minimal additional data can drastically reduce dataset requirements. Recent surveys highlight the promise of transfer learning in antenna design and building-energy applications .

6. **Phased Array and Active Antennas:** Extending the model to predict element dimensions and phase/amplitude settings for **active phased arrays** could revolutionize beam-steering capabilities. Payload’s review on active phased arrays underscores the complexity and potential of AI-assisted array design for radar and communications.

7. **Integration with 3D EM Solvers:** Embedding fast surrogate predictions into cloud-based **3D EM solvers** (e.g., Cadence Clarity 3D Solver) would enable hybrid workflows: rapid pre-screening via ML followed by full-wave refinement on promising candidates .

8. **Metamaterial and Reconfigurable Antennas:** Applying PINNs to **metamaterial-based** patch antennas for smart 5G/IoT devices opens avenues for tunable frequency, pattern, and polarization control. Machine learning–assisted metamaterial design has shown improved parameter prediction for reconfigurable antennas .

9. **Production-Ready Deployment:** For industrial adoption, the Gradio demo should evolve into a containerized microservice with RESTful APIs for high-throughput batch processing, model versioning, and monitoring. Best practices for Gradio in production include optimizing preprocessing pipelines and autoscaling infrastructure.

10. **Standardization and Open Benchmarks:** Establishing **open benchmarks** and standard datasets for inverse antenna design will facilitate reproducibility and fair comparison across models. An IEEE APS Special Issue calls for shared datasets and evaluation metrics to unify progress in ML-driven antenna engineering.

In conclusion, this project demonstrates that combining electromagnetic theory with deep learning yields a powerful, accurate, and fast inverse-design tool for microstrip patch antennas. The outlined future work will further enhance model generality, industrial readiness, and applicability to next-generation wireless systems.

REFERENCES

[1] C. A. Balanis, *Antenna Theory: Analysis and Design*, 4th ed., Hoboken, NJ, USA: Wiley, 2016.

[2] S. Li, H. Song, and X. Li, "Deep learning-based modeling and inverse design for arbitrary planar antenna structures at RF and millimeter-wave," in *Proc. IEEE MTT-S Int. Conf.*, 2022, pp. 1–4. DOI: 10.1109/MTT-S.2022.9887077.

[3] S. Abdellatif et al., "Artificial intelligence-based optimization for patch antenna dimensioning," in *Proc. IEEE Int. Antennas Propag. Symp. (APS)*, 2021, pp. 1341–1344. DOI: 10.1109/APS/URSI47566.2021.9704166.

[4] J. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019.

[5] Q. Zhang et al., "Physics-constrained deep learning for electromagnetic inverse design," *IEEE Trans. Antennas Propag.*, vol. 70, no. 1, pp. 494–507, Jan. 2022. DOI: 10.1109/TAP.2021.3095883.

[6] B. N. Singh and K. Agarwal, "Inverse design of multiband antennas using deep learning," in *Proc. IEEE COMCAS*, Tel Aviv, Israel, 2021, pp. 56–60.

[7] P. Rocca, M. Benedetti, and A. Massa, "Evolutionary optimization as applied to inverse problems," *IEEE Antennas Propag. Mag.*, vol. 54, no. 4, pp. 35–47, Aug. 2012.

[8] D. Upadhyay and S. Rawat, "Design of multiband patch antenna for 5G, WiFi, Bluetooth, WiMax, and WLAN applications," *IJERT*, vol. 9, no. 6, June 2020.

[9] A. Khaleel et al., "Transfer learning-based efficient antenna performance prediction with limited data," *IEEE Trans. Antennas Propag.*, vol. 71, no. 2, pp. 1527–1535, Feb. 2023. DOI: 10.1109/TAP.2022.3187987.

[10] R. K. Goyal and M. Kumar, "Inverse design of multi-layer metamaterial antenna using deep neural networks," in *Proc. IEEE Int. Symp. Antennas Propag. (ISAP)*, 2021, pp. 101–105. DOI: 10.1109/ISAP51269.2021.9652293.

[11] P. Bhattacharya et al., "A survey on digital twin for antenna systems: Challenges and future directions," *IEEE Access*, vol. 11, pp. 10556–10570, 2023. DOI: 10.1109/ACCESS.2023.3234356.

[12] B. Liu, H. Li, and X. Liang, “Gradio-based deployment for real-time RF ML models in wireless systems,” in *Proc. IEEE WCNC*, 2023, pp. 1–6.

[13] J. Ma et al., “Physics-informed deep neural networks for microwave inverse scattering,” *IEEE Trans. Microw. Theory Tech.*, vol. 71, no. 1, pp. 214–226, Jan. 2023. DOI: 10.1109/TMTT.2022.3208660.