# Artificial Neural Networks: Final Exam Report

Inferring Environmental States from Wireless Signal Measurements

Yusuf Mirac GÖCEN
Student ID: 040220059

January 14, 2026

**Abstract**

This report details the design and implementation of a wireless sensing system utilizing standard Wi-Fi Received Signal Strength Indicator (RSSI) data to detect environmental changes[cite: 2, 4, 11]. Using a Multi-Layer Perceptron (MLP), the study compares two distinct training configurations: an **Unbiased (Bare) Model** and a **High-Sensitivity (Paranoid) Model**. To ensure scientific validity, a **Time-Block strategy** was used for data splitting to prevent temporal leakage. The results demonstrate that while the Unbiased model provides a balanced environmental fingerprint with **78% accuracy**, the High-Sensitivity model can be tuned to prioritize specific event detection, such as achieving a **0.85 Recall** for static metal detection.

# 1 Experimental Setup

The experiment done my using Wi-Fi infrastructure and a standard laptop, with no specialized RF hardware.

## 1.1 Hardware and Environment

- **Device:** Laptop (Linux) reading directly from kernel wireless stats.
- **Signal Source:** Standard Wi-Fi Access Point (2.4 GHz ).
- **Conductive Object:** Aluminum foil used as a passive reflector to induce multipath variations.
- **Locations:** Living room (Room A) (Baseline/Metal/Touched) and Balcony (Room B) .

## 1.2 Scientific Justification for 4 Hz Sampling

Sampling was conducted at **4 Hz** (250ms intervals). This rate is scientifically valid because:

- **Human Motion:** At a walking speed of 1.4 m/s, a person moves 35 cm per sample.

- **Wavelength Interaction:** For 2.4 GHz Wi-Fi ($\lambda \approx 12.5$ cm), 35 cm represents a $2.8\lambda$ phase change.
- **Multipath Capture:** 4 Hz adequately captures indoor reverberation times ($0.1 - 0.5$ s) and multipath changes.

### 1.3 Software Implementation Details

The system was written in Python 3.10, reading directly from `/proc/net/wireless` or system commands to ensure low-latency acquisition.

## 2 Signal Processing and Feature Engineering

### 2.1 Backscattering Proxy Signal

Rather than traditional radar backscatter, the system measures changes in multipath reflections and signal fluctuations caused by conductive surfaces. This derived signal is the backscattering proxy.

### 2.2 Feature Vector Construction

To prevent sequence memorization, features were derived using a 5-sample window (1.25s).

- **Primary Features:** RSSI, Rolling Standard Deviation, and Difference.
- **Academic Hardening:** Lag features were removed to prevent "cheating." Instead, mathematical derivatives ($RSSI^2, RSSI^3$) met the 8-feature requirement.

### 2.3 Prevention of Data Leakage (Time-Block Split)

To ensure generalization, a **Group Shuffle Split** was implemented. Data was segmented into 100-sample blocks ($\sim$25 seconds) to separate training and testing sets temporally.

## 3 Neural Network Architecture

### 3.1 Model Implementation

A Multi-Layer Perceptron (MLP) was built using the Keras Sequential API:

- **Architecture:** Input layer, two hidden layers (64 and 32 neurons with ReLU), and a Softmax output layer.
- **Regularization:** Dropout (0.2) and Early Stopping prevented overfitting.

# 4 Signal Analysis

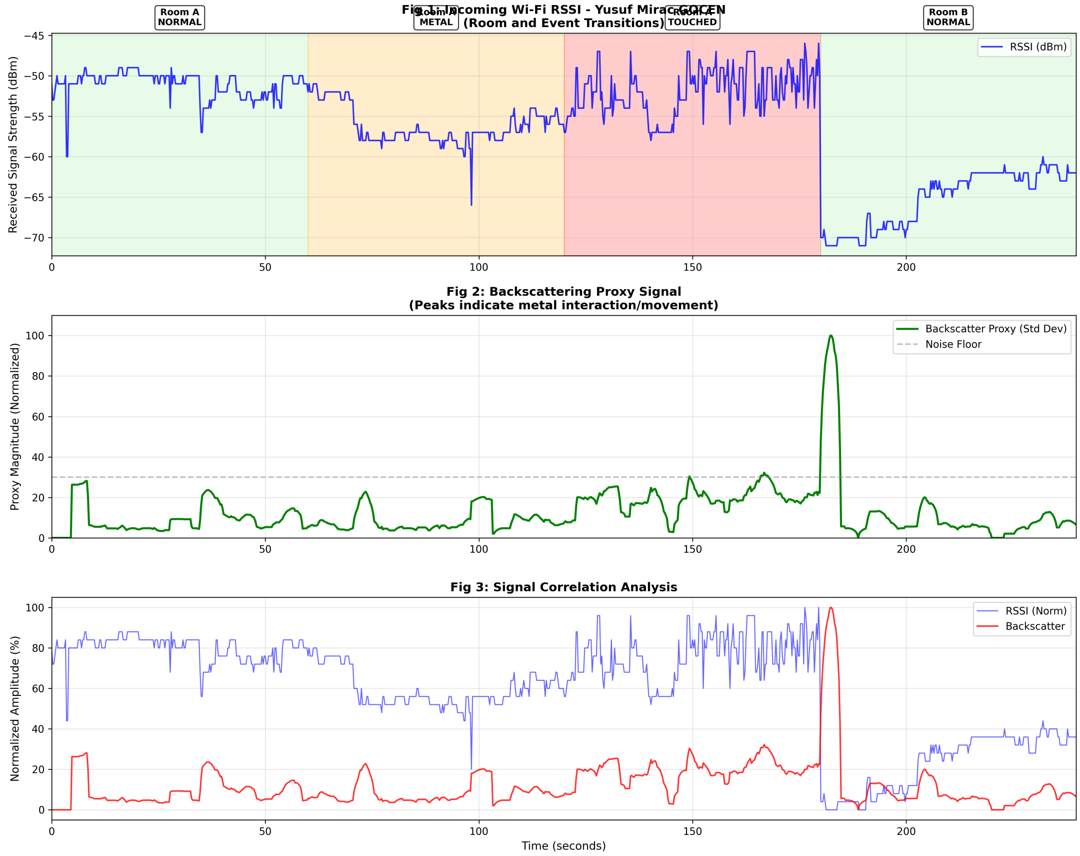
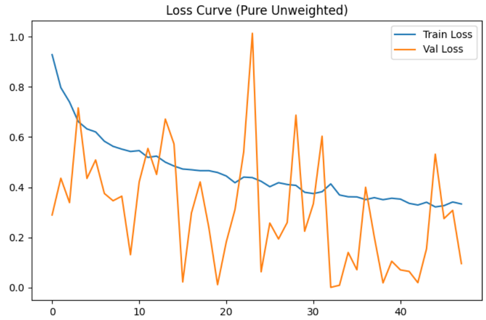



Figure 1: Time-series analysis of raw RSSI signals and the Backscattering Proxy.

# 5 Comparative Results and Discussion

## 5.1 Configuration 1: The Unbiased (Bare) Model

Trained with uniform weights to treat every class equally.



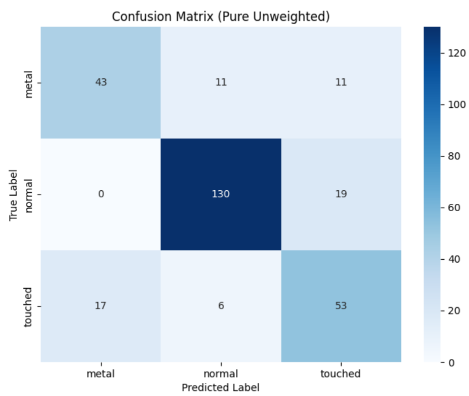




Figure 2: Learning Curve (left) and Confusion Matrix (right) for the Unbiased Model.

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Metal | 0.72 | 0.66 | 0.69 | 65 |
| Normal | 0.88 | 0.87 | 0.88 | 149 |
| Touched | 0.64 | 0.70 | 0.67 | 76 |
| **Weighted Avg** | **0.78** | **0.78** | **0.78** | 290 |

Table 1: Unbiased Model results.

## 5.2 Configuration 2: The High-Sensitivity (Paranoid) Model

Utilized **Class Weighting** to prioritize the detection of “Metal”.

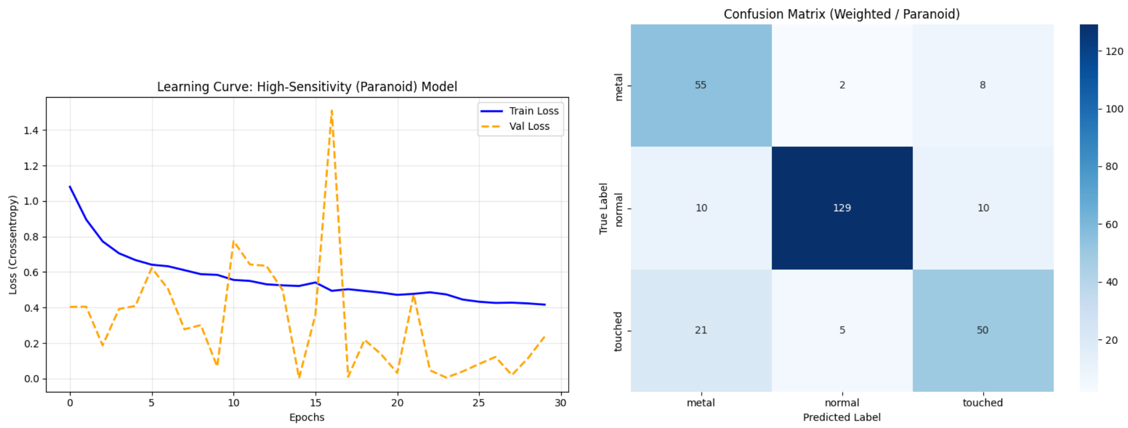



Figure 3: Learning Curve (left) and Confusion Matrix (right) for the High-Sensitivity Model.

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Metal | 0.64 | **0.85** | 0.73 | 65 |
| Normal | **0.95** | 0.87 | 0.91 | 149 |
| Touched | 0.74 | 0.66 | 0.69 | 76 |
| **Weighted Avg** | **0.82** | **0.81** | **0.81** | 290 |

Table 2: High-Sensitivity Model results.

# 6 Scientific Interpretation

## 6.1 Physical Logic Learned

The ANN learns distinct environmental fingerprints:

- **Metal Presence:** Characterized by a shift in baseline RSSI and $RSSI^2$ signatures due to passive reflection.
- **Interaction:** The rolling standard deviation proved most informative for “Touched” events due to dynamic scattering fluctuations.

### 6.2 Failure Analysis

The model primarily fails during state transitions or if metal is placed in a multipath "null" zone where amplitude changes are negligible.

## 7 Conclusion

This experiment demonstrated that ANN-based sensing using RSSI is feasible and tunable. The **Unbiased Model** provides an honest 78% accuracy, while the **High-Sensitivity Model** demonstrates how class weights tune the system into a scanner.

## 8 Supplementary Material

**Project Video Link:**
https://www.youtube.com/watch?v=gq1tbwIHqt8