Design and Development of FPGA DSP Core Architecture for Integrated RF Signal Detection and Classification

M.Tech Thesis

By

Anismita Biswas



DEPARTMENT OF ASTRONOMY, ASTROPHYSICS AND SPACE ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

May, 2026

Design and Development of FPGA DSP Core Architecture for Integrated RF Signal Detection and Classification

A THESIS

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Technology

By

Anismita Biswas



Supervisor: Prof. Abhirup Datta

DEPARTMENT OF ASTRONOMY, ASTROPHYSICS AND SPACE ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

May, 2026

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **"Design and Development of FPGA DSP Core Architecture for Integrated RF Signal Detection and Classification"** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF TECHNOLOGY** and submitted in the **Department of Astronomy, Astrophysics and Space Engineering, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from July, 2024 to May, 2026 under the supervision of Prof. Abhirup Dutta (Dean of Research and Development)

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Anismita Biswas
(07/05/2026)

(Anismita Biswas)

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

07/05/2026

Signature of the Supervisor of
M.Tech. thesis (with date)
(Prof. Abhirup Datta)

Anismita Biswas has successfully given his/her M.Tech. Oral Examination held on **4th May, 2026**.

Signature(s) of Supervisor(s) of MTech thesis
Date: 07/05/2026

Programme Coordinator, M.Tech.
Date:

Suman Majumder
Convenor, DPGC
Date:

HoD, DAASE
Date:

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, **Prof. Abhirup Datta**, for his continuous guidance, support, and encouragement throughout the course of this thesis. His insights in FPGA-based signal processing and satellite communication systems have been fundamental to the successful completion of this work.

I am also deeply thankful to **Dr. Harsha Avinash Tanti** for his valuable guidance and support, which significantly contributed to the progress of this project. I would like to extend my appreciation to **Abhijeet Dutta** for his continuous support and assistance in the laboratory.

I also thank the faculty and staff of the Department of Astronomy, Astrophysics and Space Engineering, IIT Indore, for providing the necessary academic and laboratory infrastructure.

Finally, I express my deepest gratitude to my family and friends for their unwavering support and encouragement.

ABSTRACT

This thesis presents the design and implementation of an FPGA-based software-defined radio (SDR) system for real-time wideband RF signal detection, analysis, and classification. The architecture integrates signal acquisition, processing, dataset generation, and machine learning-based classification into a unified pipeline, using a dual-stage approach that combines wideband spectrum sensing with narrowband signal analysis.

A Red Pitaya SDR platform performs wideband acquisition (15.36 MHz) using FFT-based scanning with local oscillator sweeping to reconstruct a 0–61.44 MHz spectrum. Detected signals are tracked using a narrowband receiver (192 kHz) to extract high-resolution I/Q samples, followed by preprocessing steps such as windowing, FFT, and spectral feature extraction.

A key contribution is the development of a large-scale custom RF dataset generation framework that simulates realistic UHF-band (390–480 MHz) communication environments, including public safety radio, walkie-talkie systems, radiosondes, SCADA telemetry, RFID, paging systems, and remote controls. By modeling protocol-specific characteristics, the dataset captures realistic spectrum usage patterns, producing approximately 4.4×10^5 labeled samples using time- and frequency-domain features.

Machine learning models (ANN, CNN, CNN–LSTM) are used to validate the dataset and evaluate its ability to capture discriminative signal features. Model performance provides insight into dataset quality, robustness, and generalization across varying signal conditions.

The system is designed for FPGA-based deployment, with DSP operations implemented in hardware and classification supported through a streaming framework. Results demonstrate reliable detection, accurate classification, and real-time performance, establishing a scalable framework for RF signal intelligence applications.

The final system enables RF signal identification through protocol classification, modulation recognition, carrier frequency estimation, and bandwidth analysis. Experimental results demonstrate accurate detection, reliable classification, and stable real-time performance, establishing a scalable framework for FPGA-based RF intelligence applications.

Contents

1 Introduction **1**

1.1 RF Signal Detection and Classification 2

1.1.1 Applications Across Communications and Intelligent RF Systems . . . 2

1.2 From Classical DSP to FPGA-Based Acceleration 3

1.2.1 Limitations of CPU/GPU Solutions 3

1.2.2 Advantages of FPGA Architectures 3

1.3 Research Gap and Motivation 4

1.3.1 Motivation 5

1.3.2 Research Objectives 6

2 Theoretical Background **7**

2.1 Direct Digital Synthesizer (DDS) 7

2.1.1 Role of DDS in an RF Processing Chain 7

2.1.2 Conceptual Basis of DDS 8

2.1.3 Frequency Control and Resolution 8

2.1.4 Sinusoidal Generation 8

2.1.5 Digital Mixing for Frequency Translation 9

2.1.6 Implementation Considerations 9

2.2 CIC Decimation Filter Architecture 9

2.2.1 Role of Decimation in an RF Processing Chain 9

2.2.2 Conceptual Basis of CIC Filtering 10

2.2.3 CIC Decimator Structure 10

2.2.4 Parameter Configuration in This Work 11

2.2.5 Integrator and Comb Operation 11

2.2.6 Frequency Response and Bit Growth 12

2.2.7 Implementation Considerations 12

2.3 FIR Compensation Filtering 12

2.3.1 Implemented FIR Architecture 13

2.4 Windowed FFT for Spectrum Stitching 13

2.4.1 FFT-Based Signal Representation 13

2.4.2 Windowing and Spectral Leakage Reduction . . . 14
2.4.3 Spectrum Stitching for Wideband Analysis . . . 14
2.5 Protocol-Centric Dataset Design . . . 14
2.5.1 Public Safety Systems . . . 15
2.5.2 Walkie-Talkie Systems . . . 15
2.5.3 Meteorological Systems . . . 15
2.5.4 Supervisory Control and Data Acquisition (SCADA Systems) . . . 16
2.5.5 RFID Systems . . . 16
2.5.6 Paging Systems: POCSAG . . . 16
2.5.7 Remote Control Systems . . . 16
2.5.8 Signal Signature Perspective . . . 17
2.5.9 Summary of Modulation Diversity . . . 18
2.5.10 Feature Structuring and Dimensional Consistency . . . 18
2.5.11 Design Rationale . . . 19
2.6 Machine Learning Modeling . . . 19
2.6.1 Problem Formulation . . . 20
2.6.2 Representation Learning . . . 22
2.6.3 Neural Network Models and Their Functioning . . . 22
2.6.4 Modeling Advantages for RF Signal Classification . . . 26
2.6.5 Training Strategy . . . 26
2.6.6 Hybrid Learning with Domain Knowledge . . . 27
2.6.7 Evaluation Metrics . . . 27
2.6.8 Key Insight . . . 29
2.7 Summary . . . 29

3 System Design and Implementation **30**
3.1 Overview . . . 30
3.2 Hardware Platforms . . . 30
3.2.1 Red Pitaya STEMlab 122-16 (SDR Front-End) . . . 30
3.2.2 Xilinx ZCU104 MPSoC (ML Inference Platform) . . . 31
3.3 System Overview . . . 32
3.4 Data Acquisition System Architecture . . . 32
3.5 Wideband Signal Detection . . . 33
3.5.1 Wideband Processing Architecture . . . 34
3.5.2 LO Sweep Mechanism . . . 36
3.5.3 FFT-Based Spectrum Analysis . . . 37
3.5.4 Spectrum Stitching . . . 38
3.5.5 Signal Detection . . . 38
3.6 Narrowband Signal Analysis . . . 39

3.6.1 Frequency Tuning and Locking . . . 39
3.6.2 Digital Downconversion and Decimation . . . 39
3.6.3 Narrowband Processing Architecture . . . 39
3.6.4 IQ Data Processing . . . 41
3.6.5 Key Role of Narrowband Stage . . . 41
3.7 Dataset Generation and Preprocessing . . . 42
3.7.1 Motivation and UHF Band Selection . . . 42
3.7.2 Protocol Selection, Devices, and Application Context . . . 43
3.7.3 Standards-Based Parameterization . . . 44
3.7.4 Sampling Strategy and Channel Modeling . . . 45
3.7.5 Signal Synthesis . . . 45
3.7.6 Preprocessing Pipeline . . . 45
3.7.7 Summary . . . 46
3.7.8 Channel Modeling . . . 46
3.8 Dataset Generation and Feature Extraction . . . 48
3.8.1 Signal Generation . . . 48
3.8.2 Modulation Schemes . . . 48
3.8.3 Channel Modeling . . . 49
3.8.4 Feature Extraction . . . 49
3.8.5 Dataset Structure . . . 50
3.8.6 Dataset Generation Flowchart . . . 51
3.9 Machine Learning Model Development . . . 52
3.9.1 Artificial Neural Network (ANN) . . . 52
3.9.2 Convolutional Neural Network (CNN) . . . 55
3.9.3 Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) 57
3.9.4 Model Comparison and Performance Evaluation . . . 61
3.10 FPGA Implementation . . . 62
3.10.1 System Architecture . . . 62
3.10.2 Memory Mapping and Addressing . . . 63
3.10.3 Resource Utilization . . . 64
3.10.4 System-Level Integration . . . 64
3.11 Summary . . . 65

4 Results and Discussion **66**
4.1 Overview . . . 66
4.2 Wideband Detection Performance . . . 66
4.3 Narrowband Signal Analysis Results . . . 67
4.4 Machine Learning Model Performance . . . 68
4.4.1 Model Architecture and Deployment Strategy . . . 69

4.4.2 Performance Evaluation 69
4.4.3 Model Performance Comparison 70
4.4.4 Modulation Classification (CNN) deployed in ZCU104 MPSOC+ . . . 70
4.4.5 Performance Across SNR 72
4.4.6 Key Insight 72
4.5 Real-Time System Performance 73
4.6 Comparative Analysis 75
4.7 Discussion 76

5 Summary and Future Work **78**

List of Figures

2.1 Multirate CIC decimation filter showing integrator stages, programmable rate change, and comb stages. 10
2.2 Block diagram of a CIC decimator followed by an FIR compensation filter. The CIC stage performs efficient coarse decimation, while the FIR filter compensates for passband attenuation and provides improved amplitude accuracy. 12
2.3 Illustrative spectral occupancy of communication protocols in the 390–480 MHz UHF band. 17
2.4 Structure of an Artificial Neural Network showing input, hidden, and output layers. 23
2.5 General CNN architecture illustrating convolution, pooling, and fully connected layers. 25
2.6 Architecture of a CNN–LSTM hybrid model where convolutional layers extract spatial or spectral features from input signals, and LSTM layers capture temporal dependencies across the sequence for final prediction. 26

3.1 Red Pitaya STEMlab 122-16 used for RF signal acquisition and FPGA-based preprocessing. 31
3.2 Xilinx ZCU104 MPSoC platform used for machine learning inference and system control. 32
3.3 Experimental setup showing the Red Pitaya SDR platform connected to an antenna and signal generator for real-time RF signal acquisition and testing. . . 33
3.4 Enhanced block diagram of the wideband signal detection system. The processing system controls the DDS through a control register for LO sweeping, while system status and data transfer are managed through a DMA interface and status register via the ACP (Accelerator Coherency Port). The FPGA performs signal acquisition, digital downconversion, and filtering, while the receiver performs FFT-based spectrum analysis and signal detection. 35

3.5 Vivado block design of the implemented wideband signal detection system on the Red Pitaya FPGA. The architecture includes ADC interfacing, DDS-based digital downconversion, DSP48-based mixing, CIC decimation, FIR compensation filtering, AXI streaming infrastructure, and DMA-based data transfer to the processing system through the ACP (Accelerator Coherency Port). Control and status registers are used to manage LO sweeping and monitor system operation. 36
3.6 Enhanced conceptual block diagram of the narrowband receiver (RX_0). The DDS-based local oscillator is controlled by the processing system through a control register, enabling precise frequency tuning. The downconverted signal is processed through CIC and FIR filtering, buffered in FIFO memory, and streamed to the receiver. System status and buffer information are monitored through a status register, facilitating reliable real-time operation. 40
3.7 Vivado block design of the implemented narrowband receiver on the Red Pitaya FPGA. The architecture includes DDS-based frequency tuning, DSP48-based mixing, CIC decimation, FIR compensation filtering, AXI streaming modules, and FIFO buffering for real-time data transfer to the processing system. 41
3.8 Flowchart of the RF dataset generation pipeline. The process begins with parameter initialization and message generation, followed by protocol-specific modulation, channel modeling (frequency offset and noise), and segmentation. Features are extracted from both time and frequency domains and stored as CSV samples. The process iterates over multiple channels and SNR conditions to generate a balanced dataset. 51
3.9 Channel-aware ANN architecture for RF signal classification. 53
3.10 Confusion matrix for ANN-based modulation classification. 53
3.11 Accuracy comparison of ANN and rule-based inference across different SNR levels. 54
3.12 Residual CNN architecture for RF signal classification. 56
3.13 Confusion matrix for CNN-based modulation classification. 56
3.14 Accuracy comparison between CNN and rule-based inference across different SNR levels. 57
3.15 CNN-LSTM architecture combining convolutional feature extraction with bidirectional LSTM-based temporal modeling. 59
3.16 Confusion matrix for CNN–LSTM-based modulation classification. 59
3.17 Accuracy comparison of CNN-LSTM and rule-based inference across different SNR levels. 60
3.18 FPGA block diagram showing wideband and narrowband processing chains . . 62
3.19 AXI address mapping for wideband and narrowband modules 63
3.20 Post-synthesis FPGA resource utilization 64

4.1 Wideband spectrum showing stitched FFT output across the scanned frequency range. Detected signals are visible across multiple frequency segments. 66
4.2 Narrowband IQ and FFT visualization at a fixed LO frequency of 45.3 MHz. . . 68
4.3 Confusion matrix for CNN-based modulation classification. 71
4.4 Accuracy comparison of CNN and rule-based inference across different SNR levels. 72
4.5 End-to-end hardware setup showing signal transmission from the signal generator, RF reception using the Red Pitaya SDR (122-16), and signal processing on the ZCU104 FPGA platform. 74
4.6 Real-time dashboard showing wideband spectrum sensing and narrowband signal tracking. The system detects signals in the wideband view and dynamically locks the narrowband receiver (±384 kHz) onto the detected frequency for detailed analysis. 75
4.7 Terminal output from the ZCU104 platform showing real-time signal processing and analysis and system-level execution. 75

Chapter 1

Introduction

Modern communication, surveillance, and scientific sensing systems rely heavily on the ability to detect, characterize, and classify radio-frequency (RF) signals in real time. As terrestrial, airborne, and spaceborne communication infrastructures continue to expand, the RF spectrum has become increasingly crowded and dynamic. This congestion introduces challenges in interference mitigation, autonomous spectrum management, and situational awareness. Consequently, there is a growing demand for highly efficient, low-latency, and intelligent RF processing architectures capable of operating reliably under stringent timing and power constraints.

Field-Programmable Gate Arrays (FPGAs) have emerged as one of the most promising platforms for implementing real-time digital signal processing (DSP) pipelines. Their inherent hardware-level parallelism, deterministic execution, and reconfigurable nature make them ideally suited for time-critical RF applications where CPUs and GPUs fall short. Recent advances in FPGA architectures, such as the Zynq-7000 and Zynq UltraScale+ MPSoC families, combine programmable logic with embedded processors and specialized DSP slices, enabling compact, intelligent, and hardware-accelerated RF systems.

This thesis presents the design and implementation of an intelligent wideband RF signal detection and classification system using Software-Defined Radio (SDR) and FPGA platforms. The proposed architecture follows a dual-stage approach, integrating wideband spectrum sensing with narrowband signal analysis and machine learning-based classification. A front-end SDR performs wideband scanning using FFT-based techniques and local oscillator (LO) sweeping to detect active signals across a large frequency span. Once detected, a narrowband receiver dynamically locks onto the signal of interest and extracts high-resolution I/Q samples for detailed analysis and classification.

In addition to real-time signal detection, the system incorporates a data-driven machine learning pipeline supported by a large-scale custom dataset. The proposed dataset models multiple real-world communication protocols operating in the UHF band (390–480 MHz), including public safety radio, walkie-talkie systems, meteorological radiosondes, SCADA telemetry, RFID devices, paging systems, and remote control transmissions. By incorporating protocol-specific

bandwidths, modulation schemes, and carrier frequencies, the dataset closely replicates realistic spectrum usage patterns. Furthermore, channel effects such as additive white Gaussian noise (AWGN) across a wide SNR range (-20 dB to +25 dB) and randomized frequency offsets are introduced to simulate practical RF environments. Each signal is processed to extract both time-domain and frequency-domain features, enabling the learning of distinct spectral signatures for robust classification.

The chapter begins by outlining the fundamental concepts behind RF detection and classification, followed by a discussion of the limitations associated with traditional DSP techniques when applied to high-throughput and low-latency scenarios. It then motivates the adoption of FPGA-oriented architectures and concludes by identifying the research gaps and defining the objectives that guide the development carried out in this work.

1.1 RF Signal Detection and Classification

RF signal detection determines whether a meaningful emission is present within a monitored bandwidth, while classification identifies the underlying modulation format or signal type. Together, these tasks enable essential functionalities such as spectrum monitoring, intelligent signal routing, adaptive communications, and interference mitigation.

In the proposed system, detection is performed using FFT-based spectral analysis combined with power and signal-to-noise ratio (SNR)-based peak detection. A wideband scanning receiver sweeps across the frequency range and reconstructs the spectrum through stitched FFT segments, enabling efficient identification of signal presence. Classification is then performed on high-resolution I/Q samples extracted by a narrowband receiver, allowing the system to distinguish between different modulation schemes and communication protocols.

Historically, these tasks were performed at ground stations or on high-performance computing clusters. However, emerging domains such as CubeSats, unmanned aerial systems (UAS), remote sensing payloads, and distributed spectrum-monitoring networks demand that detection and classification occur locally and autonomously. This requirement motivates the development of hardware-accelerated, embedded platforms such as FPGAs.

1.1.1 Applications Across Communications and Intelligent RF Systems

RF detection and classification play a crucial role in a wide range of applications:

- **Satellite and Deep-Space Communications:** Onboard monitoring of signal health, interference events, and adaptive modulation enables autonomous communication links with minimal ground intervention.
- **Electronic Warfare and Spectrum Security:** Identifying hostile transmissions, jammers, or anomalous emitters requires real-time detection and classification under noisy

and adversarial conditions.

- **Software-Defined Radio (SDR) and Cognitive Radio:** Systems dynamically sense and adapt to spectrum availability, requiring fast detection and intelligent classification.
- **Autonomous RF Monitoring Systems:** Wideband detection combined with narrowband analysis enables scalable and real-time spectrum intelligence platforms.
- **Radio Astronomy and RFI Mitigation:** High-speed spectral analysis is essential for detecting weak astrophysical signals and filtering terrestrial interference.

Across all these domains, common requirements emerge: high throughput, deterministic latency, and robust performance in noisy environments. These attributes are ideally met by FPGA-based DSP architectures.

1.2 From Classical DSP to FPGA-Based Acceleration

Although CPUs and GPUs offer flexibility and high floating-point capability, they often fail to meet the strict timing constraints of real-time RF systems. Their architectural limitations make them unsuitable for deterministic, high-throughput streaming workloads.

1.2.1 Limitations of CPU/GPU Solutions

- **Non-deterministic execution:** OS scheduling, cache behavior, and interrupts introduce unpredictable latency.
- **High power consumption:** GPUs consume significant power, making them unsuitable for embedded or spaceborne systems.
- **I/O bottlenecks:** Continuous high-rate data streams often exceed memory and bus-transfer capabilities.
- **Inefficiency in fine-grained parallelism:** RF DSP operations such as FFT, filtering, and decimation are inherently pipeline-friendly and better suited for hardware implementation.

1.2.2 Advantages of FPGA Architectures

FPGAs offer several benefits over CPU/GPU platforms:

- **Hardware-level parallelism** enabling simultaneous wideband scanning and narrowband processing.
- **Deterministic execution** with cycle-accurate timing.

- **Energy-efficient processing** suitable for embedded systems.
- **Custom high-speed interfaces** such as AXI-Stream and DMA pipelines.
- **Tight integration with ML inference** on MPSoC platforms.

These strengths enable the realization of a dual-stage RF processing architecture combining wideband detection and narrowband classification in real time.

1.3 Research Gap and Motivation

Understanding and analyzing the RF communication spectrum requires a complete processing pipeline that operates continuously in real time:

$$\text{Acquire} \rightarrow \text{Detect} \rightarrow \text{Classify} \rightarrow \text{Report}$$

The **acquire** stage involves wideband signal capture from the RF environment, followed by **detection** to identify the presence and location of signals within the spectrum. The **classification** stage determines the modulation or protocol type, while the **report** stage provides actionable information such as signal parameters and system-level insights.

Although significant research has been conducted on individual components of this pipeline, existing literature predominantly focuses on isolated stages such as signal detection, modulation classification, or machine learning-based analysis. Very few works address the integration of all stages into a unified, real-time system.

The key research gaps identified are:

- **Lack of End-to-End Real-Time Pipeline:** Most existing systems do not implement the complete backend pipeline from acquisition to classification and reporting. Instead, they rely on fragmented approaches where different stages are handled independently.
- **Limited Realistic RF Datasets:** Available datasets often consist of isolated modulation schemes under controlled conditions. They do not capture realistic spectrum scenarios where multiple signals coexist, overlap, and share frequency bands, which is a common characteristic of practical wireless environments.
- **Absence of Wideband–Narrowband Integration:** Many systems either focus on wideband sensing for detection or narrowband processing for analysis, but do not combine both in a structured manner. The transition from detection to detailed signal extraction is often missing.
- **Limited Embedded ML Integration:** While machine learning models have shown strong performance in RF signal classification, they are typically implemented in offline

or GPU-based environments. Their integration within FPGA-based real-time systems remains limited.

- **Scalability and Throughput Challenges:** Real-time RF systems generate extremely high data rates. Let R_s denote the sampling rate and B the bit-width. The resulting data throughput is:

$$T_{\text{raw}} = R_s \times B \times 2$$

For wideband SDR systems, this leads to massive data volumes, making storage and post-processing impractical. CPU/GPU-based solutions also introduce non-deterministic latency, which is unsuitable for time-critical applications.

These limitations highlight the need for a unified system capable of handling the entire RF processing pipeline in real time. In particular, there is a requirement for an architecture that combines:

- Efficient wideband spectrum acquisition and scanning
- Reliable signal detection and localization
- Dynamic narrowband tuning for detailed analysis
- Real-time machine learning-based classification
- Continuous operation with minimal latency and data loss

This work is motivated by addressing these challenges through the development of a fully integrated FPGA-based SDR framework. By combining wideband sensing, adaptive signal extraction, and embedded machine learning, the proposed system aims to bridge the gap between theoretical RF analysis techniques and practical, real-time spectrum intelligence applications.

1.3.1 Motivation

This work is motivated by the need to build intelligent RF systems capable of:

- autonomous wideband spectrum monitoring,
- real-time detection and classification,
- low-latency adaptive response,
- efficient handling of high-throughput RF data, and
- scalable deployment in embedded environments.

The combination of SDR-based acquisition, FPGA-based DSP, and ML-driven inference provides a powerful framework for next-generation RF intelligence systems.

1.3.2 Research Objectives

The primary objectives of this thesis are:

- To design a dual-stage architecture where wideband sensing continuously monitors the spectrum, and narrowband processing analyzes detected signals in detail.
- To extract and preprocess features from detected signals using narrowband analysis for machine learning-based identification of bandwidth, carrier frequency, and modulation.
- To develop a realistic large-scale RF dataset representing UHF communication protocols.
- To evaluate the quality and effectiveness of the generated dataset using machine learning models (ANN, CNN, CNN-LSTM) for RF signal classification.
- To deploy the complete system on FPGA platforms with hardware-software co-design.
- To validate system performance in terms of detection accuracy, classification accuracy, latency, and throughput.

This work establishes a practical and scalable framework for intelligent RF sensing systems, bridging the gap between traditional DSP pipelines and modern AI-enabled spectrum intelligence.

Chapter 2

Theoretical Background

This chapter presents the theoretical foundations and hardware implementation aspects of the key digital signal processing (DSP) modules employed in the FPGA-based RF processing chain developed in this work. The primary components include the Cascaded Integrator–Comb (CIC) decimator for efficient multirate processing, the FIR compensation filter for correcting passband distortion, and the Fast Fourier Transform (FFT) with Hanning windowing for accurate spectral analysis and signal frequency identification.

These modules collectively perform signal conditioning and feature extraction, forming the bridge between raw RF data and machine learning-based classification. The discussion covers both the mathematical principles underlying each algorithm and the practical considerations involved in their realization using Verilog and FPGA IP integrator tools. Particular emphasis is placed on windowing effects in FFT, bit-width growth, timing control, numerical stability, saturation behavior, and architectural trade-offs dictated by resource constraints and real-time throughput requirements.

2.1 Direct Digital Synthesizer (DDS)

2.1.1 Role of DDS in an RF Processing Chain

In digital RF receivers, frequency translation is required to shift an incoming signal from its carrier frequency to baseband or an intermediate frequency for further processing. A Direct Digital Synthesizer (DDS) is used to generate precise, digitally controlled sinusoidal waveforms that act as a local oscillator for this purpose through digital mixing. Compared to conventional analog local oscillators, a DDS offers fine frequency resolution, accurate phase control, and rapid, phase-continuous tunability, making it highly suitable for FPGA-based Software Defined Radio (SDR) systems.

2.1.2 Conceptual Basis of DDS

A DDS generates a sinusoidal waveform by accumulating phase over time and mapping the phase to amplitude using a lookup table. The core components of a DDS are:

- Phase accumulator
- Phase-to-amplitude converter (lookup table)
- Optional amplitude scaling or modulation block

The phase accumulator updates according to:

$$\phi[n] = \phi[n-1] + \Delta\phi \pmod{2\pi}$$

where $\Delta\phi$ is the phase increment that controls the output frequency.

2.1.3 Frequency Control and Resolution

The output frequency of the DDS is given by:

$$f_{\text{out}} = \frac{M}{2^N} f_{\text{clk}}$$

where:

- N = phase accumulator width
- M = frequency tuning word
- f_{clk} = system clock frequency

The frequency resolution is:

$$\Delta f = \frac{f_{\text{clk}}}{2^N}$$

Thus, increasing the accumulator width results in finer frequency resolution.

2.1.4 Sinusoidal Generation

The accumulated phase is used to address a sine/cosine lookup table to generate discrete samples of a complex exponential:

$$x[n] = \cos(\phi[n]) + j \sin(\phi[n])$$

This provides the in-phase (I) and quadrature (Q) components required for coherent digital signal processing.

2.1.5 Digital Mixing for Frequency Translation

The DDS output acts as a numerically controlled local oscillator and is multiplied with the incoming RF signal. For a complex input signal $r[n]$, the downconverted output is:

$$y[n] = r[n] \cdot x^*[n]$$

This operation shifts the signal spectrum to baseband (or a desired intermediate frequency), while preserving amplitude and phase information, enabling efficient filtering and decimation.

2.1.6 Implementation Considerations

When implemented in FPGA, the DDS design must consider:

- Phase accumulator bit-width (determines frequency resolution and spectral purity)
- Lookup table size (trade-off between memory usage and amplitude accuracy)
- Phase truncation and quantization effects leading to spurious tones
- Pipeline design and word-length optimization for high-speed operation

Efficient implementations often use truncated phase values and exploit symmetry properties of sine functions to reduce memory requirements while maintaining acceptable performance.

The Direct Digital Synthesizer (DDS) offers several advantages, including fine frequency and phase tuning resolution, fast and phase-continuous frequency switching, and a fully digital, repeatable implementation. It can be easily integrated with FPGA-based DSP chains and supports direct generation of quadrature (I/Q) signals. In this work, the DDS plays a crucial role in enabling digital downconversion, allowing wideband RF signals to be translated to baseband prior to CIC decimation, FIR filtering, and FFT-based spectral analysis.

2.2 CIC Decimation Filter Architecture

2.2.1 Role of Decimation in an RF Processing Chain

In the implemented RF processing pipeline, the analog-to-digital converter (ADC) operates at a high sampling rate of 122.88 MHz to preserve signal fidelity and avoid aliasing. However, subsequent digital signal processing (DSP) blocks such as filtering, feature extraction, and classification do not require such a high data rate. Therefore, decimation is essential to reduce the sampling rate, thereby lowering computational complexity, easing timing constraints, and minimizing power consumption.

To prevent aliasing during downsampling, a low-pass filtering stage must precede or be integrated with the rate reduction process. In this design, a Cascaded Integrator-Comb (CIC) filter

is employed, as it efficiently combines filtering and decimation without requiring multipliers, making it highly suitable for FPGA-based implementations.

2.2.2 Conceptual Basis of CIC Filtering

The CIC filter is derived from the concept of a recursive moving-average filter, implemented efficiently using integrator and comb stages. Instead of performing explicit convolution, the CIC structure uses accumulative summation (integration) followed by finite differencing (comb operation), significantly reducing hardware complexity.

Cascading multiple stages produces a sinc-shaped frequency response, which provides inherent low-pass filtering with regularly spaced nulls. While this response is computationally efficient, it introduces passband droop, which is typically corrected using a compensation FIR filter in later stages of the signal chain.

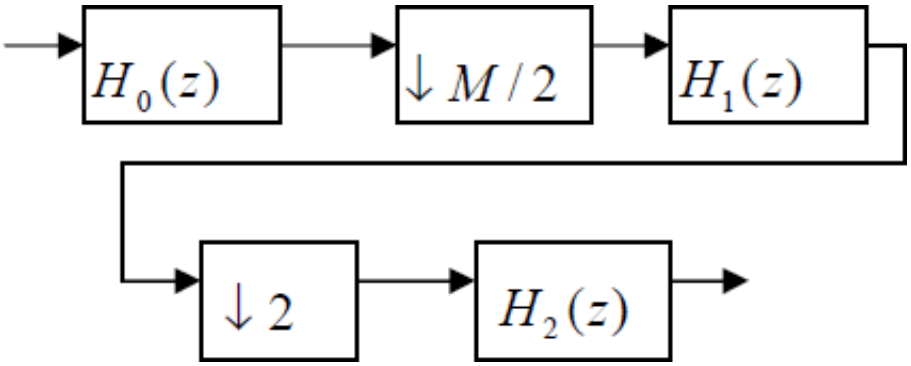


Figure 2.1: Multirate CIC decimation filter showing integrator stages, programmable rate change, and comb stages.

2.2.3 CIC Decimator Structure

The CIC decimator implemented in this work uses the Xilinx CIC Compiler IP configured for decimation mode. The structure consists of $N = 6$ integrator stages operating at the input sampling rate, followed by a programmable downsampler with decimation factor R, and $N = 6$ comb stages operating at the reduced rate. The differential delay is set to $M = 1$.

The general transfer function of the CIC filter is given by:

$$H_{\text{CIC}}(z) = \left(\frac{1 - z^{-RM}}{1 - z^{-1}} \right)^N .$$

With $N = 6$ and $M = 1$, the response becomes:

$$H_{\text{CIC}}(z) = \left(\frac{1 - z^{-R}}{1 - z^{-1}} \right)^6 .$$

2.2.4 Parameter Configuration in This Work

The CIC filter is configured using the Xilinx CIC Compiler with the following parameters:

- Number of stages: $N = 6$
- Filter type: Decimation
- Programmable decimation factor: $R \in [160, 2560]$
- Initial decimation factor: $R = 1280$
- Differential delay: $M = 1$
- Input sampling frequency: 122.88 MHz
- Clock frequency: 122.88 MHz
- Input data width: 24 bits
- Output data width: 32 bits
- Quantization: Truncation

The decimation factor is dynamically configurable at runtime via an AXI-stream configuration interface, enabling flexible adaptation to different signal bandwidth requirements.

2.2.5 Integrator and Comb Operation

Each integrator stage performs accumulation at the input rate:

$$I_k[n] = I_k[n-1] + I_{k-1}[n],$$

where $k = 1, 2, ..., 6$.

After decimation by factor R, the comb stages operate at the reduced sampling rate and compute finite differences:

$$C_k[n] = C_{k-1}[n] - C_{k-1}[n-R].$$

This multirate structure ensures that high-speed accumulation is performed efficiently, while subtraction operations are executed at a lower rate, reducing switching activity and resource usage.

2.2.6 Frequency Response and Bit Growth

The magnitude response of the CIC filter follows a sinc^N characteristic:

$$|H(e^{j\omega})| = \left| \frac{\sin(\omega R/2)}{\sin(\omega/2)} \right|^6 .$$

The filter introduces a gain of:

$$G = (RM)^N = R^6,$$

which can be very large for high decimation factors. Therefore, internal word lengths are increased, and truncation is applied to produce a 32-bit output. This ensures a balance between numerical precision and hardware resource utilization.

2.2.7 Implementation Considerations

The CIC filter is implemented using FPGA fabric without dedicated DSP slices, relying solely on adders and registers. The design is clocked using a 122.88 MHz PLL output and synchronized with an active-low reset signal. Input data is streamed from the preceding multiplier stage, and valid control signals ensure proper data flow.

Due to its multiplier-free architecture and programmable decimation capability, the CIC filter forms a critical component in the digital downconversion chain, efficiently reducing data rate while preparing the signal for subsequent compensation filtering and spectral analysis.

2.3 FIR Compensation Filtering

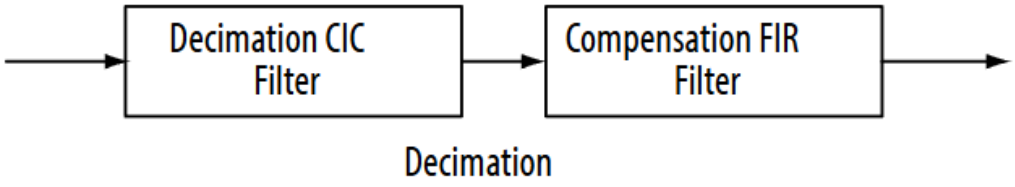


Figure 2.2: Block diagram of a CIC decimator followed by an FIR compensation filter. The CIC stage performs efficient coarse decimation, while the FIR filter compensates for passband attenuation and provides improved amplitude accuracy.

CIC decimators provide efficient rate reduction but introduce noticeable passband droop due to their inherent sinc^N response. To restore amplitude flatness and suppress residual aliasing, a fixed-coefficient FIR compensation filter is applied immediately after the CIC stage. Operating at the already-reduced sampling rate, the FIR filter improves spectral fidelity with minimal resource impact.

2.3.1 Implemented FIR Architecture

The implemented FIR uses 32 taps with 16-bit coefficients stored in a ROM array. Incoming samples are buffered through a 31-sample delay line, and each output is computed using the standard convolution sum

$$y[n] = \sum_{k=0}^{31} h[k]\, x[n-k].$$

The internal accumulator width is sized to

$$\text{ACC_WIDTH} = \text{DATA_WIDTH} + \text{TAP_WIDTH} + \lceil \log_2(32) \rceil,$$

ensuring sufficient headroom during the multiply–accumulate (MAC) cycle. Because data arrives at the decimated rate, the entire MAC loop executes efficiently within a single clock cycle without stressing DSP resources.

For each valid input, the filter performs a full MAC sweep over all taps, updates the delay line, and outputs the upper 32 bits of the accumulator as the final filtered sample. This pipelined structure keeps latency predictable, maintains throughput at the decimated sampling rate, and provides the necessary passband correction for the CIC stage.

2.4 Windowed FFT for Spectrum Stitching

The windowed FFT with spectrum stitching is fundamental to the proposed system, as it enables accurate wideband spectrum reconstruction from limited bandwidth observations. The use of windowing reduces spectral leakage, improving detection of weak and closely spaced signals, while stitching ensures continuous frequency coverage essential for reliable wideband sensing and tracking.

2.4.1 FFT-Based Signal Representation

RF signals acquired in the time domain are transformed into the frequency domain using the Fast Fourier Transform (FFT), enabling identification of spectral components. The FFT efficiently computes the Discrete Fourier Transform (DFT):

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

where $X[k]$ represents the spectral content at each frequency bin. This transformation allows detection of dominant frequency components within the signal.

2.4.2 Windowing and Spectral Leakage Reduction

In practical systems, finite-length signal observation introduces discontinuities, leading to spectral leakage. To mitigate this, windowing is applied prior to FFT:

$$x_w[n] = x[n] \cdot w[n]$$

In this work, the Hann window is used:

$$w[n] = 0.5\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right)$$

This reduces leakage by smoothly tapering the signal edges, providing a balance between frequency resolution and sidelobe suppression, making it suitable for general RF analysis.

2.4.3 Spectrum Stitching for Wideband Analysis

Since the receiver has limited instantaneous bandwidth, wideband spectrum analysis is achieved by combining multiple FFT segments obtained through frequency sweeping. Each segment is downconverted, windowed, and transformed via FFT, and then mapped to its absolute frequency:

$$f_{\text{global}} = f_{\text{DDS}} + f_{\text{FFT}}$$

Overlapping adjacent segments ensures smooth transitions and avoids discontinuities in the reconstructed spectrum. This approach enables efficient wideband signal detection and analysis within FPGA-based SDR systems.

The combined use of windowed FFT and spectrum stitching provides accurate spectral representation, reduced leakage, and reliable wideband reconstruction for real-time RF applications :contentReference[oaicite:0]index=0.

2.5 Protocol-Centric Dataset Design

The dataset is constructed using a protocol-driven approach, where each class corresponds to a specific wireless communication system operating within the UHF band. Rather than modeling abstract modulation schemes, the dataset incorporates standard-compliant protocols to ensure that the generated signals reflect realistic spectrum usage and device behavior.

The selected protocols span multiple domains, including public safety, industrial telemetry, meteorological sensing, RFID systems, paging, and short-range devices. Each protocol is defined by its modulation scheme, channel bandwidth, symbol structure, and information encoding domain (frequency, phase, or amplitude), resulting in distinct time-frequency characteristics. :contentReference[oaicite:0]index=0

2.5.1 Public Safety Systems

Public safety communication systems operate under strict regulatory constraints and require high reliability in congested environments.

- **Narrowband FM (NBFM):** Uses continuous frequency deviation (typically ±2.5–3 kHz) within 12.5 kHz channels. The constant-envelope property provides robustness against noise and nonlinear distortion, making it suitable for analog voice transmission.
- **Digital Mobile Radio (DMR):** Employs 4-FSK modulation, mapping 2 bits to one of four discrete frequency tones. It operates using two-slot TDMA within a 12.5 kHz channel, achieving approximately 9.6 kbps per timeslot. The discrete tone structure creates identifiable spectral peaks.
- **Terrestrial Trunked Radio (TETRA):** Uses $\pi/4$-DQPSK with 4-slot TDMA in a 25 kHz channel. Information is encoded in phase transitions rather than amplitude or frequency, resulting in characteristic phase trajectories in the complex signal.

2.5.2 Walkie-Talkie Systems

Walkie-talkie systems operate in the 403–480 MHz band and are widely used for short-range voice communication in both civilian and industrial environments.

- **Analog Walkie-Talkies:** Typically use NBFM modulation, providing smooth spectral characteristics and robustness to noise due to their constant-envelope nature.
- **Digital Walkie-Talkies:** Often employ DMR-based 4-FSK modulation, introducing discrete spectral components and improved spectral efficiency through TDMA operation.

2.5.3 Meteorological Systems

Meteorological radiosonde systems are designed for long-range atmospheric data transmission with reliable performance under varying environmental conditions.

- **Radiosondes:** Operate in the 400.15–406 MHz band using frequency modulation schemes for transmitting temperature, pressure, and humidity data.
- **Gaussian-Filtered FM:** In some implementations, Gaussian filtering is applied to reduce spectral sidelobes and improve bandwidth efficiency.

2.5.4 Supervisory Control and Data Acquisition (SCADA Systems)

SCADA systems are used for industrial monitoring and control, requiring reliable and interference-resistant communication.

- **Binary FSK:** Uses frequency shift keying within narrowband channels (12.5 kHz or 25 kHz), enabling robust transmission in noisy environments.
- **Low Data Rate Telemetry:** Supports data rates from approximately 4.8 kbps to 19.2 kbps, optimized for reliability rather than throughput.

2.5.5 RFID Systems

RFID systems operating in the 433 MHz ISM band are designed for low-power identification and tracking applications.

- **ASK/OOK Modulation:** Uses amplitude-based schemes such as Amplitude Shift Keying and On-Off Keying, resulting in strong amplitude variations in the transmitted signal.
- **Low-Power Operation:** Emphasizes energy efficiency and simplicity, often leading to sparse spectral occupancy.

2.5.6 Paging Systems: POCSAG

Paging systems based on the POCSAG protocol are designed for reliable message delivery over wide coverage areas.

- **Binary FSK:** Employs fixed frequency deviation (e.g., ±4.5 kHz) within 25 kHz channels.
- **Structured Signaling:** The regular and predictable frequency transitions produce distinctive and repeatable spectral signatures.

2.5.7 Remote Control Systems

Remote control systems are used in short-range wireless applications such as key fobs, alarms, and IoT devices.

- **OOK Modulation:** The carrier is switched on and off to encode information, resulting in sharp temporal transitions.
- **Sparse Spectral Content:** Due to intermittent transmission, these signals exhibit discontinuous and sparse frequency-domain characteristics.

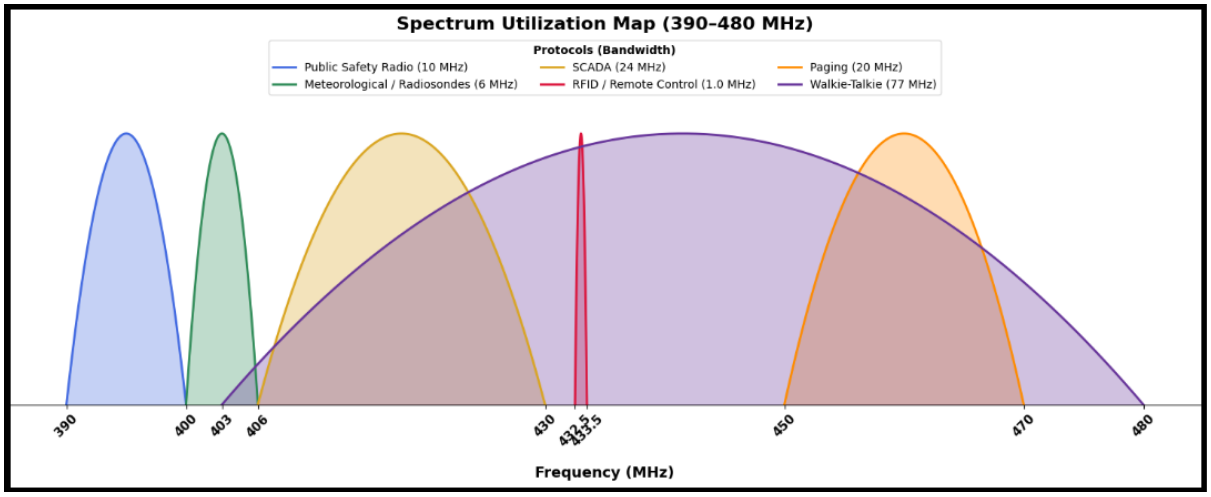


Figure 2.3: Illustrative spectral occupancy of communication protocols in the 390–480 MHz UHF band.

2.5.8 Signal Signature Perspective

To further understand the distinguishability of different protocols, their characteristic signatures in both time and frequency domains are summarized in Table 2.1. These signatures form the basis for feature extraction and classification.

Protocol	**Signature in FFT**	**Signature in Time Domain**
NBFM	Smooth continuous spectrum with no discrete tones	Continuous waveform with smooth phase variation
DMR (4-FSK)	Distinct spectral peaks corresponding to discrete frequency tones	Piecewise constant frequency transitions
TETRA (DQPSK)	Broad spectrum with phase-dependent structure	Discrete phase jumps between symbols
GFSK	Compact spectrum with reduced sidelobes	Smooth frequency transitions due to Gaussian filtering
BFSK (SCADA/POCSAG)	Two dominant frequency components	Binary switching between two frequency states
ASK	Strong carrier with amplitude variations	Envelope variation reflecting data symbols
OOK	Intermittent carrier (on/off bursts)	Sharp amplitude transitions (presence/absence)

Table 2.1: Signal Signature Perspective Across Protocols

These signatures directly correspond to the underlying modulation principles and are further illustrated using representative signal plots in the subsequent figures. The combination of time-domain and frequency-domain characteristics provides complementary information, enabling robust discrimination between protocols that may appear similar in one domain but differ in another.

2.5.9 Summary of Modulation Diversity

The dataset incorporates three primary information encoding domains:

- **Frequency-based modulation:** NBFM, FSK, GFSK
- **Phase-based modulation:** $\pi/4$-DQPSK (TETRA)
- **Amplitude-based modulation:** ASK, OOK

This protocol-centric modeling approach ensures that the dataset captures realistic communication behavior, spectral occupancy, and modulation diversity, providing a strong foundation for RF signal classification in complex wireless environments.

2.5.10 Feature Structuring and Dimensional Consistency

The extracted features are organized into a structured multi-domain representation that preserves both temporal dynamics and spectral characteristics of the RF signals. Specifically, each signal instance is represented using:

- **Time-domain sequences:** $I[n], Q[n], |x[n]|, \angle x[n]$
- **Frequency-domain vectors:** $|X[k]|, \angle X[k]$

The time-domain features capture instantaneous signal behavior, including amplitude variations, phase continuity, and transient dynamics. These are particularly important for distinguishing modulation schemes such as FM (smooth phase evolution), FSK (discrete frequency transitions), and OOK/ASK (amplitude variations).

The frequency-domain features provide a complementary global representation of the signal, capturing bandwidth occupancy, carrier offset, and spectral shape. This is essential for distinguishing protocols with similar time-domain behavior but different spectral footprints, such as narrowband versus wideband systems.

These feature components are concatenated to form a unified feature matrix, where each row represents a time sample and each column corresponds to a specific feature dimension. This structured representation ensures compatibility with neural network architectures, allowing the model to jointly learn temporal correlations and spectral patterns.

To maintain dimensional consistency across all samples, each signal is represented using a fixed-length window (2048 samples), ensuring uniform input size regardless of the original signal duration. This standardization is critical for batch-based training in deep learning models and enables efficient parallel processing.

Furthermore, all features are normalized to a consistent scale. Normalization prevents high-energy components (e.g., strong carriers or wideband signals) from dominating the learning

process and ensures numerical stability during training. This improves convergence behavior and allows the model to focus on relative feature variations rather than absolute amplitude differences.

2.5.11 Design Rationale

The feature extraction pipeline is designed to bridge classical signal processing and data-driven learning by preserving physically meaningful signal characteristics while remaining computationally efficient.

The primary design principles are as follows:

- **Modulation-Agnostic Representation:** Instead of performing explicit demodulation, the feature set retains raw signal characteristics. This allows the model to learn modulation patterns directly from data, making it adaptable to multiple protocols.
- **Time–Frequency Complementarity:** RF signals exhibit both local temporal behavior and global spectral structure. By combining time-domain and frequency-domain features, the representation captures complementary information necessary for robust classification.
- **Invariance to Channel Impairments:** Features such as magnitude, phase, and spectral shape are selected to remain informative under noise, frequency offset, and other channel effects, improving generalization to real-world conditions.
- **Scalability and Efficiency:** The feature representation is designed to support large-scale dataset generation and training, with fixed-size inputs and computationally efficient transformations such as FFT.
- **Compatibility with Deep Learning Architectures:** The structured feature matrix enables direct integration with convolutional and fully connected neural networks, allowing the model to exploit both spatial (feature-wise) and temporal correlations.

Overall, this feature representation forms a critical interface between classical DSP-based signal modeling and machine learning-based RF signal classification. By preserving essential signal characteristics while ensuring computational tractability, it enables the development of robust and generalizable RF intelligence systems.

2.6 Machine Learning Modeling

Machine learning (ML), particularly deep learning (DL), has emerged as a powerful approach for automatic radio frequency (RF) signal classification. Unlike traditional methods that de-

pend on handcrafted features or explicit demodulation, modern data-driven approaches learn discriminative representations directly from raw or minimally processed signal data.

Conventional RF classification techniques typically rely on manually engineered features such as higher-order statistics, cyclostationary properties, or demodulated parameters. Although effective under controlled conditions, these methods require prior knowledge of signal structure and often degrade in the presence of noise, frequency offsets, or unknown protocols. In contrast, deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) automatically extract relevant features from In-phase/Quadrature (I/Q) signals or time–frequency representations, enabling robust performance in complex and non-ideal environments.

In this work, machine learning is primarily used as a validation framework to assess the quality and realism of the generated RF dataset. The models operate on structured feature representations derived from baseband signals, capturing both spectral and temporal characteristics without requiring explicit demodulation.

2.6.1 Problem Formulation

The RF signal classification task is formulated as a supervised multi-task learning problem:

$$f : \mathbf{x} \rightarrow \{y_p, y_m, \mathbf{y}_r\}$$

where:

- $\mathbf{x}$ represents the input signal feature vector (e.g., time-domain and frequency-domain features)
- y_p denotes the protocol class (classification output)
- y_m denotes the modulation type (classification output)
- $\mathbf{y}_r = [f_c, BW]$ represents regression outputs, namely carrier frequency (f_c) and bandwidth (BW)

Here, $f(\cdot;\theta)$ represents a parametric function (typically a neural network), where θ denotes the set of learnable parameters (weights and biases). The goal of training is to optimize θ such that the predicted outputs match the ground truth values.

For classification tasks, the categorical cross-entropy loss function is used:

$$\mathcal{L}_{cls} = -\sum_{i=1}^{C} y_i \log \hat{y}_i$$

where:

- C is the total number of classes
- y_i is the true label (one-hot encoded)
- $\hat{y}_i$ is the predicted probability for class i

This loss function measures the difference between the true class distribution and the predicted probability distribution. It penalizes incorrect predictions more strongly when the model is confident but wrong.

For regression tasks, the Mean Squared Error (MSE) is used:

$$\mathcal{L}_{reg} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where:

- n is the number of samples
- y_i is the true value (e.g., actual frequency or bandwidth)
- $\hat{y}_i$ is the predicted value

MSE computes the average squared difference between predicted and actual values, ensuring that larger errors are penalized more heavily.

The overall loss is defined as a weighted combination of classification and regression losses:

$$\mathcal{L} = \alpha \mathcal{L}_{proto} + \beta \mathcal{L}_{mod} + \gamma \mathcal{L}_{reg}$$

where:

- $\mathcal{L}_{proto}$ is the cross-entropy loss for protocol classification
- $\mathcal{L}_{mod}$ is the cross-entropy loss for modulation classification
- $\mathcal{L}_{reg}$ is the regression loss (MSE)
- α, β, γ are weighting coefficients that balance the importance of each task

This formulation enables **multi-task learning**, where a single model simultaneously performs classification and regression, improving overall performance through shared feature learning.

2.6.2 Representation Learning

A key strength of deep learning lies in its ability to perform **representation learning**, where meaningful features are automatically extracted from raw input data instead of being manually designed.

The learned representations are hierarchical:

- **Low-level features:** basic waveform characteristics such as amplitude variations, phase continuity, and short-term signal fluctuations
- **Mid-level features:** structured patterns such as spectral envelopes, bandwidth occupancy, and modulation-specific signatures
- **High-level features:** abstract representations including protocol-specific behavior and temporal dependencies across signal sequences

This hierarchical feature extraction allows the model to learn invariant representations, making it robust to noise, channel impairments, frequency offsets, and variations in signal conditions.

2.6.3 Neural Network Models and Their Functioning

This work employs multiple deep learning architectures to model and classify RF signals. Each model captures different characteristics of the signal, including global feature relationships, local spectral structures, and temporal dependencies.

Artificial Neural Network (ANN):

An Artificial Neural Network (ANN) is a feedforward computational model inspired by biological neural systems, designed to approximate complex non-linear mappings between input features and outputs. It is widely used in classification and regression tasks due to its ability to learn global relationships across high-dimensional feature spaces.

An ANN consists of three primary components:

- **Input Layer:** Receives the input feature vector, such as combined time-domain and frequency-domain representations of RF signals
- **Hidden Layers:** Perform non-linear transformations to extract intermediate representations
- **Output Layer:** Produces final predictions such as protocol class, modulation type, or signal parameters

Each neuron performs a linear transformation followed by a non-linear activation:

$$y = \sigma\left(\sum_i w_i x_i + b\right)$$

where x_i are input features, w_i are learnable weights, b is a bias term, and $\sigma(\cdot)$ denotes an activation function such as Rectified Linear Unit (ReLU) or Sigmoid.

The ANN operates through a sequence of steps:

- **Forward Propagation:** Input features are propagated through the network to generate predictions
- **Loss Computation:** The predicted output is compared with ground truth using a loss function
- **Backpropagation:** Gradients are computed and used to update weights
- **Optimization:** Parameters are iteratively updated to minimize the loss

In this work, the ANN processes flattened feature vectors derived from RF signals, learning global relationships across features. However, due to its fully connected nature, it does not explicitly exploit local temporal or spectral structures.

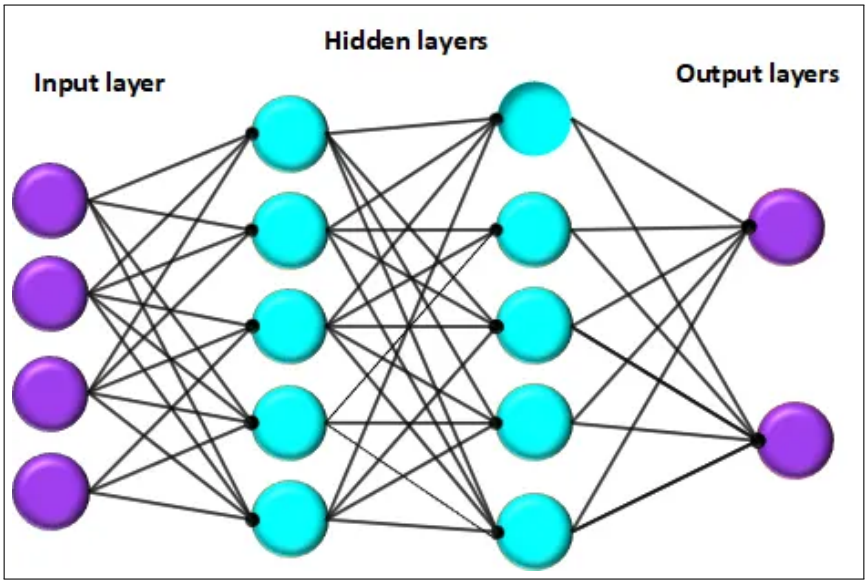


Figure 2.4: Structure of an Artificial Neural Network showing input, hidden, and output layers.

Convolutional Neural Network (CNN):

A Convolutional Neural Network (CNN) is a deep learning architecture designed to extract localized features from structured data using convolution operations. Unlike ANNs, CNNs exploit local connectivity and parameter sharing, making them highly efficient for processing signal and time-series data.

The fundamental operation in a CNN is convolution:

$$y[n] = \sum_{k} x[n-k] \cdot w[k]$$

where $x[n]$ is the input signal, $w[k]$ represents a learnable kernel, and $y[n]$ is the resulting feature map. The convolution operation captures local dependencies within the signal through sliding filters.

A typical CNN consists of:

- **Convolution Layers:** Extract local features such as spectral peaks, edges, and modulation patterns
- **Activation Functions (ReLU):** Introduce non-linearity
- **Pooling Layers:** Reduce dimensionality and improve robustness
- **Batch Normalization:** Stabilizes training and accelerates convergence
- **Fully Connected Layers:** Perform final classification or regression

CNNs learn hierarchical representations:

- Low-level: waveform variations and amplitude patterns
- Mid-level: spectral structures and bandwidth features
- High-level: modulation signatures and protocol characteristics

In this work, a one-dimensional CNN (1D-CNN) is used to process multi-channel RF signal representations (e.g., I, Q, magnitude, phase, and spectral features). The architecture incorporates residual connections to facilitate training of deeper networks and improve feature extraction capability.

CNNs are particularly suitable for RF signal analysis due to their ability to capture localized time-frequency structures, reduce model complexity through weight sharing, and learn robust representations under noisy conditions.

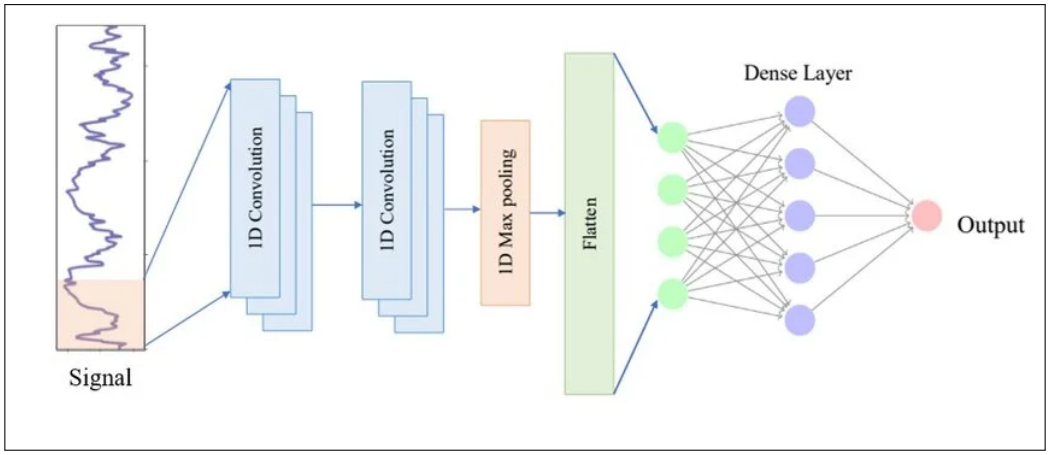


Figure 2.5: General CNN architecture illustrating convolution, pooling, and fully connected layers.

CNN–Long Short-Term Memory (CNN–LSTM) Hybrid Model:

The CNN–Long Short-Term Memory (CNN–LSTM) model combines convolutional feature extraction with temporal sequence modeling, enabling joint learning of spatial and temporal characteristics of RF signals.

In this architecture, CNN layers first extract discriminative local features from the input signal. These features are then passed to Long Short-Term Memory (LSTM) layers, which model temporal dependencies across the sequence.

The LSTM updates its hidden state as:

$$h_t = \text{LSTM}(x_t, h_{t-1})$$

where x_t is the input at time step t, h_{t-1} is the previous hidden state, and h_t is the updated state. LSTM networks employ gating mechanisms (input, forget, and output gates) to regulate information flow, allowing them to retain relevant temporal information and mitigate vanishing gradient issues.

The CNN–LSTM model operates in two stages:

- **CNN Stage:** Extracts local spectral and structural features
- **LSTM Stage:** Captures temporal dependencies and sequence dynamics

This hybrid approach is particularly effective for RF signals, where both instantaneous spectral features and time-varying behaviors (e.g., modulation transitions and burst patterns) are important.

Thus, the CNN–LSTM model provides a unified framework for learning joint spatio-temporal representations, leading to improved performance in RF signal classification and parameter estimation.

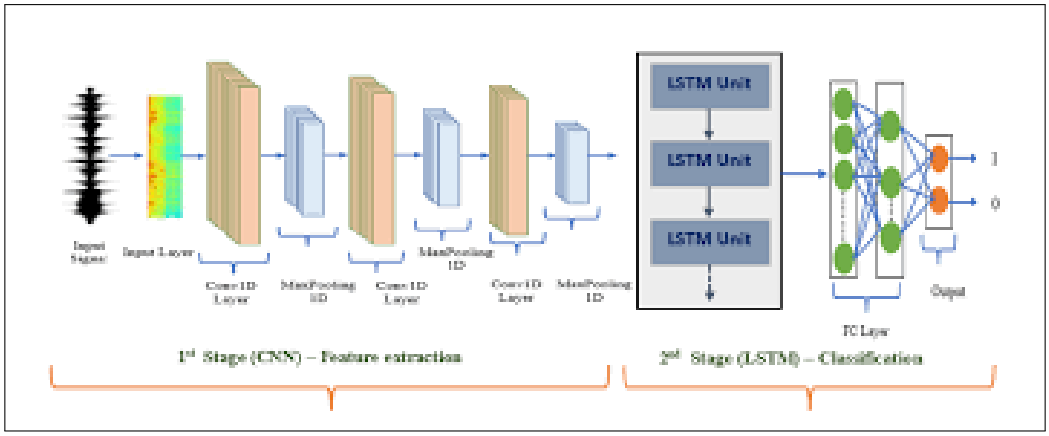


Figure 2.6: Architecture of a CNN–LSTM hybrid model where convolutional layers extract spatial or spectral features from input signals, and LSTM layers capture temporal dependencies across the sequence for final prediction.

2.6.4 Modeling Advantages for RF Signal Classification

The use of deep learning models in RF signal analysis offers several advantages:

- **End-to-End Learning:** Eliminates the need for explicit demodulation or manual feature engineering
- **Robustness:** Maintains performance under noise, signal-to-noise ratio (SNR) variations, and frequency offsets
- **Scalability:** Supports multiple protocols and modulation schemes within a unified framework
- **Generalization:** Adapts to unseen signal environments and heterogeneous RF conditions

2.6.5 Training Strategy

Model parameters are optimized using the Adaptive Moment Estimation (Adam) optimizer:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}$$

To improve training stability and generalization, the following techniques are employed:

- Batch Normalization: stabilizes learning by normalizing activations
- Dropout: reduces overfitting by randomly disabling neurons
- Early Stopping: prevents degradation in validation performance
- Gradient Clipping: avoids instability due to large gradients

2.6.6 Hybrid Learning with Domain Knowledge

To enhance classification reliability, a rule-based refinement stage is applied. The predicted modulation type, carrier frequency, and bandwidth are compared with known protocol characteristics, enabling improved discrimination between similar signal classes.

2.6.7 Evaluation Metrics

Model performance is evaluated using both classification and regression metrics to assess the effectiveness of the proposed framework.

Classification Metrics:

- **Accuracy:** Accuracy measures the proportion of correctly classified samples over the total number of samples:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

  where TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative) represent classification outcomes.

  Significance: Accuracy provides an overall measure of performance but may be misleading in imbalanced datasets.

- **Precision:** Precision measures the proportion of correctly predicted positive samples among all predicted positives:

$$\text{Precision} = \frac{TP}{TP + FP}$$

  Significance: High precision indicates low false positive rates, which is important when incorrect classification is costly.

- **Recall (Sensitivity):** Recall measures the proportion of actual positive samples that are correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN}$$

  Significance: High recall ensures that most relevant signals are detected, which is critical in RF monitoring applications.

- **F1-Score:** The F1-score is the harmonic mean of precision and recall:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Significance: Provides a balanced measure when both false positives and false negatives are important.

Regression Metrics:

- **Mean Absolute Error (MAE):** MAE measures the average absolute difference between predicted and actual values:

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

Significance: Provides a direct measure of average prediction error without penalizing large errors disproportionately.

- **Root Mean Squared Error (RMSE):** RMSE measures the square root of the average squared prediction error:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

Significance: Penalizes larger errors more strongly, making it useful when large deviations are undesirable.

- **Coefficient of Determination (R^2):** The R^2 score measures how well the model explains the variance in the data:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

where $\bar{y}$ is the mean of actual values.

Significance: Indicates goodness-of-fit, where $R^2 = 1$ represents perfect prediction and $R^2 = 0$ indicates no explanatory power.

2.6.8 Key Insight

The performance of the machine learning models serves as an indicator of dataset quality. High classification accuracy and reliable parameter estimation suggest that the dataset successfully captures realistic RF signal characteristics.

Thus, machine learning not only enables RF signal classification but also provides a systematic framework for validating the realism and effectiveness of the generated dataset for RF intelligence applications.

2.7 Summary

This chapter presented the data-driven component of the system, including dataset representation, feature extraction, and machine learning modeling. The integration of these elements with the DSP pipeline enables a complete framework for RF signal detection and intelligent classification in wideband environments.

Chapter 3

System Design and Implementation

3.1 Overview

This chapter presents the complete design and implementation of the proposed RF signal detection and classification system. The system integrates Software Defined Radio (SDR), digital signal processing (DSP), and machine learning (ML) into a unified end-to-end pipeline. It enables real-time wideband spectrum monitoring, automatic signal detection, and detailed narrowband analysis, along with intelligent classification of detected signals.

The data acquisition stage is implemented using the Red Pitaya 122-16 SDR platform, which provides high-speed analog-to-digital conversion for wideband RF signal capture. The acquired signals are then processed through FPGA-based DSP modules for preprocessing, including downconversion, filtering, and feature extraction.

Subsequently, the processed data is utilized for dataset generation and machine learning-based analysis. The machine learning models are designed for deployment on the Xilinx ZCU104 MPSoC (Multiprocessor System-on-Chip) platform, where programmable logic accelerates signal processing and the processing system supports inference and system control.

3.2 Hardware Platforms

3.2.1 Red Pitaya STEMlab 122-16 (SDR Front-End)

The Red Pitaya STEMlab 122-16 serves as the primary SDR front-end for RF signal acquisition. The platform is built around a Xilinx Zynq-7020 SoC, which integrates a dual-core ARM Cortex-A9 processing system with programmable logic, enabling a flexible hardware–software co-design framework.

It features a 14-bit analog-to-digital converter (ADC) operating at 122.88 MSPS, allowing high-resolution capture of wideband RF signals. The onboard FPGA is utilized to implement real-time DSP operations, including digital downconversion, filtering, and high-speed data



Figure 3.1: Red Pitaya STEMlab 122-16 used for RF signal acquisition and FPGA-based preprocessing.

transmission.

The system is designed with two parallel receiver paths:

- **Wideband Path:** Performs continuous spectrum scanning for signal detection
- **Narrowband Path:** Extracts high-resolution I/Q samples for detailed analysis

This dual-path architecture enables efficient wideband monitoring along with focused narrowband processing of detected signals. The reconfigurable FPGA fabric, combined with high-speed data interfaces, makes the platform well suited for real-time SDR applications and rapid prototyping of RF signal processing systems.

3.2.2 Xilinx ZCU104 MPSoC (ML Inference Platform)

The Xilinx ZCU104 MPSoC platform is used for machine learning-based signal classification. It combines a processing system (ARM Cortex-A53) with programmable logic, enabling a heterogeneous computing architecture.

The platform is designed to support hardware-accelerated inference using the Deep Processing Unit (DPU), allowing efficient execution of convolutional neural networks on streaming RF data. The narrowband I/Q data received from the SDR front-end is processed into structured features and provided to the ML model for classification.

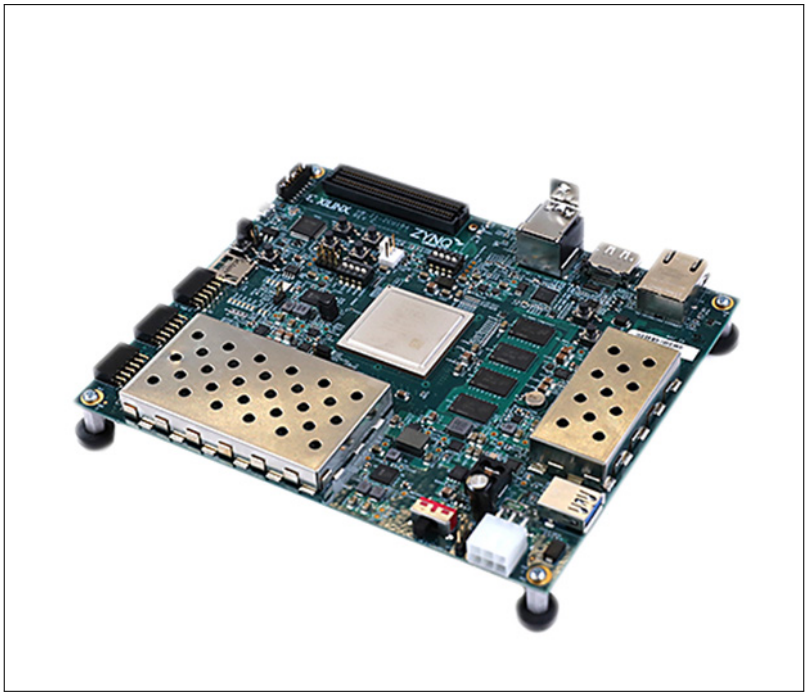


Figure 3.2: Xilinx ZCU104 MPSoC platform used for machine learning inference and system control.

This division of tasks—FPGA-based signal processing on the Red Pitaya and ML inference on the ZCU104—enables a scalable and real-time system architecture.

3.3 System Overview

The overall system is divided into four major stages:

- Wideband Signal Detection
- Narrowband Signal Analysis
- Dataset Generation and Machine Learning
- Hardware Implementation and Real-Time Deployment

3.4 Data Acquisition System Architecture

The architecture consists of FPGA-based signal processing blocks and a host-side processing unit. The system employs two parallel receiver chains:

- **Wideband Path**: Used for spectrum scanning and signal detection with a sampling rate of 15.36 MHz

- **Narrowband Path**: Used for high-resolution I/Q and FFT analysis of detected signals with a sampling rate of 768 kHz

The wideband receiver continuously scans the spectrum, while the narrowband receiver dynamically locks onto detected signals for detailed processing.

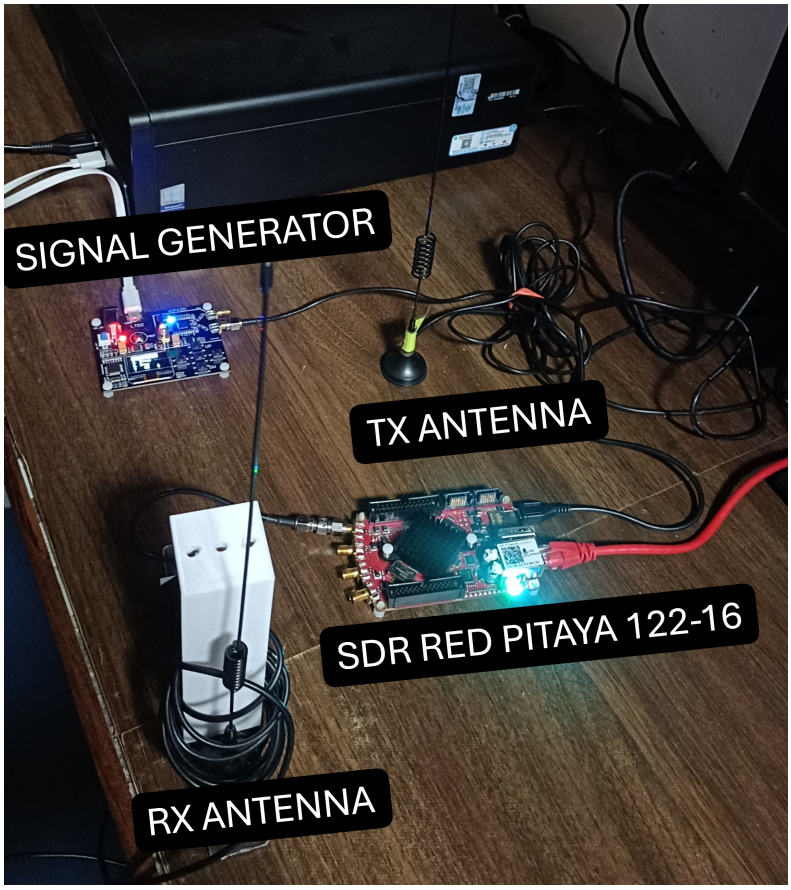


Figure 3.3: Experimental setup showing the Red Pitaya SDR platform connected to an antenna and signal generator for real-time RF signal acquisition and testing.

3.5 Wideband Signal Detection

The wideband signal detection stage is designed to continuously monitor a large RF spectrum and identify active signals in real time. The implementation follows a hardware–software co-design approach, where high-speed data acquisition and downconversion are performed on the FPGA (Red Pitaya), while spectral analysis and detection are carried out on the receiver (PC).

The system operates with an initial sampling rate of:

$$F_s = 122.88 \text{ MHz}$$

To reduce data rate while preserving the required bandwidth, a multi-stage decimation process is implemented using a Cascaded Integrator–Comb (CIC) filter followed by an FIR

compensation filter:

$$F_s' = \frac{F_s}{R_{\text{CIC}} \cdot R_{\text{FIR}}} = \frac{122.88}{4 \times 2} = 15.36 \text{ MHz}$$

This reduced sampling rate defines the instantaneous bandwidth processed per LO step.

3.5.1 Wideband Processing Architecture

The wideband processing chain is partitioned as follows:

- **FPGA (Red Pitaya):** ADC acquisition, digital downconversion (DDS + mixer), CIC decimation, FIR filtering, and AXI-based streaming
- **Processing System (ARM):** Control of LO sweep and data transfer via DMA (ACP interface)
- **Receiver (PC):** Windowing, FFT computation, spectrum stitching, visualization, and signal detection

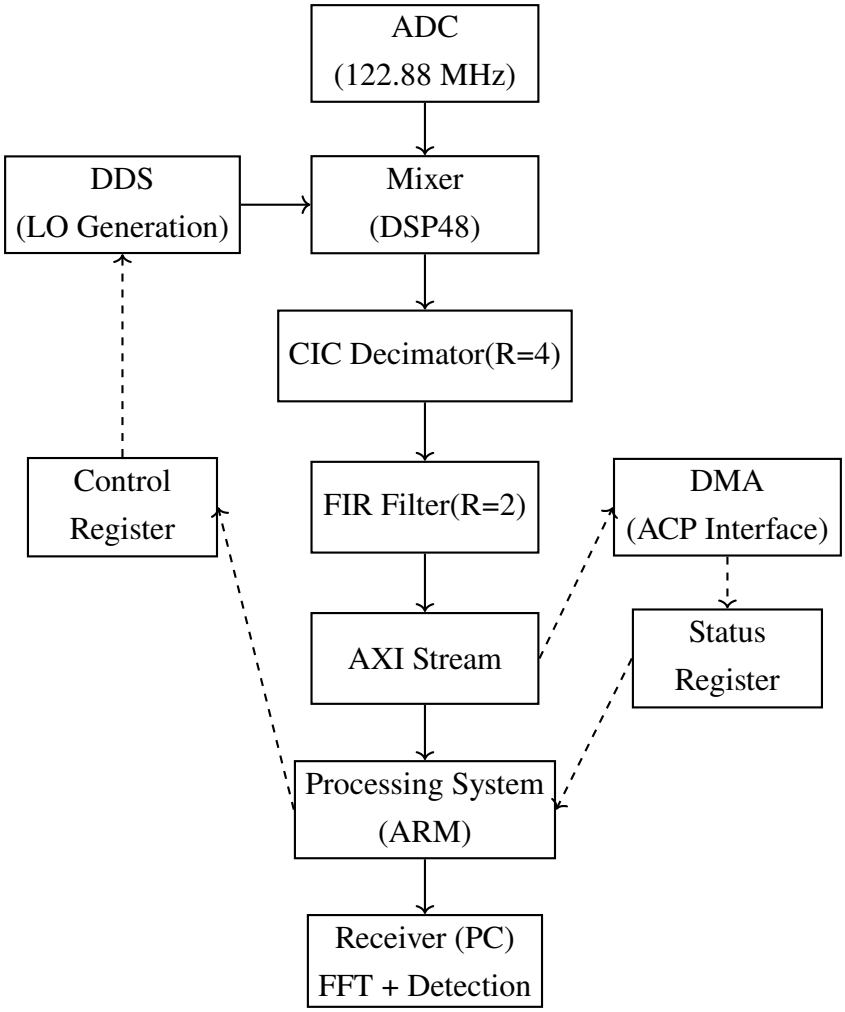


Figure 3.4: Enhanced block diagram of the wideband signal detection system. The processing system controls the DDS through a control register for LO sweeping, while system status and data transfer are managed through a DMA interface and status register via the ACP (Accelerator Coherency Port). The FPGA performs signal acquisition, digital downconversion, and filtering, while the receiver performs FFT-based spectrum analysis and signal detection.

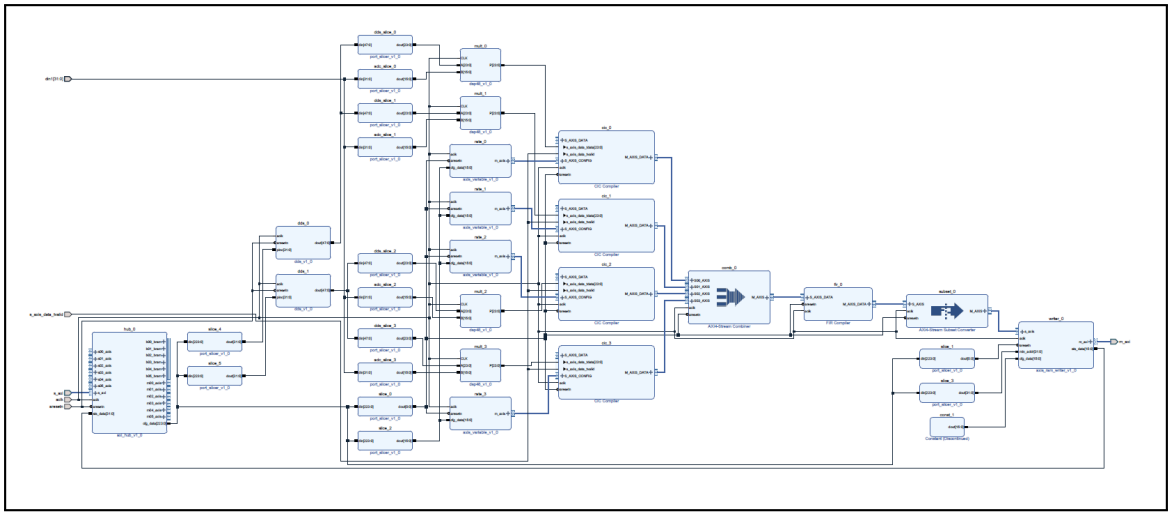


Figure 3.5: Vivado block design of the implemented wideband signal detection system on the Red Pitaya FPGA. The architecture includes ADC interfacing, DDS-based digital downconversion, DSP48-based mixing, CIC decimation, FIR compensation filtering, AXI streaming infrastructure, and DMA-based data transfer to the processing system through the ACP (Accelerator Coherency Port). Control and status registers are used to manage LO sweeping and monitor system operation.

3.5.2 LO Sweep Mechanism

To cover a wide frequency span, a Local Oscillator (LO) sweeping mechanism is implemented using a Direct Digital Synthesizer (DDS). The LO is updated in discrete steps to scan the entire spectrum.

$$f_{\mathrm{LO}}(k) = f_0 + k \cdot \Delta f$$

where:

- $f_0 = 0$ MHz is the starting frequency
- Δf is the frequency step size
- k is the sweep index

The step size is chosen as:

$$\Delta f = [\frac{F_s'}{2}] - 0.768 = [\frac{15.36}{2}] - 0.768 = 6.98 \text{ MHz}$$

This ensures that we observe the positive baseband and additionlly we have overlap 10% overlap between adjacent frequency segments, reducing the risk of signal loss at band boundaries.

The DDS generates the LO using a phase accumulator:

$$\Delta\phi = \frac{f_{\text{LO}}}{F_{\text{clk}}} \cdot 2^{32}$$

where $F_{\text{clk}} = 122.88$ MHz is the system clock. This enables precise and fine-resolution frequency control.

The LO is updated after processing a fixed number of data chunks:

$$T_{\text{hop}} = N_{\text{chunks}} \cdot T_{\text{chunk}}$$

where $N_{\text{chunks}} = 15$, corresponding to approximately one second of data per frequency segment.

3.5.3 FFT-Based Spectrum Analysis

The FPGA streams complex I/Q samples to the receiver via TCP. On the receiver side, spectral analysis is performed.

A Hann window is applied to reduce spectral leakage:

$$x_w[n] = x[n] \cdot w[n]$$

The Fast Fourier Transform (FFT) is computed as:

$$X[k] = \sum_{n=0}^{N-1} x_w[n] e^{-j2\pi kn/N}$$

The frequency resolution is:

$$\Delta f = \frac{F_s'}{N}$$

For $N = 4096$:

$$\Delta f \approx 3.75 \text{ kHz}$$

The magnitude spectrum is computed in decibels:

$$|X[k]|_{\text{dB}} = 20 \log_{10}(|X[k]|)$$

Only the positive frequency bins are retained.

To mitigate LO leakage, DC bins are suppressed by replacing them with neighboring spectral values during processing.

3.5.4 Spectrum Stitching

Each LO step produces a partial spectrum of bandwidth 15.36 MHz. These segments are stitched together to reconstruct the full spectrum up to 61.44 MHz.

The absolute frequency mapping is given by:

$$f_{\text{abs}} = f_{\text{LO}} + k \cdot \Delta f$$

In implementation, each FFT segment is placed into a global spectrum buffer based on the LO index:

$$\text{bin index} = \left\lfloor \frac{f_{\text{LO}}}{\Delta f} \right\rfloor \times \frac{N}{2}$$

Overlapping regions are handled by retaining the maximum power:

$$P_{\text{merged}}(f) = \max\left(P_1(f),\ P_2(f)\right)$$

This ensures smooth transitions across frequency segments.

A waterfall buffer is also maintained for temporal visualization of spectrum evolution.

3.5.5 Signal Detection

Signal detection is performed on the stitched spectrum using magnitude thresholding and Signal-to-Noise Ratio (SNR) estimation.

The noise floor is estimated using the median:

$$P_{\text{noise}} = \text{median}(|X[k]|)$$

The SNR is computed as:

$$\text{SNR}(k) = |X[k]| - P_{\text{noise}}$$

A signal is detected if:

$$\text{SNR}(k) > \gamma$$

where $\gamma \approx 6$ dB.

Detected spectral peaks are grouped to estimate signal center frequencies. These frequencies are fed back to the FPGA, enabling narrowband tuning for detailed analysis.

Thus, the wideband stage continuously monitors the spectrum, identifies active signals, and provides coarse frequency estimates, which are further refined in the narrowband processing stage.

3.6 Narrowband Signal Analysis

The narrowband signal analysis stage performs high-resolution inspection of signals identified during wideband detection. Once a signal is detected, its center frequency is used to configure a dedicated narrowband receiver (RX_0), enabling precise time-domain and spectral analysis.

Compared to the wideband stage, which prioritizes coverage, the narrowband receiver focuses on accuracy and resolution by operating at a significantly reduced bandwidth.

3.6.1 Frequency Tuning and Locking

A Direct Digital Synthesizer (DDS) is used to generate a tunable Local Oscillator (LO) for digital downconversion. The detected frequency is converted into a Frequency Tuning Word (FTW), which controls the DDS phase accumulator:

$$\text{FTW} = \frac{f_{\text{target}}}{F_{\text{clk}}} \cdot 2^N$$

where $F_{\text{clk}} = 122.88$ MHz. This allows precise frequency locking to the target signal.

3.6.2 Digital Downconversion and Decimation

The incoming RF signal is mixed with the DDS-generated LO using DSP48 blocks, translating it to baseband. The signal is then passed through a multi-stage decimation chain:

- CIC decimator (coarse rate reduction)
- FIR filter (passband correction)

With a decimation factor of $R = 160$:

$$F_s' = \frac{F_s}{R_{\text{CIC}} \cdot R_{\text{FIR}}} = \frac{122.88}{40 \times 2} = 1.536 \text{ MHz}$$

resulting in an effective bandwidth of approximately 1.536 kHz.

3.6.3 Narrowband Processing Architecture

The narrowband processing pipeline is implemented across FPGA and receiver as follows:

- **FPGA:** DDS-based tuning, mixing, decimation, FIFO buffering
- **Processing System:** Configuration and TCP streaming
- **Receiver (PC):** IQ visualization and FFT-based spectral analysis

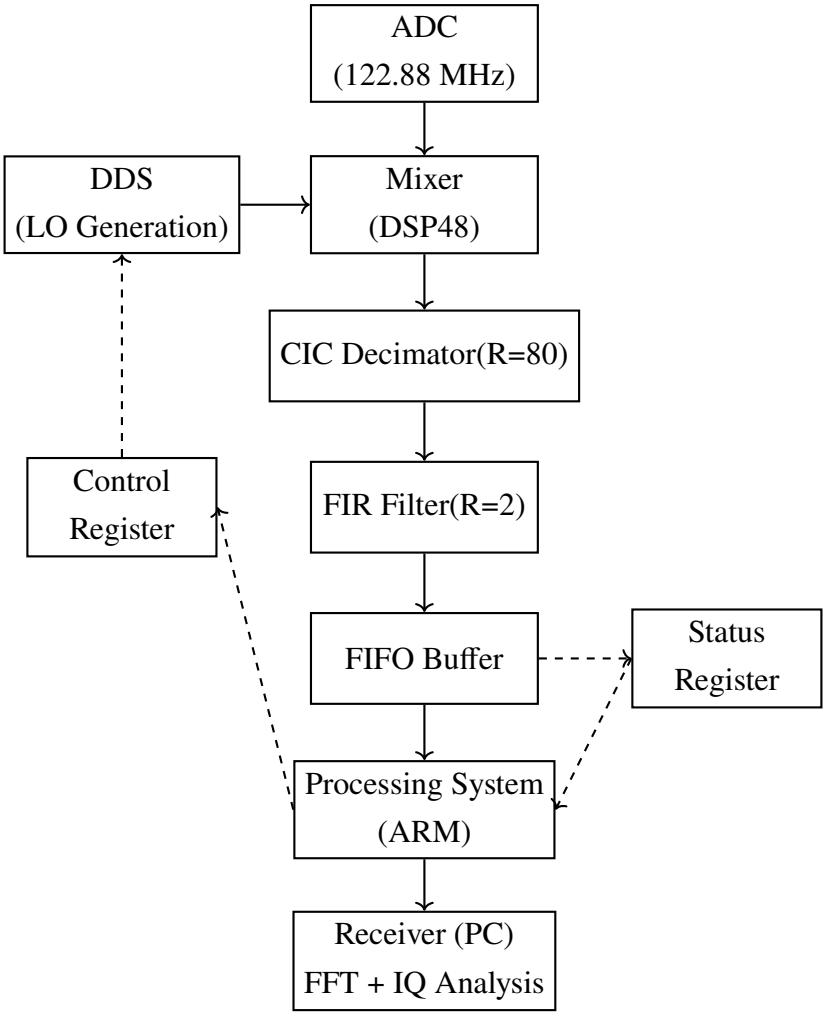


Figure 3.6: Enhanced conceptual block diagram of the narrowband receiver (RX_0). The DDS-based local oscillator is controlled by the processing system through a control register, enabling precise frequency tuning. The downconverted signal is processed through CIC and FIR filtering, buffered in FIFO memory, and streamed to the receiver. System status and buffer information are monitored through a status register, facilitating reliable real-time operation.

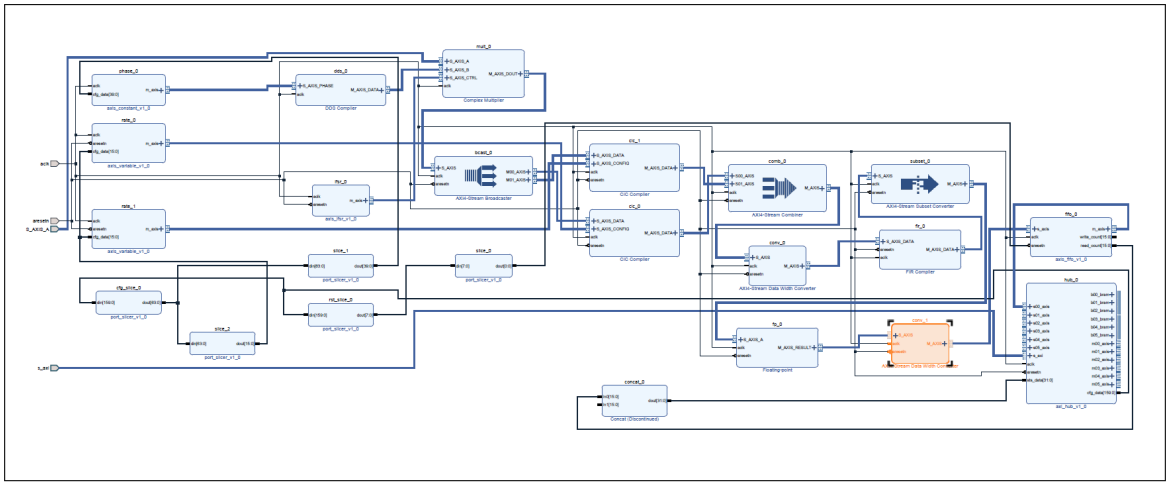


Figure 3.7: Vivado block design of the implemented narrowband receiver on the Red Pitaya FPGA. The architecture includes DDS-based frequency tuning, DSP48-based mixing, CIC decimation, FIR compensation filtering, AXI streaming modules, and FIFO buffering for real-time data transfer to the processing system.

3.6.4 IQ Data Processing

The receiver outputs complex I/Q samples, which are processed on the PC. The complex signal is reconstructed as:

$$x[n] = I[n] + jQ[n]$$

To improve spectral quality, DC offset removal is performed:

$$x'[n] = x[n] - \mu$$

A window function is applied prior to FFT:

$$x_w[n] = x'[n] \cdot w[n]$$

The FFT is computed as:

$$X[k] = \sum_{n=0}^{N-1} x_w[n] e^{-j2\pi kn/N}$$

The magnitude spectrum is:

$$|X[k]|_{\text{dB}} = 20 \log_{10}(|X[k]|)$$

3.6.5 Key Role of Narrowband Stage

The narrowband receiver refines coarse detections from the wideband stage by providing:

- High-resolution spectral analysis
- Accurate signal characterization
- Clean I/Q data for machine learning

Thus, it acts as a critical bridge between signal detection and intelligent classification.

3.7 Dataset Generation and Preprocessing

A custom dataset is generated to train and evaluate machine learning models for RF signal classification. The dataset is designed to emulate realistic wireless environments by incorporating multiple communication protocols operating in the UHF band (390–480 MHz). These protocols are selected to represent diverse modulation schemes, bandwidths, and real-world application domains commonly encountered in practical RF sensing and spectrum monitoring scenarios.

3.7.1 Motivation and UHF Band Selection

The UHF band (390–480 MHz) is chosen due to its widespread usage across both licensed and unlicensed wireless systems. This spectrum range includes public safety communication bands, Private Mobile Radio (PMR) allocations, meteorological services, paging systems, and portions of the Industrial, Scientific, and Medical (ISM) ecosystem (e.g., around 433 MHz). These bands are characterized by:

- High device density and heterogeneous signal coexistence
- Mix of analog and digital communication protocols
- Presence of both mission-critical and consumer-grade systems
- Frequent spectral congestion and interference scenarios

Unlike higher-frequency bands or highly specialized allocations, the UHF region provides a realistic environment where multiple protocols operate simultaneously with overlapping spectral characteristics. This makes it particularly suitable for evaluating RF classification systems and spectrum intelligence frameworks.

To accurately represent this diversity, a set of representative wireless protocols operating within the UHF band is selected and parameterized based on standard specifications and practical deployment characteristics. The key parameters used for dataset generation, including sampling rate, channel bandwidth, guard band, channel spacing, and modulation type, are summarized in Table 3.1.

Protocol	Sampling Rate (Hz)	Bandwidth (Hz)	Guard (Hz)	Spacing (Hz)	Channels	Modulation
PS_NBFM	72,000	12,500	8,000	20,500	400	NBFM
PS_DMR	72,000	12,500	8,000	20,500	400	4-FSK
TETRA	72,000	25,000	3,000	28,000	400	$\pi/4$-DQPSK
WT_NBFM	96,000	12,500	2,500	15,000	160	NBFM
WT_DMR	96,000	12,500	2,500	15,000	160	4-FSK
MET_GFSK	48,000	15,000	5,000	20,000	100	GFSK
SCADA_2FSK	48,000	12,500	2,500	15,000	200	BFSK
RFID_ASK	500,000	200,000	50,000	250,000	10	ASK
RFID_OOK	500,000	200,000	50,000	250,000	10	OOK
POCSAG	48,000	20,000	5,000	25,000	150	BFSK
RC_OOK	500,000	200,000	50,000	250,000	20	OOK

Table 3.1: UHF Band Protocol Parameters Used for Dataset Generation

The parameters listed in Table 3.1 reflect the operational diversity of real-world systems. Narrowband protocols such as public safety radio and SCADA operate with tight bandwidth constraints to maximize spectral efficiency, whereas systems such as RFID and remote control devices utilize wider bandwidths to support rapid transitions and simple hardware implementations. Similarly, the variation in sampling rates is chosen to satisfy the Nyquist criterion while accommodating different modulation complexities.

A key challenge motivating this work is the lack of publicly available, large-scale, labeled RF datasets that capture such heterogeneous protocol behavior. Existing datasets are often limited to isolated modulation schemes or controlled environments. Therefore, this work adopts a standards-driven signal synthesis approach, where signals are generated by studying protocol specifications, regulatory documents, and real-world device behavior, as summarized in the table above.

3.7.2 Protocol Selection, Devices, and Application Context

The dataset models multiple communication systems spanning different operational domains, along with their corresponding real-world transmitting devices:

- **Public Safety (DMR, NBFM):** Handheld and mobile radios used by emergency services (e.g. police radios, fire department transceivers)
- **Walkie-Talkie Systems:** Consumer and professional handheld radios used for short-range communication
- **Meteorological Radiosondes:** Weather balloon transmitters (e.g. Vaisala RS41-type devices) for atmospheric sensing
- **SCADA Telemetry:** Industrial radio modems used in power grids, pipelines, and water systems
- **RFID Systems:** Active RFID tags and readers operating in the 433 MHz ISM band

- **Paging Systems (POCSAG):** Paging transmitters and receivers used in hospitals and industrial facilities
- **Remote Control Devices:** Low-cost RF transmitters such as key fobs, garage door openers, and wireless sensors

The inclusion of device-level context is critical, as each protocol is tightly coupled with its hardware implementation and operational requirements:

- Public safety radios prioritize robustness and interoperability, using narrowband channels and reliable modulation schemes.
- Walkie-talkies are optimized for low-latency voice communication with limited bandwidth.
- Radiosondes operate over long distances and harsh atmospheric conditions, requiring stable and interference-resistant modulation.
- SCADA devices are designed for low-data-rate, high-reliability telemetry in industrial environments.
- RFID systems emphasize low power consumption and simple hardware for large-scale deployment.
- Paging systems are optimized for one-way, high-reliability message delivery.
- Remote control devices prioritize simplicity and low cost, typically using minimal modulation complexity.

This mapping between protocol and device ensures that the generated dataset reflects not only theoretical signal structures but also practical transmission characteristics.

3.7.3 Standards-Based Parameterization

All signals are synthesized using parameters derived from established technical references and protocol specifications in the UHF band, as summarized in Table 3.1.

Each signal class is defined by parameters such as sampling rate, channel bandwidth, guard band, channel spacing, modulation type, and encoding principle. These parameters are selected based on how real devices operate, ensuring that the generated signals accurately reflect practical communication behavior.

The dataset incorporates multiple modulation schemes, including Narrowband FM, 4-FSK, $\pi/4$-DQPSK, Gaussian FSK, Binary FSK, ASK, and OOK, enabling the model to learn features across frequency-, phase-, and amplitude-based signaling.

3.7.4 Sampling Strategy and Channel Modeling

To ensure accurate digital representation, sampling rates are selected according to the Nyquist–Shannon sampling criterion:

$$f_s > 2f_{\max}$$

Sampling rates of 48 kHz, 72 kHz, 96 kHz, and 500 kHz are used depending on signal bandwidth and modulation complexity. Lower sampling rates are sufficient for narrowband telemetry systems, while higher sampling rates are required for wideband and amplitude-based signals.

Channelization is modeled by incorporating both channel bandwidth and guard bands, resulting in:

$$\text{Channel Spacing} = \text{Channel Bandwidth} + \text{Guard Band}$$

This enables realistic modeling of spectral occupancy and adjacent channel interference.

3.7.5 Signal Synthesis

Each RF signal is generated using baseband models corresponding to its modulation scheme and then upconverted to carrier frequencies within the UHF band. Protocol-specific parameters such as symbol rate, frequency deviation, and filtering are chosen to reflect real-world device operation.

For example, DMR signals use 4-FSK modulation with defined symbol rates, paging systems use fixed-deviation binary FSK, and RFID/remote control systems use amplitude-based modulation with rapid ON/OFF transitions.

3.7.6 Preprocessing Pipeline

The generated signals undergo preprocessing before being used for machine learning:

- **Filtering:** Bandpass filtering to isolate desired channels
- **Resampling:** Adjustment of sampling rates for consistency
- **Normalization:** Scaling of signal amplitudes
- **Segmentation:** Division into fixed-length input frames
- **Feature Extraction:** Conversion to time-domain and frequency-domain representations (e.g., FFT)

3.7.7 Summary

The resulting dataset provides a realistic, standards-informed, and device-aware representation of heterogeneous RF signals in the UHF band. By combining protocol-level understanding, hardware context, and signal modeling, the dataset enables robust training and evaluation of RF classification systems in complex and congested spectral environments.

3.7.8 Channel Modeling

To emulate realistic RF propagation and receiver conditions, each synthesized signal is passed through a stochastic channel model incorporating frequency translation effects, additive noise, and spectral variability. The inclusion of these impairments ensures that the generated dataset does not represent idealized signals, but instead reflects the uncertainties and distortions encountered in practical wireless environments.

Frequency Offset and Channelization Each signal is assigned to a randomly selected channel within the protocol-specific channel grid. The effective frequency offset is computed as:

$$f_{\text{offset}} = (ch - \frac{N_{ch}}{2}) \cdot \Delta f + \delta f$$

where ch is the channel index, N_{ch} is the total number of channels, Δf is the channel spacing, and δf is a uniformly distributed random offset within half the channel spacing.

This modeling captures several real-world effects. In practical RF systems, transmitter and receiver oscillators are never perfectly synchronized, leading to carrier frequency mismatch. Additionally, dynamic spectrum allocation and channel hopping introduce uncertainty in signal placement within the band. The inclusion of a random offset within each channel further simulates imperfect tuning and drift, which are common in low-cost hardware and high-mobility scenarios.

This approach also implicitly models adjacent channel interference, as signals may partially overlap in frequency depending on the offset. By exposing the learning model to such variations, it is encouraged to learn robust spectral features rather than relying on fixed carrier locations.

The baseband signal is then frequency-shifted using complex exponential mixing to simulate RF carrier offset, ensuring that the model learns to handle frequency-translated signals rather than assuming a fixed center frequency.

Additive Noise (AWGN) To simulate varying signal quality conditions, complex Additive White Gaussian Noise (AWGN) is introduced:

- Signal-to-Noise Ratio (SNR) range: -20 dB to $+25$ dB
- Discrete SNR steps: $\{-20, -15, -10, -5, 0, 5, 10, 15, 20, 25\}$ dB

In real-world wireless systems, signal quality varies significantly due to path loss, interference, multipath fading, and hardware limitations. By spanning a wide SNR range, the dataset captures both extreme low-SNR conditions (where signals are barely distinguishable from noise) and high-SNR conditions (where signals are clean and well-defined).

Noise power is computed relative to the instantaneous signal power, ensuring consistent SNR representation across different modulation types and bandwidths. This is particularly important because wideband and narrowband signals respond differently to noise. The inclusion of low-SNR samples forces the model to learn noise-robust features, preventing overfitting to ideal signal conditions and improving generalization to real-world deployments.

Bandwidth and Spectral Variability Bandwidth variations are inherently introduced through protocol-specific configurations and signal generation processes. Each protocol operates with distinct channel bandwidths and filtering characteristics, which are enforced during signal synthesis using FIR-based band-limiting filters.

This step ensures that the generated signals exhibit realistic spectral confinement and roll-off behavior. In practice, different communication systems occupy different portions of the spectrum depending on their application, data rate, and regulatory constraints. For example, narrowband voice systems occupy significantly less bandwidth than wideband RFID or telemetry signals.

By incorporating these variations, the dataset captures differences in spectral occupancy, sidelobe behavior, and transition smoothness. This allows the model to distinguish between protocols based not only on modulation type but also on their spectral footprint, which is a key identifying characteristic in RF signal intelligence.

Randomized Temporal Windowing To avoid deterministic feature learning and improve generalization, each IQ signal is randomly segmented before feature extraction. A random window of fixed length (2048 samples) is selected from the full signal duration.

In real-world signal acquisition, receivers rarely capture perfectly aligned or complete transmissions. Instead, signals are observed as arbitrary fragments in time, often with unknown starting points and varying phase alignment. Random windowing simulates this condition by ensuring that the model is exposed to different portions of the signal waveform.

This prevents the model from memorizing specific temporal patterns or fixed symbol boundaries and instead encourages learning of invariant features such as modulation structure, statistical properties, and spectral characteristics.

3.8 Dataset Generation and Feature Extraction

This section describes the methodology used for generating the RF signal dataset and extracting features for machine learning. The dataset is synthetically generated to emulate realistic RF environments, incorporating multiple protocols, modulation schemes, channel conditions, and noise levels. The objective is to create a diverse, balanced, and scalable dataset suitable for training deep learning models.

3.8.1 Signal Generation

Each sample begins with the generation of a baseband message signal that mimics real-world communication signals. Two primary components are considered:

- **Voice-like signals:** Generated using multiple sinusoidal components in the range of 300–3400 Hz
- **Music-like signals:** Generated using a broader frequency range (100–15000 Hz)

A random combination of these signals is used to introduce variability:

$$m(t) = \alpha v(t) + (1 - \alpha)u(t)$$

where $v(t)$ represents voice, $u(t)$ represents music, and $\alpha \in [0, 1]$ is randomly selected.

The generated signal is normalized and smoothed to emulate realistic transmission characteristics.

3.8.2 Modulation Schemes

The baseband message is modulated using different modulation techniques depending on the protocol configuration:

- Narrowband Frequency Modulation (NBFM)
- Frequency Shift Keying (FSK, BFSK, 4FSK)
- Gaussian Frequency Shift Keying (GFSK)
- Differential Quadrature Phase Shift Keying (DQPSK)
- Amplitude Shift Keying (ASK)
- On-Off Keying (OOK, RFID, RC)

For example, FM modulation is expressed as:

$$x(t) = \exp\left(j2\pi \int m(t)\, dt\right)$$

while FSK-based schemes use discrete frequency tones corresponding to binary symbols.

3.8.3 Channel Modeling

To simulate realistic RF propagation conditions, the generated signal is passed through a channel model incorporating frequency offset and noise.

Frequency Offset

$$x(t) \leftarrow x(t)\, e^{j2\pi f_{\text{offset}} t}$$

Additive White Gaussian Noise (AWGN)

$$y(t) = x(t) + n(t)$$

The noise level is controlled using the Signal-to-Noise Ratio (SNR):

$$\text{SNR} = 10 \log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{noise}}}\right)$$

SNR values range from -20 dB to 25 dB, ensuring robustness under varying noise conditions.

3.8.4 Feature Extraction

Each complex IQ signal is segmented into fixed-length windows of:

$$N = 2048$$

The signal is normalized as:

$$x[n] = \frac{x[n]}{\max(|x[n]|) + \epsilon}$$

Time-Domain Features

- In-phase component: $I[n] = \Re\{x[n]\}$
- Quadrature component: $Q[n] = \Im\{x[n]\}$
- Magnitude: $|x[n]|$
- Phase: $\angle x[n]$

These features preserve amplitude and phase information essential for distinguishing modulation schemes.

Frequency-Domain Features

A Fast Fourier Transform (FFT) is applied:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

The FFT components are normalized:

$$\text{FFT}_I = \frac{\Re\{X[k]\}}{\max(|X[k]|)}, \quad \text{FFT}_Q = \frac{\Im\{X[k]\}}{\max(|X[k]|)}$$

Feature Representation

The final feature vector is constructed as:

$$\mathbf{F} = [I, Q, \text{FFT}_I, \text{FFT}_Q, |x|, \angle x]$$

This multi-domain representation captures both temporal and spectral characteristics of RF signals, enabling effective classification.

3.8.5 Dataset Structure

The dataset is generated in a balanced manner across multiple protocols, channels, and SNR conditions. Each protocol contributes approximately 4×10^4 samples, resulting in a total of about 4.4×10^5 samples.

Each sample is stored as a CSV file with associated metadata, including protocol label, SNR (dB), frequency offset, RF carrier frequency, and channel bandwidth.

This structured design ensures diversity in modulation types and channel conditions, leading to improved model generalization.

3.8.6 Dataset Generation Flowchart

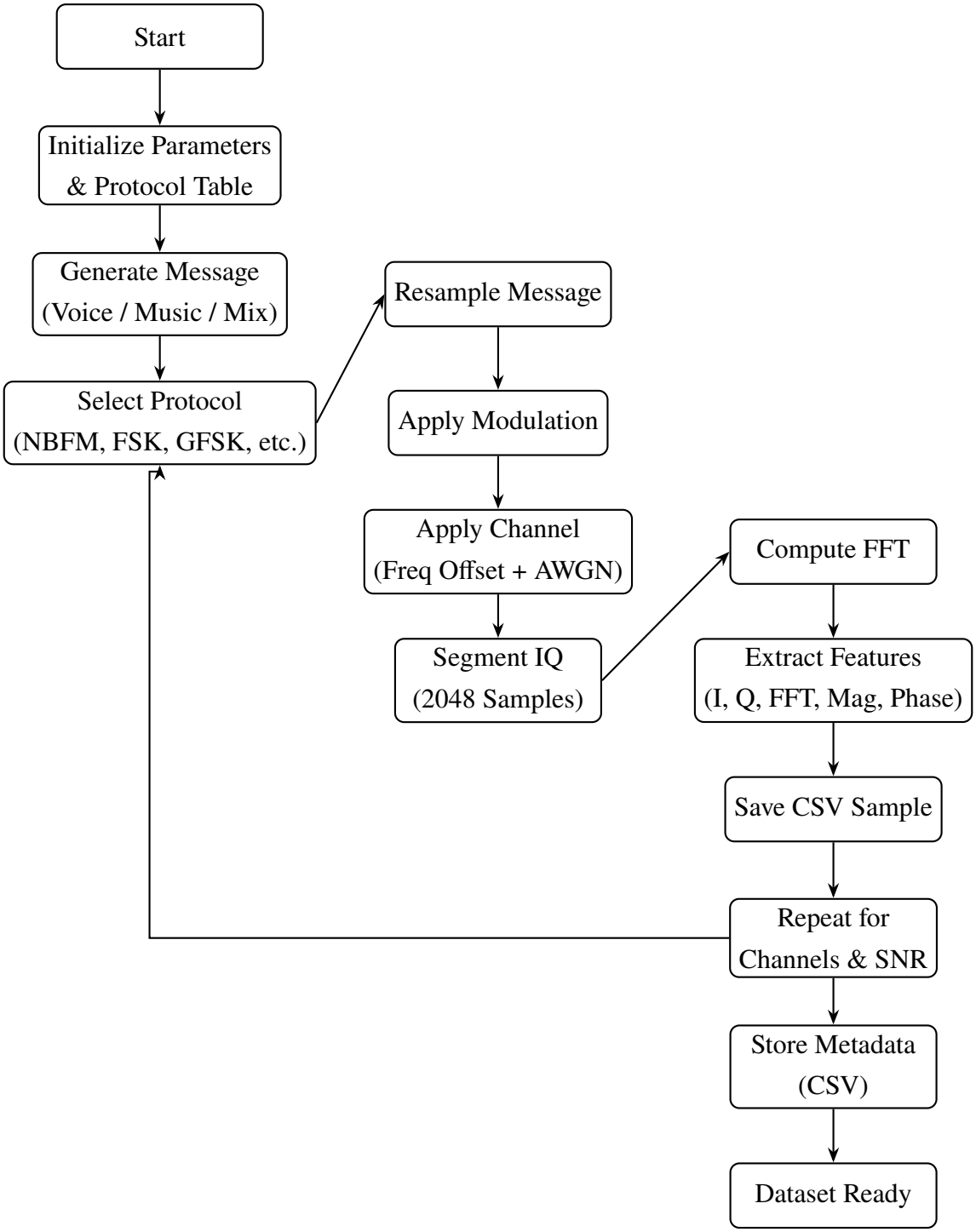


Figure 3.8: Flowchart of the RF dataset generation pipeline. The process begins with parameter initialization and message generation, followed by protocol-specific modulation, channel modeling (frequency offset and noise), and segmentation. Features are extracted from both time and frequency domains and stored as CSV samples. The process iterates over multiple channels and SNR conditions to generate a balanced dataset.

3.9 Machine Learning Model Development

Multiple models are developed and evaluated:

3.9.1 Artificial Neural Network (ANN)

A baseline Artificial Neural Network (ANN) was implemented to evaluate the effectiveness of fully connected architectures for RF signal classification. Unlike convolutional models, the ANN does not explicitly exploit temporal or spatial structure, relying instead on learned global feature representations.

Each input sample is represented as a 2048×6 tensor consisting of I, Q, magnitude, phase, and FFT components. The input is permuted to $(6, 2048)$, allowing each channel to be processed independently in a channel-aware manner.

Each channel is passed through a shared fully connected transformation:

$$\mathbf{z}_c = \sigma\left(\mathrm{BN}(\mathbf{W}\mathbf{x}_c + \mathbf{b})\right), \quad \mathbf{z}_c \in \mathbb{R}^{512}$$

The resulting feature vectors are concatenated:

$$\mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_6] \in \mathbb{R}^{3072}$$

This fused representation is processed through additional fully connected layers:

$$\mathbf{h}_1 = \sigma(\mathbf{W}_1\mathbf{z}), \quad \mathbf{h}_2 = \sigma(\mathbf{W}_2\mathbf{h}_1)$$

resulting in a compact latent representation $\mathbf{h} \in \mathbb{R}^{64}$.

Multi-Task Learning Framework

The original ANN model was trained using a multi-task objective with three heads:

- Protocol classification
- Modulation classification
- Regression (frequency and bandwidth)

$$\mathcal{L} = \lambda_1\mathcal{L}_{\text{proto}} + \lambda_2\mathcal{L}_{\text{mod}} + \lambda_3\mathcal{L}_{\text{reg}}$$

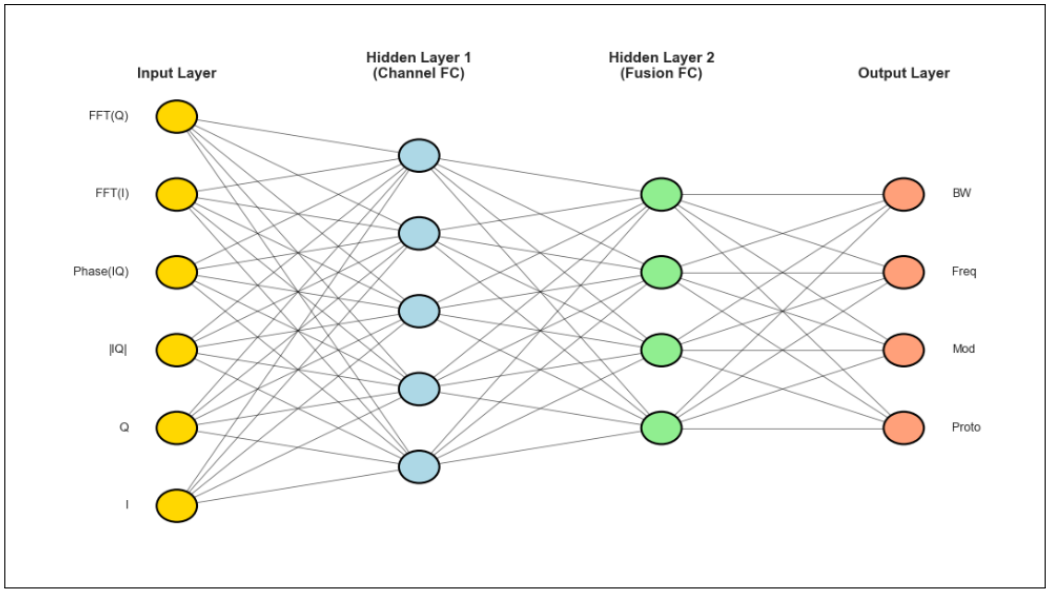


Figure 3.9: Channel-aware ANN architecture for RF signal classification.

Modulation Classification (ANN)

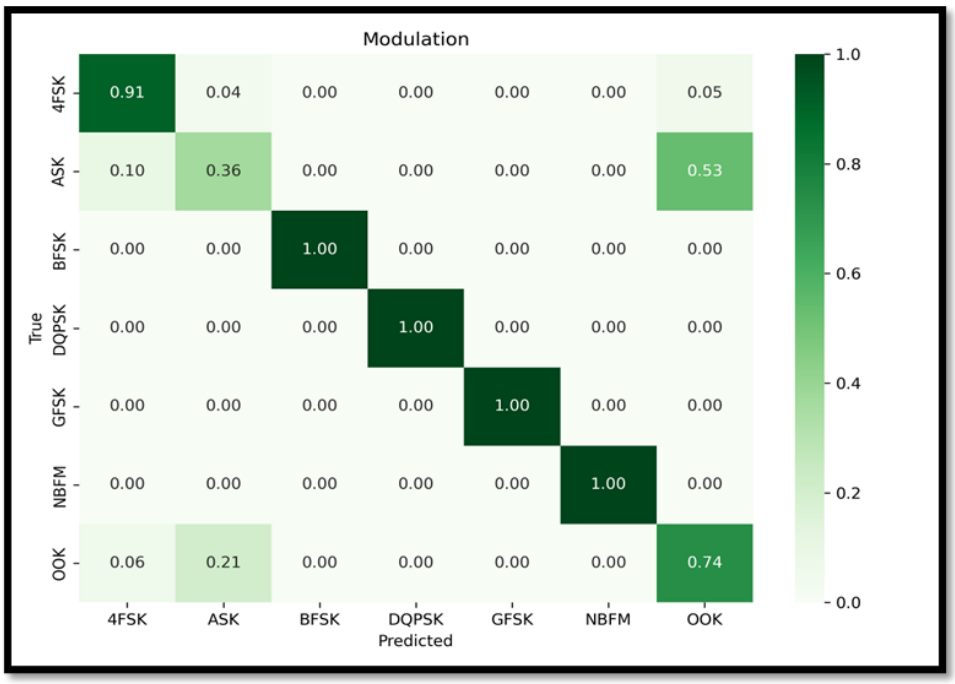


Figure 3.10: Confusion matrix for ANN-based modulation classification.

The confusion matrix illustrates the performance of the ANN model in classifying different modulation schemes. Frequency- and phase-based modulations such as BFSK, DQPSK, GFSK, and NBFM achieve near-perfect classification accuracy, as indicated by strong diagonal values close to 1.0.

However, significant confusion is observed among amplitude-based modulation schemes.

ASK is frequently misclassified as OOK (approximately 53%), while only about 36% of ASK samples are correctly classified. Similarly, OOK exhibits partial confusion with ASK (around 21%) and minor misclassification into 4FSK.

In addition, a small amount of confusion is observed in 4FSK classification, where some samples are incorrectly predicted as ASK or OOK. These errors indicate that the ANN struggles to distinguish modulation schemes with similar amplitude characteristics.

Overall, while the ANN is effective in learning global feature relationships, it lacks the ability to capture localized temporal and spectral patterns, resulting in reduced performance for closely related modulation types.

Performance Across SNR

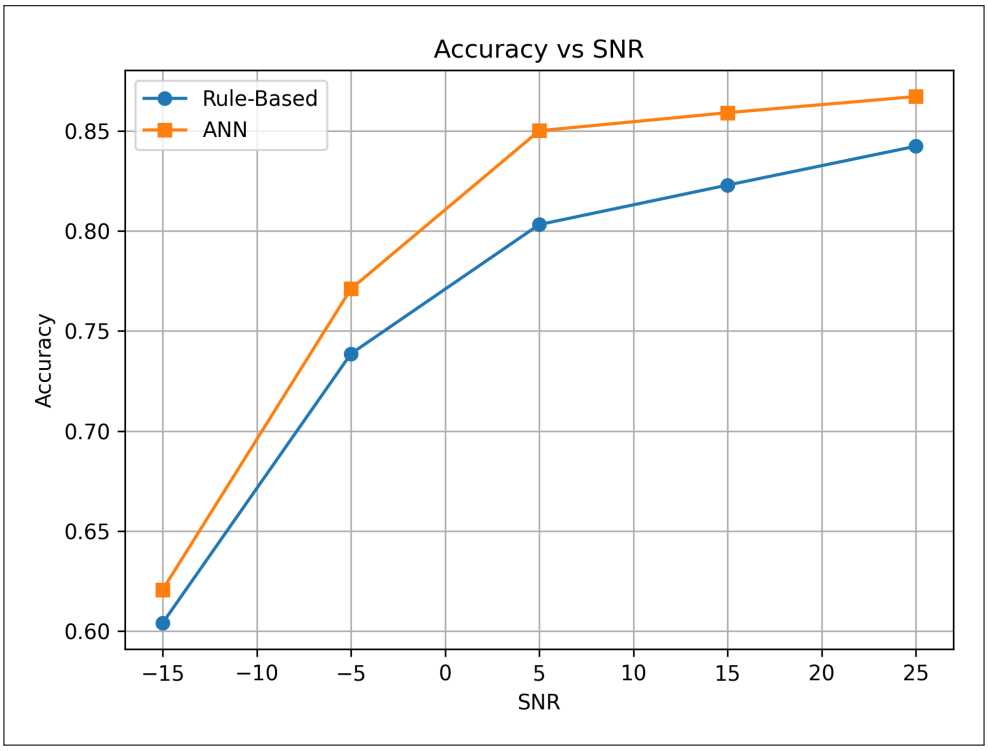


Figure 3.11: Accuracy comparison of ANN and rule-based inference across different SNR levels.

The ANN model shows improvement with increasing SNR; however, its performance remains consistently lower than convolutional models. At low SNR (-15 dB), the model struggles to extract meaningful features, resulting in reduced accuracy.

As SNR increases, the accuracy improves steadily, but the gain saturates earlier compared to CNN-based models. This indicates limited robustness to noise and reduced generalization capability.

3.9.2 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) was implemented to exploit local temporal dependencies and structured feature relationships present in RF IQ signals. Unlike the ANN baseline, the CNN directly operates on sequential data, enabling hierarchical feature extraction.

Each input sample is represented as a 2048×6 tensor consisting of I, Q, magnitude, phase, and FFT components. The input is permuted to $(6, 2048)$ to match the channel-first format required for 1D convolution.

The network begins with an initial convolutional block:

$$\mathbf{X}_1 = \text{MaxPool}(\sigma(\text{BN}(\text{Conv1D}(\mathbf{X}))))$$

To enable deeper feature learning, residual blocks are used:

$$\mathbf{Y} = \sigma(\text{BN}(\text{Conv1D}(\mathbf{X})))\mathbf{Z} = \text{BN}(\text{Conv1D}(\mathbf{Y}))\mathbf{H} = \sigma(\mathbf{Z} + \mathbf{X}_{\text{skip}})$$

These residual connections improve gradient flow and allow the network to learn deeper representations. The architecture progressively increases feature depth while reducing temporal resolution.

Global temporal features are aggregated using:

$$\mathbf{z} = \text{GlobalAvgPool}(\mathbf{H})$$

This vector is passed through fully connected layers to obtain a latent representation $\mathbf{h} \in \mathbb{R}^{64}$.

Multi-Task Learning Framework

The original CNN model was trained using a multi-task objective with three heads:

- Protocol classification
- Modulation classification
- Regression (frequency and bandwidth)

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{proto}} + \lambda_2 \mathcal{L}_{\text{mod}} + \lambda_3 \mathcal{L}_{\text{reg}}$$

For deployment on the ZCU104 DPU, the model was simplified to retain only modulation classification. Protocol identification is performed using rule-based inference based on predicted modulation, bandwidth, and frequency.

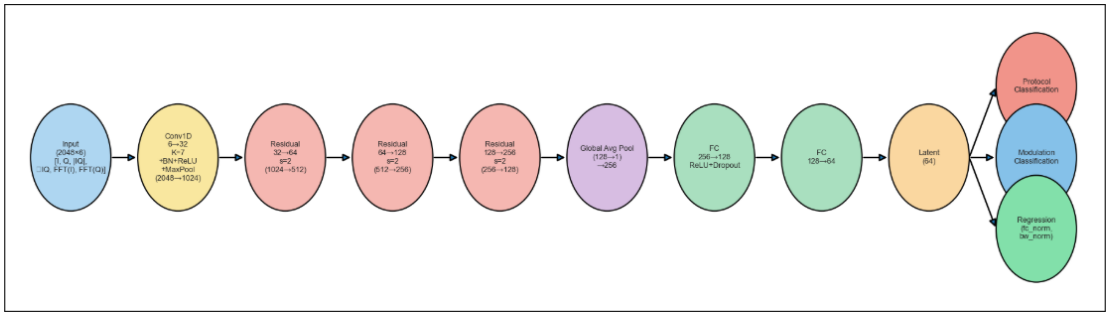


Figure 3.12: Residual CNN architecture for RF signal classification.

Modulation Classification (CNN)



Figure 3.13: Confusion matrix for CNN-based modulation classification.

The confusion matrix demonstrates that the CNN model achieves strong performance in modulation classification. Frequency- and phase-based modulation schemes such as BFSK, DQPSK, GFSK, and NBFM achieve near-perfect classification accuracy, with values close to 1.0 along the diagonal.

Compared to the ANN model, the CNN significantly improves separability among modulation types. In particular, amplitude-based modulation classification shows notable improvement. OOK achieves high accuracy (approximately 93%), indicating that the model effectively captures its temporal and spectral characteristics.

However, some confusion persists for ASK, where a large portion of samples (approximately 91%) are misclassified as OOK, and only a small fraction is correctly identified. This behavior arises due to the similarity in amplitude envelope characteristics between ASK and OOK.

Minor misclassification is also observed in 4FSK, where a small percentage of samples are incorrectly predicted as OOK. Despite these overlaps, the CNN demonstrates a substantial improvement over fully connected architectures by effectively capturing localized time-frequency patterns.

Overall, the CNN provides more discriminative feature representations, resulting in higher modulation classification accuracy and improved robustness across different signal conditions.

Performance Across SNR

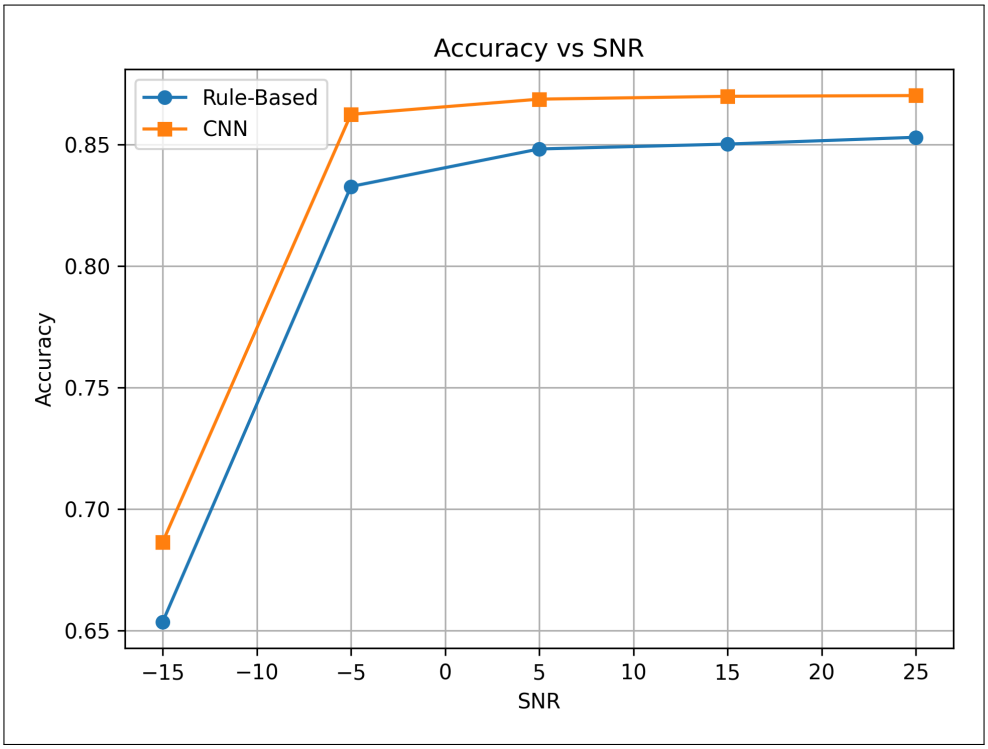


Figure 3.14: Accuracy comparison between CNN and rule-based inference across different SNR levels.

The CNN model demonstrates strong robustness across varying SNR conditions. At low SNR (-15 dB), the CNN significantly outperforms rule-based detection, indicating better noise resilience due to learned feature representations.

As SNR increases, both approaches improve; however, the CNN consistently maintains higher accuracy across all SNR levels. The performance saturates beyond 10 dB, indicating stable detection capability under favorable signal conditions.

3.9.3 Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM)

A hybrid Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) model was implemented to capture both local feature structures and temporal dependencies present in RF

IQ signals. While the CNN extracts hierarchical spatial features, the LSTM models sequential relationships across the signal.

Each input sample is represented as a 2048×6 tensor consisting of I, Q, magnitude, phase, and FFT components. The input is permuted to $(6, 2048)$ to match the channel-first format required for convolutional layers.

The network begins with a convolutional feature extraction stage:

$$\mathbf{X}_1 = \text{MaxPool}(\sigma(\text{BN}(\text{Conv1D}(\mathbf{X}))))$$

Residual convolutional blocks are then applied:

$$\mathbf{Y} = \sigma(\text{BN}(\text{Conv1D}(\mathbf{X})))\mathbf{Z} = \text{BN}(\text{Conv1D}(\mathbf{Y}))\mathbf{H} = \sigma(\mathbf{Z} + \mathbf{X}_{\text{skip}})$$

These blocks improve feature extraction while preserving gradient flow.

Temporal Modeling using LSTM

The convolutional output $\mathbf{H}$ is reshaped from (B, C, T) to (B, T, C) and passed through a bidirectional LSTM:

$$\mathbf{h}_t = \text{BiLSTM}(\mathbf{H})$$

The LSTM captures temporal dependencies such as symbol transitions and modulation patterns. Global temporal features are aggregated using mean pooling:

$$\mathbf{z} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{h}_t$$

This representation is passed through fully connected layers to obtain a compact latent vector $\mathbf{h} \in \mathbb{R}^{64}$.

Multi-Task Learning Framework

The CNN-LSTM model was initially trained using a multi-task objective:

- Protocol classification
- Modulation classification
- Regression (frequency and bandwidth)

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{proto}} + \lambda_2 \mathcal{L}_{\text{mod}} + \lambda_3 \mathcal{L}_{\text{reg}}$$

For deployment on the ZCU104 DPU, the model was simplified to retain only modulation classification. Protocol identification is performed using rule-based inference based on predicted modulation and signal parameters.

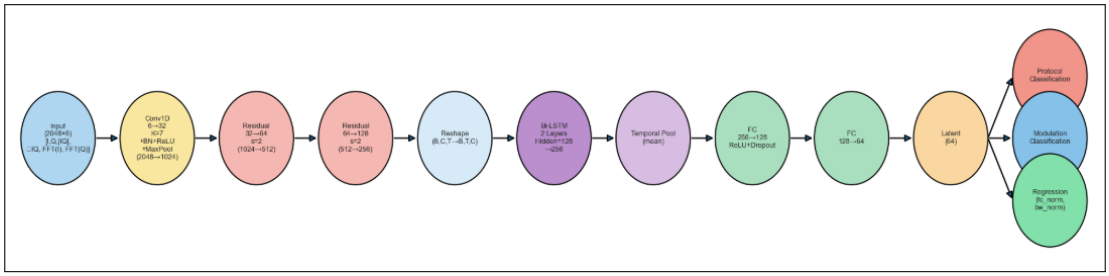


Figure 3.15: CNN-LSTM architecture combining convolutional feature extraction with bidirectional LSTM-based temporal modeling.

Modulation Classification (CNN–LSTM)

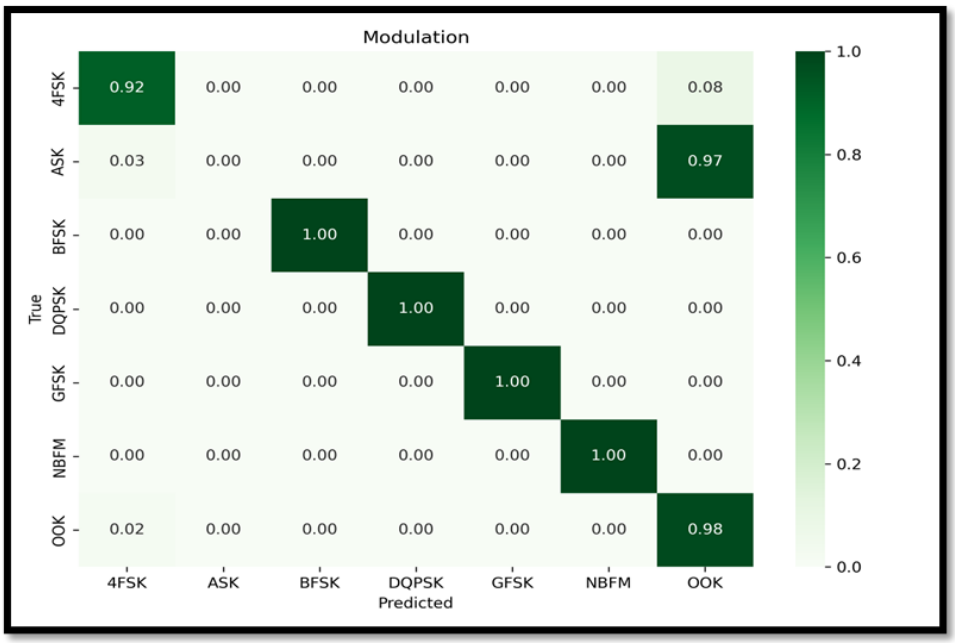


Figure 3.16: Confusion matrix for CNN–LSTM-based modulation classification.

The confusion matrix demonstrates that the CNN–LSTM model achieves the highest modulation classification performance among the evaluated architectures. Frequency- and phase-based modulation schemes such as BFSK, DQPSK, GFSK, and NBFM achieve perfect classification accuracy, with values of 1.0 along the diagonal.

The hybrid architecture significantly improves the classification of amplitude-based modulations. OOK achieves near-perfect accuracy (approximately 98%), while the confusion between ASK and OOK is further reduced compared to both ANN and CNN models. However, ASK is still predominantly misclassified as OOK (around 97%), indicating that distinguishing these

two modulation types remains inherently challenging due to their similar amplitude envelope characteristics.

Minor misclassification is observed in 4FSK, where a small portion of samples is incorrectly predicted as OOK. Despite this, the overall performance remains highly robust.

The combination of convolutional feature extraction and temporal modeling enables the CNN–LSTM to capture both local spectral patterns and sequential dependencies in RF signals. This results in improved feature discrimination and enhanced classification accuracy across all modulation types.

Overall, the CNN–LSTM model provides the most effective representation for RF modulation classification, offering superior robustness and generalization compared to ANN and CNN architectures.

Performance Across SNR

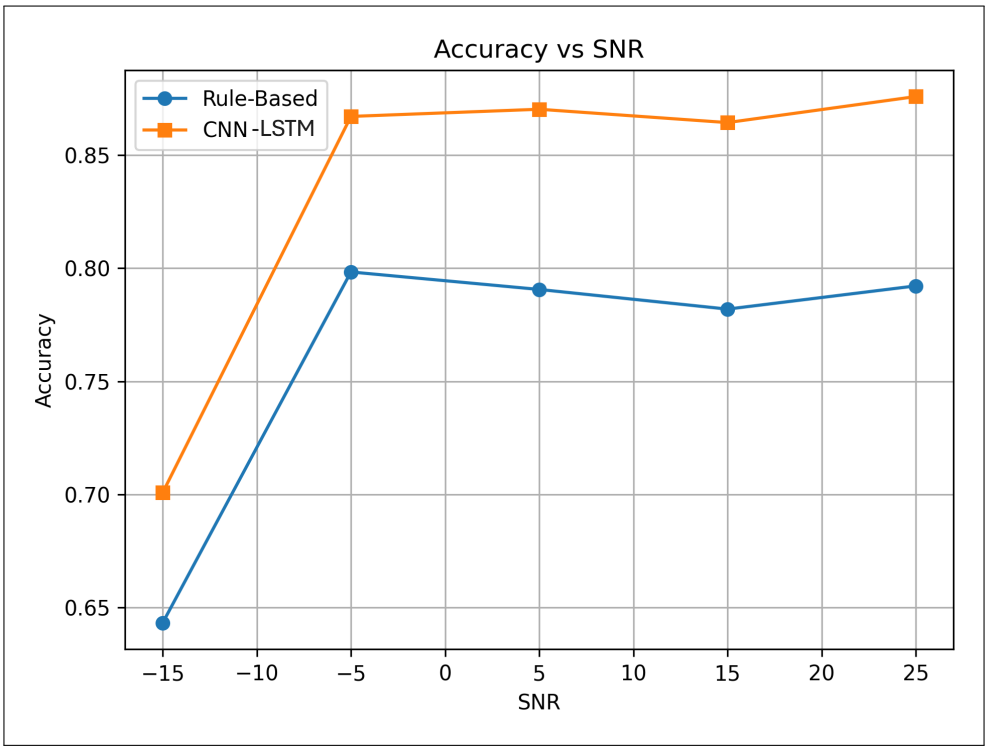


Figure 3.17: Accuracy comparison of CNN-LSTM and rule-based inference across different SNR levels.

The CNN-LSTM demonstrates improved robustness across varying SNR conditions. At low SNR (-15 dB), it significantly outperforms rule-based inference, indicating better noise resilience due to temporal modeling.

As SNR increases, the accuracy stabilizes and remains consistently higher than rule-based inference. However, the improvement over CNN is marginal, especially at high SNR levels.

3.9.4 Model Comparison and Performance Evaluation

The three architectures—ANN, CNN, and CNN-LSTM—were trained using supervised learning and evaluated on a held-out test dataset. Performance was assessed using standard classification metrics, including accuracy, precision, recall, and F1-score. Additionally, regression performance for center frequency and bandwidth estimation was evaluated using RMSE, MAE, and R^2 score.

It is important to note that protocol performance reported in this section is obtained using rule-based inference applied to the predicted modulation and signal parameters, rather than direct protocol classification.

Quantitative Comparison

Model	Rule-Based Protocol Accuracy	Modulation Accuracy	F1-score	Precision	RMSE	R^2
ANN	0.7396	0.8774	0.7950	0.7980	0.1304	0.8126
CNN	0.8080	0.8856	0.8120	0.8281	0.1070	0.8650
CNN-LSTM	0.7616	**0.8903**	**0.8184**	**0.8422**	**0.1026**	**0.8715**

Table 3.2: Performance comparison of ANN, CNN, and CNN-LSTM models

Analysis of Results

The ANN model serves as a baseline and achieves moderate performance. However, it exhibits higher regression error (RMSE = 0.1304) and reduced generalization capability due to its inability to capture localized temporal and spectral structures in RF signals.

The CNN model significantly improves performance across all metrics. By leveraging convolutional filters and residual connections, it effectively captures local correlations in IQ data, leading to improved modulation classification and more accurate parameter estimation. This results in a higher rule-based protocol accuracy (80.80%) and reduced regression error.

The CNN-LSTM model achieves the best overall performance in terms of modulation classification and regression metrics. By incorporating temporal modeling through bidirectional LSTM layers, it captures long-range dependencies in the signal. However, despite improved feature representation, its rule-based protocol accuracy (76.16%) is lower than CNN.

Rule-Based Inference Performance

Protocol classification in the deployed system is performed using a rule-based mapping:

$$\text{Signal} \rightarrow \text{Modulation + Parameters} \rightarrow \text{Protocol}$$

While CNN-LSTM provides better feature extraction, small deviations in predicted frequency and bandwidth negatively impact rule-based inference. As a result, CNN outperforms CNN-LSTM in final protocol identification despite slightly lower classification accuracy.

Training Behavior and Generalization

The ANN model shows signs of overfitting, with training accuracy exceeding 93% while validation accuracy remains significantly lower. In contrast, CNN and CNN-LSTM exhibit stable training behavior due to the use of batch normalization, dropout, and residual connections.

The CNN-LSTM demonstrates smoother convergence and improved generalization, benefiting from temporal modeling and regularization techniques.

Overall Observations

- ANN provides a simple baseline but lacks the ability to model complex RF signal structures.
- CNN achieves a strong balance between accuracy, robustness, and computational efficiency.
- CNN-LSTM offers the best feature representation but introduces higher computational complexity.

Overall, while CNN-LSTM achieves the highest classification performance, the CNN model provides the best trade-off between accuracy, robustness, and hardware efficiency. Therefore, CNN is selected for deployment on the ZCU104 platform for real-time RF signal detection and classification.

3.10 FPGA Implementation

The proposed RF signal processing system is implemented on FPGA platforms such as Red Pitaya and ZCU104, leveraging both programmable logic (PL) and processing system (PS) resources for efficient computation and data handling.

3.10.1 System Architecture

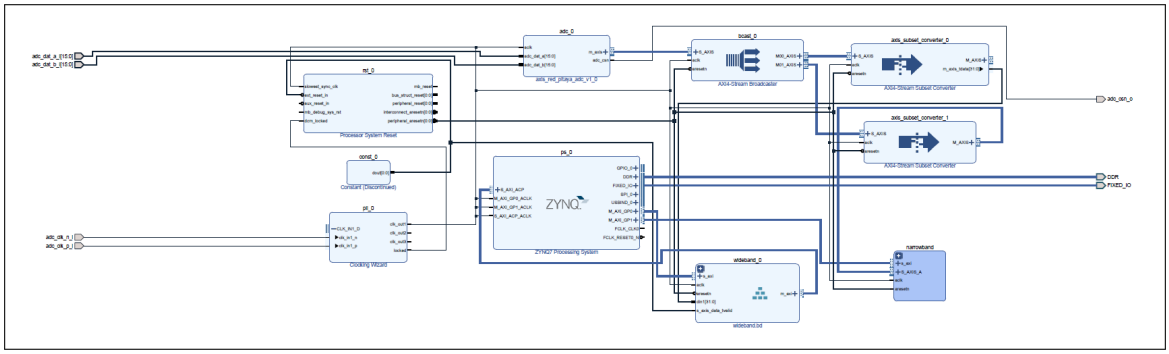


Figure 3.18: FPGA block diagram showing wideband and narrowband processing chains

The design consists of two parallel signal processing paths:

- **Wideband Processing Chain:** Designed for high data-rate signals (15.35 MHz), where data is streamed directly to DDR memory through the ACP (Accelerator Coherency Port).
- **Narrowband Processing Chain:** Designed for low data-rate signals (in ksps range), where FIFO buffering is used for efficient handling of bursty or slow data streams.

Both chains utilize AXI-Stream interfaces for high-throughput data transfer and are integrated with FFT IP cores for spectral analysis and DDS modules for frequency tuning.

3.10.2 Memory Mapping and Addressing

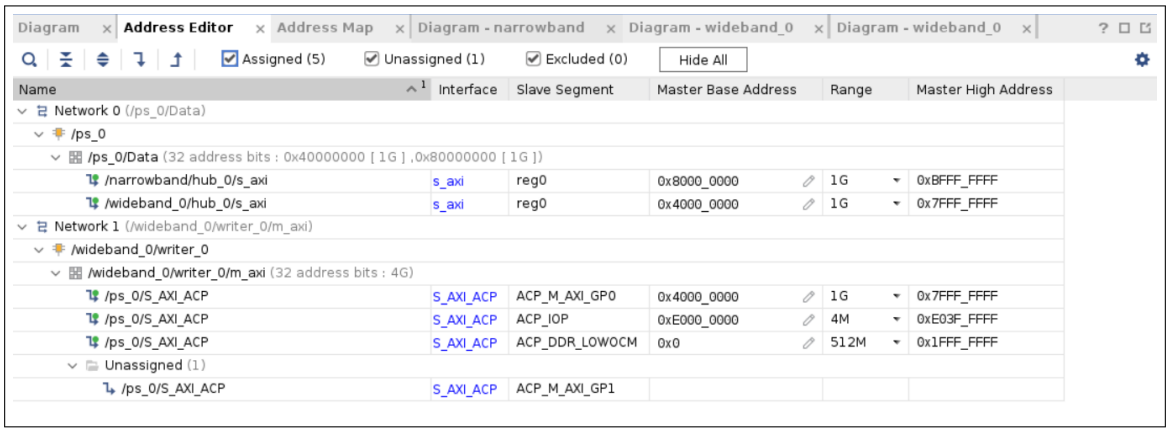


Figure 3.19: AXI address mapping for wideband and narrowband modules

The memory architecture is designed to optimize data throughput and latency:

- **Wideband Module:** Mapped to base address `0x4000_0000`. This module uses the ACP interface to directly access DDR memory, enabling high-speed data transfer required for 15.35 MHz sampling rates.
- **Narrowband Module:** Mapped to base address `0x8000_0000`. Since the data rate is significantly lower, FIFO buffers are employed to efficiently manage data flow and prevent underflow/overflow conditions.

The ACP-based architecture allows cache-coherent memory access, reducing latency and improving throughput for wideband signal processing.

3.10.3 Resource Utilization

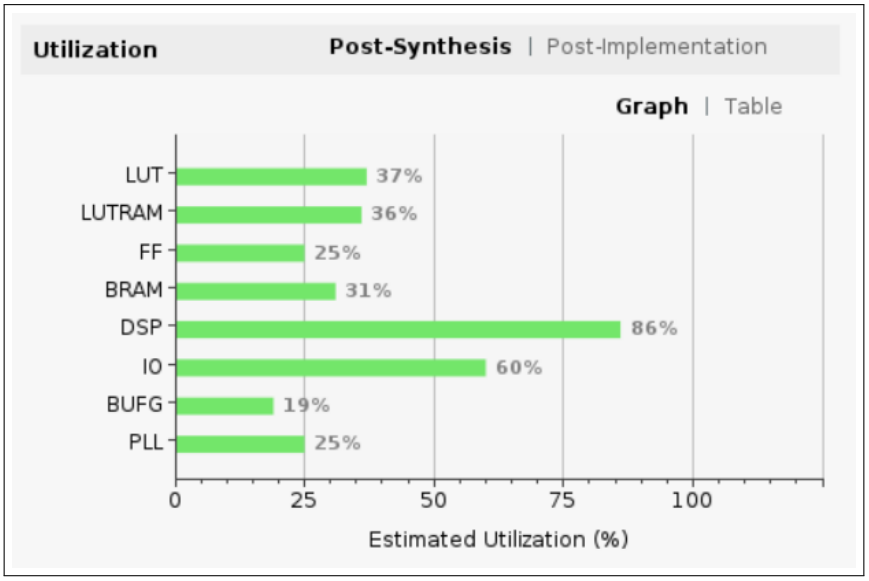


Figure 3.20: Post-synthesis FPGA resource utilization

Efficient utilization of FPGA resources is critical for high-performance signal processing. The implementation demonstrates optimized usage of DSP resources:

- Total available DSP slices: **220**
- DSP slices utilized: **189** ($\approx 86\%$)

The high DSP utilization indicates that the design effectively leverages dedicated hardware multipliers for computationally intensive operations such as filtering, FFT computation, and modulation processing.

Other resource usage includes LUTs, BRAM, and flip-flops, which remain within acceptable limits, ensuring that the design is both efficient and scalable.

3.10.4 System-Level Integration

The FPGA (PL) is responsible for:

- High-speed digital signal processing (Decimation and filtering)
- Streaming data acquisition and preprocessing

The Processing System (PS) handles:

- Control and configuration of FPGA modules
- Machine learning inference for signal classification

- Data management and communication

This hardware-software co-design ensures efficient handling of both high-throughput signal processing and intelligent decision-making tasks.

3.11 Summary

This chapter presented the complete system design integrating SDR-based acquisition, FPGA-based DSP, and machine learning-based classification. The system successfully performs wideband scanning, signal detection, and narrowband analysis in real time, forming a scalable foundation for intelligent RF sensing and classification systems.

Chapter 4

Results and Discussion

4.1 Overview

This chapter presents the performance evaluation of the proposed wideband–narrowband RF signal detection system. The results cover wideband spectrum scanning, narrowband signal analysis, real-time communication performance, and system-level trade-offs. The effectiveness of the FPGA–receiver co-design is demonstrated through real-time visualization and reliable signal detection.

4.2 Wideband Detection Performance

The wideband detection stage was evaluated based on its ability to scan, reconstruct, and detect signals across a large RF bandwidth using FPGA-based LO sweeping and receiver-side FFT processing.

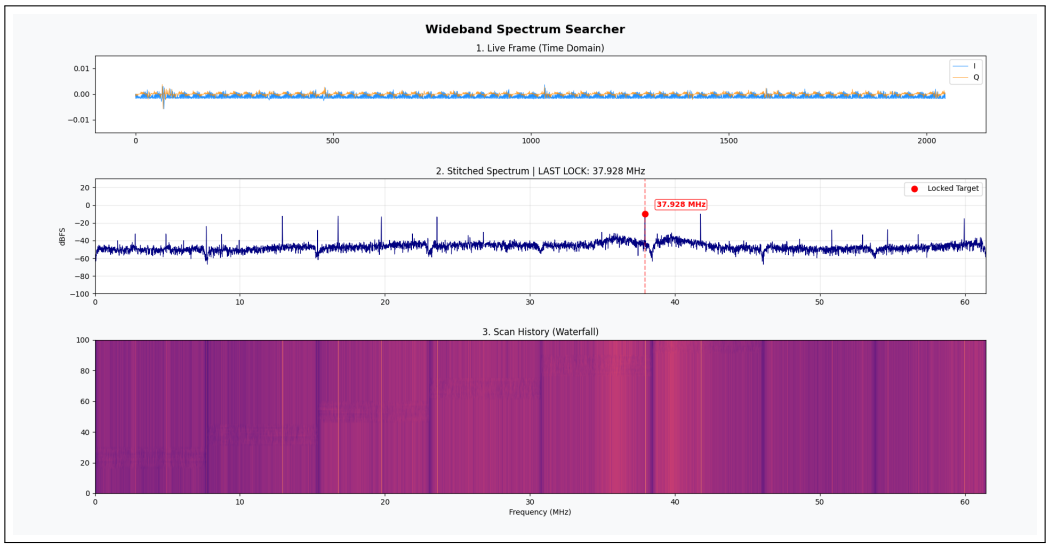


Figure 4.1: Wideband spectrum showing stitched FFT output across the scanned frequency range. Detected signals are visible across multiple frequency segments.

The FPGA streams wideband data using a high-throughput DMA-based ping-pong buffering mechanism, where 8 MB of contiguous memory is divided into two 4 MB chunks. Each completed chunk is transmitted over TCP to the receiver, ensuring continuous data flow without interruption.

A synchronized LO hopping mechanism is implemented in the FPGA, where the LO frequency is incremented after each transmitted chunk. The frequency step is set to 12.288 MHz, allowing controlled overlap between adjacent spectral segments. This approach ensures complete coverage of the target bandwidth (0–61.44 MHz) while maintaining efficient data acquisition.

On the receiver side, each 4 MB chunk is processed by extracting a fixed-length IQ segment and performing FFT-based spectral analysis. The resulting partial spectra are stitched together into a global spectrum buffer based on the corresponding LO frequency. An exponential averaging mechanism is applied during stitching to improve spectral stability and suppress noise fluctuations.

To enhance detection reliability, several signal conditioning techniques are applied:

- DC component suppression to eliminate LO leakage
- Edge masking to remove boundary artifacts during stitching
- Spur filtering to suppress known hardware-induced interference

Signal detection is performed using a dynamic SNR-based approach, where the noise floor is estimated using the median of the spectrum and peaks exceeding a predefined threshold are identified as valid signals.

The reconstructed wideband spectrum demonstrates accurate frequency mapping and consistent signal detection across the entire scanned range. The combination of DMA-based streaming, synchronized LO sweeping, and adaptive spectral stitching enables robust real-time wideband monitoring with minimal artifacts and no significant ghost frequencies.

4.3 Narrowband Signal Analysis Results

Once a signal is identified in the wideband stage, the system switches to a narrowband mode for high-resolution analysis of the detected frequency.

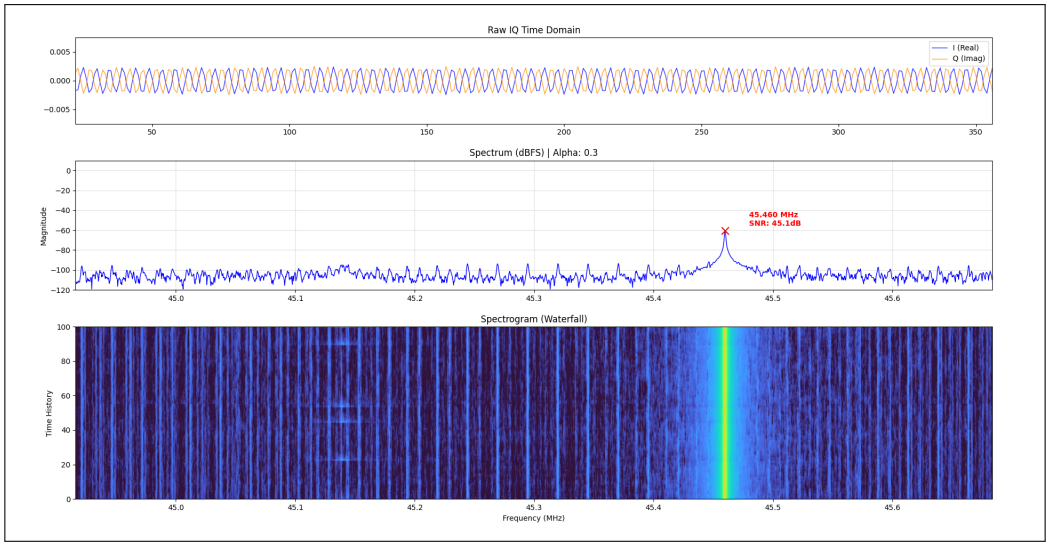


Figure 4.2: Narrowband IQ and FFT visualization at a fixed LO frequency of 45.3 MHz.

In the implemented system, the narrowband receiver operates at a reduced sampling rate of approximately 768 kSPS, obtained through FPGA-based decimation of the 122.88 MHz input clock. The Local Oscillator (LO) is fixed at 45.3 MHz during this stage, enabling focused analysis around the detected signal band. :contentReference[oaicite:0]index=0

The FPGA streams complex I/Q samples in chunks of 2048 samples (16 KB) using a FIFO-based interface. A flow-control mechanism monitors the FIFO occupancy and automatically resets the receiver in case of overflow, ensuring stable and continuous data acquisition.

At the receiver, the incoming I/Q samples are processed in real time. DC offset is removed, followed by windowed FFT computation to obtain the frequency spectrum. The spectrum is further stabilized using exponential averaging, which reduces noise fluctuations and improves visual clarity.

The narrowband plots demonstrate stable time-domain I/Q signals along with a high-resolution spectrum centered around the LO frequency. A dynamic SNR-based peak detection mechanism highlights dominant frequency components, enabling precise localization of the signal within the narrowband window.

This focused analysis stage complements the wideband scanning process by providing detailed spectral and temporal information, allowing accurate characterization of detected signals.

4.4 Machine Learning Model Performance

The extracted multi-domain features (I, Q, FFT, magnitude, and phase) were used to train machine learning models for RF signal classification. The dataset, generated across multiple protocols, SNR levels, and channel conditions, ensured robust training and improved generalization capability.

4.4.1 Model Architecture and Deployment Strategy

Initially, a multi-task learning framework was adopted, where a shared feature extraction backbone simultaneously supported:

- Protocol classification
- Modulation classification
- Regression (carrier frequency and bandwidth)

This architecture enabled the model to learn direct mappings from signal features to protocol labels, capturing complex relationships between modulation patterns, spectral characteristics, and protocol-specific signatures.

However, for deployment on the Xilinx ZCU104 platform using the Vitis AI DPU, the model was simplified to meet hardware constraints such as quantization efficiency, reduced latency, and limited support for multi-head outputs. The protocol classification head was removed, and the deployed model retains:

- Modulation classification
- Regression outputs (frequency and bandwidth)

Protocol identification is therefore performed using a rule-based inference mechanism:

$$\text{Signal} \rightarrow \text{Modulation + Parameters} \rightarrow \text{Rules} \rightarrow \text{Protocol}$$

While this approach improves deployability and efficiency, it introduces a reduction in protocol classification accuracy compared to the original end-to-end learned model.

4.4.2 Performance Evaluation

The models were evaluated using standard metrics including accuracy, precision, recall, F1-score, and regression error metrics (RMSE, MAE, and R^2). The results for different architectures are summarized below.

4.4.3 Model Performance Comparison

Table 4.1: Performance comparison of different machine learning models (modulation classification + rule-based protocol inference)

Metric	**CNN**	**CNN-LSTM**	**ANN**
Modulation Accuracy	0.8873	0.8898	0.6802
F1 Score	0.8896	0.8556	0.5999
Precision	0.8926	0.8836	0.7738
Recall	0.8873	0.8898	0.6802
RMSE	0.1075	0.1043	2.0351
MAE	0.0463	0.0493	0.7227
R^2 Score	0.8642	0.8689	-25.3440

The results indicate that CNN and CNN-LSTM models achieve high modulation classification accuracy (approximately 88%), demonstrating the effectiveness of the proposed feature representation. The CNN model provides better protocol classification performance, while CNN-LSTM shows slightly improved temporal feature learning.

In contrast, the ANN model exhibits significantly lower performance, particularly in regression tasks, as indicated by high RMSE and negative R^2 values. This highlights the importance of deep feature extraction and hierarchical learning in RF signal classification.

4.4.4 Modulation Classification (CNN) deployed in ZCU104 MPSOC+

The confusion matrix demonstrates strong classification performance across most modulation schemes. Frequency- and phase-based modulations such as BFSK, DQPSK, GFSK, and NBFM achieve near-perfect classification accuracy, as indicated by the strong diagonal values.

The overall **modulation classification accuracy of the CNN model is 88.73%**, confirming its effectiveness in capturing discriminative signal features.

However, confusion is observed among amplitude-based modulations. ASK and OOK exhibit overlap due to similar amplitude envelope characteristics, leading to mutual misclassification. Minor confusion is also observed in 4FSK classification under certain conditions.

Despite these overlaps, the CNN model demonstrates robust performance, making it suitable for downstream protocol inference.

Modulation classification is performed using rule-based inference derived from signal characteristics such as bandwidth, spectral shape, and temporal features.

The confusion matrix shows that frequency- and phase-based modulation schemes such as BFSK, DQPSK, GFSK, and NBFM achieve near-perfect classification accuracy, as their spectral signatures are distinct and easily separable.

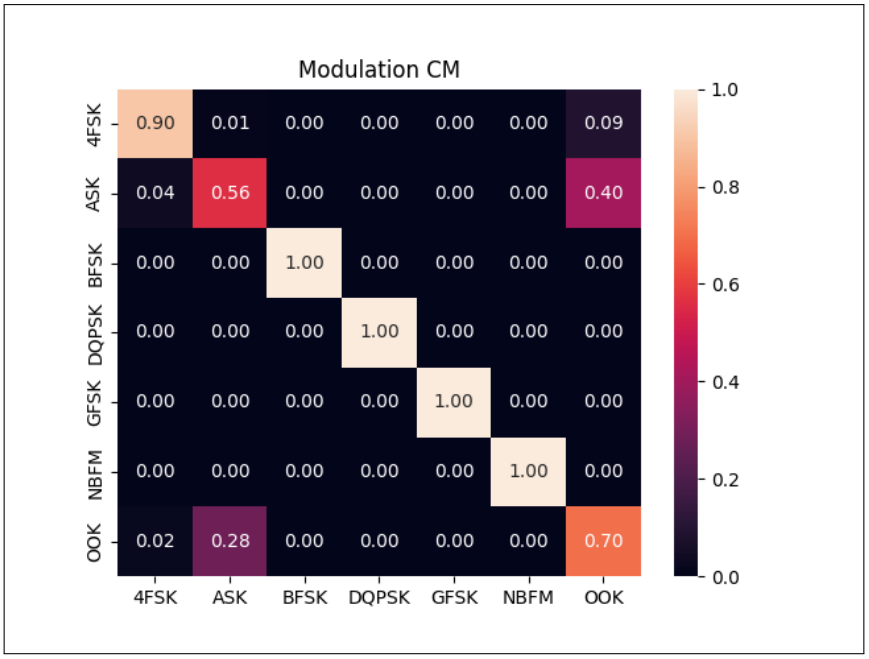


Figure 4.3: Confusion matrix for CNN-based modulation classification.

However, significant confusion is observed among amplitude-based modulation schemes. ASK is correctly classified in only about 56% of cases, while approximately 40% of ASK samples are misclassified as OOK. Similarly, OOK shows partial confusion with ASK (around 28%) and minor misclassification into 4FSK.

Additionally, small errors are observed in 4FSK classification, where a fraction of samples is incorrectly predicted as OOK. These results indicate that rule-based methods struggle to distinguish modulation schemes with similar amplitude envelope characteristics.

Overall, while rule-based inference provides a simple and interpretable approach, its performance is limited in handling closely related modulation types, especially under varying signal conditions.

4.4.5 Performance Across SNR

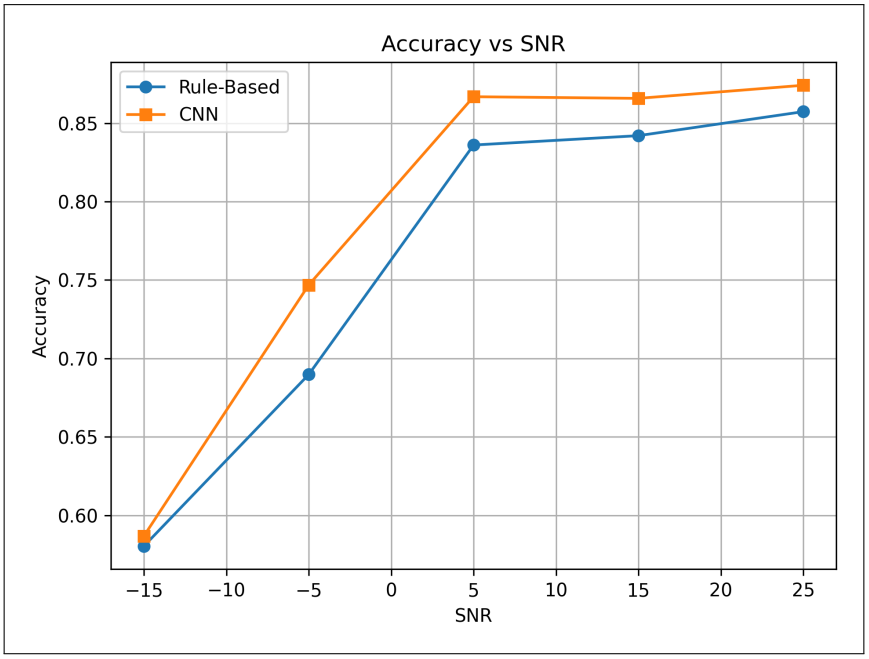


Figure 4.4: Accuracy comparison of CNN and rule-based inference across different SNR levels.

The performance of the system improves with increasing Signal-to-Noise Ratio (SNR).

At low SNR (−15 dB), the CNN achieves an accuracy of approximately 58%, demonstrating resilience under noisy conditions. As SNR increases, the accuracy improves significantly, reaching approximately **87–88% at high SNR levels (25 dB)**.

The rule-based inference follows a similar trend but consistently performs slightly lower than the CNN, as it depends on the accuracy of modulation predictions and parameter estimation.

4.4.6 Key Insight

The results highlight that:

- CNN achieves high modulation classification accuracy (**88.7%**)
- Rule-based inference provides reliable protocol classification (**80.7%**)
- System performance improves with increasing SNR, indicating robustness
- Most errors arise from similar signal structures (e.g., ASK vs OOK, FSK variants)

The CNN model is designed for deployment on the ZCU104 MPSoC platform using a hardware-accelerated inference framework. The trained model is quantized and compiled for execution on the Deep Processing Unit (DPU), enabling efficient real-time inference on incoming narrowband I/Q data. The signal is first processed through feature extraction, where

multi-domain features (time-domain and frequency-domain) are generated and provided as input to the CNN model.

The confusion matrix demonstrates strong classification performance across most modulation schemes. Frequency- and phase-based modulations such as BFSK, DQPSK, GFSK, and NBFM achieve near-perfect classification accuracy, as indicated by the strong diagonal values.

The overall **modulation classification accuracy of the CNN model is 88.7%**, confirming its effectiveness in capturing discriminative signal features.

However, confusion is observed among amplitude-based modulations. ASK and OOK exhibit overlap due to similar amplitude envelope characteristics, leading to mutual misclassification. Minor confusion is also observed in 4FSK classification under certain conditions.

Despite these overlaps, the CNN model demonstrates robust performance in simulation and dataset-based evaluation. While the model has been designed for FPGA-based deployment on the ZCU104, full end-to-end validation with real RF signals in the 390–480 MHz band was limited due to the absence of a heterodyne receiver for upconversion.

Nevertheless, the proposed architecture, feature pipeline, and inference framework are fully compatible with real-time deployment, indicating strong potential for embedded RF signal classification applications when integrated with an appropriate RF front-end.

4.5 Real-Time System Performance

The TCP/IP-based communication framework was evaluated for real-time operation between the FPGA-based SDR (Red Pitaya) and the host processing unit. The system enables continuous acquisition, transmission, and analysis of RF signals with minimal latency.

The system supports simultaneous streaming of:

- Wideband data (DMA-based transfer for high-throughput spectrum sensing)
- Narrowband data (2048-sample I/Q packets for detailed signal analysis)

Tagged packet headers are used to ensure reliable synchronization and correct data parsing across the communication pipeline.

Experimental results demonstrate that continuous wideband streaming, combined with low-latency command feedback, enables a fast detection-to-lock transition. Detected frequencies from the wideband spectrum are fed back to the FPGA, allowing the narrowband receiver to dynamically tune and track the signal in real time.

To ensure that no signal is missed, the narrowband receiver locks onto the detected frequency with a fixed bandwidth window of approximately ±384 kHz. This provides stable signal tracking while enabling high-resolution analysis of the selected signal.

Efficient data handling is achieved through the use of DMA for wideband transfer and FIFO buffering for narrowband data, ensuring continuous operation without data loss. Real-

time visualization is supported through buffered streaming, multi-threaded processing, and non-blocking updates on the host system.

The complete system operation, including hardware setup, real-time visualization, and on-board signal analysis, is illustrated in Figures 4.5, 4.6, and 4.7.

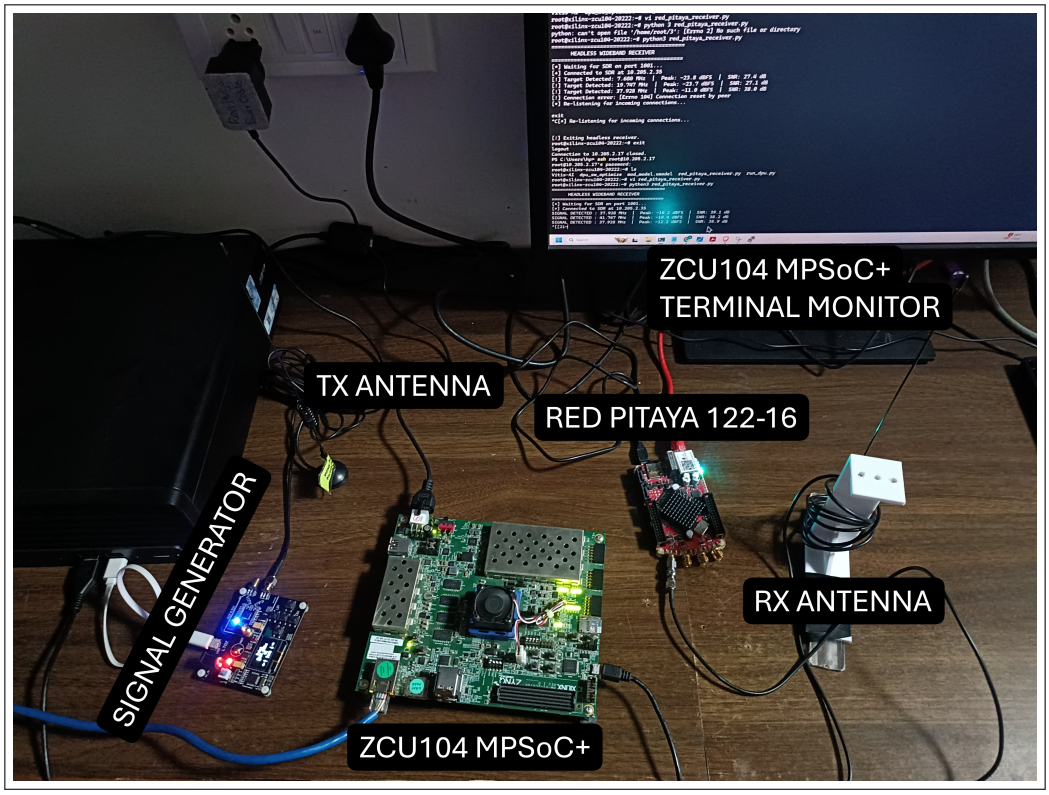


Figure 4.5: End-to-end hardware setup showing signal transmission from the signal generator, RF reception using the Red Pitaya SDR (122-16), and signal processing on the ZCU104 FPGA platform.

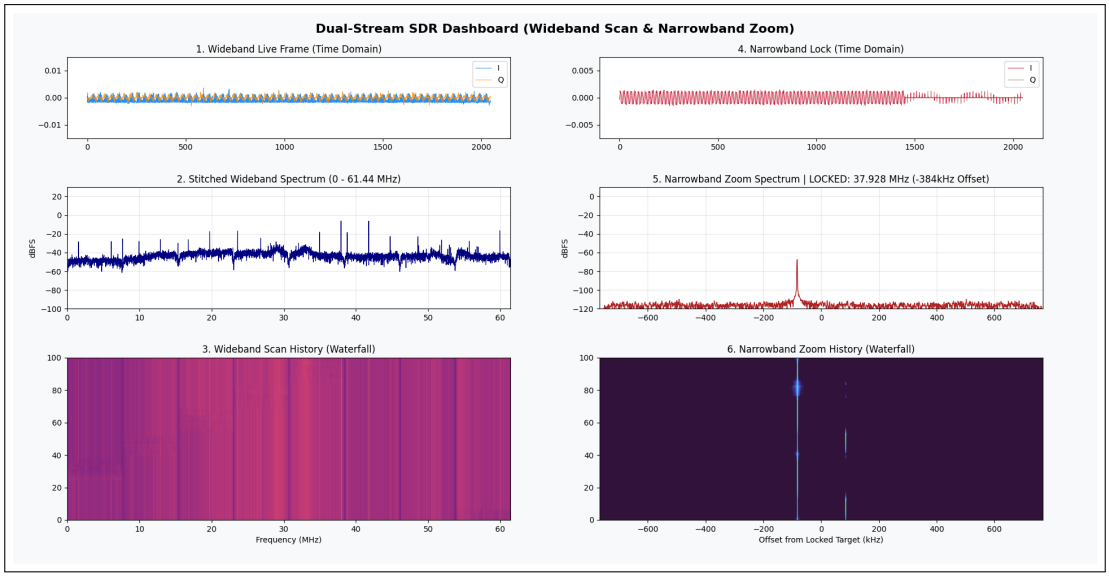


Figure 4.6: Real-time dashboard showing wideband spectrum sensing and narrowband signal tracking. The system detects signals in the wideband view and dynamically locks the narrowband receiver (±384 kHz) onto the detected frequency for detailed analysis.

```
PS C:\Users\hp> ssh root@10.205.2.17
root@10.205.2.17's password:
root@xilinx-zcu104-20222:~# ls
Vitis-AI  dpu_sw_optimize  mod_model.xmodel  red_pitaya_receiver.py  run_dpu.py
root@xilinx-zcu104-20222:~# vi red_pitaya_receiver.py
root@xilinx-zcu104-20222:~# python3 red_pitaya_receiver.py
========================================
      HEADLESS WIDEBAND RECEIVER
========================================
[*] Waiting for SDR on port 1001...
[+] Connected to SDR at 10.205.2.35
SIGNAL DETECTED : 37.928 MHz  |  Peak: -10.2 dBFS  |  SNR: 39.1 dB
SIGNAL DETECTED : 41.767 MHz  |  Peak: -10.4 dBFS  |  SNR: 38.2 dB
SIGNAL DETECTED : 37.928 MHz  |  Peak: -12.2 dBFS  |  SNR: 35.9 dB
^[[21~SIGNAL DETECTED : 41.767 MHz  |  Peak: -8.7 dBFS  |  SNR: 39.6 dB
SIGNAL DETECTED : 37.928 MHz  |  Peak: -11.9 dBFS  |  SNR: 36.5 dB
[!] Connection error: [Errno 104] Connection reset by peer
[*] Re-listening for incoming connections...

exit
^C[*] Re-listening for incoming connections...


[!] Exiting headless receiver.
root@xilinx-zcu104-20222:~# exit
logout
Connection to 10.205.2.17 closed.
PS C:\Users\hp>
```

Figure 4.7: Terminal output from the ZCU104 platform showing real-time signal processing and analysis and system-level execution.

4.6 Comparative Analysis

The proposed system demonstrates significant improvements over conventional RF monitoring approaches by combining efficient spectrum scanning with real-time signal tracking.

- **Efficient Spectrum Scanning:** Wideband sensing is performed using local oscillator (LO) sweeping, avoiding the need for full-rate FFT processing over the entire bandwidth and reducing computational complexity.
- **Staged Processing Architecture:** The system separates wideband detection and narrowband analysis, minimizing processing load while enabling focused high-resolution signal characterization.
- **Improved Detection Reliability:** Overlapping frequency segments during wideband scanning ensure continuous coverage and reduce the probability of missing transient or weak signals.
- **Real-Time Adaptive Tracking:** Detected signals are fed back to the system, allowing dynamic tuning of the narrowband receiver for continuous signal tracking and analysis.

In contrast to purely software-based SDR systems, the FPGA-based implementation provides higher throughput, deterministic latency, and efficient memory handling through DMA and FIFO-based streaming, enabling reliable real-time operation. The integrated TCP/IP communication framework further ensures synchronized wideband and narrowband data transfer between the hardware and host system for real-time visualization and processing

4.7 Discussion

The proposed wideband–narrowband architecture demonstrates an effective balance between spectrum awareness and detailed signal analysis. The wideband sensing stage enables efficient detection over a large frequency range, while the narrowband stage provides focused, high-resolution characterization of detected signals. This staged approach reduces computational complexity compared to conventional methods and ensures reliable detection without missing transient signals.

From a machine learning perspective, the results highlight the effectiveness of multi-domain feature representation in RF signal classification. CNN-based models significantly outperform ANN due to their ability to capture localized spectral and temporal patterns, while CNN-LSTM offers only marginal improvement, indicating that most discriminative features are already extracted by convolutional layers. The transition to a simplified deployment model introduces reliance on rule-based inference for protocol identification, which, although efficient, shows limitations in distinguishing signals with similar characteristics, particularly among amplitude-based and closely related FSK-based protocols.

The overall system demonstrates strong real-time performance through efficient FPGA–receiver integration, leveraging DMA-based wideband streaming and FIFO-based narrowband processing. The feedback-driven tuning mechanism enables rapid detection-to-lock transitions, ensuring continuous signal tracking. While minor limitations remain in protocol inference and

flexibility for highly dynamic signals, the system provides a scalable and practical framework for real-time RF monitoring and spectrum intelligence applications.

Chapter 5

Summary and Future Work

This work presents the design, implementation, and evaluation of a unified FPGA-based SDR framework for real-time RF signal detection and classification. The proposed system integrates wideband spectrum sensing, narrowband signal extraction, and machine learning-based classification within a single end-to-end architecture. By combining efficient DSP techniques, structured dataflow design, and embedded-oriented learning models, the system achieves reliable real-time performance in complex RF environments.

Summary of Contributions

This thesis makes several key contributions toward practical RF signal intelligence systems.

First, a unified wideband–narrowband processing architecture was developed, where wideband sensing enables spectrum awareness and signal detection, followed by narrowband processing for detailed I/Q extraction and signal characterization. This staged approach reduces computational overhead while maintaining high detection reliability.

Second, a protocol-centric dataset generation framework was introduced, where signals are modeled based on real-world communication protocols rather than abstract modulation schemes. This includes the incorporation of protocol-specific characteristics such as bandwidth allocation, guard bands, and channelization, resulting in a more realistic and representative RF dataset.

Third, a structured feature extraction pipeline was designed to combine time-domain (I/Q, magnitude, phase) and frequency-domain (FFT) representations into a unified format suitable for machine learning. This multi-domain representation enables robust classification under varying noise and channel conditions.

Fourth, multiple machine learning models (ANN, CNN, CNN–LSTM) were developed and evaluated using a multi-task learning framework. While CNN-based models demonstrated superior performance, the final system adopts an embedded-optimized approach where modulation classification is performed using neural networks, and protocol identification is achieved

through rule-based inference using predicted signal parameters.

Finally, the complete system was implemented on FPGA-based SDR platforms with real-time streaming, dynamic frequency tuning, and efficient data handling. The integration of wideband acquisition, narrowband tracking, and classification demonstrates a scalable and practical solution for real-time RF monitoring.

Future Work

Several directions can further enhance the capability and applicability of the proposed system.

One important extension is the development of fully end-to-end deep learning models capable of directly performing protocol classification without relying on rule-based inference. This would improve robustness in scenarios involving overlapping or ambiguous signal characteristics.

Another promising direction is the incorporation of advanced spectrum processing techniques such as polyphase channelization, adaptive filtering, and interference suppression to improve performance in highly congested environments. Support for wideband signals with dynamic bandwidths and frequency hopping behavior can also be explored.

From a hardware perspective, the system can be further optimized through tighter integration of DSP and machine learning components, including on-FPGA inference pipelines and reduced data movement. Exploring alternative high-speed interfaces and optimizing memory utilization would enhance scalability and throughput.

Additionally, extending the dataset to include real captured signals and more diverse channel impairments would improve model generalization and real-world applicability. Incorporating online learning or adaptive models could further enable the system to evolve with changing RF environments.

Conclusion

This thesis demonstrates that FPGA-based SDR platforms provide a powerful and practical foundation for real-time RF signal detection and classification. By integrating wideband sensing, narrowband analysis, and data-driven learning within a unified architecture, the proposed system achieves a balance between accuracy, efficiency, and deployability.

The combination of protocol-aware modeling, structured feature representation, and embedded-oriented machine learning establishes a scalable framework for next-generation RF intelligence systems. With further enhancements, the proposed approach can be extended to a wide range of applications, including spectrum monitoring, wireless surveillance, cognitive radio, and adaptive communication systems.

Bibliography

[1] Red Pitaya, "Advanced FPGA Topics," *Red Pitaya Knowledge Base*. [Online]. Available: https://redpitaya-knowledge-base.readthedocs.io/en/latest/learn$_{f}pga/advanced.html$

[2] Red Pitaya, "Digital Systems & FPGA Programming," [Online]. Available: https://learn.redpitaya.com/digital-systems-fpga-programming/

[3] S. Tang, S. Moua, Y. Xie, and Y. Zheng, "FPGA-Based Implementation of an Audio Signal Processing System on ZedBoard," *Journal of Software Technology and Application*, vol. 3, no. 1. [Online]. Available: https://gvpress.com/journals/JSTA/vol3$_{n}o1/1.pdf$

[4] R. Kumar *et al.*, "FPGA Based Data Acquisition Using RedPitaya STEMlab 125-10," ResearchGate, 2023. [Online]. Available: https://www.researchgate.net/publication/372510406

[5] G. Cameli and M. Gallieni, "A System for Digital Acquisition and Processing of Marine Seismic Profiling Data," in *OCEANS 1974*, pp. 1–7, 1974. Available: https://doi.org/10.1109/OCEANS.1974.1161450

[6] S. Bhattacharya, "Development of a Summing Module with Vivado on RedPitaya Board," Fermilab Technical Report. Available: https://indico.fnal.gov/event/66163/contributions/300032/attachments/182058/249879/FermiLabPaperReport.pdf

[7] C. Cassinelli, "ADC Noise Measurement on FPGA Board," Master's thesis, Politecnico di Torino, 2018. [Online]. Available: https://webthesis.biblio.polito.it/12535/1/tesi.pdf

[8] L. Vsandi, "Xilinx Zynq DMA HDL Example," [Online]. Available: https://lauri.vsandi.com/hdl/zynq/xilinx-dma.html

[9] Xilinx, "Fast Fourier Transform v9.1 LogiCORE IP Product Guide (PG109)," May 2022. [Online]. Available: https://www.xilinx.com/support/documents/ip$_{d}ocumentation/xfft/v9_{1}/pg109-xfft.pdf$

[10] UviDTE-FPSoC, "Vitis AI Installation Guide for Zynq-7000," [Online]. Available: https://github.com/UviDTE-FPSoC/Zynq7000-dnn-inference/blob/master/vitis$_{a}i_{i}nstallation.md$

[11] Xilinx, "UG1414: Vitis AI User Guide," [Online]. Available: https://www.scribd.com/document/888938189/Ug1414-Vitis-Ai

[12] "Introduction to Real-Time Spectrum Analysis," Everything RF. [Online]. Available: https://cdn.everythingrf.com/live/Introduction$_{t}o_{R}eal_{T}ime_{S}pectrum_{A}nalysis_{136}$38761395960670098.$pdf$

[13] Y. P. et al., "MIT RF Challenge Dataset," [Online]. Available: https://people.lids.mit.edu/yp/homepage/data/2024$_{r}fchallenge_{f}ull.pdf$

[14] Xilinx, "Vitis AI Quick Start for MPSoC," [Online]. Available: https://xilinx.github.io/Vitis-AI/3.0/html/docs/quickstart/mpsoc.html

[15] ETSI, "TETRA Standard EN 300 392-1," [Online]. Available: https://www.etsi.org/deliver/etsi$_{e}n$/300300$_{3}$00399/30039201/01.03.01$_{6}$0/en_{3}0039201v010301$p.pdf$

[16] Panoradio SDR, "RF Training Data Generation for Machine Learning," [Online]. Available: https://panoradio-sdr.de/rf-training-data-generation-for-machine-learning/

[17] UCSD, "RF Signal Classification Report," [Online]. Available: https://noiselab.ucsd.edu/ECE228$_{2}$019/$Reports/Report$42.pdf

[18] Xilinx Wiki, "Zynq UltraScale+ MPSoC Architecture," [Online]. Available: https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/1186693157/

[19] A. John, "CNN-LSTM for Signal Classification," Kaggle. [Online]. Available: https://www.kaggle.com/code/afajohn/cnn-lstm-for-signal-classification-lb-0-513

[20] S. Yang, Y. Luo, W. Miao, C. Ge, W. Sun, and C. Luo, "RF Signal-Based UAV Detection and Mode Classification Using Deep Neural Networks," 2024.

[21] M. Al-Sa'd *et al.*, "DroneRF Dataset: RF-Based Drone Detection and Classification," Mendeley Data, 2019. DOI: 10.17632/f4c2b4n755.1

[22] Federal Communications Commission (FCC), "47 CFR § 90.210 – Private Land Mobile Radio Services," [Online]. Available: https://www.law.cornell.edu/cfr/text/47/90.210

[23] ETSI, "TS 102 361-1: Digital Mobile Radio (DMR) Air Interface," [Online]. Available: https://www.etsi.org/deliver/etsi$_{t}s$/102300$_{1}$02399/10236101/02.06.01$_{6}$0/ts_{1}0236101v020601$p.pdf$

[24] ETSI, "EN 300 392-2: TETRA Voice plus Data Air Interface," [Online]. Available: https://www.etsi.org/deliver/etsi$_{e}n$/300300$_{3}$00399/30039202/03.04.01$_{6}$0/en_{3}0039202v030401$p.pdf$

[25] FCC, "47 CFR § 95.573 – Family Radio Service," [Online]. Available: https://www.ecfr.gov/current/title-47/chapter-I/subchapter-D/part-95/

[26] Vaisala, “RS41 Radiosonde Datasheet,” [Online]. Available: https://hu-bc.org/wp-content/uploads/2024/07/RS41$_SGP - Datasheet.pdf$

[27] ETSI, “EN 302 054 – Meteorological Aids (Radiosondes),” [Online]. Available: ETSI official repository

[28] ETSI, “EN 300 113 – Land Mobile Service Data Transmission,” [Online]. Available: https://www.etsi.org/deliver/etsi$_en/300100_300199/300113/02.02.01_60/en_300113v020201p.pdf$

[29] ISO/IEC, “ISO/IEC 18000-63 – UHF RFID Air Interface,” [Online]. Available: https://www.iso.org/standard/81723.html

[30] ITU-R, “Recommendation M.584-2 – Radio Paging,” [Online]. Available: https://www.itu.int/dms$_pubrec/itu-r/rec/m/R-REC-M.584-2-199711-I!!PDF-E.pdf$

[31] FCC, “47 CFR § 15.231 – Short Range Devices,” [Online]. Available: https://www.ecfr.gov/current/title-47/chapter-I/subchapter-A/part-15/subpart-C/section-15.231

[32] H. Nyquist, “Certain Topics in Telegraph Transmission Theory,” *Transactions of the AIEE*, vol. 47, 1928.

[33] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 3rd ed. Prentice Hall, 2010.