Deep Learning-Based Signal Classification (Using RTL-SDR and Convolutional Neural Networks)

<u>Team – D14</u>

Asmi K - CB.SC.U4AIE23351

Vibhu Sanchana - CB.SC.U4AIE23347

Kowshik - CB.SC.U4AIE23314

Mhokesh - CB.SC.U4AIE23316

Introduction

Traditional RF signal classification requires manual feature extraction, making it inefficient for real-time applications.

- Automated RF signal classification benefits prevent interference, optimize communication.
- Deep Learning (CNNs) can learn features automatically, improving accuracy and scalability.
- CNNs can classify AM, FM, BPSK, and other modulation types without manual processing.

Why Use CNN for RF Signal Classification rather than traditional ML models?

- CNNs reduce Preprocessing effort
- Automatic Feature Extraction – No need for manual feature engineering like in traditional ML models.
- Robust to Noise – CNNs can generalize well even with noisy signals.
- Handles Spectrograms Efficiently – CNNs work well with time-frequency representations like spectrograms (high-dimensional images).
- Scalable – Can classify multiple modulation types effectively.

Literature Review

SL No	Title	Journal & Year	Authors	Key Outcomes	Limitations
1	Software Defined Radio (SDR) based sensing	2024	Dahal, Ajaya	Explores SDR-based sensing techniques	Lacks deep learning implementation
2	A cognitive radio spectrum sensing implementation based on deep learning and real signals	Innovations in Smart Cities Applications, 2021	Mohamed Saber et al.	Uses deep learning for cognitive radio sensing	Limited real-world testing

3	Practical implementation of modulation classification and adversarial attacks using Universal Software Radio Peripheral with deep learning	2023	Salma Sultana	Demonstrates modulation classification with adversarial attacks	Requires specialized hardware
4	Software Defined Radio using MATLAB & Simulink and the RTL-SDR	Strathclyde Academic Media, 2015	Robert W. Stewart et al.	Provides a tutorial on SDR using MATLAB	Lacks modern deep learning integration
5	Low cost digital transceiver design for Software Defined Radio using RTL-SDR	IEEE, 2013	M. B. Sruthi et al.	Develops a low-cost digital transceiver	Limited computational capabilities

Objectives

- Develop an AI-driven RF signal classification system using RTL-SDR and CNNs.
- Train a CNN model to classify signals like AM, FM, BPSK, etc.
- Implement feature extraction using spectrograms & raw IQ data.
- Compare CNN-based classification vs. traditional ML techniques.
- Explore real-time IoT applications using RF messaging.

Expected Outcome

- A trained deep learning model that accurately classifies radio signals.
- Demonstration of improved performance over traditional ML methods.
- Low-cost, real-time IoT integration using RF messaging - alerts users or systems about detected **unauthorised** RF signals, enabling real-time monitoring

Relevance to Machine Learning :

Feature Extraction & Preprocessing :

1. Synthetic RF signals
2. Spectrograms (image-based transformation of RF signals)

Feature Scaling Techniques :

1. Min-Max Scaling → Normalizes signal amplitudes.
2. Standardization (Z-score) → Improves CNN stability.

Machine Learning Models Used :

1. CNNs (Convolutional Neural Networks).

Evaluation Metrics

Data Collection

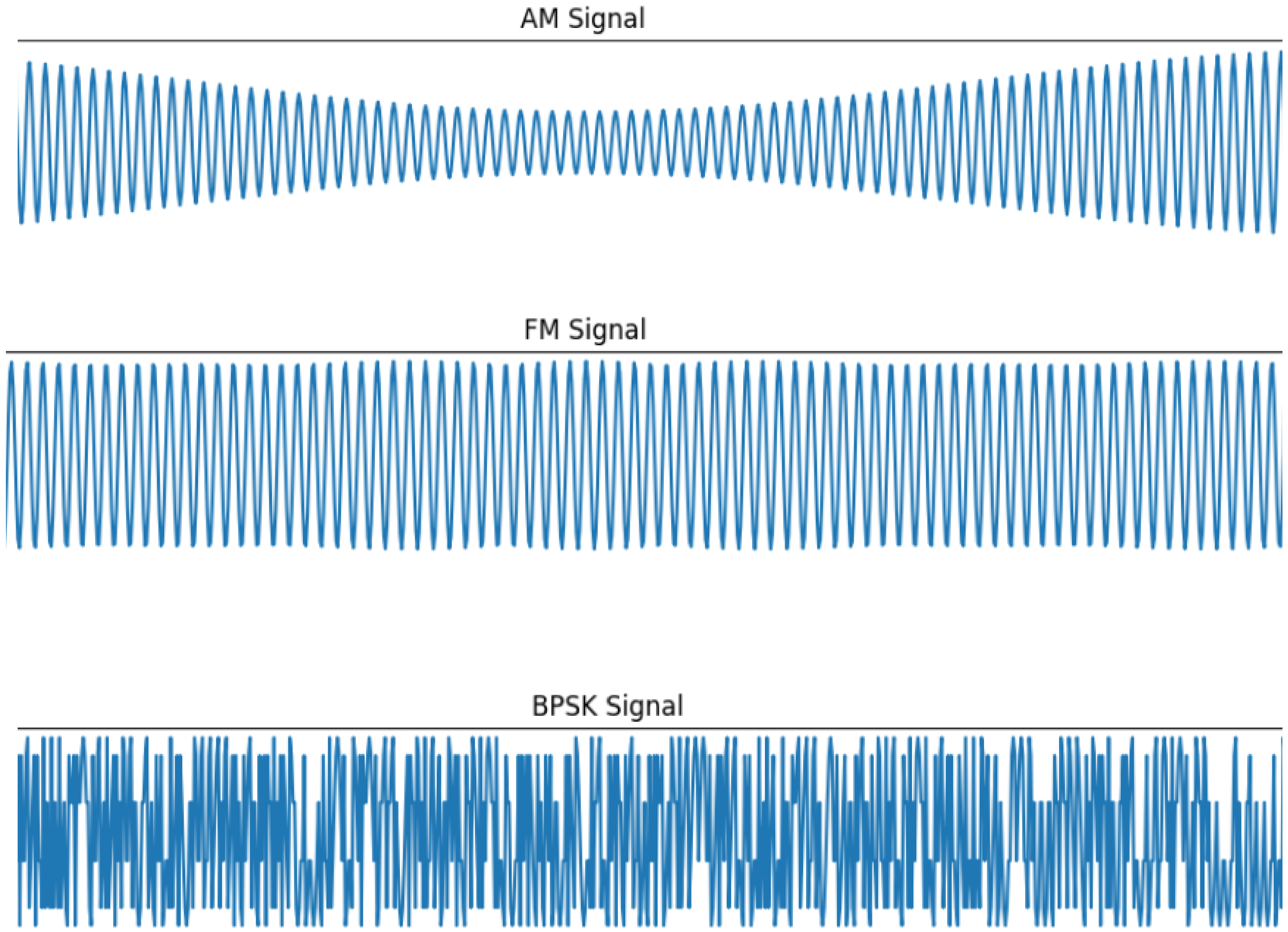
Since we aim to classify RF signals (e.g., AM, FM, PM, PSK), we need a dataset of RF signals.

- Using Pre-Existing Datasets
- Raw IQ (In-phase & Quadrature) data
- Labels for signal types (AM, FM, PM, BPSK, etc.)

Dataset Overview

"The **RadioML 2016.10A** dataset is designed for training machine learning models in **radio signal classification**. It simulates real-world radio signals and includes both analog and digital modulations."

- **11 types of modulation** : 8 digital (e.g., QPSK, BPSK, 16QAM) and 3 analog (e.g., AM, FM).



The dataset contains signals with various **modulation types at different noise levels (SNRs)** to mimic real-world conditions.

A **higher SNR** means the signal is much **stronger than the noise**, making it **easier to detect and classify**. A **lower SNR** means the signal is buried in noise, making it **difficult to recognize.**

The dataset includes **signals at 20 different SNR levels**, ranging from **-20 dB to +18 dB** in steps of 2 dB, **Each modulation type appears at all these SNR levels**, allowing models to be **trained under both noisy and clear conditions**.

Each signal consists of **1,000 samples**, where each sample contains **128 IQ points.**

- IQ samples are stored as complex numbers, representing the real and imaginary parts of the signal:
  - I (In-phase component) → Real part
  - Q (Quadrature component) → Imaginary part

CNN Architecture

1. Input Layer → Takes 128×128 spectrograms.
2. Convolutional Layers → Extract signal patterns using 3×3 filters.
3. Pooling Layers → MaxPooling (2×2) reduces size & prevents overfitting.
4. Fully Connected (Dense) Layers → Learns patterns from extracted features.
5. Output Layer → Uses Softmax to classify signals (AM, FM, BPSK, etc.).

Implement an RF-based IoT Messaging System

1)Simulated Transmission (Software-Only)

Instead of actually transmitting over the air, the signal is saved as a waveform file (.wav or .iq file).

The receiver reads the waveform file, simulating real transmission.

The CNN model then classifies and decodes the signal.

1. Encode message ("Hello" → BPSK modulated RF signal).
2. Save it as a file (hello_signal.wav).
3. Load it on the receiver and classify it with CNN.
4. Demodulate it back to text ("Hello").

Benefits of Real-Time RF Messaging in IoT Security:

- Enhanced Security: Detecting unauthorized signals in real time allows for faster responses to potential threats, improving overall security.
- Automation: The system can automate the detection and response process, reducing the need for manual intervention.
- Cost-Effective: Implementing RF-based security systems can be a cost-effective solution compared to traditional wired security setups, especially for large-scale environments.

Timeline

Research & Setup (Jan 24 – Feb 4)
Conduct literature review & finalize objectives.
Install required tools (TensorFlow, GNU Radio, etc.).
collect initial signal samples

Data Collection & Preprocessing (Feb 5 – Feb 14)
Capture IQ data & generate spectrograms.
Label dataset & prepare training data.
Explore feature extraction techniques.
.

Model Development & Training (Feb 15 – Mar 1)
Implement CNN model for signal classification.
Tune hyperparameters & optimize architecture.
Compare with traditional ML models (SVM, KNN, etc.)

Real-Time Testing & Optimization (Mar 2 – Mar 14)
Test model with live signals from RTL-SDR.
Optimize inference speed for real-time performance..

Documentation & Presentation Prep (Mar 15 – Mar 24)
Draft final report with methodology & results.
Conduct trial presentations & get feedback.

Final Review & Submission (Mar 25 – April 1)

References

[1]https://www.reddit.com/r/RTLSDR/comments/7itckh/neural_network_deep_learning_signal_recognition/

[2] https://arxiv.org/pdf/1602.04105

[3] Stewart, Robert W., et al. "A low-cost desktop software defined radio design environment using MATLAB, simulink, and the RTL-SDR." IEEE Communications Magazine 53.9 (2015): 64-71.