OPEN-SOURCE REAL-TIME AUTOMATIC MODULATION CLASSIFICATION WITH DEEP LEARNING FOR INTERNET OF THINGS DEVICES

Simon Boka, Oladayo Bello, Innocent Davidson

Presented by: **Simon Boka**
Realtime Embedded Software Engineer

sboka@vols.utk.edu

About the Presenter

Educational Background:
Master of Science in Computer Science (Data Mining and Intelligence Systems),
The University of Tennessee Knoxville, 2025

Professional Experience:
Current Role: Embedded Software Engineer at Raytheon, integrating mission-critical, real-time C++ software on multi-core ARM/VxWorks platforms.

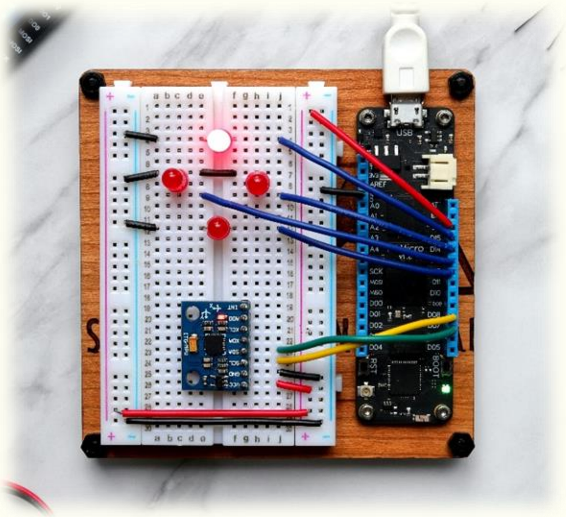
Previous Experience: Senior Engineer/Scientist at EPRI, Inc., developing data analysis tools and software for DER communication protocols and HEMS.

Key Research Interests:
- ❖ Edge AI and deploying machine learning models on resource-constrained embedded hardware.
- ❖ Architecting real-time embedded systems for mission-critical AI applications.
- ❖ Distributed intelligence, IoT device management, and fog computing architectures.

Simon Boka
Realtime Embedded Software Engineer
sboka@vols.utk.edu

Presentation Outline

- **Introduction & Motivation**
  The challenge of Automatic Modulation Classification (AMC) on IoT devices.
- **Proposed Solution**
  An open-source platform combining Software-Defined Radio and RISC-V.
- **Contributions & Significance**
  Key benefits for IoT deployments.
- **Hardware Selection**
  Survey of RISC-V platforms and SDR front-ends.
- **Methodology & Workflow**
  The end-to-end pipeline from signal to deployed model.
- **Conclusion & Future Work**
  Summary of findings and next steps.



Source: Jorge Ramirez - Unsplash (2bJ2OH9e9J8)

MOTIVATION AND CONTRIBUTION

Introduction & Motivation

What is AMC?

Deep Learning (DL) has redefined Automatic Modulation Classification (AMC), using neural networks to process raw signal data directly.

High Performance, High Cost:

Modern CNN and hybrid models achieve high accuracy (often over 90%) at good Signal-to-Noise Ratios (SNRs), but their computational cost is significant.

The IoT Challenge:

These high costs make it difficult to deploy advanced AMC models on resource-constrained Internet of Things (IoT) edge devices.

Persistent Issues:

Key challenges remain, including poor performance at low SNR, difficulty generalizing from synthetic data, and maintaining real-time processing on limited hardware.

Proposed Solution: An Open-Source Platform

This work proposes a low-cost, open-source radio platform for real-time Automatic Modulation Classification (AMC) on edge devices.

- Core Components:
  - **Signal Acquisition:** A commodity Realtek Software-Defined Radio (RTL-SDR) is used for inexpensive and reliable signal capture.
  - **Accelerated Inference:** A Reduced Instruction Set Computer Five (RISC-V) processor with vector extensions (RVV) is used to accelerate AI model execution.
- **Key Idea:** Pair modern deep learning techniques with vectorized execution on widely available, open-source RISC-V hardware to make spectrum intelligence more accessible.

Contributions & Significance for IoT

1. **On-Device Intelligence:** Equips IoT nodes with the ability to monitor, classify, and react to the local RF environment in real-time without cloud dependency.

2. **Shortens Path to Deployment:** Bridges the gap between simulated data and real-world performance, enabling use cases like interference detection in smart cities and factories.

3. **Enables In-Situ Adaptation:** Allows IoT devices to adapt their radio operations on-the-fly, such as selecting robust modulations to handle congestion.

4. **Facilitates Wide-Area Monitoring:** Distributed, low-power IoT gateways can classify signals locally and share only compact summaries, improving scalability and privacy for applications like utility metering and asset tracking.

OVER THE COUNTER OPEN-SOURCE DEVICES SURVEY

Hardware Survey: SDR Front-Ends

Software-Defined Radio (SDR) front-ends are used for signal capture.

For IoT, the primary task is often passive spectrum monitoring, so receive-only (RX) devices are sufficient.

SDRs were evaluated on:

- Frequency coverage
- Sampling rate and ADC depth
- Frequency stability (TCXO)
- Cost

Selection Rationale: The RTL-SDR Blog V4 was selected for its stable frequency control, good performance, and integrated filtering at a very low cost.

Hardware Survey: RISC-V Compute Platforms

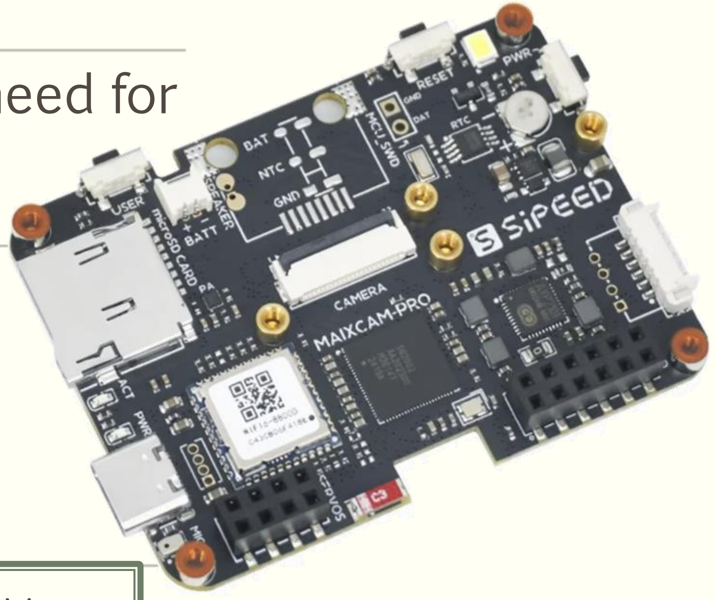
RISC-V platforms are well-suited for IoT nodes as they balance affordability and power efficiency.

They enable local feature extraction and inference, reducing the need for constant cloud backhaul and improving scalability.

Platforms evaluated on:

- CPU microarchitecture
- Vector / NPU acceleration
- Memory capacity
- Cost for Bill of Materials (BOM) planning

Selection Rationale: The Sipeed MaixCAM chosen for its integer SIMD capabilities and a small NPU, which accelerate inference under tight power and memory budgets.



END-TO-END WORKFLOW

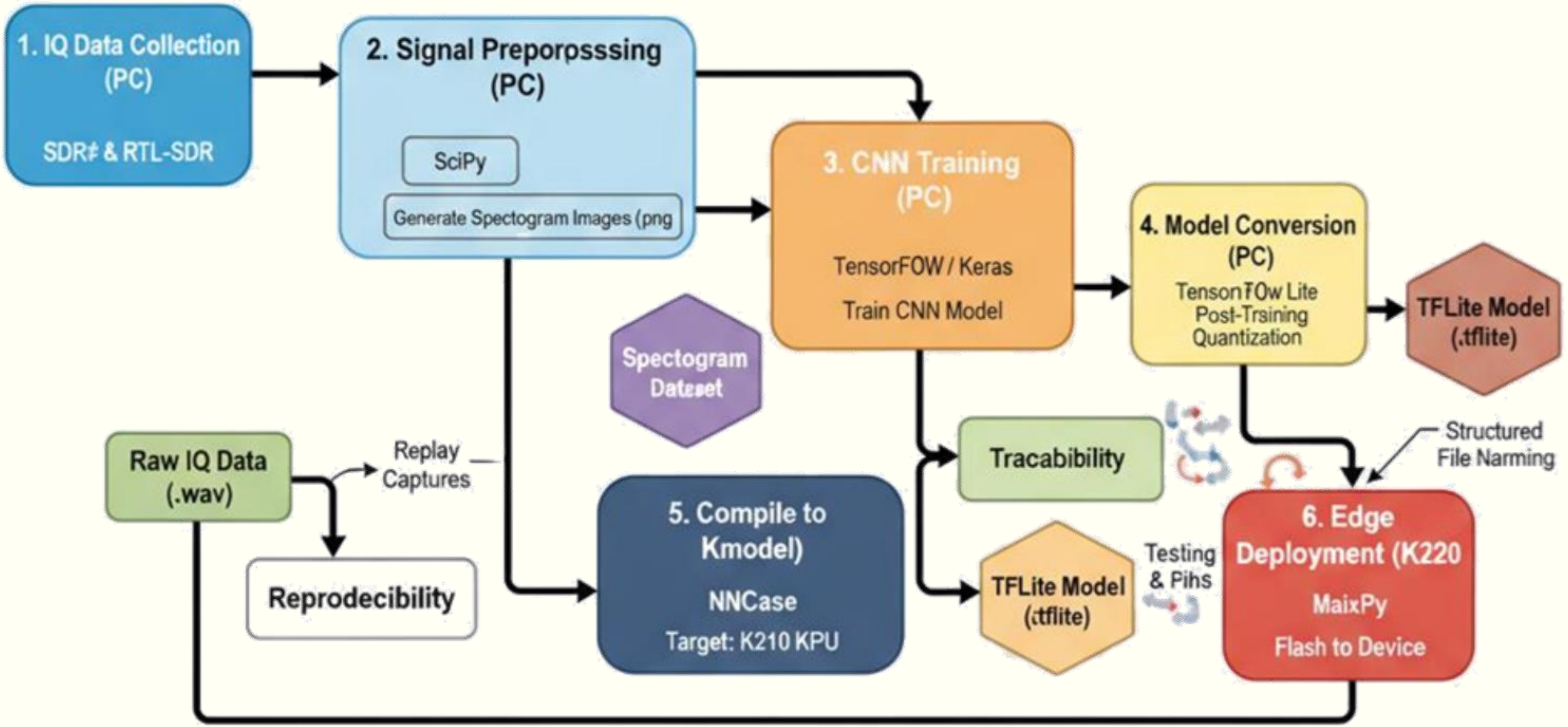
End-to-End Workflow Methodology

A reproducible, end-to-end pipeline was developed to deploy signal classification models on resource-constrained devices.

The entire pipeline runs on a desktop host but outputs a compact kmodel artifact specifically for the Kendryte K210's Processing Unit (KPU).

This approach ensures that the resulting models are fully compatible with IoT edge devices for autonomous spectrum classification.

Workflow Approach



Workflow Step 1: Data Collection

Tool:

SDR# software with an RTL-SDR is used for interactive signal capture.

Process:

Raw complex baseband (I/Q) streams are recorded and saved as WAV files.

This allows for precise offline replay and deterministic dataset creation.

A small, labeled dataset was created by capturing:

- A strong local FM broadcast.
- A NOAA Weather Radio transmission.
- An unoccupied channel for a noise baseline.

Workflow Step 2: Signal Preprocessing

Goal:	Transform the 1D complex I/Q arrays into 2D time-frequency images suitable for a CNN.
Method:	A Short-Time Fourier Transform (STFT) is used to generate a spectrogram.
Process:	Each I/Q recording is loaded. Spectrograms are computed with a fixed window and overlap for consistent resolution. Images are log-scaled, normalized, and saved to per-class folders for training.

Workflow Step 3: CNN Training

Framework: TensorFlow/Keras is used to define and train a compact Convolutional Neural Network (CNN).

Architecture: The model uses a simple stack of convolutional and pooling layers, ending in a softmax output layer for classification.

Process: The model is trained on the directory of labeled spectrogram images.

After validation, the host-side model is saved before any conversion for edge deployment.

Workflow Step 4 & 5: Model Conversion

Conversion to TensorFlow Lite (.tflite)

- The validated Keras model is converted into a .tflite FlatBuffer using the TFLite Converter API.
- Post-training quantization can be enabled to reduce model size and improve inference speed on edge hardware.

Compilation to kmodel

- The Kendryte K210's KPU requires a vendor-specific .kmodel format.
- The .tflite model is compiled using the nncase tool to produce the final kmodel artifact, ready for deployment.

Workflow Step 6: Edge Device Deployment

Setup:

- Install appropriate firmware (MaixPy or C SDK) on the edge device.
- Copy the compiled kmodel file to an SD card or directly into the device's flash memory.

Execution:

- A simple runtime script loads the kmodel from its location.
- The script captures input, pre-processes it to match the model's expected shape, and runs inference on the KPU.
- Classification results are then emitted via serial, display, or GPIO.

CONCLUSION AND FUTURE WORK

Conclusion & Keys Take Away

- ✓ This work presented an end-to-end, reproducible pipeline for on-device AMC using open-source hardware and software.

- ✓ The workflow successfully converts raw RF signals into an optimized kmodel for inference on a resource-constrained Kendryte K210 microcontroller.

- ✓ **Key Achievement:** This approach establishes a verifiable pathway from signal acquisition to on-device classification, addressing the memory and operator limitations of edge hardware.

- ✓ **Impact:** It makes spectrum intelligence accessible for a wide range of IoT deployments, enhancing local decision-making, reducing latency, and conserving network bandwidth.

Future Work

- **Port to Ratified RVV 1.0:** Migrate the workflow to a newer RISC-V platform with the official 1.0 Vector Extension to quantify performance gains from the advanced programming model.
- **Comparative Benchmarking:** Compare the platform's performance against other edge AI accelerators like the Google Coral TPU or NVIDIA Jetson Nano.
- **Advanced Model Architectures:** Explore efficient models like MobileNets or SqueezeNets to improve accuracy while maintaining low latency.
- **Expanded OTA Dataset:** Collect and train on a more diverse Over-the-Air dataset (with QPSK, GMSK, 16-QAM, etc.) to improve robustness in real-world RF conditions.
- **Explore New Applications:** Use the validated platform for other signal intelligence tasks

ACKNOLEDGEMENTS AND THANKS

Acknowledgments

The authors would like to thank the French South African Institute of Technology (FSATI), Cape Peninsula University of Technology, Cape Town, South Africa for funding this research.

Thank You


Thank You
Merci
Dank Je
Obrigado
Dyakuyu
Salamat
Arigato
Spasibo
Grazie
Kiitos
Juspaxar
Mamana
Multumesc
Asante
Vinaka
Mochchakkeram
Chokrane
Kia Ora