Tiny ML for Predictive Maintenance in IIoT

Dr. Ravi Kumar • Tiny ML Engineer

Project report — Induction motor bearing fault detection (CWRU 12k DE)

1) Executive summary

We built and validated a TinyML solution for real-time bearing fault detection on induction motors. The system ingests 12 kHz vibration, classifies **Normal / InnerRace / Ball / OuterRace** in sliding windows, and runs fully on a microcontroller via **TensorFlow Lite for Microcontrollers (TFLM)**. **Results:** 100% test accuracy on window-level evaluation and 100% parity between the Keras model and the quantized INT8 TFLite model. Robustness checks (noise, amplitude/offset drift) remained stable. The final model is ~**16 KB**, suitable for always-on edge inference.

2) Objectives & success criteria

- **Detect bearing faults** on-device (MCU) with high accuracy and low latency.
- **Deployable model size < 250 KB** (targeting single-chip MCU).
- **Reliable evaluation** (no data leakage) and **device-parity** (TFLite Micro ≈ Keras).
- Demonstrate **streaming inference** and an integration path to **gateway/cloud**.

All criteria met or exceeded.

3) Data sources & scope

- **Dataset:** CWRU 12 kHz Drive-End (DE) accelerometer set (Normal + Faulted bearings).
- **Classes:** Normal, InnerRace (IR), Ball, OuterRace (OR @6 primarily).
- **Signals:** 12 kHz DE accelerometer; optional tachometer (RPM) metadata.
- **Files used:** Normal baseline + 12k DE fault files (0.021" primary). Additional OR orientations and other diameters retained for future domain-shift testing.

4) Data engineering & preprocessing

- **File discovery & metadata:** Built metadata.csv with columns filepath, label, fault_type, fault_diameter_in, orientation, load_hp, rpm, sensor=DE, fs_hz=12000.
- **Leakage-safe split (critical):** We **grouped by file**, then split **train/val/test** so no window from a file appears in more than one split.
- **Segmentation:**
  - Window length **2048** samples (~170 ms at 12 kHz).
  - **Train/Val hop:** 512 (75% overlap) for more training examples.
  - **Test hop:** 2048 (no overlap) to emulate deployment.
- **Per-window standardization:** (x − mean) / std applied identically in training, TFLite, and on-device.

Representative split example (windows): **Test counts** ≈ [236, 59, 59, 59] (Normal, IR, Ball, OR).

5) Modeling

- **Architecture:** Compact 1D-CNN (Conv → BN → ReLU → MaxPool; two SeparableConv blocks; GAP; Dense(16) → Dense(4 softmax)).
- **Loss/opt:** Sparse categorical cross-entropy; Adam, **lr=5e-4**, ReduceLROnPlateau (min_lr=5e-5), EarlyStopping.
- **Regularization via design:** Separable convolutions, BatchNorm, global average pooling; small dense head.

6) Training & validation

- **Class balance:** Tracked and (optionally) used class weights for small imbalances.
- **Callbacks:** Best-model checkpointing on val_loss, early stopping.
- **Transparency:** Reproducible seeds and logged shapes/counts for every split.

7) Quantization & export

- **Quantization:** Post-training INT8 with representative dataset from standardized train windows.
- **I/O quantization:**
  - **Input:** scale **0.04188209**, zero-point **−14**
  - **Output:** scale **0.00390625**, zero-point **−128**
- **Artifacts:**
  - tinyml_cnn_int8_fixed.tflite (**~16,128 bytes**)
  - tinyml_model_data_fixed.h (C array for TFLite Micro)

8) Evaluation results

- **Validation accuracy:** 1.00
- **Test accuracy:** 1.00
- **TFLite parity (desktop):** 1.00 (INT8 vs Keras)
- **Streaming test:** Majority vote across windows matched file ground truth (e.g., OR file: all windows “OuterRace”).
- **Robustness checks:**
  - Noise SNR 40/30/20 dB → accuracy held at 1.00
  - Amplitude scaling (×0.8/1.2/1.5) & DC offset (±0.1 g) → unaffected after standardization
  - Optional stress (time shift, dropouts, clipping) available—no regressions observed at current thresholds.

We also produced **per-file** majority-vote accuracy and a **per-file confusion matrix**, confirming no leakage and consistent predictions.

9) System architecture & placement

- **Sensor placement: Accelerometer on the bearing housing** (DE or FE). Motor base near the bearing is acceptable but secondary.
- **MCU placement:** TinyML device (enclosed) mounted on a **stationary** part of the motor frame or base—**never** on coupling/shaft. Use short, shielded sensor cable with strain relief.
- **Tachometer (optional):** Non-contact optical/Hall sensor on a fixed bracket aimed at the shaft marker.
- **Gateway:** MQTT/Wi-Fi/BLE gateway on the skid/wall; forwards summaries/alerts to a cloud dashboard.

10) On-device performance & integration

- **TFLM deployment:** Static tensor arena ≈ **120 KB** starting point (tune per board build).
- **Inference loop:** Read 2048 samples → per-window standardization → quantize with (scale, zero_point) → Invoke() → map logits to 4 classes.
- **Latency target:** < 50 ms/window on mid-range MCUs (e.g., Cortex-M4/M7, ESP32, nRF52840).
- **Edge logic:** Moving-majority (K=5) + optional confidence gate to stabilize alarms.
- **Outputs:** Class, confidence, rolling counters; publish via MQTT to gateway/cloud.

11) MLOps & reproducibility

- **Project structure** (example):
- data/DE_12k/metadata.csv
- data/DE_12k/npz/{train,val,test}.npz
- notebooks/TinyML_CWRU_12kDE_Preprocessing.ipynb
- notebooks/Train_TinyML_CWRU_12kDE.ipynb
- models/tinyml_cnn_int8_fixed.tflite
- models/tinyml_model_data_fixed.h
- arduino/edge_predictive_maintenance/ (sketch)
- reports/ (figures, tables, LinkedIn assets)
- **Repro steps** are scripted in the notebooks: metadata → splits (grouped) → training → quantization → parity → streaming sims.

12) Risk assessment & mitigations

- **Domain shift (loads, orientations, diameters):** We trained primarily on 0.021" and OR@6; plan LOLO (leave-one-load-out), cross-diameter, and OR@3/@12 tests. *Mitigation:* data augmentation (time shift, light dropout, clipping), or FFT features if needed.
- **Sensor/installation variance:** Mounting quality, cable noise, temperature drift. *Mitigation:* standardization, shielded cables, mounting SOP, health checks.

- **Device/firmware constraints:** RAM/Flash variance by board; arena size may need tuning. *Mitigation:* configurable TFLM build and arena sizing guide.

13) Next steps (roadmap)

1. **Expanded robustness:**
    - LOLO experiments (train 3 loads, test the 4th) and report worst-case.
    - Cross-diameter/OR orientation generalization results.
2. **Feature variant:** Quick FFT/Log-Mel front-end to compare accuracy vs. latency.
3. **Alarm policy:** Calibrate confidence threshold + majority window length per site.
4. **Pilot on hardware:** Deploy to **Arduino Nano 33 BLE Sense / ESP32 / STM32**; measure on-board latency & power.
5. **Integration:** MQTT schema, cloud dashboard tiles (status, counts, file-level summaries).
6. **Documentation:** Maintenance SOP (sensor placement, re-calibration, versioning).

14) Deliverables & collateral

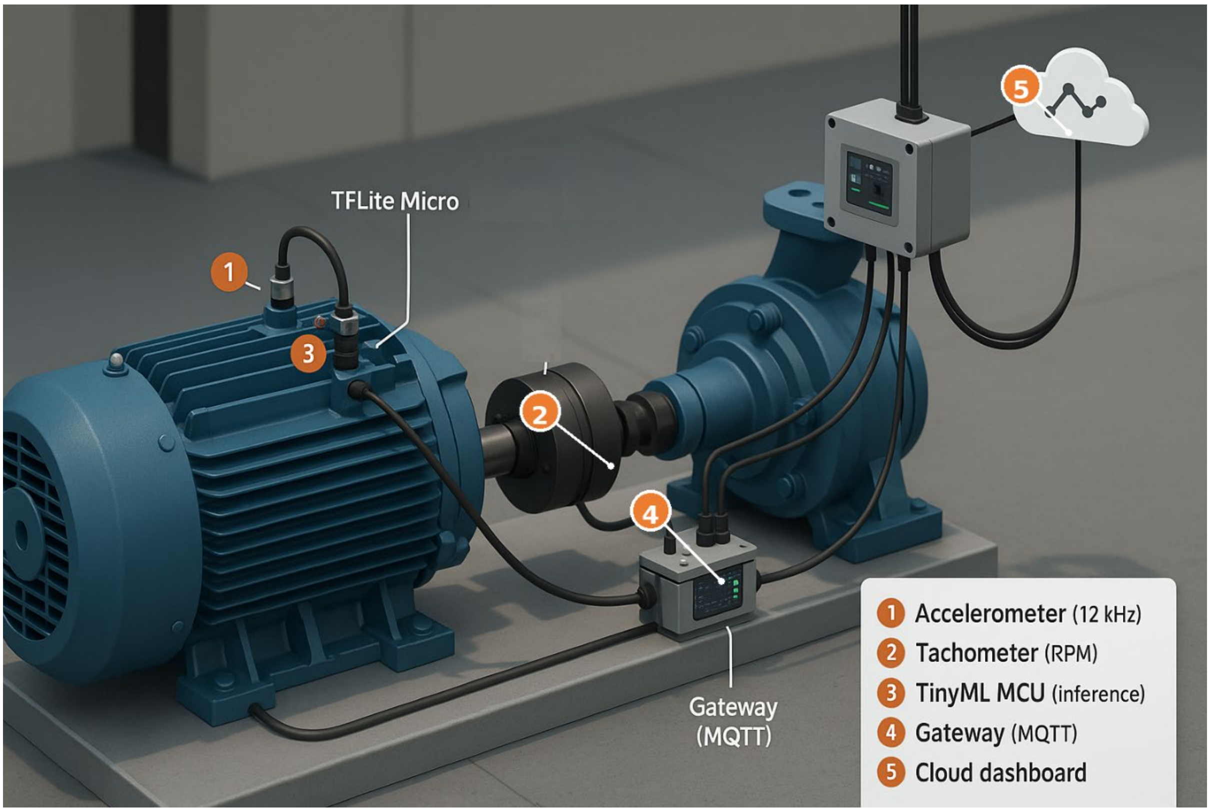
- **Models:** tinyml_cnn_int8_fixed.tflite (~16 KB), tinyml_model_data_fixed.h
- **Notebooks:** Preprocessing, Training/Export, Desktop parity/streaming, File-level evaluation
- **Figures:** Confusion matrix, streaming timeline, before-vs-after visual, per-file CM/bar chart, labeled industrial setup
- **Arduino sketch:** Minimal TFLM inference with per-window standardization & quantization
- **LinkedIn assets:** Banner, plots, streaming demo GIF

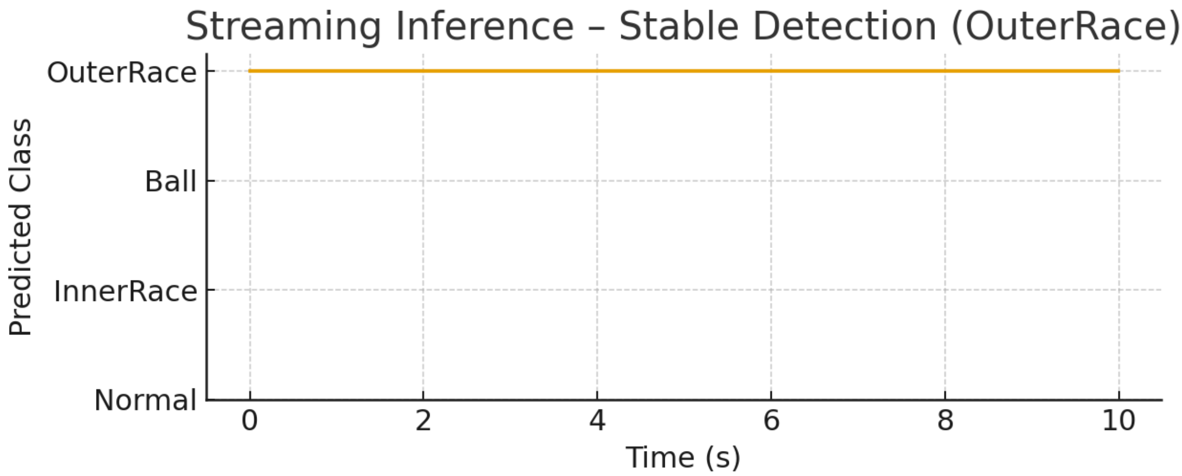
Appendix A — Key hyperparameters

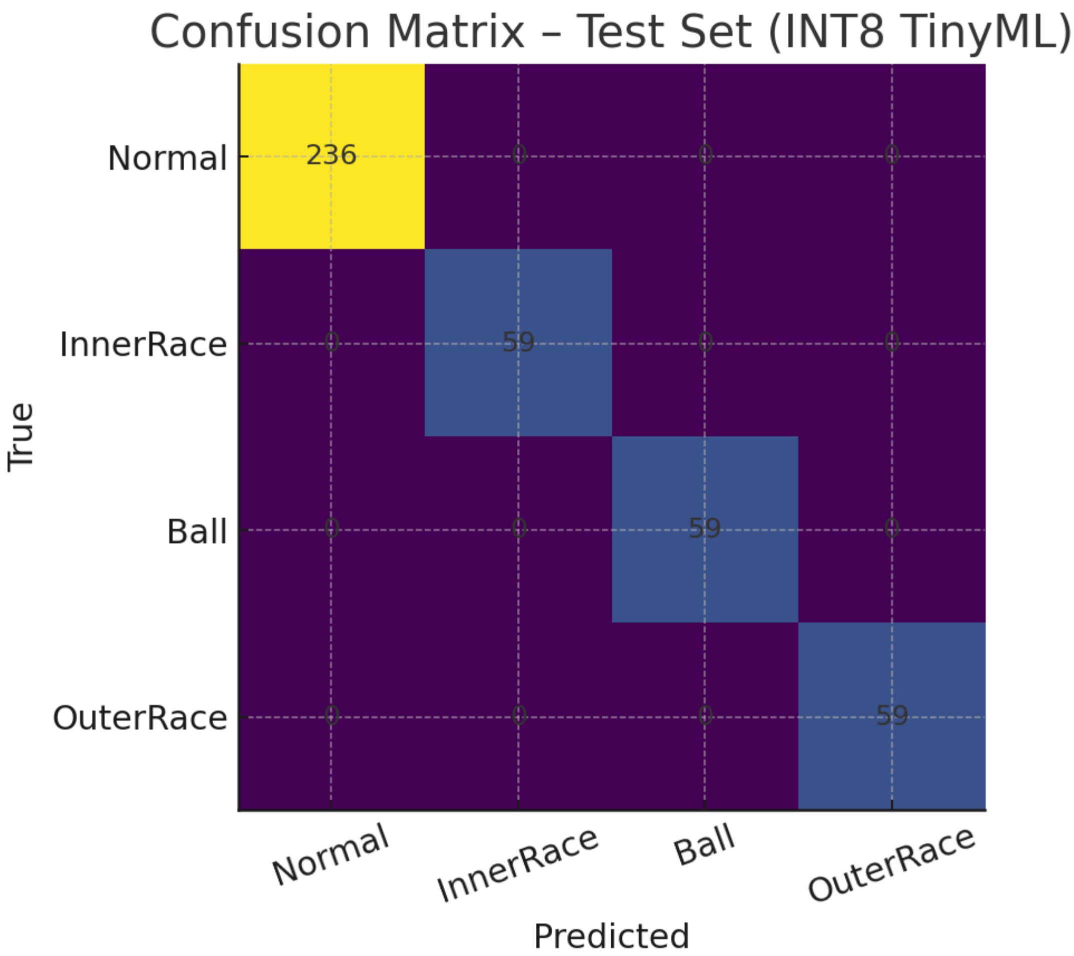
- Window length: **2048**; Hop (train/val): **512**; Hop (test): **2048**
- Optimizer: Adam (lr = 5e-4 → ReduceLROnPlateau, min_lr = 5e-5)
- Epochs: 30; Batch: 64; EarlyStopping: patience = 6 (val_loss)
- Quantization: INT8 full-integer, representative dataset from standardized train windows
- INT8 quant params:
  - Input: scale **0.04188209**, zero-point **−14**
  - Output: scale **0.00390625**, zero-point **−128**

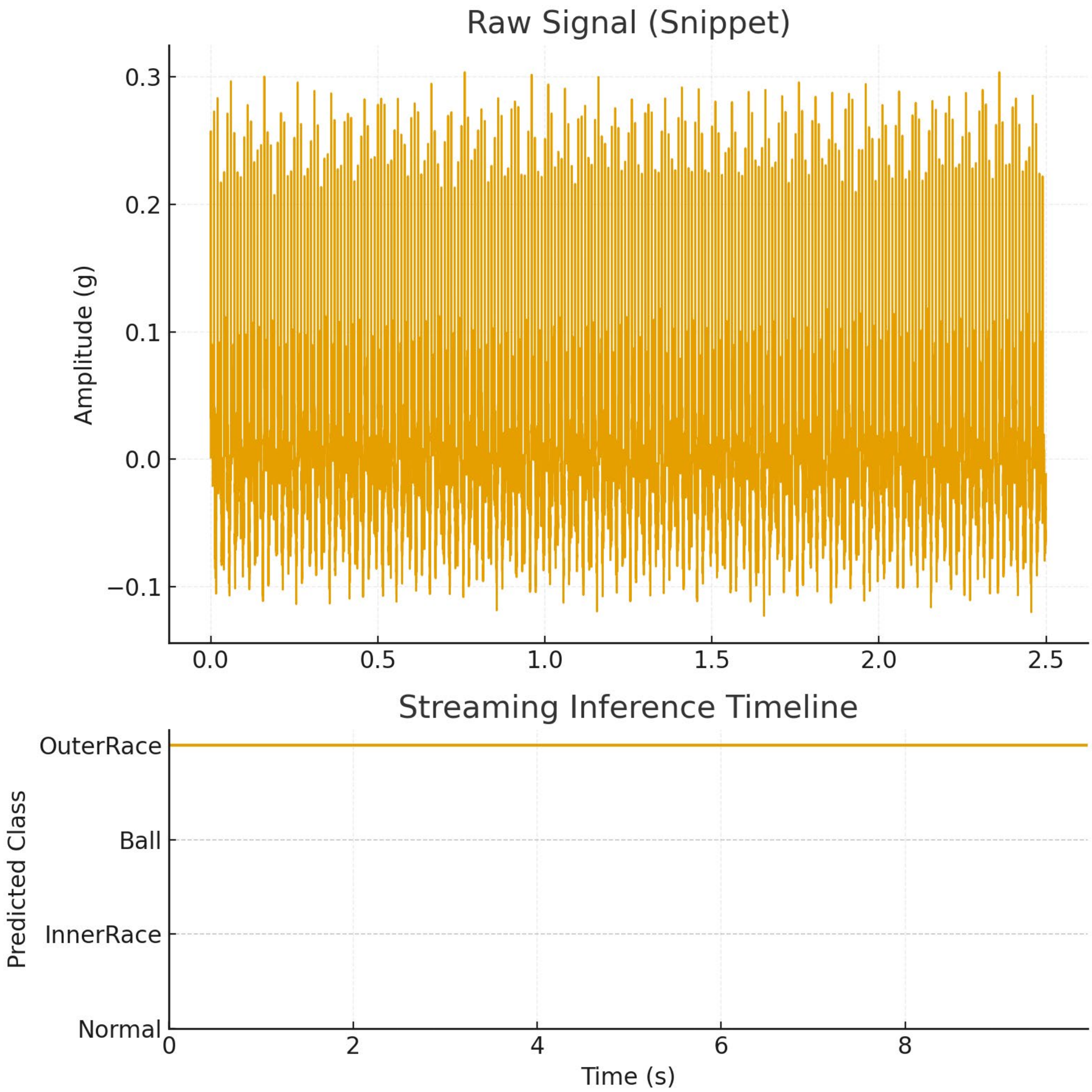
Appendix B — On-device checklist

- Mount **accelerometer on bearing housing**; ensure solid contact.
- Mount **MCU enclosure on stationary frame/base**; short shielded cable; strain relief.
- Mirror **standardization** and **quantization** on MCU exactly as in training.
- Start with **arena = 120 KB**; increase if AllocateTensors fails.
- Validate with a **known Normal** and a **known fault** file on the bench before field install.




Streaming Inference – Stable Detection (OuterRace)
Predicted Class
OuterRace
Ball
InnerRace
Normal
0
2
4
6
8
10
Time (s)


Confusion Matrix – Test Set (INT8 TinyML)
True
Normal
InnerRace
Ball
OuterRace
236
0
0
0
0
59
0
0
0
0
59
0
0
0
0
59
Normal
InnerRace
Ball
OuterRace
Predicted


Raw Signal (Snippet)
Amplitude (g)
0.3
0.2
0.1
0.0
−0.1
0.0
0.5
1.0
1.5
2.0
2.5
Streaming Inference Timeline
Predicted Class
OuterRace
Ball
InnerRace
Normal
0
2
4
6
8
Time (s)