Predictive Maintenance with an Arduino-based LoRa solution

Sebastian Romero
s.romero@arduino.cc
What is Arduino?

Arduino is an open electronics platform based on easy-to-use hardware and software.

- Started 2005 as a low cost prototyping solution.
- Arduino Pro brings the knowledge and the experience that we collected over the years to the professionals.
The Arduino Platform

- A complete platform to simplify developing hardware solutions
- Thousands of libraries to support almost any sensor / actuator
- Huge worldwide community to provide support and inspiration
Maintenance vs Repair

**Repair**: Mending something that is already broken. Usually urgent for time sensitive processes.

**Maintenance**: Preventive measures to keep a machine functional for as long as possible. → Failure prevention

Scheduled vs **predictive** maintenance
Predictive Maintenance

- Techniques to **analyse** the condition of hardware and **predict** when maintenance should be performed **before** an inconvenient moment comes.

- Replacement for scheduled maintenance.
Predictive Maintenance Benefits

- Save cost on repair of broken equipment
- Shorter outage (if any)
- Optimized use of maintenance staff
- Optimized spare part stocking
- Technology agnostic (mostly)
- No hardware modification → no warranty issues
LoRa Based Predictive Maintenance

1. Choose the Hardware You Need
2. Make It Smart
3. Connect It to the Cloud
4. Inspect the Data
1. Choose the Hardware You Need
MKR WAN 1310

- SAMD21 Cortex-M0+ 32bit low power ARM MCU 48 MHz
- CMWX1ZZABZ LoRa Module
- Battery connector
- ATECC508A Secure Element
- Carrier frequency: 433/868/915 MHz
Portenta H7 + Vision Shield

- STM32H747 dual Cortex ®
  480 + 240 MHz
- Murata 1DX dual WiFi / Bluetooth
- LiPo battery charger
- Murata CMWX1ZZABZ LoRa Module
- Himax HM-01B0 Lo-Power camera
- 2 microphones (directional sound)
- SD-Card connector
Nicla Sense ME

- Cortex-M4 nRF52832
- ANNA B112 Bluetooth module
- 6 axis IMU, pressure sensor, magnetometer, gas sensor
- LiPo battery charger
2. Make it Smart 🧠
Machine Learning on MCUs

Augment the intelligence of billions of appliances

- Low-cost hardware: Easily embed in everyday products
- Low power: Works with a battery (portable)
- No internet connection required
- Data stays on device (privacy)
How: Sound & Machine Learning

Use machine learning to detect anomalies in sound.

- E.g. the rattling sound of a broken dishwasher.
- E.g. the sound of broken glass in a factory
- Accuracy influenced by background noise 🔄
How: Machine Vision

Use machine vision to detect visual anomalies.

- E.g. broken light bulbs
- E.g. distorted saw blade
- Accuracy influenced by lighting 💄
How: Gas & Machine Learning

Use machine learning to detect anomalies in gases.

- E.g. detect wine going bad while ageing in a barrel
- E.g. leaking gases in a factory
How: Vibration & Machine Learning

Use machine learning to detect anomalies in vibrations (focus of the demo).

- e.g. worn out drill bits (life expectancy estimation)
- e.g. displaced washing machine drum
3. Connect it to the Cloud 🌓
IoT Cloud meets The Things Network

- MKR WAN boards can connect to IoT Cloud via a TTN LoRaWAN® backend 🌞

- Automatic configuration of the TTN app.
Why LoRa is a Good Choice

- Resistance to noise
- Cover wider area (even public infrastructure)
- Simplified wiring (one gateway connected to internet)
- Low power (can operate with batteries/solar panel)
- No network configuration (no IPs etc.)
- Outdoor usage (e.g. predictive maintenance in the nature)
4. Inspect the Data 🧐
IoT Cloud Dashboard

- Inspect hardware status
- Read sensor data
- Track sensor history
- Send messages
  (e.g. alert maintenance staff)
- Trigger actions
  (e.g. turn device off)
APPLICATION EXAMPLE

Detecting Vibration Anomalies
Predict Mechanical Failure Through Vibration

- **Analyse** vibration patterns
- **Predict** if a machine may fail in the near future.
- Perform **maintenance** before it fails.
Rule Based vs Machine Learning

- The simple **intensity** or **frequency** of a vibration anomaly could be detected with a rule based approach...
- ...but vibration patterns are not exactly same every time.
- ML can easily deal with these variations. 🏋️‍♂️
Goal: IoT Cloud Dashboard

- Track device status
- Inspect failure history
- Intervene (e.g. turn device off)
Hardware Setup for Demo

- MKR WAN 1310
- Nicla Sense ME
- ESLOV Cable
- PC Fan
- Finger 🤚
Simulate mechanical wear with a finger 👍
Training Process

PC
- ARDUINO IDE
- TRAINING SKETCH
- NICLA SENSE ME
- EI DATA FORWARDER

WEB APP
- EDGE IMPULSE
- SPECTRAL ANALYSIS
- ML MODEL TRAINING
- ML SKETCH DEPLOYMENT
Deployment

SENSOR NODE
- ARDUINO IDE
- ML SKETCH
- NICLA SENSE ME

HOST BOARD
- ARDUINO IDE
- IOT CLOUD SKETCH
- MKR WAN 1310
Detection Process

1. Read accelerometer data
2. Classify vibration
3. Send data over Ethernet to MKR WAN
4. Read data from Nicla Sense Me
5. Upload data to IoT cloud
6. Display status in dashboard
Machine Learning: Edge Impulse Studio

- Gather Raw Data
- Process Data
- Extract Features
- Train ML Model
1. Gather Raw Data

- On-board accelerometer to gather vibration data
- Motion on 3 axes
- Orientation matters

- Simplification: Magnitude of motion vector

\[ |M| = \sqrt{x^2 + y^2 + z^2} \]
1. Gather Raw Data

- On-board accelerometer to gather vibration data
- Motion on 3 axes
- Orientation matters

- Simplification: Magnitude of motion vector

\[ |M| = \sqrt{x^2 + y^2 + z^2} \]
2. Process Data

Window Size

- Defines how many ms of sensor data should be considered for a classification.
- Depends on vibration pattern
- For constant vibration a small window may suffice.
2. Process Data
Spectral Analysis

- Filter relevant frequency
- Vibration characteristics
- Find peaks
3. Extract Features

- Unique characteristics
- Separation of classes
- Find bad training data
4. Train ML Model

- Learns based on provided vibration samples
- Adjust learning cycles as needed
- Watch out for overfitting ⚠

Last training performance (validation set)

<table>
<thead>
<tr>
<th>ACCURACY</th>
<th>LOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>93.5%</td>
<td>0.77</td>
</tr>
</tbody>
</table>

Confusion matrix (validation set)

<table>
<thead>
<tr>
<th></th>
<th>NORMAL</th>
<th>VIBRATING</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORMAL</td>
<td>91.4%</td>
<td>8.6%</td>
</tr>
<tr>
<td>VIBRATING</td>
<td>3.7%</td>
<td>96.3%</td>
</tr>
<tr>
<td>F1 SCORE</td>
<td>0.94</td>
<td>0.93</td>
</tr>
</tbody>
</table>
Thank you!