Advanced Anomaly Detection with Edge Impulse

José Antonio Bagur Nájera

Universidad del Valle de Guatemala

Scientific Use of Machine Learning on Low-Power Devices 2023

Agenda

In this presentation we are going to cover the following:

- Define what is know as advanced anomaly detection.
- Show some use cases of advanced anomaly detection.
- Hands-on with Edge Impulse and Arduino®.

About me

Hello, my name is José and I am from Guatemala. I work at Universidad del Valle de Guatemala (UVG) as a lecturer in the Mechanical Engineering Department; I am also involved in the Aerospace Laboratory at UVG. In addition to my work at the university, I'm part of the Arduino team, where I contribute as a content creator for Arduino Pro.

Outside my professional life, I love to explore the beautiful mountains and volcanoes in Guatemala through hiking. I am also an avid coffee drinker and I enjoy reading and listening to music in my free time.

About me

Anomaly detection

Anomaly detection is the process of **identifying unusual patterns, events, or data points** that deviate significantly from what is considered normal or expected.

Anomaly detection is a crucial and challenging task that has gained importance across various fields, including **finance**, **cybersecurity**, **healthcare** and **industrial equipment monitoring** (we are going to focus on this last one).

Anomaly detection (2)

Why is anomaly detection so important in industrial equipment?

- Early detection of equipment anomalies.
- Equipment efficiency improvement.
- Safer working environment.
- Predictive maintenance.

Anomaly detection use cases

Some use cases of anomaly detection in industrial equipment are the following:

- Vibration analysis and bearing fault detection.
- Temperature monitoring and overheating detection.
- Pressure monitoring in pipelines and vessels.
- Energy consumption analysis.
- Acoustic emissions and ultrasonic testing.

Hands-on will focus on **vibration analysis**.

Hands-on with Edge Impulse and Arduino

In the hands-on we are going to develop an anomaly detection model with Edge Impulse and an Arduino supported board.

The model will be a **fan anomaly detection model** (vibration)

Hands-on with Edge Impulse and Arduino (2)

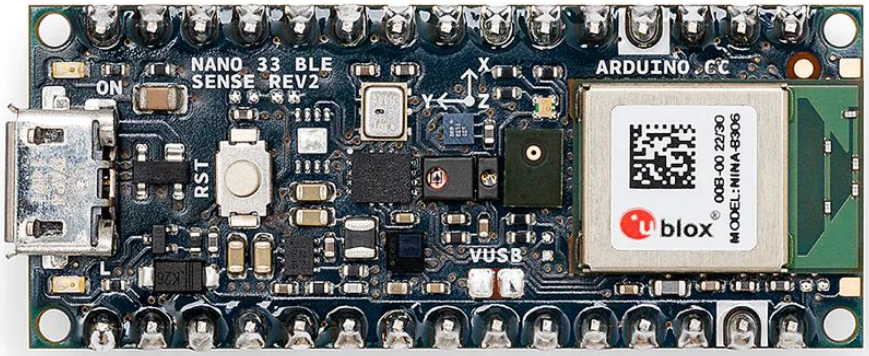
We will detect vibrations in a fan using an accelerometer placed on the top of a fan.

For demonstration purposes, we will use a **cheap computer fan** that can be found in any electronics store (or electronics dumpster 😅). This example should also work in real-life industrial environments.

Hands-on with Edge Impulse and Arduino (3)

The Arduino board we are going to use is the **Nano 33 BLE Sense**.

The Nano 33 BLE Sense is a compact and versatile development board based on the nRF52840 microcontroller from Nordic Semiconductor.



The board was designed for low-power applications and features Bluetooth Low Energy (BLE) connectivity.

Hands-on with Edge Impulse and Arduino (4)

Key features of the Nano 33 BLE Sense board:

- Microcontroller: **32-bit ARM Cortex-M4** processor running at 64 MHz (nRF52840), **1 MB of Flash** and **256 kB of SRAM**.
- Connectivity: BLE, allowing easy communication with other BLE-enabled devices.
- **11 onboard sensors**:
  - 9-axis Inertial Measurement Unit (IMU).
  - Temperature, humidity and barometric pressure.
  - Digital microphone.
  - Gesture, proximity, and ambient light.
  - Color sensor.

Hands-on with Edge Impulse and Arduino (5)

Important note: there are currently two versions of the Nano 33 BLE Sense board, **Rev1** and **Rev2**. What is the difference between the boards?

- Different IMU's.
- Different temperature and humidity sensors.
- Different microphones.

This means also means **different libraries to work with the board's onboard sensors and different official Edge Impulse firmware. Currently Edge Impulse is working on adding dedicated "official firmware" to Rev2 board**.

Hands-on with Edge Impulse and Arduino (6)

For using this board with TinyML, we need to install the following:

- Arduino IDE 2.0+
- Core and libraries:
  - Arduino Mbed OS Nano Boards
  - TensorFlow Lite
  - For Rev1, Arduino_LSM9DS1

Hands-on with Edge Impulse and Arduino (7)

Follow these instructions to configure the Nano 33 BLE Sense board with Edge Impulse.

Basically, the **configuration can be divided into four main steps**:

1. Edge Impulse CLI installation.
2. Edge Impulse Studio account registration.
3. Arduino CLI installation.
4. Latest Edge Impulse firmware upload to the board.

Installation process can be tricky, don't hesitate to send your questions to the **network Discord server**!

Hands-on with Edge Impulse and Arduino (8)

The **number of samples required for an anomaly detection model** depends on several factors. **There is no one-size-fits-all answer**, but some general guidelines to determine the appropriate sample size are the following:

- **Data complexity**: with complex patterns, larger sample size.
- **Type of anomaly detection technique**: supervised methods require more samples than unsupervised methods.
- **Model performance**: more data is not always better.
- **Rarity of anomalies**: a larger sample size is needed if anomalies are rare events.

As a starting point, **consider collecting at least several hundred to a few thousand samples** to train an anomaly detection model.

Hands-on with Edge Impulse and Arduino (9)

A general **workflow** we can follow for developing an anomaly detection model is the following:

1. **Data collection**: aim for at least several hundreds of samples per class to start with.
2. **Data preprocessing**: ensure data is suitable for training a machine learning model.
3. **Model training**: split your data into training, test and validation sets. **A common split ratio is 70% for training, 15% for testing and 15% for validation**.
4. **Model evaluation**: understand how well the model is performing.
5. **Model deployment**: deploy your model to your target hardware.

Hands-on with Edge Impulse and Arduino (10)

Comments about data collection

For the hands-on, we will two different types of samples or **classes**: **”idle”** and **”nominal”**. Since this is a simple binary classification problem, the required sample size may be smaller compared to complex scenarios.

Other classes like **“broken”** or **“obstructed”**, for example, are not necessary but can be added to improve the model significantly.

Hands-on with Edge Impulse and Arduino (11)

Comments about data collection (2)

In this example, we are going to collect **4 minutes of data** ("not great, not terrible"). In this case, we need to allocate approximately the following amounts of data for each set:

- **Training**: 2.8 minutes (70% of 4 minutes).
- **Testing**: 0.6 minutes (15% of 4 minutes).
- **Validation**: 0.6 minutes (15% of 4 minutes).

Hands-on with Edge Impulse and Arduino (12)

Comments about model training

In this example we are going to use an **Spectral Analysis** processing block, which is great for analyzing repetitive motion, focusing only in the accelerometer readings.

Edge Impulse can also combine different sensor readings or outputs to get a new outcome of collected data.

Hands-on with Edge Impulse and Arduino (13)

Comments about model training (2)

For this example we are going to use **two learning blocks**: a **Classification** block which learn patterns from data and an **Anomaly Detection** (K-means) block which can recognize unknown states.

For the Anomaly Detection block is important to calculate the feature importance since this can tell us which aspects are contributing to the original classes (so we can focus on those features).

Hands-on with Edge Impulse and Arduino (13)

Comments about preprocessing data

We can achieve a more accurate model in less time by **standardizing** (i.e. to a mean of 0 and unit variance) and decorrelating input data. Great paper about this here.

The standardization process assumes our data is normally distributed (i.e. Gaussian), if your data does not follow a Gaussian distribution, we should perform a normalization.

Take a look into this Google Colab notebook from Shawn Hymel and Edge Impuse.

Hands-on with Edge Impulse and Arduino (14)

The complete anomaly detection demonstration can be found here (it is **public**, you can clone it to you Edge Impulse account and use it freely).

Conclusions

Anomaly detection is a powerful technique for identifying abnormal behavior in industrial equipment, helping to prevent costly failures, improve efficiency and enhance safety in the workplace.

The combination of Edge Impulse and Arduino provides a practical and accessible platform for implementing and deploying real-world TinyML applications. These tools enable rapid prototyping, streamlined data collection, and seamless integration with various supported boards, sensors and actuators.

Thank you for your attention

Let’s keep in touch!

- **#icpt-workshop** at the network Discord server
- Email: **jabagur@uvg.edu.gt**