

Sensor Data and Anomaly Detection

Student guide, Hands-On AI for High School, grades 9 to 12

You are looking after a spinning machine in a factory, such as a large motor or pump. It has three sensors: one for vibration, one for temperature and one for speed (RPM). The sensors took 500 readings over 50 seconds. Somewhere in the data, a part called a bearing starts to fail. Your task is to find the failure without being told where it is.

Before you start

- You need a device with a web browser and a Google account.
- Open the course page, github.com/aalbata/ai-labs-for-high-school, and click the Open in Colab badge next to “Sensor data and anomaly detection”.
- Choose File, then Save a copy in Drive, so that your answers are kept.
- The sensor data is already inside the notebook, so there is nothing to upload.
- Click the play button on the left of each code cell and wait until it finishes before you run the next one.

Step 1: Load the data

Run the first cell. It loads 500 readings of time, vibration (in g), temperature (in degrees Celsius) and RPM, and prints a summary of each signal.

Read the smallest, largest and average value of each signal. Does anything already look unusual? Write your first thoughts in Checkpoint 1.

Step 2: Look at the three signals

Run the plot cell. You will see one graph per signal across the 50 seconds. Healthy signals look steady, and a failure shows up as a change in the pattern. Study all three graphs for at least a minute before you answer.

Answer in Checkpoint 2:

- At about what time does the trouble start? Write your best guess.
- Which signal gave the clearest warning: vibration, temperature or RPM?
- Would you stop the machine based on these graphs? Why?

Step 3: Zoom in

This cell compares two time windows, one normal and one where you suspect a problem, and measures the standard deviation of the vibration in each. The standard deviation measures how jumpy a signal is. A steady signal has a low value and a jumpy one has a high value.

You can change `WINDOW_A_START` and `WINDOW_B_START` and run the cell again to look at other parts of the signal.

- How many times jumpier is the problem window than the normal window?

Step 4: Detect the problem automatically

This is the main algorithm. It measures how jumpy the vibration is over a sliding window of the last 30 readings. When the jumpiness rises above the threshold, set to 2.5 times the normal level, it flags a problem. The output shows when the algorithm first raised the alarm, next to the true start of the fault ($t = 32.0$ s).

Answer in Checkpoint 3:

- How close was the algorithm's first alarm to the true start of the fault?
- How close was it to your guess from Step 2?
- What could happen to a real machine during the delay before the alarm?

Step 5: The sensitivity trade-off

You can change `THRESHOLD_FACTOR` in the Step 4 cell and run it again. The Step 5 cell then compares five thresholds at once: 1.5, 2.0, 2.5, 3.5 and 5.0 times the normal level. The table shows the first alarm, the delay and the number of false alarms for each.

Answer in Checkpoint 4:

- What happens to the false alarms as the threshold gets lower?
- What happens to the delay as the threshold gets higher?
- Which threshold would you choose for a coolant pump in a nuclear power plant, and which for an office air-conditioning fan? Why?

Step 6: Use all three sensors together

This cell puts each signal on a 0 to 1 scale and combines them into one score. Temperature counts double, because friction in a failing bearing makes it heat up.

Answer in Checkpoint 5:

- Did the combined score catch the problem sooner than vibration alone? By how much?
- Why does a bearing heat up when it starts to fail?
- Name two other sensors you would add to this machine, and what each would tell you.

Step 7: Compare with real systems

Run the last code cell. It shows what you built next to the predictive-maintenance systems that companies sell to factories.

- What is the most important thing a real system does that yours does not?
- What does a human engineer still need to do, even with the most advanced system?

Final reflection

Answer these in the reflection cell at the end of the notebook.

1. Explain how the rolling-standard-deviation detector works, as if you were telling a factory manager who has never taken a statistics class.

2. Your threshold was 2.5x and the fault started at $t = 32.0$ s. How much sooner could a lower threshold have caught it, and what would that cost?
1. A worker who walks past this machine every day might notice a problem before any sensor does. What might they notice, and how is that different from what the algorithm sees?
2. This machine had a worn bearing. Name two other ways a machine can break, and describe what you would expect to see in the sensor data for each.
3. A factory wants to put this system on 500 machines and text the maintenance team every time an alarm goes off. What could go wrong, and what safety checks would you add?

Words to know

Term	Meaning
Accelerometer	A sensor that measures vibration, often in g. One g is the acceleration of gravity, about 9.8 meters per second squared.
Standard deviation	A measure of how spread out a set of values is. Low means steady and high means jumpy.
Rolling window	A calculation over the most recent readings that moves forward one reading at a time.
Baseline	The normal, healthy behavior of a machine. Problems show up as changes from the baseline.
Threshold	The level above which a reading counts as a problem. Choosing it well is the key engineering decision.
False alarm	An alarm when nothing is wrong. Too many of them cause needless shutdowns, and people stop trusting the system.
Missed detection	A real problem the algorithm does not flag. It can lead to a sudden, serious failure.
Predictive maintenance	Using sensor data to predict when a machine will fail, so that it can be fixed before it breaks.

If something goes wrong

Problem	What to do
"Runtime disconnected"	Reconnect and run the cells again from Step 1. The data is inside the notebook, so nothing is lost.
"NameError: name 'vib' is not defined"	Run the Step 1 cell first. Its variables are lost when the runtime disconnects.
A plot does not appear	Make sure the Step 1 cell has run.
A cell runs but prints nothing	Look for a red error message under the cell and read its last line.