

Without fault labels: unsupervised machine learning in industry

Learn what unlabelled sensor history can and cannot tell you, how to evaluate a detector without labels, and why operating modes need attention first.

In short

- Plants usually hold plenty of sensor history but few labelled failures, so most methods learn normal behaviour and flag departures.
- An unsupervised model can say a reading is unusual, but it cannot say which fault it is; every alert needs operator review.
- Evaluate an unlabelled detector by replaying past incidents, counting alerts per week, and splitting the data by time.

01 / Why labels are scarce

A plant may hold years of sensor history, yet labelled failures inside that history are usually few. Failures are rare. Maintenance records can be incomplete, inconsistent or free text, and the root cause of an event is often never written down. Unsupervised learning refers to methods that use no target labels during training. They learn what normal operation looks like and flag departures from it. Semi-supervised methods use a few labels when they exist.

The wording matters. A model can say that a reading is unusual, but it cannot say that a failure is happening. An unusual point is not necessarily a faulty one: a different operating condition, a measurement error or a new product recipe can produce the same departure. Keep that distinction explicit in reports and in operator training.

02 / Method families

Statistical limits are the simplest family. They track each sensor against its normal range, using a mean and standard deviation or robust equivalents. They are easy to explain but blind to relationships between variables.

Distance and density methods, such as Mahalanobis distance, account for correlations between variables and measure how far a point lies from the normal cloud. Isolation trees rely on the idea that rare and different points are separated with fewer random splits. Reconstruction methods, such as PCA-based models, compress normal data into a lower-dimensional structure and reconstruct it; the part that cannot be reconstructed indicates departure. Clustering with k-means is used to find operating modes. The right family depends on the data size, the strength of relationships between variables and how much the model must be explained.

03 / Operating modes come first

A machine may not have a single normal state. A pump running at two speeds, or a line making three products, has a separate normal distribution for each mode. If you train one model without separating the modes, even a legitimate change of mode can keep triggering alarms.

Two fixes are common. One is to find the modes first, with clustering, and then build a separate model for each. The other is to include the variable that shows the operating mode, such as speed, product code or load, as an input. If no mode information exists, derive a signal from speed or load changes, locate the transition periods and remove them from evaluation. Once the modes are identified, a separate threshold for each mode gives a result that is easier to explain.

04 / Evaluating without labels

Accuracy cannot be computed without labels, so the team has to use indirect measures. The first is replaying past incidents: at what time did the model signal, and when did the maintenance team act? The second is operator review of flagged periods. The third is the number of alerts per week.

A hypothetical calculation: if a 12-week period produces 48 alerts, the average is $48 / 12 = 4$ alerts per week. If 3 of 5 known events were flagged in advance, the catch rate is $3 / 5 = 60\%$. Five events is a very small sample, so treat that figure with care. If the three advance flags came 36, 10 and 4 hours before their events, the median lead time is 10 hours. Finally, split the data by time. Shuffling rows randomly lets information from the future leak into training.

05 / Limits and misreadings

An unsupervised model does not know the fault type or the root cause. It can only say that data differ from what it saw in training. If a fault period is already in the training data, the model may treat it as normal. Drift from seasonal change, new raw material or a replaced sensor will also produce false alarms. For that reason, the output should be presented as an inspection priority, not a diagnosis.

The evaluation choice is a risk that is often overlooked. A low weekly alert count looks reassuring, but if the maintenance team cannot connect those alerts to real events, the model is not useful. Remember also that an average score can hide a poor period.

06 / Turning alerts into decisions and labels

A raw alert is not a decision. Define a triage rule: for example, notify the operator only if the threshold is exceeded in three consecutive windows, and merely log a single spike. Store the decision the operator gives for every alert, whether it was real, false or uncertain.

Over time, that feedback becomes a label set. Once there are enough verified events and a consistent labelling process, a supervised model may be worth considering. Until then, the unsupervised method is a tool for producing labels. How quickly that transition can happen depends on how consistently field decisions are recorded.

Try it

In Sentinel's sample sensor table, compare two flagged periods and write one sentence on whether either one resembles an operating-mode transition rather than a real departure.

References and online guide

[Open the online guide](#)

Unsupervised learning (Wikipedia)

Anomaly detection (Wikipedia)

scikit-learn: novelty and outlier detection

k-means clustering (Wikipedia)

Proposed pilot steps do not claim field accuracy. Synthetic examples are not measurements from a production site.