Critical manufacturing
an ASM PT company

Outlier Detection using Machine Learning

11.2

February 2026

DOCUMENT ACCESS

Public

Outlier Detection using Machine Learning

Estimated time to read: 6 minutes

This guide outlines how to create and configure an outlier detection workflow using IoT Workflows in the Data Platform.

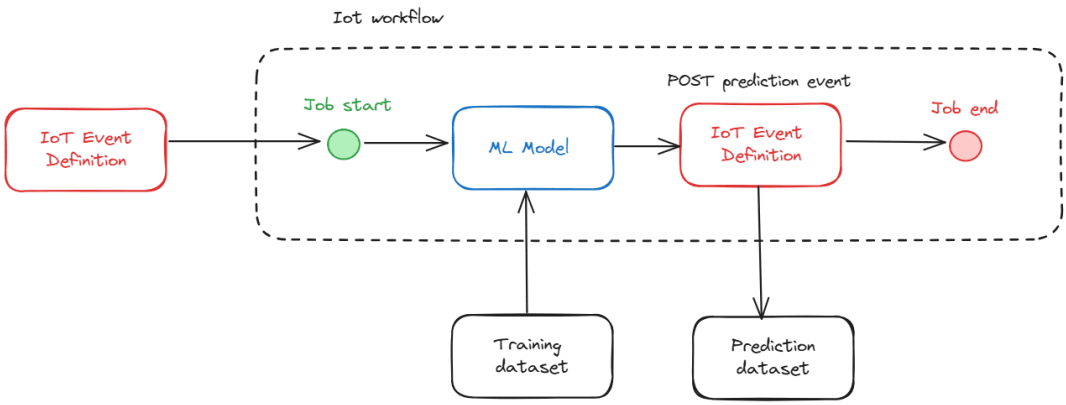
Overview

Outlier detection is a process of identifying data points that deviate significantly from the normal pattern in a dataset. These outliers often indicate potential issues such as equipment failures, anomalies in production, or irregular environmental conditions.

By setting up an outlier detection workflow, you can:

- Proactively address issues before they escalate.
- Monitor production data for unusual trends.
- Integrate alerts and actions for seamless operations.

A generic Machine Learning (ML) workflow may look like the diagram below where we can see that we need at least two tasks and two IoT Event Definitions as we have to, first, trigger the ML Model for inference and, and then to store the predictions into a Data Set. Once stored, we may visualize the data in Grafana.



ML models need a dataset to be trained. This dataset is usually composed of a training set and a test set. The training set is used to train the model, while the test set is used to evaluate the model performance. Please check on the user guide into this topic.

Preconditions

Before creating an outlier detection workflow, ensure the following:

1. **IoT Event Definitions**

   Define the events or data streams you want to monitor, such as sensor data, production metrics, or environmental readings.

2. **Access to Historical Data**

   Set up a dataset with a fair amount of high quality data to be used during machine learning experiments.

3. **Access to Data Platform ML Model**

   Ensure you have the right license to unlock ML capabilities in Data Platform.

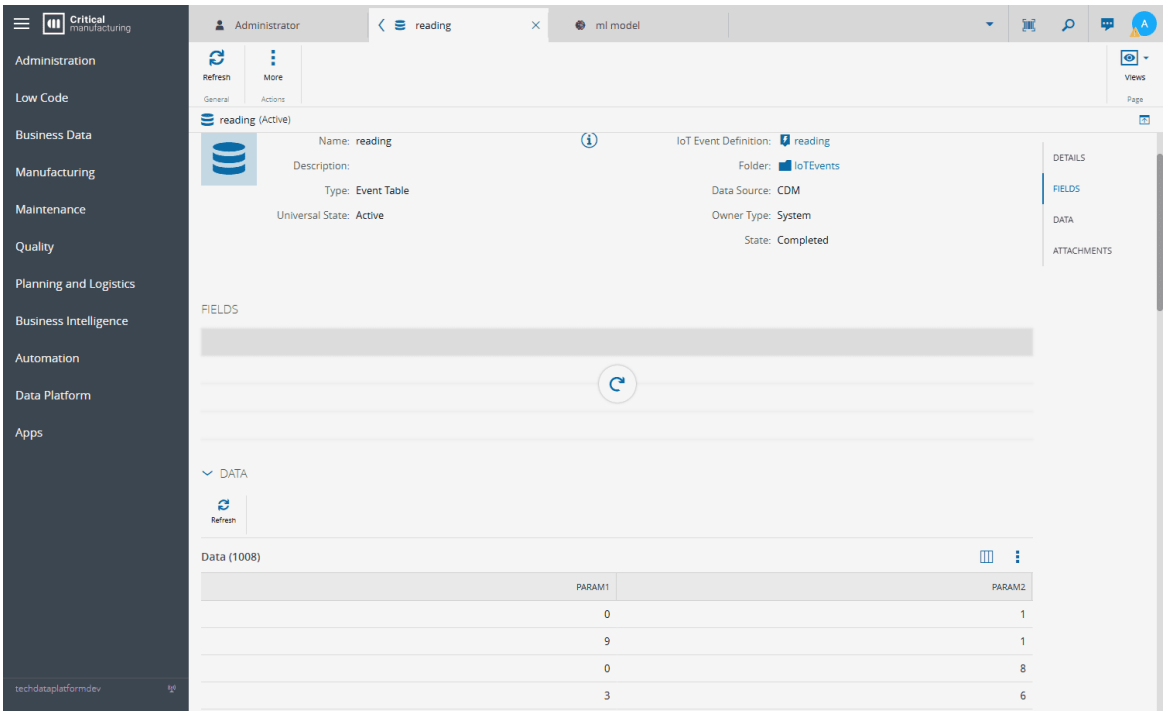
4. **Permissions and Roles**

   Verify that you have the required permissions to create and configure IoT Events / IoT Workflows.

Steps to create an outlier detection workflow

Step 1: Define the IoT Event

1. Navigate to the IoT Event Definition view.
2. Create or Select an IoT Event Definition:
   - Define the data streams or events that may seem relevant to anomaly detection (example: temperature spikes, production downtime, or sensor malfunctions). For the present scenario, we will use dummy data and we will monitor Parameter 1 and Parameter 2 for unusual behavior. For that we have created an Iot Event Definition that is called `reading`, which comprises parameters of type `Decimal`.
   - Enable the data to be stored in a **Data Set** as the events are being sent. This Data Set will be used to train our ML Model.
   - A **Data Set** will be created with the same name and can be consulted in the Iot Events folder.
   - In our exercise the **Data Set** will be called `reading`.

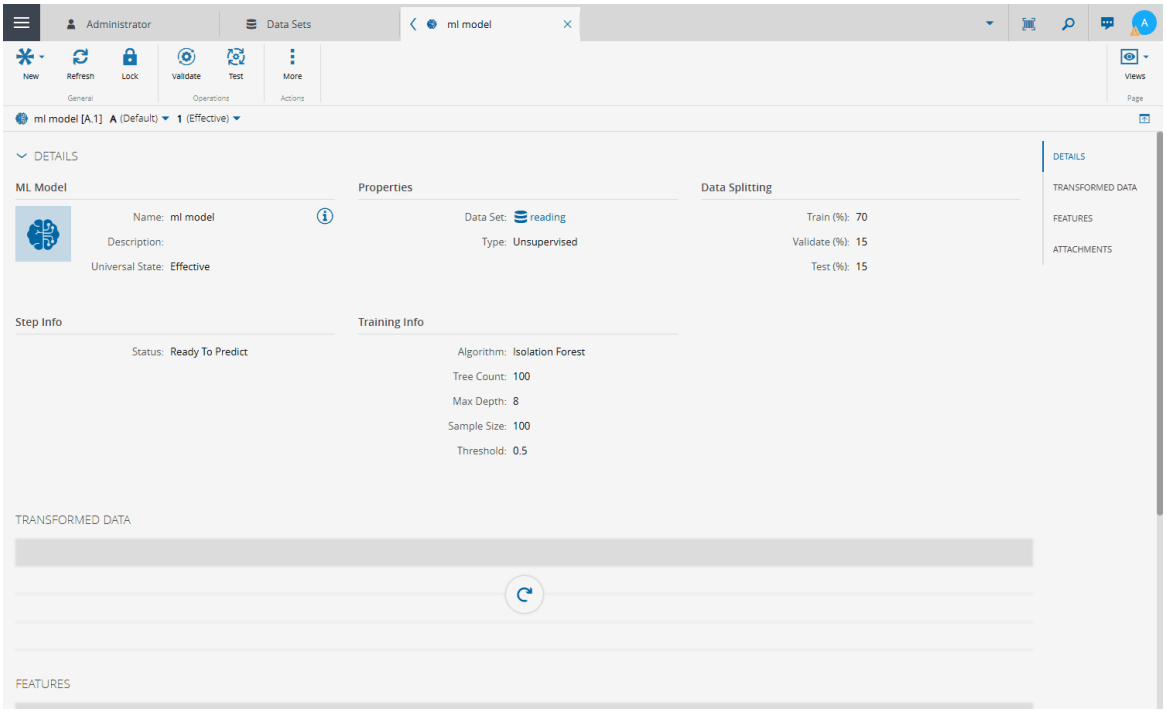


Step 2: Train an ML Model

For this step, you need a dataset with a fair amount of high quality data. For this particular use case we have a sample with two clear patterns. We need a quick way of segmenting data into two groups: normal

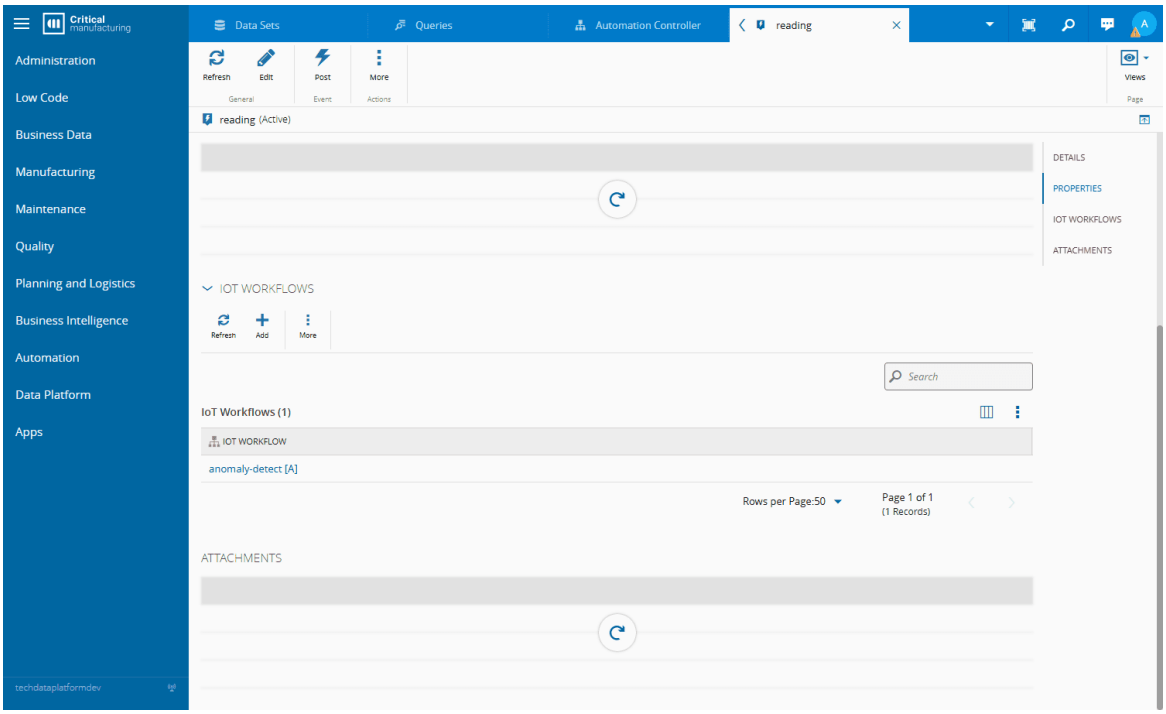
and abnormal.

For that we will use isolation forest algorithm provided by Data Platform. After it is trained, we can use the score parameter to identify anomalies by applying thresholds either within the workflow or in the Grafana dashboard.



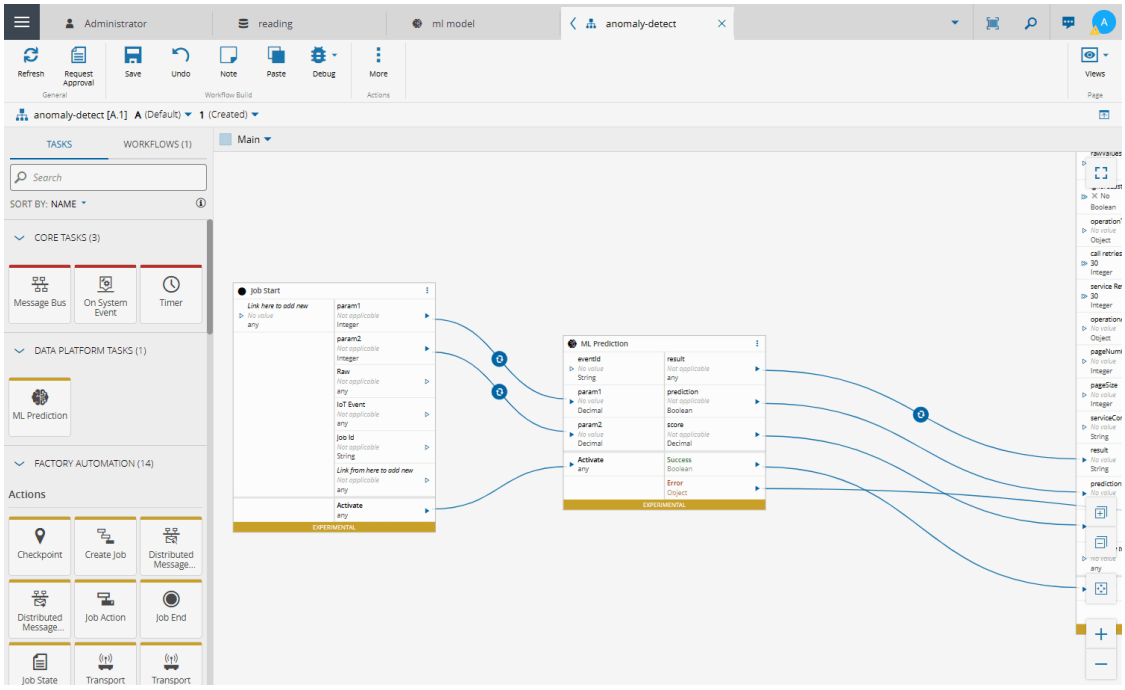
Step 3: Create the IoT Workflow

1. Go to the **IoT Workflow** section of the IoT Event Definition `reading`.
2. Select + to start creating a new workflow.
3. The **Automation Controller Wizard** will open.
4. Define the following:
   - **Change Set** Name: A version-controlled label for the workflow.
   - Workflow Name: A descriptive name - we shall call it `anomaly-detect`.
   - Scope: Confirm it defaults to `Data Platform`.
5. Select Create to complete the operation. The IoT workflow will open and can be consulted in the IoT Workflows section.



Step 4: Configure the Anomaly Detection Workflow

1. Add the ML Task:
   - From the task library on the left, add a **ML Task** to the workflow.
   - Select settings and specify the pre-trained **ML Model** to apply for anomaly detection. The parameters the ML Model expect as inputs should appear in the inputs tab of the ML task.
2. Set Input Data in ML Task
   - Connect the parameters between both bocks, this is, the job start and ML task.

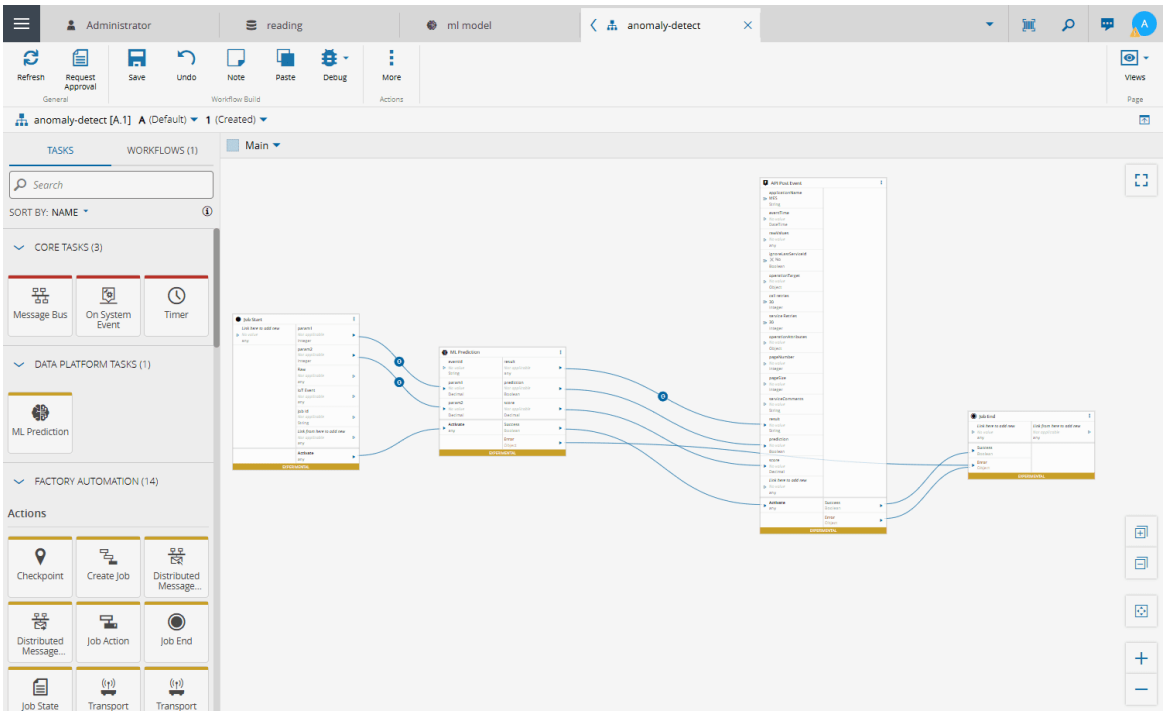


> **Info**
>
> Note we have created an IoT Event Definition whose parameters have type Integer. To successfully connect the parameters between the two blocks we need to apply converters. Over the lines, right select, and then select the Converters button.

3. Add an API Post Event Task:
   - Add an API Post Event task to send anomaly detection results to a specific endpoint:
     - To achieve this, we have created an Iot Event called predictions where we can store both parameters and the prediction outputs.
   - Select settings, select the IoT event and confirm the inputs. Select OK.
   - Connect the ML task output to the API Post Event task inputs, as needed.

> **Info**
>
> Do not forget to connect each block state to activate the execution of the block ahead.

The image that follows is an example of a workflow, but without the event enrichment since we are not storing the parameters next to the ML predictions:



Step 5: Save and Deploy

1. Save the configured workflow.
2. Deploy it in a production environment to start detecting anomalies in real-time.
3. Monitor the workflow performance and adjust thresholds as needed.

Step 6: Visualize Anomalies in Grafana

1. Set Up Dashboards:

- Create a Grafana dashboard to visualize detected anomalies.
- Use panels like time-series charts, tables, or heatmaps to display anomaly trends and severity.

2. Add Filters and Annotations:
   - Include filters for time ranges, event types, or severity levels.
   - Annotate graphs with anomaly events for better context.
3. Enable Real-Time Monitoring:
   - Configure panels to refresh frequently, ensuring real-time updates.

Here, we can immediately observe the abnormal data points by looking into the scatter plot and how the ML Model efficiently detects the outliers. We can see from the time-series the period when the abnormal behavior started.

> **Info**
>
> Keep in mind that within the IoT Workflow, the user may want to trigger a notification in CM MES for the engineers to take action.

Best Practices

- **Fine-Tune Models**: Regularly update ML models with new data to improve detection accuracy.
- **Set Appropriate Thresholds**: Balance sensitivity to avoid excessive false positives or negatives.
- **Monitor Workflow Performance**: Use Grafana dashboards to ensure the workflow is operating as intended.

By following these steps, you will have a robust anomaly detection workflow integrated with your Data Platform, thus enabling proactive monitoring and rapid responses to unexpected events.

Legal Information

Disclaimer

The information contained in this document represents the current view of Critical Manufacturing on the issues discussed as of the date of publication. Because Critical Manufacturing must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Critical Manufacturing, and Critical Manufacturing cannot guarantee the accuracy of any information presented after the date of publication. This document is for informational purposes only.

Critical Manufacturing makes no warranties, express, implied or statutory, as to the information herein contained.

Confidentiality Notice

All materials and information included herein are being provided by Critical Manufacturing to its Customer solely for Customer internal use for its business purposes. Critical Manufacturing retains all rights, titles, interests in and copyrights to the materials and information herein. The materials and information contained herein constitute confidential information of Critical Manufacturing and the Customer must not disclose or transfer by any means any of these materials or information, whether total or partial, to any third party without the prior explicit consent by Critical Manufacturing.

Copyright Information

All title and copyrights in and to the Software (including but not limited to any source code, binaries, designs, specifications, models, documents, layouts, images, photographs, animations, video, audio, music, text incorporated into the Software), the accompanying printed materials, and any copies of the Software, and any trademarks or service marks of Critical Manufacturing are owned by Critical Manufacturing unless explicitly stated otherwise. All title and intellectual property rights in and to the content that may be accessed through use of the Software is the property of the respective content owner and is protected by applicable copyright or other intellectual property laws and treaties.

Trademark Information

Critical Manufacturing is a registered trademark of Critical Manufacturing.

All other trademarks are property of their respective owners.