SIEMENS

Analyze MyDrives /Edge

Operating Manual

Preface

Industrial Security 1

Security instructions 2

Introduction 3

Preparing 4

Commissioning and configuring 5

Starting Analyze MyDrives / Edge 6

User interface Analyze MyDrives /Edge 7

Basic information on Models and CMS Dashbord 8

Operating Analyze MyDrives /Edge 9

Mechanical Anomaly Detector Starter Kit 10

Customizable AI Model 11

V1.4.0

09/2023
A5E50643086B AE

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ **DANGER**
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ **WARNING**
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ **CAUTION**
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ **WARNING**
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E50643086B AE
Ⓟ 10/2023 Subject to change

Preface

Purpose of this publication

This documentation provides the basic information you need to commission, operate and diagnose "Analyze MyDrives /Edge".

This documentation is intended for commissioning engineers, operators, and technologists.

The documentation helps you with the following tasks:

- Configuring "Use Cases" for "Energy Efficiency" and "Mechanical Anomaly Detector"
- Importing and activating AI models
- Configuring "Customizable AI Model"
- Monitoring Key Performance Indicators ("KPIs") with "Overview Widgets" and "Details Pages"
- Configuring the "Condition Monitoring System Dashboard" for monitoring the state of the vibration sensors with "Overview Widgets" and "Details Pages"

Basic knowledge required

- Solid knowledge of SINAMICS drives
- Solid knowledge of engineering software such as STARTER, Startdrive, or TIA Portal
- In-depth knowledge of SINAMICS TRCDATA
- Solid knowledge of operating the Drive System Framework application
- Solid knowledge of operating and commissioning Industrial Edge
- In-depth knowledge of Data Service
- Solid knowledge of personal computers
- General knowledge of automation technology

Scope of this document

Commissioning, operation and diagnostics of "Analyze MyDrives /Edge".

Convention

- The term "Data Source" as used in this documentation refers to the data collected by the SINAMICS drives about "High-Speed-Adapter" or "Low-Speed-Adapter".
- The terms "AI Model" and "Model" as used in this documentation refer to the "Artificial Intelligence Model" used to analyze the data provided via "Data Source".

- The term "Use Case" as used in this documentation refers to the linking of a "Data Source" with a "model" in order to generate real-time and historical KPIs and to graphically represent them graphically in an "Overview Widget" and a "Details Page".
- The terms "Condition Monitoring System" and "CMS" as used in this documentation refer to the Condition Monitoring System SIPLUS CMS 1200.
- The terms "Condition Monitoring System Dashboard" and "CMS Dashboard" as used in this documentation refer to the "Overview Widgets" and "Details Pages" of data collected from SIPLUS CMS 1200 SM 1281 modules.
- The term "Mechanical Anomaly Detector" refers to the "Model" used to detect mechanical anomalies using data from the "High-Speed-Adapters".
- The term "Energy Efficiency" refers to the "Model" used to analyze the energy consumption of drives using data from the "Low-Speed-Adapters".
- The term "drive" used in this documentation refers to SINAMICS drives.
- The term "drive object" used in this documentation refers to subcomponents of the SINAMICS drive.
- The term "Industrial Edge" used in this documentation refers to the "SIEMENS Industrial Edge Ecosystem".
- The term "application" used in this documentation refers to "Industrial Edge Apps".
- The terms "TRCDATA" and "Trace Data" as used in this documentation refer to "SINAMICS TRCDATA".
- The terms "Drive System Framework" and "DSF" as used in this documentation refer to the Drive System Framework Industrial Edge application.
- The term "High-Speed-Adapter" as used in this documentation refers to the "Drive System Framework high-speed data acquisition adapter".
- The term "Low-Speed-Adapter" as used in this documentation refers to the "Drive System Framework Low Speed data acquisition adapter".
- The term "Customizable AI Model" as used in this documentation refer to "models" created and uploaded by users. "Customizable AI Models" allow users to define conditions and to analyze these specific conditions.
- The terms "Operational State Filter" and "OSF", as used in this documentation, refer to filters that are used by Models. You can use "OSF" to filter out irrelevant input data if the corresponding Model supports "OSF".
- The term "Basic Anomaly Detection" used in this documentation refers to the "Model" used to produce statistical analysis results of drives using data from the "Low-Speed-Adaptern".

Table of contents

Preface ... **3**

1 Industrial Security ... **9**

2 Security instructions ... **11**

3 Introduction ... **13**

4 Preparing ... **15**

4.1 System requirements ... 15

4.2 Requirements ... 16

5 Commissioning and configuring ... **19**

5.1 Configuring Industrial Edge Databus ... 19

5.2 Configuring SINAMICS TRCDATA ... 20

5.3 Configuring the Drive System Framework Low-Speed-Adapter ... 23

5.4 Configuring the Drive System Framework High-Speed-Adapter ... 25

5.5 Installing Analyze MyDrives /Edge ... 27

6 Starting Analyze MyDrives /Edge ... **29**

6.1 Starting the Analyze MyDrives /Edge ... 29

7 User interface Analyze MyDrives /Edge ... **31**

7.1 Layout and operator controls ... 31

7.2 Operating areas ... 32

8 Basic information on Models and CMS Dashbord ... **39**

8.1 Energy Efficiency Model ... 39
8.1.1 Energy Efficiency Model Parameter ... 39
8.1.2 Energy Efficiency Model KPIs ... 40
8.1.3 Energy Efficiency Overview Widget ... 41
8.1.4 Energy Efficiency Use Case Details Page ... 43

8.2 Mechanical Anomaly Detector Model ... 44
8.2.1 Mechanical Anomaly Detector Model Parameter ... 44
8.2.2 Mechanical Anomaly Detector Model KPIs ... 44
8.2.3 Mechanical Anomaly Detector Overview Widget ... 44
8.2.4 Mechanical Anomaly Detector Use Case Details Page ... 46

8.3 CMS Dashboard ... 48
8.3.1 CMS Dashboard Overview Widget ... 48
8.3.2 CMS Dashboard Details Page ... 49

8.4 Basic Anomaly Detection ... 53
8.4.1 Basic Anomaly Detection Model Parameter ... 53
8.4.2 Basic Anomaly Detection Model KPIs ... 53

8.4.3 Basic Anomaly Detection Overview Widget 54
8.4.4 Basic Anomaly Detection Details Page 55

9 Operating Analyze MyDrives /Edge 57

9.1 Model Management 59
9.1.1 Inserting a "Model" 59
9.1.2 Activating a "Model" 63
9.1.3 Editing a "Model" 67
9.1.4 Updating a "Model" 70
9.1.5 Deactivating a "Model" 73
9.1.6 Deleting a "Model" 76

9.2 Use Cases 79
9.2.1 Inserting a "Use Case" 79
9.2.2 Activating a "Use Case" 84
9.2.3 Deactivating a "Use Case" 86
9.2.4 Editing a "Use Case" 88
9.2.5 Deleting a "Use Case" 90
9.2.6 Opening the "Use Case" details page 92
9.2.7 Adapting the time interval on the details page 93

9.3 CMS Dashboard 98
9.3.1 Inserting a "CMS Dashboard" element 98
9.3.2 Editing a "CMS Dashboard" element 103
9.3.3 Deleting a "CMS Dashboard" element 105
9.3.4 Opening the "CMS Dashboard" details page 107

9.4 Operational State Filter 109
9.4.1 Inserting an "Operational State Filter" 109
9.4.2 Starting/stopping "Raw Data Collection" 112
9.4.3 Finding suitable raw data for OSF 112
9.4.4 Defining filters 115
9.4.5 Editing filters 119
9.4.6 Deleting filters 119

9.5 Settings 121
9.5.1 Changing the time zone 121
9.5.2 Change Databus credentials 123

10 Mechanical Anomaly Detector Starter Kit 125

10.1 Introduction 126
10.1.1 Mechanical Anomaly Detector Starter Kit 126
10.1.2 Mechanical Anomaly Detector Starter Kits functionality 126
10.1.3 Information about the software license 127

10.2 Security instructions 128
10.2.1 Protection of the host computer 128
10.2.2 Notes on protecting administrator accounts 128
10.2.3 Notes regarding use 128

10.3 Installing Mechanical Anomaly Detector Starter Kit 129
10.3.1 Installing and running 129

10.4 Using Mechanical Anomaly Detector Starter Kit 132
10.4.1 Preparing data for training 132
10.4.2 Training models 134

10.4.3 Packaging models into an inference pipeline 135
10.4.4 Locally testing the pipeline configuration package 138
10.4.5 Mockup of the logging of the AI Inference Server 141
10.4.6 Deployment of the packaged inference pipeline for AI @Edge 141

11 Customizable AI Model 143

11.1 Introduction 143

11.2 Creating a "Customizable AI Model" 144
11.2.1 AI Software Development Kit 144
11.2.2 Adapting AI SDK project templates 144
11.2.2.1 Inference.py file format 145
11.2.2.2 Folder hierarchy for project templates 145
11.2.2.3 Encrypting the model ZIP file 146
11.2.3 Using the "Analyze My Drives/Edge" template example 146
11.2.3.1 Installing and running 146
11.2.3.2 Adapting and generating the model 148

11.3 Data format for "Customizable AI Model" 149
11.3.1 Continuous parameter 149
11.3.2 Read Once and User Defined parameters 150

11.4 Configuring "Customizable AI Model" 151
11.4.1 Overview 151
11.4.2 Structure of the configuration file 151
11.4.3 Element "name" 152
11.4.4 Element "version" 152
11.4.5 Element "family" 152
11.4.6 Elements "type" und "model" 152
11.4.7 Element "datasource" 153
11.4.8 Element "read_once_input_configuration" 153
11.4.9 Element "user_defined_configuration" 155
11.4.10 Element "continuous_input_configuration" 157
11.4.11 Element "output_configuration" 158
11.4.12 Element "ui_config" 161
11.4.13 Element "aggregation_method" 165
11.4.14 Element "execution_configuration" 165
11.4.15 Element "trigger" 166

11.5 Creating and uploading a Use Case for a "Customizable AI Model" 167

11.6 Configuration files - examples 168
11.6.1 Example: Configuration file with "Low-Speed-Adapter" as "Data Source" 168
11.6.2 Example: Configuration file with "High-Speed-Adapter" as "Data Source" 171

Industrial Security

1

Siemens offers products and solutions with Industrial Security functions that support the secure operation of plants, systems, machines, and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement (and continuously maintain) a holistic Industrial Security concept that is state of the art. The products and solutions from Siemens constitute only one component of such a concept.

The customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Systems, machines and components should only be connected to the company network or the Internet if and to the extent that this is necessary and appropriate protective measures (e.g. use of firewalls and/or network segmentation) have been taken.

More information about Industrial Security measures that can be implemented can be found here (http://www.siemens.com/industrialsecurity).

Siemens products and solutions are constantly being further developed to make them even more secure. Siemens expressly recommends that updates be made as soon as the corresponding updates are available and that only the latest product versions be used. Using product versions that are no longer supported and not applying the latest updates may increase the customer's exposure to cyber threats.

Notifications regarding product updates are available via the Siemens Industrial Security RSS Feed, which you can subscribe to here (https://support.industry.siemens.com/cs/start?lc=en-WW).

Security instructions

2

Also refer to the safety instructions in the "Industrial Edge Management - Security overview and requirements (https://support.industry.siemens.com/cs/document/109781002)" manual, which also apply to this documentation.

Introduction

3

"Analyze MyDrives /Edge" is an Industrial Edge application for monitoring and analyzing the SINAMICS drives High-Speed and Low-Speed data.

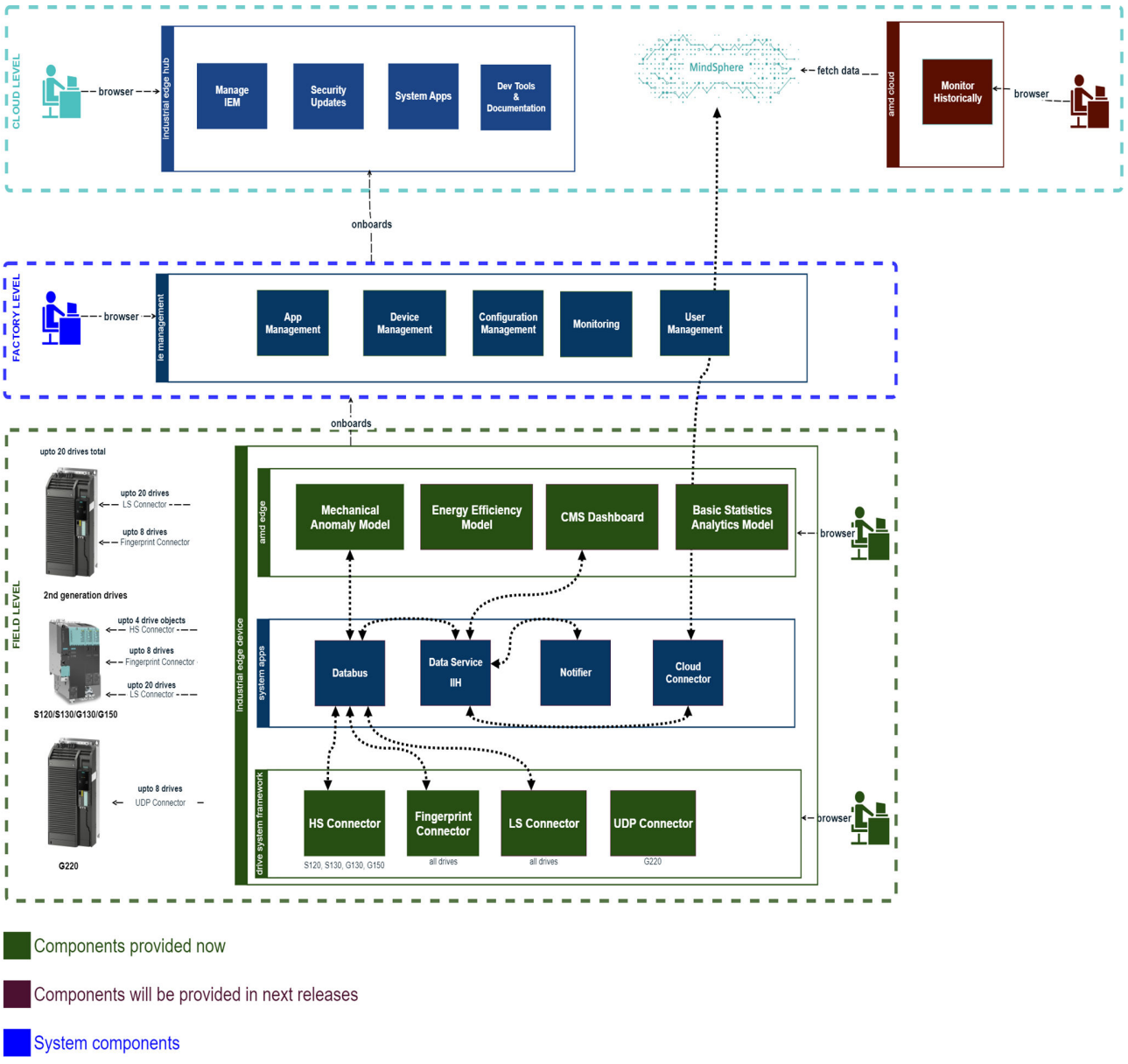
The main advantage of this application is the reduction of downtimes and the increase in sustainability for the connected systems. "Analyze MyDrives /Edge" processes and calculates data provided by the "Drive System Framework" application using artificial intelligence and data analysis models. The Key Performance Indicators calculated by these models (KPIs) are displayed in real-time and historical graphical representations.

Features:

- The integrated "Mechanical Anomaly Detector" is used to detect anomalies in mechanical systems by analyzing drive data.
- The integrated energy efficiency function calculates energy consumption, including operating costs and the CO2 footprint for 2nd generation SINAMICS Type G drives.
- A "Customizable AI Model" analyzes specific conditions defined by the user.
- The "Condition Monitoring System Dashboard" monitors the vibration data and the KPIs calculated by SIPLUS CMS 1200 systems.
- "Basic Anomaly Detection" allows the user to visualize basic statistical characteristics such as mean value, standard deviation, etc. for specific drive parameters and for configurable operational states.

Key benefits:

- Increased plant availability and avoidance of downtimes.
- Prediction of mechanical anomalies to generate optimized service and maintenance strategies.
- The "Condition Monitoring System Dashboard" creates transparency about the status of the connected CMS 1200 systems and enables use cases, such as predictive maintenance of mechanical components, at all levels.
- Transparency of the energy consumption, operating costs, and CO2 footprint of the connected G-type SINAMICS drives.
- Transparency for the systematic variation of statistical characteristics of drive data as they result from the degradation of drive train and application components. This permits evaluation, long-term degradation monitoring, and anomaly detection.


CLOUD LEVEL
browser
industrial edge hub
Manage IEM
Security Updates
System Apps
Dev Tools & Documentation
MindSphere
fetch data
mind cloud
Monitor Historically
browser
onboards
FACTORY LEVEL
browser
ie management
App Management
Device Management
Configuration Management
Monitoring
User Management
onboards
FIELD LEVEL
upto 20 drives total
upto 20 drives
LS Connector
upto 8 drives
Fingerprint Connector
2nd generation drives
upto 4 drive objects
HS Connector
upto 8 drives
Fingerprint Connector
upto 20 drives
LS Connector
S120/S130/G130/G150
upto 8 drives
UDP Connector
G220
industrial edge device
amd edge
Mechanical Anomaly Model
Energy Efficiency Model
CMS Dashboard
Basic Statistics Analytics Model
browser
system apps
Databus
Data Service IIH
Notifier
Cloud Connector
drive system framework
HS Connector
S120, S130, G130, G150
Fingerprint Connector
all drives
LS Connector
all drives
UDP Connector
G220
browser
Components provided now
Components will be provided in next releases
System components

Preparing

4

4.1 System requirements

- Drive System Framework V1.4.0 or higher
- SINAMICS drives with CU320-2 and TRCDATA technology extension for the Model "Mechanical Anomaly Detector"
- 2nd generation SINAMICS G drives for the Model "Energy Efficiency". The drives which are supported for the Model"Energy Efficiency" can be found in the section "Energy Efficiency Model (Page 39)".
- Industrial Edge Device as SIEMENS IPC 227E (6ES7647-8BD31-0CW1)
- Industrial Edge Virtual Device (6ES7823-0EE10-0AA1)
- Industrial Edge Device Runtime for SIEMENS IPC227E with version V1.2.0 or higher
- Industrial Edge Databus app version V1.2.0 or higher
- Industrial Edge Data Service app version V1.3.2 or higher
- Industrial Edge SIMATIC S7 Connector app version V1.7.0 or higher

  Note

  The SIMATIC S7 Connector is only mandatory when collecting data from CMS modules via S7-1200 PLC and for the transfer of this data to Data Service for use with the "CMS Dashboard".

- Google Chrome version 83.0.4103.116 or higher
- Work memory: 1 GB (min.)
- CPU: at least one core
- Hard disk: 10 GB (min.)

4.2 Requirements

Before commissioning "Analyze MyDrives /Edge", the following requirements must be met.

Preconditions

1. The app is available at the Industrial Edge Hub and available in the Industrial Edge Management catalog.
   See "Industrial Edge Management - Getting Started", Section 3.4 "Market"
   Document ID: A5E50177870-AC, Version V1.2.0
   Siemens Industry Online Support link:
   https://support.industry.siemens.com/cs/document/109795387 (https://support.industry.siemens.com/cs/document/109795387)
2. The Edge Core network has been configured, Industrial Edge Management and the SINAMICS drives are accessible.
   See "Industrial Edge Device – Operation", Section 6.1.2 "Creating the Edge Device configuration file"
   Document ID: A5E50541859-AB, Version V1.2.0
   Siemens Industry Online Support link:
   https://support.industry.siemens.com/cs/document/109795385 (https://support.industry.siemens.com/cs/document/109795385)
3. The Edge Core for IED is created and connected to the IEM. The status of the connection is indicated by LEDs on the Edge Device. The Edge app can be downloaded when connected online.
   See "Industrial Edge Device – Operation", Section 6.1.8 "LED status"
   Document ID: A5E50541859-AB, Version V1.2.0
   Siemens Industry Online Support link:
   https://support.industry.siemens.com/cs/document/109795385 (https://support.industry.siemens.com/cs/document/109795385)
4. The "IE Databus" app is installed on IED.
   See "Industrial Edge Management - Getting Started", Section 5.3 "Installing system apps on Edge Devices ".
   Document ID: A5E50177870-AC, Version V1.2.0
   Siemens Industry Online Support link:
   https://support.industry.siemens.com/cs/document/109795387 (https://support.industry.siemens.com/cs/document/109795387)
5. The "IE Data Service" app is installed on IED.
   See "Industrial Edge Management - Getting Started", Section 5.3 "Installing system apps on Edge Devices ".
   Document ID: A5E50177870-AC, Version V1.2.0
   Siemens Industry Online Support link:
   https://support.industry.siemens.com/cs/document/109795387 (https://support.industry.siemens.com/cs/document/109795387)

6. TRCDATA OA is installed and activated for the respective drive component
   See "SINAMICS TRCDATA Function Manual", Chapter 3 "Installation and activation"
   Document ID: A5E37025411B
   The document can be provided through Support.

   Note

   - With the TRCDATA OA application for SINAMICS drives, signal data is recorded with a high sampling rate. To record signal data, you must install TRCDATA and activate it for the particular drive component.
   - TRCDATA OA is required to activate the "High-Speed adapter" from the "DSF" for CU320-2-based drives. TRCDATA OA is mandatory when using the "Mechanical Anomaly Detector" model.

7. "Drive System Framework", version V1.3 or higher, is installed on the IED.
   See Operating Manual "Drive System Framework".
8. "SIMATIC S7 Connector" is installed on the IED.
   See "Industrial Edge Management - Getting Started", Section 5.3 "Installing system apps on Edge Devices ".
   Document ID: A5E50177870-AC, Version V1.2.0
   Siemens Industry Online Support link:
   https://support.industry.siemens.com/cs/document/109795387 (https://support.industry.siemens.com/cs/document/109795387)

   Note

   The SIMATIC S7 Connector is only mandatory when collecting data from CMS modules via S7-1200 PLC and for the transfer of this data to Data Service for use with the "CMS Dashboard".

Commissioning and configuring 5

5.1 Configuring Industrial Edge Databus

"Analyze MyDrives /Edge" uses the "IE Databus" application for communication and data transfer with a high sampling rate. You must configure the "IE Databus" application for IED before commissioning "Analyze MyDrives /Edge".

You configure the IE Databus in the Industrial Edge Managemen (IEM). The configuration created must be provided on the IED.

Procedure

The procedure for configuring the IED is described in the following manual:

"Industrial Edge Databus Configurator – Operation", Chapter 5 "Working with Industrial Edge Databus Configurator"

Document ID: A5E50594959-AA, version V1.2.0

Siemens Industry Online Support link:

https://support.industry.siemens.com/cs/document/109795600 (https://support.industry.siemens.com/cs/document/109795600)

Note

Login data in the IE Databus Configurator

Make sure to use the following login data for the login:

- User name: Define a user name. You can see the standard user name on the "Settings" page under "Analyze MyDrives /Edge".
- Password: Define a password. Please contact your local Siemens Support organization to obtain a standard password.
- Topic name: amdedge/#
- Approval: Publish and Subscribe

The IE data bus configuration is sensitive and must be executed exactly as described.

5.2 Configuring SINAMICS TRCDATA

After activating TRCDATA for the drive object which is to be monitored for mechanical anomalies, below variables must be set and downloaded to the SINAMICS drive using the requested engineering software like STARTER or Startdrive.

Note

- Parameters p32046, p32048 and p32050 are not relevant for "Analyze MyDrives /Edge" as only signals with a FloatingPoint32 data type are processed.
- Example values for parameters apply to the current version of the "Mechanical Anomaly Detector" model. Depending on the required sampling period of the model, you may need to change these values.

TRCDATA trace mode [p32040]:

To be able to function, the trace mode of the desired drive object must be equal to "1" as for the Continuous Trace.

TRCDATA sampling time factor signal sampling [p32041]

To monitor the belt elongation, the "Mechanical Anomaly Detector" model in "Analyze MyDrives / Edge" requires a sampling period of 500 Hz, this means a time interval of 2 ms between each data point.

- **Example:**
  p32041 must be set to 16 if p0115[0] is set to 125 µs.

For other models, the value must correspond to the "frequency" attribute of parameter "continuous_input_configuration" in file "config.json".

- Example:
  For example, if a frequency of 500 Hz is set and p0115[0] is set to 125 µs, then p32041 must be set to 16, as 1/500 s = 16 × 125 µs.

Note

You may have to adapt the values corresponding to the model configuration file.

TRCDATA sampling time factor trace buffer [p32042]

The trace buffer must be set to 1, that means signals will not be aggregated.

TRCDATA trace buffer number of data points [p32044]

The number of data points must be 200 or an integer multiple of 1200 (e.g. 100, 200, 300, 400, 600).

TRCDATA float signal sources number [p32045]

The number of signals with the data type FloatingPoint32 must be equal or greater than 2. As a minimum, speed and torque values are required to calculate anomaly scores.

TRCDATA float signals compression operation [p32047[0...n]]

As default, for all speed and torque values of the Float signals compression operation is set to 3. However, the setting does not become active as p32042 is equal to 1.

CI: TRCDATA float signals signal source [p32049[0...n]]

The following signals for SINAMICS S120 and SINAMICS S150 drives must be included for the "Mechanical Anomaly Detector" model to monitor the belt elongation:

r63: Actual speed value

r80: Torque actual value

The following signals must be included for the SINAMICS G130 and SINAMICS G150 drives:

r63[1]: Actual speed value

r80[0]: Torque actual value

For user-defined models, the values must correspond to the "id" attributes of the "continuous_input_configuration" parameters in configuration file "config.json" of the configuration package.

Each signal must have an associated set of parameters. For example, for parameters r63 and r80, configuration file "config.json" must include a parameter set for id 63[0] and id 80[0]:

```
"continuous_input_configuration": [
    {
        "id": "63[0]",
        "key": "speed",
        "type": "float",
        "description": { "en": "Actual Speed Smoothed"},
        "frequency": 500
    },
    {
        "id": "80[0]",
        "key": "torque",
        "type": "float",
        "description": { "en": "Actual Torque Smoothed"},
        "frequency": 500
```

```
            }
        ]
```

5.3 Configuring the Drive System Framework Low-Speed-Adapter

So that you can use the "Energy Efficiency" model, you must perform the following operations in "Drive System Framework".

Note

Basic Anomaly Detection

The configuration example for a "Low-Speed-Adapter" based on the "Energy Efficiency" Model. For "Basic Anomaly Detection", the Model is configured in a similar way, but with different parameters (see section "Basic Anomaly Detection" (Page 53)).

Procedure

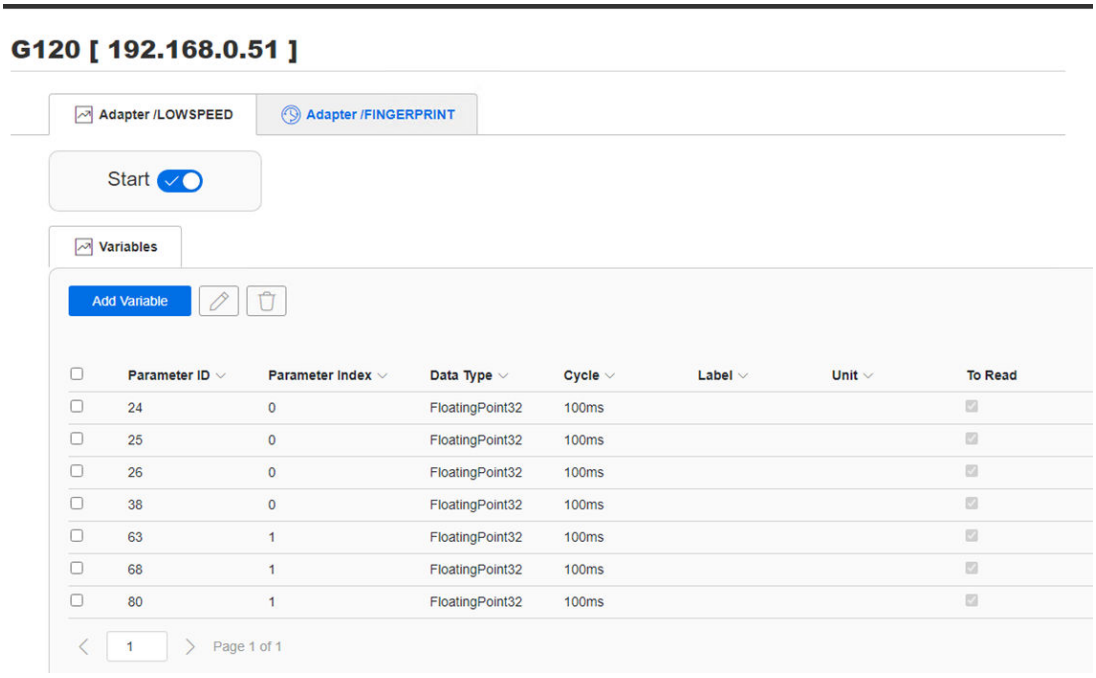
1. Define the drive.
2. Switch to the "Adapter /LOWSPEED" tab.
3. Add the following variables with 100 ms cycle time:
   - R24[0]: Output frequency smoothed
   - R25[0]: Output voltage smoothed
   - R26[0]: Trigger parameter - DC link voltage smoothed
   - R38[0]: Power factor smoothed
   - R63[1]: Speed actual value
   - R68[1]: Absolute actual current smoothed
   - R80[1]: Actual torque value smoothed
4. Start the adapter.

References

See "Drive System Framework" Operating Manual.

Result

After successful configuration in "Drive System Framework", the result should look like this:



5.4 Configuring the Drive System Framework High-Speed-Adapter

So that you can use the "Mechanical Anomaly Detector" model, you must perform the following operations in "Drive System Framework".

Procedure

1. Define the drive.
2. Switch to the "Adapter /HIGHSPEED" tab.
3. Update the topology.
4. Signals
   - r0063: Speed actual value smoothed
   - r0080: Actual torque value smoothed, is checked in the "To Read" column.
5. Perform the calibration. The calibration must be completed successfully; in other words, there must be no error icon on the drive object.
6. Start the adapter.

References

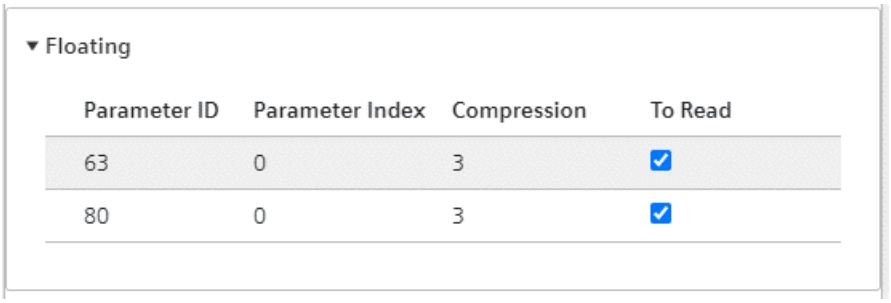
See Operating Manual "Drive System Framework".

Result

After successful configuration in "Drive System Framework", the result should look like this:

Trace Data Configuration

Parameter ID	Parameter Name	Value
p115[0]	Basic Sampling Time	125 µs
p32040	Trace Mode	Continuous
p32041	Signal Sampling Factor	16
p32042	Buffer Sampling Factor	1
p32044	Number of Data Points	200
p32045	Float Signal Count	2
p32046	Integer Signal Count	0



Note

For the SINAMICS G130 and SINAMICS G150 drives, the index for parameter 63 must be 1.

5.5 Installing Analyze MyDrives /Edge

"Analyze MyDrives /Edge" must be installed from IEM to IED. The services and the application start automatically after the installation is finished.

Note

The installation of "Analyze MyDrives /Edge" requires no specific configuration and is subject to the standard installation of Edge applications.

Note

"Analyze MyDrives /Edge" requires no specific configuration at the IEM level. The IP address of the drive is transferred to IED via the "Analyze MyDrives /Edge" user interface.

Starting Analyze MyDrives /Edge

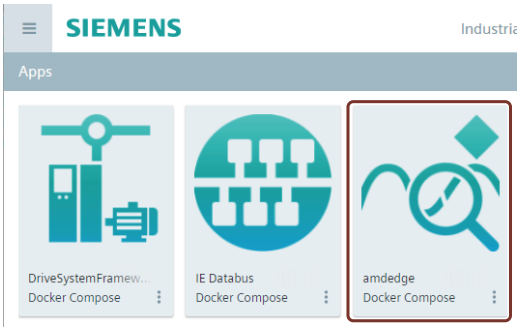
6

6.1 Starting the Analyze MyDrives /Edge

You can call the Industrial Edge application "Analyze MyDrives /Edge" using the two procedures described below.

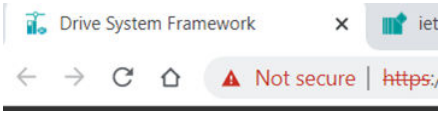
Start via IED user interface

1. Open the web browser.
2. To call the IED user interface, enter https://[Edgebox IP ADDRESS] in the address line.
3. Log on to the IED.
4. Navigate to "Apps"
5. To start, click on the button of the app.



Additional notes:

- Opening the IED user interface requires a secure connection via "**https**".
- After you have logged on at the IED, you can call application "Analyze MyDrives /Edge" by entering the following address in the address line:
  https://[Edgebox IP ADDRESS]/analyzemydrivesedge/
- "Analyze MyDrives /Edge" uses a self-signed certificate. The certificate must be confirmed as trustworthy by you.

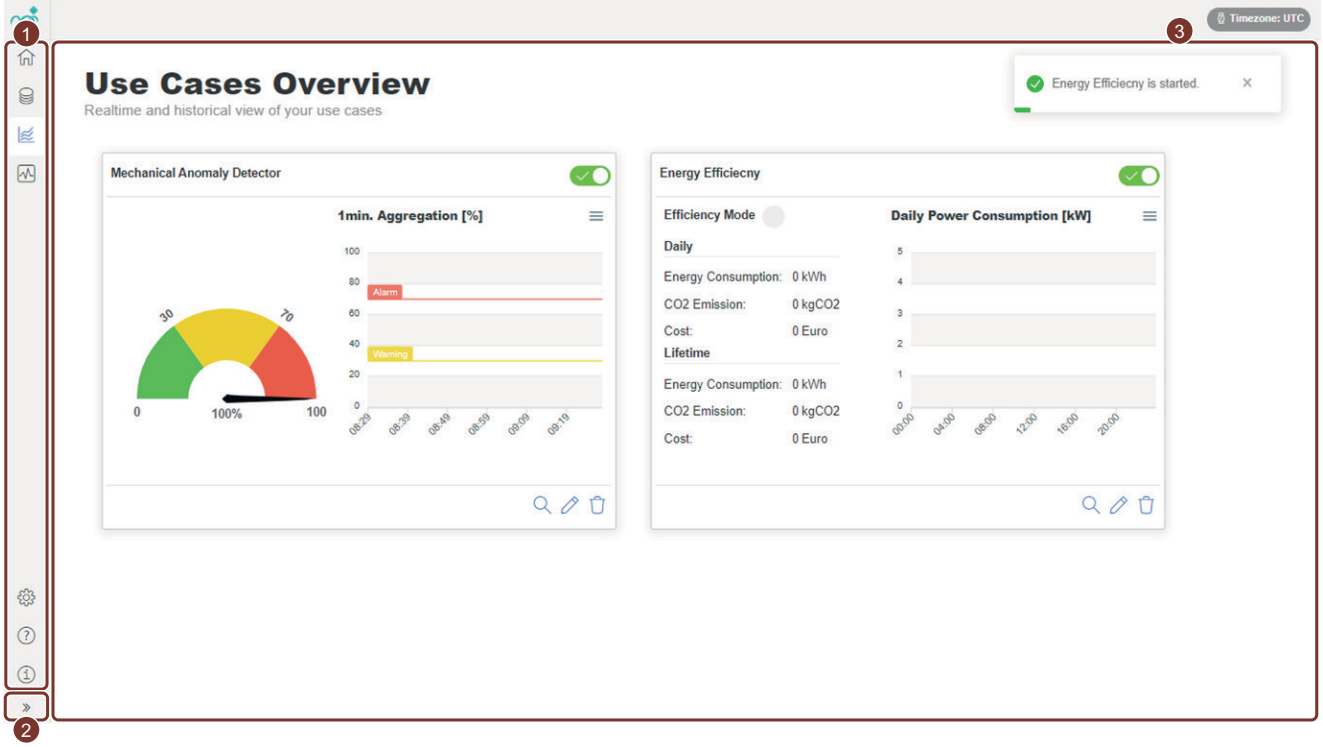


User interface Analyze MyDrives /Edge

7

7.1 Layout and operator controls

The following figure shows the structure and the operator controls of the Industrial Edge App "Analyze MyDrives /Edge".



① Navigation bar: Via the navigation bar, you can call the operating areas "Home", "Use Cases", "Model Management", "CMS Dashboard", "Settings", "Help" and "About".

② Navigation bar extender: Icon for expanding or reducing the navigation bar

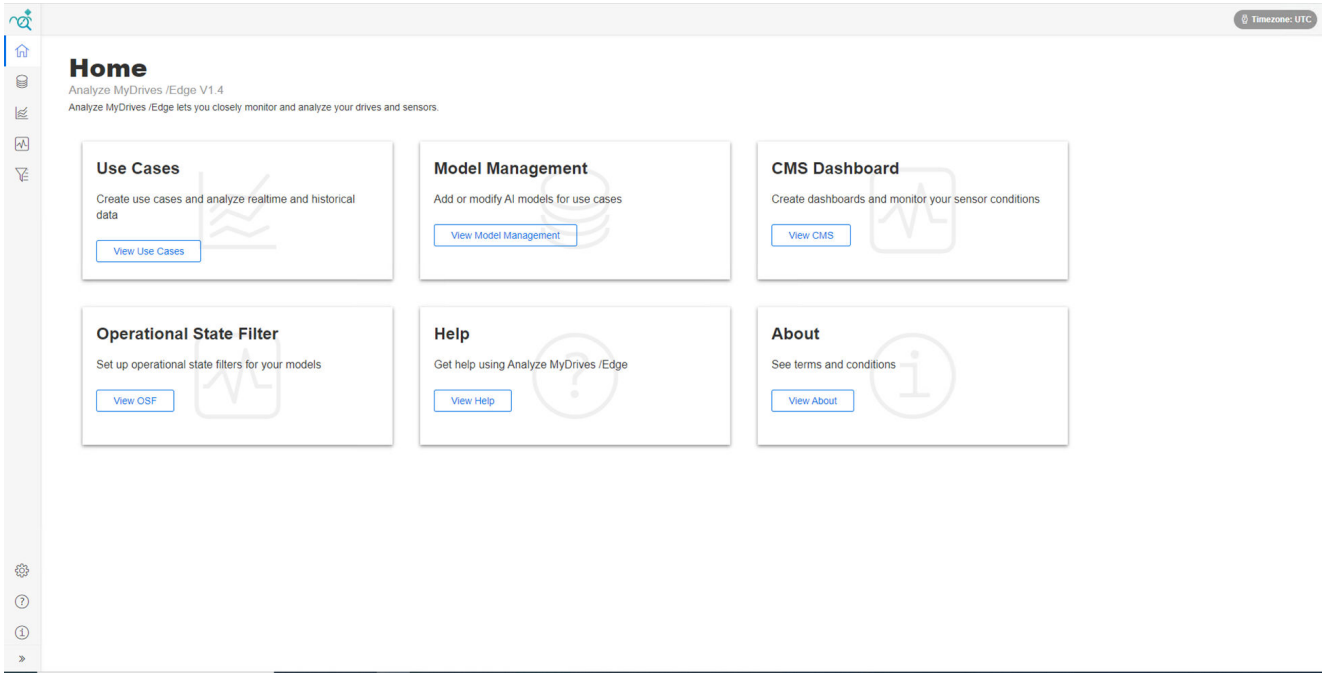
③ Workspace

7.2 Operating areas

The following is an overview of the operating areas of the "Analyze MyDrives /Edge" application.

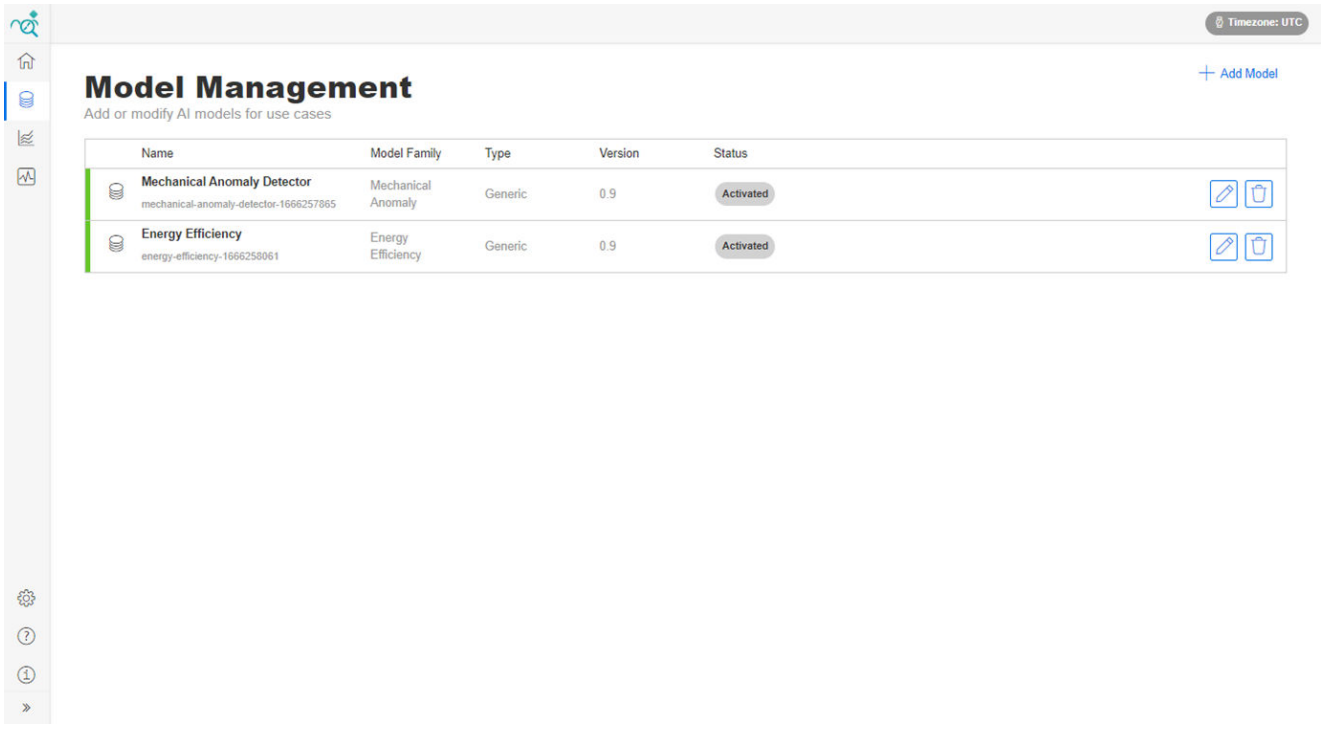
Home

Via the Home page, you can call the operating areas "Use Cases Overview", "Model Management", "CMS Dashboard", "Operational State Filter", "Help" and "About".



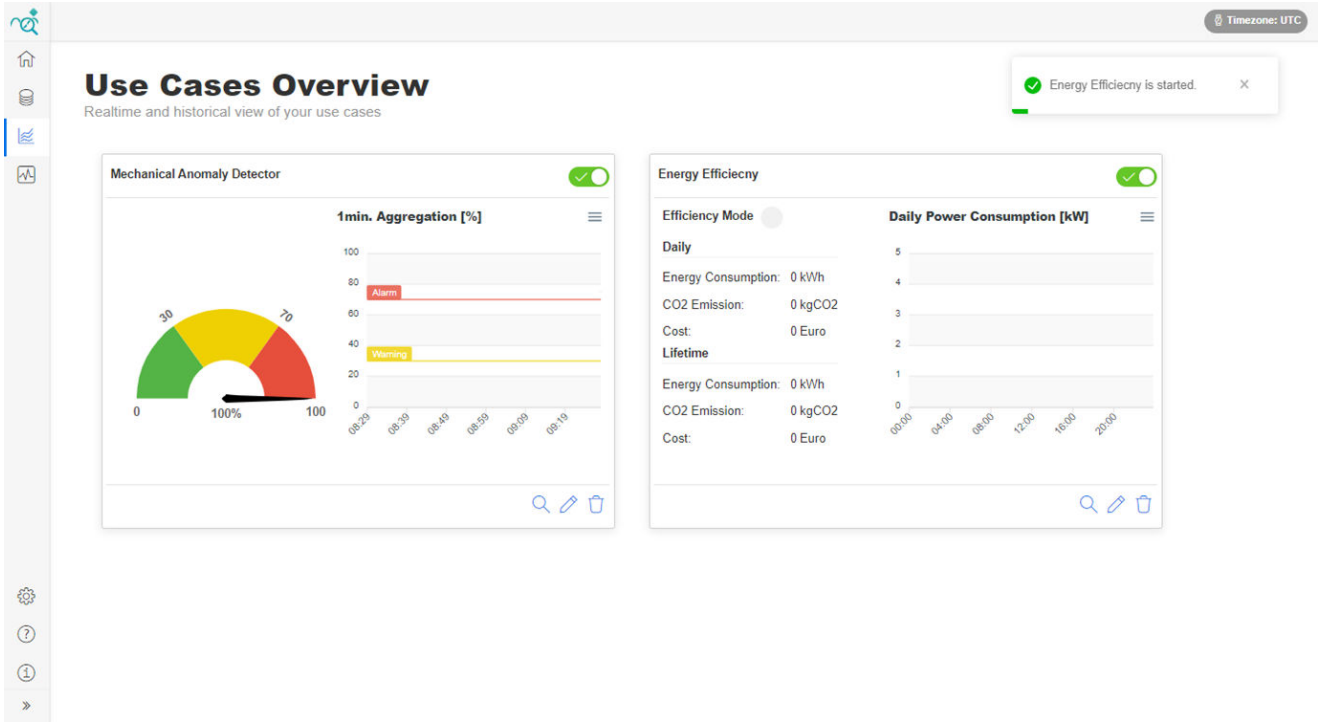
Model Management

In the operating area "Model Management", you can create new "Models" for use with your "Use Cases" and activate them. From this operating area, you can configure or update your "Models".



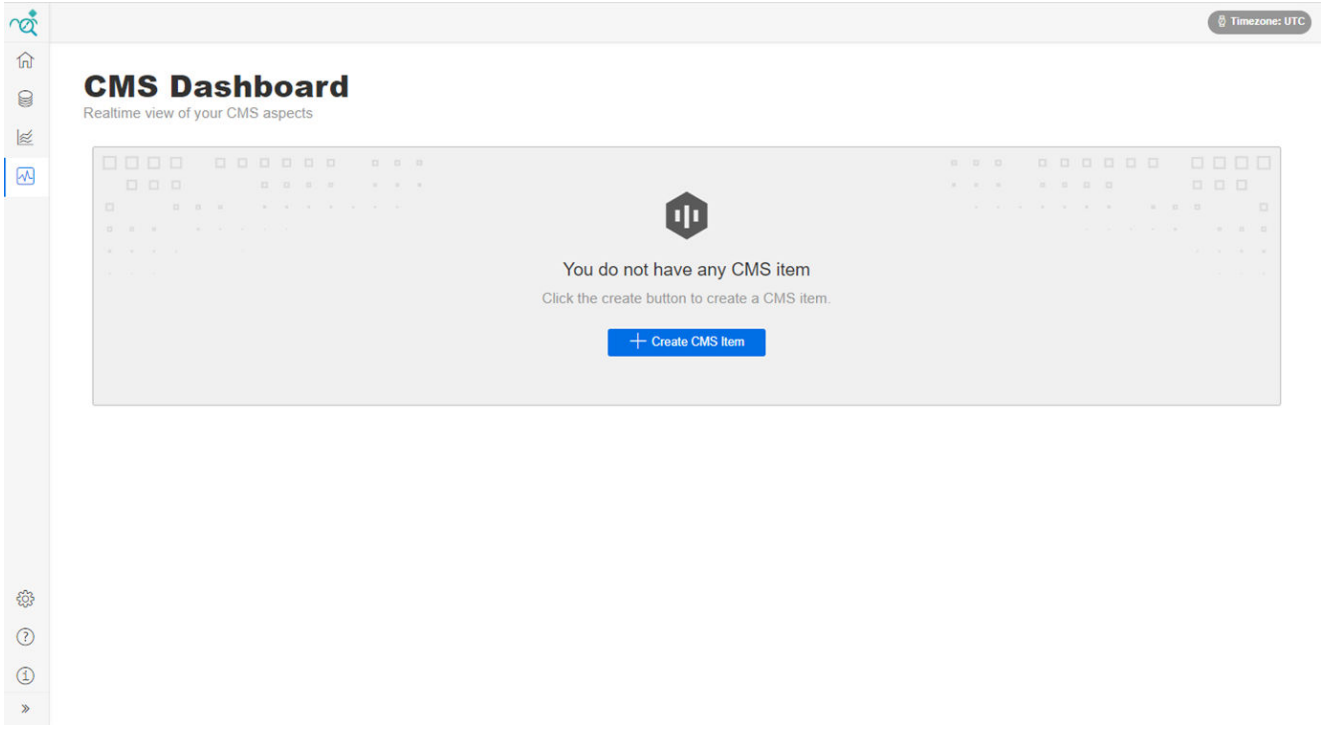
Use Cases Overview

In the "Use Cases Overview" operating area, the real-time monitoring for the models "Energy Efficiency" and "Mechanical Anomaly Detector" with "Overview Widgets" is displayed. From this area, you can also access "details pages" of your use cases to get detailed historical data.



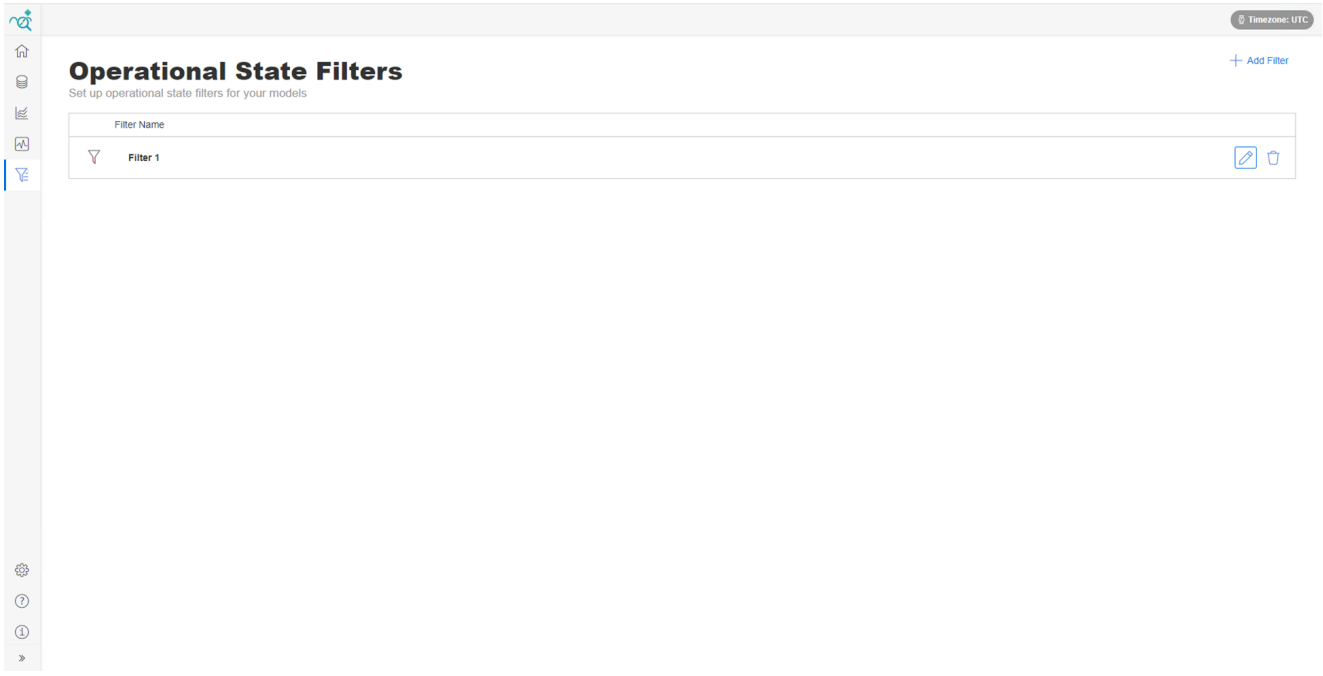
CMS Dashboard

In the "CMS Dashboard" operating area, you can monitor the status of your CMS modules in real time. You can also access the "details pages" of your "CMS Dashboard" to obtain detailed information.



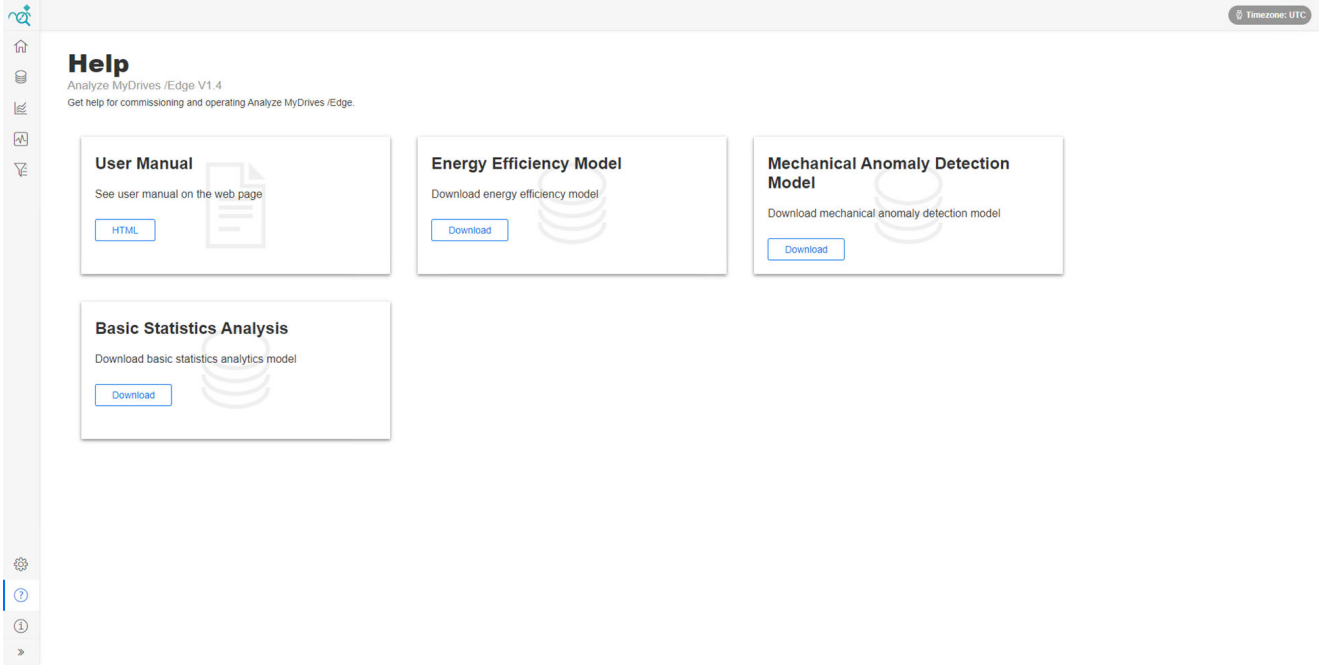
Operational State Filter

In the "Operational State Filter" operating area, you can view the data collected by the "Low-Speed-Adapter" and create filters for your Models.



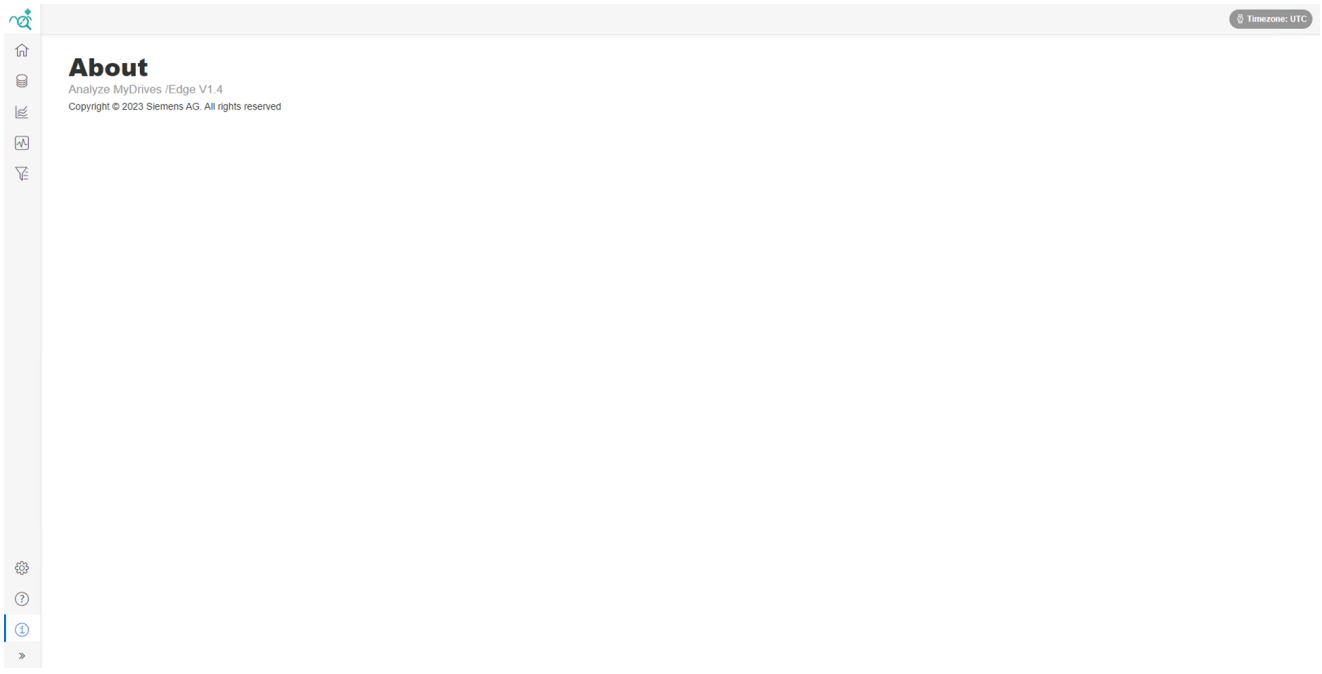
Help

In operating area "Help", you can find the HTML links to the Operating Manual and to product notes. You can find information about the notes, restrictions for the models and the functions in Analyze My Drives Edge in the product notes in Section "Notes on configuration and operation". In this area, you can download the Models "Energy Efficiency", "Mechanical Anomaly Detector", and "Basic Anomaly Detection".



About

The "Analyze MyDrives /Edge" version is displayed under "About".



Basic information on Models and CMS Dashbord 8

8.1 Energy Efficiency Model

The "Energy Efficiency Use Case" allows you to calculate the energy consumption including operating costs and CO2 footprint for SINAMICS G drives of the 2nd generation. The Model "Energy Efficiency" uses the "Low-Speed-Adapter" data provided by the "Drive System Framework" and calculates metrics on energy consumption and drive efficiency for 2nd generation SINAMICS G drives. The results of the "Energy Efficiency" Model are composed of power consumption, energy consumption, operating costs and CO2-equivalent emissions of the drive system, power loss of the frequency converter and power loss of the motor, efficiency of the frequency converter. It also calculates energy savings, CO2 emission savings, and cost savings compared to IE2 devices.

The Model "Energy Efficiency" V.1.0.18 provided with Analyze MyDrives Edge supports the following drives:

Compatible drives supported by Model "Energy Efficiency" V.1.0.18		
No.	**Product family**	**Grid Conditions (IEC vs NEC)**
1	SINAMICS G120	PM240-2 (200V)
2	SINAMICS G120	PM240-2 (400V)
3	SINAMICS G120	PM240-2 (690V)
4	SINAMICS G120C	FSAA-FSF (400V)
5	SINAMICS G130	All frame sizes
6	SINAMICS G150	All frame sizes incl. NEM

8.1.1 Energy Efficiency Model Parameter

The "Energy Efficiency" Model uses the following drive parameters provided by the "Low-Speed-Adapter":

- R24[0]: Output frequency smoothed
- R25[0]: Output voltage smoothed
- R26[0]: Trigger parameter - DC link voltage smoothed
- R38[0]: Power factor smoothed
- R63[1]: Speed actual value
- R68[1]: Absolute actual current smoothed
- R80[1]: Actual torque value smoothed

8.1.2 Energy Efficiency Model KPIs

The "Energy Efficiency" Model generates the following KPIs every 5 minutes:

- **Power consumption of the drive system [kW]:**
  The total power of the drive system required at the frequency converter input to generate the work. The total power consumption, calculated for the period from the activation of the Models to the current time, is displayed in the "Widget" view.
- **Energy consumption of the drive system [kWh]:**
  The energy consumption of the entire drive system during 5 minutes / one day / one month / one year, depending on the period you select. The energy consumption is specified in kilowatt hours. The total energy consumption, calculated for the period from the activation of the Model to the current time, is displayed in the "Widget" view.
- **Operating costs for drive system [euros]:**
  The operating costs of the entire drive system based on the consumed energy with a resolution of 5 minutes / one day / one month / one year, depending on the period you have selected, for graphical representation in detail view. Costs are expressed in user-defined currencies such as euros. The overall operating costs of the drive system, calculated from the activation of the Model, are displayed in the "Widget" view.
- **CO2-equivalent emissions of the drive system per kWh [kgCO2]:**
  The CO2 equivalent emissions of the overall drive system based on the energy consumed in the corresponding/selected period. You can set the kgCO2/kWh coefficient when setting up "Analyze MyDrives".
- **Power loss of the frequency converter [W]:**
  The power loss of the installed frequency converter connected to "Analyze MyDrives /Edge", expressed in watts.
- **Power loss of the motor [W]:**
  The power loss of the built-in motor expressed in watts.
- **Frequency converter degree of utilization [%]:**
  The ratio between the actual output current and the nominal output current of the frequency converter, expressed in percent.
- **Losses ratio between real and reference frequency converter (RCDM) [%]:**
  The ratio between the power loss of the real frequency converter and the power loss of the corresponding reference frequency converter or Reference Complete Drive Module (RCDM), as described in the ecodesign standard IEC 61800-9-2, expressed in percent. The RCDM device has the same or the next larger apparent power compared to the real device in the plant.
- **Avoided amount of losses in comparison to IE2 frequency converter (%):**
  The loss difference in percent between the actual frequency converter compared to 75% of the corresponding RCDM. The IE2-equivalent device is a device that complies with the maximum permissible losses for frequency converters on the EU market.
- **Energy savings compared to IE2 frequency converter [kWh]:**
  The energy savings of the actual frequency converter compared to an IE2-equivalent device that is right at the IE2 limits (namely, a device that has 75% loss of the equivalent RCDM). This IE2-equivalent device is a device that complies with the maximum permissible losses for frequency converters on the EU market.

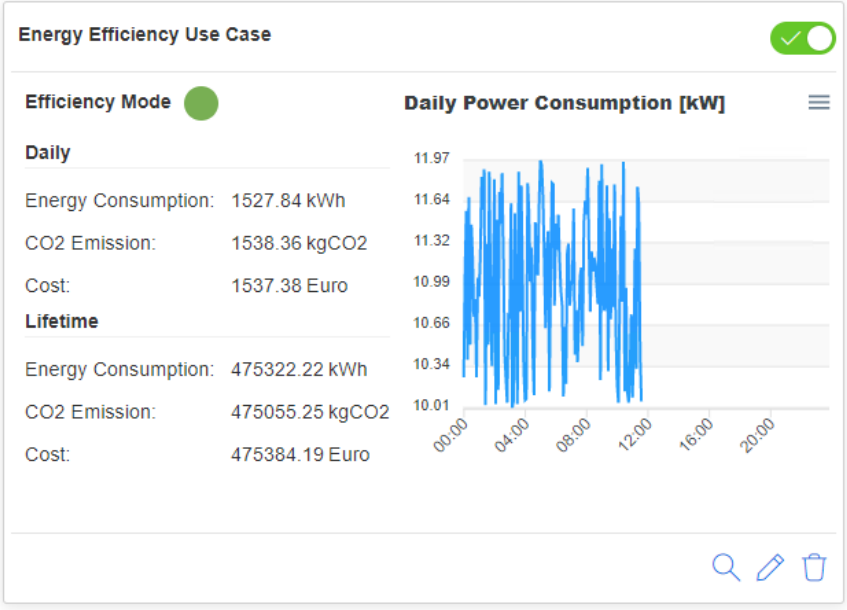
- **CO2 emissions saved in comparison to IE2 device [kgCO2]:**
  The CO2 emission savings of the actual frequency converter compared to an IE2-equivalent device, which are exactly at the IE2 efficiency limits.
- **Costs saved in comparison to IE2 device [euros]:**
  The cost savings of the actual frequency converter compared to an IE2-equivalent device, which are exactly at the IE2 efficiency limits

8.1.3 Energy Efficiency Overview Widget

This section describes the data and graphics that are displayed in the "Overview Widget" for "Energy Efficiency".

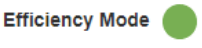
The "Overview Widget" for "Energy Efficiency" is divided into the following 4 sections:

- Efficiency Mode
- "Daily" accumulated data
- "Lifetime" accumulated data
- "Daily power consumption" diagram



Efficiency Mode

The "Efficiency Mode" displays the average energy efficiency of your drive for the last full hour. This value is calculated based on the loss ratio between the real and reference frequency converter (RCDM).



Representation according to the following criteria:

- Green: if the loss ratio is less than 65%
- Orange: if the loss ratio is between 65% and 75%
- Red: if the loss ratio is above 75%

Note

The "Efficiency Mode" is updated once every hour using aggregated data from the last few hours.

Example:

If you check the "Efficiency Mode" between 14:00 and 15:00, the data collected between 13:00 and 14:00 is used to determine the loss rate. After 15:00, the data collected between 14:00 and 15:00 is used to determine the loss rate.

"Daily" accumulated data

The "Daily" accumulated data shows the daily sum of energy consumption, CO2 emissions, and costs from the beginning of the current day. The daily diagram is reset every day at 00:00.

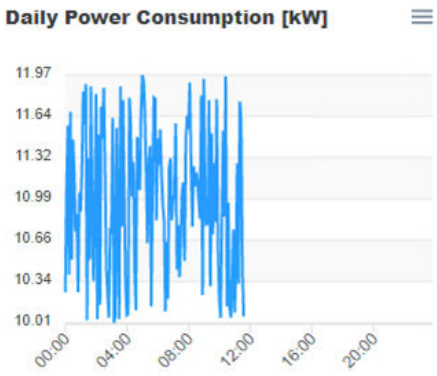
"Lifetime" accumulated data

The "Lifetime" accumulated data shows the sum of all available data on energy consumption, CO2 emissions and costs.

The lifetime of data of the Use Case "Energy Efficiency" is 10 years. Data that is older than 10 years is discarded.

"Daily power consumption" diagram

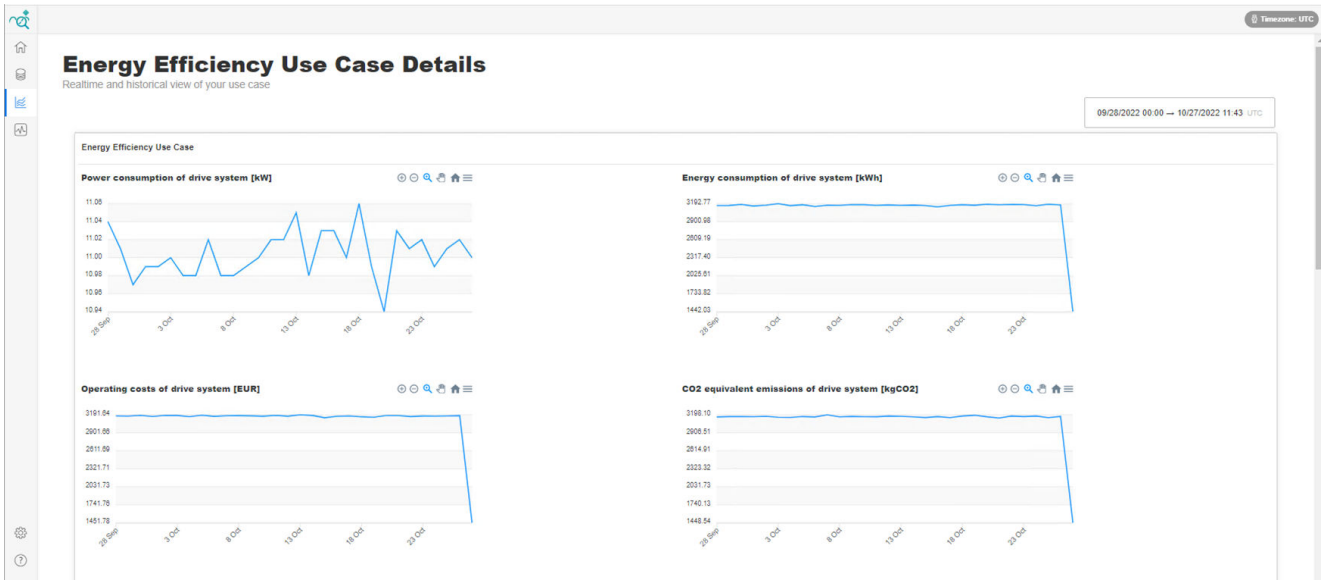
The "Daily power consumption" diagram displays the power consumption as a line graph. The "Daily power consumption" diagram is reset every day at 00:00.



8.1.4 Energy Efficiency Use Case Details Page

This section describes the data and graphics that are displayed on the "Energy Efficiency Use Case Details" page.

The details page contains 12 line diagrams for the KPIs described in the section "Energy Efficiency Model KPIs (Page 40)".



By default, the details page displays the data collected in the last 30 days, including the current day's data. You can adjust the time interval for the diagrams or select "Quickrange". See section "Adapting the time interval on the details page (Page 93)".

8.2 Mechanical Anomaly Detector Model

"Mechanical Anomaly Detector Use Cases" allows you to monitor the condition of mechanical drive train components with a data-drive model, which is trained for High-Speed drive data. An anomaly indicator is displayed, indicating how far the component's condition is from the trained good condition.

By adjusting the indicator thresholds, warnings can be generated when a critical condition has been reached. This allows you to perform appropriate maintenance in a timely manner.

Templates of trained models for specific drive train components, such as belts, are available. If, for example, a belt of a belt system is too slack or too tight, you will be informed and you can change the belt tension accordingly.

8.2.1 Mechanical Anomaly Detector Model Parameter

The "Mechanical Anomaly Detector" Model uses the following drive parameters provided by the "High-Speed-Adapter":

- r0063: Speed actual value smoothed
- r0080: Actual torque value smoothed

Models can be trained for different sample and data point values. The belt elongation template is based on 2 ms and an integral divisor of 1200 data points.

8.2.2 Mechanical Anomaly Detector Model KPIs

The "Mechanical Anomaly Detector" Model generates an anomaly score approximately every 2.4 seconds. The significance of the anomaly score depends on the data used for model training.

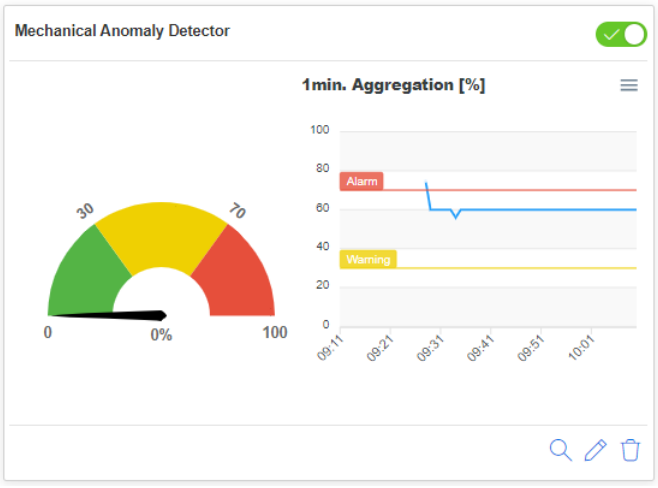
The available belt elongation template described in the section "Using Analyze MyDrives / Edge (Page 57)" uses an anomaly score to detect the belt tension.

8.2.3 Mechanical Anomaly Detector Overview Widget

This section describes the data and graphics that are displayed in the "Overview Widget" for the "Mechanical Anomaly Detector".

The "Overview Widget" for the "Mechanical Anomaly Detector" is divided into the following two sections:

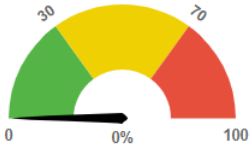
- Gauge diagram for the last calculated anomaly value
- Line diagram for the calculated anomaly value of the last 60 minutes



Gauge diagram

The gauge diagram shows the last calculated anomaly value. Refreshing takes place approximately every 2.4 seconds, depending on the Model setting in the configuration file. The diagram has three ranges:

- Normal range: green area
- Warning range: yellow area
- Alarm range: red area

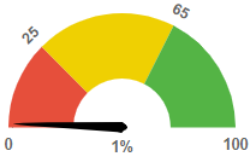


In order to display a high Score, set the warning level higher than the alarm level in the "Model Management" operating area.

Example:

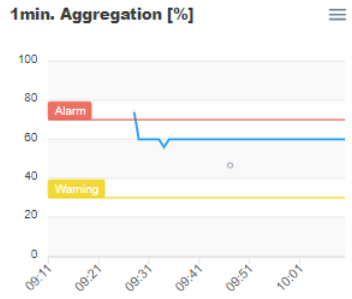
Alarm level: 25

Warning level: 65



Line diagram for the calculated Anomaly Score

The line graph in the "Overview Widget" aggregates the anomaly value collected in the last 60 minutes by calculating the average anomaly score value for each minute. Alarm level and warning level are also displayed in this diagram and are also reversible, as described in the explanation "Gauge diagram".



Note

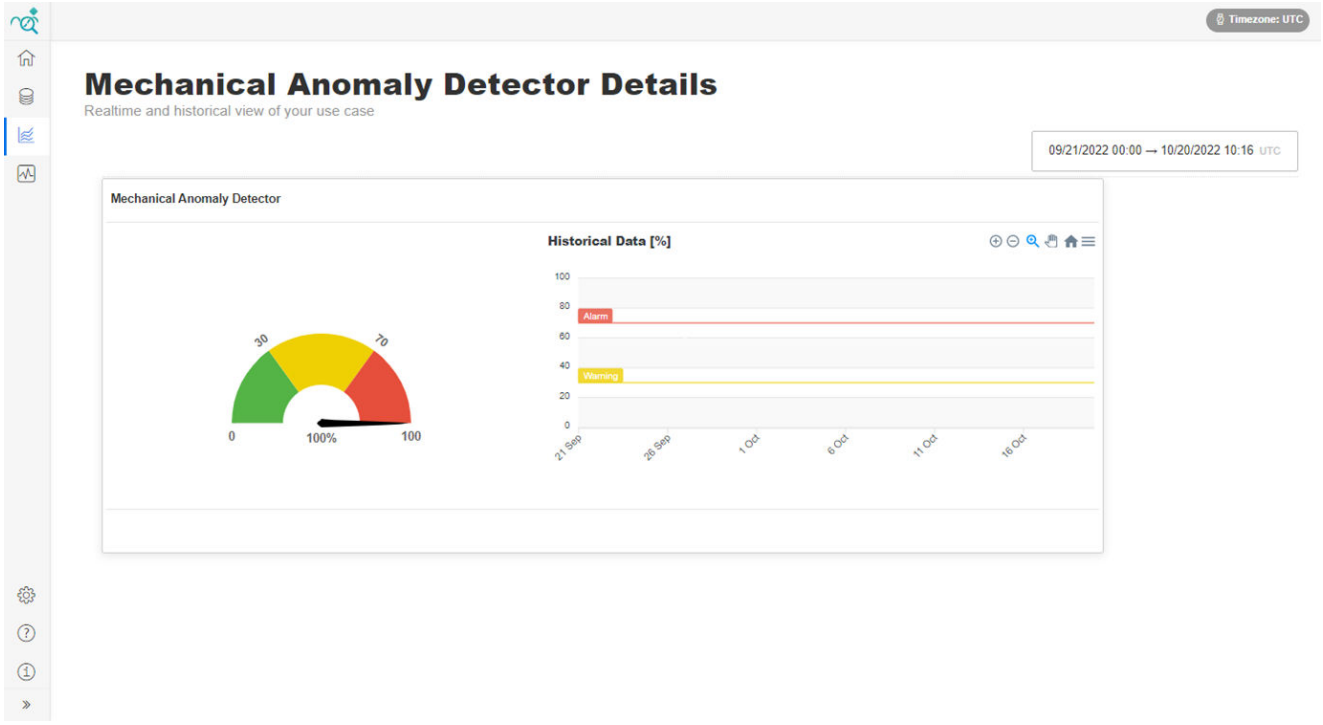
The line diagram shows the average anomaly score of each minute. Therefore, the average Score must exceed the warning level or alarm level to be displayed in the warning or alarm zone.

8.2.4 Mechanical Anomaly Detector Use Case Details Page

This section describes the data and graphics that are displayed on the "Mechanical Anomaly Detector Use Case Details" page.

The details page contains two diagrams for the calculated KPIs.

- Gauge diagram for the last calculated anomaly value
- Line diagram for the calculated anomaly value for the selected time interval



The gauge diagram of the details page is identical in function to the gauge diagram in the "Overview Widget". It displays the latest measured value in the corresponding zone, according to the Model configuration.

By default, the details page displays the data collected in the last 30 days, including the current day's data. You can adjust the time interval for the diagrams or select "Quickrange". See section "Adapting the time interval on the details page (Page 93)".

8.3 CMS Dashboard

The "CMS Dashboard" monitors the vibration data of SIPLUS CMS 1200 Condition Monitoring systems. The "CMS Dashboard" creates transparency about the status of the connected CMS 1200 systems. This enables predictive maintenance of mechanical components across all levels, for example. With "CMS Dashboard", vibration and state variables from SIPLUS CMS 1200 systems can be monitored. The variables are explained and verified in the following chapters.

8.3.1 CMS Dashboard Overview Widget

This section describes the data and graphics displayed in the "CMS Dashboard Overview Widget". The "CMS Dashboard Overview Widget" consists of two areas:

- Information on Dashboard elements
- Channel status indicators

Information on Dashboard elements

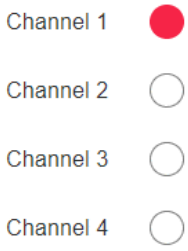
For the respective Dashboard elements, the CMS type, the Asset name, and the Aspect name are displayed. This information is retrieved by the Industrial Edge Data Service application.

Channel status indicators

The Channel status indicators show the status of the channel by evaluating the data collected by sensors.

The colors of the Channel status indicators have the following meanings:

- Green: All variables of the channel are in normal state, there are no warnings or alarms.
- Yellow: At least one of the channel's variables has a warning status.
- Red: At least one of the channel's variables has an error status.
- Gray: The channel is configured to collect data, but there has been no new data in the last hour.
- White: The channel is not configured for data acquisition.

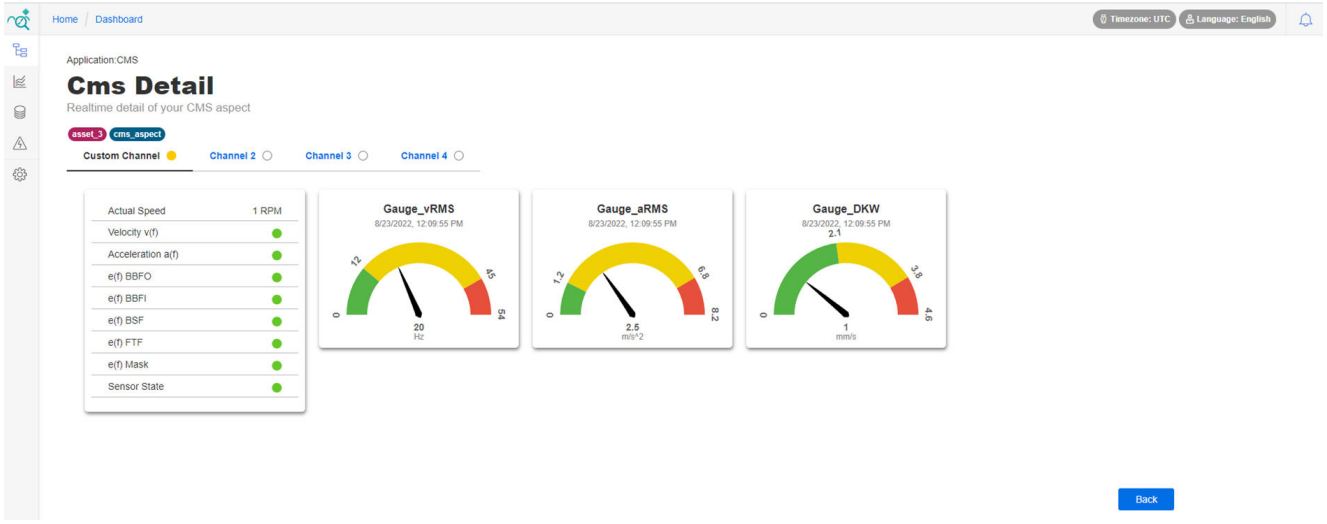


8.3.2 CMS Dashboard Details Page

This section describes the data and graphics that are displayed on the "CMS Detail" page.

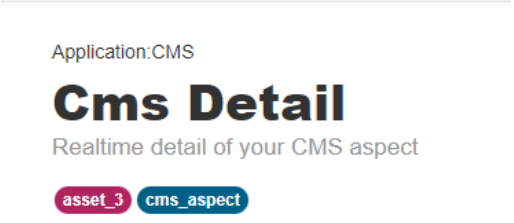
The "CMS Detail" page comprises the following four areas:

- Information on the Dashboard element
- Channel navigation
- Status indicator
- Variable Gauges



Information on the Dashboard element

The CMS type, the name of the selected Asset, and the name of the selected Aspect are displayed for the Dashboard element. This information is retrieved by the Industrial Edge Data Service application.



Channel navigation

The Channel navigation allows switching between data collected by different sensors.

Status indicator

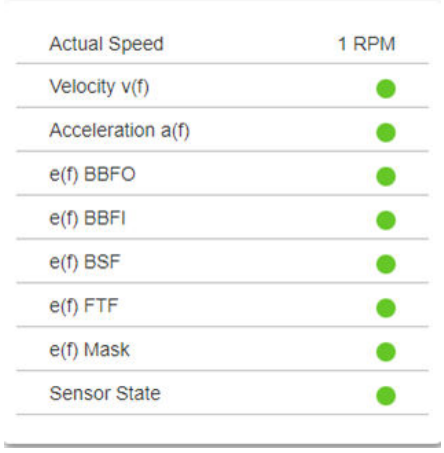
Status indicators show the current speed and status information of the following variables:

- Velocity v(f)
- Acceleration a(f)
- e(f) Mask
- e(f) BBFO
- e(f) BBFI

- e(f) BSF
- e(f) FTF
- Sensor State

The displayed colors have the following meaning:

- Green: Normal status of the variable
- Yellow: Warning status of the variable
- Red: Alarm status of the variable
- Gray: No information is available for the variable



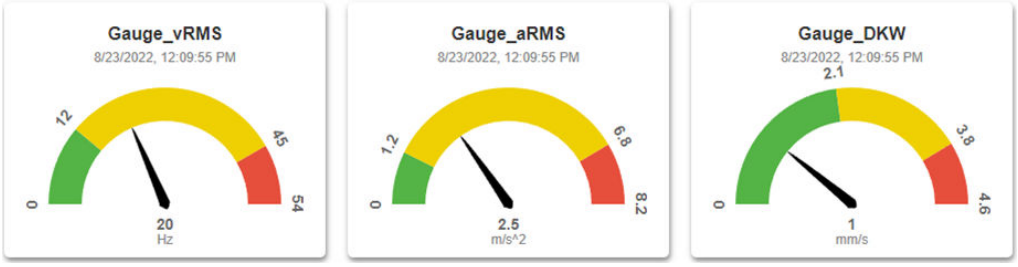
Variable Gauges

The Variable Gauges show information about the following variables:

- vRMS
- aRMS
- DKW

The following information is displayed for each variable:

- Variable name
- Last reading time of variable
- Variable warning level
- Variable alarm level
- Last value of the variable


Gauge_vRMS
8/23/2022, 12:09:55 PM
0
12
45
54
20
Hz
Gauge_aRMS
8/23/2022, 12:09:55 PM
0
1.2
6.8
8.2
2.5
m/s^2
Gauge_DKW
8/23/2022, 12:09:55 PM
0
2.1
3.8
4.6
1
mm/s

8.4 Basic Anomaly Detection

8.4.1 Basic Anomaly Detection Model Parameter

The "Basic Anomaly Detection" Model uses the following drive parameters provided by the " Low-Speed-Adapter" with a resolution of 100 ms:

- R63[0]: Actual speed value
- R80[0]: Actual torque smoothed
- R26 : DC link voltage smoothed
- R62 : Speed setpoint after the filter
- R79 : Torque setpoint
- R35 : Motor temperature
- R37[1] : Power unit temperatures - depletion layer maximum value

8.4.2 Basic Anomaly Detection Model KPIs

The "Basic Anomaly Detection" Model generates the following set of KPIs every 5 minutes for the drive parameters listed in section "Basic Anomaly Detection Model Parameter (Page 53)". The calculated 5-minute KPIs are smoothed with a time resolution, see step 7 in section Inserting a "Use Case" (Page 79).

Level-related KPIs

- Minimum:
  The minimum value of the given parameter within the evaluated period.
- 20 percent percentile:
  The 20 percent percentile is the value below which 20 percent of the values of the given parameter lie within the evaluated period. 80 percent of the values of the given parameter within this period lie above this value.
- Mean value:
  The mean value is the average value of the given parameter within the evaluated period.
- Median:
  The median is the value below which 50 percent of the values of the given parameter lie within the evaluated period. 50 percent of the values of the given parameter within this period lie above this value. It is less influenced by outliers than the mean value.
- 80 percent percentile:
  The 80 percent percentile is the value below which 80 percent of the values of the given parameter lie within the evaluated period. 20 percent of the values of the given parameter within this period lie above this value.

- Maximum:
  The maximum value of the given parameter within the evaluated period.
- Root-mean-square value:
  The root-mean-square value is the square root of the mean value of the squared values of the parameter within the evaluated period. It is especially relevant for alternating signals and provides a robust estimate how the values of the parameter deviate on average from 0.

Spread-related KPIs

- Standard deviation:
  The standard deviation measures the variation in the distribution of the given parameter within the evaluated period. The standard deviation has the same unit as the parameter values. A low standard deviation indicates that the observed values are close to the mean value. A higher standard deviation indicates that the observed values are spread out over a wider range.

Distribution characteristics

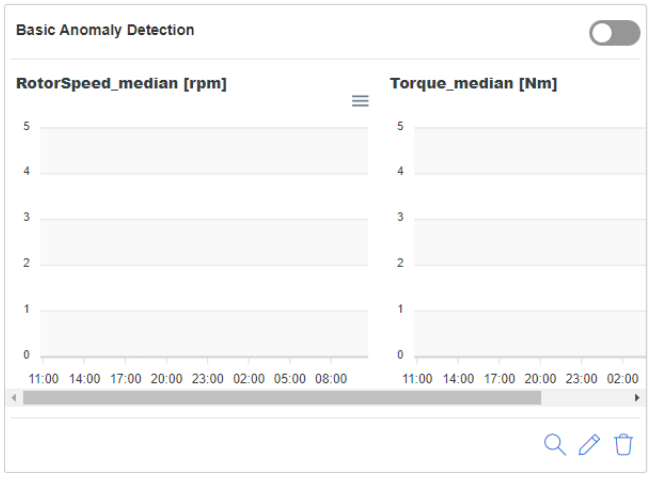
- Kurtosis
  The kurtosis is a measure of the weight of the tails compared to the center of the distribution of the given parameter within the evaluated period.
  A kurtosis value close to 0 means: The weight of the tails of the distribution corresponds to the weight that the tails of a normal distribution have.
  A negative kurtosis value means: The weight of the tails of the distribution is lower than the weight of the tails in a normal distribution.
  A positive kurtosis value means: The weight of the tails of the distribution is higher than the weight of the tails in a normal distribution.
  So the kurtosis indicates if extreme values in the given distribution occur less/more frequent than would be expected for a normal distribution.
- Skewness
  Skewness is a measure of the asymmetry of the distribution of the given parameter within the evaluated period.
  For unimodal distributions (= distributions with one peak value), a negative skewness indicates that there is more weight in the left tail (= parameter values smaller than the most frequently occurring parameter value) of the distribution. Likewise, a positive skewness indicates that there is more weight in the right tail (= parameter values greater than the most frequently occurring parameter value) of the distribution.
  A skewness value close to 0 indicates that the tails around the most frequent parameter value have equal weight. This is always true for symmetric unimodal distributions but can also occur for asymmetric distributions.

8.4.3 Basic Anomaly Detection Overview Widget

This section describes the data and diagrams that are displayed in the Overview Widget for the "Basic Anomaly Detection".

The Overview Widget for "Basic Anomaly Detection" is divided into the following areas:

- Rotorspeed Median [rpm]
- Torque Median [Nm]

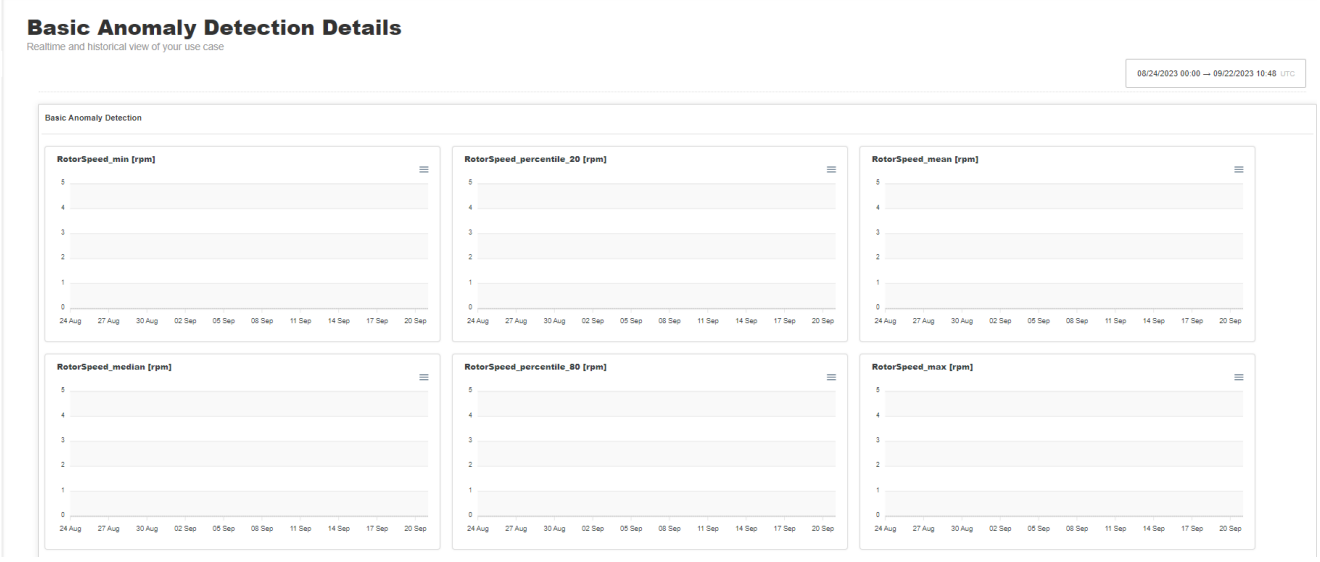


- "RotorSpeed Median [rpm]" diagram
  The "RotorSpeed Median [rpm]" diagram shows the average rotor speed values of the last 24 hours as revolutions per minute.
- "Torque Median [Nm]" diagram
  The "Torque Median [Nm]" diagram shows the average torque values of the last 24 hours in Newton meters.

8.4.4 Basic Anomaly Detection Details Page

This section describes the data and diagrams that are displayed on the "Basic Anomaly Detection Details Page".

The Details Page contains line graphs for each of the KPIs described in the section "Basic Anomaly Detection Model KPIs (Page 53)".



By default, the Details Page displays the data collected in the last 30 days, including the current day's data. You can adjust the time interval for the diagrams or select "Quickrange". See section "Adapting the time interval on the Details Page (Page 93)".

On the Details Page, 10 KPIs are displayed per page. You can navigate back and forth between pages using the page navigation.

Operating Analyze MyDrives /Edge

9

The following sections describe the functions of the Industrial Edge app "Analyze MyDrives /Edge".

Requirement

The following requirements must be met in order to use the Industrial Edge app "Analyze MyDrives /Edge".

- You have activated the "Low-Speed-Adapter" with the required parameters in the Industrial Edge app "Drive System Framework" for the "Energy Efficiency" model. See section Configuring the Drive System Framework Low-Speed-Adapter (Page 23).
- You have configured and activated the "High-Speed-Adapter" with the required parameters in the Industrial Edge app "Drive System Framework" for the "Mechanical Anomaly Detector" model. See section Configuring the Drive System Framework High-Speed-Adapter (Page 25).
- The data of the "Condition Monitoring Systems" must be available in the "Data Service" so that you can set up and monitor "CMS Dashboard" elements.

Work steps

The steps for operating "Analyze MyDrives /Edge" are described below:

- Model Management
  - Inserting a "Model" (Page 59)
  - Activating a "Model" (Page 63)
  - Editing a "Model" (Page 67)
  - Updating a "Model" (Page 70)
  - Deactivating a "Model" (Page 73)
  - Deleting a "Model" (Page 76)
- Use Cases
  - Inserting a "Use Case" (Page 79)
  - Activating a "Use Case" (Page 84)
  - Deactivating a "Use Case" (Page 86)
  - Editing a "Use Case" (Page 88)
  - Deleting a "Use Case" (Page 90)
  - Opening the "Use Case" details page (Page 92)
  - Adapting the time interval on the details page (Page 93)

- CMS Dashboard
  - Inserting a "CMS Dashboard" element (Page 98)
  - Editing a "CMS Dashboard" element (Page 103)
  - Deleting a "CMS Dashboard" element (Page 105)
  - Opening the details page of the "CMS Dashboard" element (Page 107)
- Operational State Filter
  - Inserting an "Operational State Filter" (Page 109)
  - Starting/stopping "Raw Data Collection" (Page 112)
  - Finding suitable raw data for OSF (Page 112)
  - Setting up filters (Page 115)
  - Editing filters (Page 119)
  - Deleting filters (Page 119)
- Mechanical Anomaly Detector Starter Kit (Page 125)

See also

Mechanical Anomaly Detector Starter Kit (Page 125)

9.1 Model Management

This section describes the steps required to create, configure and activate "Models" for use with "Use Cases".

9.1.1 Inserting a "Model"

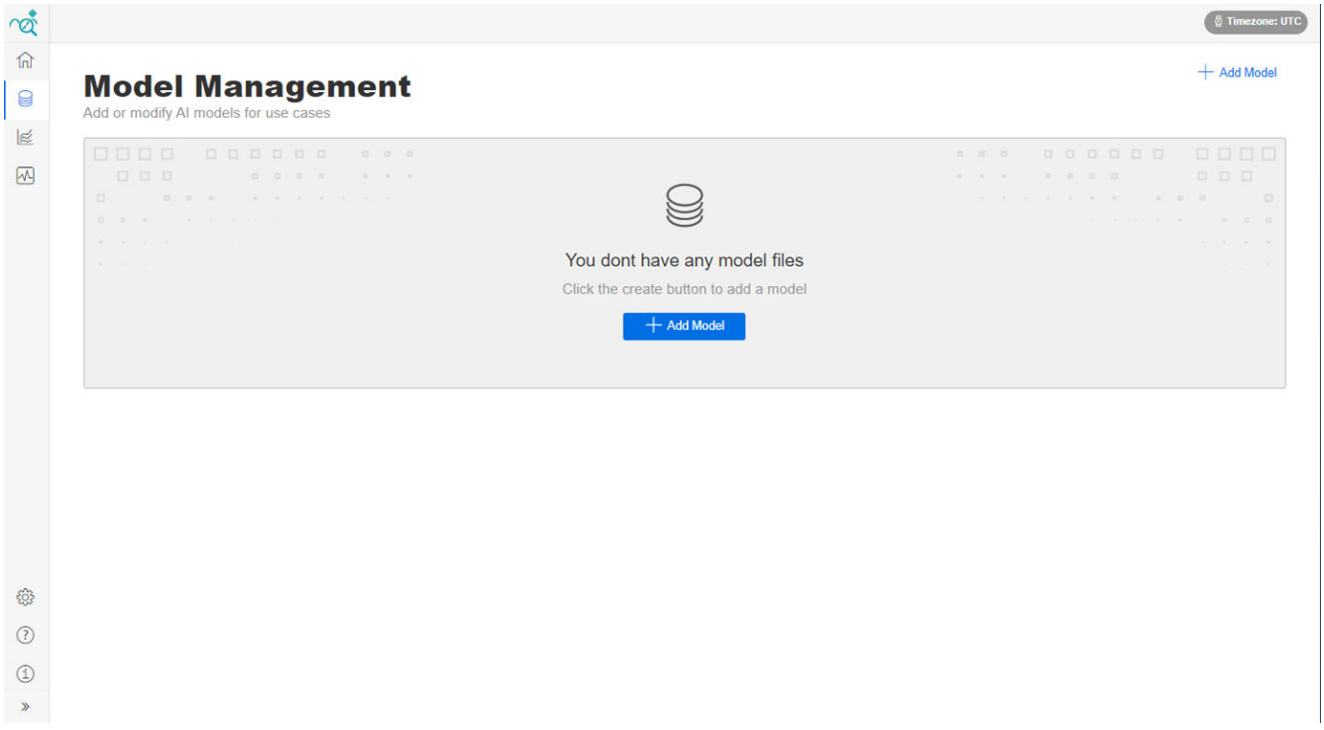
To insert a Models, proceed as described below.

Requirement

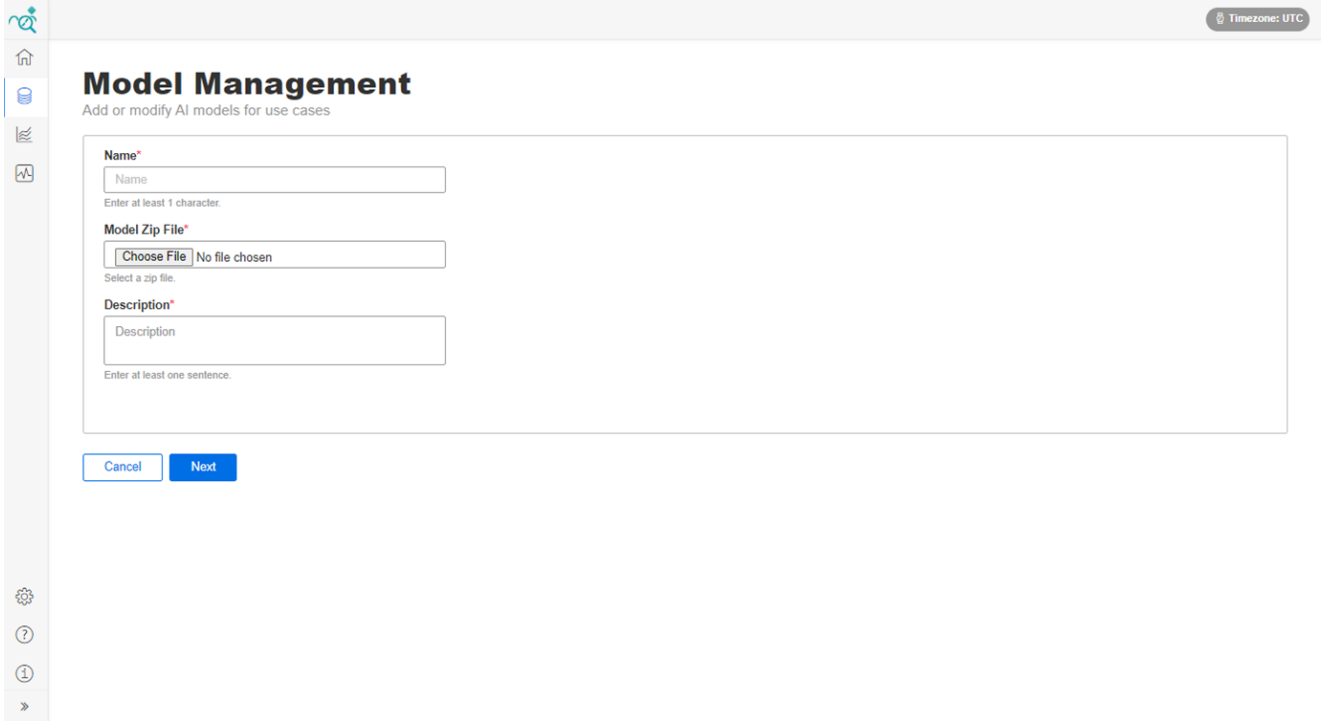
A Model file ("zip" file) for "Energy Efficiency", "Mechanical Anomaly Detector" or "Customizable AI Model" is available.

Procedure

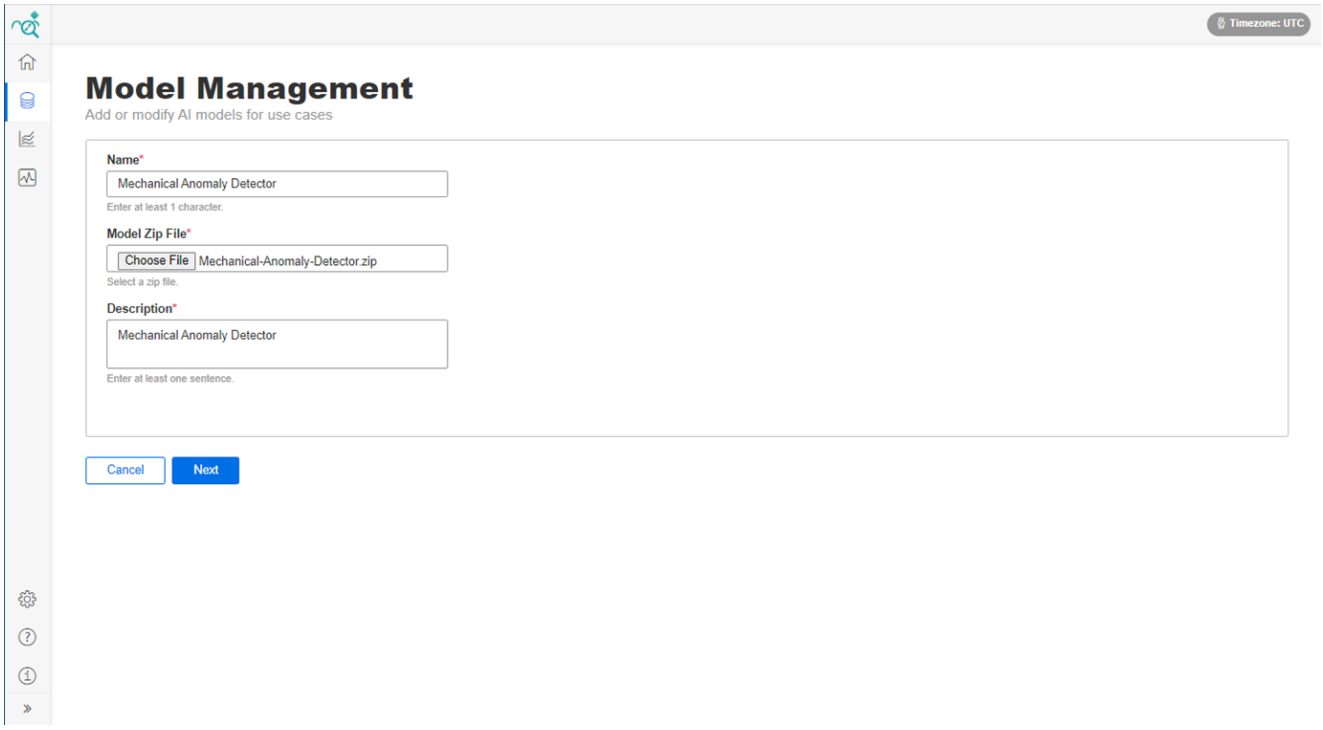
1. In the navigation bar, click on "Model Management". The "Model Management" page is displayed.



2. Click on the "+ Add Model" button (at the top right or in the middle of the screen). The following dialog is displayed.



3. Enter a name for your Model.
   Observe the following rules when entering data:
   - The name of the Models must be entered (mandatory field).
   - The same name cannot be assigned more than once.
   - Only ASCII characters can be used.
   - The name must not be longer than 25 characters.
   - The characters #, +, / and spaces must not be used.
4. Upload the Model file. In the "Help" operating area, you can download Model files for "Energy Efficiency" and "Mechanical Anomaly Detector".
   - In the "Model Zip File" field, click the "Choose File" button to open the dialog for uploading the file.
   - Select a Model Zip File and accept your selection.
   - The selected Model file button is displayed next to the "Choose File" button
   - To change the selected file, repeat the procedure.
5. In the "Description" field, enter a description for your Model.



6. Click the "Next" button. Configuration parameters for the Model are displayed.
   Continue with step 7 to configure the "Energy Efficiency" model.
   Continue with step 8 to configure the "Mechanical Anomaly Detector" model.
   Continue with step 9 to configure the "Basic Anomaly Detection" Model.
   Continue with step 10 to configure the "Customizable AI Model" Model. The parameters should be configured similar to those for the Models "Energy Efficiency", "Mechanical Anomaly Detector" or "Basic Anomaly Detection".

7. The "Energy Efficiency" Model has the following specific configuration parameters:
   - Nominal grid frequency [Hz]
   - Rated power unit line supply voltage [V]
   - Costs per kWh [ct]
   - CO2 emission intensity [g CO2/kWh]

   You can keep the default values of these parameters or change them by editing the actual values. These values can also be changed later when editing the Models. Continue editing with step 10.

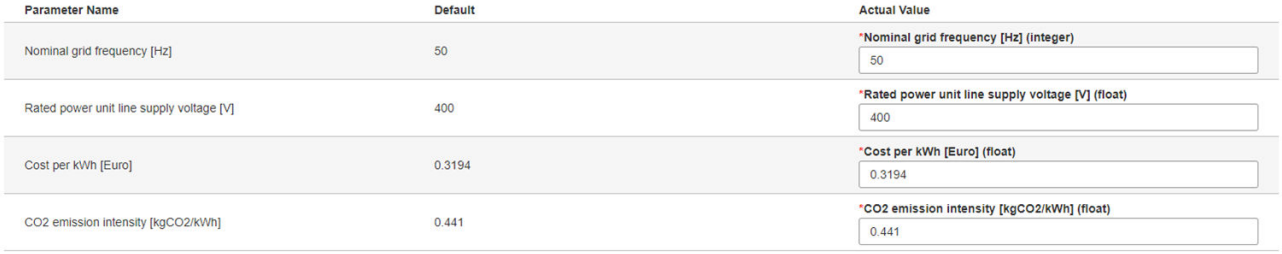
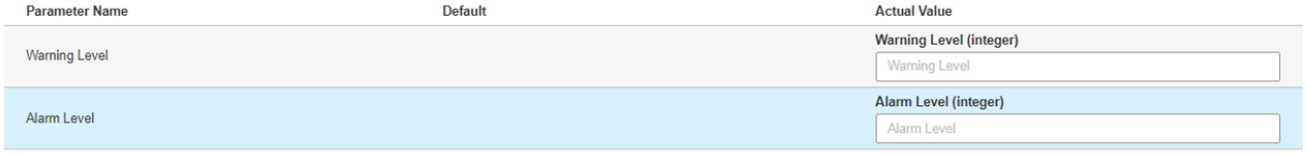


Figure 9-1 Insert model Step 7

8. The "Mechanical Anomaly Detector" Model has the following specific configuration parameters:
   - Warning level: Assumes a value between 0 and 100 and displays the warning level in diagrams
   - Alarm level: Assumes a value between 0 and 100 and displays the alarm level in diagrams

   You can keep the default values of these parameters or change them by editing the actual values. These values can also be changed later when editing the Models.



Note

Depending on the "Model Training", either a low Score or a high Score may indicate an anomaly. If a low Score indicates an anomaly, the alarm level can be set lower than the warning level. The diagrams automatically adapt to this situation.

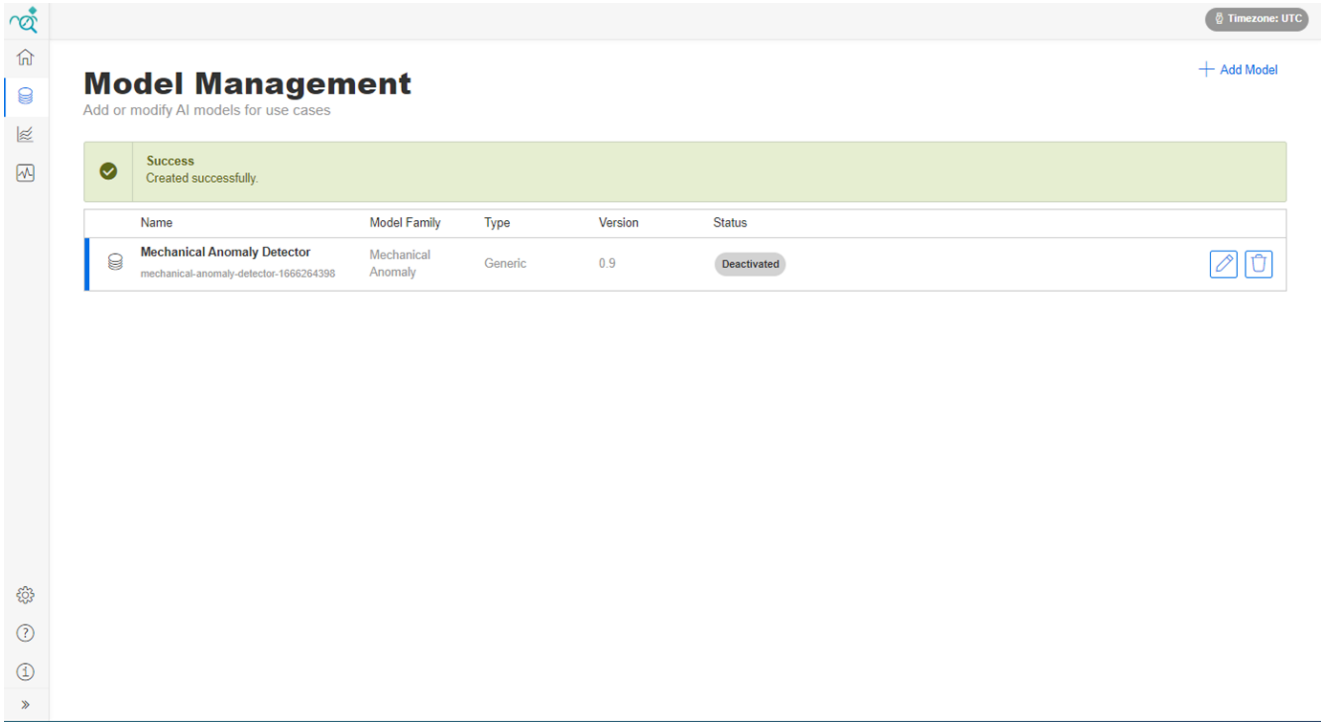
Continue editing with step 10.

9. The "Basic Anomaly Detection" Model has the following specific configuration parameters: Minimum segment length: Assumes values between 0 and 300. This parameter is relevant if the "Basic Anomaly Detection" Model is used with a "Operational State Filter". Only the data segments that meet the defined filter conditions are processed by the Model. These must have at least the minimum segment length (in seconds). Data segments that are shorter than the specified segment length are not processed by the Model.

10. Model "Customizable AI Model" has configuration parameters that are defined under "user_defined_configuration" in file "config.json" in the configuration package. You can keep the default values from file "config.json" when uploading the model. However, you can adapt these corresponding to the actual values. The values can also be adapted at a subsequent point in time, see Section Edit "Model" (Page 67).
    Continue editing with step 10.
11. Click the "OK" button.

Result

The Model has been added to the list of Models. The model is not activated yet.



Note

To be able to use the added Model with a Use Case, you must activate the model. See section Activating a "Model" (Page 63).

9.1.2 Activating a "Model"

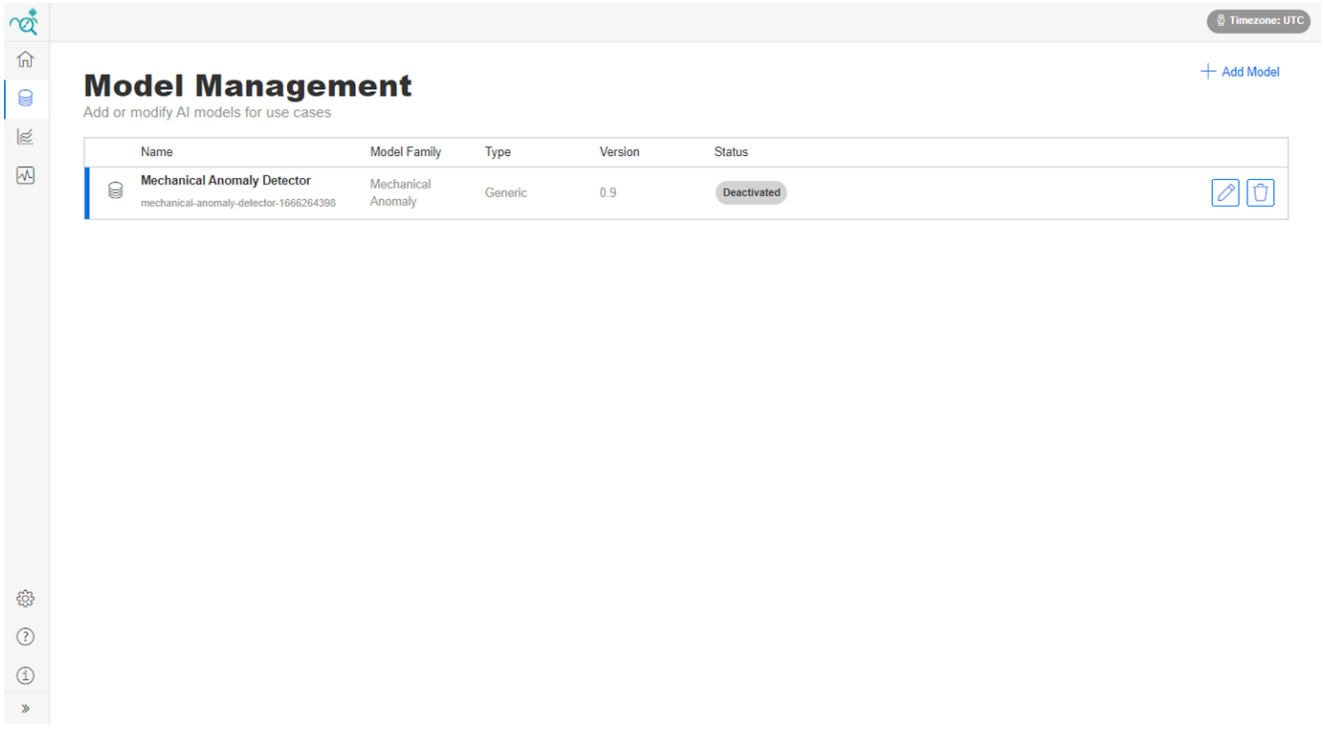
A "Model" must be activated to be able to use it with a "Use Case". Proceed as described to activate an inactive Model.

Requirement

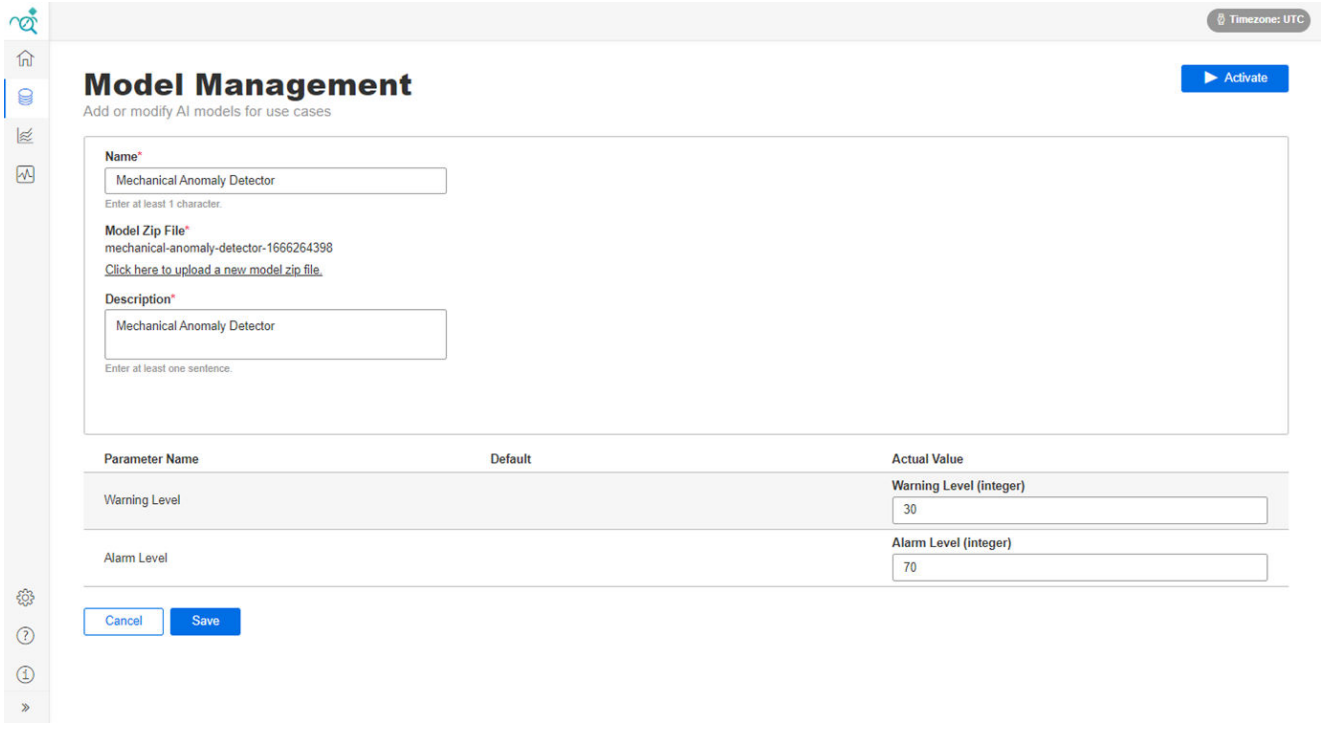
At least one Model is created in the application and the status is inactive.

Procedure

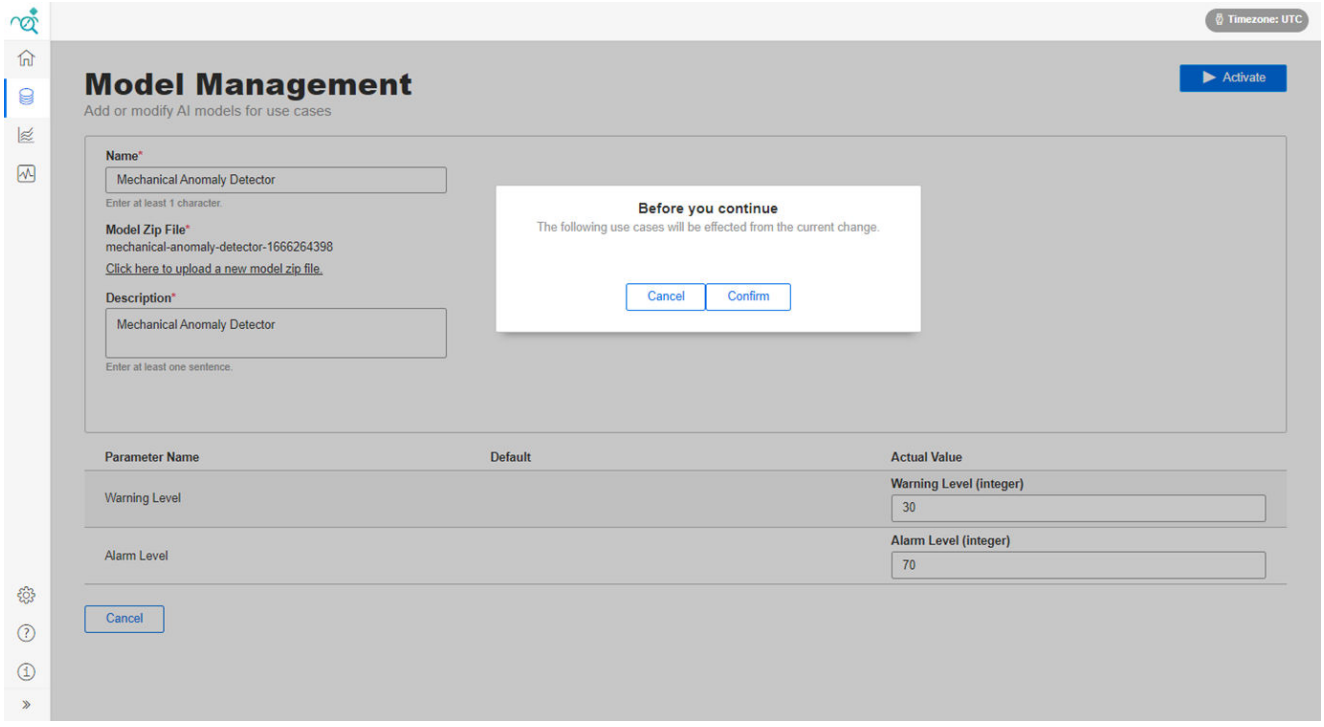
1. In the navigation bar, click on "Model Management". The "Model Management" page is displayed.



2. Click the button of the Model that you want to activate. The following dialog is displayed.



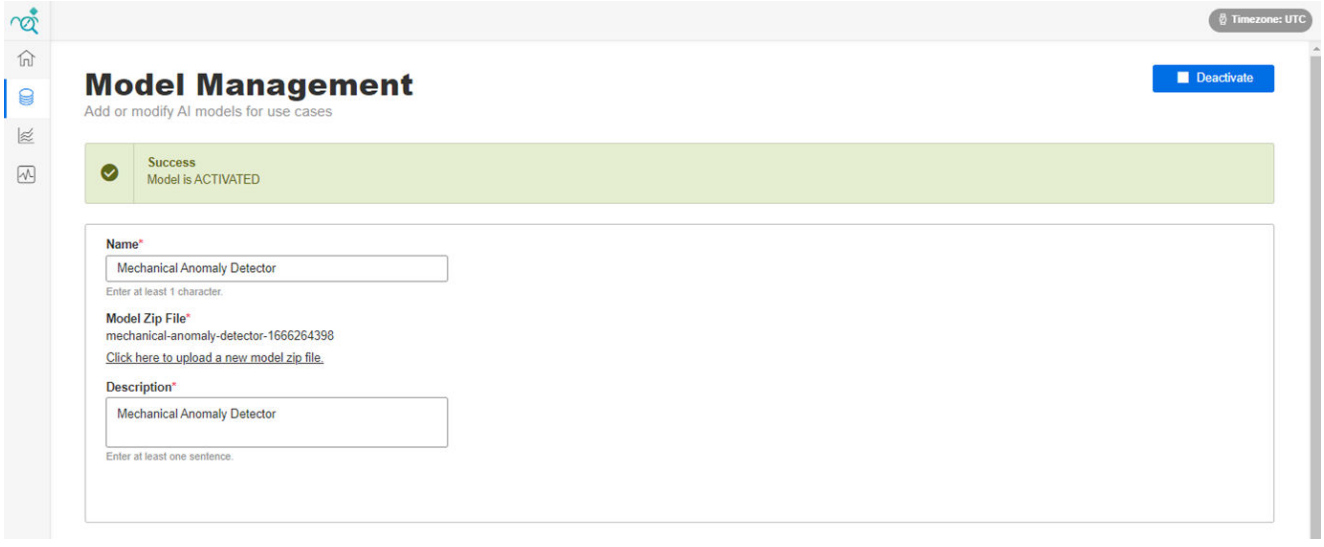
3. Click the "Activate" button. A message with the list of affected active "Use Cases" is displayed.



4. Click on "Confirm" to continue.

Result

The Model is activated.



9.1.3 Editing a "Model"

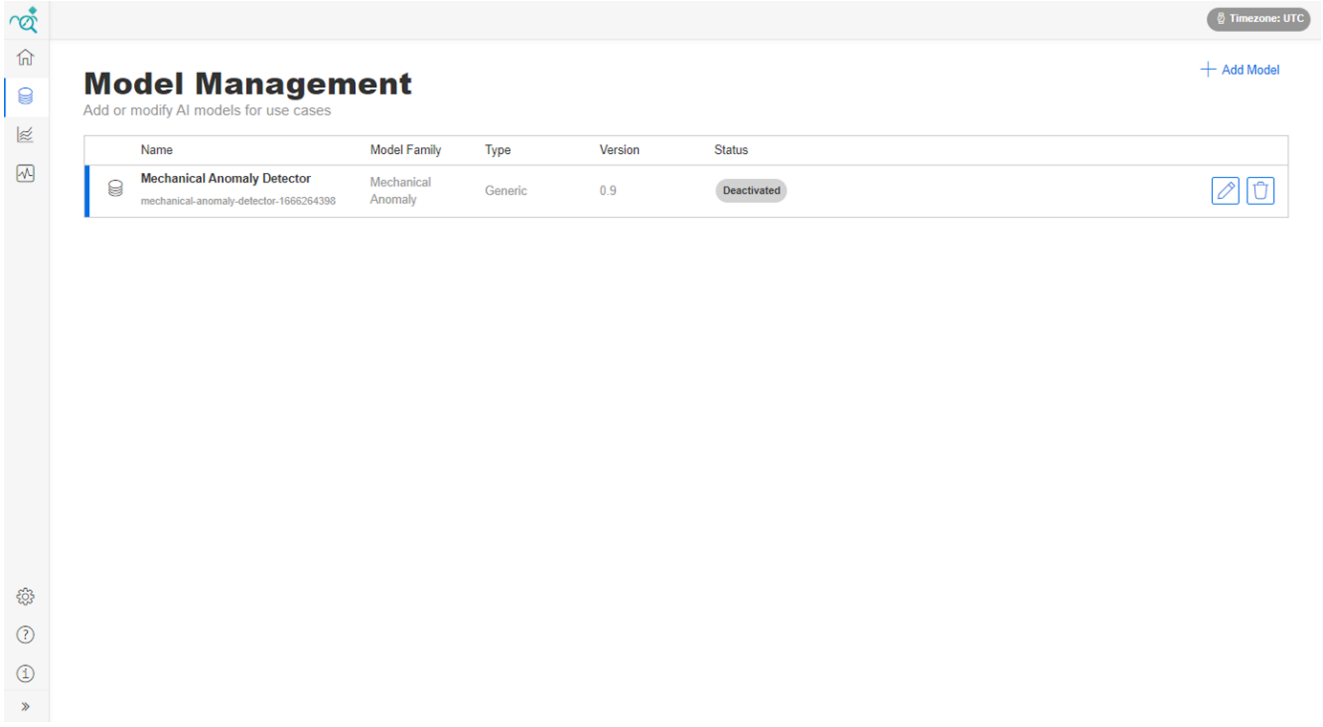
Proceed as described to change the name, description, or configuration parameters of a "Model".

Requirement

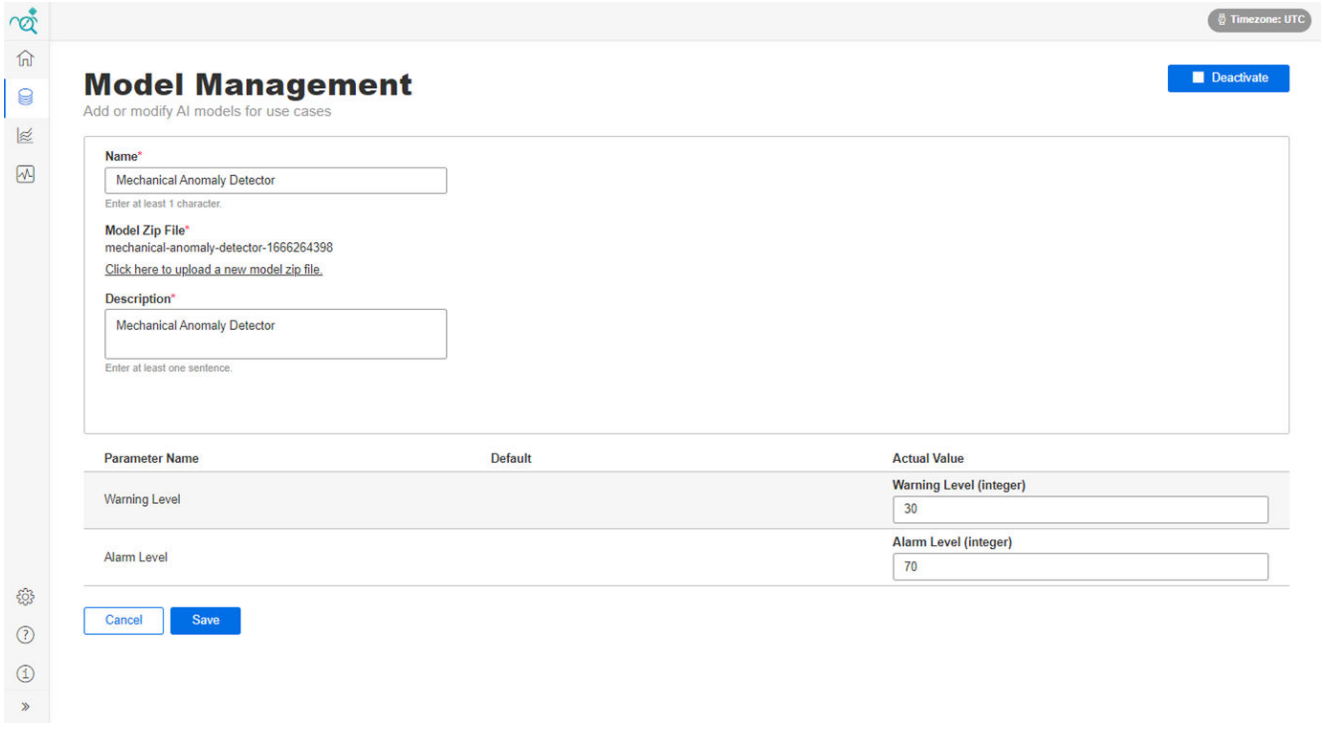
At least one Model is created in the application.

Procedure

1. In the navigation bar, click on "Model Management". The "Model Management" page is displayed.

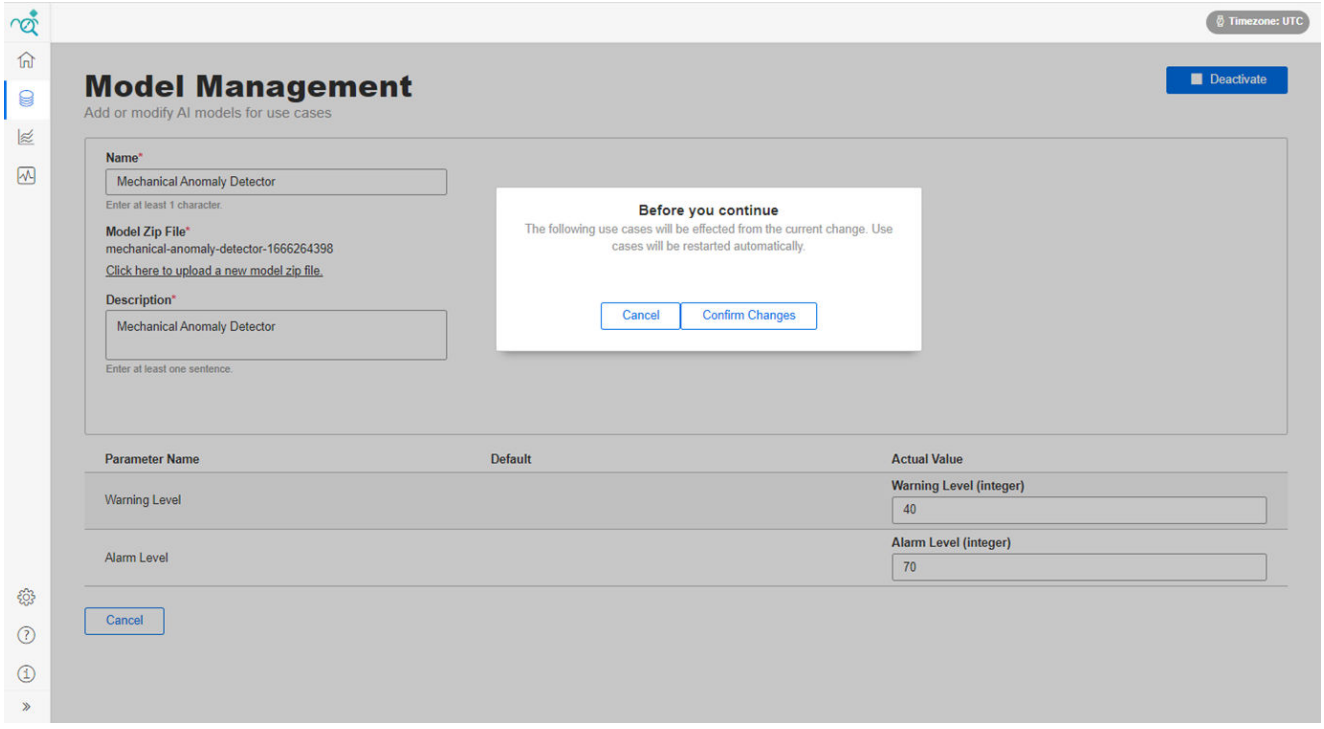


2. Click the button of the Model that you want to edit. The following dialog is displayed.



3. You can change the name, description, or configuration parameters of the Model.

4. Click the "Save" button. A message with the list of affected active "Use Cases" is displayed.



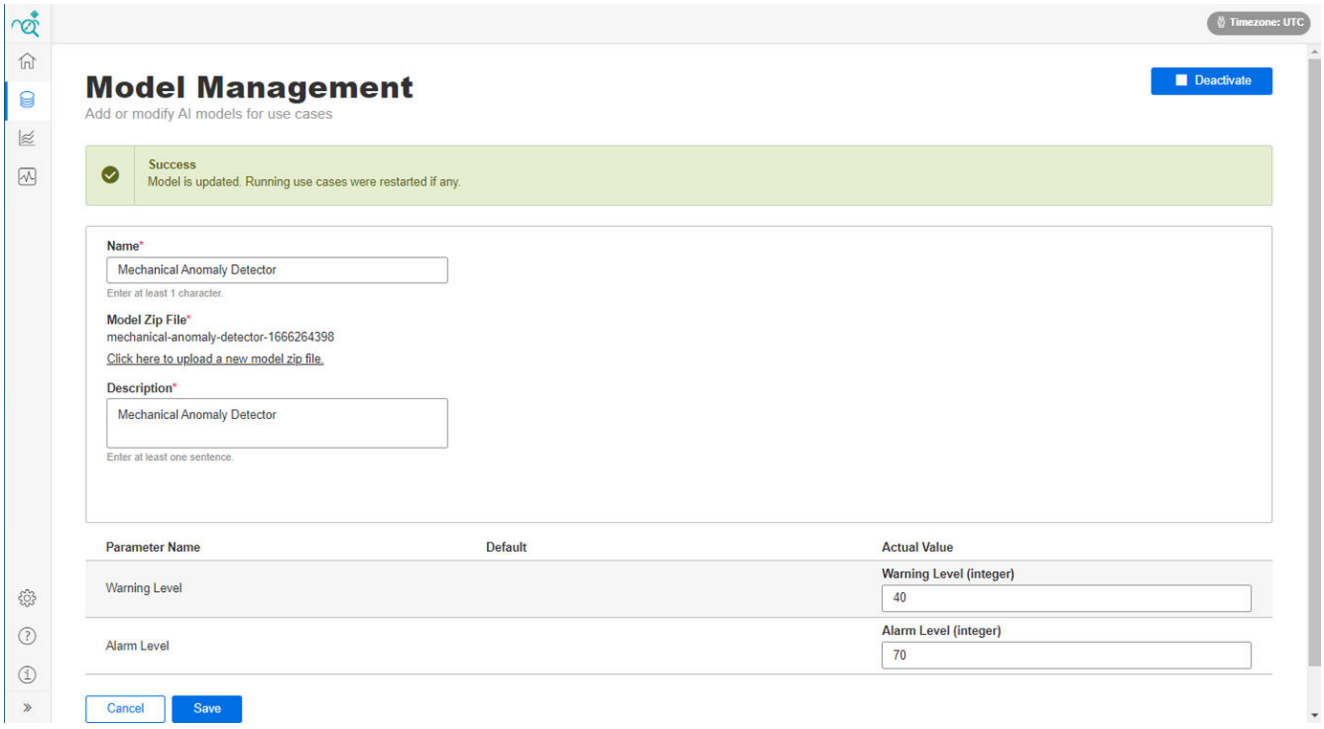
5. If you want to proceed with the changes, click "Confirm Changes". If you want to discard your changes, click "Cancel".

Note

The changes in the "Model" will take effect immediately and associated active "Use Cases" are executed with the updated configuration.

Result

After confirming the changes, the new configuration is applied.



9.1.4 Updating a "Model"

When a new version of a "Model" file is available or a "Mechanical Anomaly Detector" Model is trained with new data, the "Model" can be updated by uploading the updated "zip" files. When the update is completed, all active "Use Cases" that refer to the selected Model will be executed with the updated version.

Note

- In this version of "Analyze MyDrives /Edge", the update of Models is supported with limitation. If the new "Model" files contain new input parameters or new KPIs, add the "Model" as a new "Model".
- You can only update "Models" that belong to the same family. A "Mechanical Anomaly Detector" Model can only be updated with another "Mechanical Anomaly Detector" Model.

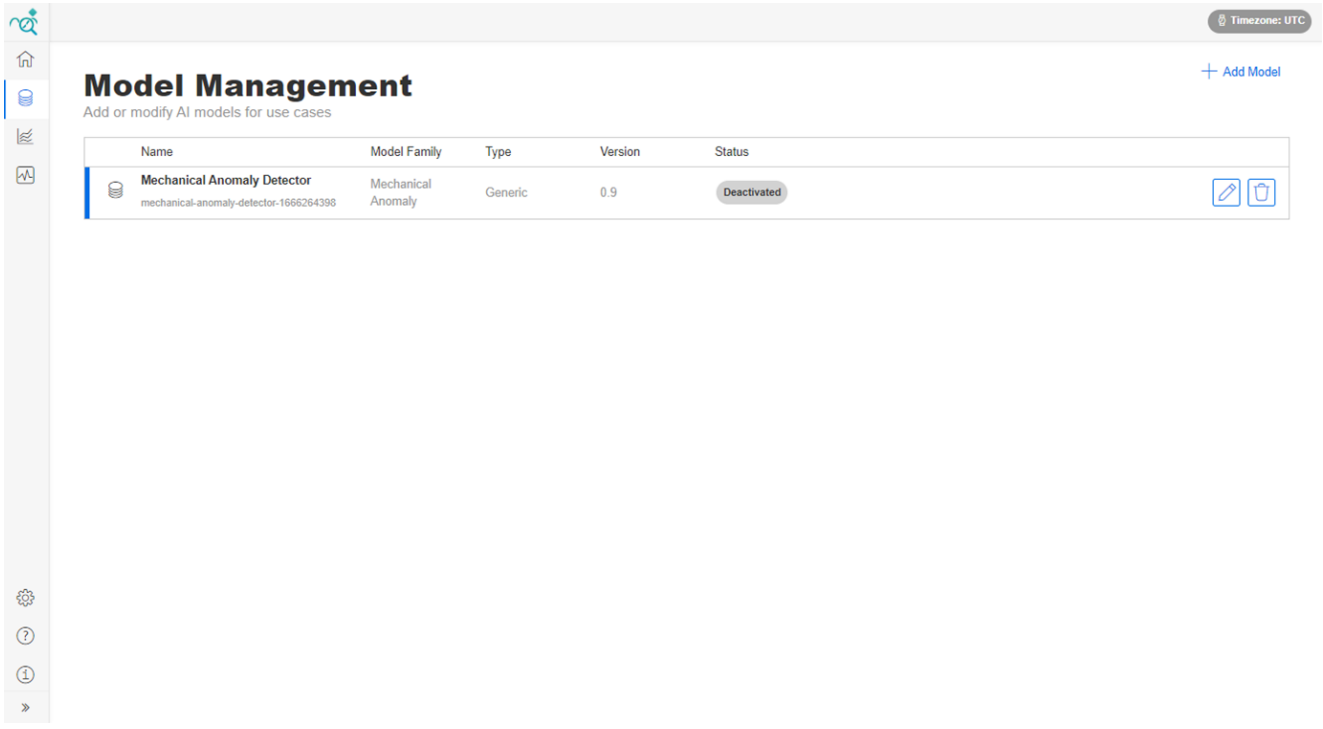
Proceed as described below to activate the "Model".

Requirement

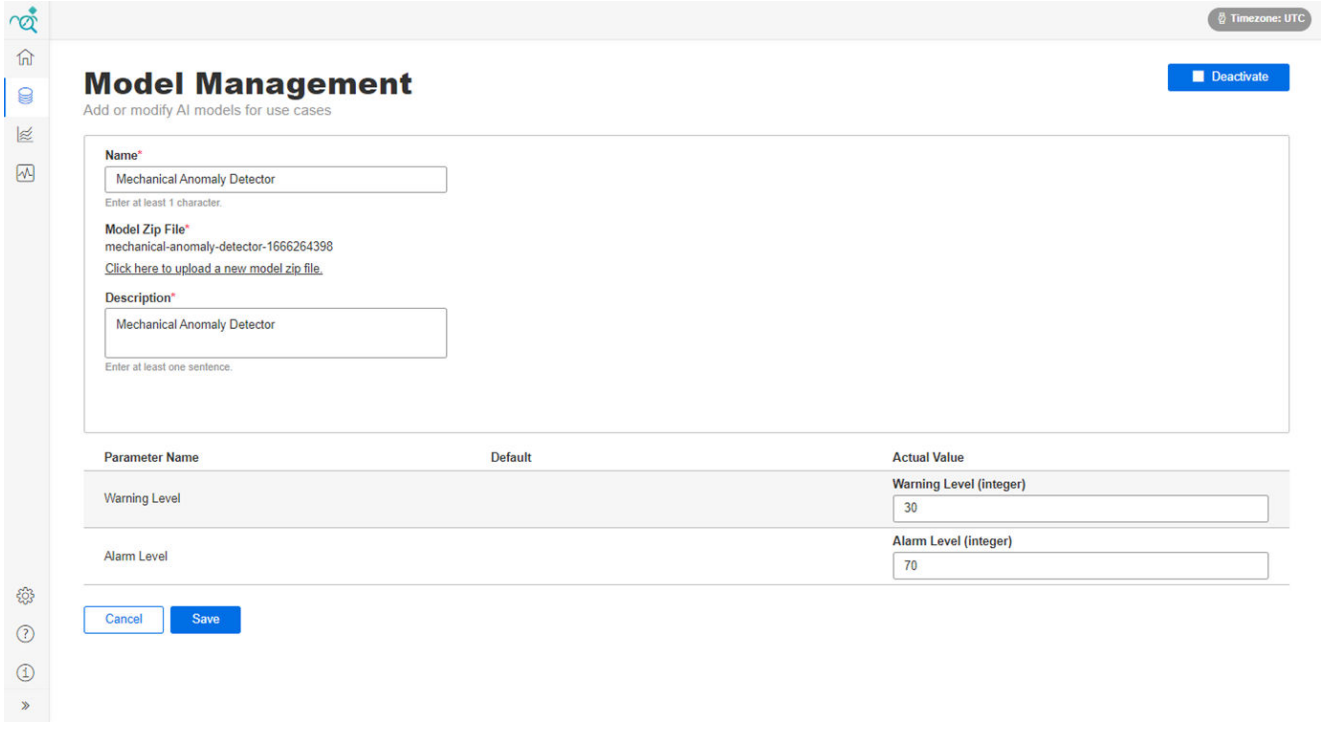
At least one Model is created in the application.

Procedure

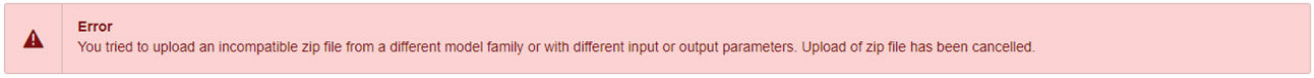
1. In the navigation bar, click on "Model Management". The "Model Management" page is displayed.



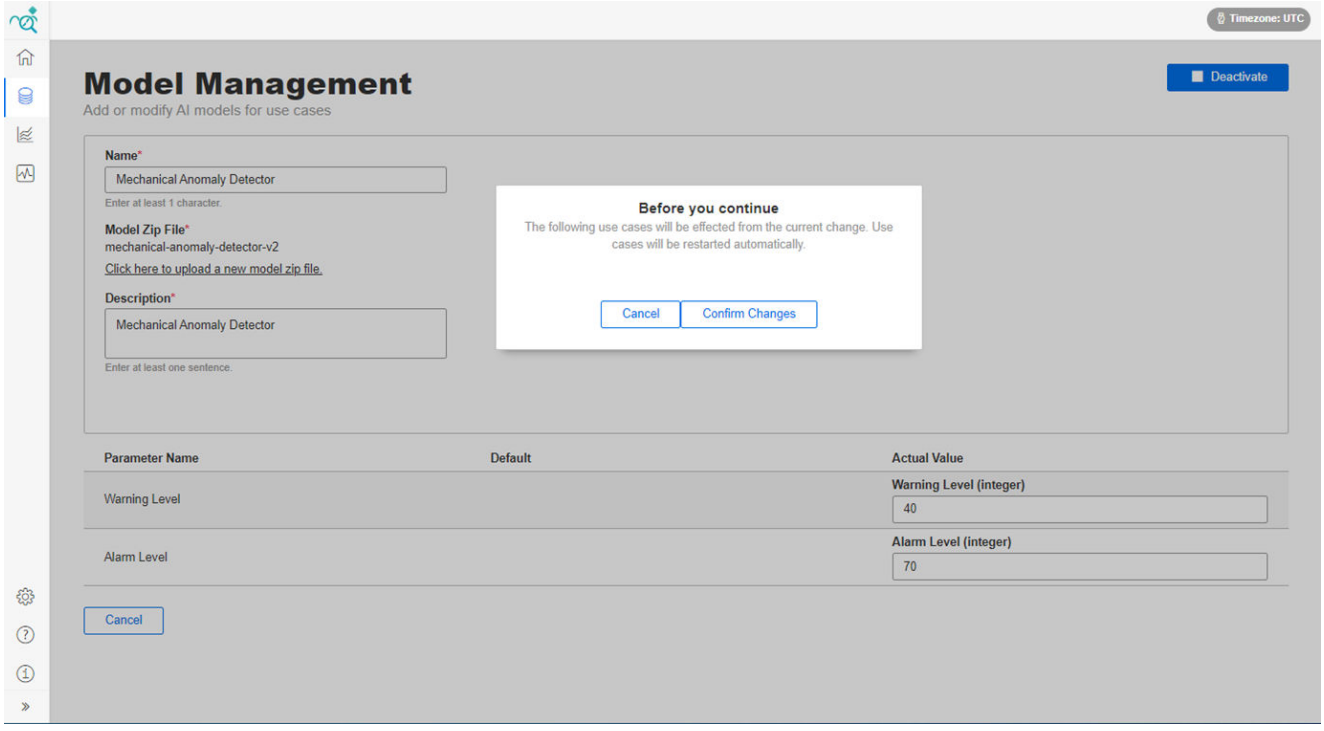
2. Click the button of the Model that you want to activate. The "Edit" dialog is displayed.



3. Under "Model Zip File", click the link "Click here to upload new model zip file". Then click on "Choose File".
4. Select the new "zip" file and click on "Next".
5. If the selected file is not compatible with the current "Model" file, the following error message will be displayed and the update is cancelled.



6. If the selected file is compatible with the current "Model" file, the configuration parameters will be displayed. Click on "Save" to apply the updated "Model".
   A message with the list of affected active "Use Cases" is displayed.



7. If you want to proceed with the update, click "Confirm Changes". If you want to discard the update, click "Cancel".

Note

The changes in the "Model" will take effect immediately and associated active "Use Cases" are executed with the updated configuration.

Result

After confirming the update, the updated "Model" is applied to the corresponding "Use Cases".

9.1.5 Deactivating a "Model"

Proceed as described below to deactivate the "Model".

Note

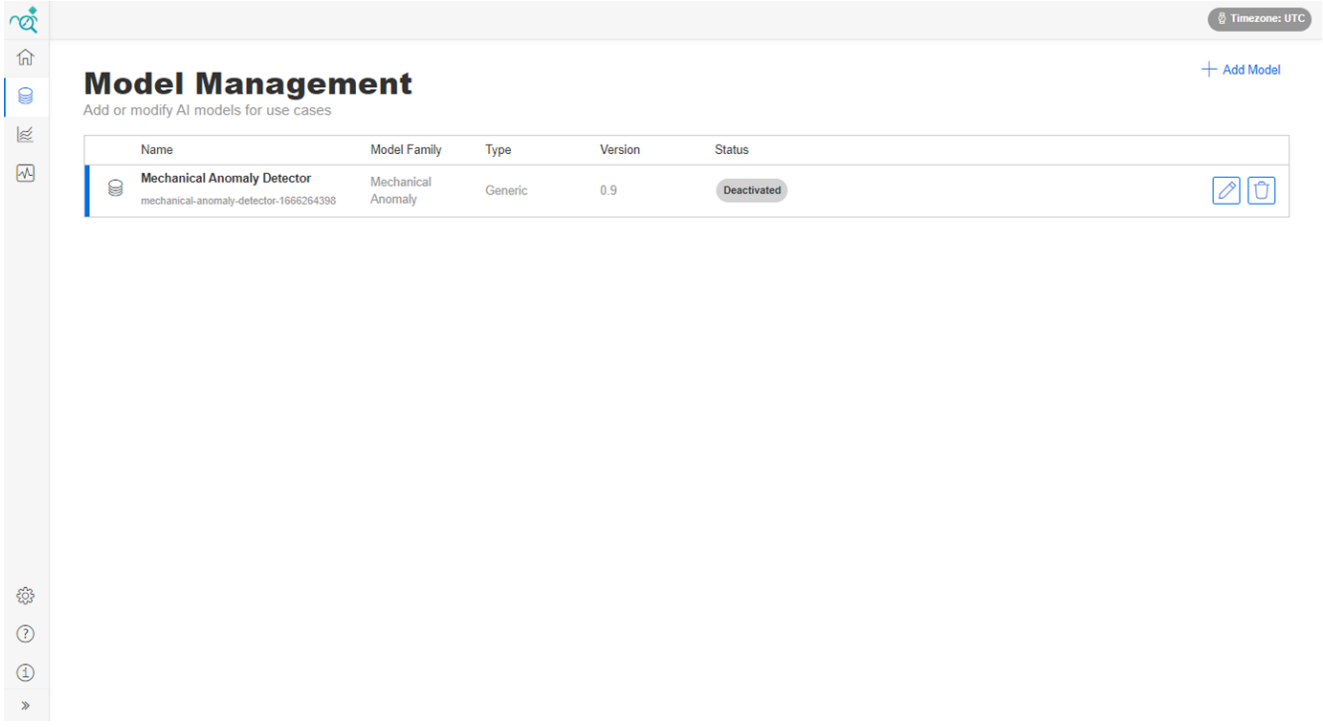
- When you deactivate a "Model", the associated active "Use Cases" are deactivated. For the analysis of the historical KPIs, deactivated "Use Cases" are available, but no new KPIs are created for these "Use Cases".
- The deactivated "Use Cases" can be reactivated after reactivating the corresponding "Model".

Requirement

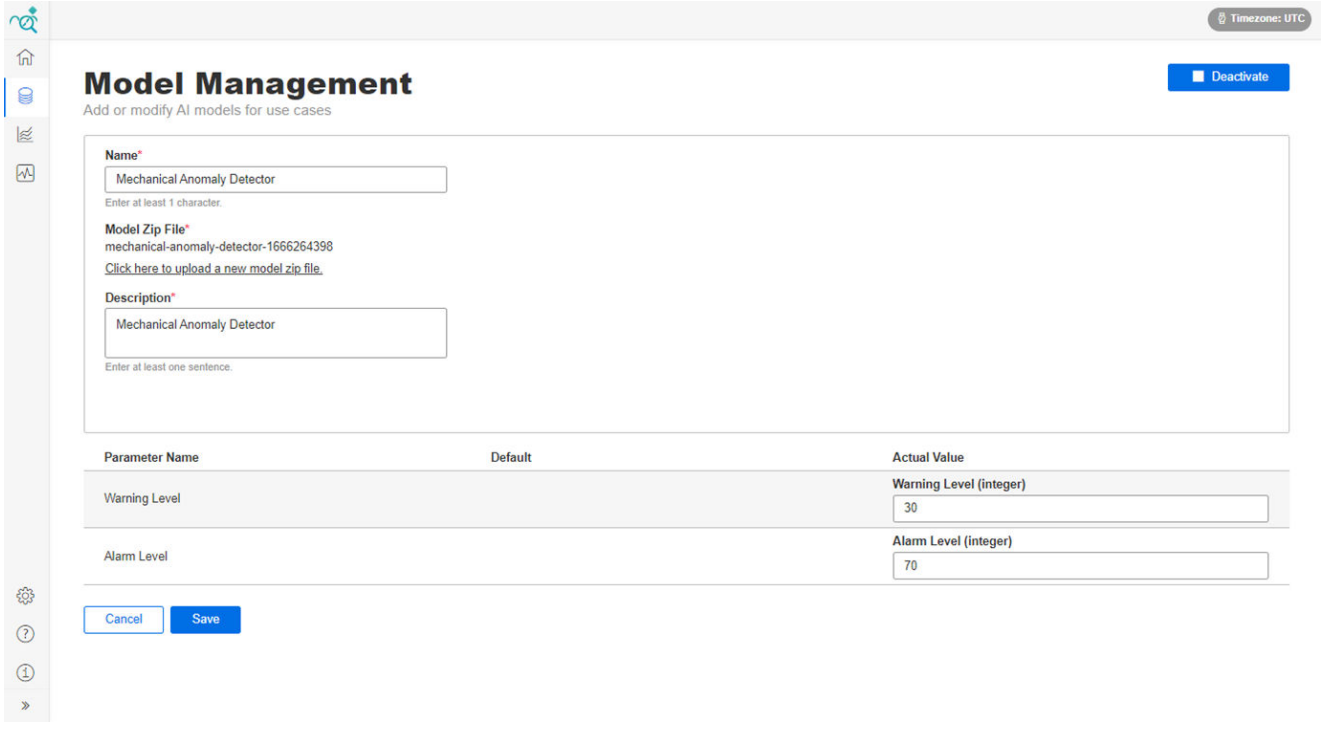
At least one Model is created in the application.

Procedure

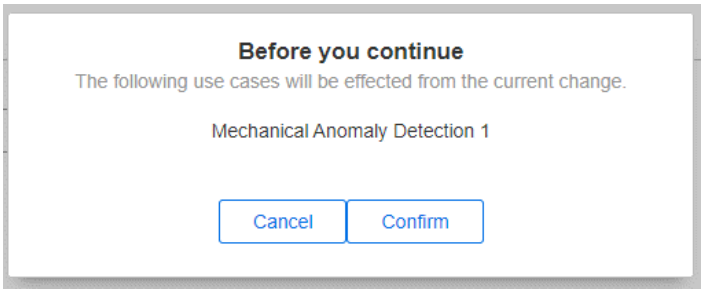
1. In the navigation bar, click on "Model Management". The "Model Management" page is displayed.



2. Click the button of the Model that you want to deactivate. The "Edit" dialog is displayed.



3. Click the "Deactivate" button. A message with the list of affected active "Use Cases" is displayed.



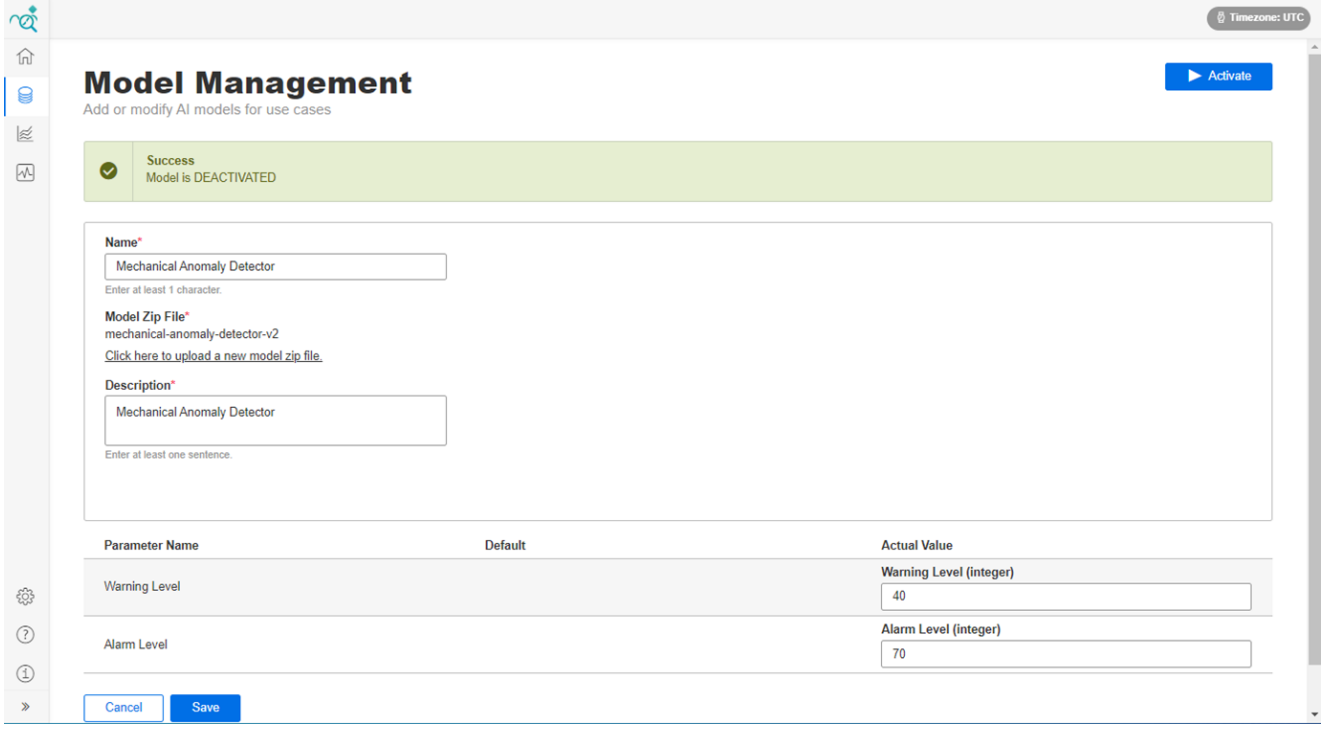
4. If you want to proceed with the deactivation, click "Confirm". If you want to cancel the deactivation, click "Cancel".

Note

The deactivation of the "Model" will take effect immediately and associated active "Use Cases" are deactivated.

Result

After confirmation with "Confirm", the "Model" and the affected "Use Cases" are deactivated.



9.1.6 Deleting a "Model"

Proceed as described below to delete a "Model".

Note

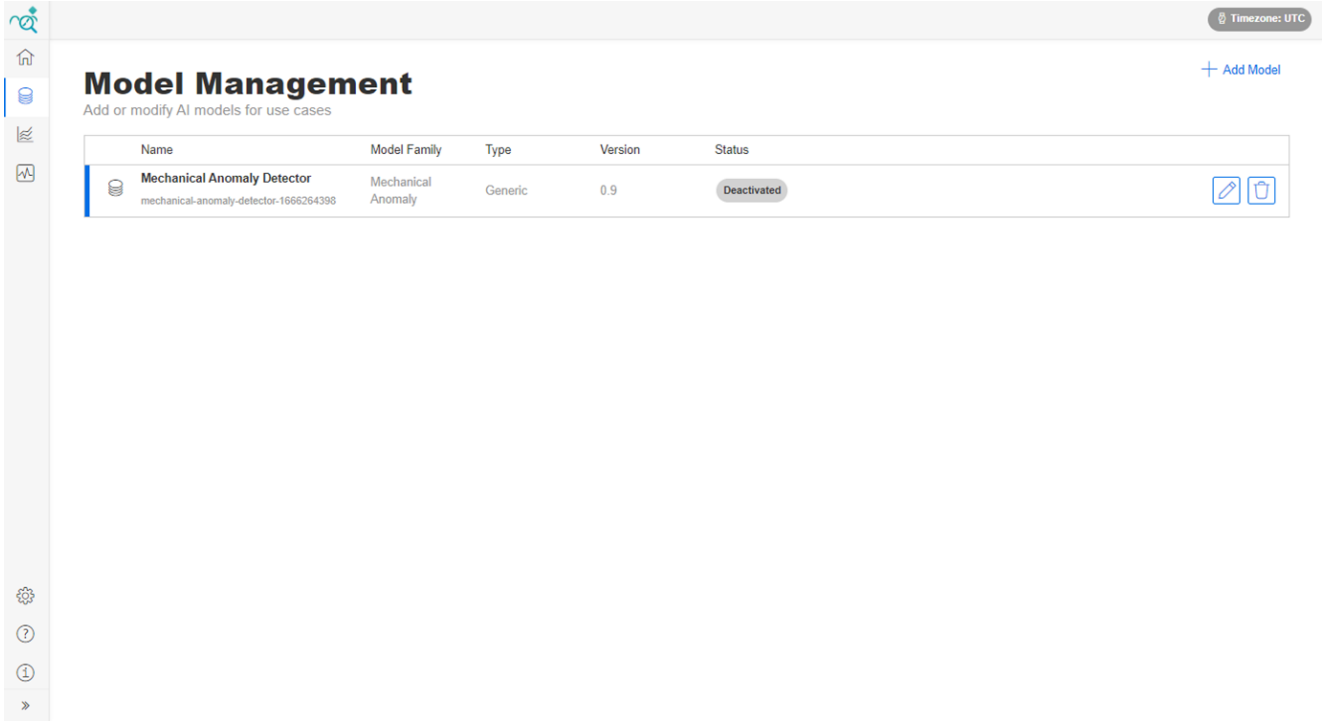
An active "Model" cannot be deleted directly, you have to deactivate the "Model" first. See section Deactivating a "Model" (Page 73).

Requirement

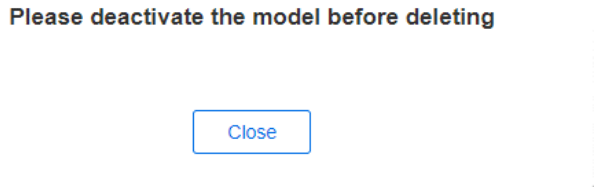
At least one inactive Model has been created in the application.

Procedure

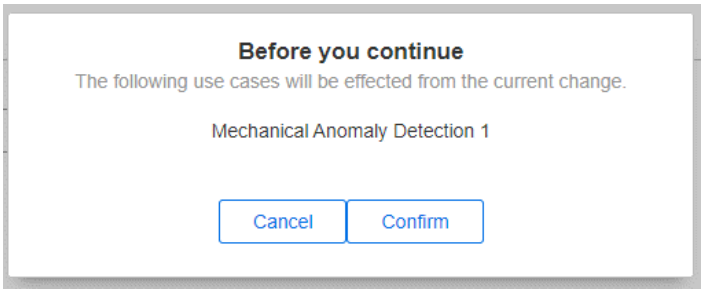
1. In the navigation bar, click on "Model Management". The "Model Management" page is displayed.



2. Click the button of the Model that you want to delete.
If the selected Model is still active, the following message is displayed and the deletion process is aborted.



Otherwise, a message is displayed with the list of affected "Use Cases".



3. If you want to proceed with the deletion process, click "Confirm". If you want to cancel the deletion process, click "Cancel".

Note

The deletion of the "Model" becomes effective immediately. With the "Use Cases" referring to this "Model", you can then only access historical data. You cannot reactivate the "Use Cases".

Result

After confirmation with "Confirm Delete", the "Model" is permanently deleted.

9.2 Use Cases

This section describes the steps required to create, configure and activate "Use Cases".

See also

Energy Efficiency Model (Page 39)

Mechanical Anomaly Detector Model (Page 44)

9.2.1 Inserting a "Use Case"

Proceed as described below to insert a "Use Case".

Note

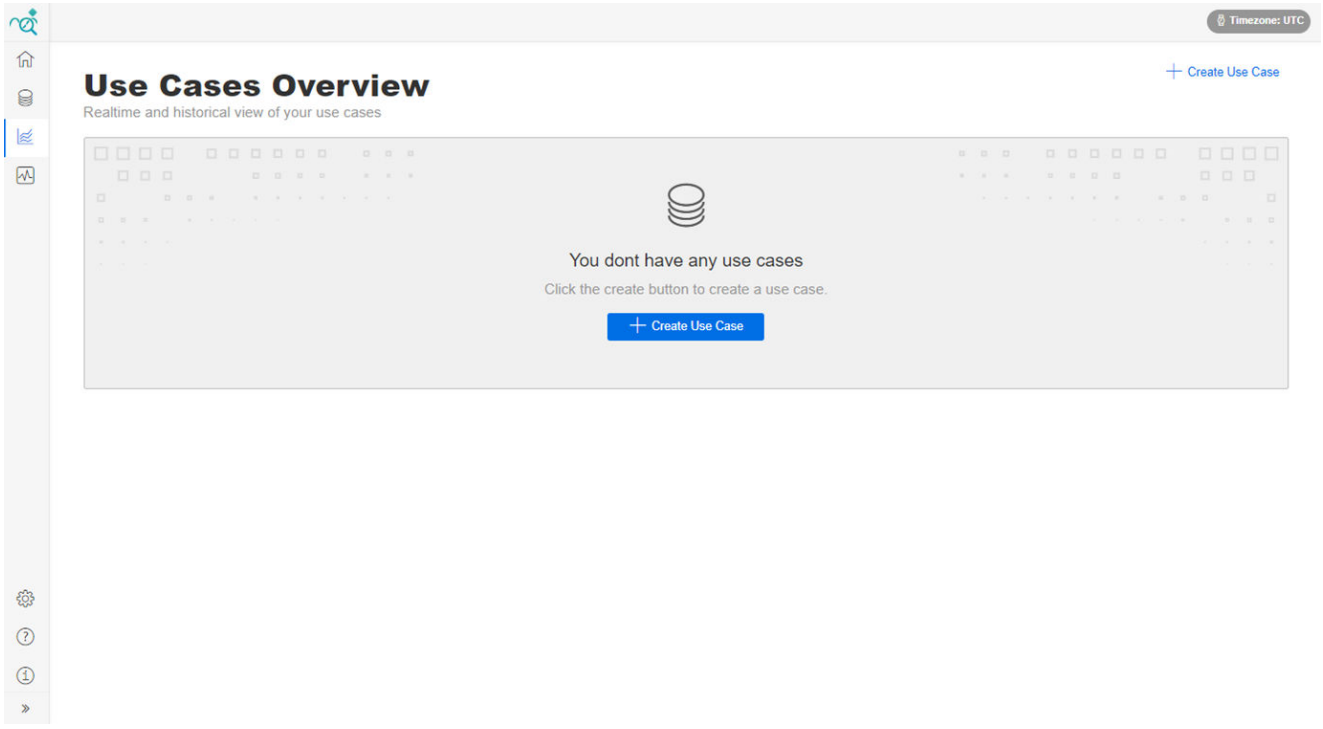
- Only "Low-Speed-Adapter" of the type "Data Sources" can be used for the "Energy Efficiency" and "Basic Anomaly Detection" Models.
- Only "High-Speed-Adapter" of the type "Data Sources" can be used for the "Mechanical Anomaly Detection" Model.
- Depending on the model configuration, "High-Speed-Adapter" or "Low-Speed-Adapter" type "Data Source" can be used for "Customizable AI Model".
- Once a "Use Case" has been created, the "Model" and "Data Source" can no longer be changed.
- A maximum of 16 "Uses Cases" can be added.

Requirement

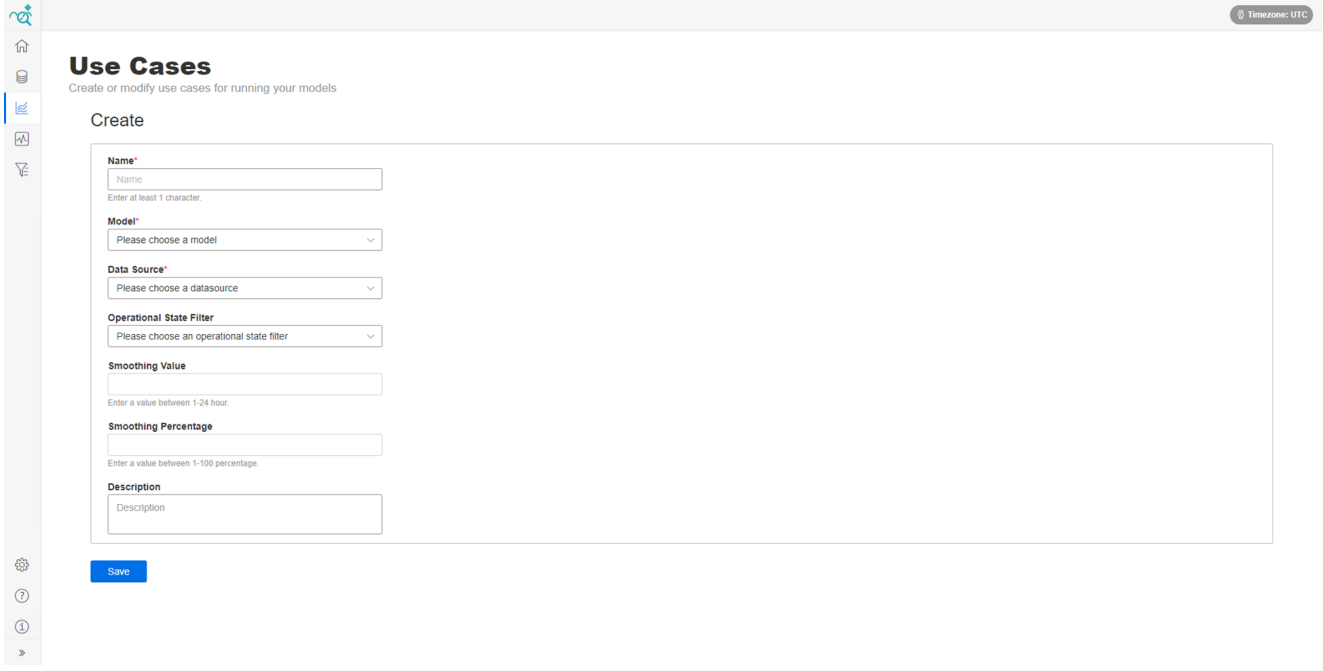
At least one active Model is available in the application. See sections Inserting a "Model" (Page 59) and Activating a "Model" (Page 63).

Procedure

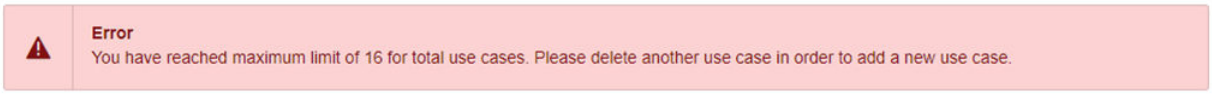
1. In the navigation bar, click on "Use Cases". The "Use Cases Overview" page is displayed.



2. Click on the "+ Create Use Case" button (at the top right or in the middle of the screen). The following dialog is displayed.



Note that a maximum of 16 "Use Cases" are possible. This includes both active and inactive "Use Cases". If the number of 16 "Use Cases" is exceeded, the following error message is displayed.



3. Enter a name for your "Use Case".
   Observe the following rules when entering data:
   - The name of the "Use Cases" must be entered (mandatory field).
   - The same name cannot be assigned more than once.
   - Only ASCII characters can be used.
   - The name must not be longer than 25 characters.
   - The characters #, +, / and spaces must not be used.
4. In the "Model" field, select a "Model" from the drop-down list. Only active "Models" are displayed in this list.
5. In the "Data Source" field, select a "Data Source" from the drop-down list. This list contains "Data Sources" provided by the "Drive System Framework". Please check your "DSF" configuration if a "Data Source" is missing.

Note

- The "Data Source" name has the following format:
- [Drive Name]-[Drive Object Name]-[Adapter Type]
- [Drive Name] shows the name of the drive as it is configured in "DSF"
- [Drive Object Name] shows the name of the drive object as it is configured on the drive for SINAMICS S drives. For SINAMICS G drives, the [Drive Object Name] is always DO#1.
- [Adapter Type] shows the name of the adapter configured in "DSF" for this "Adapter Type". The abbreviation LS stands for "Low-Speed-Adapter", and HF stands for "High-Speed-Adapter"
- Examples of "Data Source" names:
  - S120-SERVO2-HF
  - G120-DO#1-LS
- If the selected Model is an instance of the Model "Energy Efficiency" or "Basic Anomaly Detection", you must select a "Low-Speed-Adapter" of the type "Data Source".
- If the selected Model is an instance of the Model "Mechanical Anomaly Detector", you must select a "High-Speed-Adapter" of the type "Data Source".

6. Select an "Operational State Filter" if you want to filter out input parameters from the selected "Data Source".

Note

- The selection of "Operational State Filter" is optional.
- The selected Model must support filtering in order to be able to use the "Operational State Filter". Currently only the "Basic Anomaly Detection" Model supports this function.

More information on setting up a filter can be found in section "Operational State Filter (Page 109)".

7. Enter "Smoothing Value" and "Smoothing Percentage" if you want to use smoothing for the output. The smoothing function aggregates the KPIs produced by the Model by averaging them. Furthermore, aggregated values are filtered out if the created KPIs in the selected smoothing time window are below a certain threshold.

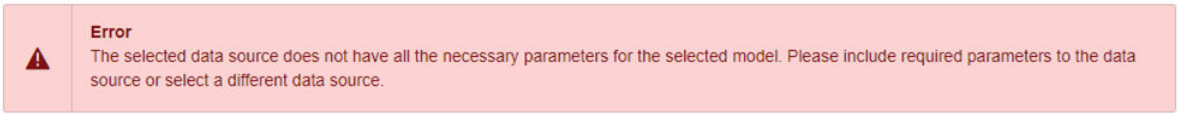
   – "Smoothing Value": This value specifies the time window in hours for smoothing. Values between 1 and 24 are possible.
   Example:
   The "Smoothing Value" is 4. This means that every 4 hours AMD /Edge aggregates the KPIs generated by the Model. The values are averaged and a single value is stored for each 4-hour period.

   – "Smoothing Percentage": Average aggregation is used in smoothing. With this approach, if the number of values created for a KPI is below a threshold, the smoothed values can be misleading. By entering a "Smoothing Percentage" between 1 and 100, smoothed values with low sampling rates can be filtered out.
   Example:
   "Basic Anomaly Detection" is executed every five minutes and creates up to 12 values for each KPI. A "Smoothing Value" of 4 hours and a "Smoothing Percentage" of 50 results in the following values:
   Maximum possible number of KPIs = 4 x 12 = 48
   Required minimum number of KPIs for smoothing = 48 x 0.5 = 24
   In this case, if the Model has generated less than 24 values for a characteristic value during a 4-hour window, the smoothed value for this window is evaluated as non-significant and is not displayed in the "Overview Widget" and on the "Details Page". If the Model has generated ≥ 24 values, this time window is evaluated as significant and the smoothed value is displayed in the "Overview Widget" and on the "Details Page".

Note

- Smoothing is not mandatory.
- You cannot change the "Smoothing Value" after the "Use Case" has been created.
- You can change the "Smoothing Percentage" even after the "Use Case" has been created.
- If smoothing is defined, only smoothed values are displayed on the "Overview Widget" and "Details Page" of the "Use Case".

8. Enter a description for your model.
9. Click the "Save" button. The application checks whether the selected "Model" and the "Data Source" are compatible.
   If the selected "Model" and the "Data Source" are not compatible, the following message will be displayed.

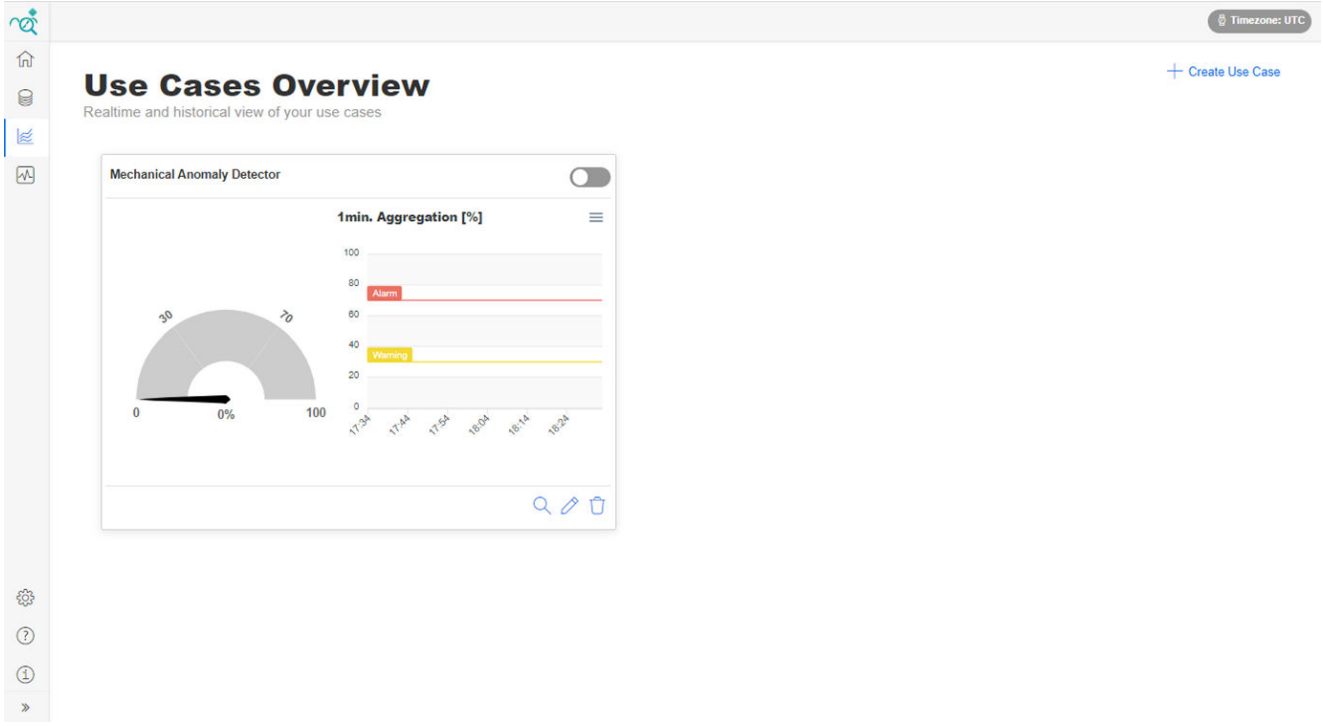


There can be two possible causes for incompatibility:

- The selected type "Data Source" is not compatible with the "Model". Select a "Low-Speed-Adapter" type "Data Source" for the Models "Energy Efficiency" and "Basic Anomaly Detection" and select a "High-Speed-Adapter" type "Data Source" for the Model "Mechanical Anomaly Detector".
- The selected "Data Source" type is correct, but either a parameter is missing or the sampling time is incorrect in the configuration of the "Data Source". In this case, you must check the configuration in "DSF". More information is provided in the sections "Configuring a Drive System Framework Low-Speed-Adapter (Page 23)" or "Configuring a Drive System Framework High-Speed-Adapter (Page 25)".

Result

If the selected "Model" and the "Data Source" are compatible, the "Use Case" is added to the "Use Cases Overview" page with inactive status.



9.2.2 Activating a "Use Case"

Proceed as described to activate an inactive "Use Case".

Note

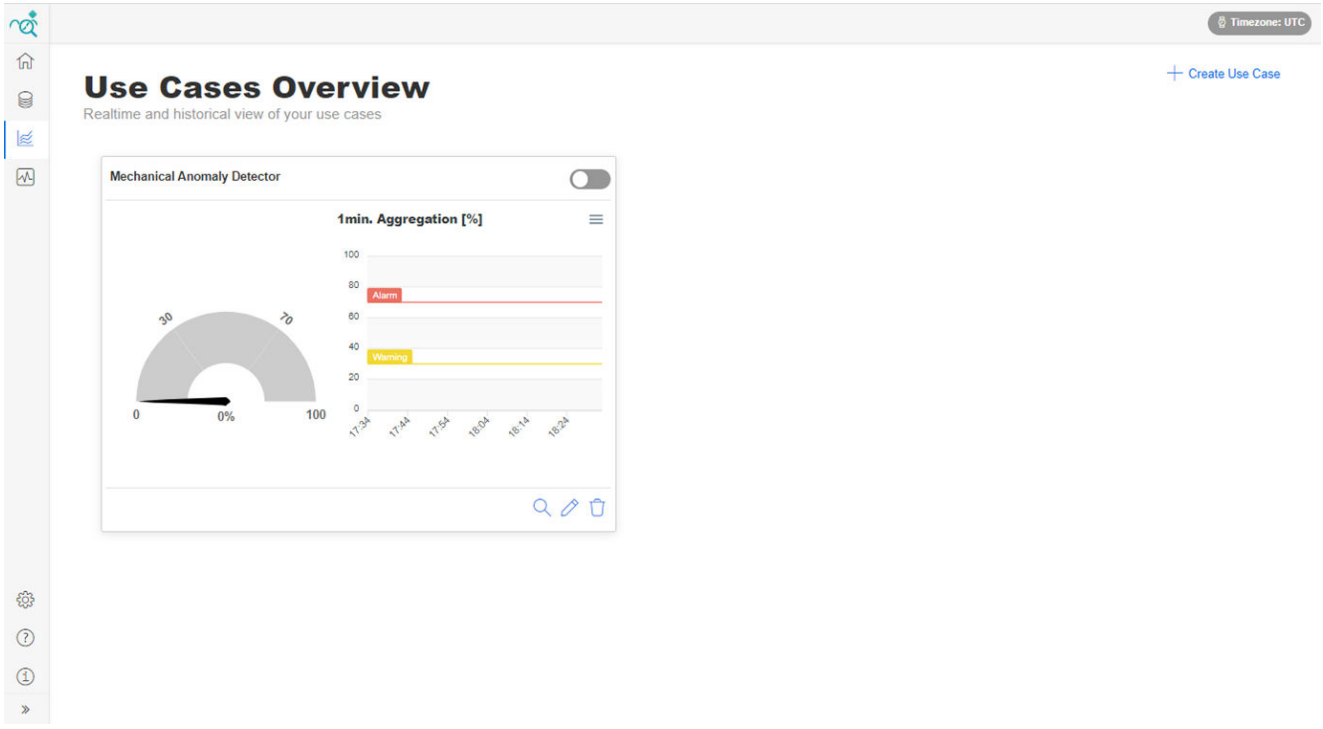
- A maximum of 8 "Use Cases" which use "Data Source" of the type "Low-Speed-Adapter" can be simultaneously active.
- A maximum of 4 "Use Cases" which use "Data Source" of the type "High-Speed-Adapter" can be simultaneously active.
- The total number of all active "Use Cases" must not exceed 8.

Requirement

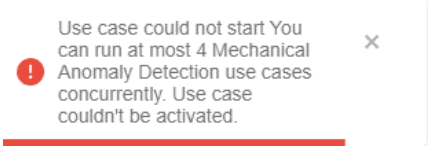
At least one "Use Case" is created in the application and the status is inactive.

Procedure

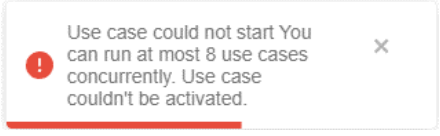
1. In the navigation bar, click on "Use Cases". The "Use Cases Overview" page is displayed.



2. Click on the slider of the "Overview Widget" of the "Use Case" that you want to activate.
3. If 4 "Use Cases" that use a "High-Speed-Adapter" are already active, then the following error message is displayed when trying to activate a 5th "Use Case" of this type and activation is canceled.

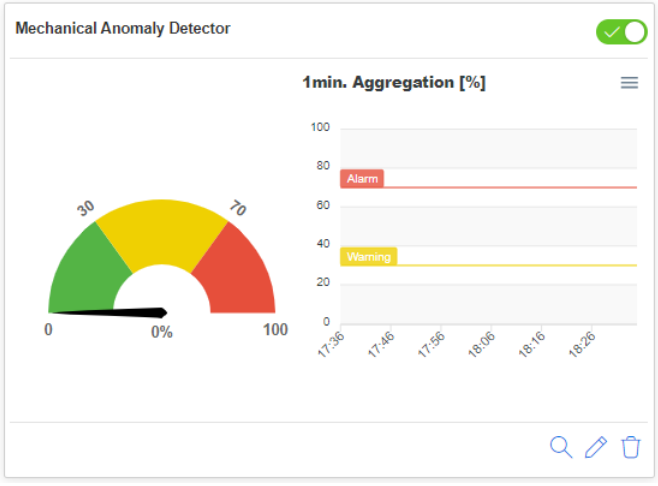


If 8 "Use Cases" that use a "Low-Speed-Adapter" are already active, then the following error message is displayed when trying to activate a 9th "Use Case" of this type and activation is canceled.



Result

The "Use Case" is activated. The associated "Model" starts generating KPIs depending on the data provided by the associated "Data Source".



9.2.3 Deactivating a "Use Case"

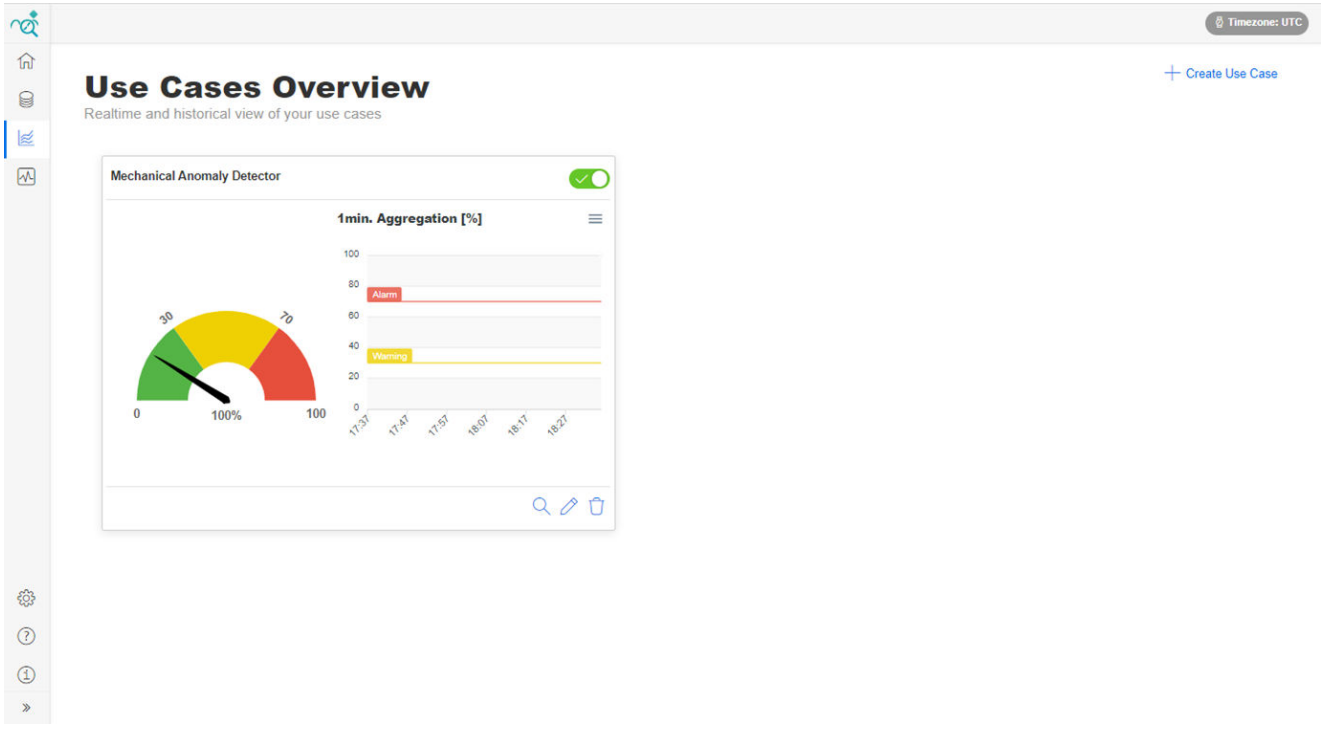
Proceed as described to deactivate an active "Use Case".

Requirement

At least one "Use Case" is created in the application and the status is active.

Procedure

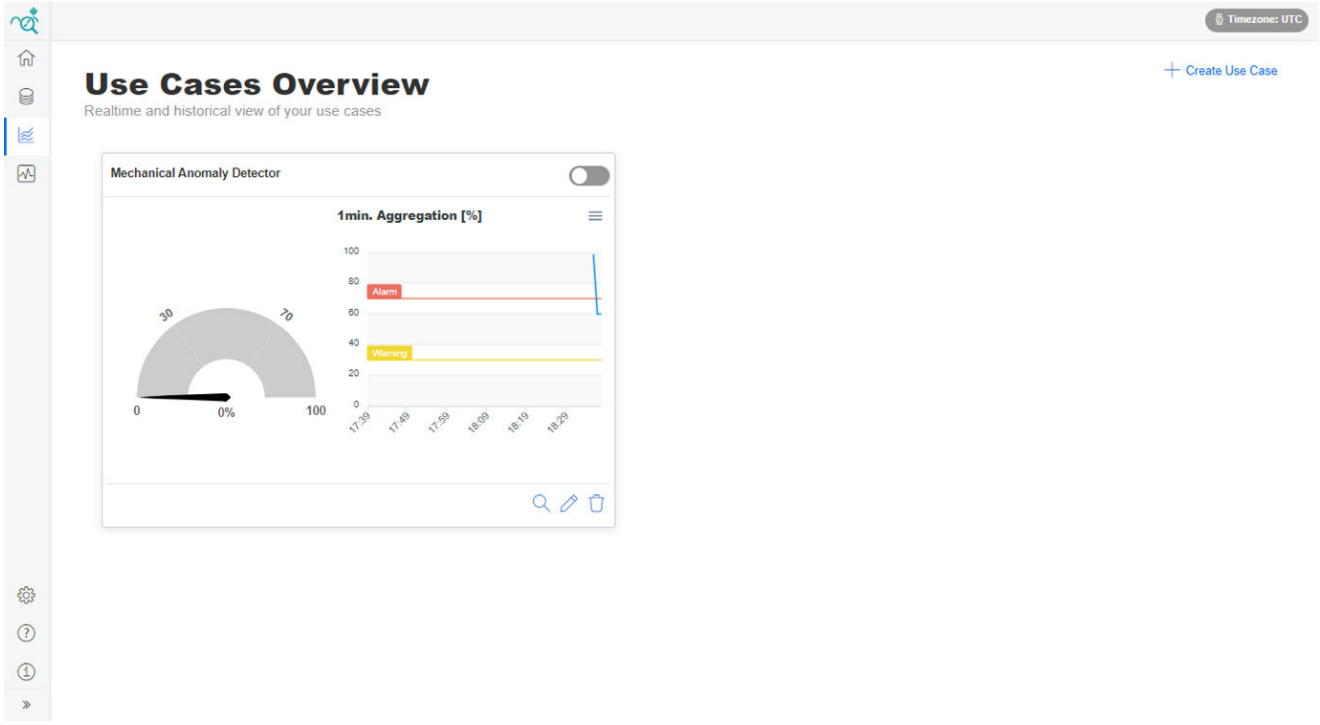
1. In the navigation bar, click on "Use Cases". The "Use Cases Overview" page is displayed.



2. Click on the slider of the "Overview Widget" of the "Use Case" you want to deactivate.

Result

The "Use Case" is deactivated and no longer generates any KPIs. You can access the historical data of this "Use Case" via the Details Page.



9.2.4 Editing a "Use Case"

Proceed as described below to edit a "Use Case".

Note

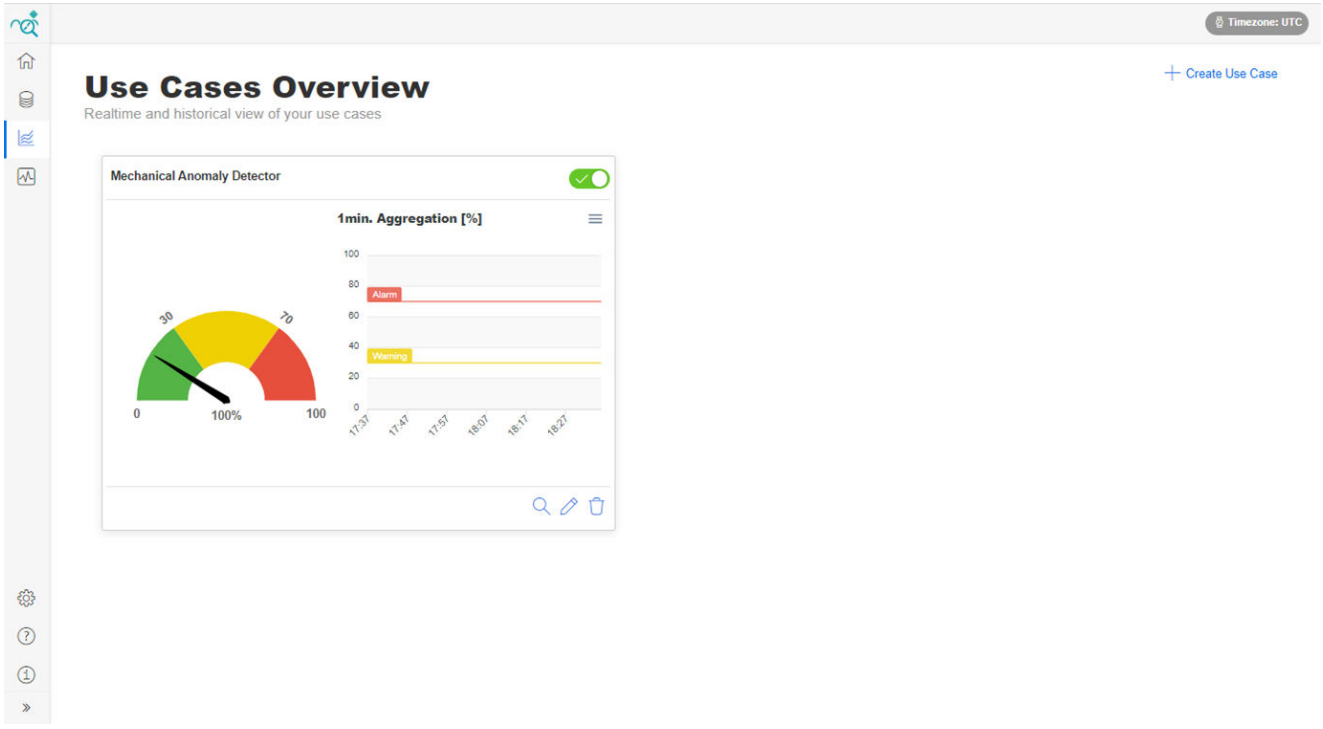
You can only change the name and description of the "Use Case". The "Model" and "Data Source" of a created "Use Case" can no longer be changed.

Requirement

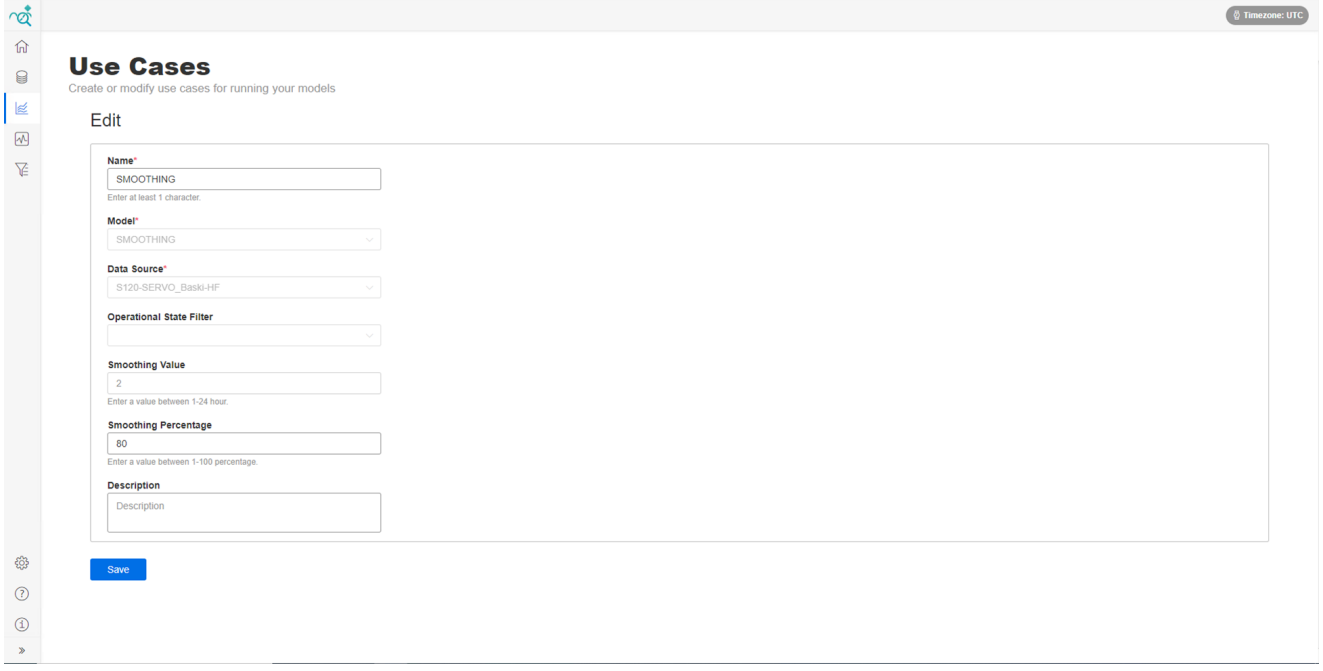
At least one "Use Case" is created in the application.

Procedure

1. In the navigation bar, click on "Use Cases". The "Use Cases Overview" page is displayed.



2. Click on the button of the "Use Case" you want to edit. The following dialog is displayed.



3. You can change the name of the "Use Case", as well as "Smoothing Percentage" and "Description".
4. Click the "Save" button to accept the changes.

Result

The changes have been saved.

9.2.5 Deleting a "Use Case"

Proceed as described below to delete a "Use Case".

Note

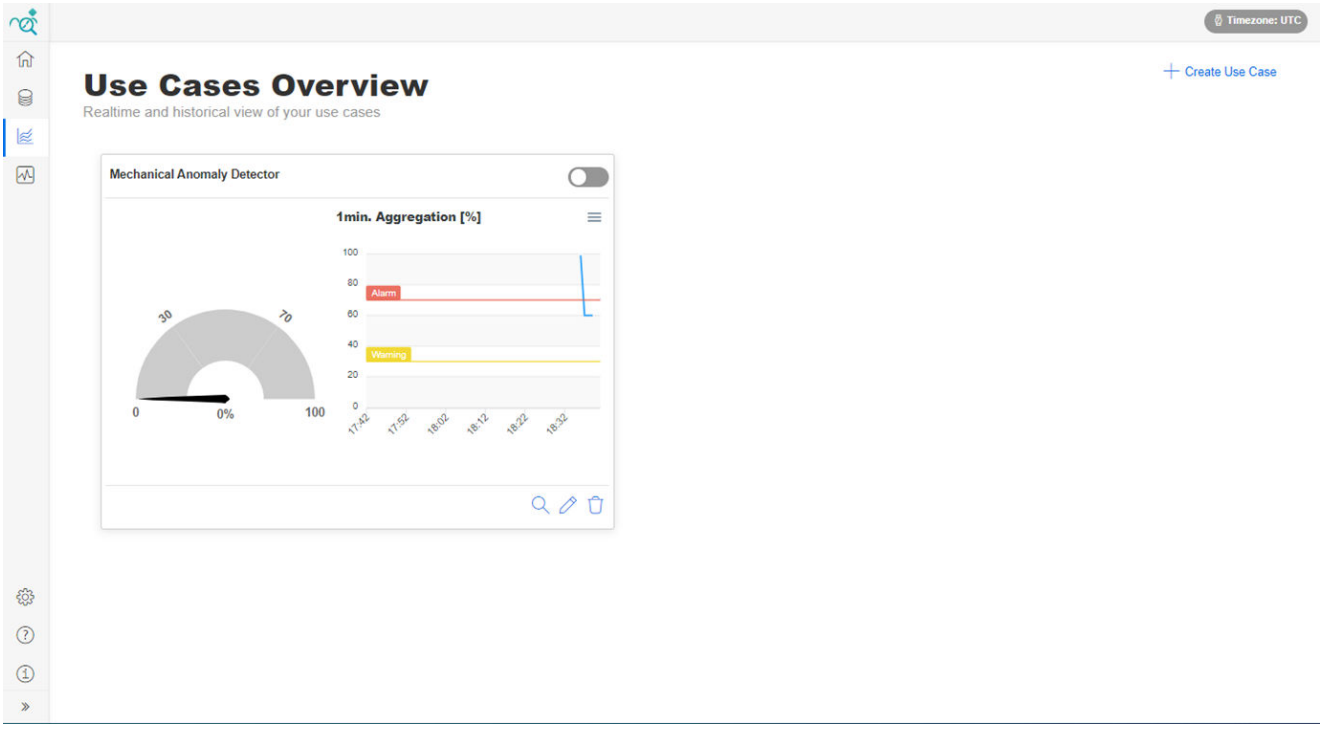
An activated "Use Case" must first be deactivated before you can delete it. See section "Deactivating a Use Case (Page 86)".

Requirement

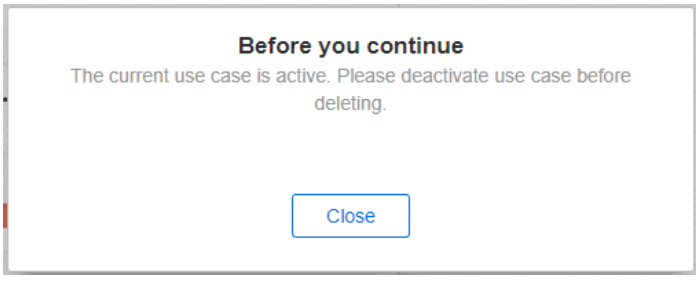
At least one "Use Case" is created in the application.

Procedure

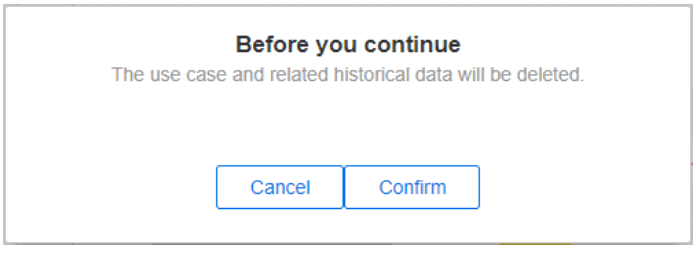
1. In the navigation bar, click on "Use Cases". The "Use Cases Overview" page is displayed.



2. Click on the button in the "Overview Widget" of the "Use Case" you want to delete. The following message is displayed if the selected "Use Case" is active. Deactivate the "Use Case" and continue with the deletion process.



3. Before the "Use Case" including all historical data is deleted, the following message is displayed.



Note

The deletion of the "Use Case" becomes effective immediately. If you delete a "Use Case", all recorded historical data will also be permanently deleted.

If you want to proceed with the deletion process, click the "Confirm" button. If you want to cancel the deletion process, click "Cancel".

Result

The "Use Case", including all recorded historical data, is permanently deleted.

9.2.6 Opening the "Use Case" details page

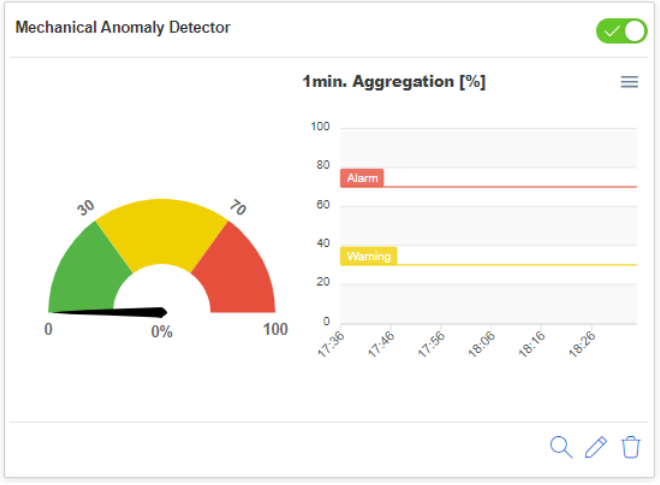
Proceed as described below to open the details page of a "Use Case".

Requirement

At least one "Use Case" is created in the application.

Procedure

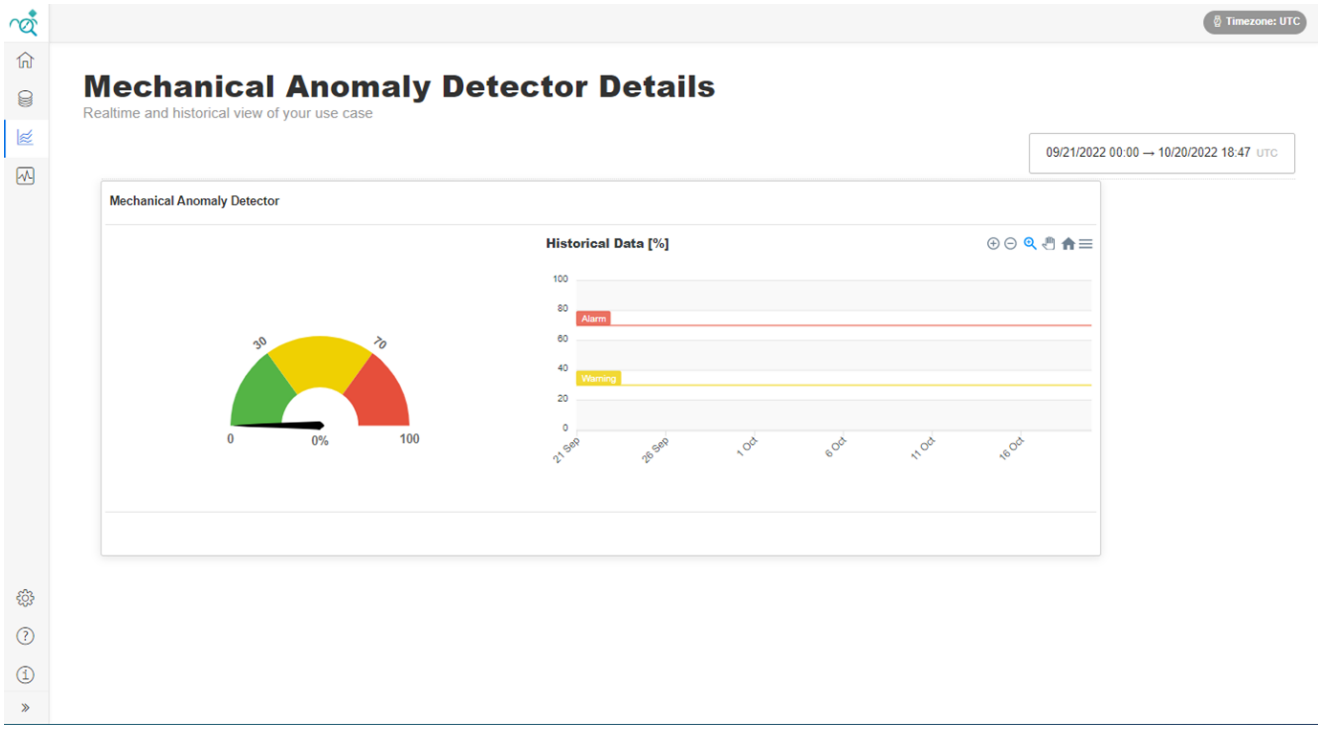
1. In the navigation bar, click on "Use Cases". The "Use Cases Overview" page is displayed.



2. Click on the button of the "Use Case" for which you want to open the details page. The details page is displayed.

Result

The details page is displayed. The represented graphics show the data for the last 30 days with daily aggregations.



9.2.7 Adapting the time interval on the details page

Proceed as described below to change the time interval on the "details page".

Note

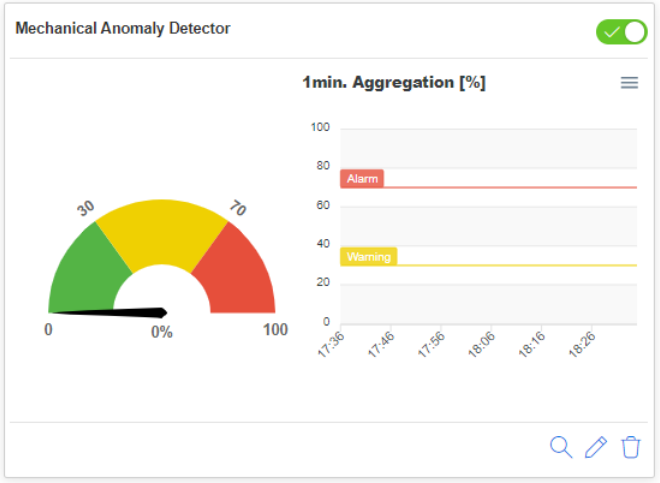
- All diagrams displayed on the details page have the same interval. It is not possible to change the time interval of a single diagram.
- Diagrams are based on the aggregated data of the respective selected time interval. For a more detailed view, you can select a smaller time interval.

Requirement

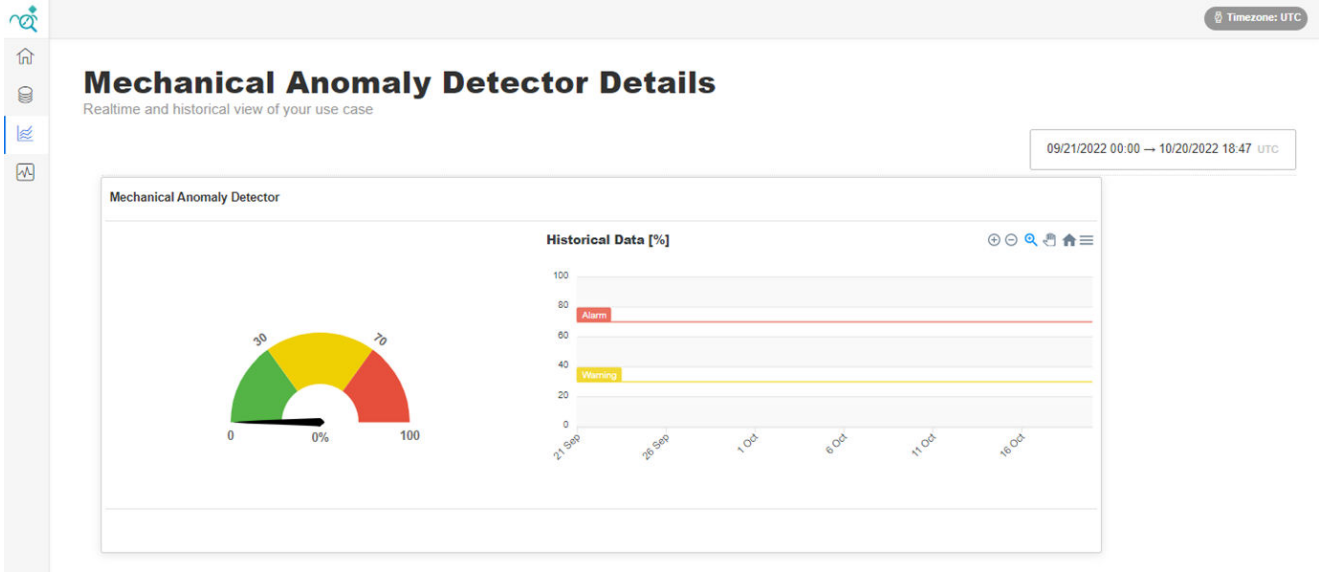
At least one "Use Case" is created in the application.

Procedure

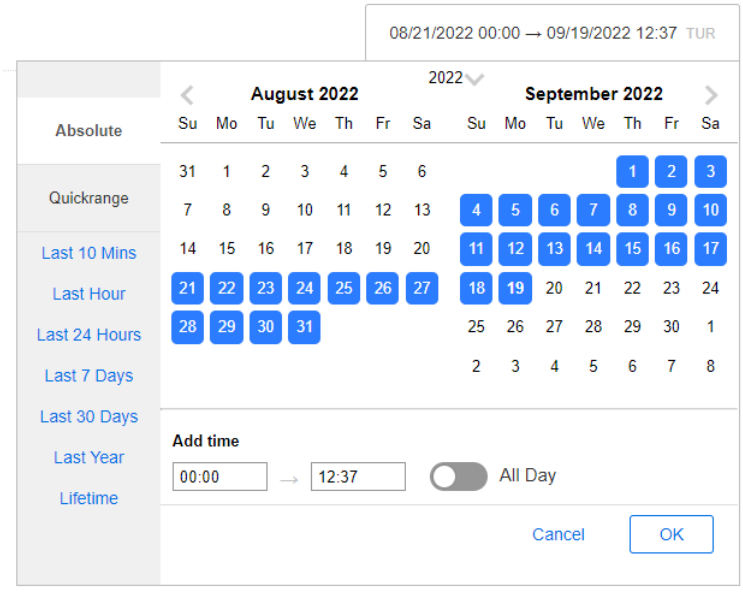
1. In the navigation bar, click on "Use Cases". The "Use Cases Overview" page is displayed.
2. Click on the button of the "Use Case" for which you want to open the details page.



3. The details page of the "Use Case" is displayed. At the top right of this page is the time interval selection for detailed diagrams. The default interval for all "Use Cases" is the last 30 days, displayed as daily aggregations.



4. Click on the time interval selection (top right). The following dialog is displayed.

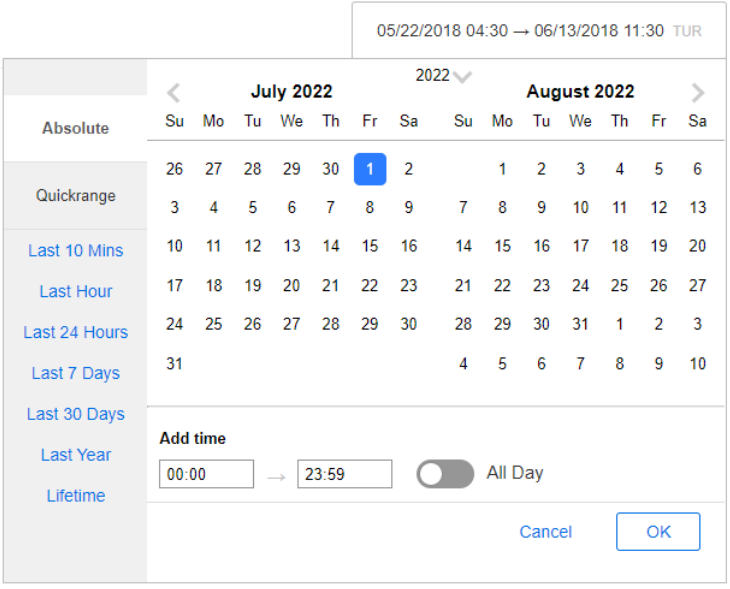


You can select the time interval by specifying a start and end date as absolute values or by using preset quick ranges.

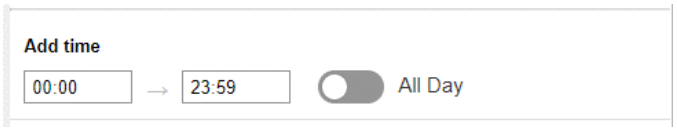
Note

- All diagrams displayed on the details page have the same interval. It is not possible to change the time interval of a single diagram.
- Diagrams are based on the aggregated data of the respective selected time interval. For a more detailed view, you can select a smaller time interval.

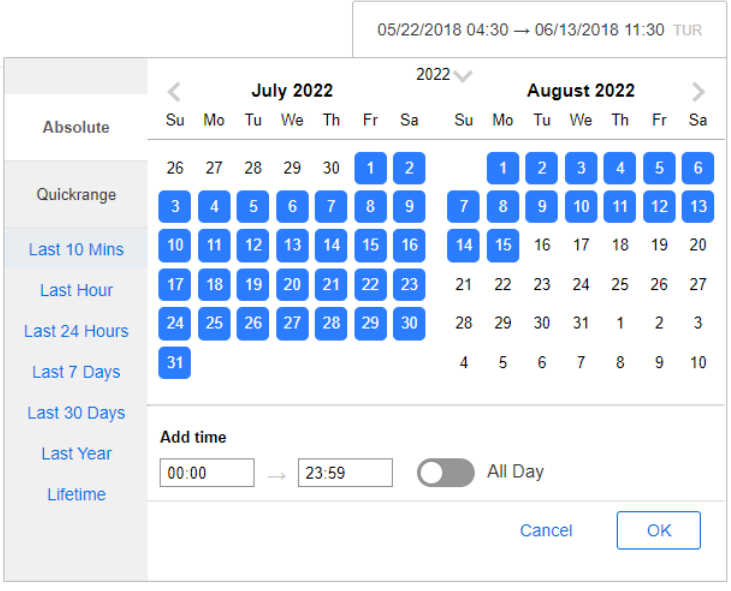
5. This step describes the selection of the date interval by absolute values:
   – Select the year, month and day of the start date of the time interval.



   – If you want your start date to begin at a time other than the beginning of the day (00:00), enter the desired time in the left field in the "Add time" area.



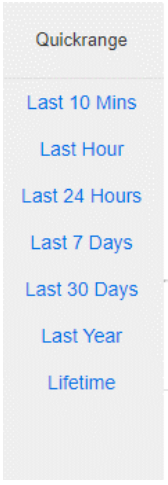
   – Select the year, month and day of the end date of the time interval.



– If you want your end date to end at a time other than the end of the day (23:59), enter the desired time in the right field in the "Add time" area.

– The time selection under "Add time" can be discarded by activating the option "All Day". If this option is enabled, it is not necessary to enter the time for the start date and end date. The fields are deactivated.

6. This step describes the selection of the time interval via preset quick ranges:
As an alternative to entering the time interval in absolute values, you can select preset time intervals.



According to your selection, the data collected from the beginning of the range to the current time will be displayed on the "details page".

7. Click the "OK" button after you have selected the desired time interval.

Result

The time interval selection shows the selected interval and the diagrams are updated accordingly.

9.3 CMS Dashboard

This section describes the steps required to create, configure and delete "CMS Dashboard" elements.

9.3.1 Inserting a "CMS Dashboard" element

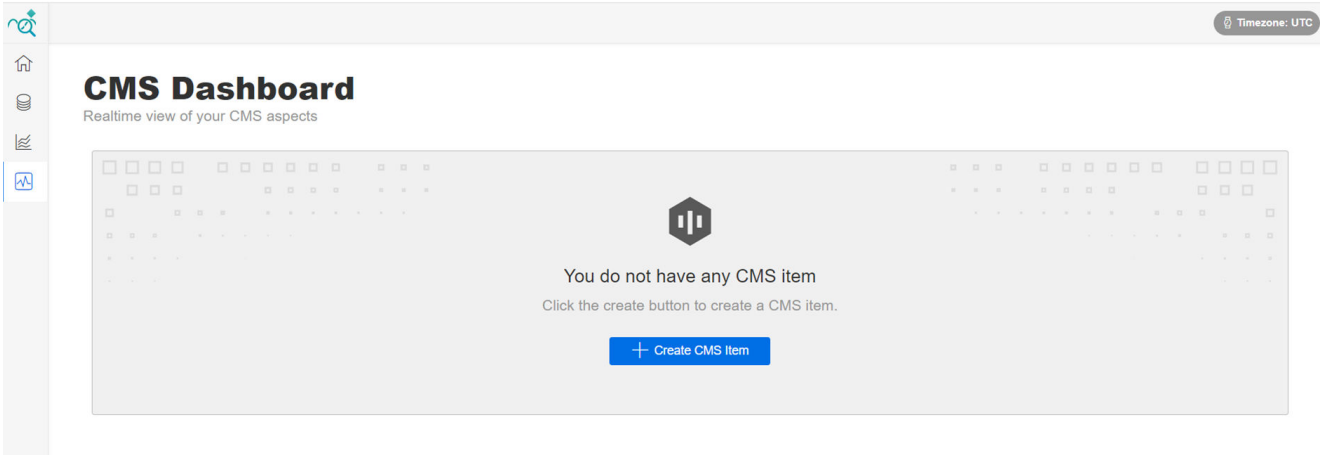
Proceed as described below to add a new "CMS Dashboard" element.

Requirement

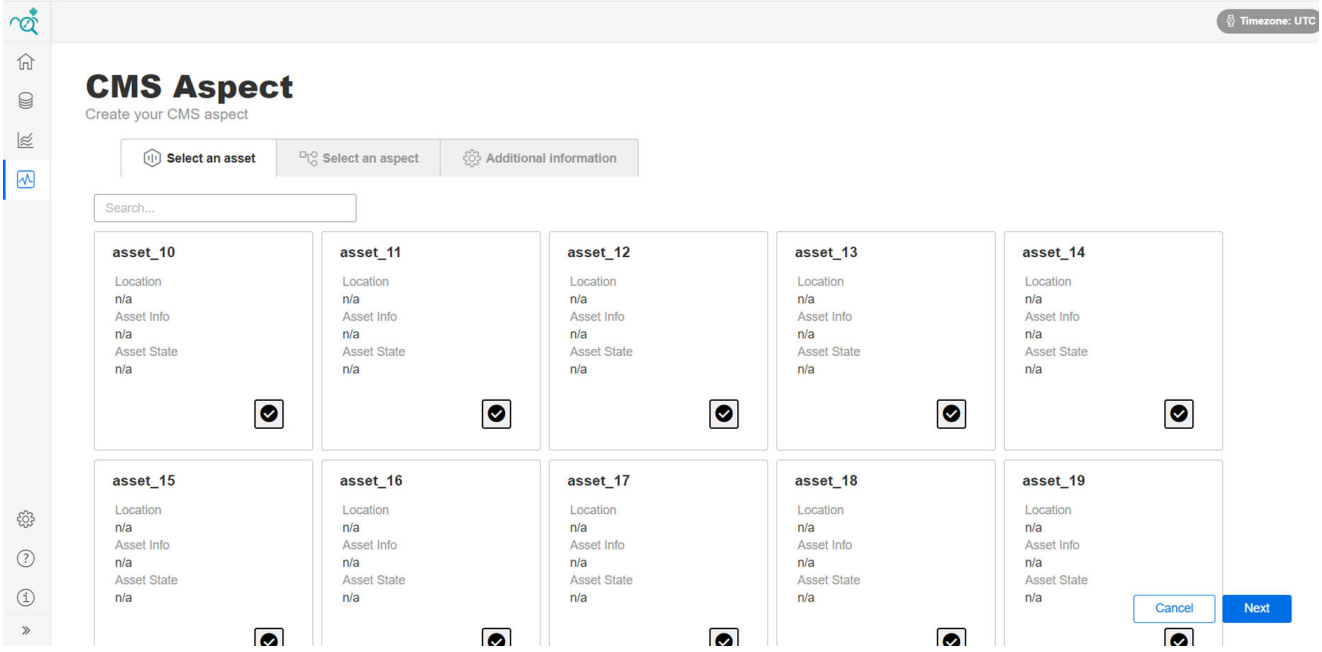
The Asset, Aspect and variable data for the CMS module have been configured in the Industrial Edge Data Service application.

Procedure

1. In the navigation bar, click on "CMS Dashboard". The "CMS Dashboard" page is displayed.

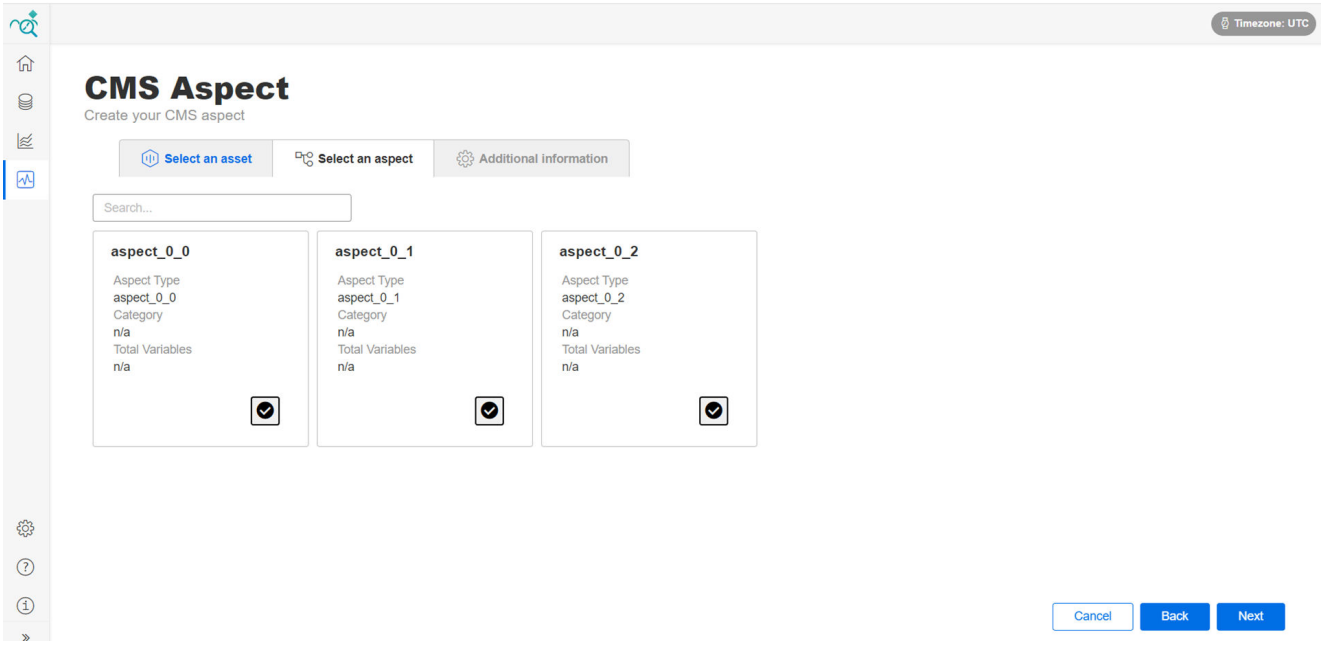


2. Click the "+ Create CMS Item" button. The "CMS Aspect" page with the "Select an asset" tab is displayed.



3. Select an Asset. You can filter Assets by name using the search function "Search...".

4. After you have selected an Asset, click "Next". The "Select an Aspect" tab is displayed. Click on "Cancel" if you want to cancel the addition of a new "CMS-Dashboard" element.

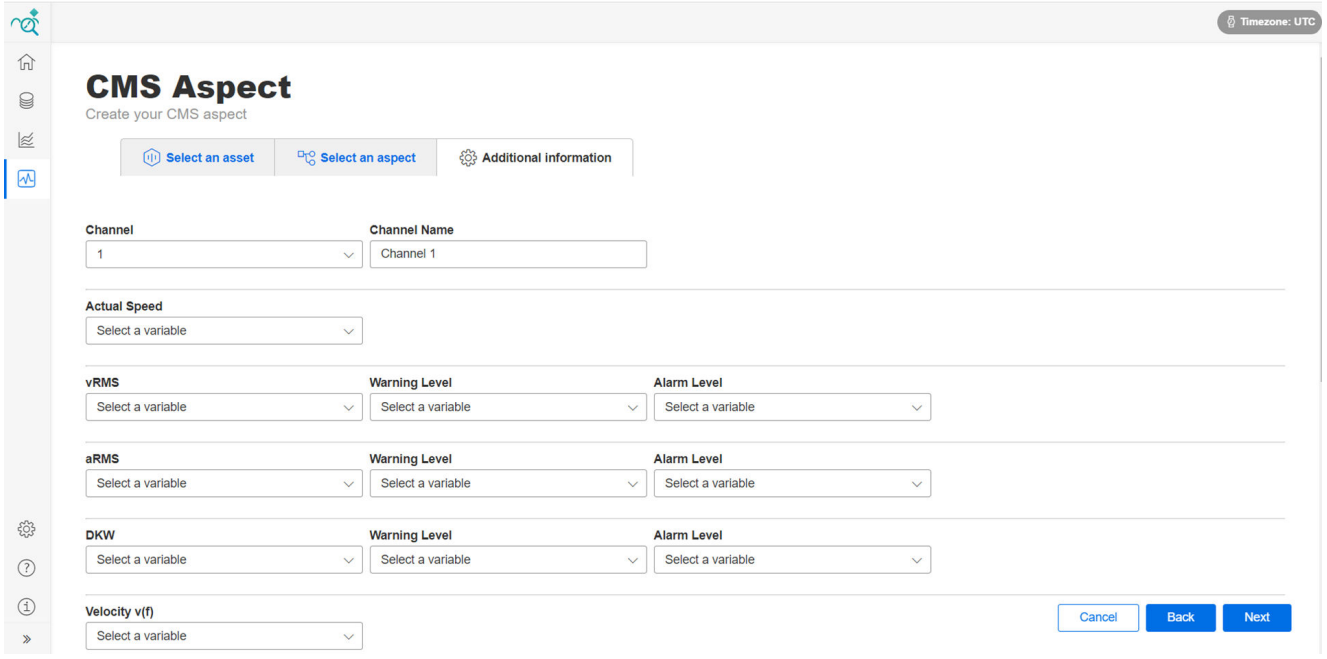


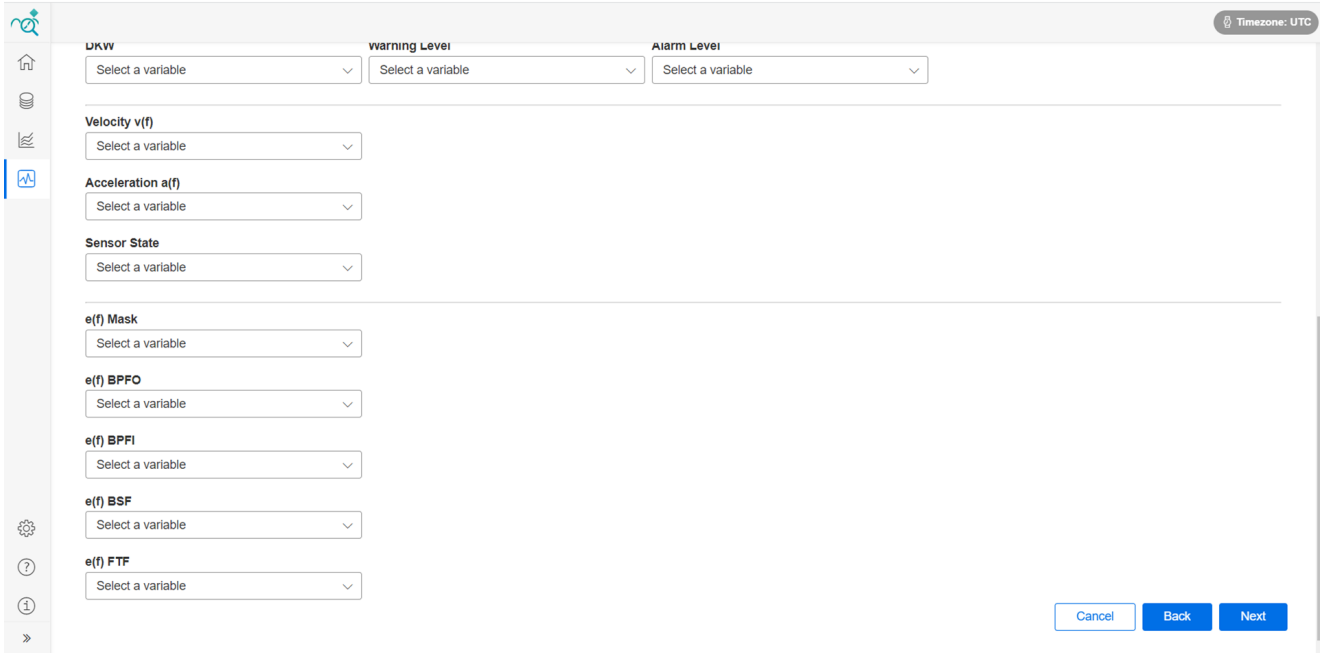
5. Select an Aspect. You can filter Aspects by name using the search function "Search...".
6. After you have selected an Aspect, click "Next". The "Additional information" tab is displayed. Click on "Back" to go to the "Select an asset" page if you want to change the selected Asset. Click on "Cancel" if you want to cancel the addition of a new "CMS-Dashboard" element.

7. Select a channel between one and four. The "Channel Name" field displays a default name, which you change as required.
   After selecting the channel, link the variables defined in the Industrial Edge Data Service application with the "CMS-Dashboard" element by selecting the relevant variables from the drop-down fields.

Note

- If a channel is not used, you do not need to configure it.
  Example: Only channels 1 and 2 can be configured, without configuring channels 3 and 4.
- You do not need to configure channels sequentially.
  Example: Only channel 3 can be configured, without configuring channels 1, 2 and 4.

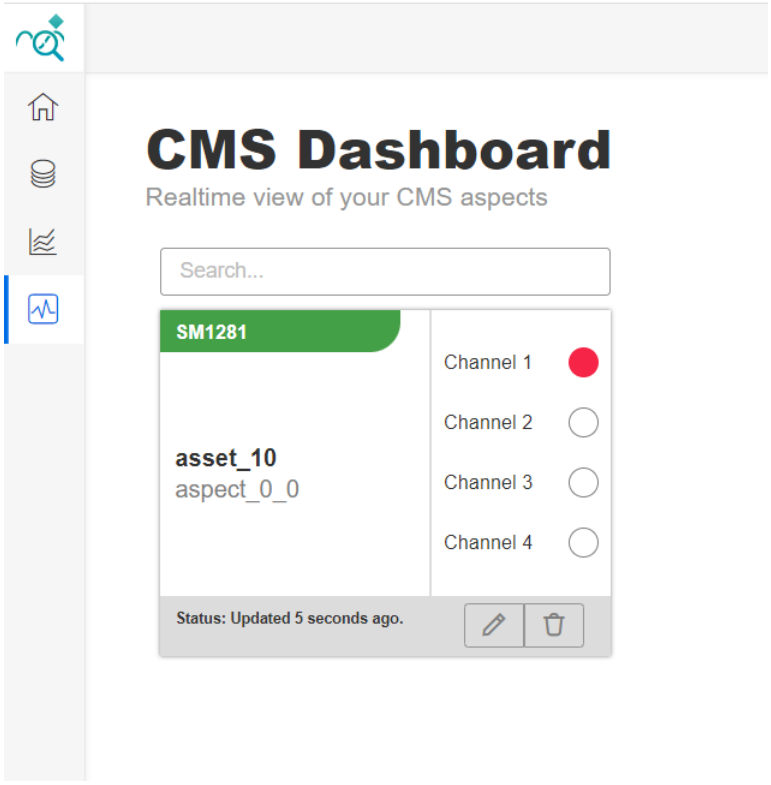




8. Click on "Next" to accept the configuration. The new element is added to the "CMS-Dashboard" page.
   Click on "Back" to go to the "Select an aspect" page if you want to change the selected Aspect.
   Click on "Cancel" if you want to cancel the addition of a new "CMS-Dashboard" element.

Result

The new "CMS-Dashboard" element is added to the "CMS-Dashboard" page.



9.3.2 Editing a "CMS Dashboard" element

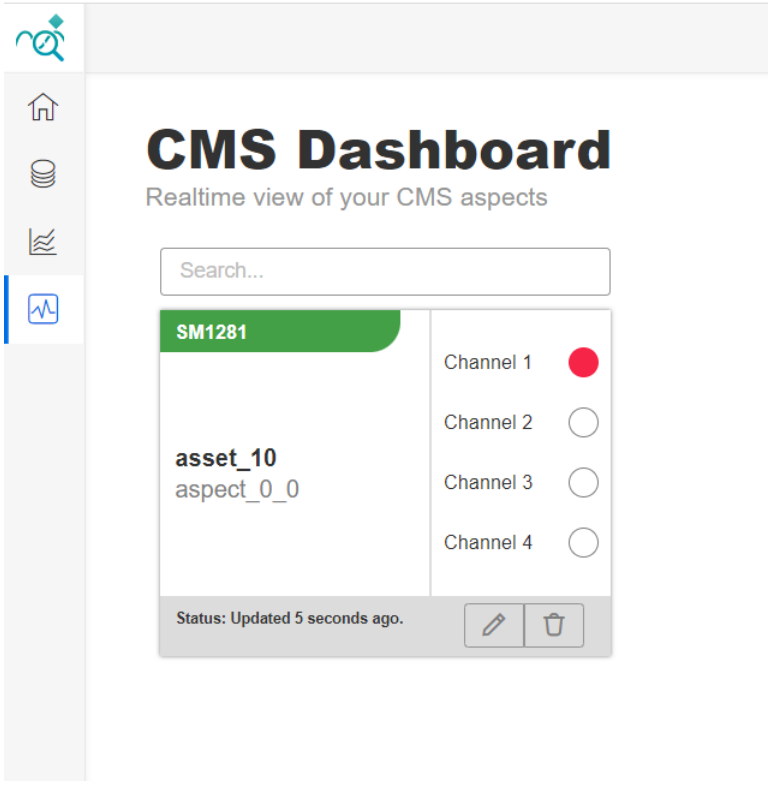
Proceed as described below to edit a "CMS Dashboard" element.

Requirement

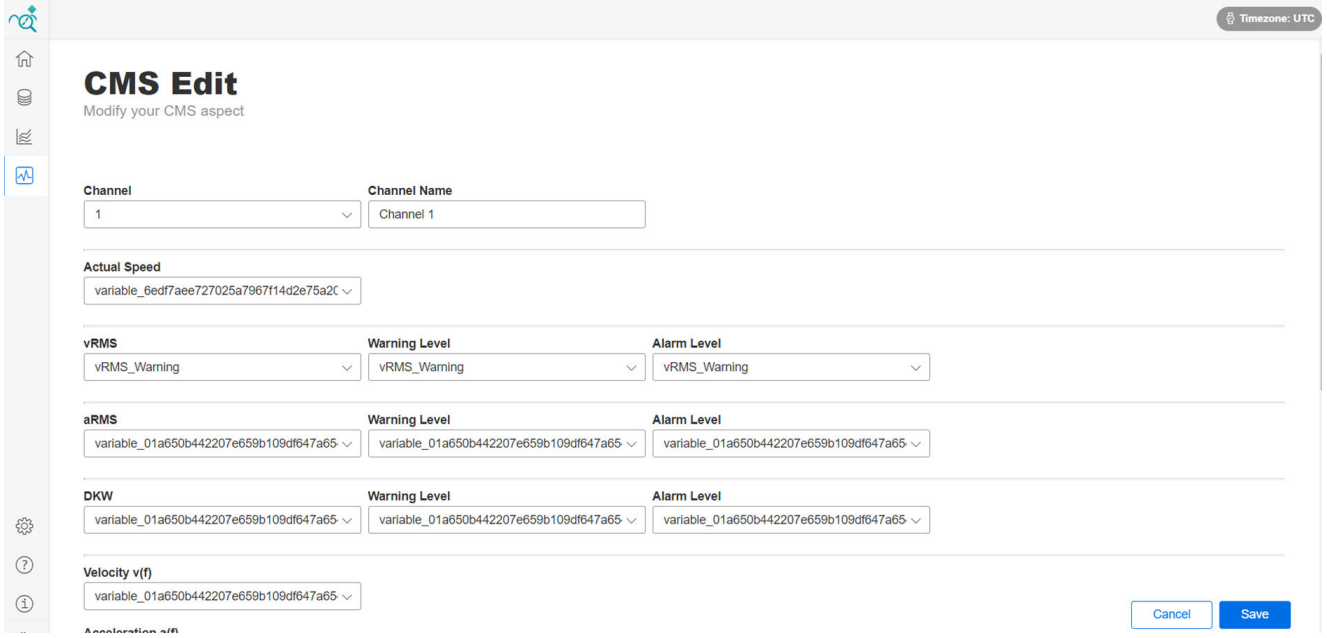
At least one "CMS Dashboard" element is created in the application.

Procedure

1. In the navigation bar, click on "CMS Dashboard". The "CMS Dashboard" page is displayed.



2. Click the button of the "CMS Dashboard" element that you want to edit. The "CMS Edit" page is displayed.



3. You can change the "Channel Name" or the selected variables.
4. After completing the changes, click "Save" to apply the changes. Click "Cancel" to discard the changes.

Result

The changes have been accepted.

9.3.3 Deleting a "CMS Dashboard" element

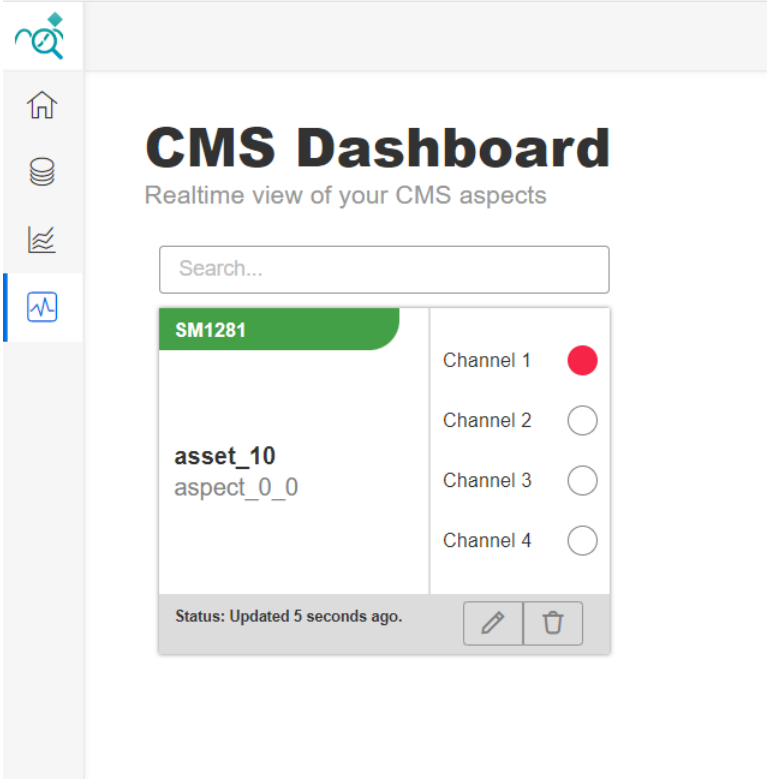
Proceed as described below to delete a "CMS Dashboard" element.

Requirement

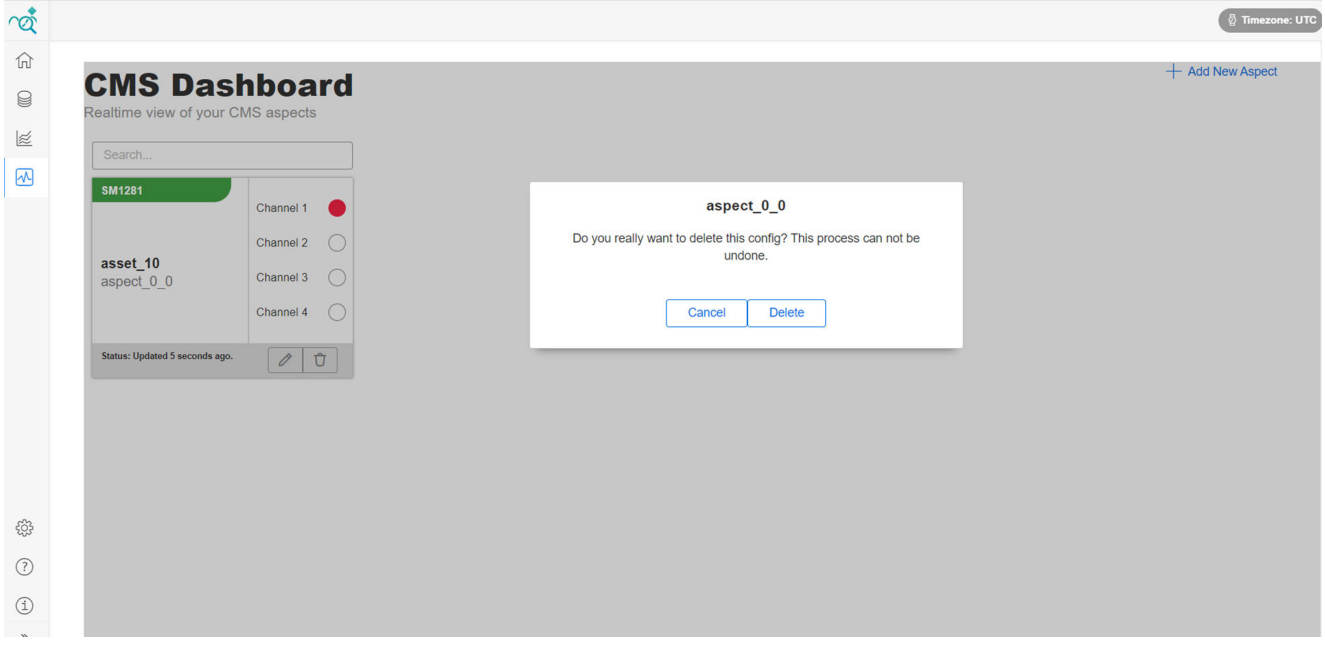
At least one "CMS Dashboard" element is created in the application.

Procedure

1. In the navigation bar, click on "CMS Dashboard". The "CMS Dashboard" page is displayed.



2. Click the button of the "CMS Dashboard" element that you want to delete.
The deletion of the "CMS Dashboard" element becomes effective immediately. If you want to proceed with the deletion process, click on "Delete" in the following message. If you want to cancel the deletion process, click "Cancel".



Result

The "CMS Dashboard" element has been deleted.

9.3.4 Opening the "CMS Dashboard" details page

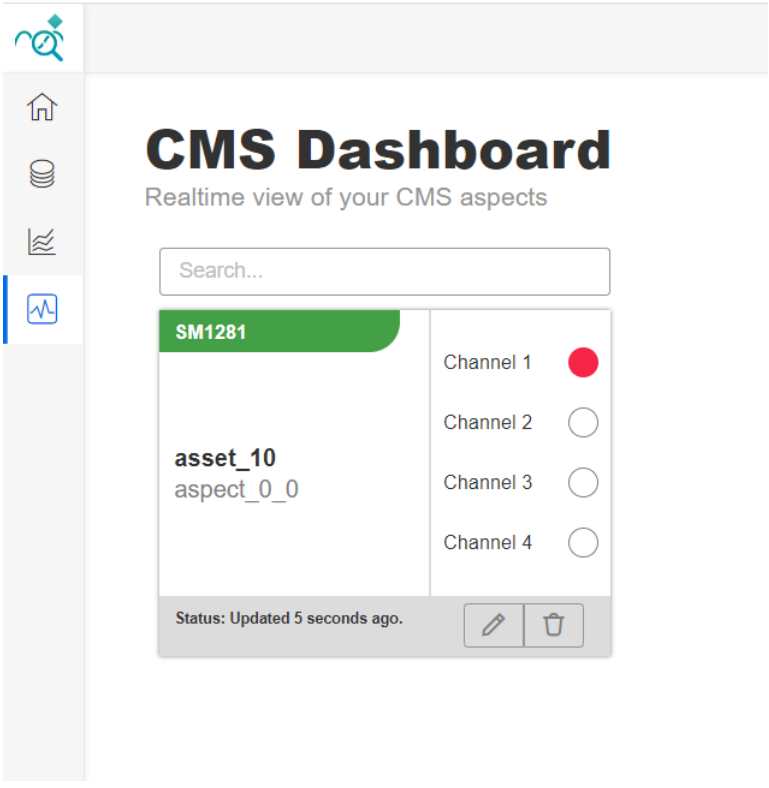
Proceed as described below to open the details page of a "CMS Dashboard" element.

Requirement

At least one "CMS Dashboard" element is created in the application.

Procedure

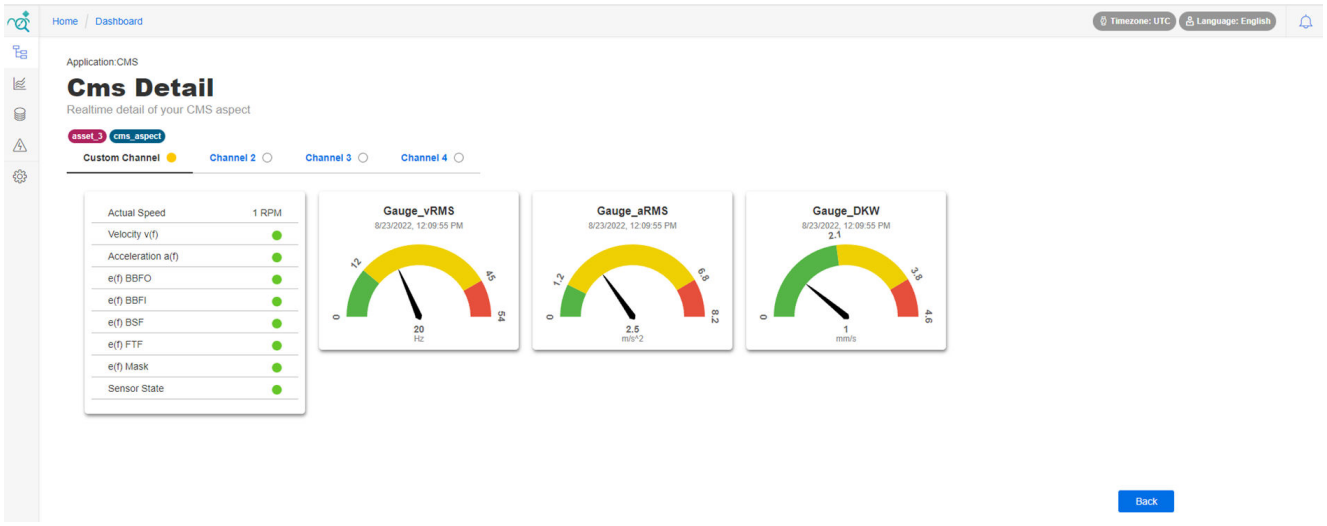
1. In the navigation bar, click on "CMS Dashboard". The "CMS Dashboard" page is displayed.



2. Click on the "CMS Dashboard" element (except for the buttons and) to open the "details page".

Result

The details page of the "CMS Dashboard" element is displayed.



9.4 Operational State Filter

The "Operational State Filter" is used to define minimum and maximum thresholds for input parameters to define the operational state of the monitored system. The "Operational State Filter" allows a Model to filter out input data when the system is not in the operational state. This enables the creation of characteristic values, provided that the system is in the operational state, and improves the quality of the KPIs created for the Model.

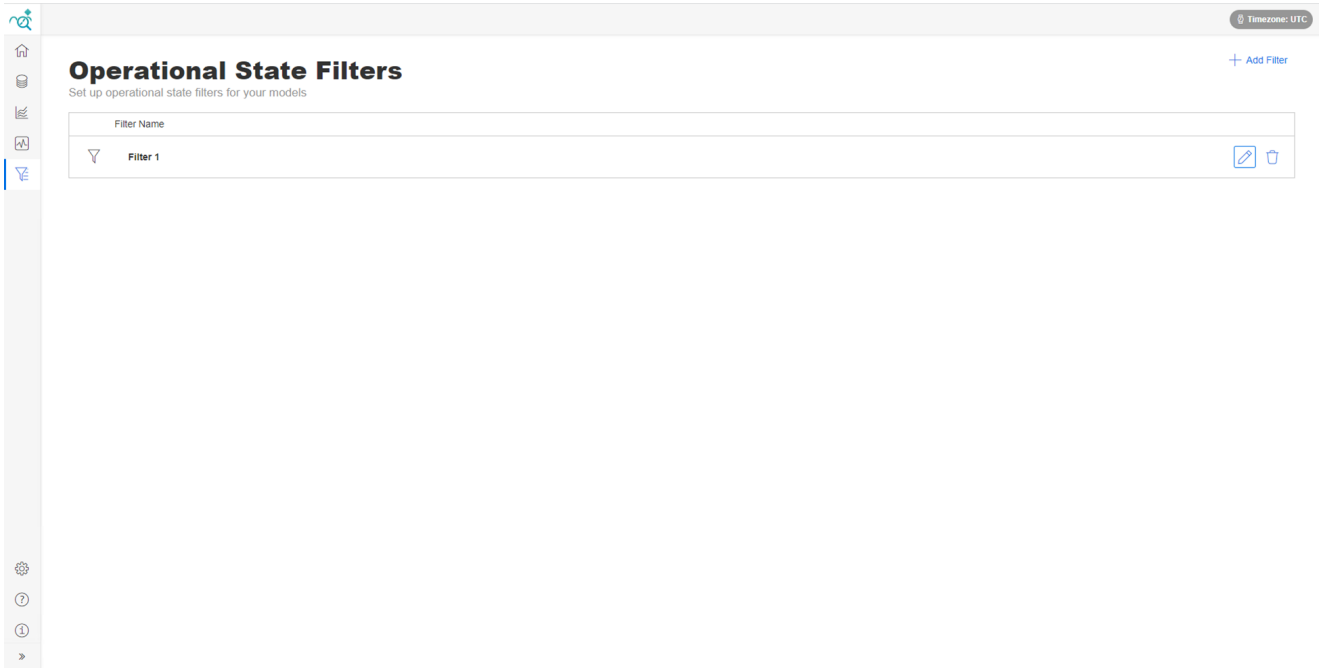
With the "Operational State Filter" you can monitor the up to two parameters from the "Data Source" for up to 48 hours and define operational state filters based on this information.

9.4.1 Inserting an "Operational State Filter"

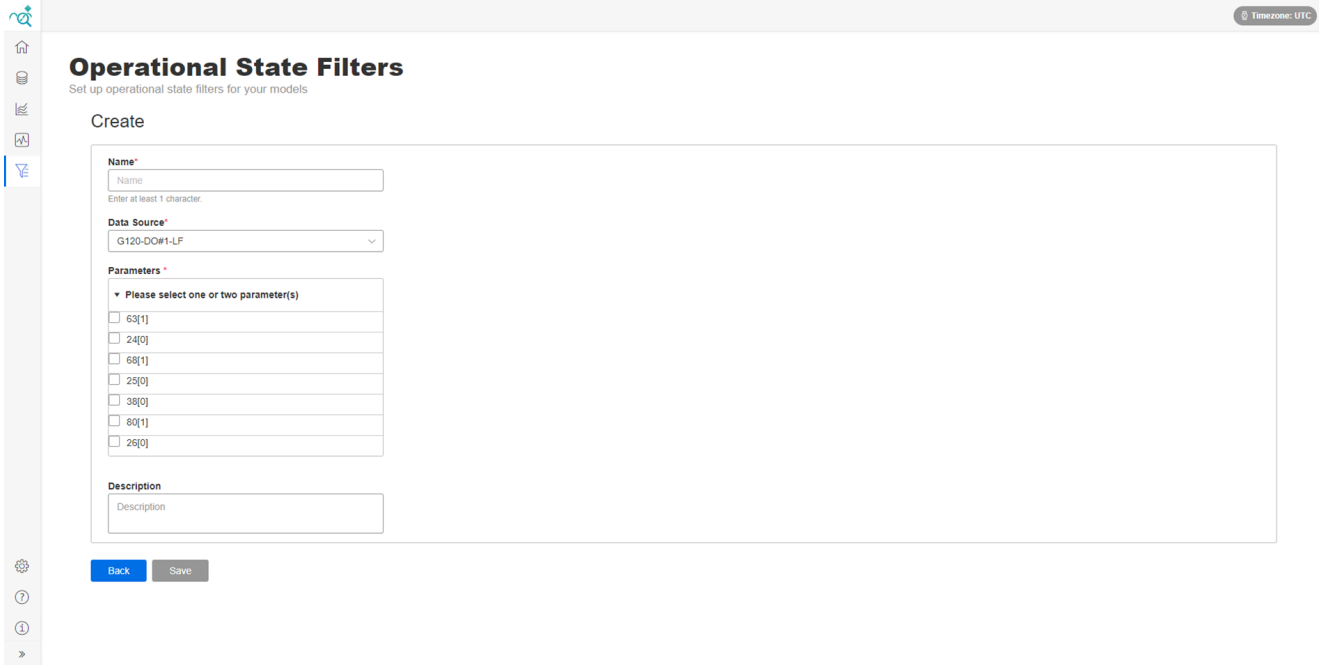
Proceed as described below to insert an "Operational State Filter".

Procedure

1. In the navigation bar, click on "Operational State Filter". The "Operational State Filter" page is displayed.



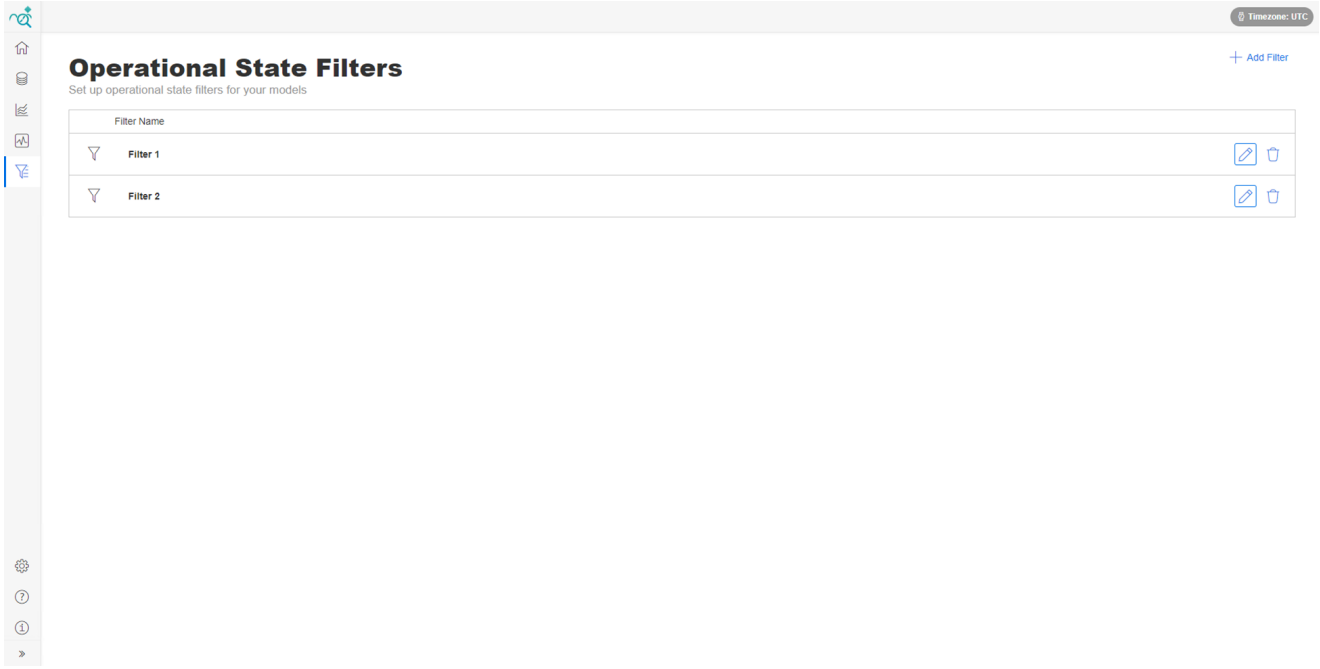
2. Click on the "+ Add Filter" button (at the top right or in the middle of the screen). The following dialog is displayed.



3. Enter a name for your filter.
   Observe the following rules when entering data:
   - The name of the filter must be entered (mandatory field).
   - The same name cannot be assigned more than once.
   - Only ASCII characters can be used.
   - The name must not be longer than 25 characters.
   - The characters #, +, / and spaces must not be used.
4. In the "Data Source" field, select a "Data Source" from the drop-down list. The drop-down list contains all "Data Sources" that are provided by the "Drive System Framework". If the required "Data Source" is missing, check the configuration in "Drive System Framework".
5. Select up to two parameters.
6. If required, enter a description of the filter in the "Description" field.
7. Click the "Save" button to save the entered values.

Result

The newly created filter is displayed in the list of "Oprational State Filters".



9.4.2 Starting/stopping "Raw Data Collection"

The "Operational State Filter" allows you to check the values of 2 input parameters for the last 48 hours. To start or stop data collection, proceed as described below.

Note

Defined "Operational State Filter" are applied to the associated "Use Cases" even when raw data collection has been stopped.

Requirement

- An "Operational State Filter" has been defined.
- The "Data Source" has been correctly configured in "Drive System Framework".

Procedure

1. In the navigation bar, click on "Operational State Filter". The "Operational State Filters" page is displayed.
2. Click on the button of the "Operational State Filter" for which you want to collect raw data. The "Operational State Filters Details Page" is displayed.
3. Click the "Play" button to start raw data collection.

4. Click the "Stop" button to stop raw data collection.

Result

Raw data collection has been started or stopped.

9.4.3 Finding suitable raw data for OSF

To define filters for Models, you need to check the raw data and find a suitable time period for setting up the filters.

The diagrams on the "Operational State Filters Details Page" can be used in 2 different modes. Besides the filter mode, you can check raw, minimum, maximum and average values of the tracked parameters in the navigation mode to find a suitable period for setting up the filters.

Proceed as described below to view the data.

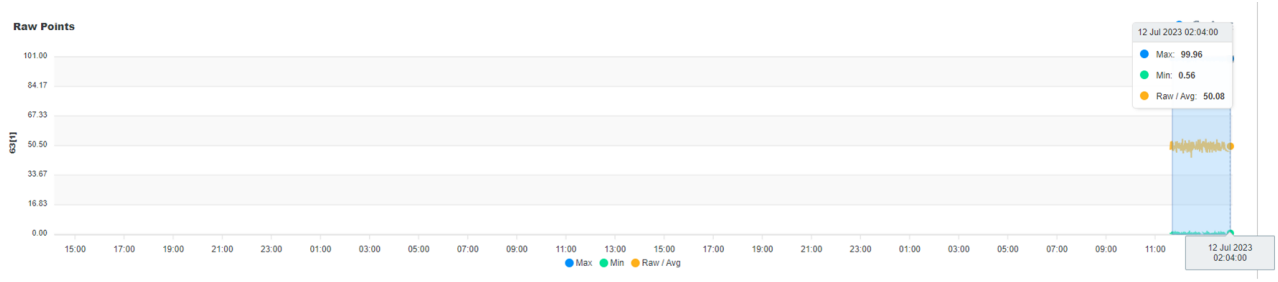
Requirement

An "Operational State Filter" has already been defined.

Procedure

1. In the navigation bar, click on "Operational State Filter". The "Operational State Filters" page is displayed.
2. Click on the button  of the "Operational State Filter" for which you want to collect raw data. The "Operational State Filters Details Page" is displayed.
3. Check whether the "Navigation mode" is activated. If this is not the case, switch to "Navigation mode" by clicking on the slider.

4. The graphs first show the time span of 48 hours. You can zoom in by selecting an area with the mouse. After selecting an area, both diagrams are refreshed according to the selection. You can zoom in multiple times.



After zooming in, you can move the diagrams to the left and right accordingly using the and buttons below the diagrams. This action does not change the zoom level.
You can reset zooming by clicking the button (top right next to the slider for navigation mode/filter mode). This resets the diagrams back to the 48-hour window.

5. Each diagram shows 3 different lines:

   – Rav / Avg: This line shows the average aggregated or non-aggregated values of the measured value. If the selected range contains too many data points to be displayed on the screen, the measured values are aggregated according to the length of the selected range.

   – Max: This line displays the maximum values of the measured value for the selected range. The Max values help the user to understand the measured values by showing the maximum limit for the selected range.

   – Min: This line displays the minimum values of the measured value for the selected range. The Min values help the user to understand the measured value by showing the minimum limit for the selected range.

Result

You have found a suitable period for the definition of filters.

9.4.4 Defining filters

Proceed as described below to define a filter. You can define multiple filters in an "Operational State Filter".

Requirements

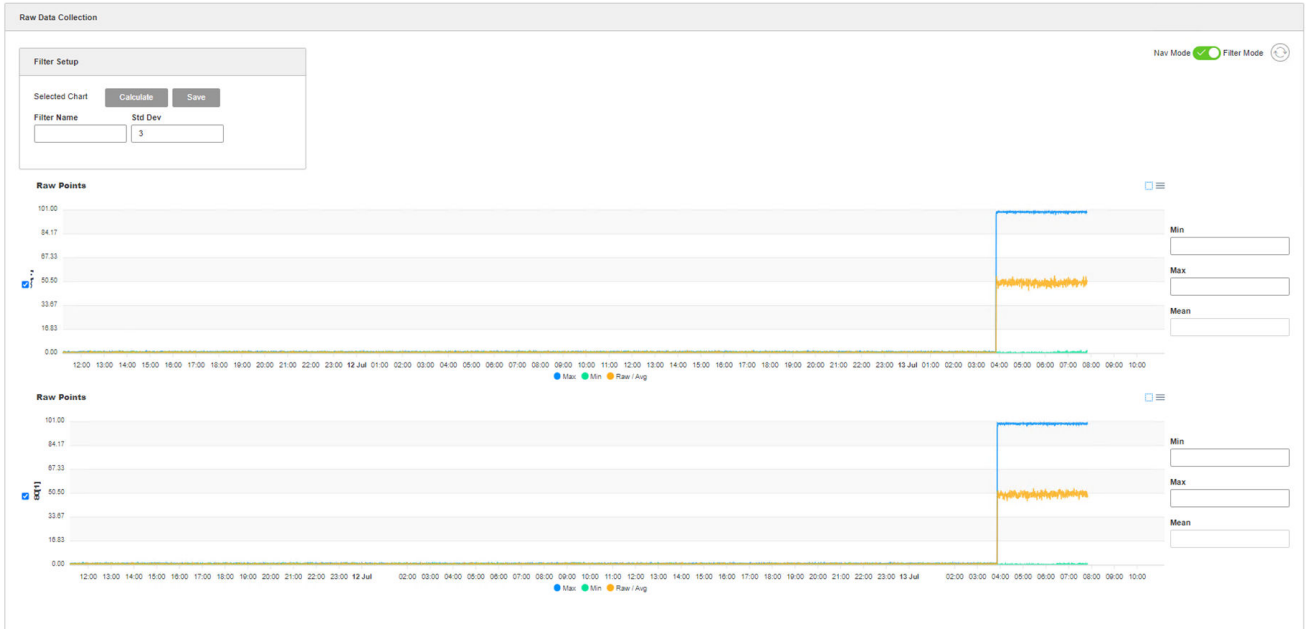
An "Operational State Filter" is defined and a suitable period is selected.

Procedure

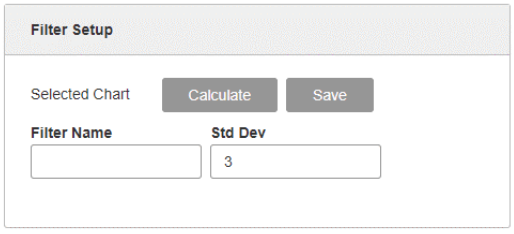
1. In the navigation bar, click on "Operational State Filter". The "Operational State Filters" page is displayed.
2. Click on the button of the "Operational State Filters" you want to examine. The "Operational State Filters Details Page" is displayed.

3. Switch to "Filter mode" by clicking on the slider.

The filter control elements are displayed.

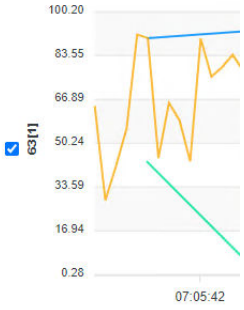


– The "Filter Setup" control element features the following elements:

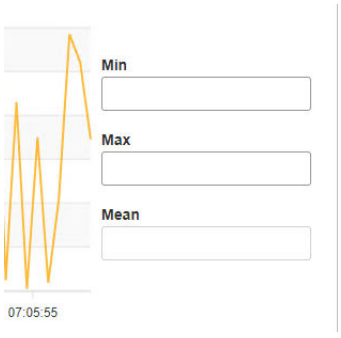


Calculate: The "Calculate" button is used to set the maximum and minimum limits based on the average value of the selected region. The values are calculated by adding/deriving a multiple of the standard deviation from the average value. You can adjust the values after the calculation.
Save: You save the filter by pressing the "Save" button.
Filter Name: Name of the filter
Std Dev: In this field you can change the standard deviation used by the "Calculate" function.

– The control elements for parameter selection indicate which parameter will be included in the defined filter. You can select both parameters or only a single parameter. If you select both parameters, the "Basic Anomaly Detection" Model will only generate characteristic values if both parameters are within the specified limits.
Click the check box to the left of the diagrams to include or exclude parameters.

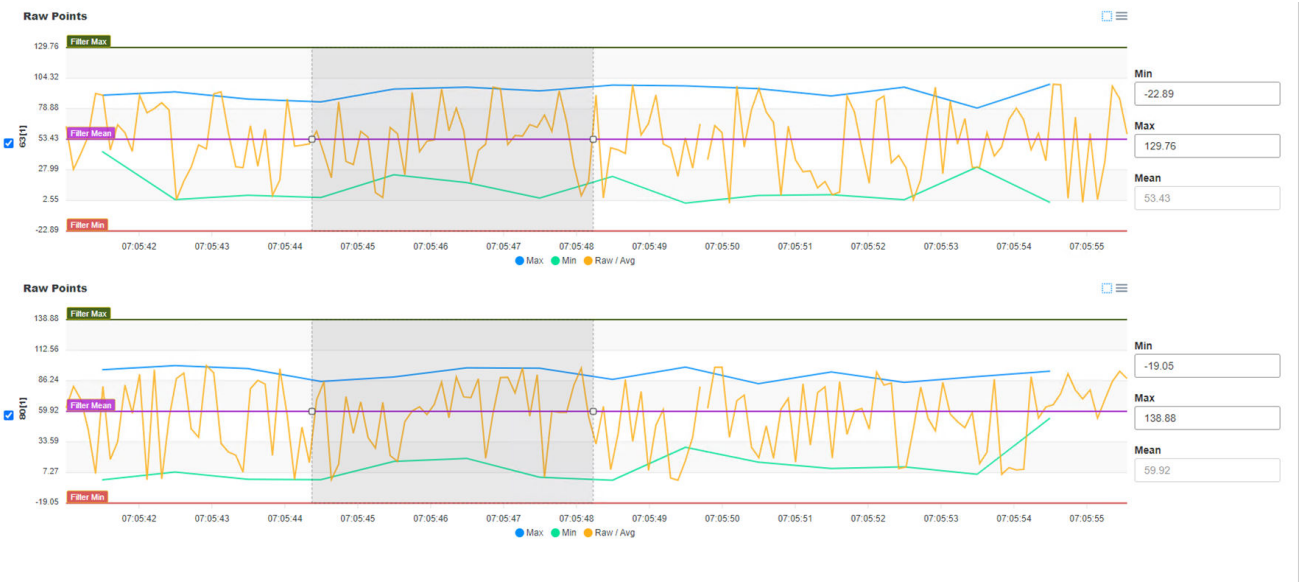


– The control elements for limit value adjustment allow you to adjust the automatically calculated minimum and maximum values.

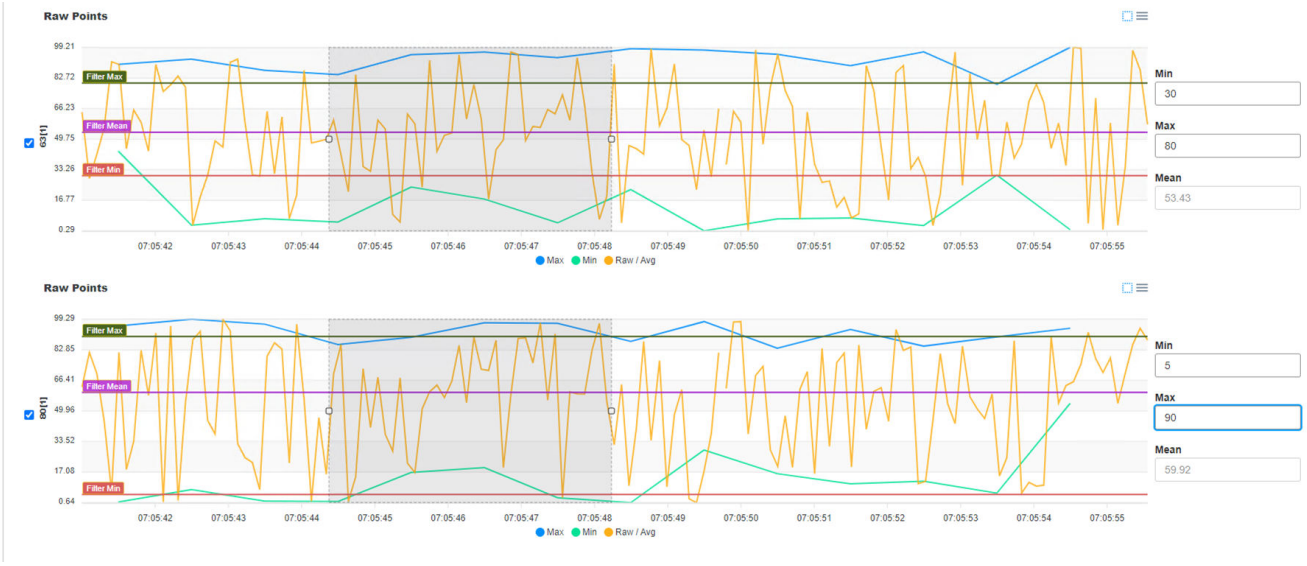


4. Enter the filter name.
5. Adjust the standard deviation value.
6. Select the parameters you want to include in the filter.

7. Select a suitable area in the diagram and click the "Calculate" button. Proposed minimum and maximum limits are calculated and displayed in the diagrams.



8. After the initial calculation, adjust the suggested limit values as needed.



9. After making the adjustments, click the "Save" button to save the filters in the filter list. You can define several filters in one "Operational State Filter". If you define multiple filters, each filter condition is applied separately to the input data. If the input data is within the ranges of a defined filter, the input is processed by the Model and KPIs are created.

Result

A new filter has been defined in the "Operational State Filter".

9.4.5 Editing filters

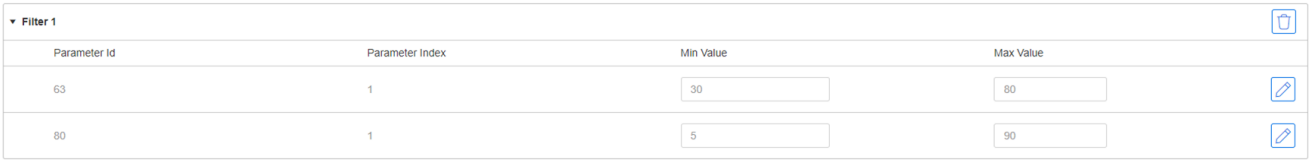
Proceed as described below to change the minimum and maximum threshold values for a parameter of the filter.

Requirement

A filter is defined in the "Operational State Filter".

Procedure

1. In the navigation bar, click on "Operational State Filter". The "Operational State Filters" page is displayed.
2. Click on the button of the "Operational State Filters" you want to edit. The "Operational State Filters Details Page" is displayed.
3. Click the button of the parameter that you want to change.



4. After making changes, click the button to confirm or click the button to cancel.
5. The changes are not applied immediately. To apply changes, click the "Apply Changes" button.
   If you exit the page without applying the changes, the changes may be lost.
   After applying the changes with "Apply Changes", related "Use Cases" are restarted so that the changes are applied.

Result

The filter values have been updated and applied to the "Use Cases".

9.4.6 Deleting filters

Proceed as described below to delete a filter.

Requirement

A filter is defined in the "Operational State Filter".

Procedure

1. In the navigation bar, click on "Operational State Filter". The "Operational State Filters" page is displayed.
2. Click the button of the "Operational State Filters" for which you want to delete a filter. The "Operational State Filters Details Page" is displayed.
3. Click the button of the filter you want to delete.
4. Click the button to confirm or click the button to cancel.

Result

The filter has been deleted.

9.5 Settings

This section describes steps for changing time zone settings and configuring data bus login information.

9.5.1 Changing the time zone

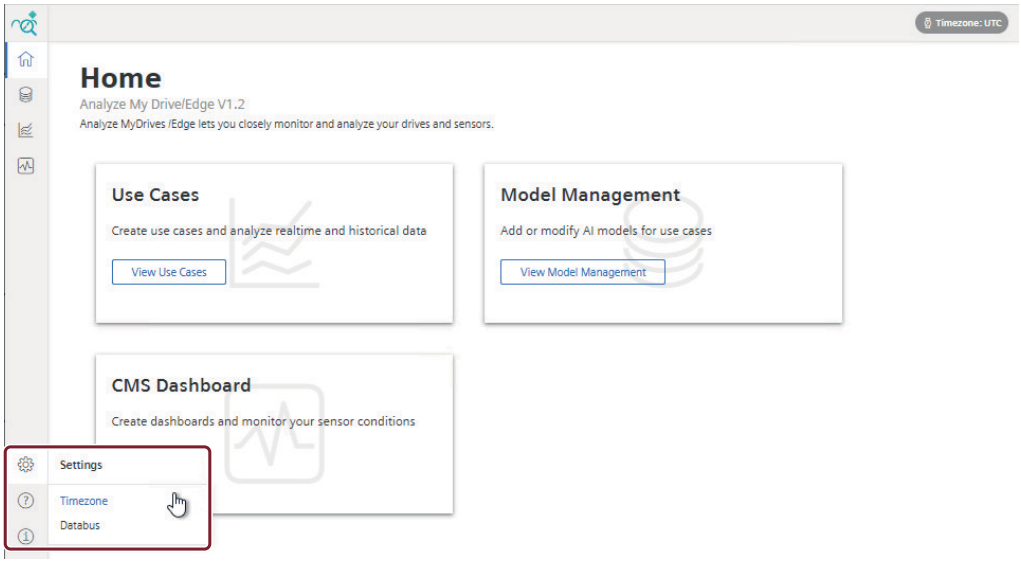
Proceed as described below to change the time zone.

Note

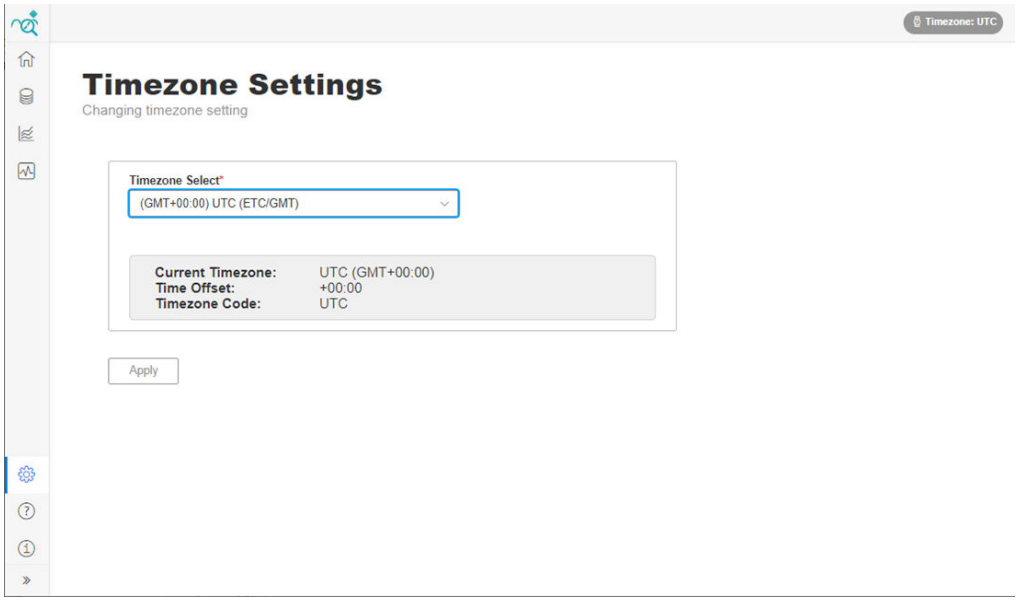
If the selected time zone is not correct, KPIs specified in "Overview Widgets" and "Details Pages" of "Use Cases" can shift. Hour or date information becomes inaccurate.

Procedure

1. In the navigation bar of the home page, click on "Time zone" under "Settings".



2. Select the required time zone from the drop-down menu.



3. Apply the selection with "Apply".

Result

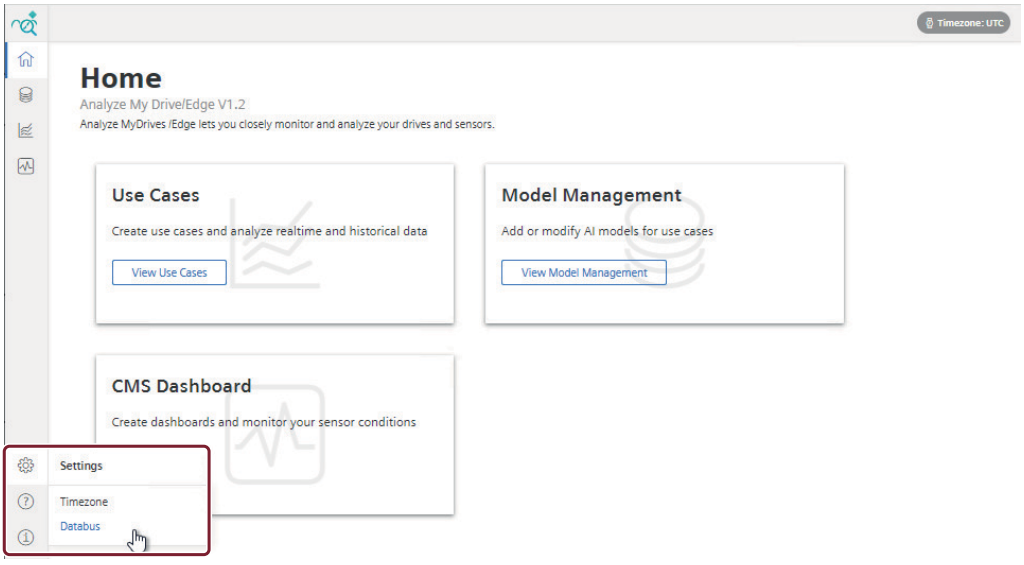
The new time zone has been set in the application.

9.5.2 Change Databus credentials

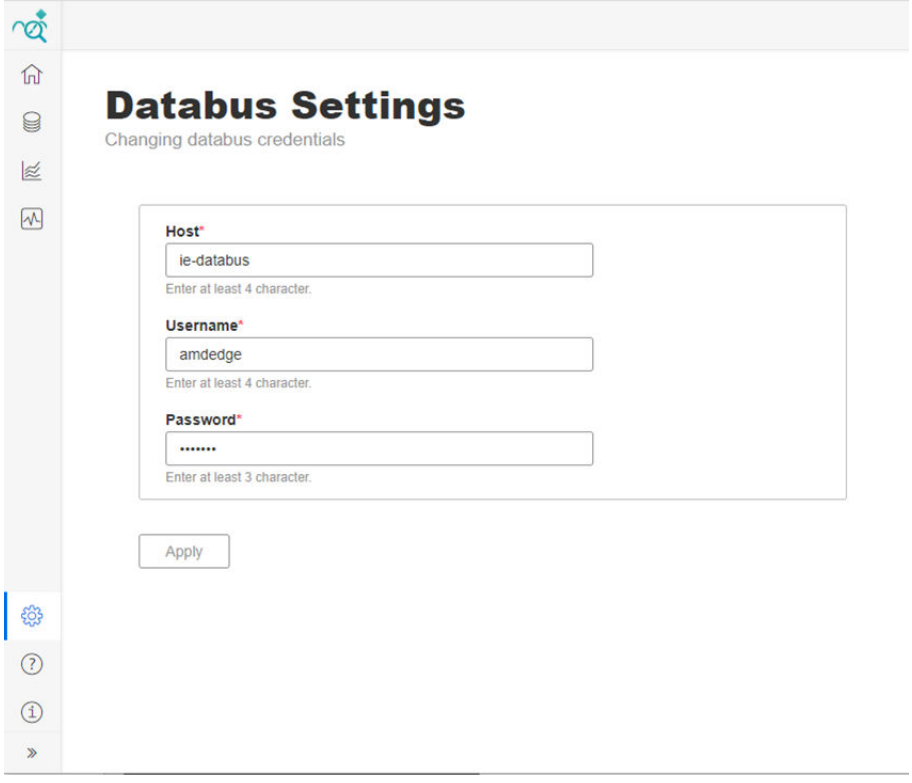
Proceed as described below to change the Databus credentials.

Procedure

1. In the navigation bar of the home page, click on "Databus" under "Settings".



2. Configure the log-on information for Databus.



3. Apply the selection with "Apply".

Result

The log-on information has been changed.

Mechanical Anomaly Detector Starter Kit 10

This section describes the procedures for training the "Mechanical Anomaly Detector" model.

Note

The Starter Kit for the "Mechanical Anomaly Detector" can be downloaded via the help area of the application.

10.1 Introduction

10.1.1 Mechanical Anomaly Detector Starter Kit

The Starter Kit for the "Mechanical Anomaly Detector" supports you when training and establishing AI models for "Mechanical Anomaly Detector" applications of Analyze MyDrives /Edge. These applications close the gap between data acquisition using SINAMICS STARTER or the Visual Flow Creator and provision of the model in the Analyze MyDrives /Edge application. The integration of AI and Analyze MyDrives /Edge offers an efficient and convenient environment to introduce artificial intelligence into production and to optimize the various steps in your development lifecycle.

The "Mechanical Anomaly Detector" component in Analyze MyDrives /Edge provides AI models that support the first stages of the life cycle of AI models, such as data acquisition, model training, model deployment and model inference. These models can be run in Analyze MyDrives /Edge on different SIEMENS hardware products, which are compatible with the SIEMENS Industrial Edge platform. Additional hardware, which is compatible with SIEMENS Industrial Edge is listed on the official homepage for Industrial Edge (https://www.siemens.com/de/de/produkte/automatisierung/themenfelder/industrial-edge.html).

The models for the "Mechanical Anomaly Detector" are completely integrated in the Analyze MyDrives /Edge application, which is completely compatible with system applications and other applications provided by SIEMENS.

10.1.2 Mechanical Anomaly Detector Starter Kits functionality

The Starter Kit for the "Mechanical Anomaly Detector" is a Notebook Template, which allows you to create, train and package AI model pipelines using the AI Software Development Kit (AI SDK). AI SDK is a Python library with which you can create, package and test AI Inference Pipelines. It provides building blocks for automating these steps, and contains fundamental Scikit-learn pipeline elements for processing time series.

When AI SDK is delivered, it includes project templates, such as the Starter Kit for the "Mechanical Anomaly Detector", which provides workflows based on notebooks for training models, and packages these with a suitable configuration file for deployment in Analyze MyDrives /Edge.

The Starter Kit for the "Mechanical Anomaly Detector" assumes a machine learning workflow that includes the following steps:

- Preparing data for training
- Training models
- Packaging models into an inference pipeline
- Adapting the pipeline configuration
- Deploying packaged inference pipelines to Analyze MyDrives /Edge

The Starter Kit for the "Mechanical Anomaly Detector" is designed to be used exploratively from interactive Python Notebooks, but can also be adapted to be used from a pure programming perspective as part of an automated ML workflow.

Most of the Starter Kit functions for the "Mechanical Anomaly Detector" are independent of any specific application. The trained model pipeline is known to work for use cases such as constant speed, belt elongation, chain elongation and tray blockage monitoring and can be modified to address additional use cases.

10.1.3 Information about the software license

Software from third-party suppliers

The Starter Kit for "Mechanical Anomaly Detection" contains open-source software and/or other software from third parties such as the AI Software Development Kit.

10.2 Security instructions

10.2.1 Protection of the host computer

Protect your host computer against unauthorized access. Apply the following measures:

- Deploy the host computer only in isolated system networks, never in office networks.
- Activate the screen saver and lock the screen when leave your work area.
- Install suitable anti-virus software.
- Install updates and patches for the operating system and software on the host PC at the earliest opportunity.

10.2.2 Notes on protecting administrator accounts

A user with administrator rights has extensive access to the system, including possibilities of manipulating it.

Therefore, ensure there are adequate safeguards for protecting the administrator accounts to prevent unauthorized changes. To do this, use secure passwords and a standard user account for normal operation. Other measures, such as the use of security strategies, should be applied as required.

10.2.3 Notes regarding use

Accessing the Starter Kit for the "Mechanical Anomaly Detector" is only possible from the host computer. **DO NOT** allow any other computer in the system network to access the Starter Kit for the "Mechanical Anomaly Detector".

- The current Starter Kit for the "Mechanical Anomaly Detector" can only be used for non-security-critical applications.
- The Starter Kit for the "Mechanical Anomaly Detector" saves the project data without encryption on the host computer. You are solely responsible for "CIA" (Confidentiality, Integrity and Availability) for the created, saved, downloaded or exported files from the Starter Kit for the "Mechanical Anomaly Detector".
- The Starter Kit for the "Mechanical Anomaly Detector" is designed for the use in conjunction with JupyterLab; it encompasses a web server, which can be locally and remotely accessed. You are responsible for configuring JupyterLab with activated HTTPS (https://jupyter-notebook.readthedocs.io/en/stable/public_server.html#using-ssl-for-encrypted-communication).
- If you use the Starter Kit for the "Mechanical Anomaly Detector" to create pipeline configuration packages, carefully ensure that you only incorporate source code and Python packages from trusted sources.
- If you use the Starter Kit for the "Mechanical Anomaly Detector" to run pipeline configuration packages locally, ensure that you only use pipeline configuration packages from trusted sources.

10.3 Installing Mechanical Anomaly Detector Starter Kit

10.3.1 Installing and running

You can use the Jupyter Notebook template for the Starter Kit for the "Mechanical Anomaly Detector" with a Notebook Editor of your choice, or you can run the code in a pure Python 3.8 environment.

To become familiar with the Starter Kit for the "Mechanical Anomaly Detector" we recommend starting with the project templates. These are sample solutions that contain all the necessary dependencies and allow for a quick and smooth start. Once you have left your interactive exploration phase, you can consider switching to a pure Python approach and adjust the model pipelines to address your needs.

Requirements

First, ensure that you have internet access. If you access the internet through a proxy, e.g. because you are working in a corporate network directly or via VPN, ensure that you have configured pip and `Conda` to use the correct proxy. Setting environment variables `http_proxy` and `https_proxy` covers both.

More information on alternative solutions:

- Using a proxy server (https://pip.pypa.io/en/stable/user_guide/#using-a-proxy-server)
- Using Anaconda behind a company proxy (https://docs.anaconda.com/anaconda/user-guide/tasks/proxy/)

Using the Starter Kit for the "Mechanical Anomaly Detector" with project templates

Individual "zip" packages with project templates encompass a prepared work directory with Notebooks and sources. You can use these in a Python or JupyterLab environment of your choice. Preconditions for this are:

- Python 3.8 is either native (https://www.python.org/downloads/release/python-3810) or installed using Conda.
- A Notebook Editor such as Jupyter Notebook, JupyterLab or Visual Studio code.

The Notebook template uses 7zip to encrypt the model package, which is to be uploaded to "Analyze MyDrives /Edge". Ensure that the path to the "7zip" program is added to the environment variables.

Set up a separate Python environment specifically for the project template, as described in the README for the template.

To create the Python environment you can use your preferred Python Environment Manager. Under Windows, open the command prompt as administrator. To do this, press the Windows key on your keyboard and enter cmd. In the command prompt, enter the following commands, e.g. for Conda or Python `venv`.

```
# create Conda environment including Python and activate it

conda create -n mad_starterkit python=3.8.13
```

```
conda activate mad_starterkit

# create Python virtual environment on Linux and activate it

python -m venv ~/venv/mad_starterkit

source ~/venv/mad_starterkit/bin/activate

# create Python virtual environment on Windows and activate it

python -m venv %USERPROFILE%\venv\mad_starterkit

%USERPROFILE%\venv\mad_starterkit\Scripts\activate.bat
```

Once the environment is created and activated, you must register it as an interactive Python kernel so that it becomes available in your Notebook Editor. This is usually achieved with the following commands:

```
# install and register interactive Python kernel

pip install ipykernel

python -m ipykernel install --user --name mad_starterkit --display-
name "Python (mad_starterkit)"
```

Your Python environment is now ready to use for the project template. Unzip the project template from its package, change the working directory to the extracted project folder and execute the following command.

```
# install packages required for the template including the AI SDK

pip install daielib/

pip install -r requirements.txt -f <root directory path of starter
kit>
```

Make another attempt in order to resolve the problem if this command fails in the Windows operating system.

Once the required packages have been installed, the notebooks can be explored, starting with the Notebook login screen and executed in your Notebook Editor.

Ensure that you select a suitable interactive Python kernel to execute the Notebooks, which in this example is: `Python (mad_starterkit)`.

Note that by default, pip installs the latest version of required packages that are compatible with the AI SDK and the project template. If you want to make sure to use the versions that are listed in Readme_OSS, you can apply the appropriate constraints during installation as follows:

```
pip install -r requirements.txt -f <directory path containing
simaticai wheel> -c constraints.txt
```

Note that this increases the probability that `pip` installs older package versions with associated security issues.

After installing Python 3.8, you can install Jupyter Notebook or JupyterLab with the following command at the command prompt:

```
pip install jupyterlab
```

or

```
pip install notebook
```

After installation, start Jupyter Notebook or JupyterLab with the following command in the command prompt:

```
jupyter lab
```

or

```
jupyter notebook
```

10.4 Using Mechanical Anomaly Detector Starter Kit

10.4.1 Preparing data for training

Preparing data for training the models is predominantly outside the scope of Starter Kits for the "Mechanical Anomaly Detector". Prepared data set examples are available for the template pipeline, allowing you to try out these templates without involving tedious data acquisition. The Starter Kit includes a template for the Visual Flow Creator, which supports you when downloading a data set example from the Drive System Framework. Alternatively, data can be acquired using the SINAMICS STARTER.

Processing time series data

The Starter Kit for the "Mechanical Anomaly Detector" provides basic building blocks for establishing machine learning (ML) models for specific use cases such as belt elongation, chain elongation and tray blockage monitoring that process time series of aligned speed (SINAMICS parameter r63) and torque (SINAMICS parameter r80) signals with a sampling time of 2 ms. Aligned signals mean that the input of the processing pipeline consists of rows, containing a value for each signal. For example, a row consisting of signals from the SERVO_02 drive object and a time stamp looks like this:

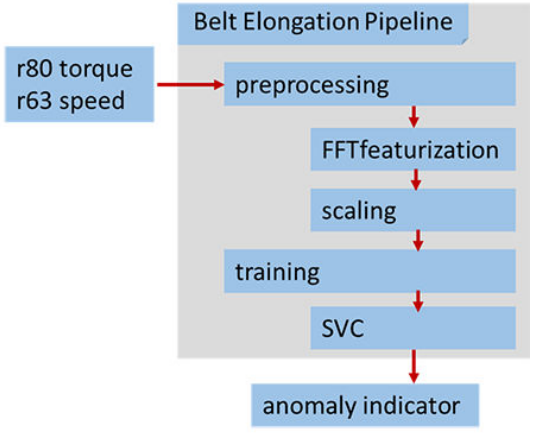
Time stamp (ms)	C0:SERVO_02.r63	C1:SERVO_02.r80
0	1.2	202

To receive the appropriate data from SINAMICS converters, set the TRCData parameters in the SINAMICS converter to the appropriate settings:

1377	p32040		TRCDATA Aufzeichnungsmodus	[1] Aufzeichnung kontin...		IBN (P9=3)	3		
1378	p32041		TRCDATA Abtastzeit Faktor Signalabtastung	[16] T = p0115[0] * 16		IBN (P9=3)	3		
1379	p32042		TRCDATA Abtastzeit Faktor Aufzeichnungspuffer	1		IBN (P9=3)	3	1	32000
1380	r32043		TRCDATA Abtastzeit Aufzeichnungspuffer Anzeige	2.000	ms		3		
1381	p32044		TRCDATA Aufzeichnungspuffer Datenpunkte Anzahl	600		IBN (P9=3)	3	1	65535
1382	p32045		TRCDATA Float Signalquellen Anzahl	2		IBN (P9=3)	3	0	36
1383	p32046		TRCDATA Integer Signalquellen Anzahl	0		IBN (P9=3)	3	0	12
1384	⊟ p32047		TRCDATA Float Signale Verdichtungsoperation						
1385	├ p32047[0]		TRCDATA Float Signale Verdichtungsoperation	[3] Mittelwert		Betrieb	3		
1386	└ p32047[1]		TRCDATA Float Signale Verdichtungsoperation	[3] Mittelwert		Betrieb	3		
1387	⊟ p32049		CI: TRCDATA Float Signale Signalquelle						
1388	├ p32049[0]		CI: TRCDATA Float Signale Signalquelle	SERVO_02 : r80		Betrieb	3		
1389	└ p32049[1]		CI: TRCDATA Float Signale Signalquelle	SERVO_02 : r63		Betrieb	3		
1390	p32055		BI: TRCDATA Freigabe	1		Betrieb	3		
1391	⊟ r32056		CO/BO: TRCDATA Zustandswort	17H			3		
1392	├ r32056.0		Freigabe aktiv	Ja			3		
1393	├ r32056.1		Aufzeichnung läuft	Ja			3		
1394	├ r32056.2		Aufzeichnungsmodus "Kontinuierlich" eingestellt	Ja			3		
1395	├ r32056.3		Aufzeichnungsmodus "Einmalig" eingestellt	Nein			3		
1396	└ r32056.4		Aufzeichnungsdaten lesbar	Ja			3		
1397	r32057		CO: TRCDATA Datenpuffer intern Füllstand	92.833252	%		3		
1398	⊟ r32058		CO: TRCDATA Aufzeichnungspuffer Zeitstempel						
1399	├ r32058[0]		Millisekunden	4124820			3		
1400	└ r32058[1]		Tage	488			3		
1401	r32059		CO: TRCDATA Aufzeichnungspuffer Intervallzähler	501628			3		

Select reading out the corresponding data in the Drive System Framework (DSF):

The Starter Kit for the "Mechanical Anomaly Detector" supports you when creating a time series pipeline that processes a stream of such rows according to the following pattern:



The role of pipeline elements is as follows:

- Preprocessing applies a function calculation with a Fast Fourier Transformation and a scaling to the input rows of torque and speed time series.
- When training, a trained support vector classifier is applied to the functions and an anomaly indicator is output.

Most real pipelines contain other pre-processing elements, such as imputers to enter missing values or windows to select parts of the time series. These pipelines can be used by appropriately modifying the template.

To train the classifier in such a pipeline, the input data must be routed through the preprocessing steps during the training process.

Therefore, this processing pipeline must be defined as part of data preparation before training. This is where the blocks in the AI SDK play a role.

These blocks are based on a well-established Python package `Scikit-learn` for machine learning. Scikit-learn provides a framework to define pipelines, which allows you to combine data transformers with classifiers or other types of estimator functions. The blocks are located in the `simaticai.pipeline` module. The main blocks include:

- `WindowTransformer`, which transforms a series of input rows into a series of windows of rows
- `FeatureTransformer`, which transforms a window of rows into the function values according to user-defined functions.

In addition to these transformers, there is a transformer called `FillMissingValues`, which corrects input data for simple cases. For more difficult cases, you should use a more sophisticated imputer to correct your input data.

References:

- Details and specific examples are provided in the training notebooks in the project template for "Mechanical Anomaly Detection" and in the reference manual for AI SDK API.
- Documentation on Scikit-learn is available if you want to know the principle of operation of Scikit-learn in detail, or you would like to use your own transformers.

Mapping predicted classes of data windows to data points

As described in the previous chapter, time series data is typically classified window by window. This means that the class of a single data row is not defined per se. Nevertheless, there are cases where it is convenient to map the classes defined window by window to the data points themselves. One such case could be if you want to visualize the data points according to their classification, e.g. plot the data points color-coded with their classes.

AI SDK provides the utility function `ack_propagate_labels` in module `simaticai.pipeline` to perform this mapping.

References:

For more details and specific examples, refer to the AI SDK API reference manual.

10.4.2 Training models

The AI SDK does not restrict how you train your models and save the trained models. You can review the training notebooks in the project templates as examples; however, you can select any ML libraries that are compatible with Analyze MyDrives /Edge AI Runtime. The project templates include examples of Scikit-learn.

Some ML frameworks, such as Scikit-learn, rely on persistent Python runtime objects. In the latter case, when saving objects after training and when calling in the Analyze MyDrives / Edge AI Runtime, the same versions of the Python libraries are available. The packaging function of the AI SDK supports this by requiring exact version specifiers for Python packages required in a pipeline package.

To train a model for "Mechanical Anomaly Detection" to identify specific errors as anomalies, unzip the template for the Starter Kit for the "Mechanical Anomaly Detector". Save the acquired data in the good and fault cases as "csv" files, with data in rows and column headers as shown above, in two directories in your development environments, e.g. '../data/raw/s120minimodel/0/' and '../data/raw/s120minimodel/1/'. Execute file "20a-TrainModelAMDEdge_s120minimodel.ipynb" in a Jupyter notebook sequence in the Python environment. Alternatively, you can also change the naming of each data column in the Jupyter Notebook script. The outcome is a trained model saved under "../models/beltelongation_s120minimodel-model.joblib".

10.4.3 Packaging models into an inference pipeline

The AI SDK provides the functionality to create a pipeline configuration package that encompasses trained models. By adding and appropriately adapting configuration file "config.json", the package can be uploaded to Analyze MyDrives /Edge AI Runtime and can be run on Analyze MyDrives /Edge AI Runtime on an Industrial Edge Device. The appropriate functions are located in the simaticai.deployment module and the "config.json" file can be found in the starter package. The Starter Kit for the "Mechanical Anomaly Detector" supplies a template for Jupyter Notebook to create a package suitable for Analyze MyDrives /Edge AI Runtime.

Creating pipeline components implemented in Python

The implementation of an inference pipeline component in Python is described in detail in Section "Customizable AI Model" (Page 143).

In the following, an overview is only provided for models that are compatible with "Mechanical Anomaly Detector" in Analyze MyDrives /Edge

A component comprises files and metadata.

- The files contain:
  - Python scripts
  - Trained models
- Metadata encompasses:
  - Component name, component version
  - Required Python version and Python packages
  - Input and output variables
  - The entry point

For project templates for the AI SDK , like the "Mechanical Anomaly Detector", template files, such as files with source code and saved trained models, need to be arranged in the file system in a predefined structure. The idea is that you keep the same relative structure for the Analyze MyDrives /Edge AI Runtime, which allows you to use the same relative references from the source code to saved models or other files.

Put together all the files for the components. In the template for "Mechanical Anomaly Detector", there should be at least one Python script for the entry point, the Inference Wrapper, a configuration file and a saved model. Create the pipeline component by running a Python script or Notebooks corresponding to the CreatePipelinePackage_s120minimodel.ipynb example that provides the following functionality and then add a configuration file:

- Creates a PythonComponent object with a specific name, a specific component version and a required Python version
- Defines the required Python packages
- Defines input and output variables
- Adds Python scripts and saved models
- Defines the entry point under Python scripts

All of this is realized with the corresponding functionality of the `simaticai.deployment` module. Concrete examples are provided in the Notebook packages in the project templates. For more information and advanced options, see the AI SDK API reference manual.

Take into consideration the following restrictions:

- AI SDK allows you to select one of the required Python versions, which is supported by various versions of Analyze MyDrives /Edge AI Runtime.
- Make sure you select a Python version that is supported by the version installed on your Industrial Edge target device. At the time that this document was created, this is Python version 3.8.
- The required Python packages must be added as Wheels to the pipeline components.
- At the time that this document was created, there is a limitation that the entry point script inference.py must be in the root folder of the package and must not be changed.

Configuring and saving the pipeline

After you have created the components, even if it is only a single one, you must configure the pipeline by using a `Pipeline` object.

For packaging a model for the "Mechanical Anomaly Detector" into an inference pipeline make sure that the trained model is saved under "../models/beltelongation_s120minimodel-model.joblib" or adapt the file name in the CreatePipelinePackage_s120minimodel.ipynb template accordingly.

Then modify name and version of the package and run the Notebook. The resulting model file is available in the package folder as a compiled "*.zip" and in an encrypted "*.zip" version. Before running the Notebook make sure that the storage location of the executable zip.exe, e.g. C:\Program Files\7-Zip, is added to the path environment variable on your computer. Add a configuration file config.json to the inference/src/si folder of the "zip" file in which inputs are described and how you would like to visualize the model output. Before you Upload the encrypted "zip" file into the model management of Analyze MyDrives /Edge, ensure that all of the required libraries are available for your Pipeline in the src folder.

Configuration file config.json of the model for the "Mechanical Anomaly Detector" is shown below:

```
{
    "name": "Belt Elongation",
    "version": 0.9,
    "family": "Mechanical Anomaly",
    "type": "Generic",
    "model": "belt",
    "read_once_input_configuration": [],
    "user_defined_configuration": [],
    "continuous_input_configuration": [
        {
            "id": "63[0]",
```

```
            "key": "speed",
            "type": "float",
            "description": { "en": "Actual Speed Smoothed"},
            "frequency": 500
        },
        {
            "id": "80[0]",
            "key": "torque",
            "type": "float",
            "description": { "en": "Actual Torque Smoothed"},
            "frequency": 500
        }
    ],
    "output_configuration": [
        {
            "name": "anomaly_score",
            "description": { "en": "Anomaly Score"},
            "type": "float",
            "aggregation": [
                {
                    "strategy": "avg",
                    "period": "24h",
                    "window": "1m"
                },
                {
                    "strategy": "avg",
                    "period": "120M",
                    "window": "1h"
                }
            ]
        }
    ],
    "execution_configuration": {
        "strategy": "window_length",
        "execution_period": "1200",
```

```
            "result_storage_period": "24h",
            "configurable": true
        },
        "trigger": {}
}
```

Change the input configurations "read_once_input_configuration", "user_defined_configuration" and "continuous_input_configuration" not. By modifying the aggregation window, strategy and period, the visualization of the anomaly score output is adapted as needed.

For current application cases, an "execution_period" of 1200 points and a "storage_period" of 24h have proven themselves in practice.

10.4.4 Locally testing the pipeline configuration package

Once you have created your inference pipeline package, you can proceed directly to the deployment in Analyze MyDrives /Edge AI Runtime.

Note

If you plan the deployment on the AI Inference Server, then locally test your package before deployment.

The advantages of local testing are as follows:

- You can identify many potential issues more quickly because you need not to go through a deployment cycle.
- You can diagnose and troubleshoot problems much more easily because you can inspect artifacts in your development environment.
- You can validate your debugs faster and move on to other issues that you haven't been able to cover due to previous issues.
- You can easily include local pipeline tests in the test automation of your build process.

Two tools for local testing

You can apply state-of-the-art software engineering practices such as unit testing and test-driven development.

This means that ideally, you already have automated Unit Tests or even integration tests that ensure that the Python code and saved models work in isolation as expected. This helps you localize errors when you assemble these parts and integrate them as a pipeline configuration package.

The AI SDK-Paket `simaticai.testing` provides two tools for local testing:

- A pipeline validator that performs static validation of the package for the availability of required Python packages.
- A pipeline runner that allows you to simulate the execution of your pipeline in your Python environment.

Note, that all of these functions apply to pipeline configuration packages, not to Edge configuration packages. You must use this before you convert your pipeline configuration package using AI Model Deployer into an Edge configuration package.

Since the conversion itself is done automatically by AI Model Deployer, most of the potential issues are already present in the package before the conversion, so a post-conversion verification would only delay the identification of these issues.

Static validation of a pipeline package

You can transfer your pipeline configuration package to the validate_pipeline_dependencies function in the simaticai.testing.pipeline_validator submodule to perform static checks. These checks include:

- Verify that the required Python version in the package is supported by a known version of the AI Inference Server.
- Verify that all required Python packages are either included in the pipeline package itself or available on pypi.org for the target platform.

References:

For specific programming details, see the AI SDK API reference manual.

Local execution of a packaged pipeline

Class `LocalPipelineRunner` in the `simaticai.testing.pipeline_runner` submodule can be used to locally mimic the behavior of the AI Inference Server for loading and running inference pipelines. This is a quick and easy way to find programming or configuration errors before deploying the package.

The local pipeline runner simulates the server environment as follows:

1. It unpacks the pipeline components into a test folder, similar to what would happen on the inference server.
2. It creates a separate Python virtual environment for each component.
3. It installs the required Python packages from the Wheels, if these are provided in the package or via `pypi.org`.
4. It installs the mockup of the `log_module` (see Mockup of the logging of the AI Inference Server (Page 141))
5. It feeds the pipeline with input data by triggering the entry points of the components accordingly.
6. It collects the sequence of pipeline outputs for a given sequence of pipeline inputs.

You can also use the local pipeline runner to run your pipeline component by component. You can feed individual components with inputs and verify the output produced.

From a testing strategy and risk-based testing perspective, we recommend that you validate the business logic within the pipeline components in unit tests as you would with any ordinary Python program and use the local pipeline runner to cover test risks such as the following:

- Deviation between pipeline and component input and output variable names
- Required Python packages not specified
- Source or other files are missing from the package
- Interface deviations between subsequent pipeline components
- The entry point cannot process input data due to a deviation in the data format.
- The entry point generates output data in the wrong format
- For some reason, the pipeline does not work as consistently as intended

An essential point for conducting a thorough local test concerning data input and output formats is to understand how data connections work in the AI Inference Server. The following data connection types are known:

- IE Databus
- IE MQTT Connector
- IE Vision Connector

For these data connection types, the AI Inference Server transfers the databus or MQTT payload as a string directly as the value of the connected pipeline input variable. In many use cases where this data connection type applies, your pipeline has a single input variable of type string. This means that you must assign the local pipeline runner a Python dictionary with a single element. For a concrete code example that shows how to feed a pipeline with a single string input variable in a local test, see the Image Classification project template in the Local Pipeline Test Notebook for AI SDK.

The SIMATIC S7 Connector data connection type requires more attention. This connector is typically used in time series use cases. Using this connection, the AI Inference Server processes the data format used by the S7 Connector and only transfers the values of the PLC variable values, but not the metadata. This means that if you intend to use your pipeline with the S7 Connector, you need to feed it with dictionaries containing the PLC variable values. A specific code example, which shows how a pipeline in a local test is fed an input row with PLC variable values, is provided in the template for project State Identifier in the Local Pipeline Test Notebook for AI SDK.

Restrictions relating to the local pipeline execution

The local Runner works with batches of input data and processes the whole input batch component by component. In the case of a sequence of pipeline inputs, the entire sequence is first processed by the first components, and only then is the output of the first component processed by the second component. This is different from the runtime environment of the AI Inference Server, where the components in the pipeline potentially start taking input data as soon as the preceding component has produced outputs.

You cannot fully test input and output data formats, as these depend on the data connection settings in the AI Inference Server, and you must provide the local Runner the input data in the representation that matches the output side of the connector. This means that if your assumptions on the data connection settings or the resulting data formats are wrong, your

tests will also provide misleading results. The local Runner can only simulate linear pipelines, where the pipeline input variables are only used by one component, each component uses only the output of the previous component, and the pipeline output only comprises variables from the last component. The local Runner does not simulate any inputs from AMD/Edge AI Runtime.

Furthermore, the results obtained in local tests are not fully representative for AI Inference Server, including but not limited to the following aspects:

- The local version of Python can differ from that on the inference server.
- The local architecture may be different, resulting in different Builds of imported Python packages being used.

Despite these restrictions, you should locally test your pipeline before deployment on the AI Inference Server. With these local tests, you are more likely to save time than when skipping the tests.

10.4.5 Mockup of the logging of the AI Inference Server

The Python environment on the AI Inference Server feeds a Python module with the designation `log_module` so that the Python scripts can be used for logging on the server. To be able to run the same code in a local development environment on a PC, the AI SDK provides a mock of the `log_module` as Wheel, which you can install, import and use in the same way. The Wheel file must not be included in the dependencies of the pipeline package.

10.4.6 Deployment of the packaged inference pipeline for AI @Edge

After you created and tested your inference pipeline, you can deploy it to the AI Inference Server. Observe the following points at the time of creation

1. Use the AI Model Deployer to convert the pipeline configuration package, created in AI SDK, into an Edge configuration package.
2. Deploy the Edge configuration package to an Industrial Edge Device by uploading it using the AI Inference Server application.

In the future, there will be more options to perform the required conversion and deployment.

If you want to deploy the "Mechanical Anomaly Detector" model created with the Analyze MyDrives /Edge Starter Kit on the Analyze MyDrives /Edge, upload the encrypted "zip" file to the Model Management page on Analyze MyDrives /Edge.

Customizable AI Model

11

11.1 Introduction

The procedure to create and configure a "Customizable AI Model" is described in this section.

The term "Customizable AI Model", as used in this documentation, refers to "models" created and uploaded by users.

"Customizable AI Models" provide you with the possibility of creating your own AI models to analyze certain conditions. You can connect these models with "Low-Speed"- or "High-Speed-Adapters", specify input parameters and output key performance indicators and create "Overview Widgets" and "Details pages". Further, you have the option of adapting the "Customizable AI Models" as required.

11.2 Creating a "Customizable AI Model"

The "Analyze My Drives/Edge" application provides you with an environment in which Machine Learning models can be run. The models are generated using the "AI Software Development Kit" and with the required configuration and "inference.py" files.

To become familiar with the model development process, start by adapting the project template example of the "AI Software Development Kit" or with a user-defined project template. You can download the project template from Siemens Industry Online Support (SIOS) from the "Analyze My Drives/Edge" page.

11.2.1 AI Software Development Kit

"AI Software Development Kit", briefly "AI SDK", is a Python library that you can use to create Machine Learning models for the "Analyze My Drives / Edge" application.

The "SDK" is accompanied by project templates, which provide Notebook-based workflows for training models. This also includes packing the models for deployment and testing them. Project template "State Identifier" contains basic scikit-learn pipeline elements for processing time series.

The "AI Software Development Kit" assumes machine learning Workflow which encompasses the following steps:

- Preparing data for training
- Training models
- Packing models in an inference pipeline
- Testing packed inference pipelines
- Converting and deploying inference pipelines for the "Analyze My Drives/Edge" application

To create a "Customizable AI Model", general knowledge about the "AI Software Development Kit" is required. More detailed information is provided in the "AI SDK operating instructions" (https://support.industry.siemens.com/cs/document/109814457/ai-software-development-kit).

11.2.2 Adapting AI SDK project templates

You must adapt the project template example of the "AI SDK" using the associated "config.json" and "inference.py" files. The configuration file format is described in detail in the next chapter.

The "ZIP" file of the model is generated as soon as the pipeline is executed with the required changes.

After the model has been generated, you must encrypt the "ZIP" file of the model using the public key of the "Analyze My Drives/Edge" application provided. The batch file example (Windows) for encrypting the "ZIP" file of the model is provided at Siemens Industry Online Support (SIOS) on the "Analyze My Drives/Edge" page.

11.2.2.1 Inference.py file format

You must modify the file corresponding to the following content. There are two functions with which the "Analyze My Drives /Edge" application interacts.

Function 'init' is called if the model is run for the first time with "read once" and user-defined parameter values.

Function 'run' is periodically called depending on the run strategy, which is defined in the configuration file with continuous parameter values.

```
import os, sys
from pathlib import Path
import json
import numpy
import pandas
import joblib

from itertools import product

current_path = os.path.dirname(os.path.abspath(__file__))
sys.path.insert(0, current_path + "/src")

with open(current_path + '/inference-model.joblib', 'rb') as rpl:
    model = joblib.load(rpl)

pipe = model['pipe']

def init(conf: str):

    configuration_data  = json.loads(conf)

    pipe.set_params(featurization__conf = configuration_data)

def run(data: str):

    input_data = json.loads(data)
    output = pipe.predict(input_data)

    return output
```

11.2.2.2 Folder hierarchy for project templates

You must save configuration and modified inference files to folder "src/si" of the project template example.

The model automatically generated by the pipeline has the following folder hierarchy:

mymodel.zip file
- mymodel (folder with same name)
  - inference (folder)
    - inference-model.joblib
    - inference.py (modified)
    - requirements.txt
    - src (folder)
      - simaticai (folder)
      - si (folder)
        - config.json (created)

11.2.2.3 Encrypting the model ZIP file

You must encrypt the ZIP file of the model generated by the pipeline using the public key of the "Analyze My Drives/Edge" application.

Use the Windows batch file provided to create an encrypted ZIP file. The "Analyze My Drives/Edge" application only accepts encrypted model files.

11.2.3 Using the "Analyze My Drives/Edge" template example

The template example for the "Analyze My Drives /Edge" application contains all of the required adaptations and Tools. The template example allows you to quickly familiarize yourself and adapt and create a compatible model.

11.2.3.1 Installing and running

You can use the Jupyter Notebook template for the Starter Kit of the "Example Custom Model" using a Notebook editor of your choice or you can run the Code in a pure Python 3.8 environment.

Preconditions

First, ensure that you have internet access. If you access the internet through a Proxy, e.g. because you are working in a corporate network directly or via VPN, ensure that you have configured pip and `Conda` to use the correct proxy. Setting environment variables `http_proxy` and `https_proxy` covers both.

More information on alternative solutions:

- Using a proxy server (https://pip.pypa.io/en/stable/user_guide/#using-a-proxy-server)
- Using Anaconda behind a company proxy (https://docs.anaconda.com/anaconda/user-guide/tasks/proxy)

Using the Starter Kit for the "Example Custom Model" with project templates

The "ZIP" packages with project templates contain a work folder with Notebooks and Sources. You can use these in a Python or JupyterLab environment of your choice.

Please observe the following requirements:

- Python 3.8 is installed either with native (https://www.python.org/downloads/release/python-3810) or with Conda.
- A Notebook editor such as Jupyter Notebook, JupyterLab or Visual Studio Code.

The Notebook template uses 7zip to encrypt the model package, which is to be uploaded to AMD/Edge. Ensure that the path to the "7zip" program is added to the environment variables.

Set up a separate Python environment specifically for the project template, as described in the README for the template.

To create the Python environment you can use your preferred Python Environment Manager. Under Windows, open the command prompt as administrator. To do this, press the Windows

key on your keyboard and enter cmd. In the command prompt, enter the following commands, e.g. for Conda or Python `venv`.

```
# create Conda environment including Python and activate it
conda create -n custommodel_starterkit python=3.8.13
conda activate custommodel_starterkit
# create Python virtual environment on Linux and activate it
python -m venv ~/venv/custommodel_starterkit
source ~/venv/custommodel_starterkit/bin/activate
# create Python virtual environment on Windows and activate it
python -m venv %USERPROFILE%\venv\custommodel_starterkit
%USERPROFILE%\venv\custommodel_starterkit\Scripts\activate.bat
```

Once the environment is created and activated, you must register it as an interactive Python kernel so that it becomes available in your Notebook Editor. This is usually achieved with the following commands:

```
# install and register interactive Python kernel
pip install ipykernel
python -m ipykernel install --user --name custommodel_starterkit --
display-name "Python (custommodel_starterkit)"
```

Your Python environment is now ready to use for the project template. Unzip the project template from its package, change the working directory to the extracted project folder and execute the following command.

```
# install packages required for the template including the AI SDK
pip install daielib/
pip install -r requirements.txt -f <root directory path of starter
kit>
```

Make another attempt in order to resolve the problem if this command fails in the Windows operating system.

Once the required packages have been installed, the notebooks can be explored, starting with the Notebook login screen and executed in your Notebook Editor.

Ensure that you select a suitable interactive Python kernel to execute the Notebooks, which in this example is: `Python (custommodel_starterkit)`.

Note that by default, pip installs the latest version of required packages that are compatible with the AI SDK and the project template. If you want to make sure to use the versions that are listed in Readme_OSS, you can apply the appropriate constraints during installation as follows:

```
pip install -r requirements.txt -f <directory path containing
simaticai wheel> -c constraints.txt
```

Note that this increases the probability that `pip` installs older package versions with associated security issues.

After installing Python 3.8, you can install Jupyter Notebook or JupyterLab with the following command at the command prompt:

```
pip install jupyterlab
```

or

```
pip install notebook
```

After installation, start Jupyter Notebook or JupyterLab with the following command in the command prompt:

```
jupyter lab
```

or

```
jupyter notebook
```

11.2.3.2 Adapting and generating the model

You must carry out the adaptation required after running the Jupyter environment. Review Section Mechanical Anomaly Detector Starter Kit (Page 125) if you intend to generate a training model. Adapt the model corresponding to your requirements and then execute the pipeline. An encrypted model ZIP file is generated when executing the pipeline.

11.3 Data format for "Customizable AI Model"

11.3.1 Continuous parameter

For "Customizable AI Model", you can configure "Data Sources" in configuration file. Continuous data are transferred on a regular basis to the execution function corresponding to the configuration in the configuration file. Two different "Data Sources" can be used for continuous data acquisition:

High Speed

Data with a frequency of up to 8 kHz are read via the High Speed Connector of the "Drive System Framework" from SINAMICS drives with CU 320-2 and TRCDATA technology extension. The data format for HS is a JSON object. It contains a time stamp and an array of values for each parameter.

Data format:

```
{
    // Starting timestamp of measurements
    "timestamp": 1678648677,
    // parameter name used as a key from configuration file and
values array
    "param_1": [23.3, 11.2, 4.5,...]
    // number of values defined in execution configuration
    "param_2": [2, 1, 3,...]
}
```

Low Speed

Data with a frequency of up to 10 Hz are read via the Low Speed Connector of the "Drive System Framework" from 2nd generation SINAMICS G drives. The data format for LS is a JSON array. It contains the measurement of the parameters for an individual time stamp per defined frequency.

Data format:

```
[
    {
        "time": 1678648677,
        "param_1": 4.54,
        "param_2": 1
    },
    {
        "time": 1678648777,
```

```
        "param_1": 4.32,
        "param_2": 2
    },
    ...
]
```

11.3.2 Read Once and User Defined parameters

For "Customizable AI Model", you can configure "Read Once" and "User Defined" parameters in the configuration file.

"Read Once" parameters are read out from SINAMICS drives via the API of the "Drive System Framework". Users enter "User Defined" parameters when creating a Use Case. These values are combined, and when the model is run for the first time, they are transferred once to the init function of the model.

Data format:

```
{
    "read_once_param_1": 4.32,
    "read_once_param_2": 1.09,
    "user_defined_param_1": 45,
    "user_defined_param_2": 32.3
}
```

11.4 Configuring "Customizable AI Model"

11.4.1 Overview

Input parameters, key performance indicators, Data Sources that are used and other configuration information are contained in file "config.json" in the ZIP file of the model. You can adapt the model configuration.

The individual configuration elements and their restrictions are described in this section.

Examples of configuration files are provided in Section "Configuration files - examples" (Page 168).

11.4.2 Structure of the configuration file

The configuration file contains the following elements:

```
{
    "name": "Custom Model",
    "version": 1.0,
    "family": "Custom HS Model",
    "type": "Generic",
    "model": "custom",
    "datasource": "HS",
    "read_once_input_configuration": [],
    "user_defined_configuration": [],
    "continuous_input_configuration": [],
    "output_configuration": [],
    "ui_config": {
        "widget_config": [],
        "details": []
    },
    "aggregation_method": "",
    "execution_configuration": {
        "strategy": "window_length",
        "execution_period": "1200",
        "result_storage_period": "24h",
        "configurable": true
    },
```

```
    "trigger": {
    }
}
```

11.4.3 Element "name"

Element "`name`" contains the model name.

Alphanumeric values are permissible.

Example:

```
"name": "Custom Model",
```

11.4.4 Element "version"

Element "`version`" contains the model version.

Floating point values are permissible.

An existing model from the same model family can be updated with a more recent version model if no changes are made to input parameters and key performance indicators.

Example:

```
"version": 1.0,
```

11.4.5 Element "family"

Element "`family`" contains the family of the model.

Alphanumeric values are permissible.

An existing model from the same model family can be updated with a more recent version model if no changes are made to input parameters and key performance indicators.

Example:

```
"family": "Custom HS Model",
```

11.4.6 Elements "type" und "model"

Elements "`type`" and "`model`" contain the model type.

The following assignment applies to all "Customizable AI Models":

```
"type": "Generic",
```

```
"model": "custom",
```

Note

Do not change these values!

11.4.7 Element "datasource"

Element "datasource" specifies the adapter type that is compatible with the model.

Possible values:

"HS" for "High-Speed-Adapter"

"LS" for "Low-Speed-Adapter"

Example:

```
"datasource": "HS",
```

Note

You can only use "LS" in conjunction with 2nd generation SINAMICS G drives.

11.4.8 Element "read_once_input_configuration"

Element "read_once_input_configuration" contains the configuration for the drive input parameters. These are only read once when "Use Case" is started. Parameter values read once are not updated if the model is run; this also applies even if these have changed on the drive side.

Up to 10 "Read Once" parameters can be defined for a user-defined model.

Every "Read Once" parameter has the following elements:

- id: Parameters that the drive should read.
  - Format: #ParameterId#[#ParameterIndex#]
  - Examples: 7845[0], 207[0]

  Note

  Enter 0 for the parameter index if the drive parameter is not of type Array.

- type: Parameter value type (supported values: "string", "integer", "float")
- description: Parameter description. You can enter the description for English (en) and German (de).

- `key`: Key value to access parameters in the model.
- `value`: Default value of the parameter. This value is overwritten if "Use Case" is started and real values are read from the drive.

Defining an individual "Read Once" parameter

```
"read_once_input_configuration": [
    {
        "id": "7845[0]",
        "type": "string",
        "description": {
            "en": "MLFB - Power Module/ Power Unit Order Number",
            "de": "MLFB - Leistungsteil Bestellnummer"
        },
        "key": "MLFB",
        "value": "6SL3210-1PE16-1AL1"
    }
],
```

Defining several "Read Once" parameters

If you define several parameters, you must separate each parameter definition by a comma after the closing curvy brackets.

```
"read_once_input_configuration": [
    {
    "id": "7845[0]",
        "type": "string",
        "description": {
            "en": "MLFB - Power Module/ Power Unit Order Number",
            "de": "MLFB - Leistungsteil Bestellnummer"
        },
        "key": "MLFB",
        "value": "6SL3210-1PE16-1AL1"
    },
    {
        "id": "207[0]",
        "type": "float",
        "description": {
```

```
                "en": "Rated power unit current [A]",
                "de": "Leistungsteil Bemessungsstrom [A]"
            },
            "key": "IN",
            "value": 8
        }
    ],
```

11.4.9 Element "user_defined_configuration"

Element "`user_defined_configuration`" contains the configuration for user-defined input parameters. These are not read from the drive, but are specified by the user when configuring the model. If "Read Once" parameters are read, as soon as they are updated via the model page, then the associated "Use Cases" are also updated.

Up to 10 "User Defined" parameters can be defined for a user-defined model.

Every "User Defined" parameter has the following elements:

- `name`: The name is used to access parameters in the model.
- `type`: Parameter value type (supported values: "`string`", "`integer`", "`float`")
- `mandatory`: Indicates whether the parameter is obligatory. It is not permissible that an obligatory parameter is empty.
  Possible values: "`true`" and "`false`".
  If "`true`", then the application executes the necessary check.
- `editable`: Specifies whether users can change the value of this parameter.
  Possible values: "`true`" and "`false`".
  If "true", then users cannot change the value of the parameter in operating area "Model Management".
- `description`: Parameter description. You can enter the description for English (`en`) and German (`de`).
- `default_value`: Specifies the default value of the parameter. If parameter "`editable`", then users can change the value of the parameter in operating area "Model Management".

Defining an individual "User Defined Configuration" parameter

```
"user_defined_configuration": [
    {
        "name": "fN",
        "type": "integer",
        "mandatory": true,
        "editable": false,
        "description": {
```

```
            "en": "Nominal grid frequency [Hz]",
            "de": "Nenn-Netzfrequenz [Hz]"
        },
        "default_value": 50
    }
],
```

Defining several "User Defined Configurations" parameters

If you define several parameters, you must separate each parameter definition by a comma after the closing curvy brackets.

```
"user_defined_configuration": [
    {
        "name": "fN",
        "type": "integer",
        "mandatory": true,
        "editable": false,
        "description": {
            "en": "Nominal grid frequency [Hz]",
            "de": "Nenn-Netzfrequenz [Hz]"
        },
        "default_value": 50
    },
    {
        "name": "UN",
        "type": "float",
        "mandatory": true,
        "editable": false,
        "description": {
            "en": "Rated power unit line supply voltage [V]",
            "de": "Nenn-Netzspannung [V]"
        },
        "default_value": 400.0
    }
],
```

11.4.10 Element "continuous_input_configuration"

Element "continuous_input_configuration" contains the configuration for the drive input parameters. These are continuously read when the "Use Case" is started. The values of "Continuous Input Configuration" parameters are updated if the model is run corresponding to the frequency specified in the configuration and in the "Drive System Framework".
Up to 10 "Continuous Input Configuration" parameters can be defined for a user-defined model.

Every "Continuous Input Configuration" parameter has the following elements:

- id: Parameters that the drive should read.
  - Format: #ParameterId#[#ParameterIndex#]
  - Examples: 7845[0], 207[0]

  Note

  Enter 0 for the parameter index if the drive parameter is not of type Array.

- type: Parameter value type (supported values: "string", "integer", "float")
- key: Key value to access parameters in the model.
- description: Parameter description. You can enter the description for English (en) and German (de).
- frequency: Specifies the frequency for reading the parameter.

Note

The value specified under "frequency" should match the frequency specified in "Drive System Framework". If the frequencies differ from those in the selected "Data Sources", then the model cannot be used in the "Use Case".

Defining an individual "Continuous Input Configuration" parameter

```
"continuous_input_configuration": [
    {
        "id": "63[1]",
        "key": "n_mot_value",
        "type": "float",
        "description": {
            "en": "Speed actual value smoothed [1/min]",
            "de": "Drehzahlistwert geglättet [1/min]"
        },
        "frequency": 10
    }
],
```

Defining several "Continuous Input Configuration" parameters

If you define several parameters, you must separate each parameter definition by a comma after the closing curvy brackets.

```
"continuous_input_configuration": [
    {
        "id": "63[1]",
        "key": "n_mot_value",
        "type": "float",
        "description": {
            "en": "Speed actual value smoothed [1/min]",
            "de": "Drehzahlistwert geglättet [1/min]"
        },
        "frequency": 10
    },
    {
        "id": "24[0]",
        "key": "fconv_out_value",
        "type": "float",
        "description": {
            "en": "Output frequency smoothed [Hz]",
            "de": "Ausgangsfrequenz geglättet [Hz]"
        },
        "frequency": 10
    }
],
```

11.4.11 Element "output_configuration"

Element "`output_configuration`" contains the configuration for KPI parameters, which are calculated by "Customizable AI Model".

Up to 15 "Output Configuration" parameters can be defined for a user-defined model.

Every "Output Configuration" parameter has the following elements:

- `name`: The name is used to access parameters in element "ui_config" (Page 161).
- `description`: Parameter description. You can enter the description for English (`en`) and German (`de`).

- `type`: Parameter value type (supported values: "`integer`", "`float`")
- `aggregation`: Specifies the aggregation type for medium-term and long-term storage. This element should contain two sets of definitions. These definitions are as follows:
  - `strategy`: Key performance indicators can be stored as average or as sum of the calculated key performance indicators. Supported values "`avg`" and "`sum`".
  - `period`: Defines how long the aggregated data should be stored. For medium-term storage, enter value "7d", for long-term storage, enter value "120M".
  - `window`: Defines the length of the aggregation window. For medium-term storage, enter value "1m", for long-term storage, enter value "1h".

Defining an individual "Output Configuration" parameter

```
"output_configuration": [
    {
        "name": "Drive_Power_Consump",
        "description": {
            "en": "Power consumption of drive system [kW]",
            "de": "Leistungsaufnahme Antriebssystem [kW]"
        },
        "type": "float",
        "aggregation": [
            {
                "strategy": "avg",
                "period": "7d",
                "window": "1m"
            },
            {
                "strategy": "avg",
                "period": "120M",
                "window": "1h"
            }
        ]
    }
],
```

Defining several "Output Configuration" parameters

If you define several parameters, you must separate each parameter definition by a comma after the closing curvy brackets.

```
"output_configuration": [
    {
        "name": "Drive_Power_Consump",
        "description": {
            "en": "Power consumption of drive system [kW]",
            "de": "Leistungsaufnahme Antriebssystem [kW]"
        },
        "type": "float",
        "aggregation": [
            {
                "strategy": "avg",
                "period": "7d",
                "window": "1m"
            },
            {
                "strategy": "avg",
                "period": "120M",
                "window": "1h"
            }
        ]
    },
    {
        "name": "Sys_EnergyConsumpt",
        "description": {
            "en": "Energy consumption of drive system [kWh]",
            "de": "Energieverbrauch des Antriebssystem [kWh]"
        },
        "type": "float",
        "aggregation": [
            {
                "strategy": "sum",
                "period": "120M",
                "window": "1h"
            },
            {
```

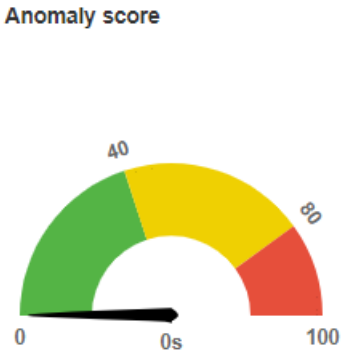
```
                    "strategy": "avg",
                    "period": "24h",
                    "window": "1m"
                }
            ]
        }
    ],
```

11.4.12 Element "ui_config"

Element "`ui_config`" contains definitions for graphically visualizing Overview Widget and the details page. These can contain several graphics, which show different key performance indicators.

There are 2 different graphic components:

- Gauge diagram component: Displays the last calculated key performance indicator in a Gauge diagram with alarm and warning levels.
  The Gauge component has the following elements:
  - `type`: Type of component. For a Gauge component, this value must be "`gauge`".
  - `title`: Displayed component name. Alphanumeric values are permissible.
  - `unit`: Specifies the unit of the selected KPI parameter.
  - `kpi_name`: Specifies which key performance indicator is displayed in the Gauge component. This value must match the name value of the displayed KPI parameter.
  - `alarm_level`: Specifies the alarm level of the selected key performance indicator. This line can be removed if there is no alarm level.
  - `warning_level`: Specifies the warning level of the selected key performance indicator. This line can be removed if there is no warning level.
  - min_value: Specifies the minimum value that should be displayed in the Gauge diagram. This line can be removed if there is no minimum value.
  - max_value: Specifies the maximum value that should be displayed in the Gauge diagram. This line can be removed if there is no maximum value.



- Line diagram component: Displays the calculated key performance indicator in a line diagram. For the Overview Widget, a time period is used for the component, and for the detail pages, a selected start and end date are used.

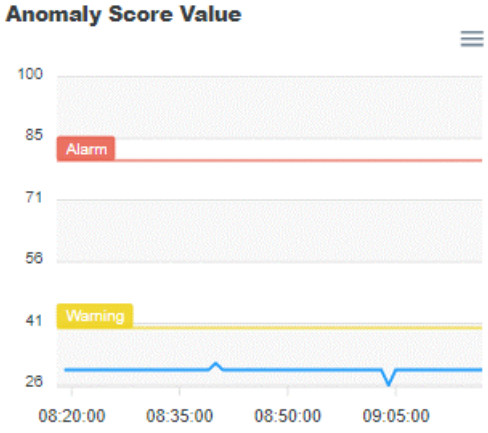
Note

You can only use `refresh_period` and `time_range` in the overview widget configurations; you can remove for detail page configurations.

The line diagram component comprises the following elements:

- `type`: Type of component. For line diagram components this value must be "`linechart`".
- `title`: Displayed component name. Alphanumeric values are permissible.
- `unit`: Specifies the unit of the selected KPI parameter.
- `kpi_name`: Specifies which key performance indicator is displayed in the line diagram component. This value must match the name value of the displayed KPI parameter.

- `refresh_period`: Specifies how frequently the line diagram is updated. The refresh_period is specified in seconds.
- `time_range`: Specifies the time period for the displayed key performance indicator. You can use a numerical value for displaying the minute format or "`today`" to display key performance indicators, which have been collected since the beginning of the day.
  Example 1: Time period is 60; the line diagram displays key performance indicators that were calculated in the last 60 minutes
  Example 2: Time period is 120; the line diagram displays key performance indicators that were calculated in the last 120 minutes
  Example 3: Time period is "today", the line diagram displays key performance indicators that were calculated from 00:00 of one day up to the current time of the same day.
- `alarm_level`: Specifies the alarm level of the selected key performance indicator. This line can be removed if there is no alarm level.
- `warning_level`: Specifies the warning level of the selected key performance indicator. This line can be removed if there is no warning level.
- `min_value`: Specifies the minimum value that should be displayed in the line diagram. This line can be removed if there is no minimum value.
- `max_value`: Specifies the maximum value that should be displayed in the line diagram. This line can be removed if there is no maximum value.



Element "`ui_config`" has two subelements to configure the Overview Widget and the details page. The Gauge components and the line diagram components can be defined in these subelements.
• Element "`widget_config`": This element can contain up to 2 graphic components. The same key performance indicator can be displayed a multiple number of times with different time periods or with a difference representation (Gauge diagram or line diagram). Two different key performance indicators can also be displayed in this element.
• Element "`details`": This element can contain up to 2 graphic components per key performance indicator. A combination of Gauge and line diagrams can be used in element "`details`". The line diagram component uses start and end data, which were selected on the details page. Elements `refresh_period` and `time_range` are therefore not used for the configuration of the details page.

Definition of "UI Config Element"

```
"ui_config": {
    "widget_config": [
        {
            "type": "gauge",
            "title": "Anomaly score",
            "unit": "t",
            "kpi_name": "anomaly_score",
            "alarm_level": 80,
            "warning_level": 40,
            "min_value": 0,
            "max_value": 100
        },
        {
            "type": "lineChart",
            "title": "Anomaly Score Value",
            "unit": "k",
            "kpi_name": "anomaly_score",
            "refresh_period": 120,
            "time_range": 60,
            "alarm_level": 80,
            "warning_level": 40,
            "min_value": 1,
            "max_value": 100
        }
    ],
    "details": [
        {
            "type": "gauge",
            "title": "Anomaly score",
            "unit": "m",
            "kpi_name": "anomaly_score",
            "warning_level": 40,
            "min_value": 0,
            "max_value": 100
```

```
            },
            {
                "type": "lineChart",
                "title": "Anomaly Score",
                "unit": "s",
                "kpi_name": "anomaly_score",
                "alarm_level": 80,
                "warning_level": 40,
                "min_value": 0,
                "max_value": 100
            }
        ]
},
```

11.4.13 Element "aggregation_method"

Element "`aggregation_method`" specifies how KPI parameters are aggregated.

For "Customizable AI Model" this value must be "time_based".

Example:

```
"aggregation_method": "time_based",
```

11.4.14 Element "execution_configuration"

Using element "`execution_configuration`", you define how data are captured before the data are sent to the model for calculations.

The "Execution Configuration" parameter contains the following elements:

- `strategy`: The strategy defines whether the data are captured based on time or on data points. For "Data Sources", type `LS`, this value must be "`time`" and for "Data Sources", type `HS`, this value must be "`window_length`".
- execution_period: The execution time period specifies how many data are captured. This value must be specified in seconds for "Data Sources", type `LS`. This value must be specified in the form of data points for "Data Sources", type `HS`.

Note

- The execution time should not be less than 2.4 seconds for "Data Sources", type `LS`.
- The execution time should not be less than 1200 data points for "Data Sources", type `HS`.

- `result_storage_period`: The result storage duration specifies how long the calculated key performance indicators are stored, without being aggregated. This value must be "`24h`".
- `configurable`: The element has no effects on this version of "Analyze MyDrives /Edge". This value must be "`false`".

Example of a "Execution Configuration" for LS Data Source

```
"execution_configuration": {
    "strategy": "time",
    "execution_period": "300s",
    "result_storage_period": "24h",
    "configurable": false
},
```

Example of a "Execution Configuration" for HS Data Source

```
"execution_configuration": {
    "strategy": "time",
    "execution_period": "300s",
    "result_storage_period": "24h",
    "configurable": false
},
```

11.4.15 Element "trigger"

Element "`trigger`" has no effects on this version of "Analyze MyDrives /Edge". This value must be empty.

Example:

```
"trigger": {}
```

11.5 Creating and uploading a Use Case for a "Customizable AI Model"

"Customizable AI Models" are handled the same as "Energy Efficiency Models" and "Mechanical Anomaly Detector Models".

Follow the steps in Section "Model Management" (Page 59) to upload and activate a "Customizable AI Model". Follow the steps in Section "Use Cases" (Page 79) to create and activate an application case for "Customizable AI Model".

11.6 Configuration files - examples

This section contains two configuration file examples for "Customizable AI Models".

The first example shows the configuration for a "Low-Speed-Adapter", which also contains the configuration for "Read Once" parameters.

The second example shows the configuration for a "High-Speed-Adapter".

11.6.1 Example: Configuration file with "Low-Speed-Adapter" as "Data Source"

The following example shows a configuration file with a "Low-Speed-Adapter" as "Data Source".

Example

```
{
    "name": "Custom LS Model",
    "version": 1.0,
    "family": "Custom LS Model",
    "type": "Generic",
    "model": "custom",
    "datasource": "LS",
    "read_once_input_configuration": [
        {
            "id": "207[0]",
            "type": "float",
            "description": {
                "en": "Rated power unit current [A]",
                "de": "Leistungsteil Bemessungsstrom [A]"
            },
            "key": "IN",
            "value": 8
        }
    ],
    "user_defined_configuration": [
        {
            "name": "sample_integer_parameter",
            "type": "integer",
            "mandatory": true,
```

```
            "editable": true,
            "description": {
                "en": "Sample Integer Parameter",
                "de": "Probe Ganzzahl Parameter"
            },
            "default_value": 50
        }
    ],
    "continuous_input_configuration": [
        {
            "id": "63[1]",
            "key": "n_mot_value",
            "type": "float",
            "description": {
                "en": "Speed actual value smoothed [1/min]",
                "de": "Drehzahlistwert geglättet [1/min]"
            },
            "frequency": 10
        }
    ],
    "output_configuration": [
        {
            "name": "sample_output_KPI",
            "description": {
                "en": "Sample Output KPI",
                "de": "Probe Ausgang KPI"
            },
            "type": "float",
            "aggregation": [
                {
                    "strategy": "avg",
                    "period": "7d",
                    "window": "1m"
                },
                {
```

```
                    "strategy": "avg",
                    "period": "60M",
                    "window": "1h"
                }
            ]
        }
    ],
    ui_config": {
        "widget_config": [
            {
                "type": "gauge",
                "title": "Sample Output KPI Gauge",
                "unit": "s",
                "kpi_name": "sample_output_KPI",
                "alarm_level": 80,
                "warning_level": 40,
                "min_value": 0,
                "max_value": 100
            },
            {
                "type": "lineChart",
                "title": "Sample Output KPI LineChart",
                "unit": "s",
                "kpi_name": "sample_output_KPI",
                "refresh_period": 60,
                "time_range": "today"
            }
        ],
        "details": [
            {
                "type": "gauge",
                "title": "Sample Output KPI Gauge",
                "unit": "s",
                "kpi_name": "sample_output_KPI",
                "alarm_level": 80,
```

```
                    "warning_level": 40,
                    "min_value": 0,
                    "max_value": 100
                },
                {
                    "type": "lineChart",
                    "title": "Sample Output KPI LineChart",
                    "unit": "s",
                    "kpi_name": "sample_output_KPI",
                    "refresh_period": 60,
                    "time_range": "today"
                }
            ]
        },
        "aggregation_method": "time_based",
        "execution_configuration": {
            "strategy": "time",
            "execution_period": "300s",
            "result_storage_period": "24h",
            "configurable": false
        },
        "trigger": {}
    }
```

11.6.2 Example: Configuration file with "High-Speed-Adapter" as "Data Source"

The following example shows a configuration file with a "High-Speed-Adapter" as "Data Source".

Example

```
    {
        "name": "Custom HS Model",
        "version": 1.0,
        "family": "Custom HS Model",
        "type": "Generic",
        "model": "custom",
```

```
"datasource": "HS",
"read_once_input_configuration": [],
"user_defined_configuration": [
    {
        "name": "sample_integer_parameter",
        "type": "integer",
        "mandatory": true,
        "editable": false,
        "description": {
            "en": "Sample Integer Parameter",
            "de": "BeispielParameter1"
        },
        "default_value": 90
    }
],
"continuous_input_configuration": [
    {
        "id": "63[0]",
        "key": "speed",
        "type": "float",
        "description": {
            "en": "Actual Speed Smoothed"
        },
        "frequency": 500
    }
],
"output_configuration": [
    {
        "name": "sample_output_KPI",
        "description": {
            "en": "Sample Output KPI",
            "de": "Probe Ausgang KPI"
        },
        "type": "float",
        "aggregation": [
```

```
                {
                    "strategy": "avg",
                    "period": "7d",
                    "window": "1m"
                },
                {
                    "strategy": "avg",
                    "period": "60M",
                    "window": "1h"
                }
            ]
        }
    ],
    "ui_config": {
        "widget_config": [
            {
                "type": "gauge",
                "title": "Sample Output KPI Gauge",
                "unit": "s",
                "kpi_name": "sample_output_KPI",
                "alarm_level": 80,
                "warning_level": 40,
                "min_value": 0,
                "max_value": 100
            },
            {
                "type": "lineChart",
                "title": "Sample Output KPI LineChart",
                "unit": "s",
                "kpi_name": "sample_output_KPI",
                "refresh_period": 60,
                "time_range": 60,
                "alarm_level": 80,
                "warning_level": 40,
                "min_value": 0,
```

```
                "max_value": 100
            }
        ],
        "details": [
            {
                "type": "gauge",
                "title": "Sample Output KPI Gauge",
                "unit": "s",
                "kpi_name": "sample_output_KPI",
                "alarm_level": 80,
                "warning_level": 40,
                "min_value": 0,
                "max_value": 100
            },
            {
                "type": "lineChart",
                "title": "Sample Output KPI LineChart",
                "unit": "s",
                "kpi_name": "sample_output_KPI",
                "refresh_period": 60,
                "time_range": 60,
                "alarm_level": 80,
                "warning_level": 40,
                "min_value": 0,
                "max_value": 100
            }
        ]
    },
    "aggregation_method": "time_based",
    "execution_configuration": {
        "strategy": "window_length",
        "execution_period": "1200",
        "result_storage_period": "24h",
        "configurable": true
    },
```

```
    "trigger": {
    }
}
```