Application Note

FESTO

Festo AX Vacuum Insights



This application note describes how to set up AX Vacuum Insights from prerequisites and data connectivity to asset onboarding and monitoring of vacuum ejectors.

GSDA-ATP-*-VAC

Title ..... Festo AX Vacuum Insights
Version 1.10
Document no. 100859
Original ..... en
Author ..... Festo

Last saved 17.11.2026

General information:

This document is intended for qualified, trained and instructed professionals. The data provided in this document are no guaranteed specifications, in particular with regard to functionality, condition or quality in the legal sense. The information in this document is intended only as simple indications for the implementation of a specific, hypothetical application, and in no way as a substitute for the operating instructions of the respective manufacturers or the design and testing of the respective application by the user. The respective operating instructions for Festo products can be found under www.festo.com. The user of this document must ensure that each function described herein works properly in his application. The user remains solely responsible for his or her own use of this document, even by studying this document and using the information mentioned therein. This also applies to any software (in source code and/or object code) that is made available to the user as an appendix to this document.

Rights of use of the software:

If software is made available to the user as an appendix to this document, the user is granted a simple and unlimited right of use hereto. This right also includes the processing and distribution of the software to third parties in edited or unedited form. The User is not permitted to use the name "Festo" to endorse or promote products derived from the Software without express prior written permission.

Software is provided to the user "as is". Festo does not assume any warranty or guarantee with regard to software. Festo's liability for damages of any kind arising from the use of the software is limited to intent and gross negligence.

Legal Notices:

©Festo SE & Co. KG, all rights reserved. A change in content and form is only permitted with the express written consent of Festo SE & Co. KG. Festo grants the user the right to reproduce this document in a form that remains unchanged in terms of content and form and to pass it on to third parties.



Need **support** with your **application**?
Get in touch with your local **Technical Support:**

www.festo.com/contact

© (Festo SE & CO. KG, D-73726 Esslingen, 2026)
Internet: http://www.festo.com

Table of contents

1 Introduction 5

2 Prerequisites and System Requirements 7

2.1 Automation equipment and required input data samples 7

2.2 Characteristics of the raw data samples 12

2.3 Required minimum data sampling interval 14

2.4 AX Industrial AI Core environment 14

2.4.1 Core version 14
2.4.2 Network setup and data connectivity solutions 14
2.4.3 License 15

2.5 Security and Intended Use 15

3 Installation and Commissioning 16

3.1 Import the AX Asset Type Package for Vacuum 16

3.2 Data provisioning via MQTT 16

3.2.1 Accessing the MQTT broker included in the AX Core 17
3.2.2 Example data connectivity solution using Node-Red 18

3.3 Onboarding of assets and initial training 19

4 Model 21

4.1 Model structure and functionality 21

4.2 Data preprocessing model 21

4.3 Anomaly detection model 21

4.3.1 Sub-Model Continuous Vacuum Pressure Value 23
4.3.2 Sub-Model Evac Time 23
4.3.3 Sub-Model Evac Time Ratio 23
4.3.4 Sub-Model Evac Time Recursive 24
4.3.5 Sub-Model Evac Time Variance 25
4.3.6 Sub-Model Festo VABF 25
4.3.7 Sub-Model General Features 26
4.3.8 Sub-Model Pressure Flow 26
4.3.9 Sub-Model Vacuum Reached 26
4.3.10 Decision fusion approach 26

4.4 Classification model 27

4.5 Model training and model update / retraining 28

4.5.1 Model retraining with the existing parameter configuration 30
4.5.2 Model retraining with a fine-tuned parameter configuration 31

5 When the application is running 34

5.1 Vacuum Asset Type Dashboard Walkthrough 34

5.1.1 Overview tab 34

5.1.2 Analytics tab 35
5.1.3 Live Data tab 36
5.1.4 Historic Data tab 38
5.1.5 System Info tab 39
5.1.6 Historic Asset Data View 39
5.1.7 Event Statistics View 40
5.2 Result Interpretation 41
6 Frequently Asked Questions 43

1 Introduction

The **Festo AX Vacuum Insights - Asset Type for Vacuum** provides continuous monitoring of vacuum processes, such as vacuum grippers and vacuum-based holding or fixture applications. It is an AX Asset Type Package used with the AX Industrial AI Core. The package contains a standardized analytics model chain and a dashboard for condition monitoring of vacuum applications.

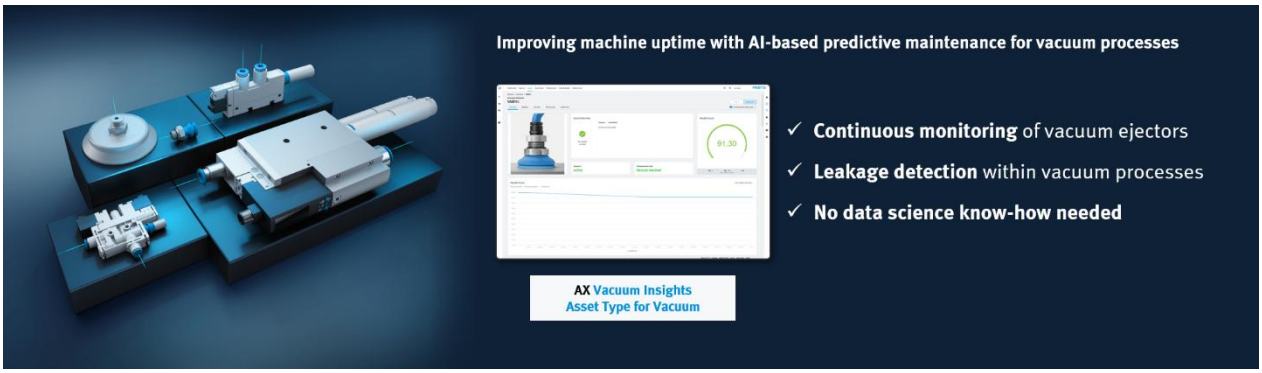


Figure 1: AX Vacuum Insights - Asset Type for Vacuum.

The vacuum asset type requires vacuum applications to provide input data in a standardized format for monitoring. This standardized data format ensures that the vacuum asset type can be applied to a wide range of vacuum-based applications in industrial automation. It supports different types of vacuum generators and vacuum ejectors as well as different vacuum pressure sensor setups. Also, the vacuum asset type can be applied for different types, sizes and configurations of suction cups.

The model chain includes a data preprocessing step, an anomaly detection model and a fault classification model. Based on the provided set of input data samples, the vacuum asset type automatically configures the data preprocessing steps, the anomaly detection model, and the classification model.
The asset type's analytical capabilities therefore depend on the provided set of input data samples, rather than on a specific vendor or vacuum generator type.

Besides this, the asset type includes an auto-onboarding configuration and asset type dashboard. The auto-onboarding configuration is used for automatically boarding assets of real-world vacuum application instances in Festo AX based on the provided input data. The asset type dashboard for vacuum summarizes the analysis results and visualizes the raw input data samples as well as the calculated features based on the vacuum process cycles.

Highlights

- Standardized app for continuous monitoring of vacuum applications.
- Standardized input data format to support a wide variety of vacuum generators and vacuum ejectors.
- Automated boarding of vacuum assets in Festo AX Industrial AI Core.
- Automated visualization of data and information-rich features from the underlying vacuum process (cf. Figure 2).
- Automated creation and training of the AX Vacuum Insights model chain.

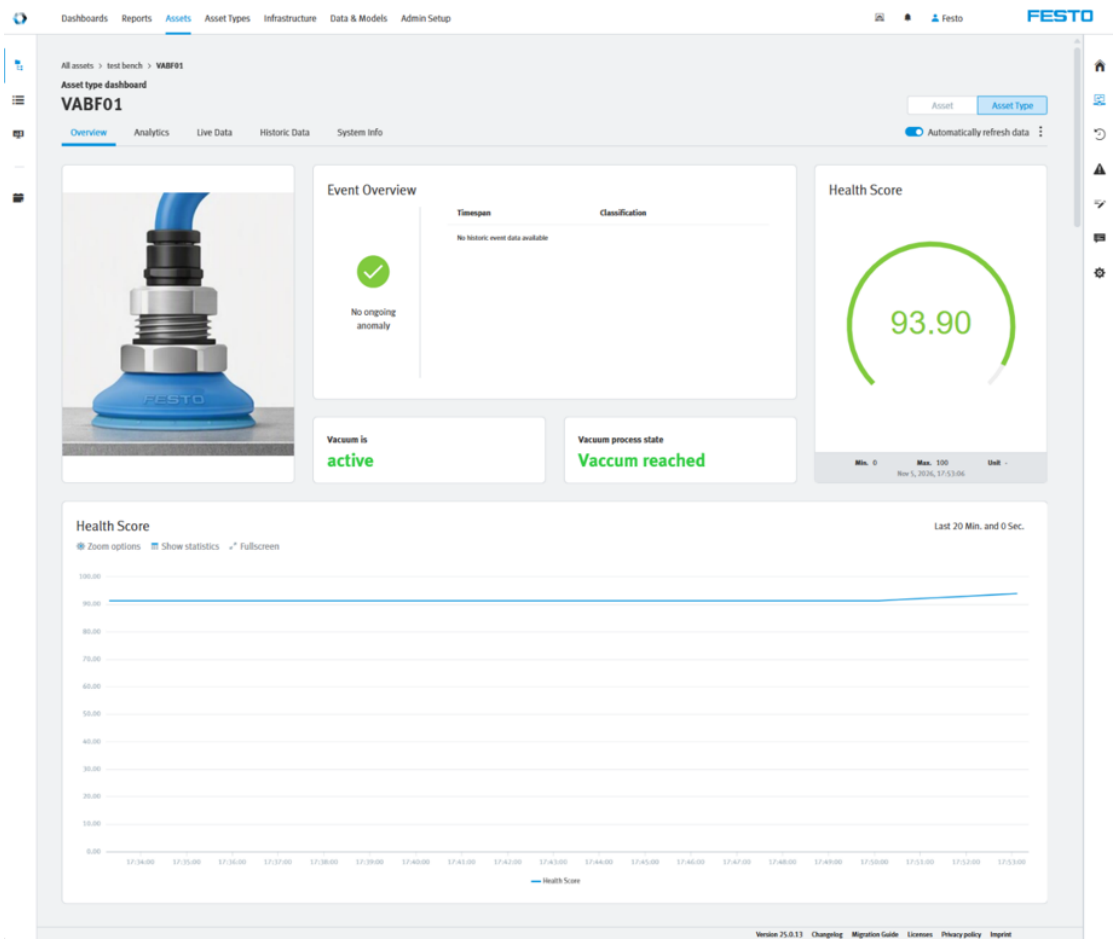


Figure 2: Overview dashboard view of the AX Asset Type for Vacuum.

Structure of the application note

This application note is structured as follows: Section 2 includes a summary of the prerequisites and system requirements for AX Vacuum Insights. Section 3 details the installation and commissioning process. In Section 4, the model chain is explained. This section also contains an explanation of the training and update process of the model chain. Section 5 presents the structure of the dashboard and outlines how the prediction results must be interpreted. Section 6 summarizes frequently asked questions.

Note

This application note covers specific information for the Festo AX Asset Type for Vacuum. It does not provide an overall overview of the AX Industrial AI Core which is the foundation for AX Asset Type applications. To get an overview of the general concepts of this foundation for AX Asset Types, please refer to the respective **application note of the Festo AX AI Core (GSDA-AIC)** that can be downloaded on www.festo.com.

2 Prerequisites and System Requirements

In this section, the prerequisites and system requirements for the AX Industrial AI Core, the AX Asset Type for Vacuum and the supported automation equipment are summarized.

2.1 Automation equipment and required input data samples

The vacuum asset type requires vacuum applications to provide input data in a standardized format. This standardized data format ensures that the vacuum asset type can be applied on a multitude of vacuum-based applications in industrial automation. The supported vacuum-based applications may include

- different types of vacuum generators and vacuum ejectors,
- different vacuum pressure sensor setups,
- different types, sizes and configurations of suction cups.

The standardized input data format comprises up to 20 variables plus the timestamp *ts*. Depending on the vacuum application and technical setup used, only a sub-set of these 20 variables plus the timestamp *ts* have to be provided to the AX Asset Type for Vacuum (cf. Table 1). The standardized input data format reflects an MQTT message payload in the JSON format. This payload has to be sent via MQTT using a specific MQTT topic structure to an MQTT broker to which the AX Industrial AI Core is connected.

The structure of the expected MQTT message including the standardized data format with all 20 variables in its payload is as follows (type hints given in *//[datatype]* notation):

```
{
  "topic": "vacuuminsights/.../myVacuumGenerator",
  "payload":{
    "valveSuction":1,                        // [binary]
    "valveBlowing":0,                        // [binary]
    "valveEjector": 0,                       // [binary]
    "vacuumPressureSwitch": 0,               // [binary]
    "vacuumPressureValue": 0.0,              // [float]
    "pressureSensor": 0.0,                   // [float]
    "flowSensor": 0.0,                       // [float]
    "diagnosisEvacuationTimeExceeded": 0.0,      // [float]
    "diagnosisEvacuationTimeExceededDouble": 0.0,   // [float]
    "diagnosisVentilationTimeExceededDouble": 0.0, // [float]
    "diagnosisVentilationTimeExceeded": 0.0,     // [float]
    "diagnosisParameter": 0.0,               // [float]
    "diagnosisControlParameter": 0.0,        // [float]
    "diagnosisProcessQuality": 0.0,          // [float]
    "diagnosisVacuumvalueA1": 0.0,           // [float]
    "openLoad_shortCircuit": 0.0,            // [float]
    "outputA": 0.0,                          // [float]
    "outputB": 0.0,                          // [float]
    "processQuality": 0.0,                   // [float]
    "errorMsg": 0.0,                         // [float]
    "ts": 1793948483769                      // Unix epoch time in milliseconds [int]
  }
}
```

As mentioned above, the AX Asset Type for Vacuum expects only a subset of the available variables depending on the underlying vacuum application. The subsets of expected input variables for each supported vacuum system and vacuum generator/ejector are summarized in the following Table 1.

Use exact, unambiguous feature names and avoid duplicate names as defined in Table 1. Do not introduce additional variables to the standardized input data format. Additionally, unknown variables will not be used by the AX Asset Type for Vacuum.

Table 1: Supported types of vacuum systems, vacuum generators/ejectors and required sets of input variables.

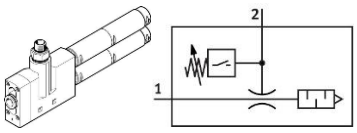
<table>
<tr><th>Setup configuration of the vacuum system</th><th>Required payload data provided via MQTT</th></tr>
<tr><td>

Simple vacuum generator (e.g., Laval/Venturi Nozzles) with a **vacuum pressure switch** or a **vacuum pressure sensor** providing continuous pressure equivalent values.

Expected input data variables:

- signal of the valve on the used valve terminal to pressurize the vacuum ejector → variable **"valveSuction".**
- signal of the valve on the used valve terminal to pressurizing the blow off channel of the vacuum ejector → variable **"valveBlowing".**
- Binary signal of a vacuum pressure switch indicating whether the target vacuum pressure level is reached or not → variable **"vacuumPressureSwitch".**
- Floating-point signal of a vacuum pressure sensor measuring the actual vacuum pressure or an equivalent value (continuous values) → variable **"vacuumPressureValue".**

Exemplary generator: Festo VN



</td><td>

Minimal required set of variables:

```
{
 "valveSuction": 1,       // [binary],
 "vacuumPressureSwitch": 0, // [binary],
 "ts": 1779887644123      // [int]
}
```

OR

```
{
 "valveSuction": 1,       // [binary],
 "vacuumPressureValue": 0.0, // [float],
 "ts": 1779887644123      // [int]
}
```

Optimal set of variables:

```
{
 "valveSuction": 1,       // [binary],
 "valveBlowing": 0,       // [binary],
 "vacuumPressureSwitch": 0, // [binary],
 "ts": 1779887644123      // [int]
}
```

OR

```
{
 "valveSuction": 1,       // [binary],
 "valveBlowing": 0,       // [binary],
 "vacuumPressureValue": 1.0, // [float],
 "ts": 1779887644123      // [int]
}
```

</td></tr>
<tr><td>

Vacuum generator/ejector with an internal ejector valve and vacuum pressure switch

Expected input data variables:

- Signal of the valve on the used valve terminal to pressurizing the vacuum ejector → variable **"valveSuction".**
- Signal of the valve on the used valve terminal to pressurizing the blow off channel of the vacuum ejector → variable **"valveBlowing".**
- Binary signal of a vacuum pressure switch indicating whether the target vacuum pressure

</td><td>

Required set of variables:

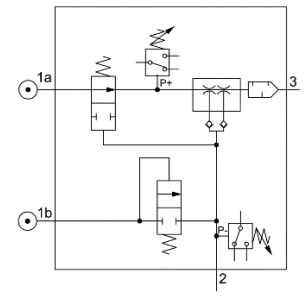
```
{
 "valveSuction": 0,       // [binary],
 "valveBlowing": 0,       // [binary],
 "valveEjector": 0,       // [binary],
 "vacuumPressureSwitch": 0, // [binary],
 "ts": 1779887644123      // [int]
}
```

</td></tr>
</table>

level is reached or not → variable **"vacuumPressureSwitch".**

- Binary signal of the switching behavior of the internal ejector valve indicating whether the ejector valve pressurizes the vacuum ejector (high, 1) or blocking the air flow for air-saving (low, 0). This valve ejector signal can also be derived via a second pressure switch reflecting the switching behavior (0/1)
  → variable **"valveEjector".**

Exemplary vacuum generator:



Festo VABF

Expected input data variables:

- Signal of the valve on the used valve terminal to pressurizing the vacuum ejector → variable **"valveSuction".**
- Signal of the valve on the used valve terminal to pressurizing the blow off channel of the vacuum ejector → variable **"valveBlowing".**
- Floating point signal of a vacuum pressure sensor measuring the actual vacuum pressure or an equivalent value (continuous values) → variable **"vacuumPressureValue".**
- Forward the VABF output signal *diagnosisEvacuationTimeExceeded* to the AX Asset Type for Vacuum
  → variable **"diagnosisEvacuationTimeExceeded".**
- Forward the VABF output signal *diagnosisEvacuationTimeExceededDouble* to the AX Asset Type for Vacuum
  → variable **"diagnosisEvacuationTimeExceeded-Double".**
- Forward the VABF output signal *diagnosisVentilationTimeExceeded* to the AX Asset Type for Vacuum
  → variable **"diagnosisVentilationTimeExceeded".**
- Forward the VABF output signal *diagnosisVentilationTimeExceededDouble* to the AX Asset Type for Vacuum
  → variable **"diagnosisVentilationTimeExceeded-Double".**

Minimal required set of variables:

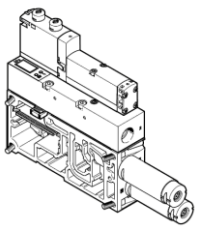
```
{
  "valveSuction":1,                 // [binary]
  "vacuumPressureValue": 0.0,       // [float]
  "ts":1793948483769                // [int]
}
```

Optimal set of variables:

```
{
  "valveSuction":1,                 // [binary]
  "valveBlowing":0,                 // [binary]
  "vacuumPressureValue": 0.0,       // [float]
  "diag...Evac...TimeExceeded": 0.0,     // [float]
  "diag...Evac...TimeExceededDouble": 0.0, // [float]
  "diag...Venti...TimeExceededDouble": 0.0,// [float]
  "diag...Venti...TimeExceeded": 0.0,    // [float]
  "diagnosisParameter": 0.0,        // [float]
  "diagnosisControlParameter": 0.0,  // [float]
  "diagnosisProcessQuality": 0.0,    // [float]
  "diagnosisVacuumvalueA1": 0.0,     // [float]
  "openLoad_shortCircuit": 0.0,      // [float]
  "outputA": 0.0,                   // [float]
  "outputB": 0.0,                   // [float]
  "processQuality": 0.0,            // [float]
  "errorMsg": 0.0,                  // [float]
  "ts":1793948483769                // [int]
}
```

<table>
<tr>
<td>

- Forward the VABF output signal *diagnosis-Parameter* to the AX Asset Type for Vacuum → variable **"diagnosisParameter".**
- Forward the VABF output signal *diagnosisControlParameter* to the AX Asset Type for Vacuum → variable **"diagnosisControlParameter".**
- Forward the VABF output signal *diagnosisProcessQuality* to the AX Asset Type for Vacuum → variable **"diagnosisProcessQuality".**
- Forward the VABF output signal *diagnosisVacuumvalueA1* to the AX Asset Type for Vacuum → variable **"diagnosisVacuumvalueA1".**
- Forward the VABF output signal *openLoad_shortCircuit* to the AX Asset Type for Vacuum → variable **"openLoad_shortCircuit".**
- Forward the VABF output signal *outputA* to the AX Asset Type for Vacuum → variable **"outputA".**
- Forward the VABF output signal *outputB* to the AX Asset Type for Vacuum → variable **"outputB".**
- Forward the VABF output signal *processQuality* to the AX Asset Type for Vacuum → variable **"processQuality".**
- Forward the VABF output signal errorMsg to the AX Asset Type for Vacuum → variable **"errorMsg".**



The details of the available data from the VABF and possible data connectivity solutions, e.g., via IO-Link, are described in the application notes of the VABF.

</td>
<td></td>
</tr>
<tr>
<td>

Festo OVEM

The OVEM can be used via two commissioning approaches. (1) Use via IO-Link mode and communication interface, (2) SIO mode.

Expected input data variables:

- Signal of the valve on the used valve terminal to pressurizing the vacuum ejector → variable **"valveSuction".**

</td>
<td>

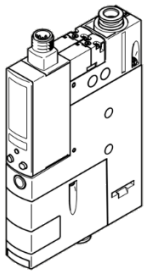
Minimal required set of variables:

```
{
  "valveSuction": 1,       // [binary]
  "vacuumPressureValue": 0,  // [float]
  "ts": 1779887644123      //[int]
}
```

</td>
</tr>
</table>

<table>
<tr>
<td>

- Signal of the valve on the used valve terminal to pressurizing the blow off channel of the vacuum ejector → variable **"valveBlowing".**
- Binary signal of a vacuum pressure switch indicating whether the target vacuum pressure level is reached or not → variable **"vacuumPressureSwitch".** The *outputA* signal of the OVEM vacuum generator can be used as this variable.
- Floating point signal of a vacuum pressure sensor measuring the actual vacuum pressure or an equivalent value (continuous values) → variable **"vacuumPressureValue".**
- Forward the OVEM output signal *outputA* to the AX Asset Type for Vacuum → variable **"outputA".**
- Forward the OVEM output signal *outputB* to the AX Asset Type for Vacuum → variable **"outputB".**



The details of the available data from the OVEM and possible data connectivity solutions, e.g., via IO-Link are described in the applications notes of the OVEM-...-LK.

</td>
<td>

Optimal set of variables:

```
{
  "valveSuction": 1,       // [binary]
  "valveBlowing": 0,       // [binary]
  "vacuumPressureValue": 1.0, // [float]
  "vacuumPressureSwitch": 1, // [binary]
  "outputA": 0.0,          // [float]
  "outputB": 0.0,          // [float]
  "ts": 1779887644123      // [int]
}
```

</td>
</tr>
<tr>
<td>

Additional pressure and flow sensors

The data samples from a pressure sensor and flow sensor installed in the air supply of the vacuum generator can be added to any set of variables listed in this Table. The naming scheme of the additional respective variables representing the measurements from the pressure and flow sensors is given on the right.

</td>
<td>

```
{
  ...
  "pressureSensor": 0.0, // [float]
  "flowSensor": 0.0,     // [float]
  "ts": 1779887644123    // [int]
}
```

</td>
</tr>
</table>

Table 2.1: Vacuum systems

2.2 Characteristics of the raw data samples

The sampled data from the vacuum process provided via MQTT messages must represent the vacuum process cycles that correspond to the vacuum-based gripping or fixture/holding processes. The sampled data of the vacuum cycles are cyclic time series data. The model chain of the AX Asset Type for Vacuum analyzes these vacuum cycles. That means, the preprocessing model of the model chain detects the vacuum cycle based on the streamed input data samples via MQTT. To ensure a valid cycle detection based on the input data samples, the input data samples must meet the following requirements:

- The data samples of the variables inside the MQTT payloads must be synchronously sampled and sent to the AX Industrial AI Core.
- A constant, equidistant sampling rate is recommended.
- The vacuum cycle start condition is associated with a rising edge of the *valveSuction* data samples.
- The vacuum cycle end condition is associated either with a falling edge of the *valveSuction* data samples (if the *valveBlowing* variable is not present) or with a falling edge of the *valveBlowing* data samples (if the *valveBlowing* variable is present). The presence or absence of the *valveBlowing* variable during the onboarding of the vacuum asset in the AX Industrial AI Core defines whether the cycle detection uses the falling edge of the *valveBlowing* variable or the *valveSuction* variable.
- The timestamp *ts* of each data sample, which is also sent via the MQTT payload, should be set by the data acquisition system, e.g., by a PLC, to ensure accurate measurements in time.
- The minimum sampling time must be guaranteed (cf. Section 2.3).
- Send one MQTT message including all respective input variables per each sampling interval.
- The vacuum cycle detection included in the data preprocessing model discards cycles at or above 50,000 timestamp units and cycles with a data gap at or above 30,000 timestamp units. These limits correspond to 50 seconds and 30 seconds only if the input timestamps use milliseconds.

Incomplete cycles are not processed. The analysis of the pressure and flow data samples requires at least 10 samples per evaluated cycle.

The following two figures show two exemplary vacuum cycles that are expected by the AX Asset Type for Vacuum. Figure 3 shows an exemplary vacuum cycle and its raw data samples of a vacuum generator with an internal ejector valve and a binary vacuum pressure switch. In Figure 4, exemplary raw data samples of a vacuum generator with a vacuum pressure sensor providing continuous vacuum pressure equivalent values are shown.

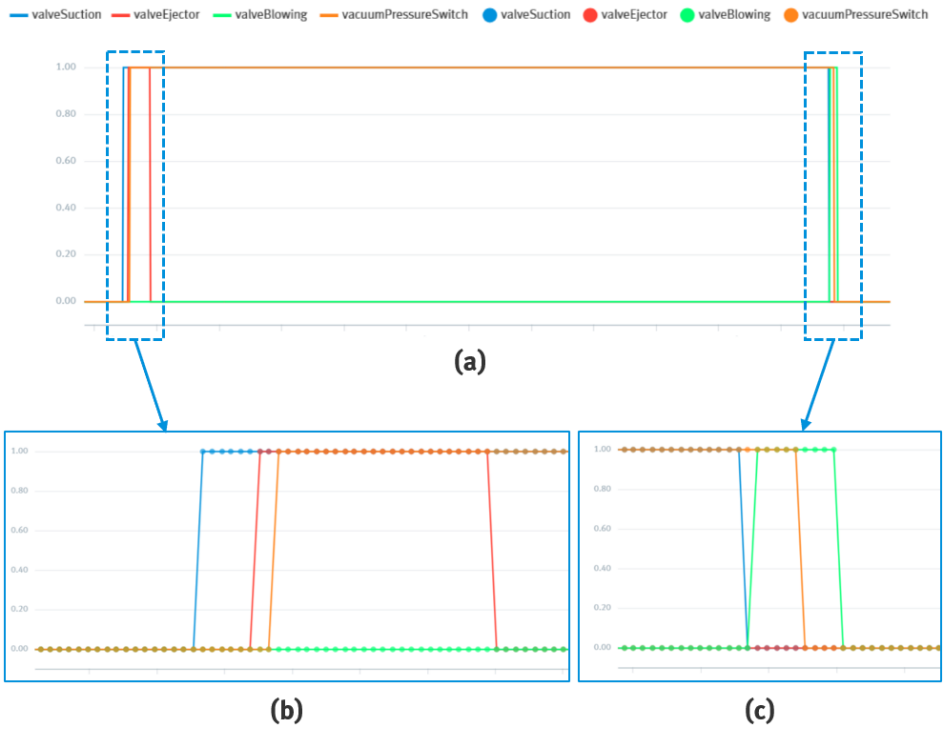


Figure 3: Exemplary raw data samples of a vacuum generator with an internal ejector valve and a binary vacuum pressure switch: (a) Signal switching behavior during one vacuum process cycle, e.g., of a vacuum-based gripping process. (b) Zoomed-in signal switching behavior at the beginning of the vacuum process cycle. (c) Zoomed-out signal switching behavior at the end of the vacuum process cycle.

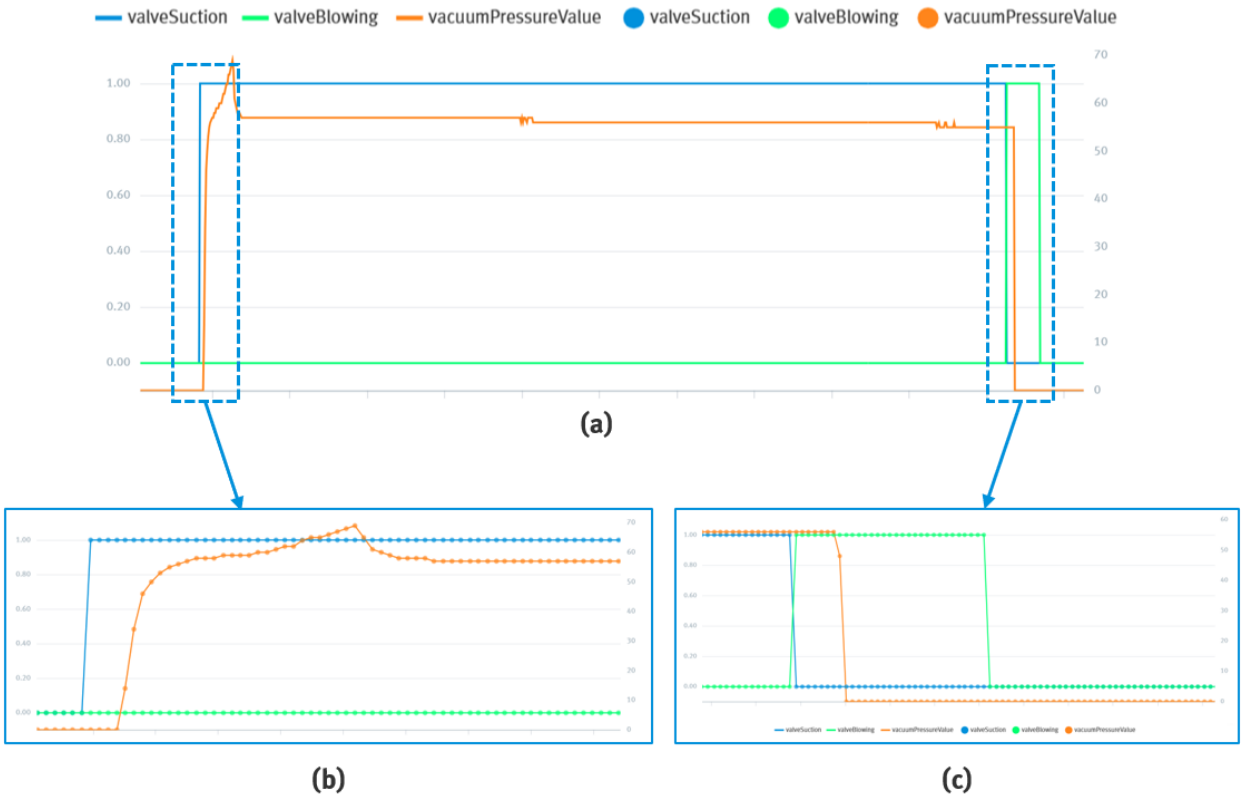


Figure 4: Exemplary raw data samples of a vacuum generator with a vacuum pressure sensor providing continuous vacuum pressure equivalent values: (a) Signal switching behavior and vacuum pressure equivalent during one vacuum process cycle, e.g., of a vacuum-based gripping process. (b) Zoomed-in signal switching behavior and vacuum pressure equivalent at the beginning of the vacuum process cycle. (c) Zoomed-out signal switching behavior and vacuum pressure equivalent at the end of the vacuum process cycle.

2.3 Required minimum data sampling interval

Since the data analytics approach is directly based on the raw data samples from the vacuum process, the AX Asset Type for Vacuum requires a high temporal resolution of the switching behavior of the binary signals and the measured continuous sensor values. The AX Asset Type for Vacuum can be correctly applied if the sampling time of the input data meets the required minimum sampling time condition:

$$evacuation\ time\ vacuum\ ejector > 2 \cdot sampling\ time\ of\ the\ raw\ data\ samples \quad .$$

To make the application of the vacuum asset type more robust, we recommend a sampling time of the input data, for which the condition

$$evacuation\ time\ vacuum\ ejector > 3 \cdot sampling\ time\ of\ the\ raw\ data\ samples \quad .$$

is at least true.

2.4 AX Industrial AI Core environment

This section summarizes the prerequisites and system requirements of the AX Industrial AI Core as well as the network setup for the data connectivity. For further details, please refer to the official application note of the Festo AX Industrial AI Core (GSDA-AIC) that can be downloaded on www.festo.com.

2.4.1 Core version

The Asset Type requires a running instance of the AX Industrial AI Core version 25.0.0 or higher.

2.4.2 Network setup and data connectivity solutions

Ethernet network access to all involved PLCs, controllers and MQTT clients must be available. The typical network setup and data connectivity solution for vacuum applications in industrial automation systems and the AX Industrial AI Core is schematically depicted in Figure 5. This deployment schematic can help to identify the vacuum equipment, acquisition/mapping layer, MQTT broker and AX Core. The user of the AX Asset Type for Vacuum must provide the required data via MQTT to the AX Industrial AI Core. Thereby, the MQTT broker included in the AX Industrial AI Core can be used for the data connectivity (blue rectangle in Figure 5). See Section 3.1 for further information for connecting to the MQTT broker of the AX Industrial AI Core. Note that the user can also use his/her own MQTT broker for the data connectivity to the AX Core. In such cases, the AX Core must be connected to the external MQTT broker of the user.

The data acquisition system must be able to publish the selected vacuum data samples to the MQTT broker used by AX. The AX platform must have network access to the services required by its approved deployment. Provide a stable network route between the acquisition system and the approved MQTT broker.

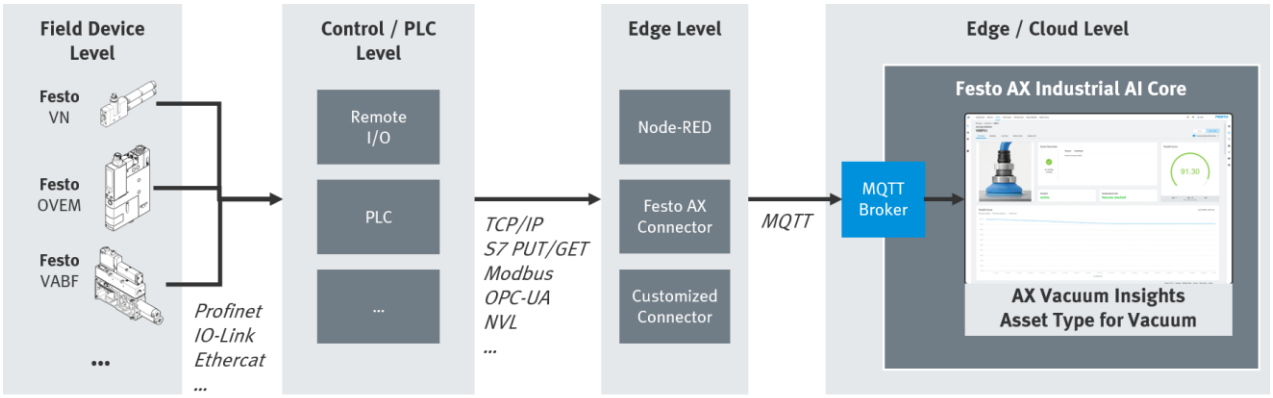


Figure 5: Overview of the typical network setup and data connectivity solutions in industrial automation systems used for the vacuum signal acquisition for the AX Industrial AI Core and the AX Asset Type for Vacuum.

2.4.3 License

Festo AX Vacuum Insights - Asset Type Vacuum consists of two parts: the Asset Type Package and the Industrial AI Core that builds the foundation for it. For both parts valid licenses are required:

- AX Asset Type for Vacuum: GSDA-ATP-*-VAC
- AX Industrial AI Core: GSDA-AIC

2.5 Security and Intended Use

Vacuum Insights is intended to support observation and investigation of vacuum-process condition. It analyzes acquired signals and presents scores, classifications and trends to authorized users. It is not described here as a safety function or a replacement for the equipment manufacturer's operating instructions, machine risk assessment or protective measures. Analysis results should be evaluated by the responsible operator or maintenance specialist in the process context. Do not automate an intervention in the machine control system solely from an anomaly score or classification presented in this application note.

Deploy the AX services within the network arrangement approved for the site. Protect MQTT, browser access and external integrations with appropriate access controls. Restrict access to administration, model configuration and acquisition settings to authorized persons. Use the authentication and encryption methods approved for the delivered broker and deployment. Do not publish registry passwords, license strings or integration credentials in screenshots, payload examples or exported documentation. Limit exposed ports and destinations to those required by the installation.

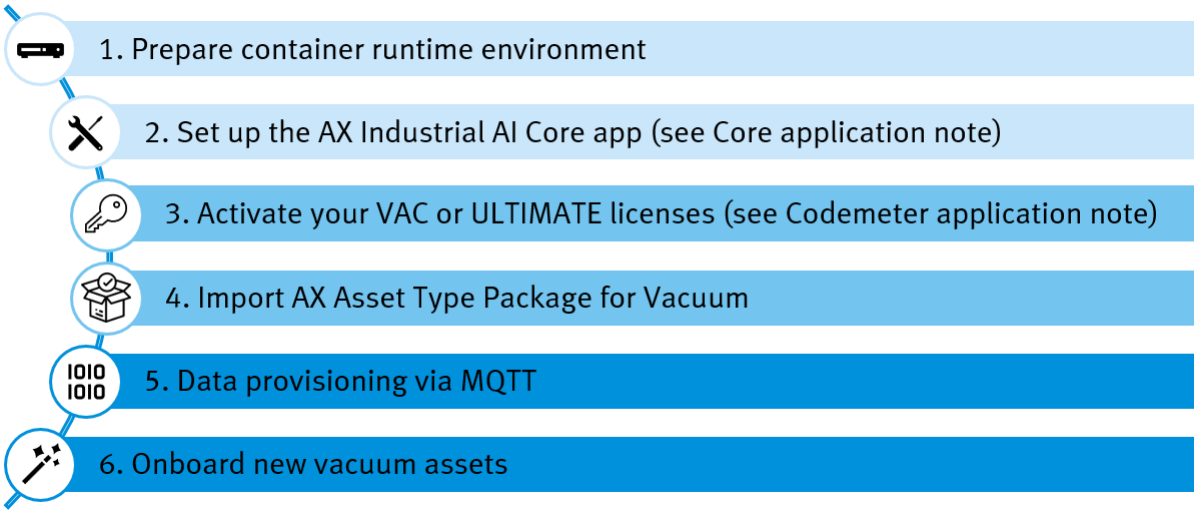


Note

Plan and deploy the data acquisition for your specific vacuum application before importing the Asset Type Package. First identify the available data from your vacuum generator or vacuum ejector, vacuum pressure sensor and additional pressure and/or flow sensors of the compressed air supply. Ensure that the data acquisition system can provide the required data samples at the required minimum sampling rate. In addition, confirm that the AX platform, license entitlement and acquisition system can support the intended number of vacuum assets.

3 Installation and Commissioning

To set up the entire AX Asset Type for Vacuum, these steps have to be taken:



3.1 Import the AX Asset Type Package for Vacuum

Before the AX Asset Type Package for Vacuum can be imported, a valid VAC license has to be activated. For details on the license activation process and the import of the asset type package in the AX Core, please refer to the application notes of the AX Industrial AI Core and Codemeter for license handling, which both can be downloaded from www.festo.com.

3.2 Data provisioning via MQTT

Import the AX Asset Type Package for Vacuum before publishing MQTT data for the automatic onboarding. Use a controlled example asset to confirm topic interpretation and input mapping before expanding the number of onboarded assets.

To onboard the vacuum assets in the AX Core, the expected input data samples have to be sent in the standardized data format as described in Section 2.1 via MQTT to the AX Core. Note, for each sampling interval of the input data samples, one MQTT message should be sent (cf. Section 2.2).

The MQTT message for the vacuum asset type has to be structured as follows:

```
{
  "topic": "vacuuminsights/.../myVacuumGenerator",
  "payload":{
    ...
    Set of variables according to Section 2.1
    ...
    "ts": 1793948483769                    // the timestamp ts is mandatory!
  }
}
```

The MQTT topic structure has to fulfil the following rules:

Base topic structure: "vacuuminsights/.../myEjector01"

Example topic strings: "vacuuminsights/lab/test_bench/myEjector"
"vacuuminsights/lab/test_bench/vacuum/myEjector01"
"vacuuminsights/lab/test_bench/vacuum/myEjector02"
"vacuuminsights/Hall/Floor/Segment/Workgroup/Station/VacuumGripper01"

Once the MQTT data, organized according to the defined MQTT topic structure, is published to the MQTT broker of the AX Core, the asset auto-onboarding process is triggered automatically. Based on the provided MQTT topic structure, the asset path and asset name of the vacuum asset boarded in the AX Core are derived. Table 2 summarizes the mappings between the topic structure, asset path and asset name.

Table 2: Configured topic and onboarding mapping.

Item	**Package configuration or illustrative value**	**Meaning**
Subscription	vacuuminsights/#	The onboarding of the AX Asset Type for Vacuum is triggered based on the **vacuuminsights** field of the MQTT topic.
Asset-topic pattern	^vacuuminsights/(.*)/(.*)$ ---------- Groups: $1 $2	Captures the asset path and final asset name used by the onboarding.
Asset path	Capture group $1 from the asset-topic pattern	Asset path supplied by the matching topic.
Asset name	Capture group $2 from the asset-topic pattern	Asset name supplied by the matching topic.
Example topic	vacuuminsights/Site/Hall/Floor/Segment/Workgroup/Station/VacuumGripper01	**Site/Hall/Floor/Segment/Workgroup/Station** becomes the asset path. **VacuumGripper01** becomes the name for boarded vacuum asset.

Table 3.1: Configured topic descriptions

3.2.1 Accessing the MQTT broker included in the AX Core

The MQTT Connection to Festo AX MQTT Broker can be established within the same network of the running container environment with the following settings:

Server: mosquitto-extern
Port: 1883
Security: none

If the data provision is to be carried out via MQTT using Node-Red, the customer must set up this running Node-Red instance themselves. Figure 6 shows an example of the connection settings in Node-Red for the MQTT broker of the AX Industrial AI Core.

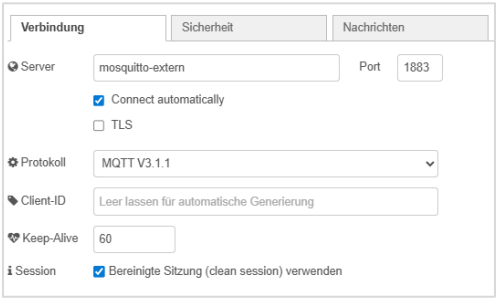


Figure 6: Connecting to the MQTT broker of the AX Core in Node-Red.

Note

Node-Red is not part of the delivered AX Industrial AI package. It must be added manually to the setup in case it is needed.

3.2.2 Example data connectivity solution using Node-Red

In the following, an exemplary data connectivity solution for the Festo VABF vacuum generator via a TCP/IP communication interface and Node-Red is presented. The data from the VABF is acquired on a PLC. The PLC sends the data via TCP/IP to a Node-Red instance running on an edge device. The data processing flow in Node-Red transforms the incoming data into the standardized data format expected by the AX Asset Type for Vacuum (cf. Figure 7). The data transformed into the standardized data format is sent via MQTT to the MQTT broker connected to the Festo AX Core. In Figure 8, an exemplary MQTT message for the Festo VABF vacuum generator is depicted.

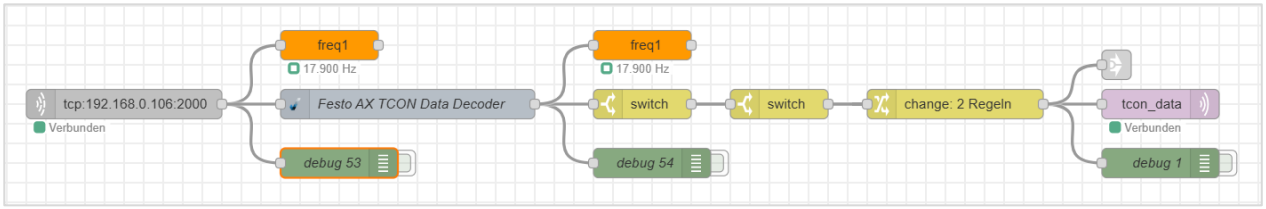


Figure 7: Example data connectivity solution for the AX Asset Type for Vacuum using Node-Red. The PLC sends the data from the Festo VABF vacuum generator via TCP/IP to a running Node-Red instance running on an edge device. A data processing flow processes the incoming data and maps the data in the standardized format to MQTT.

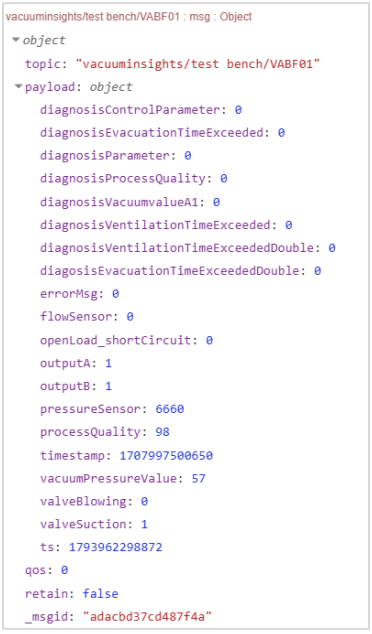


Figure 8: Example MQTT message of the data samples from the Festo VABF vacuum generator.

3.3 Onboarding of assets and initial training

If enabled, the asset auto-onboarding process is triggered automatically by new MQTT messages. The automated onboarding can be enabled and disabled in the settings of the respective asset type. If enabled, for each MQTT message sent to the AX Core with a new unique topic and valid payload, a new vacuum asset is automatically boarded in the AX Core. Details on this automated onboarding can be found in the application note of the AX Industrial AI Core.

You can navigate to already boarded vacuum assets via the asset tree as shown in Figure 9.

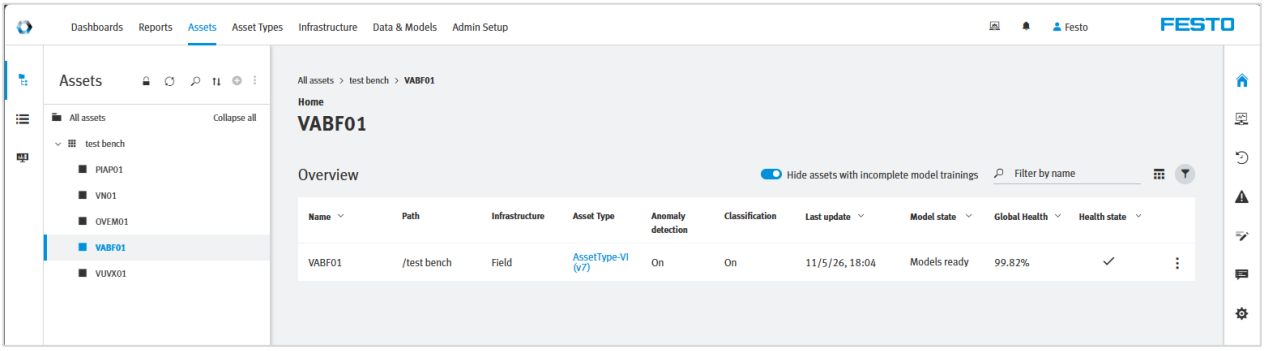


Figure 9: Navigating to assets via the asset tree.

Before the anomaly detection is active, the initial model training has to be completed. Therefore, a scheduled training is configured by default. Only the anomaly detection model requires a scheduled training after the onboarding. You can check the progress of the training phase in the asset settings of the model chain (cf. Figure 10). The AX Asset Type for Vacuum needs at least 100 data samples representing the OK/good/normal/fault-free condition of the underlying vacuum process. The data samples provided during the training phase set the “baseline” for the ongoing anomaly detection. The number of 100 training data samples corresponds to 100 vacuum cycles.

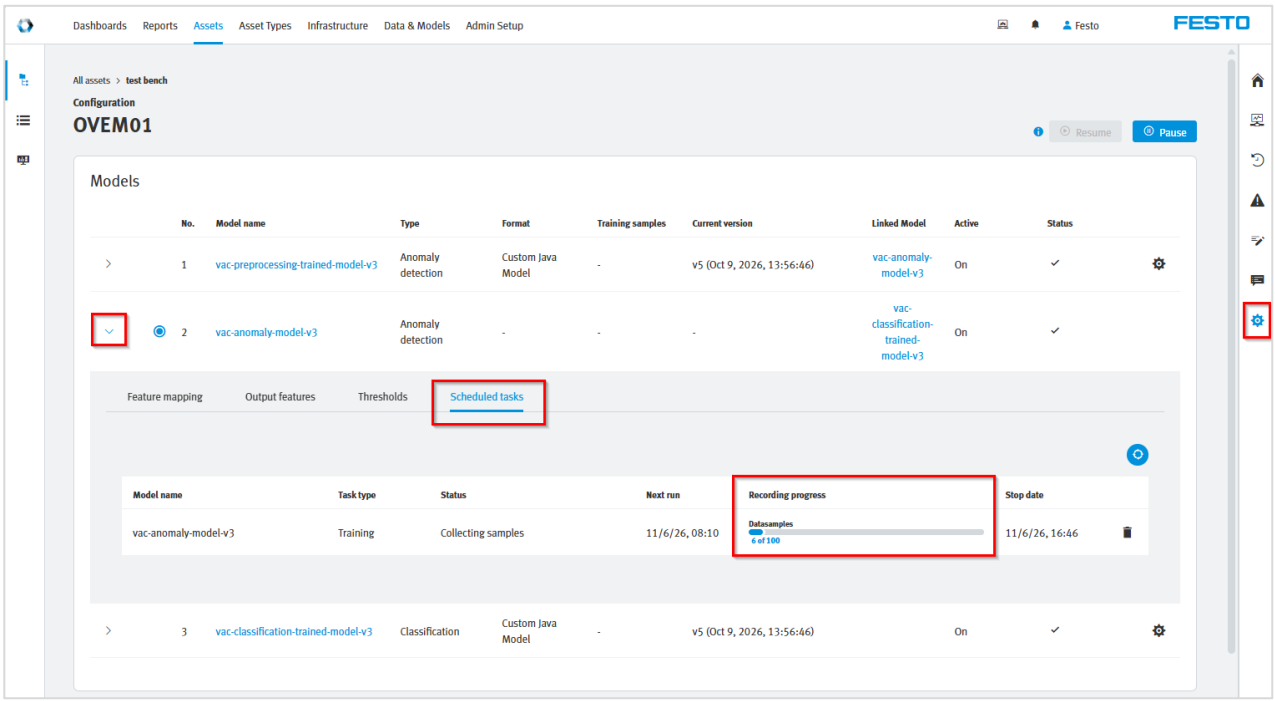


Figure 10: Initial training of a vacuum asset.

Note

The model of the AX Asset Type for Vacuum has to complete the initial training until results from the anomaly detection and fault classification can be displayed on the dashboard. It might take some time until enough training data samples are recorded for the initial training of the model.

4 Model

In this section, the data analysis approach and model chain of the Vacuum Asset Type is explained.

4.1 Model structure and functionality

To detect anomalies in the behavior of the monitored vacuum application, a model chain consisting of three models is used. Figure 11 shows the model chain of the Asset Type for Vacuum. It contains the VAC-preprocessing model, the VAC-anomaly-model and the VAC-classification-model. To support different types of vacuum generators/ejectors as well as to detect different kinds of errors and faults, the preprocessing model, anomaly detection model and classification model follow an adaptive, self-configuring approach based on the provided input data. The details of these models are described in the following subsections.

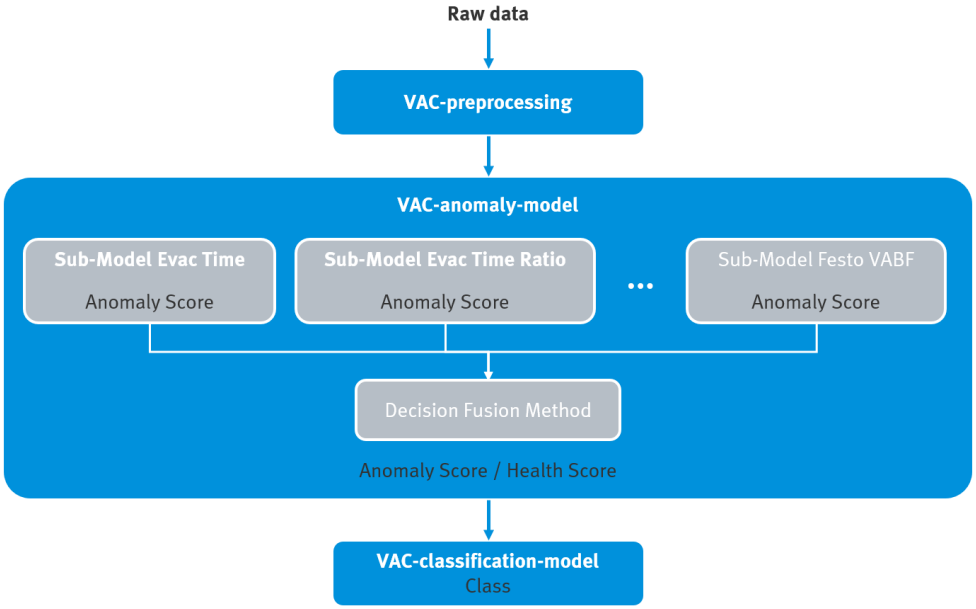


Figure 11: Model chain of the vacuum asset type.

4.2 Data preprocessing model

The preprocessing model is automatically configured at the onboarding of the vacuum asset. The configuration of its data processing steps is derived from the initially provided payload data in the first MQTT message. Based on the provided input data samples, the preprocessing model also determines the set of sub-models of the anomaly detection model that can be applied on the underlying vacuum application. This adaptive and self-configuring functionality of the preprocessing model makes it possible to support different types of vacuum generators and vacuum ejectors as well as different vacuum application setups with different vacuum pressure sensor setups and different types, sizes and configurations of suction cups.

The preprocessing model requires no training step. It detects vacuum cycles in the incoming streams of data samples from the vacuum application. After a vacuum cycle is detected, the preprocessing model computes an information-rich feature set based on the cyclic, multivariate time series data via adaptive feature extractors.

4.3 Anomaly detection model

The anomaly detection model follows an ensemble-based approach. It consists of nine sub-models. The predictions of these sub-models are combined via a decision fusion approach. The anomaly detection model predicts an overall anomaly score that represents the health state of the vacuum asset. It is computed based on the predictions of the analytical sub-models included in the anomaly detection model which can be applied on the underlying vacuum application.

The set of sub-models that can be applied on the underlying vacuum application is indicated by the model type id (model_type_id) that is also visualized on the dashboard of the AX Asset Type for Vacuum (cf. Section 5.1.5). The sub-models used by the anomaly detection model can be identified via the digit position of each sub-model in the model type id. A digit being equals to 1 indicates the selection of the respective sub-model. If the digit is equal to 0, the respective sub-model is not selected. Table 3 summarizes the coding schema of the sub-models across the digit positions of the model type id. The following example illustrates the use of the model type id more clearly: If the model type id is 100.001, the sub-models Sub-Model Festo VABF, Sub-Model Evac Time, Sub-Model Evac, Time Recursive and Sub-Model Evac Time Variance are used by the anomaly detection model.

Each of the nine sub-models included in the anomaly detection model as well as the decision fusion approach are described in the following subsections in more detail. The following subsections also include an overview of the parameters to configure the training of the sub-models and the decision fusion approach.

Table 3: Meaning of the selected digit positions in the model type id and selected sub-models for anomaly detection.

Model type id	Meaning / Analytical profile	Selected sub-models for anomaly detection
1.000.000	Data samples from a vacuum pressure sensor measuring continuous vacuum pressure values are available in the vacuum cycle detected. The preprocessing model extracts features based on these data samples for the subsequent sub-model continuous vacuum pressure value for anomaly detection.	Sub-Model Continuous Vacuum Pressure Value
0.100.000	Data samples from the Festo VABF are available in the vacuum cycle detected. The preprocessing model extracts features from these data samples that can then be analyzed from the sub-model Festo VABF for anomaly detection.	Sub-Model Festo VABF
0.010.000	Data samples from the additional pressure and flow sensors are available in the vacuum cycle detected. The preprocessing model extracts features based on these data samples. The resulting feature set can be processed by the sub-model shown in the right-hand cell.	Sub-Model Pressure Flow
0.001.000	Data samples from a vacuum pressure switch are available in the vacuum cycle detected. The preprocessing model extracts features based on these data samples. These features are used by the sub-model on the right.	Sub-Model Vacuum Reached
0.000.010	If the second digit in the model type id is set to 1, the vacuum cycle detected contains data samples from additional pressure and flow sensors and/or from the signals valveEjector and vacuumPressureSwitch. The feature set derived based on these data samples can be analyzed by the sub-model listed on the right.	Sub-Model Evac Time Ratio
0.000.001	If this digit of the model type id is set, data samples from a Festo VABF and/or data samples from the signals valveSuction, vacuumPressureSwitch and valveEjector are available in the vacuum cycle detected. The features derived from such a vacuum cycle can be processed by the three sub-models listed in the right-hand cell.	Sub-Model Evac Time Sub-Model Evac Time Recursive Sub-Model Evac Time Variance

Table 4.1: Model ID descriptions

4.3.1 Sub-Model Continuous Vacuum Pressure Value

This sub-model reacts on changes of the features computed based on the continuous-valued vacuum pressure data samples obtained using a vacuum pressure sensor. The model's influence on the final decision fusion step can be controlled with the parameter listed in Table 4.

Table 4: Parameter of the Sub-Model Continuous Vacuum Pressure Value.

Parameter and default value	Meaning	Value range	Effect of parameter change
Rater Sub-Model Continuous Vacuum Pressure Value = 1.0	Weighting parameter for the Sub-Model Continuous Vacuum Pressure Value at the decision fusion step.	[0..100]	↑: Higher weighting on the final decision making during the decision fusion step. ↓: Lower weighting on the final decision making during the decision fusion step.

Table 4.2: Sub Model descriptions

4.3.2 Sub-Model Evac Time

The Sub-Model Evac Time reacts on changes of the initial evac time computed via the valve suction signal and the vacuum pressure switch signal. The model's parameters are listed in Table 5.

Table 5: Parameters of the Sub-Model Evac Time.

Parameter and default value	Meaning	Value range	Effect of parameter change
Rater Sub-Model Evac Time = 10	Weighting parameter for the *Sub-Model Evac Time* at the decision fusion step.	[0..100]	↑: Higher weighting on the final decision making during the decision fusion step. ↓: Lower weighting on the final decision making during the decision fusion step.
Alpha = 0.05	Controls the model's sensitivity to deviations from the training data, determining which data samples are classified as anomalies.	[0..1] Recommendation: [0.005..0.2]	↑: More sensitive to changes in the data samples compared to the data distribution of the training data samples. ↓: Less sensitive or more robust to changes in the data samples compared to the data distribution of the training data samples.

Table 4.3: Sub Model Evac times

4.3.3 Sub-Model Evac Time Ratio

The Sub-Model Evac Time Ratio reacts on changes of the ratio evac time / total duration of the vacuum generation process. Besides its weighting parameter for the decision fusion, the model has two further parameters to control its prediction behavior of the anomaly score (see Table 6). In Figure 12, valid configurations of the parameters X0 and Steepness are depicted.

Table 6: Parameters of the Sub-Model Evac Time Ratio.

Parameter and default value	Meaning	Value range	Effect of parameter change
Rater Sub-Model Evac Time Ratio = 10	Weighting parameter for the *Sub-Model Evac Time Ratio* at the decision fusion step.	[0..100]	↑: Higher weighting on the final decision making during the decision fusion step. ↓: Lower weighting on the final decision making during the decision fusion step.
X0 = 0.4	Controls how large an increase in the evacuation-time ratio must be before the anomaly score rises.	[0..1] Recommendation: [0.05.. 0.95]	↑: Higher values make the model less sensitive. ↓: Lower values make it more sensitive.
Steepness = 18.0	Controls how quickly the anomaly score rises as the evacuation-time ratio increases.	[0..500] Recommendation: [10..200]	↑: Higher values create a sharper transition ↓: Lower values create a more gradual one.

Table 4.4: Evac time ratios

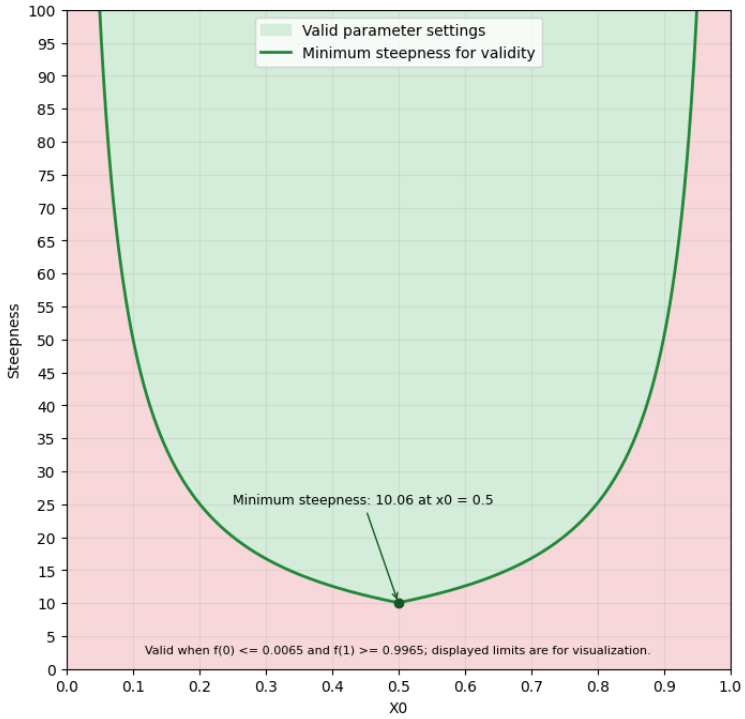


Figure 12: Valid parameter ranges for *X0* and *Steepness*.

4.3.4 Sub-Model Evac Time Recursive

This sub-model reacts on the initial evac time computed via the valve suction signal and the vacuum pressure switch signal. Table 7 summarizes the parameters of the sub-model to control the sub-model's weighting during the decision fusion step as well as to influence the anomaly score prediction behavior.

Table 7: Parameters of the Sub-Model Evac Time Recursive.

Parameter and default value	Meaning	Value range	Effect of parameter change
Rater Sub Model Evac Time Recursive = 1	Weighting parameter for the Sub-Model Evac Time Recursive at the decision fusion step.	[0..100]	↑: Higher weighting on the final decision making during the decision fusion step. ↓: Lower weighting on the final decision making during the decision fusion step.
Lambda Factor = 0.85	Controls how quickly the moving average adapts to new evacuation-time measurements during prediction.	[0..1]	↑: Higher values make it adapt more slowly ↓: Lower values make it respond more quickly
N = 3	Sets the normal range around the training data means to N standard deviations.	[1..5]	↑: Higher values widen the range, making anomaly detection less sensitive ↓: Lower values narrow it, making detection more sensitive.

Table 4.5: Evac time recursives

4.3.5 Sub-Model Evac Time Variance

This model reacts on changes of the variance of the initial evac time computed via the valve suction signal and the vacuum pressure switch signal. The two parameters listed in Table 8 control the sub-model's weighting during decision fusion and influence its anomaly-score prediction behavior.

Table 8: Parameters of the Sub-Model Evac Time Variance.

Parameter and default value	Meaning	Value range	Effect of parameter change
Rater Sub Model Evac Time Variance = 1	Weighting parameter for the Sub-Model Evac Time Variance at the decision fusion step.	[0..100]	↑: Higher weighting on the final decision making during the decision fusion step. ↓: Lower weighting on the final decision making during the decision fusion step.
Alpha F = 0.05	Controls the model's sensitivity to deviations of the variance of the training data, determining which variance of the last seen data samples are classified as anomalies.	[0..1] Recommendation: [0.005..0.2]	Controls how sensitive the model is to changes in the variance of evacuation times: ↑: Variance changes are more likely to be classified as anomalies. ↓: Variance changes are less likely to be classified as anomalies.

Table 4.6: Evac time variances

4.3.6 Sub-Model Festo VABF

The Sub-Model Festo VABF reacts on features and information from the Festo VABF vacuum ejector. This sub-model has only one parameter that controls the weight of its anomaly score during the decision fusion (cf. Table 9).

Table 9: Parameter of the Sub-Model Festo VABF.

Parameter and default value	Meaning	Value range	Effect of parameter change
Rater Sub-Model Festo VABF = 10	Weighting parameter for the Sub-Model Festo VABF at the decision fusion step.	[0..100]	↑: Higher weighting on the final decision making during the decision fusion step. ↓: Lower weighting on the final decision making during the decision fusion step.

Table 4.7: VABF sub model parameter

4.3.7 Sub-Model General Features

The Sub-Model General Features reacts on features derived from all available input data samples. Table 10 describes the effect of the sub-model's weighting parameter during the decision fusion step.

Table 10: Parameter of the Sub-Model General Features.

Parameter and default value	Meaning	Value range	Effect of parameter change
Rater Sub-Model General Features = 1	Weighting parameter for the Sub-Model General Features at the decision fusion step.	[0..100]	↑: Higher weighting on the final decision making during the decision fusion step. ↓: Lower weighting on the final decision making during the decision fusion step.

Table 4.8: Sub Model general features

4.3.8 Sub-Model Pressure Flow

This sub-model reacts on changes of the features computed based on the pressure and flow data samples. This sub-model has only one parameter that controls the weighting of its anomaly score during the decision fusion (cf. Table 11).

Table 11: Parameter of the Sub-Model Pressure Flow.

Parameter and default value	Meaning	Value range	Effect of parameter change
Rater Sub Model Pressure Flow Fault Detection = 1	Weighting parameter for the Sub-Model Pressure Flow at the decision fusion step.	[0..100]	↑: Higher weighting on the final decision making during the decision fusion step. ↓: Lower weighting on the final decision making during the decision fusion step.

Table 4.9: Sub Model pressure flow

4.3.9 Sub-Model Vacuum Reached

This sub-model reacts on the features indicating whether the vacuum pressure level is reached or not. It only comprises one parameter to control the weighting of the sub-model's anomaly score (see Table 12).

Table 12: Parameter of the Sub-Model Vacuum Reached.

Parameter and default value	Meaning	Value range	Effect of parameter change
Rater Sub Model Vacuum Reached = 10	Weighting parameter for the Sub-Model Vacuum Reached at the decision fusion step.	[0..100]	↑: Higher weighting on the final decision making during the decision fusion step. ↓: Lower weighting on the final decision making during the decision fusion step.

Table 4.10: Sub Model vacuum reached

4.3.10 Decision fusion approach

The predictions in the form of anomaly scores of all applicable sub-models are combined to an overall anomaly score ([0..1]; 0: healthy state, 1: abnormal state) and overall health score ([0..100]; 0: abnormal state, 100: healthy state) which show the overall health condition of the vacuum asset. The decision fusion method used depends on the configuration of the anomaly detection model. It can be configured with the parameter *Fusion Method* of the anomaly detection model. Table 13 shows the valid settings and their meaning of this parameter.

Besides this parameter, the sensitivity and the prediction behavior (amplification and attenuation) of the overall anomaly score or overall health score, respectively, can be influenced via the two parameters *X0 Global* and

Steepness Global (cf. Table 13). The valid parameter ranges for these two parameters have the same dependency as shown in Figure 12 for the Sub-Model Evac Time Ratio in Section 4.3.3.

The overall health score is the predicted output of the anomaly detection model. It is the indicator for an error or abnormal fault state of the underlying vacuum application. A guideline on how to interpret this health indicator can be found in Section 5.2.

Table 13: Parameter of the decision fusion approach.

Parameter and default value	Meaning	Value range	Effect of parameter change
Fusion Method = 1	With this parameter, the decision fusion method can be selected.	{1,2}	1: Use decision fusion method 1 - a rule-based, weighted average decision fusion method 2: Use decision fusion method 2 - a rule-based decision fusion method based on the maximum sub-model anomaly score or the minimum sub-model health score, respectively. **Note: This fusion method does not take the sub-models' weighting parameters into account during decision fusion.**
X0 Global = 0.5	Controls how large an increase in the fused anomaly score must be before the overall anomaly score rises. It controls how large an increase in the fused anomaly score must be before the overall health score decreases.	[0..1] Recommendation: [0.05.. 0.95]	↑: Higher values make the model less sensitive. ↓: Lower values make it more sensitive.
Steepness Global = 20	Controls how quickly the overall anomaly score rises as the fused anomaly score increases. It controls how quickly the overall health score decreases as the fused anomaly score increases.	[0..500] Recommendation: [10..200]	↑: Higher values create a sharper transition ↓: Lower values create a more gradual one.

Table 4.11: Decision fusion descriptions

4.4 Classification model

The classification model receives all the necessary information for fault classification from the data preprocessing model and the anomaly detection model. In the event of a detected fault or error, a fault classification is compiled from this ingoing information. The classification model of the AX Asset Type for Vacuum is based on a rule-based classification approach that leverages expert knowledge about the vacuum generation process to classify different vacuum process states reflected by the computed features and the predicted health score. Table 14 summarizes the classes which the classification model can predict.

In addition to the rule-based classification of the calculated features and predicted health score, the AX Asset Type for Vacuum takes also into account the on-board diagnosis results of the Festo VABF vacuum ejector during the fault diagnosis. So, if the model_type_id contains the digit 0.100.000, i.e., data samples from a VABF ejector are provided, the classification model also integrates the available fault diagnosis information from the VABF into the decision making. Table 15 summarizes the fault diagnosis codes sent by the VABF as well as the associated fault/error types and messages that are used in the classification model.

A guideline on how to interpret the fault classification results can be found in Section 5.2.

Table 14: Output of the classification model.

Classification output "Class"	Interpretation
Normal	No ongoing error or fault.
Small leakage detected	Indication of a small leakage inside the vacuum process. It is derived from an applicable evacuation time ratio rule or a rule based on re-evacuation cycle within in the total vacuum cycle.
Medium leakage detected	Indication of a medium-sized leakage inside the vacuum process. It is derived via a rule based on re-evacuation cycle within in the total vacuum cycle.
Large leakage detected	Indication of a large leakage inside the vacuum process. It is derived from an applicable evacuation time ratio rule or a rule based on re-evacuation cycle within in the total vacuum cycle.
Filter clogged	Filter-clogging indication from a specified zero-re-evacuation/ratio rule.
Vacuum pressure not reached	The vacuum application cannot reach its target vacuum pressure.
VABF-specific fault classes	See Table 15 for the integrated fault diagnosis information from the Festo VABF vacuum generator.

Table 4.12: Classification model

Table 15: Onboard diagnosis information from the Festo VABF vacuum generator that is used in the classification model of the AX Asset Type for Vacuum.

Code	Type	Exact German output	English translation
11	Fault	Fehler: Kurzschluss Ventil	Error: valve short circuit
13	Fault	Fehler: Drahtbruch Ventil	Error: valve wire break
29	Fault	Fehler: Parameter-/Teach-In-Fehler	Error: parameter/teach-in error
62	Fault	Fehler: Evakuierungs- oder Belüftungszeit überschritten	Error: evacuation or ventilation time exceeded
28	Warning	Warnung: Vakuumwert nicht erreicht	Warning: vacuum value not reached
37	Warning	Warnung: Schaltfrequenz Luftsparfunktion liegt über Grenzwert (1 Hz)	Warning: air-saving function switching frequency above limit (1 Hz)
111	Warning	Warnung: Prozessgüte unter Grenzwert	Warning: process quality below limit

Table 4.13: Diagnostic messages

4.5 Model training and model update / retraining

Before the anomaly detection model and fault classification model are active and make predictions, the first initial training of the anomaly detection model has to be completed. A scheduled training is configured by default and starts after the vacuum asset is on-boarded in the AX Core (see also Section 3.3). By default, the AX Asset Type for Vacuum performs an initial model training with 100 data samples. These data samples have to represent the OK/good/normal/fault-free condition of the underlying vacuum process to guarantee valid predictions of anomalies after the training phase have been completed, i.e., the data samples provided during the training phase sets the "baseline" for the ongoing anomaly detection. Note that 100 data samples used for the model training correspond to 100 vacuum cycles at the vacuum process.

You can check out the parameter configuration of the anomaly detection model that has been used during training in the model's overview of parameter configuration. This overview of the parameter configuration can be entered as shown in Figure 13. The parameter configuration of the anomaly detection model currently used by the respective vacuum asset is summarized in the AX Core as depicted in Figure 14. The meaning of these parameters is described in the previous Section 4.3.

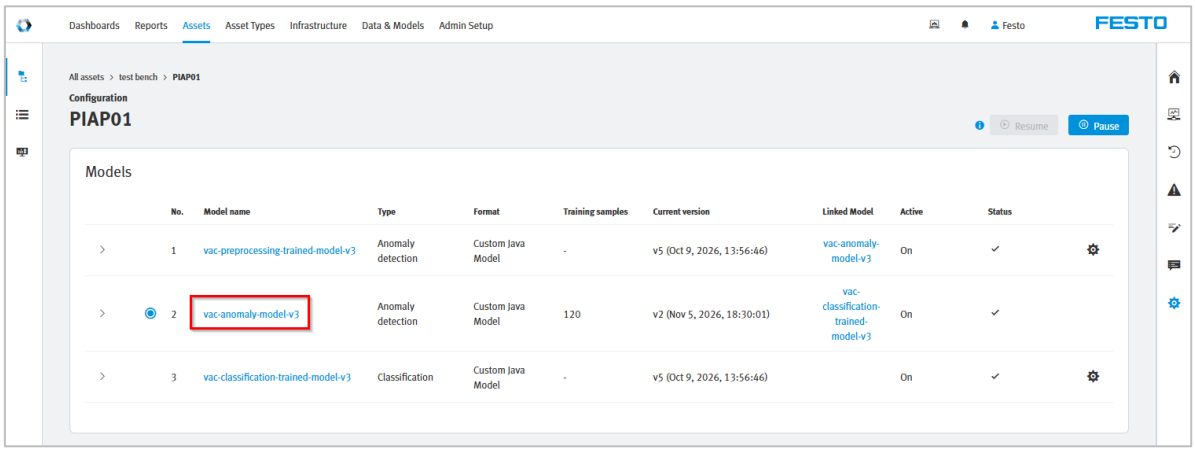


Figure 13: Entering the overview of the metadata of the anomaly detection model.

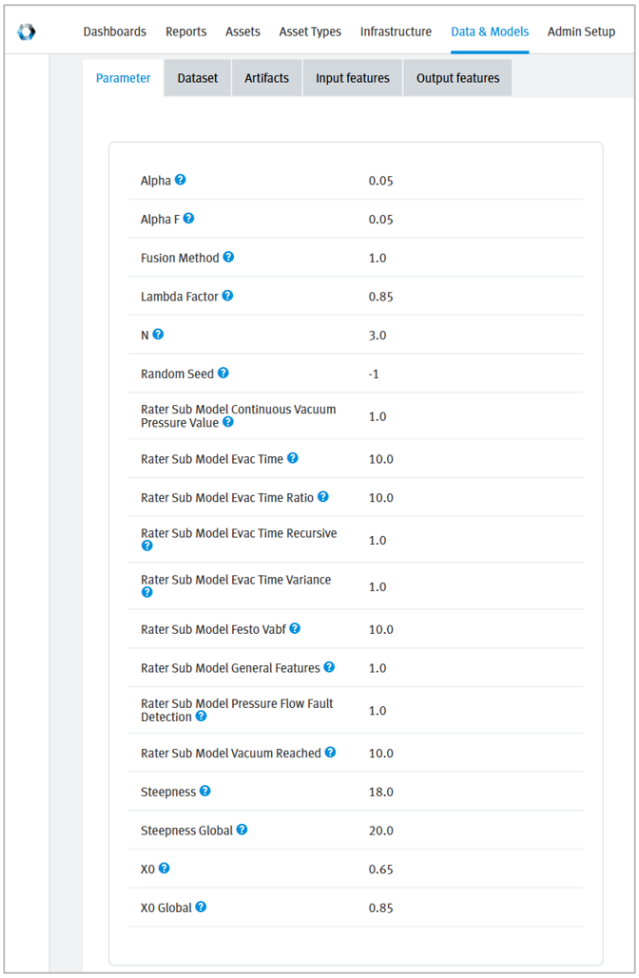


Figure 14: Parameter configuration that has been used during training of the anomaly detection model.

Sometimes it is necessary to retrain / update the anomaly detection model of an already running vacuum asset. The reasons for retraining can be:

- The data samples contained in the initial training dataset did not represent the normal condition of the vacuum application. As a result, the "baseline" of the anomaly detection model has to be updated.
- The baseline of the normal condition of the vacuum application has been changed over time, i.e., the anomaly detection model has to be re-fitted to the new data distribution representing this new baseline.
- Mechanical changes in the vacuum application, e.g., readjusting pressure levels or replacement of suction cups.

- The predictions of the anomaly detection model and classification model are incorrect or too sensitive to changes in the underlying data.
- **Note:** Vacuum Insights selects its data preprocessing paths, i.e., the applicable feature extractors, and the applicable sub-models of the anomaly detection model based on the available input variables. If the number of the provided input variables has been changed, the vacuum asset must be deleted from the AX core (see Application Note of the AX Core for further information). Subsequently, the vacuum asset has to be re-onboarded via the asset type package using the new initial MQTT input message containing the newly added input variables. A simple retraining of the anomaly detection model is not sufficient to adapt a vacuum asset to new incoming variables.

If a retraining / update of the anomaly detection model is necessary, the anomaly detection model can be retrained with the existing parameter configuration or with a fine-tuned parameter configuration. These two retraining options are explained in the following two subsections.

4.5.1 Model retraining with the existing parameter configuration

The retraining of the anomaly detection model of the AX Asset Type for Vacuum can be started via the Home Screen of the vacuum asset. This workflow to start a scheduled training with the existing or default parameter configuration of the asset is shown in Figure 15. Was the retraining process successfully started, its progress can be observed with the screen shown in Figure 16.

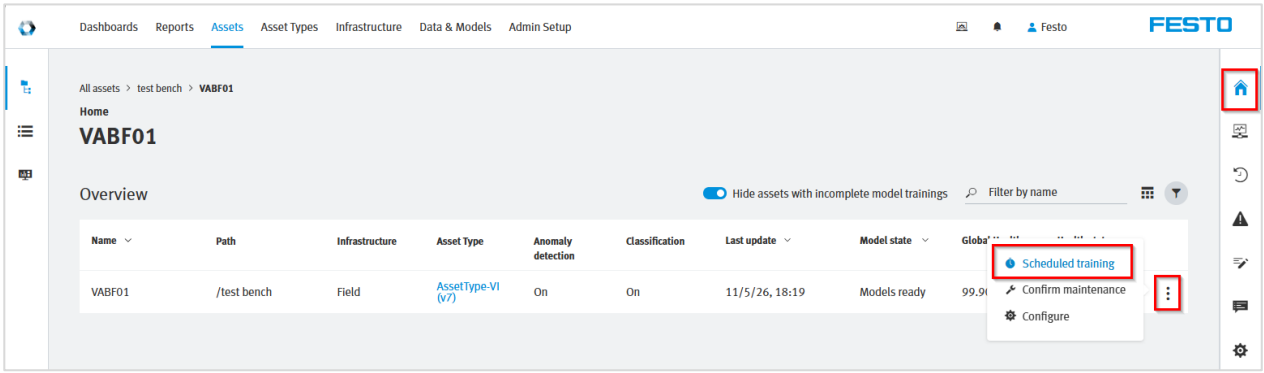


Figure 15: Starting a scheduled training via the home screen of the asset.

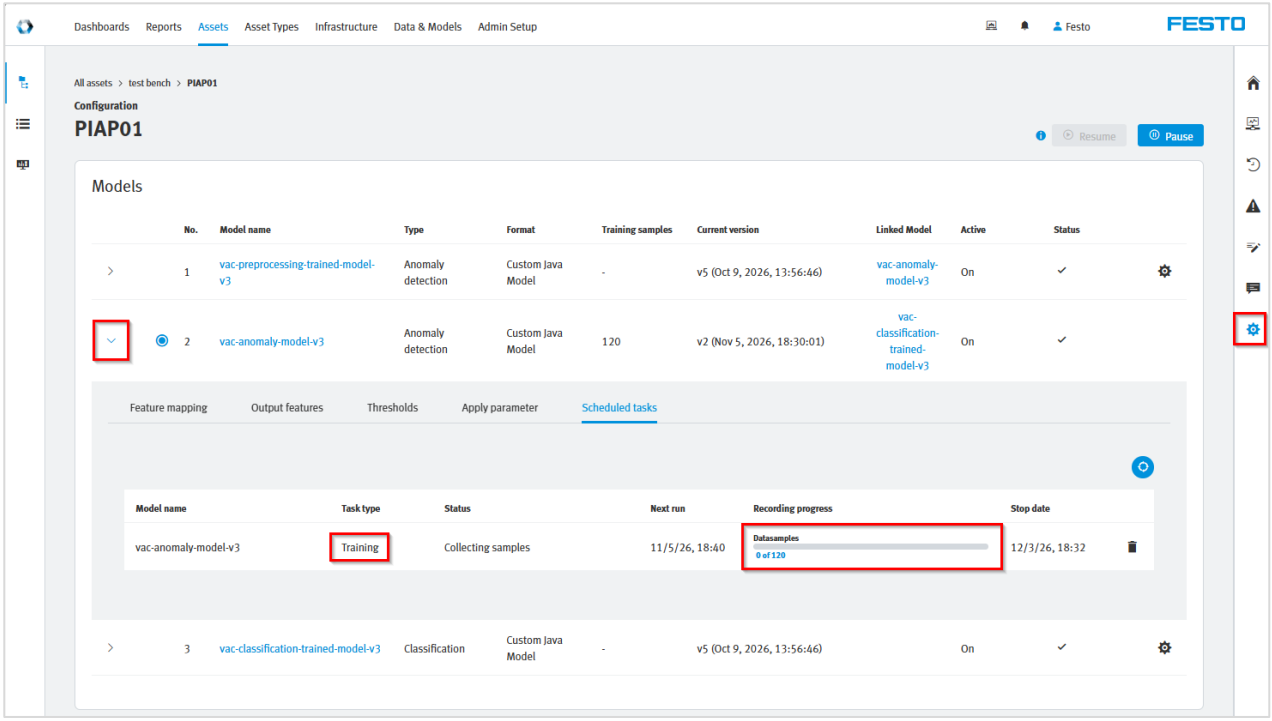


Figure 16: Model update - data samples recording and model training progress.

4.5.2 Model retraining with a fine-tuned parameter configuration

The retraining of the anomaly detection model of the AX Asset Type for Vacuum can be started with a fine-tuned parameter configuration. Fine-tuning the parameters of the anomaly detection model is particularly useful if the predictions made by the anomaly detection model and the classification model are incorrect, ambiguous (e.g., the health score is always in the interval [20, 80]) or too sensitive to normal changes in the incoming data samples.

A scheduled training with a customized number of training data samples and fine-tuned parameters can be started as depicted in Figure 17 (step 1).

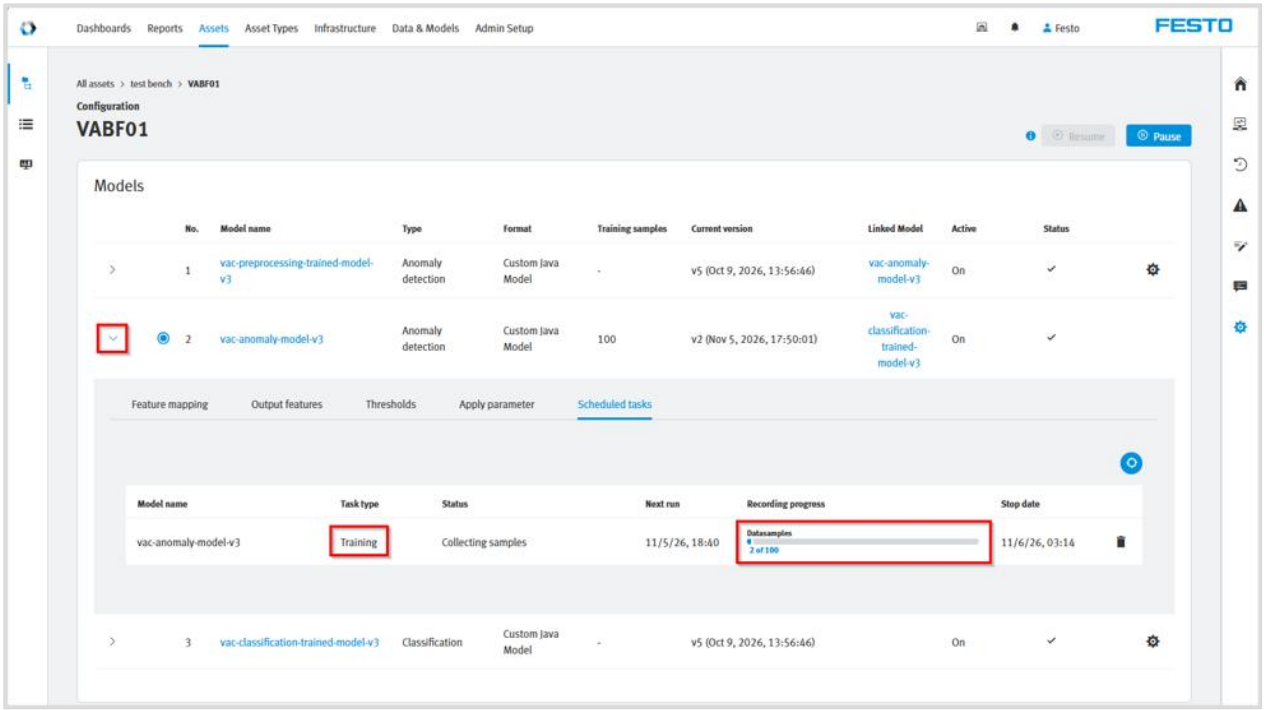


Figure 17: Model retraining / update via a scheduled training - step 1.

The second step of setting up a new scheduled training is shown in the subsequent Figure 18. In step 2, the number of training data samples can be configured. The default value is set to 100 data samples. Besides, you can set the time on the right when the data collection is forced to end for the model training.

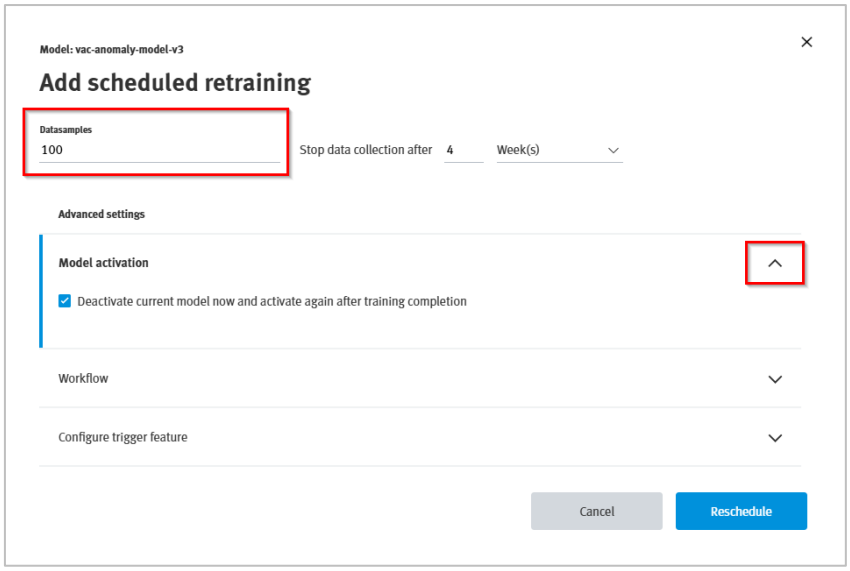


Figure 18: Model retraining / update via a scheduled training - step 2.

In step 3, the parameters of the anomaly detection model can be customized (cf. Figure 19). Please refer to Section 4.3 for a description of the different parameters and how these parameters can be changed to influence the prediction performance of the anomaly detection model.

Finally, the retraining / update process of the anomaly detection model is started by clicking on the button "Reschedule" (cf. Figure 19). The progress of the training phase can be observed via the same screen shown in Figure 16 for the scheduled training with the existing parameter configuration.

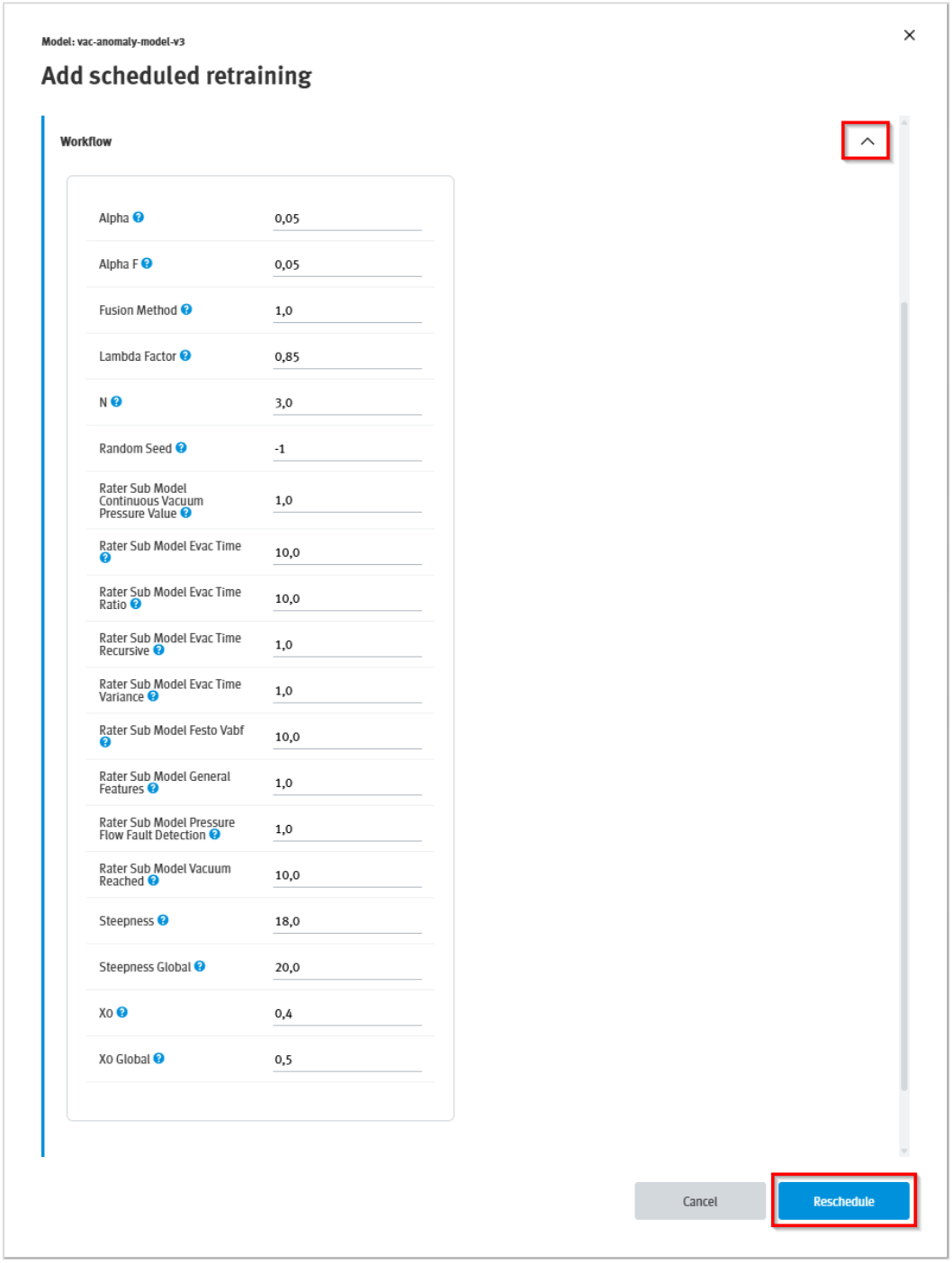


Figure 19: Model retraining / update via a scheduled training - step 3.

Note

After a retraining of the anomaly detection model, it is recommended to confirm baseline suitability with the application owner. Typically, an authorized user opens the relevant anomaly detection model, reviews its training configuration and evaluates its prediction performance. A new parameter configuration and successful model training alone does not verify the new baseline or the quality of monitoring.

5 When the application is running

This section presents the visual elements of the asset type dashboard for vacuum. In addition, this section gives hints on how to interpret the prediction results of the AX Asset Type for Vacuum and discusses the limitations of the asset type.

5.1 Vacuum Asset Type Dashboard Walkthrough

In the following, the visual views and elements of the asset type dashboard are presented. To open a vacuum asset type boarded in the AX Core, you have to navigate in the AX Core to the asset tree via the **Assets** screen (cf. Figure 20). In the asset tree, a vacuum asset can be selected. In doing so, the dashboard of the vacuum asset type is opened on the right to the asset tree as shown in Figure 20.

The asset tree provides access to the monitored assets. The onboarding of the AX Asset Type for Vacuum derives the asset path shown in the asset tree and the asset name from the configured MQTT topic expression of the respective asset. In Section 3.2, the details of the MQTT topic structure are described.

The dashboard screen of the vacuum asset type can be opened via the symbol on the right-side of the view shown in Figure 20. The dashboard of each vacuum asset comprises five tabs: **Overview**, **Analytics**, **Live Data**, **Historic Data** and **System Info**. These tabs are explained in the following five subsections. In addition, the Historic Asset Data View and the Event Statistics View of all assets in the AX Core are explained in the following.



Figure 20: Navigating through the asset tree.

5.1.1 Overview tab

The Overview tab shown in Figure 21 displays information about the health score predicted by the anomaly detection model. The event overview widget summarizes the anomaly events and classified anomalies which already occurred in the past or which are currently active. Two further text-based widgets show the current state of the vacuum process, i.e., whether the vacuum process is currently active or inactive and whether the target vacuum pressure level could be reached during the active phase of the vacuum cycle.

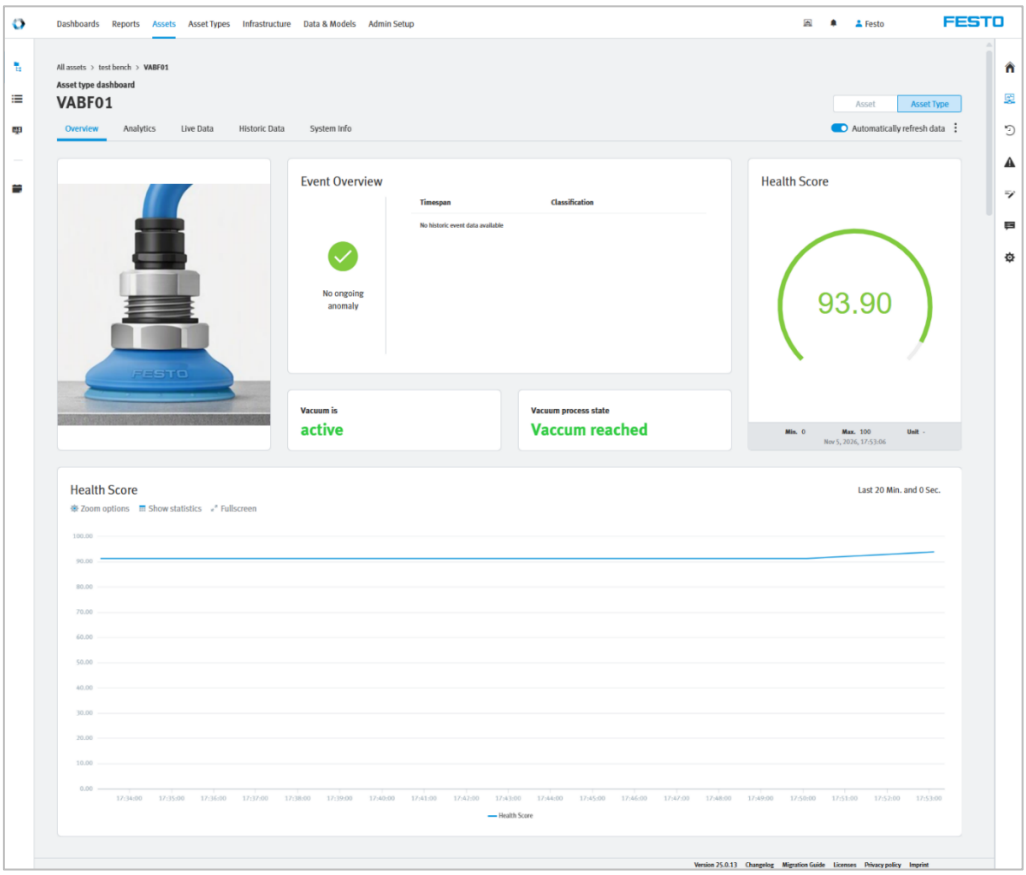


Figure 21: Vacuum asset type dashboard - Overview tab.

5.1.2 Analytics tab

The Analytics tab contains all relevant information regarding the overall anomaly score of the anomaly detection model as well as the anomaly scores predicted by the sub-models included in the anomaly detection model (see Figures 22 and 23). Each anomaly score is visualized via a gauge widget. Below each gauge widget, a description of the corresponding (sub-)model and its analytical capabilities is given. The analytics tab also includes a historical data widget that displays the trend of the overall anomaly score and the sub-model's anomaly score over the last 20 minutes.

Further information about the meaning of health / anomaly scores and on how to interpret these scores can be found in Section 5.2.

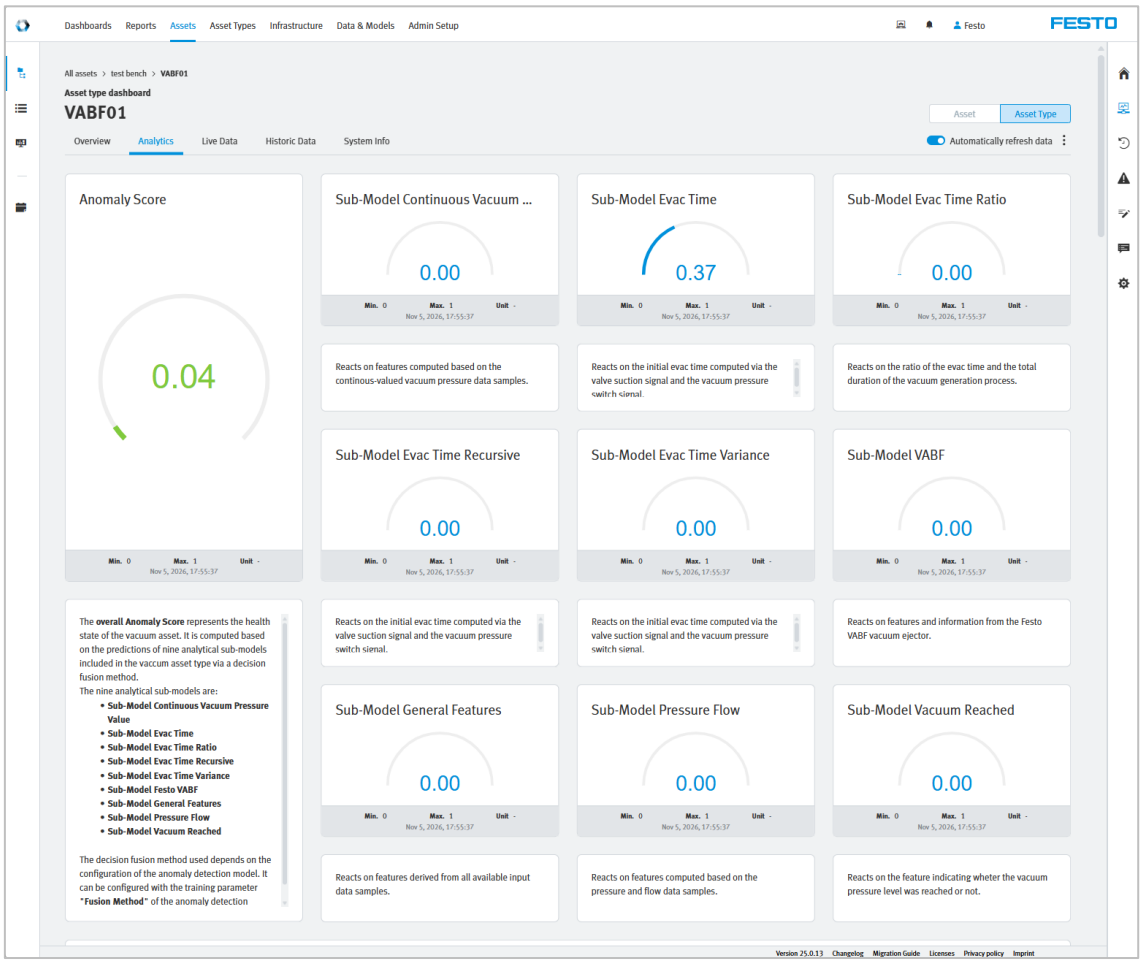


Figure 22: Vacuum asset type dashboard - Analytics tab (1 of 2).

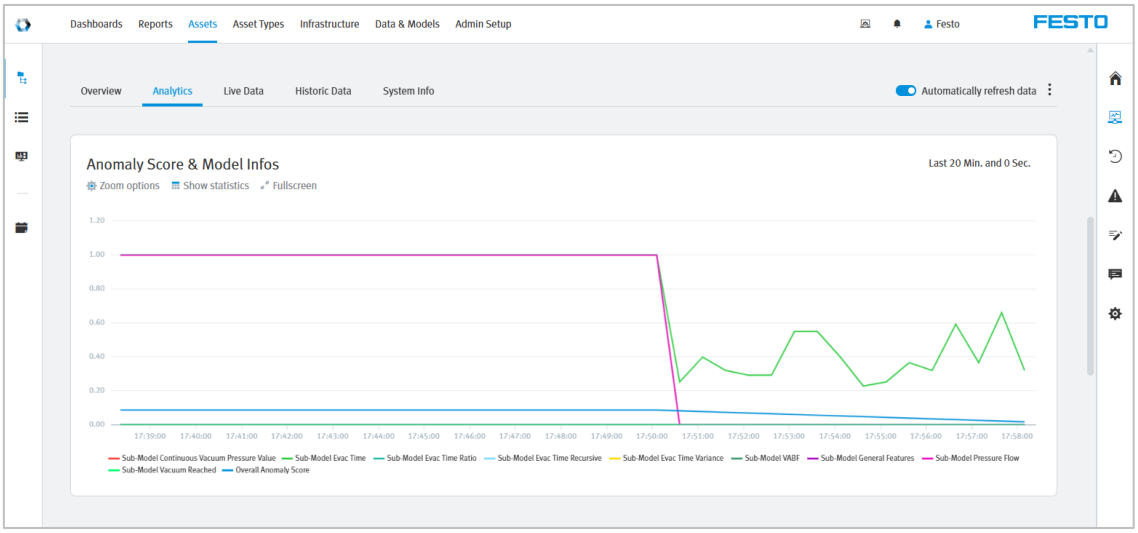


Figure 23: Vacuum asset type dashboard - Analytics tab (2 of 2).

5.1.3 Live Data tab

The live data tab displays the measurement data of the underlying vacuum process as well as the features computed by the adaptive feature extractors from the preprocessing model. For the high-frequency raw input data samples from the vacuum process, the last 5 minutes are plotted via historic data widgets (Figure 24). For the computed feature set, the last 60 minutes are plotted (Figure 25). Note, only these features are plotted in the historic data widgets that can be computed based on the provided variables in the input data samples.

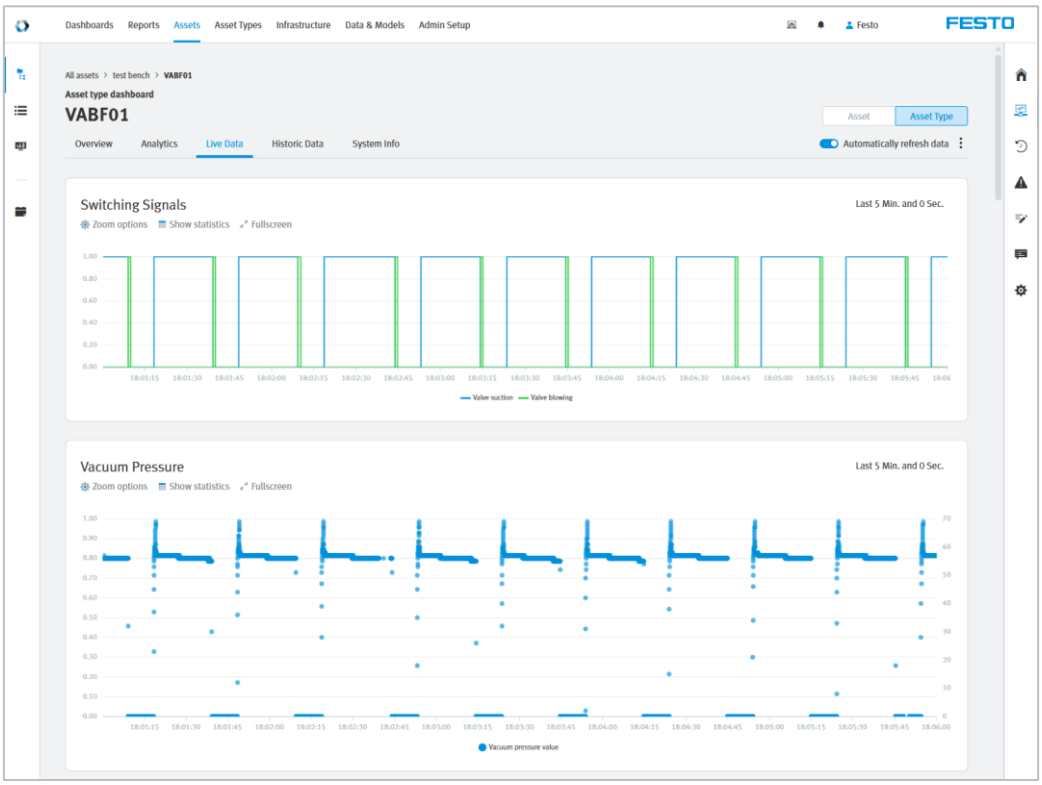


Figure 24: Vacuum asset type dashboard - Live Data tab (1 of 2).

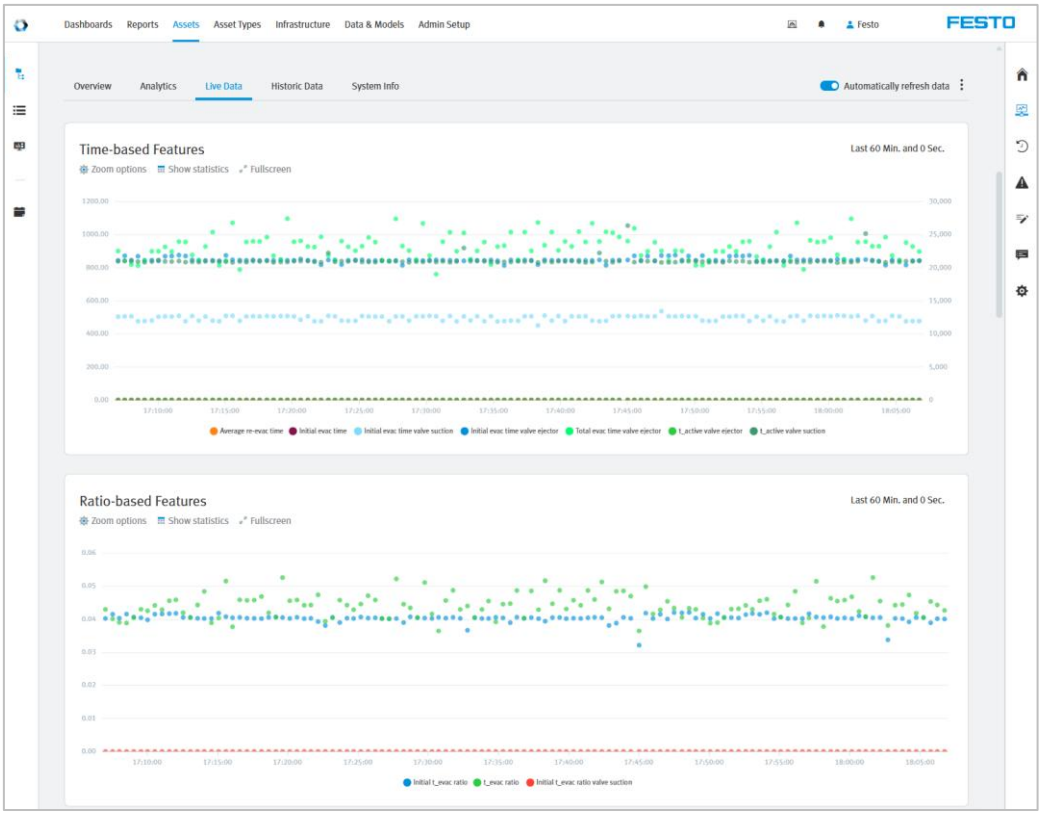


Figure 25: Vacuum asset type dashboard - Live Data tab (2 of 2).

5.1.4 Historic Data tab

Like the Live Data tab, the Historic Data tab displays the raw input data and the feature set computed by the pre-processing model but covers a longer historic period (see Figure 26). The historic data widgets show the data and features for the past week. In these historic data widgets, the raw data and features are aggregated as mean values over one-hour intervals.

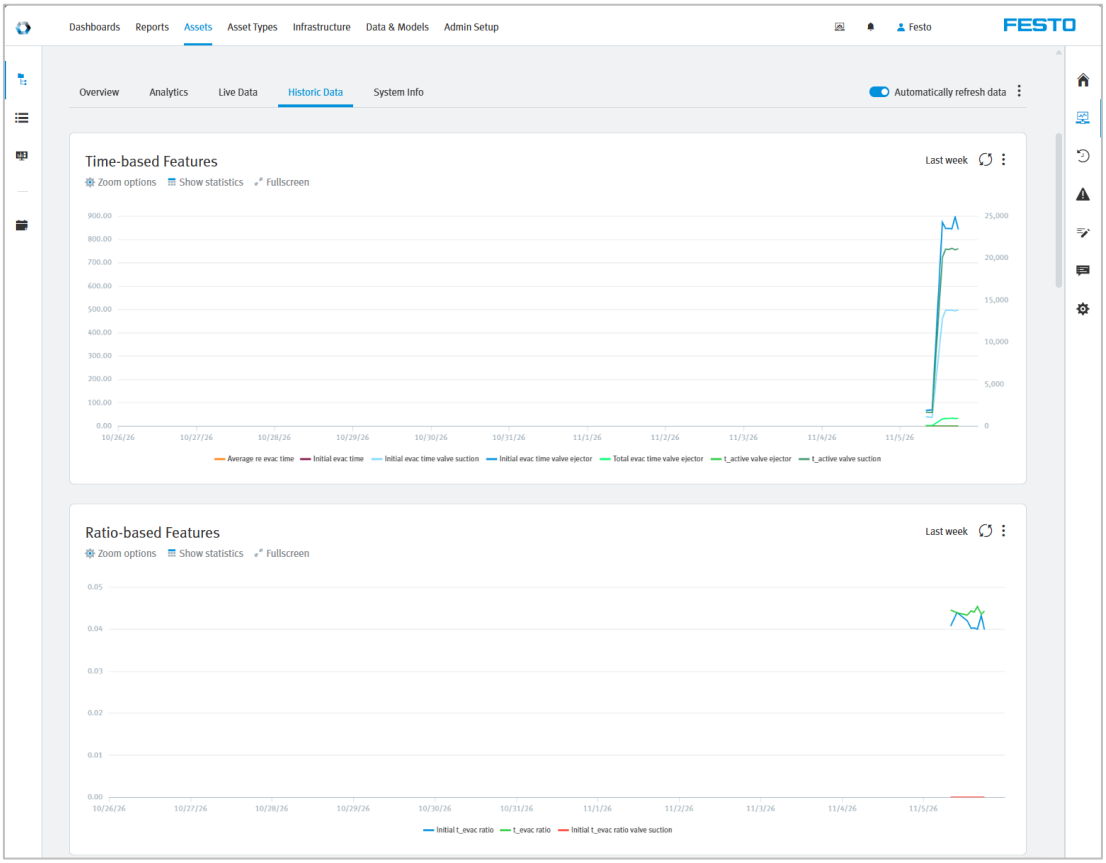


Figure 26: Vacuum asset type dashboard - Historic Data tab.

5.1.5 System Info tab

The System Info tab contains additional information about the vacuum asset, the anomaly detection model as well as the classification model (Figure 27). The additional information is visualized via key value widgets that show the last available value. The full range of additional information is available only after the initial model training has been successfully completed. The key value widget for the fault classification also shows the onboard fault diagnosis from Festo vacuum generators/ejectors, if available.

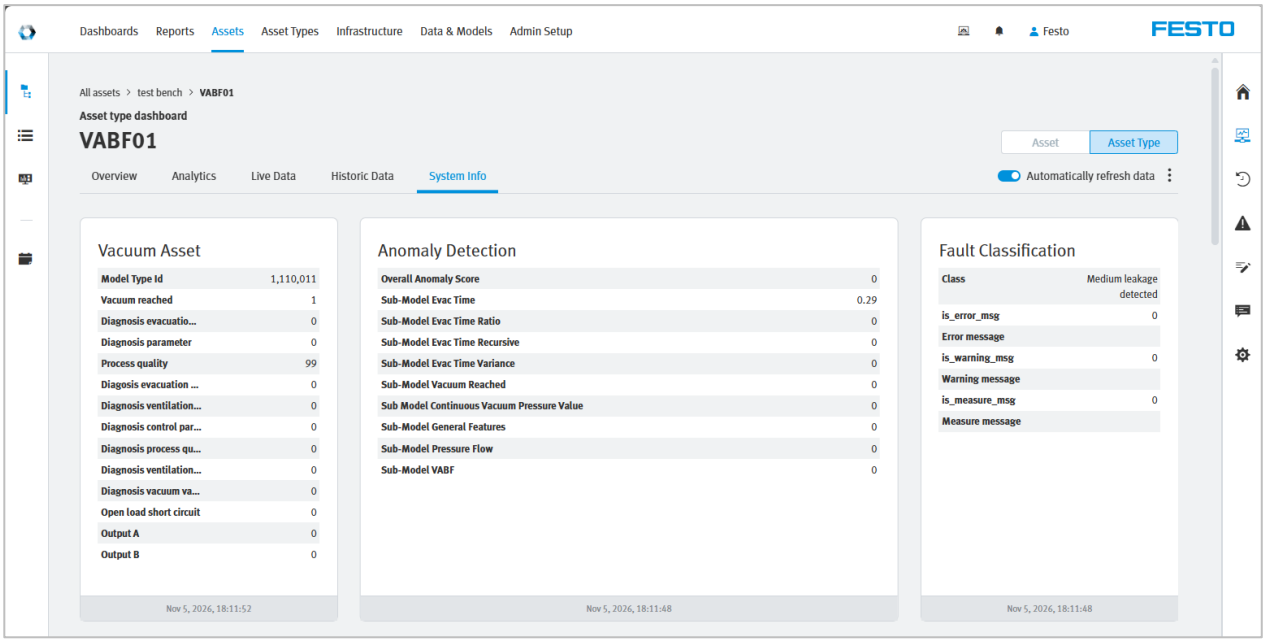


Figure 27: Vacuum Asset Type Dashboard - System Info View.

5.1.6 Historic Asset Data View

The vacuum asset type also includes the historic asset data view of AX Core. This historic data view can be entered by clicking on the symbol on the right (see Figure 28). The historic data view plots the last five minutes of the provided input data samples associated with the selected vacuum asset. The plot of the historic data view is well suited for debugging reasons, e.g., to identify faulty input signal mappings or a wrong switching behavior of the vacuum process reflected by the input data samples.

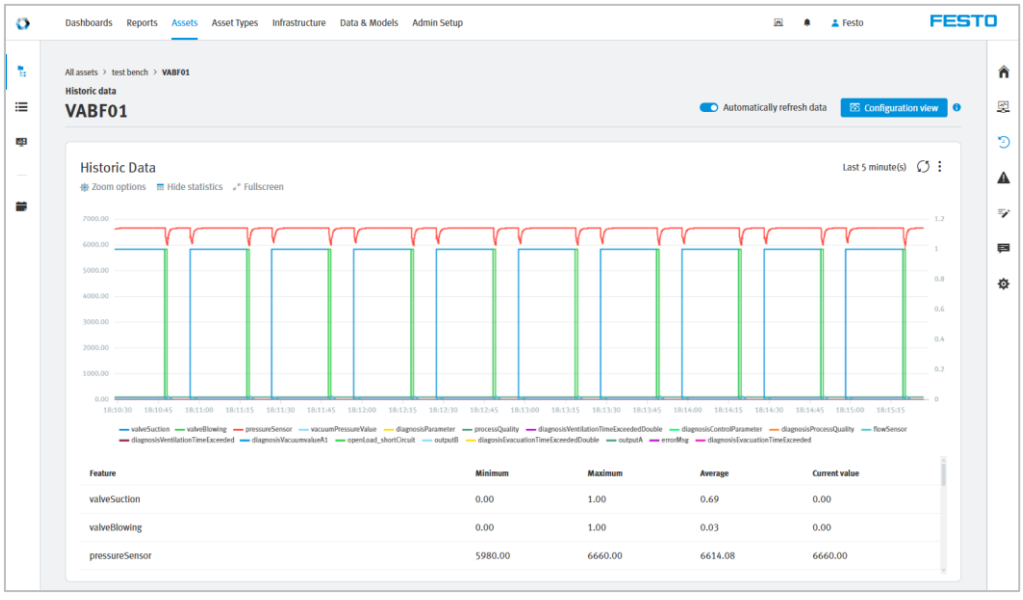


Figure 28: Vacuum Asset Type Dashboard - Historic data view.

5.1.7 Event Statistics View

The view of the event statistics of the asset can be accessed via the symbol on the right (see Figure 29). This event statistics view summarizes events and event types of the vacuum asset that occurred in the past.

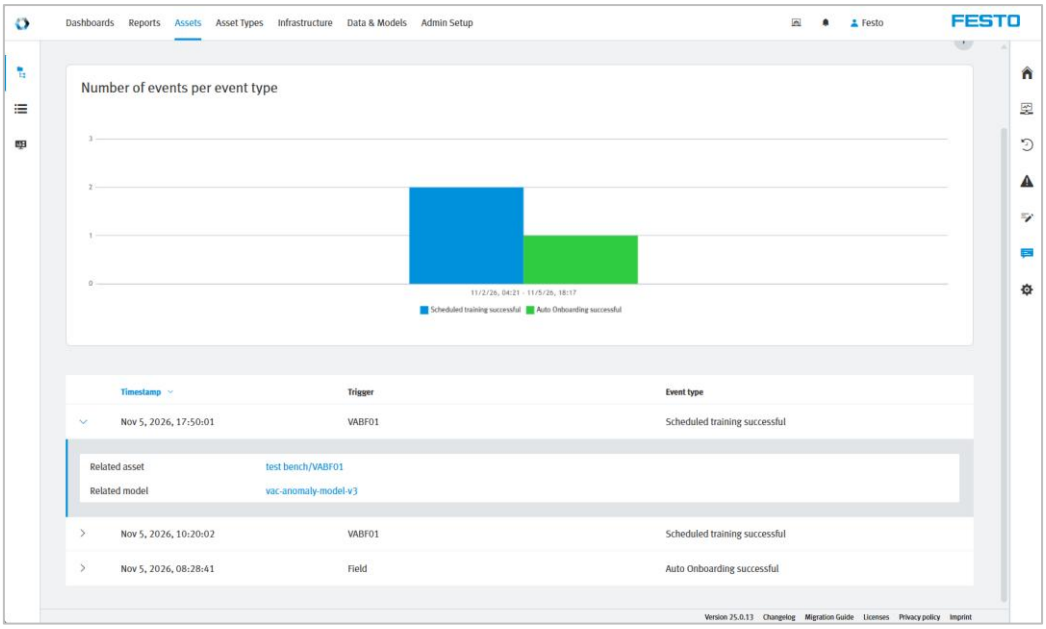


Figure 29: Vacuum asset type dashboard - Event statistics view.

5.2 Result Interpretation

This section describes how the health score of the vacuum asset, the anomaly scores of the sub-models and the class label predictions of the classification model can be interpreted with respect to fault detection and diagnosis. Table 16 summarizes the relevant diagnostic information and items presented on the vacuum asset type dashboard.

Table 16: Diagnostic information and items presented on the vacuum asset type dashboard.

Widget / item	Meaning	Value range
Anomaly Score — gauge widget and historic data widget	Displayed condition indicator derived from the predictions of the anomaly detection model	Numeric score [0..1] ≤ 0.3 green (OK) > 0.3 orange ≥ 0.7 red (Not OK)
Health Score — gauge widget and historic data widget	Displayed condition indicator derived from the anomaly score of the anomaly detection model. *Health Score = (1 – Anomaly Score) × 100*	Numeric score [0..100] ≥ 70 green (OK) > 30 orange ≤ 30 red (NOT OK)
Sub-model anomaly scores – gauge widget and historic data widget	Each sub-model predicts its own anomaly score. An anomaly score of zero can also mean inactivity. The anomaly scores of the sub-models are sensitive to the input data samples used by each sub-model (cf. Section 4.3).	Numeric score [0..1]
Class – event overview widget and key value widget	Prediction of the classification model. The class is either a fault class listed in Table 14 or an error / warning message sent by the VABF vacuum generator as listed in Table 15.	Text-based class label (cf. Section 4.4)
Error message – key value widget	VABF-specific error message.	Text-based message (cf. Section 4.4)
Warning message – key value widget	VABF-specific warning message.	Text-based message (cf. Section 4.4)
Measure message – key value widget	Measure guidance to correct errors and to fix warnings.	Text-based message (cf. Table 17)

Table 5.1: Diagnostic information

An anomaly score or health score reports unusual behavior of the vacuum asset relative to the baseline learned during the training phase of the anomaly detection model. The health score decreases as the anomaly score increases. Figure 30 shows the thresholds *Anomaly On* and *Anomaly Off* of the anomaly score predicted by the anomaly detection model. If the anomaly score is greater than the *Anomaly On* threshold, the anomaly event is triggered. This anomaly event is then shown in the Event Overview widget in the Overview tab (see Figure 21). The anomaly event is displayed as "ongoing" as long as the anomaly score is above the *Anomaly Off* threshold.

Each occurring anomaly event is classified by the classification model. The predicted class label is also shown in the Event Overview widget in the Overview tab. The classes that can be predicted by the classification model are listed in Section 4.4.

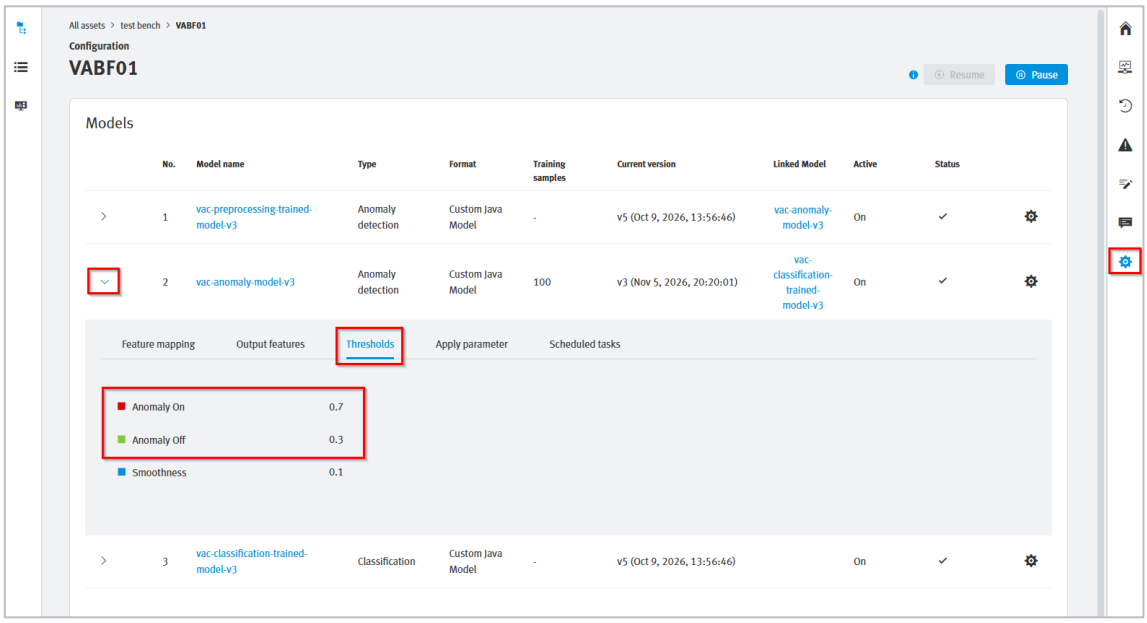


Figure 30: Thresholds for an anomaly event.

If the onboard diagnostic function of the Festo VABF vacuum generator provides further diagnostic information during the condition monitoring with the Asset Type for Vacuum, this additional diagnostic information is shown in the fault classification key value widget on the System Info tab. This additional diagnostic information can comprise an error or warning message as well as a measure message. Table 15 summarizes the error and warning codes as well as the associated error and warning messages. These error and warning messages are also thrown as a fault class by the classification model, if an anomaly event is triggered and if the VABF provides this additional fault information. The measure message indicates some corrective actions to correct the error occurred or to resolve the condition indicated by the warning message sent by the VABF. Table 17 lists the available measure texts for each error / warning code sent by the VABF.

Table 17: VABF measure texts.

Code(s)	German measure text	English translation
11, 13	Ventil tauschen	Replace valve
28	Auf Leckage prüfen	Check for leakage
29	Einstellungen prüfen (z. B. Wertebereich) oder Teachen wiederholen.	Check settings (e.g. value range) or repeat teach-in.
37	Luftsparfunktion deaktivieren oder auf Leckage prüfen.	Deactivate air-saving function or check for leakage.
62, 111	Auf Leckage prüfen.	Check for leakage.

Table 5.2: VABF codes descriptions

For each processed vacuum cycle, the anomaly score predicted by the anomaly detection model and the class label predicted by the classification model are also shown in the key value widgets on the System Info tab.

The historic data widgets plotting the health score and the anomaly scores predicted by the sub-models can be used for diagnosing anomaly events that have been occurred. Trends within the historic period of the health score or the sub-model's anomaly scores can indicate specific fault patterns. In addition, each sub-model's anomaly score is sensitive only to changes in the input data samples on which it is based (cf. Section 4.3). This means that the current predicted values and the historic trend of these anomaly scores may indicate specific fault patterns. Therefore, it is worth examining each sub-model's anomaly score and its trend, in addition to the overall anomaly score.

6 Frequently Asked Questions

How many vacuum cycles are required for the initial training?

The default number of training samples for the initial training of the anomaly detection model of the AX Asset Type for Vacuum is set to 100, i.e., 100 vacuum cycles are required for the initial training.

Does the training phase automatically discard faulty training data?

The training step of the anomaly detection model performs no health labeling or filtering of the provided training data samples. The supplied data samples during the data recording of the training phase define the baseline for the anomaly detection. It is recommended to confirm representative normal operation data with the application owner.

Which vacuum generators and ejectors are officially supported?

The asset type package supports all types of vacuum generators and ejectors that can provide the expected set of input variables. See Section 2 for further details. The supported vacuum generators and ejectors from Festo that were successfully tested are the VN, OVEM, VUVX and VABF.

What happens if the connected signal set or provided set of variables in the MQTT payload changes?

Vacuum Insights selects its data preprocessing configuration and the applicable sub-models of the anomaly detection model based on the available input variables. If the number of the provided input variables has been changed, the vacuum asset must be deleted from the AX core (see Application Note of the AX Core). Subsequently, the vacuum asset must be re-onboarded via the asset type package using the new initial MQTT input message containing the newly added input variables.

What happens after a mechanical component or process change?

It is recommended to confirm the baseline suitability of an already running anomaly detection model after mechanical, recipe or process changes. If the anomaly detection results are inaccurate even though the underlying vacuum process is stable and operating normally, the anomaly detection model must be retrained using new input data samples that represent the new baseline.

Why is a sub-model anomaly score zero?

Zero can mean an inactive sub-model as well as a low score from an active analysis. Check the input profile and model type id. Further details can be found in Sections 4.3 and 5.2

Why does the health score remain in an ambiguous range, such as between 20 and 80, instead of approaching 100, even though my vacuum application is operating normally?

The anomaly detection model may be overly sensitive to normal variations in the input data, or its training baseline may not adequately represent normal operation. Review the individual sub-model anomaly scores and their trends, then consider fine-tuning the relevant parameters and retraining the model using representative data from normal, fault-free operation. After retraining, verify that the predictions appropriately reflect the application's actual condition. The sensitivity of the health score can be directly controlled via model parameters (see Section 4).

I also have other industrial automation components, e.g., pneumatic cylinders and electrical axes. Can I monitor them as well?

Standardized AX Industrial Apps for pneumatic cylinder monitoring and e-axis-monitoring are also available. Generally, the Festo AX Industrial Intelligence suite can be used to tackle various challenges. Find out more here: https://www.festo.com/ax

What is the required minimum data sampling interval / time?

The AX Asset Type for Vacuum can be correctly applied if the sampling time of the input data samples of each vacuum cycle meets the required minimum sampling time condition:

$$evacuation\ time\ vacuum\ ejector > 2 \cdot sampling\ time\ of\ the\ raw\ data\ samples \quad .$$

I have specific questions. How can I get in contact?

Get in contact with your Festo contact person or through https://www.festo.com/contact