AN14549

How to Use Time Series Studio to Deploy Anomaly Detection Model based on MCXA156

Rev. 1.0 — 10 February 2025 Application note

Document information

Information	Content
Keywords	AN14549, Time Series Studio Anomaly detection, TSS, Time Series, FRDM-MCXA153, eIQ
Abstract	This application note illustrates the process of using eIQ Time Series Studio to train and deploy an anomaly detection model based on the FRDM-MCXA156.

1 Introduction

This document illustrates the process of using eIQ Time Series Studio to train and deploy an anomaly detection model based on the FRDM-MCXA156. eIQ Time Series Studio is an end-to-end Bring Your Own Device (BYOD) development tool that reduces the complexity and development time of time series-based AI models. Moreover, there is a demonstration titled “TSS Powered on-device learning Fan Anomaly based on MCXA156” available in the application code hub. This demonstration example senses the operating status of the fan through the acceleration sensor mounted on the fan. To determine any abnormal operating status of the fan, the demonstration example uses the model trained by Time Series Studio (TSS). Also, the example supports the data capture function, which can provide training data for the TSS tool.

2 Time Series Studio

Time Series Studio IDE provides an end-to-end solution enabling engineers to develop time series-based machine learning solutions from initial evaluation to final production.

The IDE offers:

- time series data logging
- data visualization
- data smart analysis
- automatic machine learning optimization
- model emulation
- library generation for the deployment

The following diagram illustrates the complete workflow.

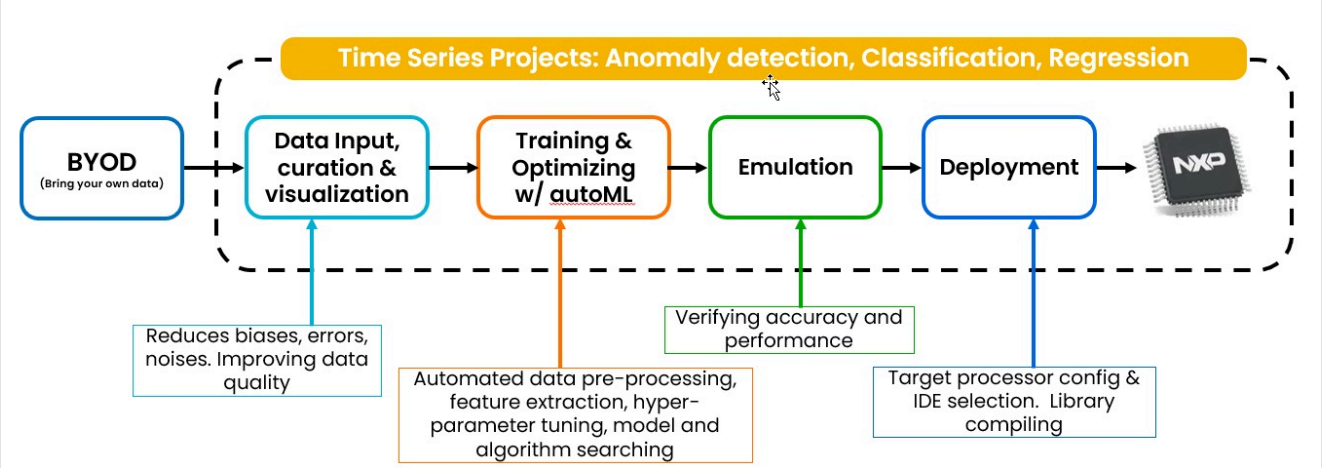

Figure 1. Time Series Studio IDE workflow

- It supports time series dataset input, which the customer completely owns.
- From the dataset input, it automatically trains out the best model list.
- Generate the algorithm header file and runtime library by selected CPU core.

3 Prepare the hardware

The following are the hardware requirements:

- Main board: FRDM-MCXA156
- LCD panel: PAR-LCD-S035

- Accelerator sensor: ACCEL-4-Click
- Type-C USB cable
- USB fan

1. Connect the LCD panel to the FRDM board.

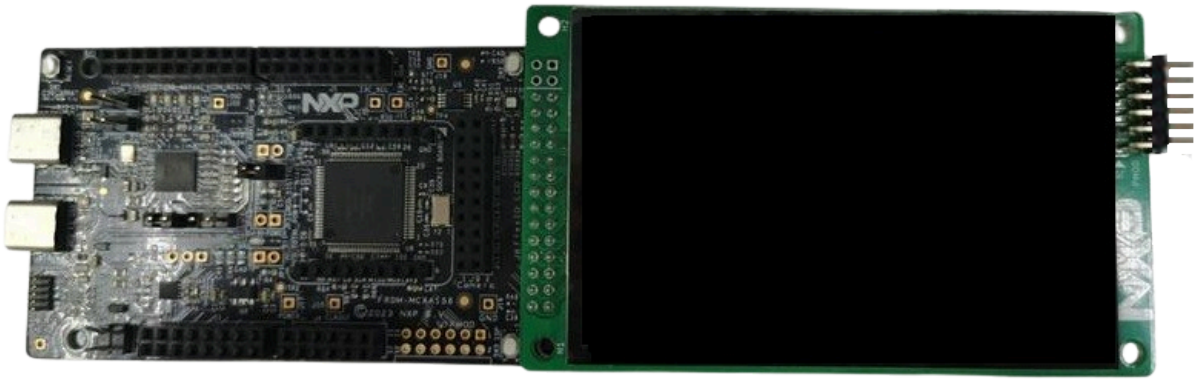

Figure 2. Connect the LCD panel to the FRDM board

2. Connect the Accelerator sensor with FRDM

Sensor connection

FRDM Mikro Bus	Accel 4 click
J6-7	3V3
J6-8	GND
J5-5	SCL
J5-6	SDA
J5-2	INT1

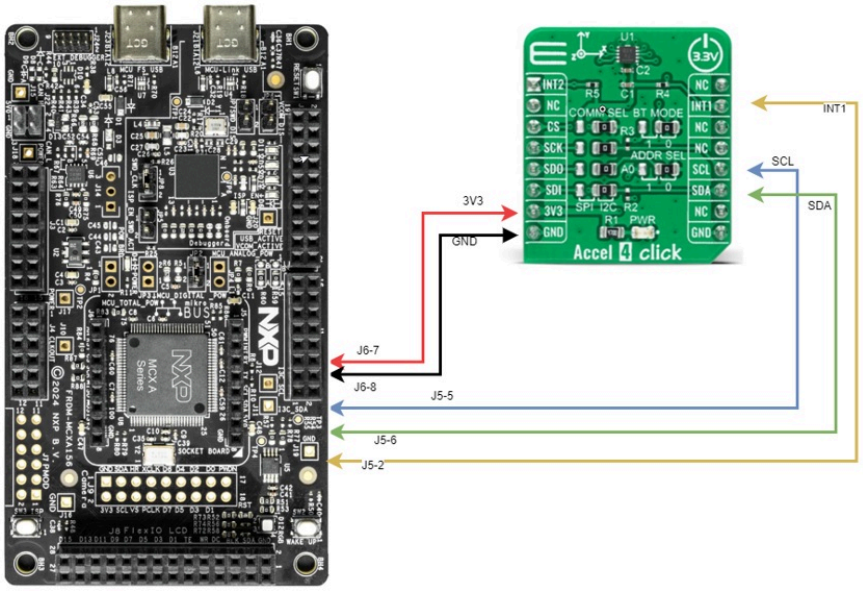

Figure 3. Connect the accelerator sensor with FRDM

3. Mount the accelerator sensor on the FAN.

Figure 4. Mount the accelerator sensor on the FAN

The complete hardware connection appears as in the following figure.

Figure 5. Complete hardware connection

4 Install the software

1. Download and install eIQ ToolKit 1.13.1. After successful installation, the IDE appears as in the following figure.

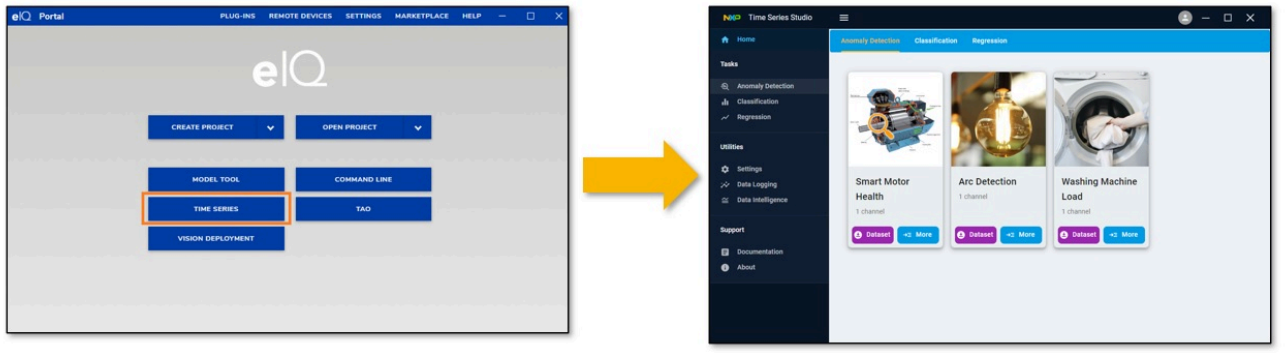

Figure 6. Install the software

2. Download and install MCUXpresso V11.9.0 or later.
3. Download and install SDK_2_16_0_FRDM-MCXA156 from the SDK builder.

5 Set up the demo

1. Open MCUXpresso from the Quickstart panel.
2. To import the mode project, click **Import from Application Code Hub**.
3. Type '`dm-tss-powered-on-device-learning-fan-anomaly-based-on-mcxa156`' in the searching text box, the project appears in the list.
4. Click the project.
   The project page appears in the window.
5. Click the **GitHub** link on the top of the window.
   The repository of the project is stored
6. Click the **Next** button at the bottom and download the project from GitHub.

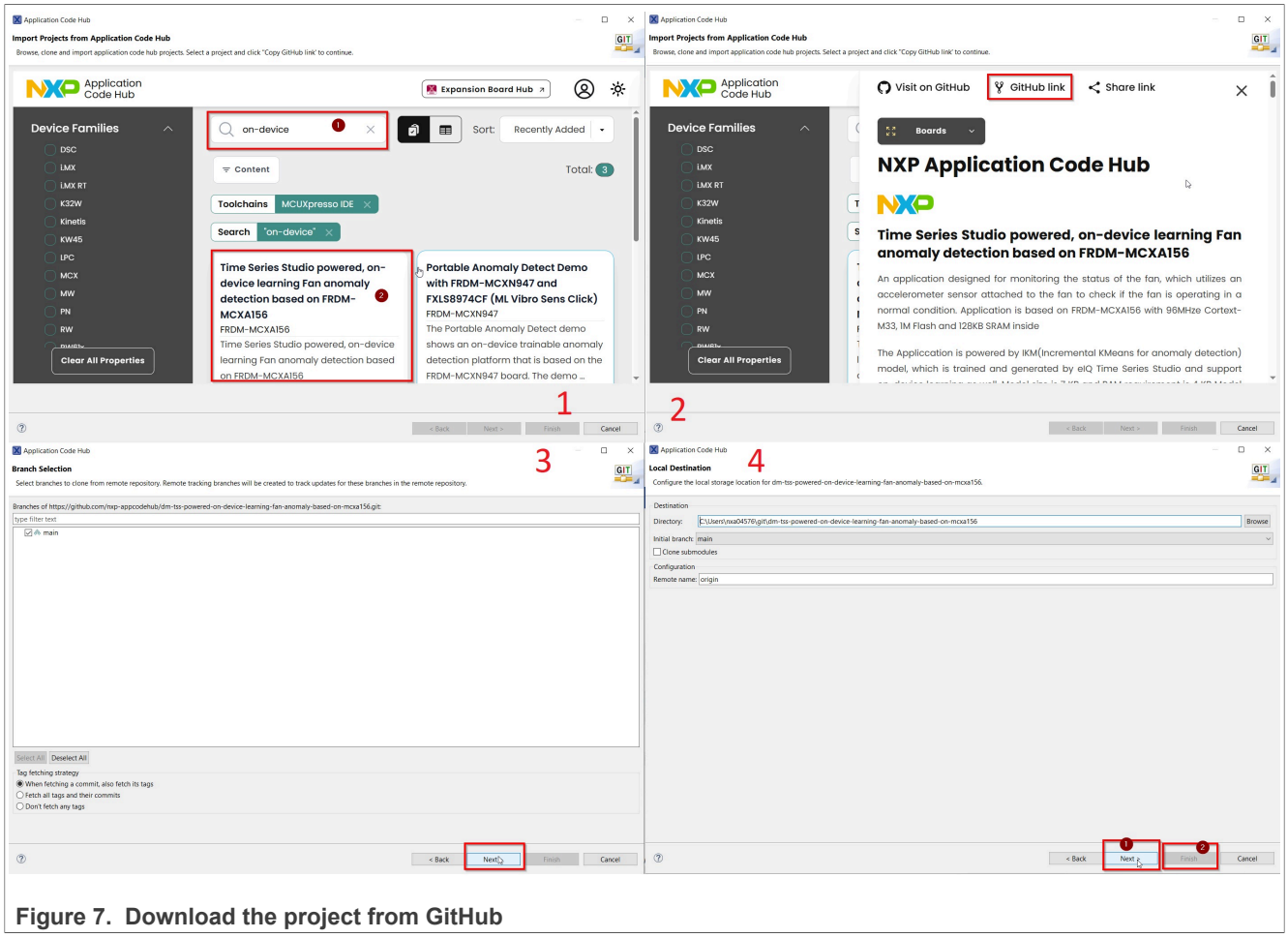

Figure 7. Download the project from GitHub

7. Connect the debug port of the FRDM board with your laptop,
8. Compile the project and download to the board.
9. After the download, completely reset the board.

6 Create a dataset

To create a dataset:

1. Click the **Data Collect** button on the device.
2. Open the **Data Collector** window.
   The UART generates the data of the accelerated sensor with the baud rate at 230400.
3. To open the data logger on your laptop, navigate to the folder `scripts` of the source code, execute the script `python make_live_ts_dataset.py` in the command line.
   The **Dataset Maker** window appears as in the following figure.

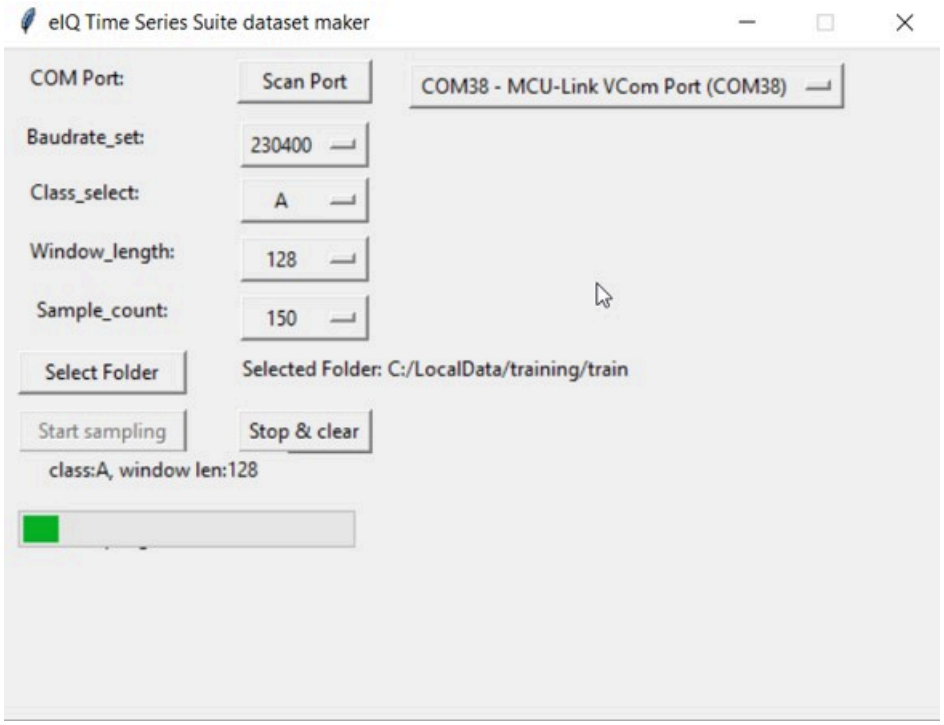

Figure 8. Data Maker

4. Connect the debug port of the FRDM board with the laptop.
5. Select the com port and set the baud rate to 230400.
6. Receiving and save the data files for train and validate data sets.
7. Create training sets.
   Select the folder and store the train sets.
   a. Turn on the fan.
   b. Set class to `A`, window length to `128`, and sample count to `150`.
   c. Click the **Start sampling** button.
   d. Wait for the progress bar to stop at the end.
   e. Click the **Stop & clear** button.
   f. Switch the fan speed to mid.
   g. Click the **Start sampling** button.
   h. Wait for the progress bar stop at the end.
   i. Click the **Stop & clear** button.
   j. Switch the fan speed to fast.
   k. Click the **Start sampling** button.
   l. Wait for the progress bar stop at the end.
   m. Click the **Stop & clear** button.
   n. Turn off the fan.
   o. Set class to B.
   p. Click the **Start sampling** button.
   q. Wait for the progress bar to stop at the end.
   r. Click the **Stop & clear** button.
8. Create validation sets.
   Select the folder and store the validated sets.
   a. Turn on the fan.

b. Set class to `A`, window length to `128`, and sample count to `50`.
c. Click the **Start sampling** button.
d. Wait for the progress bar to stop at the end.
e. Click the **Stop & clear** button.
f. Switch the fan speed to mid.
g. Click the **Start sampling** button.
h. Wait for the progress bar stop at the end.
i. Click the **Stop & clear** button.
j. Switch the fan speed to fast.
k. Click the **Start sampling** button.
l. Wait for the progress bar stop at the end.
m. Click the **Stop & clear** button.
n. Turn off the fan.
o. Set class to B.
p. Click the **Start sampling** button.
q. Wait for the progress bar to stop at the end.
r. Click the **Stop & clear** button.

7 Train the model

To train the model:

1. Open the eIQ Time Series Studio IDE.
2. Select **Anomaly Detection** under **Tasks**.
3. Create a project.
4. Select FRDM-MCXA153 in your target,
5. Set the **Number of channels** to 3, as in the following figure.

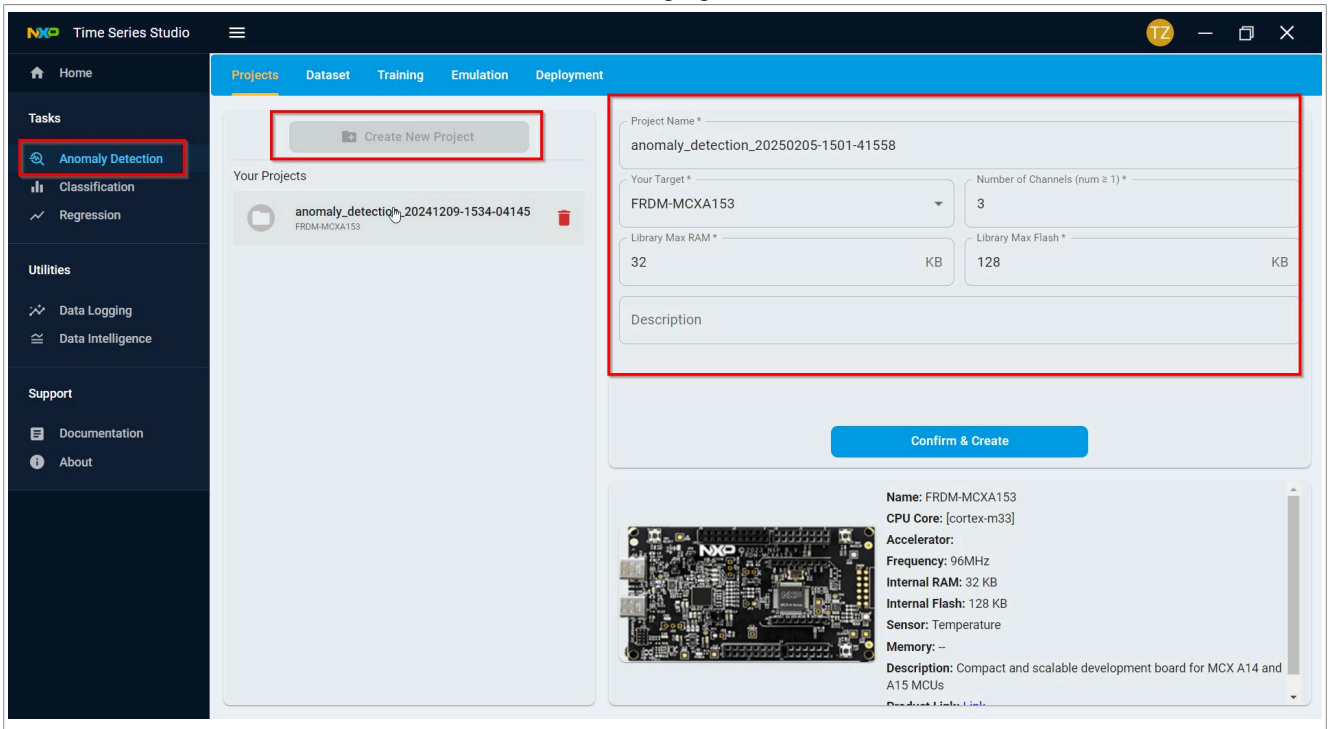

Figure 9. Set the number of channels

6. Click **Save & Next** and open the **Dataset** window.
7. Click **Normal Data**.
8. Select all the `A_xxx.csv` files from the train folder.
9. Click **Load Anomaly Data**.
10. Select the `B_xxx.csv` file from the train folder, shown in the following figure.
    In this window, you can see the visualization of the data and can view the data in different ways.

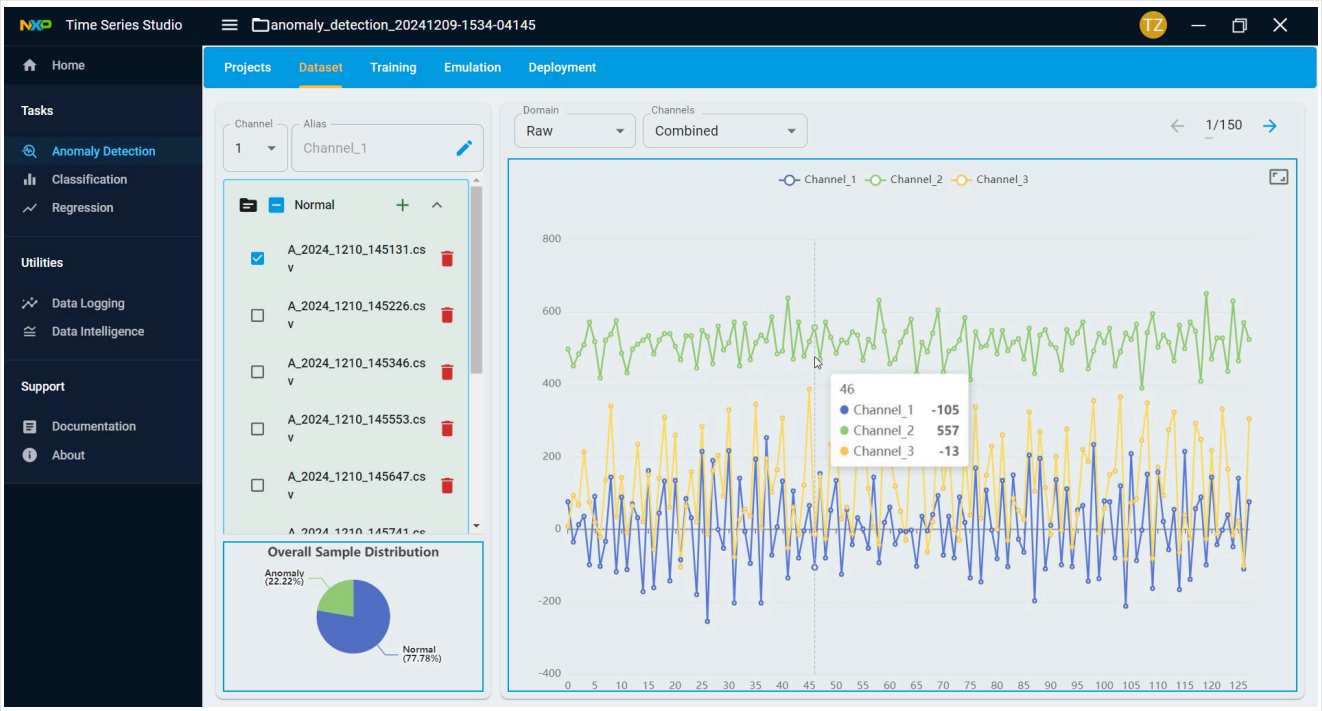

Figure 10. Create validation sets

11. Click the **Training** window.
12. Click the **Start New Training** button and start the model train.
13. If you want to enable the on-device learning feature, then turn on the switch of On-Device Learn. Supports quick search also.
14. After the training completes, accuracy, flash, RAM size, and confusion matrix appears for each model created. You can sort the models by accuracy, or by flash/RAM size.

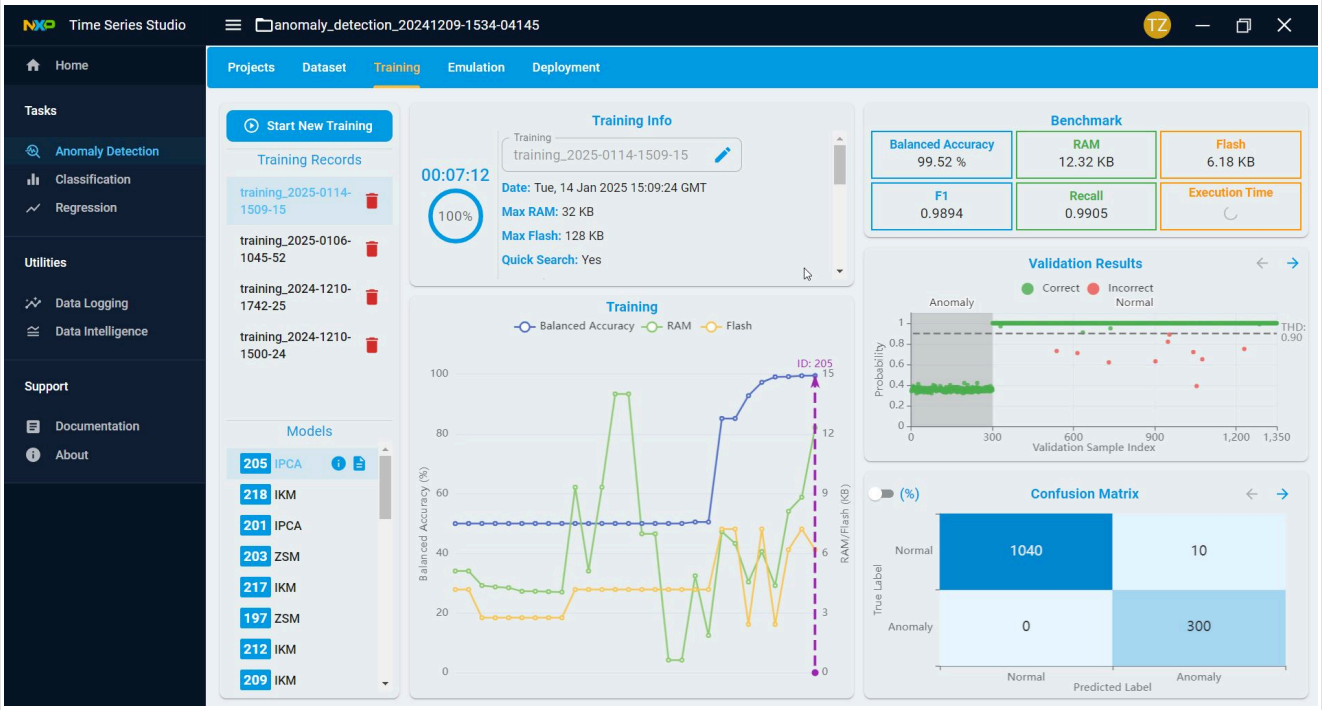

Figure 11. Training information

8 Emulate the model

To emulate the model:

1. Click into the **Emulation** window.
2. Select the models that you want to emulate.
3. Click the **New Emulation** button.
   A pop-up window appears.
4. Select the **normal and anomaly data** from the **Validate** folder.
   After emulation, the complete result appears in the report window as in the following figure.

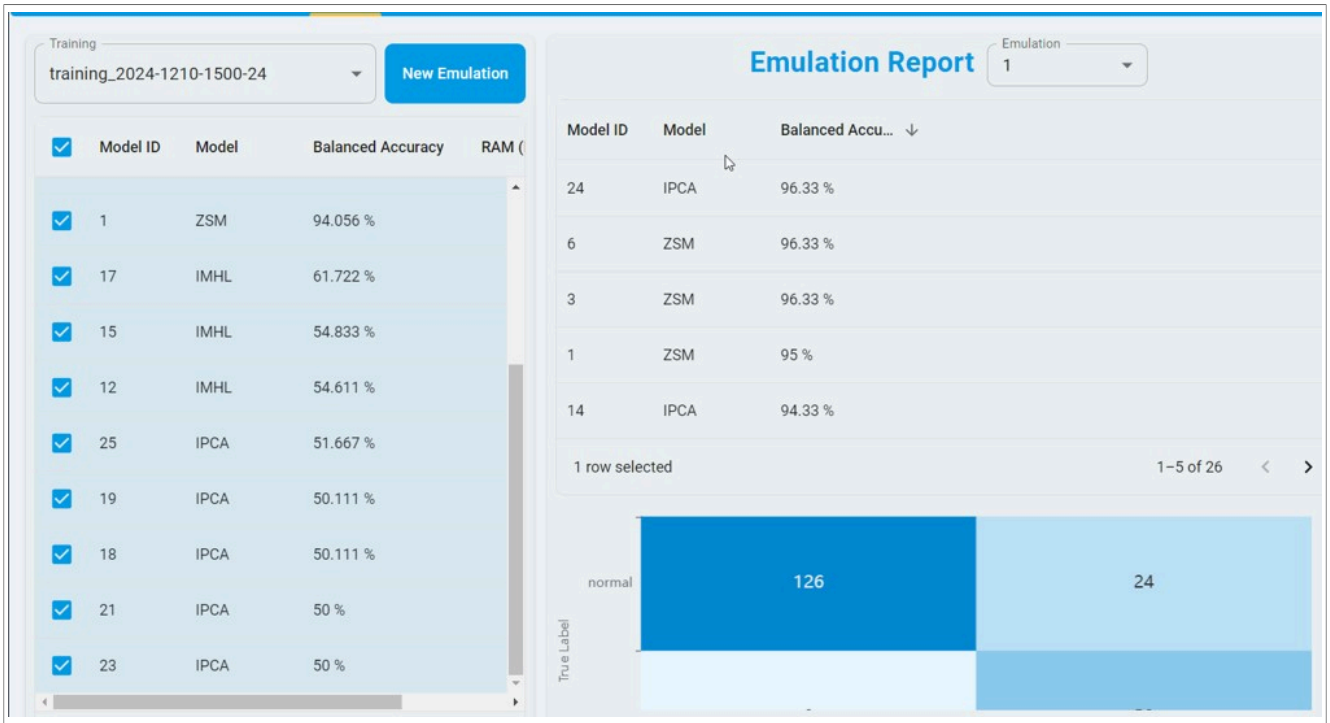

Figure 12. Emulation report

9 Deploy the model

To deploy the model:

1. Click in the **Deployment** window.
2. Select the best performance model.
   The example code of the model library appears in the right side, as shown in the following figure.

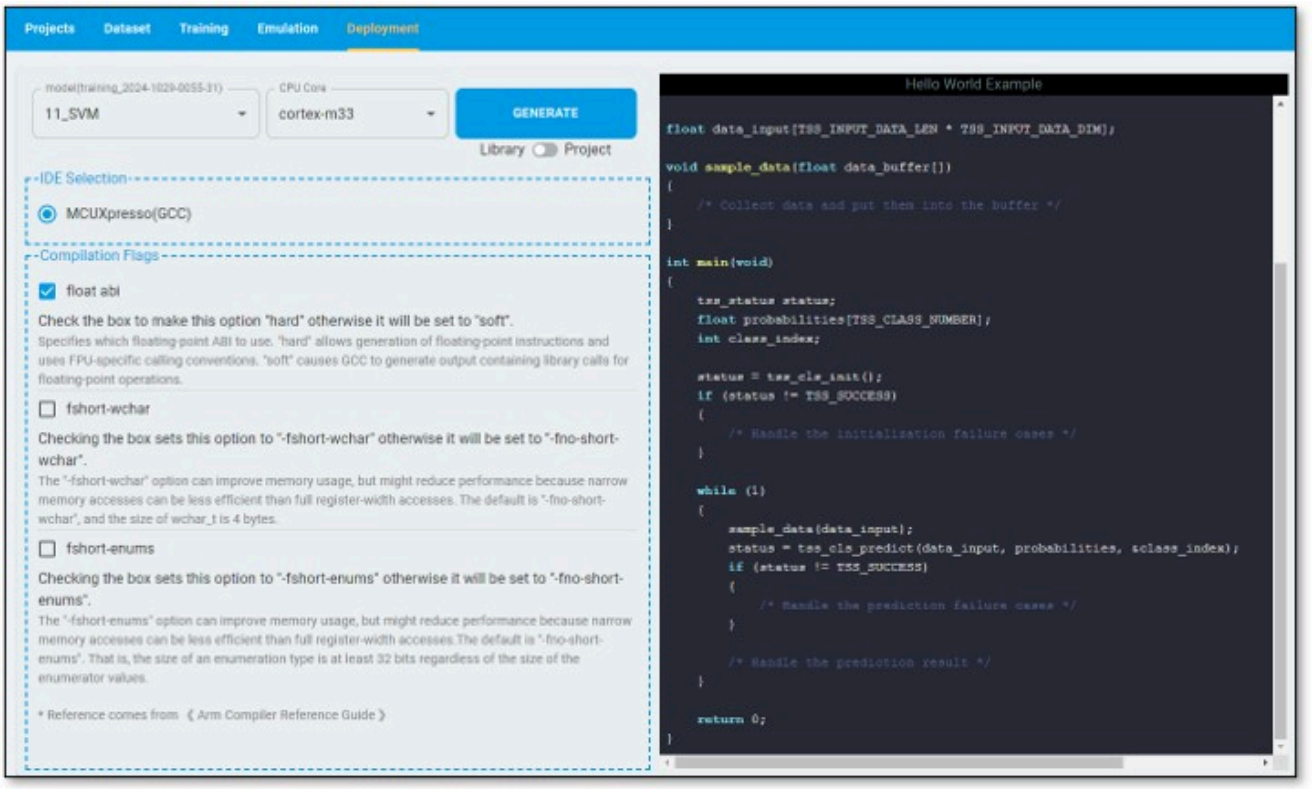

Figure 13. Example code of the model library

3. Click the **GENERATE** button.
4. Save the library zip file that contains the following four files.

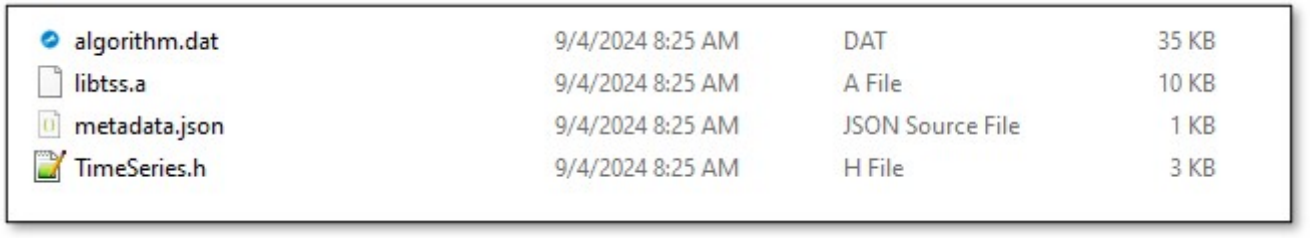

Figure 14. Files in the library zip file

- **algorithm.dat**: This encrypted file contains the algorithm details. NXP cloud server can parse and generate the source code.
- **TimeSeries.h**: This file is the API header for the `libtss.a`. The developer uses it for algorithm integration.
- **libtss.a**: This file is the core algorithm library. The developer uses it for algorithm integration.
- **metadata.json**: This file includes the meta description for the generated algorithm. It contains key information like compiler type, task type, input dataset and platform information, and also contains the minimum memory size as a reference.

5. Unzip the file and copy them into the `source\tss` of the project.
6. Compile and download the file onto the board.

10 Result

After flashing the board, reset the board, and the result appears on the LCD screen. Turn on the fan a switch from low speed to fast speed, the status of the fan must be normal as shown in the following figure.

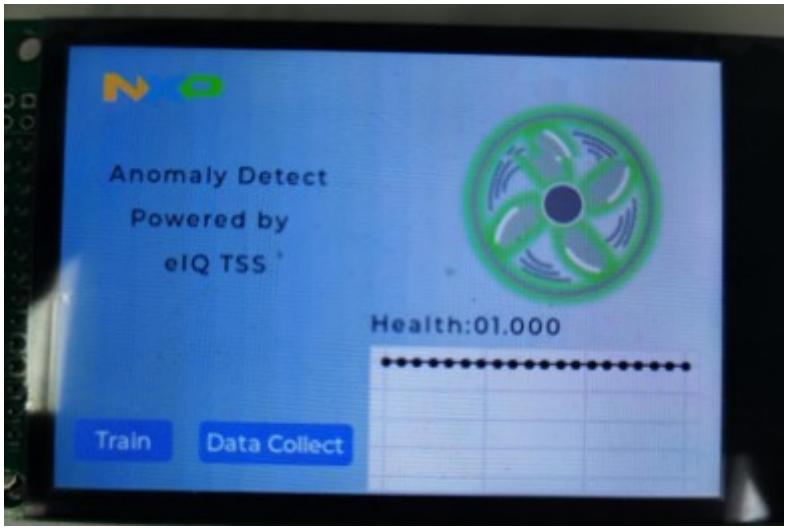

Figure 15. Status of the fan

10.1 Anomaly use case

- Knocking on the fan, detects an anomaly.
- Disrupting the operation of the fan blades, detects an anomaly.
- Turning off the Fan, detects an anomaly.

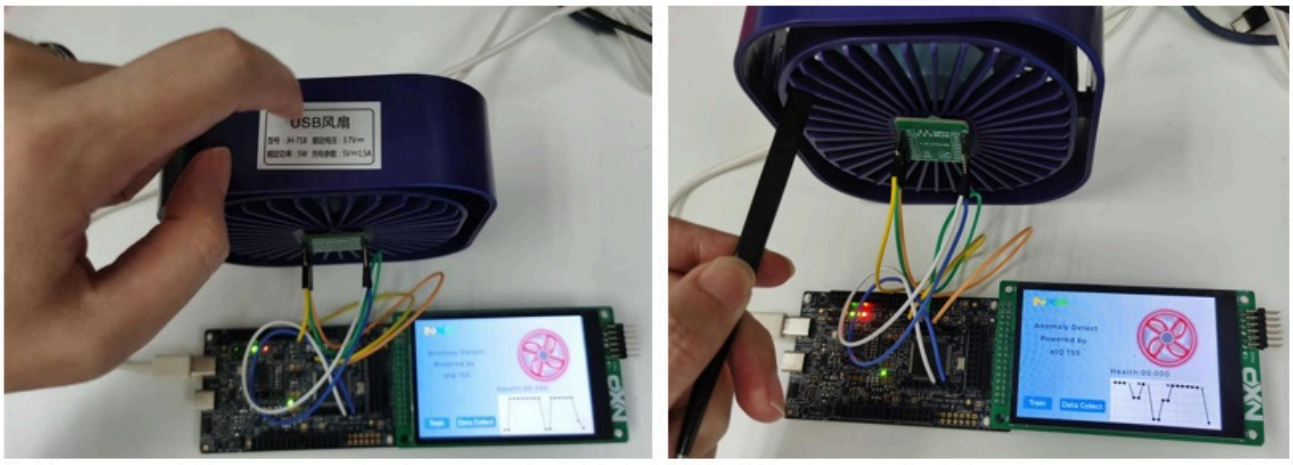

Figure 16. Figure 16

10.2 On-device learn

Adjust the mounting angle of the accelerator sensor. Notice that sometimes the sensor detects an anomaly.

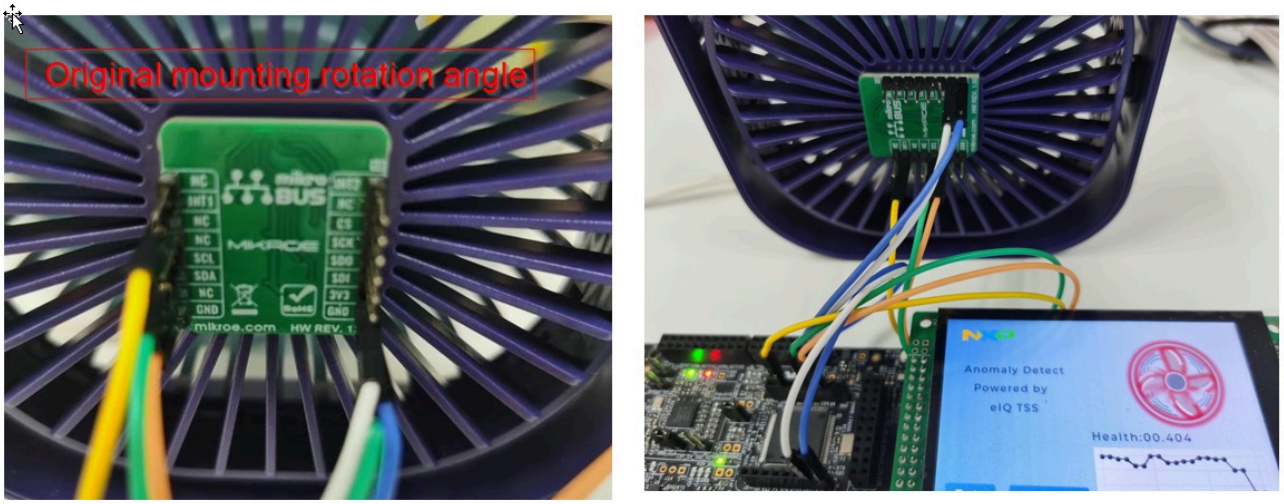

Figure 17. Figure 17

Click the Train button on the Screen, enter the trainer window, Click Start button to start the learning procedure. Return to the main window after the learning is complete. You can see the result on the screen.

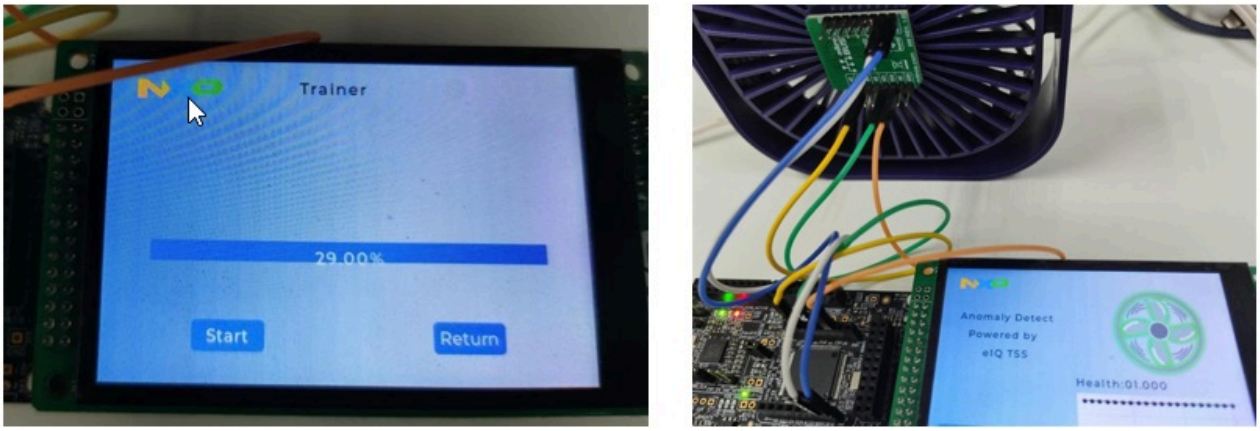

Figure 18. Figure 18

11 References

For further information, refer to the eIQ Time Series Studio documentation.

12 Revision history

The following table summarizes the changes since the initial release.

Table 1. Revision history

Document ID	Release date	Description
AN14549 v.1.0	10 February 2025	Initial public revision

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Figures

How to Use Time Series Studio to Deploy Anomaly Detection Model based on MCXA156

Contents