ModusToolbox™ Machine Learning Configurator user guide

ModusToolbox™ tools package version 3.0.0

Machine Learning Configurator version 2.0.0

About this document

Scope and purpose

The ModusToolbox™ Machine Learning (ML) Configurator is used in ML applications for adapting a pretrained learning model to an Infineon target platform. The tool accepts a pretrained ML model and generates an embedded model (as a library), which can be used along with your application code for a target device. The ModusToolbox™ ML Configurator also lets you fit the pretrained model of choice to the target device with a set of optimization parameters.

Intended audience

This document helps application developers understand how to use the ML Configurator as part of creating a ModusToolbox™ application.

Document conventions

Convention	Explanation
Bold	Emphasizes heading levels, column headings, menus and sub-menus
Italics	Denotes file names and paths.
`Courier New`	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, file/folder names, directories, command line inputs, code snippets
File > New	Indicates that a cascading sub-menu opens when you select a menu item

Abbreviations and definitions

The following define the abbreviations and terms used in this document:

- ML – Machine Learning
- NPZ – NumPy array in zipped format. When unzipped, it provides validation data as NumPy arrays used by the tool.
- MAE – Maximum Absolute Error
- MACC – Machine-learned ASAS Classification Catalog

Reference documents

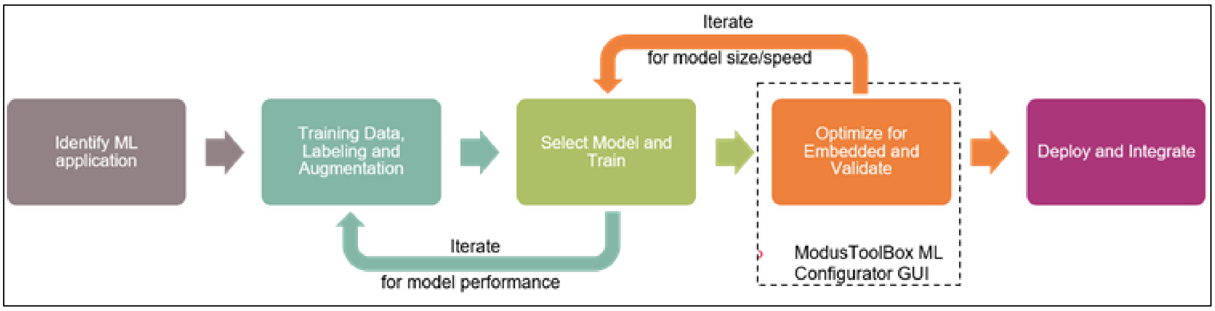
Refer to the following documents for more information as needed:

- ModusToolbox™ Machine Learning user guide
- Machine Learning Inference library readme file

Table of contents

1 Overview

The following shows the design flow for a typical application. The ModusToolbox™ ML Configurator GUI forms a vital part in fitting the model to the target platform.



The ModusToolbox™ ML Configurator is required to support the machine learning tool ecosystem. This tool forms the central asset that brings together other assets of the machine learning tool ecosystem, including Core tools, Inference engine, etc. as described in the ModusToolbox™ Machine Learning user guide. You can create a new application using the ML code example, or you can add the ML library to an existing application using the Library Manager.

1.1 Supported library

Name	Version	Link
Machine Learning Inference Engine library	2.0	https://github.com/Infineon/ml-inference

2 Launch the ML Configurator

There are numerous ways to launch the ML Configurator, and those ways depend on how you use the various tools in ModusToolbox™ software.

2.1 make command

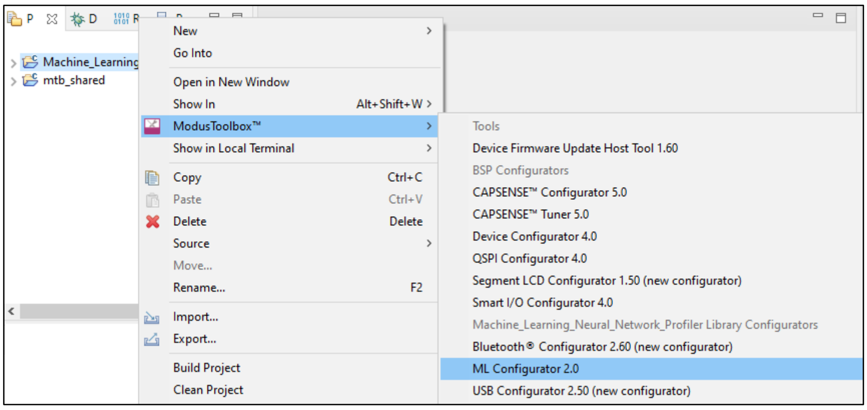
As described in the ModusToolbox™ tools package user guide build system chapter, you can run numerous make commands in the application directory, such as launching the ML Configurator. After you have created a ModusToolbox™ application, navigate to the application directory and type the following command in the appropriate bash terminal window:

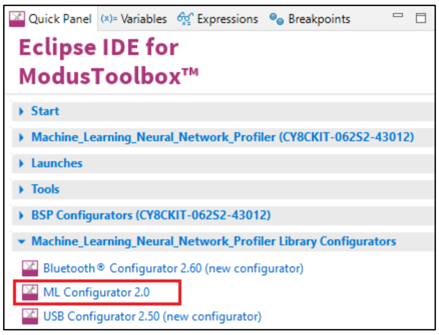
```
make ml-configurator
```

This command opens the ML Configurator GUI for the specific application in which you are working.

2.2 Eclipse IDE

If you use the Eclipse IDE for ModusToolbox™, you can launch the ML Configurator for the selected application. In the Project Explorer, right-click on the project and select **ModusToolbox™ > ML Configurator <version>**. You can also click the ML Configurator link in the IDE Quick Panel.




Quick Panel
(x)= Variables
Expressions
Breakpoints
Eclipse IDE for ModusToolbox™
Start
Machine_Learning_Neural_Network_Profiler (CY8CKIT-062S2-43012)
Launches
Tools
BSP Configurators (CY8CKIT-062S2-43012)
Machine_Learning_Neural_Network_Profiler Library Configurators
Bluetooth® Configurator 2.60 (new configurator)
ML Configurator 2.0
USB Configurator 2.50 (new configurator)

Similar to the make command method, launching the ML Configurator using the Eclipse IDE opens the tool for the selected application. Refer to the Eclipse IDE for ModusToolbox™ user guide for details about the IDE.

2.3 Executable (GUI)

If you don't have an application or if you just want to see what the configurator looks like, you can launch the ML Configurator GUI by running its executable as applicable for your operating system (for example, double-click it or select it using the Windows **Start** menu). By default, it is installed here:

<install_dir>/ModusToolbox/packs/ModusToolbox-Machine-Learning-Pack /tools/ml-configurator/

When launched this way, the ML Configurator opens without any settings configured. You can either open a specific configuration file or create a new one. See Menus for more information.

2.4 Executable (CLI)

The ML Configurator executable can be run from the command line, and it also has a "cli" version of the executable as well. Running configurator executables from the command line can be useful as part of batch files or shell scripts to re-generate the source code based on the latest configuration settings. The exit code for the executable is zero if the operation is successful, or non-zero if the operation encounters an error. For more information about the command-line options, run the executable using the `-h` option.

3 Quick start

This section provides a simple workflow for how to use the ML Configurator.

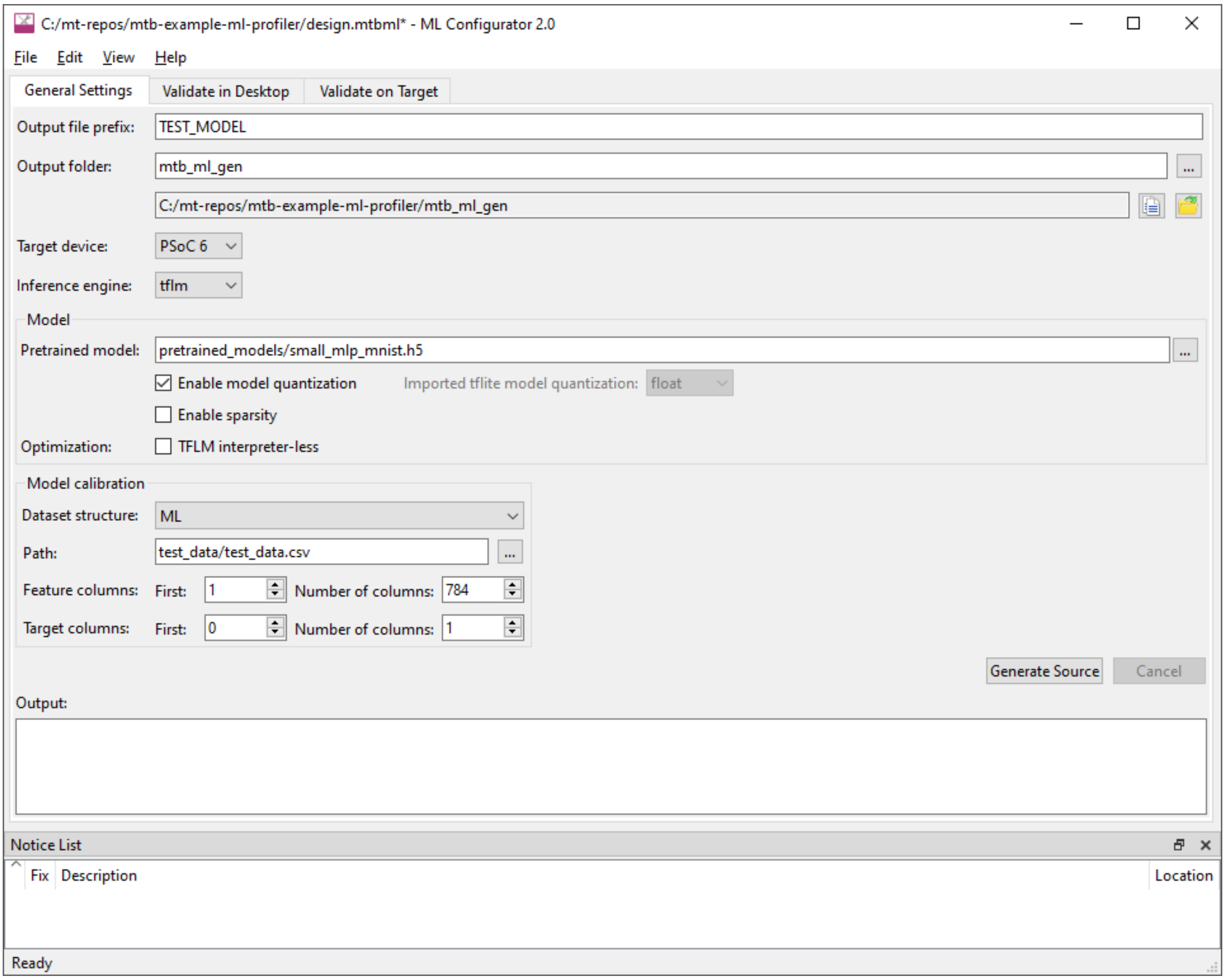
1. Create a new application based on a code example. For example:

   https://github.com/Infineon/mtb-example-ml-profiler

2. Launch the ML Configurator GUI.
3. Under **General Settings**, enter/select Project elements and Model elements.
4. Click **Generate Source** and specify the configuration (*.*mtbml*) file, as needed. See the Generate source button section.
5. Click the **Validate in Desktop** tab, and use the various setting to visualize the output data.
6. Click the **Validate on Target** tab if you have a device connected to your computer, and use the various setting to visualize the output data.

4 GUI description

The ML Configurator GUI contains menus, tabs, and various fields to generate data for the selected project.



C:/mt-repos/mtb-example-ml-profiler/design.mtbml* - ML Configurator 2.0

Output file prefix: TEST_MODEL

Output folder: mtb_ml_gen

C:/mt-repos/mtb-example-ml-profiler/mtb_ml_gen

Target device: PSoC 6

Inference engine: tflm

Model

Pretrained model: pretrained_models/small_mlp_mnist.h5

☑ Enable model quantization

Imported tflite model quantization: float

☐ Enable sparsity

Optimization: ☐ TFLM interpreter-less

Model calibration

Dataset structure: ML

Path: test_data/test_data.csv

Feature columns: First: 1 Number of columns: 784

Target columns: First: 0 Number of columns: 1

Output:

Notice List

Ready

4.1 Menus

4.1.1 File

- **New** – Creates a new file with new configuration.
- **Open** – Opens the configuration file.
- **Save** – Saves the existing file.
- **Save As** – Saves the existing file under a different name.
- **Open in System Explorer** – This opens your computer's file explorer tool to the folder that contains the **.mtbml* file.
- **Settings**
  - **Automatically generate code after save** – Select this check box to automatically generate code upon saving the configuration file. By default, this check box is not selected.
- **Exit** – Closes the configurator.

4.1.2 Edit

- **Undo** – Undoes the last action or sequence of actions.
- **Redo** – Redoes the last undone action or sequence of undone actions.

4.1.3 View

- **Notice List** – Hides or shows the Notice List pane. The pane is shown by default.
- **Toolbar** – Hides or shows the Toolbar.

4.1.4 Help

- **View Help** – Opens this document.
- **About ML Configurator** – Opens the About box for version information, as well as links to Infineon.com and the current session log file.

4.2 General Settings tab

4.2.1 Output file prefix

Provides the name for the output file. This name is used as the prefix for the generated files.

4.2.2 Output folder

This is the name and location where the generated files are placed, relative to the location of the saved configuration file.

4.2.3 Target device

Allows you to select the target device to use. Only PSoC™ 6 is available at this time.

4.2.4 Inference engine

This field allows you to select which inference engine to use. The two options are the Infineon Inference Engine and the TFLM Inference Engine.

4.2.5 Pretrained model

This parameter lets you select a pretrained model file from 3rd party ML frameworks.

Note: *For this version, the ML Configurator supports h5 and Tflite file formats (for IFX inference engine only h5 available).*

4.2.6 Enable model quantization

This check box is only available to select if the selected inference engine is TFLM and the Pretrained model is file format .h5. When you select this check box, the Model Calibration section becomes enabled to generate an 8-bit quantized model. If this option is not selected, then only a floating-point model is generated.

If the user does not import calibration data then an error message will show up in the notice list.

4.2.7 Imported tflite model quantization

This option is only available to select value if the selected inference engine is TFLM and the Pretrained model is file format .tflm. There are two possible values: float (by default) and int8x8. Select the value that corresponds with the quantization of the selected model.

Note: *If you select a quantization that does not correspond to model quantization, you will get error from ml-coretools.*

4.2.8 4.2.8 Enable sparsity

This check box enables the use of a memory-efficient packed format for any sparse weights present in the model. Available only for tflm inference engine.

4.2.9 4.2.9 Optimization

The section provides options based on the selected inference engine:

- ifx: Advanced scratch memory optimization
- tflm: TFLM interpreter-less

4.2.9.1 Advanced scratch memory optimization

This check box is selected by default. When selected, this option reduces the amount of scratch memory needed with minimal to no impact on the accuracy of the model.

4.2.9.2 TFLM interpreter-less

This check box is only available if the selected inference engine is TFLM. It is not selected by default.

Note: *Validation in Desktop is not supported for TFLM interpreter-less optimization.*

4.2.10 Model Calibration

This section allows you to import model calibration data when using the TFLM inference engine. It functions similar to the **Validate in Desktop** and **Validate on Target** tabs, but there is no option for random data. This section is only enabled if you select the Enable model quantization check box.

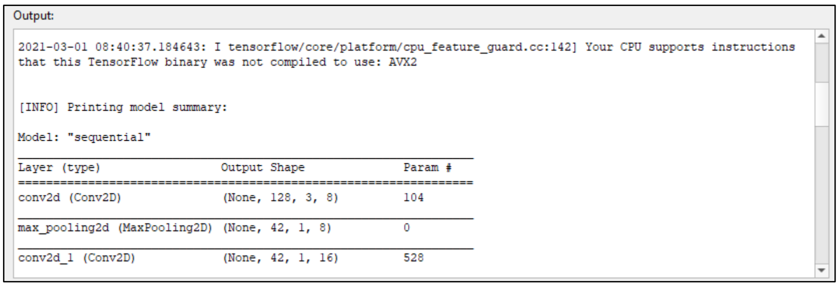
4.3 Generate source button

The main purpose of this button is to analyze and validate the pretrained model, as well as generate source data. The output of this analysis displays as output messages. The configurator also generates an output file in the specified Output file location.

When you click this button for the first time without an existing *.*mtbml* file , the configurator asks you to specify its file name and location. When you press this button on subsequent occasions after already specifying the *.mtbml file, the configurator skips this step and begins generating source data.

4.4 Output messages

The output messages section displays various messages about the model and the generation of source.

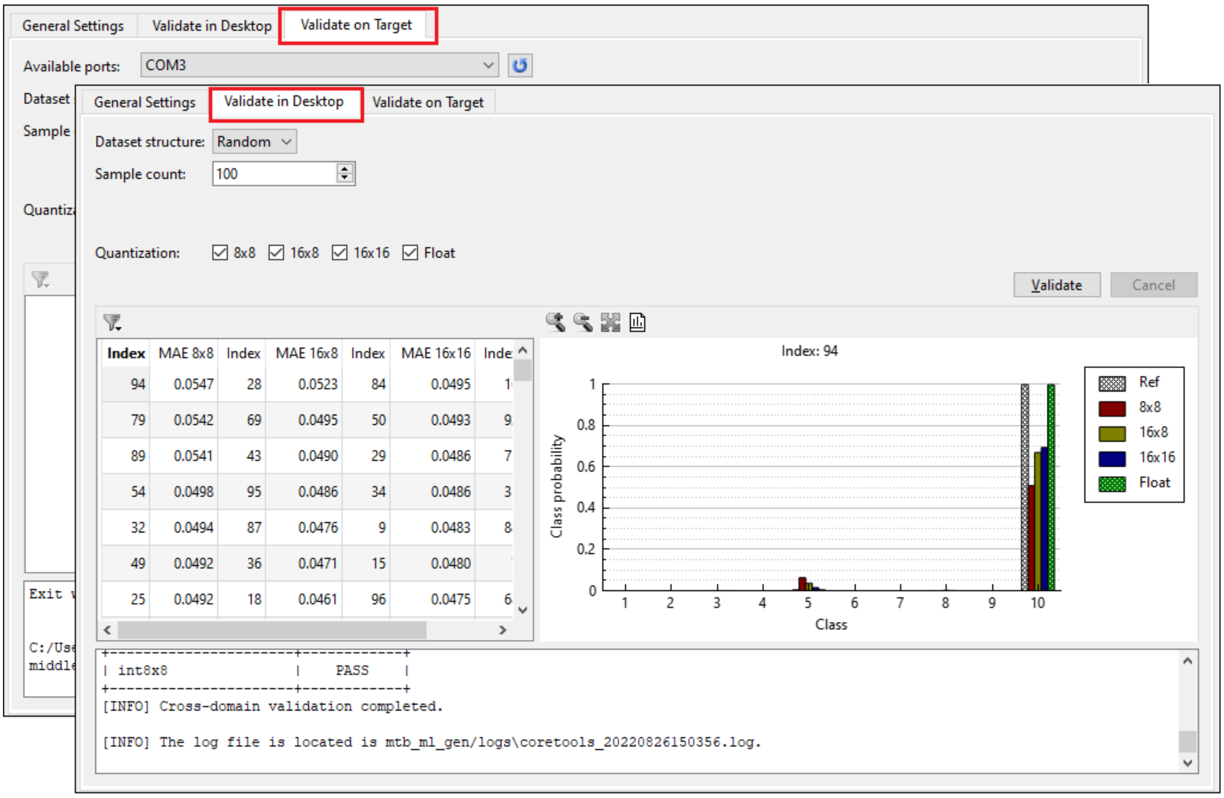


```
2021-03-01 08:40:37.184643: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions
that this TensorFlow binary was not compiled to use: AVX2


[INFO] Printing model summary:

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 128, 3, 8)         104
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 42, 1, 8)          0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 42, 1, 16)         528
```

4.5 Validate tabs

The ML Configurator contains two validate tabs: **Validate in Desktop** and **Validate on Target**. Both tabs provide controls to evaluate and visualize the output data. The only difference is that the Validate in Desktop tab allows you to run validation without being connected to a device. The Validate on Target tab requires a device be connected to the computer.

Note: *When using Validate on Target, ensure that you build the streaming validation firmware on the same host OS you are using to run the ML Configurator.*



Dataset structure: Random

Sample count: 100

Quantization: ☑ 8x8 ☑ 16x8 ☑ 16x16 ☑ Float

Index	MAE 8x8	Index	MAE 16x8	Index	MAE 16x16	Inde
94	0.0547	28	0.0523	84	0.0495	1
79	0.0542	69	0.0495	50	0.0493	9
89	0.0541	43	0.0490	29	0.0486	7
54	0.0498	95	0.0486	34	0.0486	3
32	0.0494	87	0.0476	9	0.0483	8
49	0.0492	36	0.0471	15	0.0480	
25	0.0492	18	0.0461	96	0.0475	6

```
+-----------------------+------------+
| int8x8                |    PASS    |
+-----------------------+------------+
[INFO] Cross-domain validation completed.

[INFO] The log file is located is mtb_ml_gen/logs\coretools_20220826150356.log.
```

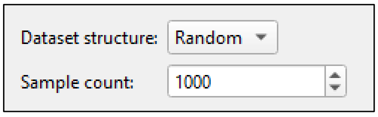
4.5.1 COM port

This pull-down menu lets you specify the COM port that connects to the target device when using the **Validate on Target** tab.

4.5.2 Dataset structure

This pull-down lets you select the type of data to use as validation input. This can be random data generated by the ML Configurator or an input file that is typically a data capture from an actual application source. Options include:

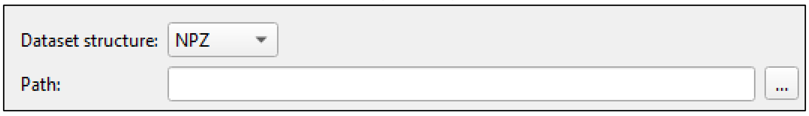
- **Random** – Random data generated by the configurator.



Dataset structure: Random

Sample count: 1000

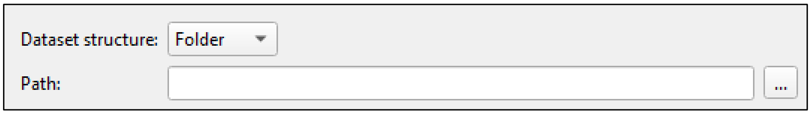
- **NPZ** – Validation data as NumPy arrays in zipped format.



Dataset structure: NPZ

Path:

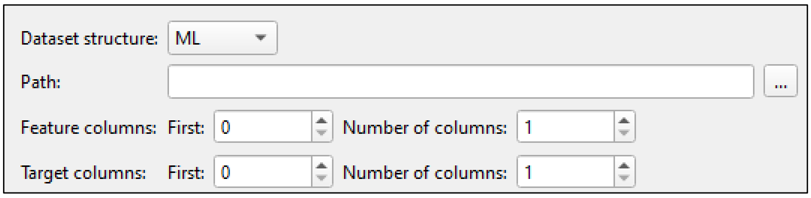
- **Folder** – Validation data as folders containing JPEG files.



Dataset structure: Folder

Path:

- **ML** – ML Format: A CSV file used by ml-coretools that contains only numeric data, with no header and no sample ID columns.



Dataset structure: ML

Path:

Feature columns: First: 0 Number of columns: 1

Target columns: First: 0 Number of columns: 1

4.5.3 Sample count

If you set the **Dataset structure** field to "Random," this field displays to specify the number of rows of data to count for the sample.

4.5.4 Path

If you select a **Dataset structure** other than "Random," this field displays to specify the file name and location to use for data validation. When **Dataset structure** is set to "Folder," the path should specify a folder location.

4.5.5 Feature columns/number of columns

If you set the **Dataset structure** field to "ML," this field specifies the first and number of feature columns. The first column is 0.

4.5.6 Target columns/number of columns

If you set the **Dataset structure** field to "ML," this field specifies the first and number of target columns. The first column is 0.

4.5.7 Quantization

These check boxes let you chose 8-bit, 16-bit, and/or float quantization options for the validation results to display in the table and graph, as well as for one or more output files to be generated. For the **Validate in Desktop** option, you can select multiple options; for the **Validate on Target** option, you can only select one.

4.5.8 Validate button

This button generates the validation result based on the selected data input and the model.

4.5.9 Cancel button

This button stops the validation process. Available only in running process.

4.5.10 Table

The Validation dialog contains a table that displays the validation result data with the input Index and its corresponding MAE value. The table may have up to three columns, depending on the selected quantization and filter options.

- Each of the table column headings provides a sorting option to view the data in ascending or descending order.
- A **Filter** button above the table allows you to select and deselect the quantization options, and thus show or hide the corresponding table.
- When you select each of the Indexes and their corresponding MAE value, the data displays in graph form on the right side of the dialog.

4.5.11 Graph

The graph is a simple representation of the data to allow for easy viewing. The graph includes several buttons as follows:

- Zoom in – Make the graph bigger.
- Zoom out – Make the graph smaller.
- Zoom to fit – Grow or shrink the graph to fit the size of the graph area.
- Export to file – Save the graph as currently presented to a file. Formats include png, jpg, and bmp.

You can also click and drag your mouse to pan the graph in any direction.

4.5.12 Results

This area displays the analysis results, which includes memory usage and usage of MACCs, if these exist in hardware.

4.5.13 Notice List

Shows errors and info messages. This must show no errors before running **Generate Sources** or **Validate**.

4.6 Status bar

The status bar displays various information about the status of the tool.

5 Version changes

This section lists and describes the changes for each version of this tool.

Version	Change Descriptions
1.0	New tool.
1.10	Added **Open in System Explorer** in **File** menu. Changed how the target/feature columns are specified in Validation dialog.
1.20	Added **Edit** menu and **Undo/Redo** commands. Added **Advanced scratch memory optimization** check box. Added support for Validate on Target.
2.0	Changed GUI to use different validation tabs instead of separate dialogs. Removed Validate in Desktop and Validate on Target buttons. Added tflm inference engine. KERAS renamed to ifx. Added support *.tflite models (only for tflm inference engine). For tflm inference engine: • Only float quantization available (8x8 if calibration data presented). • Enable model quantization check box. • Enable sparsity check. • TFLM interpreter-less check. Added Model calibration section. Added notice list for show errors and info. Added cancel button for each tab.

Revision history

Revision	Date	Description
**	2021-03-16	New document.
*A	2021-04-30	Updated to version 1.10.
*B	2021-09-08	Updated to version 1.20.
*C	2022-10-10	Updated to version 2.0.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2022-10-10

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

www.cypress.com/support

Document reference

002-32756 Rev. *C

IMPORTANT NOTICE
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics („Beschaffenheitsgarantie") .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.