EDGE IMPULSE

Next-generation Digital Health:

Automating Machine Learning (ML) Workflows in Digital Health with Edge Impulse

Table of Contents

Introduction

Edge Impulse is a cloud based machine learning platform for developing edge AI models for any hardware, from embedded MCUs to powerful CPUs and GPUs. It's low- to no-code interface provides seamless integration of the machine learning workflow from data collection to model optimization to hardware profiling and deployment. Edge Impulse (noted as EI throughout this document) offers a comprehensive solution that caters to various stages of a typical machine learning (ML) pipeline. It resembles the toolkit of a skilled architect designed for building skyscrapers. Just as an architect starts with a foundation, builds upwards, adjusts to challenges, and integrates new designs and technologies over time, EI provides the tools to start, adapt, and refine ML solutions, ensuring they're robust, current, and optimized for your data.

With a focus on data collection, cleaning, feature extraction, model training, testing, and deployment, Edge Impulse ensures that users have access to the necessary tools at each step. By seamlessly connecting these components through a unified software platform, users can efficiently manage the entire ML workflow from data ingestion to model deployment. Beyond its integrated approach, Edge Impulse provides APIs and SDKs, enabling users to customize their workflows and integrate with external tools when needed. Moreover, Edge Impulse acknowledges the importance of collaboration, incorporating collaboration tools while maintaining a strong focus on security and compliance, particularly relevant for teams working in the digital health industry. With these features, Edge Impulse empowers both beginners and experts to build and deploy ML models for the edge effectively.

In the upcoming sections of this guide, we will delve deeper into the features and capabilities that Edge Impulse offers, to help you understand and gain insights on how to use it effectively for your digital health ML applications.

Overview

This tutorial serves as a comprehensive guide for transforming your digital health Machine Learning (ML) applications, taking advantage of the enterprise-grade capabilities of the Edge Impulse (EI) platform.

We'll guide you step-by-step through the processes of collecting, integrating, transforming, visualizing, and engineering a time-series dataset. To round things off, we'll show you how to train, optimize, and deploy your trained model on the hardware of your choice!

Each segment provides an overview of the required EI components, which are highlighted with this icon. This will help you comprehend what these elements are, why they are important, and how they fit in the Edge AI platform pipeline that we are building.

To aid understanding, each part of this guide is supplemented with a diagram, demonstrating the flow of inputs and outputs through these elements. We also provide screenshots demonstrating the configuration steps along with concise explanations within each section.

Plus, for an in-depth understanding, we've provided links to our comprehensive documentation.

About the Dataset

Collecting information through brainwave analysis is a prevalent and crucial application in the digital healthcare sector. In this guide, the dataset includes EEG signal data from 10 college students, obtained while they watched a series of 20 MOOC video clips, categorized into non-confusing (basic algebra or geometry) and potentially confusing topics (Quantum Mechanics, Stem Cell Research). Each participant wore a single-channel wireless MindSet, measuring activity over the frontal lobe, and after viewing, they rated their confusion on a 1-7 scale. These subjective ratings were normalized to create binary confusion labels - 0 (not confused) and 1 (confused).

This transformed dataset is then stored into a new dataset called "HealWell_Customer_Dataset" and later fed into a project called "HealWell_customer_Project" to train the model and make predictions on unseen data.

Let's begin by exploring ways to collect and organize your data using Edge Impulse.

🔄 Here is a typical high level ML workflow in EI.

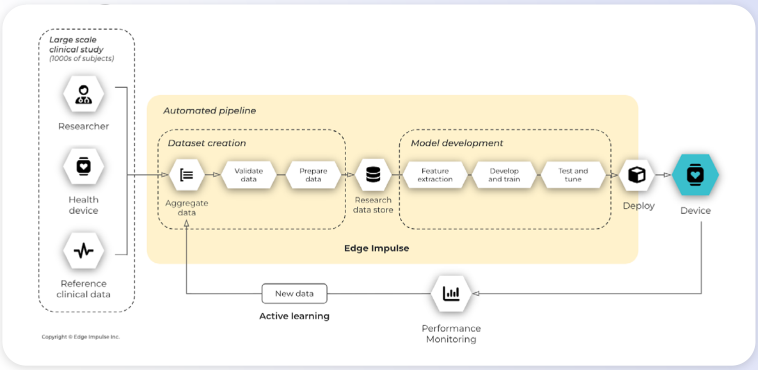


Diagram 1: High level ML workflow in EI

1. Collecting and Organizing Data

The process of creating a machine learning model begins with data. This data spans across wearable devices, electronic health records, medical imaging systems, patient surveys etc. Each data source presents its own unique data format, structure, and challenges.

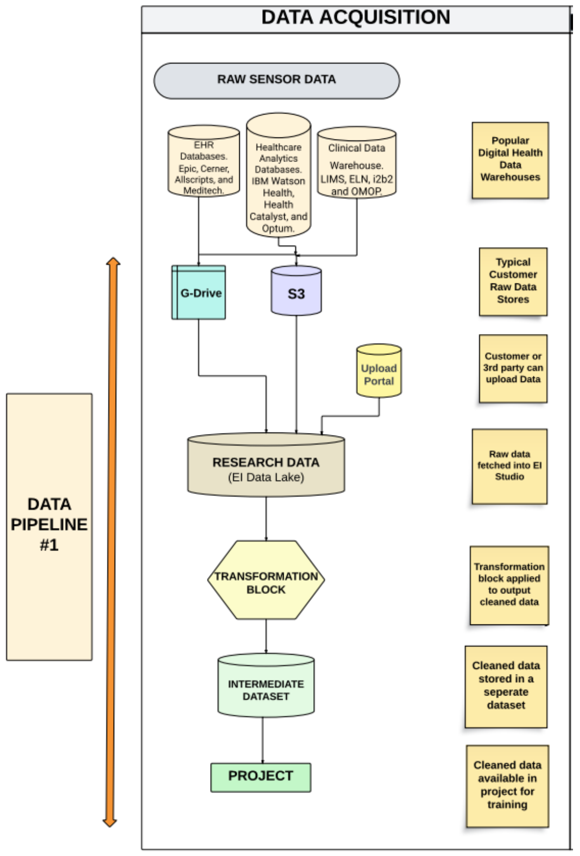
Edge Impulse's **Data** component (part of the data acquisition pipeline) is designed to ingest data from a variety of sources. This facilitates the collection of diverse, real-world data needed to train robust models.

Additionally, users can collect and store training and test data alongside their model and deployment code. Rather than relying on prebuilt datasets or requiring users to construct their own data gathering technology, Edge Impulse offers a variety of methods to gather data in real-world environments.

Diagram 2: Typical Data acquisition workflow

[Data acquisition]

Note: "Research Data" in the screenshot below refers to the "Data" tab within EI UI.



DATA ACQUISITION
RAW SENSOR DATA
EHR Databases. Epic, Cerner, Allscripts, and Meditech.
Healthcare Analytics Databases. IBM Watson Health, Health Catalyst, and Optum.
Clinical Data Warehouse. LIMS, ELN, i2b2 and OMOP.
Popular Digital Health Data Warehouses
G-Drive
S3
Typical Customer Raw Data Stores
Upload Portal
Customer or 3rd party can upload Data
RESEARCH DATA (EI Data Lake)
Raw data fetched into EI Studio
DATA PIPELINE #1
TRANSFORMATION BLOCK
Transformation block applied to output cleaned data
INTERMEDIATE DATASET
Cleaned data stored in a seperate dataset
PROJECT
Cleaned data available in project for training

Data

Often, data scientists and ML engineers like to work with a slice of the unprocessed data for preliminary tasks such as data quality checks and experimental trials. The “Data” component within EI organization stores the raw incoming data from your s3 bucket or a 3rd party research partner in a single location within EI. This data lake gives data scientists and ML engineers the freedom to segment the raw data as they please, using a query similar to SQL.

Data from your s3 bucket and/or data from a 3rd party research partner uploaded to an EI hosted bucket (a.k.a Upload Portal) feeds into a “dataset” within “Data” in your EI organization.

Upon preprocessing, semi-clean or clean data is outputted. The semi-clean data can be stored in a new or existing dataset for further cleaning or fed into a project for data exploration and/or conducting preliminary experiments. Alternatively, clean data can be fed into a project for model training.

“Data” is located under the “Organization” tab within your EI account.
Follow the steps below to configure the data sources.

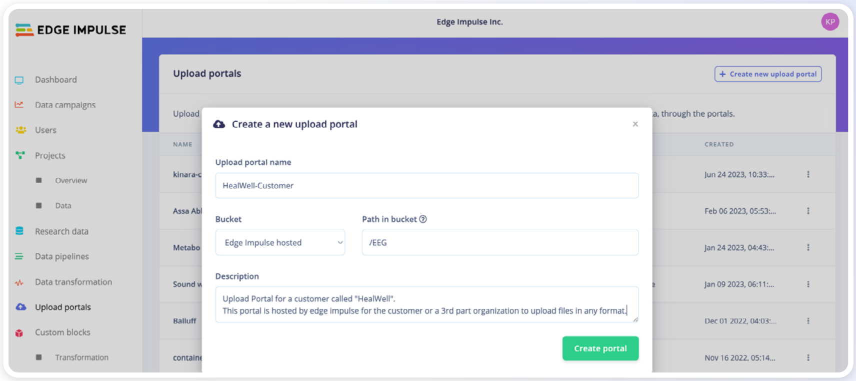
Upload Portal

Upload portals provide a secure way to let external parties upload data to your s3 bucket. Through an upload portal they get an easy user interface to add data, but they have no access to the content of the dataset, nor can they delete any files. Upload Portals offer a friendly user interface and does not require a technical person to use it, thus providing an easy way to use the data directly in Edge Impulse.

[Upload Portals]

Option 1: Upload your data to Edge Impulse hosted bucket (a.k.a Upload Portal)

Step 1: Create a new upload portal



Create a new upload portal

Upload portal name: HealWell-Customer

Bucket: Edge Impulse hosted

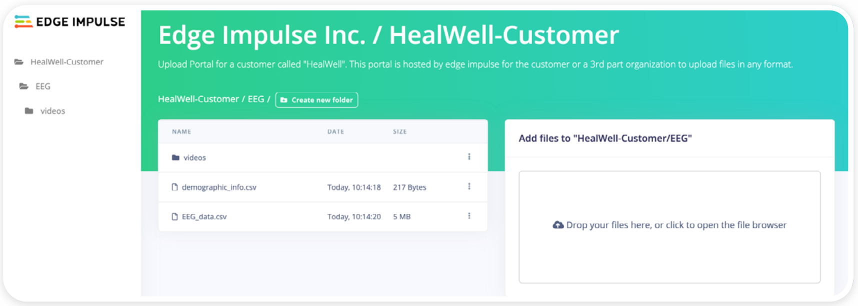
Path in bucket: /EEG

Description: Upload Portal for a customer called "HealWell".
This portal is hosted by edge impulse for the customer or a 3rd part organization to upload files in any format.

NAME	CREATED
kinara-c	Jun 24 2023, 10:33:...
Assa Abl	Feb 06 2023, 05:53:...
Metabo	Jan 24 2023, 04:43:...
Sound w	Jan 09 2023, 06:11:...
Balluff	Dec 01 2022, 04:03:...
containe	Nov 16 2022, 05:14...

Step 2: Drag and drop your files

Sample dataset with details on the use case of the dataset are available [here](#).



Edge Impulse Inc. / HealWell-Customer

Upload Portal for a customer called "HealWell". This portal is hosted by edge impulse for the customer or a 3rd part organization to upload files in any format.

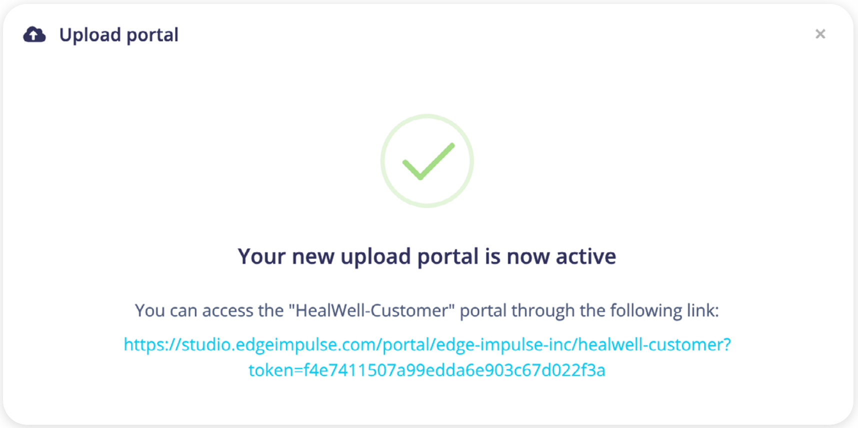
HealWell-Customer / EEG / Create new folder

NAME	DATE	SIZE
videos		
demographic_info.csv	Today, 10:14:18	217 Bytes
EEG_data.csv	Today, 10:14:20	5 MB

Add files to "HealWell-Customer/EEG"

Drop your files here, or click to open the file browser

Step 3: Follow the link to view your upload portal



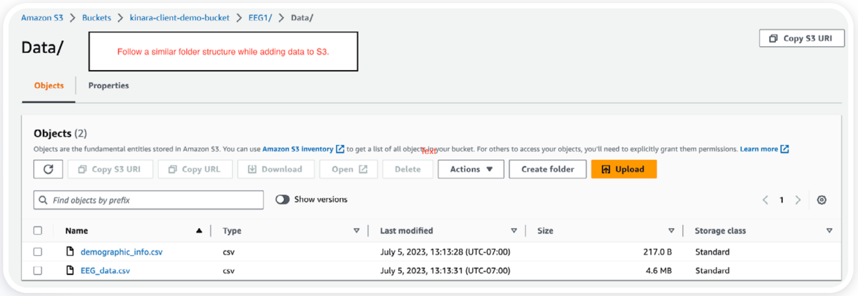
Upload portal

Your new upload portal is now active

You can access the "HealWell-Customer" portal through the following link:

https://studio.edgeimpulse.com/portal/edge-impulse-inc/healwell-customer?token=f4e7411507a99edda6e903c67d022f3a

Option 2: Upload your data to S3/GCP or S3 compatible bucket



Data/

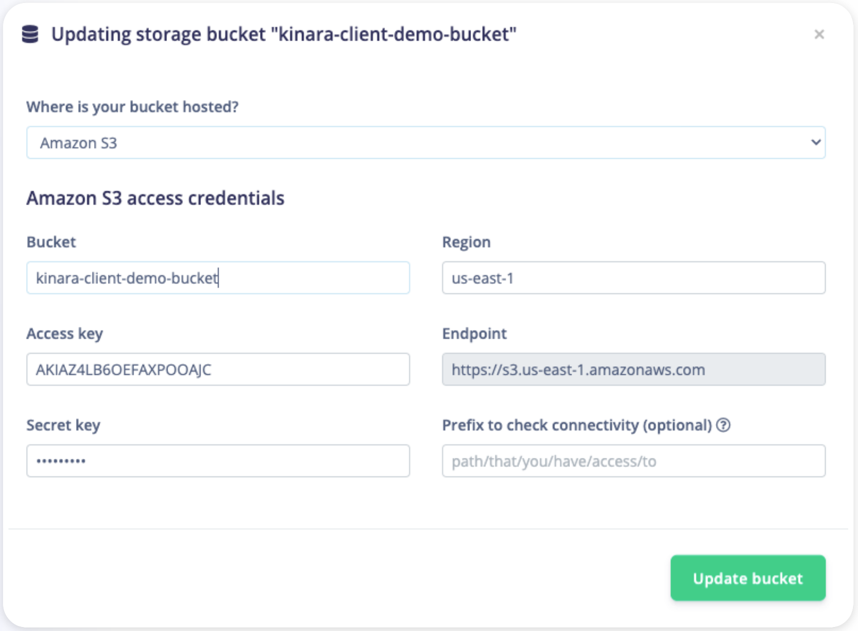
Follow a similar folder structure while adding data to S3.

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
demographic_info.csv	csv	July 5, 2023, 13:13:28 (UTC-07:00)	217.0 B	Standard
EEG_data.csv	csv	July 5, 2023, 13:13:31 (UTC-07:00)	4.6 MB	Standard

Go to “Data” → Buckets and add a new bucket



Updating storage bucket "kinara-client-demo-bucket"

Where is your bucket hosted?
Amazon S3

Amazon S3 access credentials

Bucket: kinara-client-demo-bucket	Region: us-east-1
Access key: AKIAZ4LB6OEFAXPOOAJC	Endpoint: https://s3.us-east-1.amazonaws.com
Secret key: •••••••••	Prefix to check connectivity (optional): path/that/you/have/access/to

2. Data Cleaning and Transformation

Transformation Blocks

Raw data often arrives laden with errors, inconsistencies, and missing values, making it a daunting task to ensure data integrity. These errors are often detected at a later stage, wasting precious time and resources invested during data collection and data transfer.

To address these issues, we'll use the *"Transformation blocks"* feature of EI to implement comprehensive custom *data cleaning processes, validate data accuracy* and much more. By utilizing the power of parallelization, to ensure rapid discovery of potential errors, right at the source, these custom blocks save valuable time and resources.

Transformation blocks take raw data from the previous section and transform the data into a different dataset or files that can be loaded in an Edge Impulse project. In this tutorial, we use Transformation blocks to *load data from the csv files, merge dataframes, standardize the features, save the resulting data frame in EI compatible format etc.*

Transformation blocks are located under the "Organization" tab in your EI account.
Follow the steps below to configure one as you progress through this tutorial.

Additional documentation

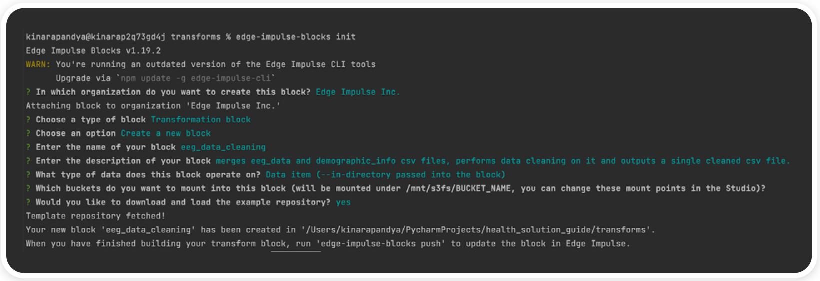
Step 1: Pre-process unclean raw data by pushing the pre-processing code to Edge Impulse in form of a Transformation Block

Prerequisite: Install **Docker** depending on your OS.

Open a new terminal and follow the steps in the screenshot below. You may replace the sample code within "transform.py" with your own custom code or clone **this** repository to use the same dataset and code within "transform.py".

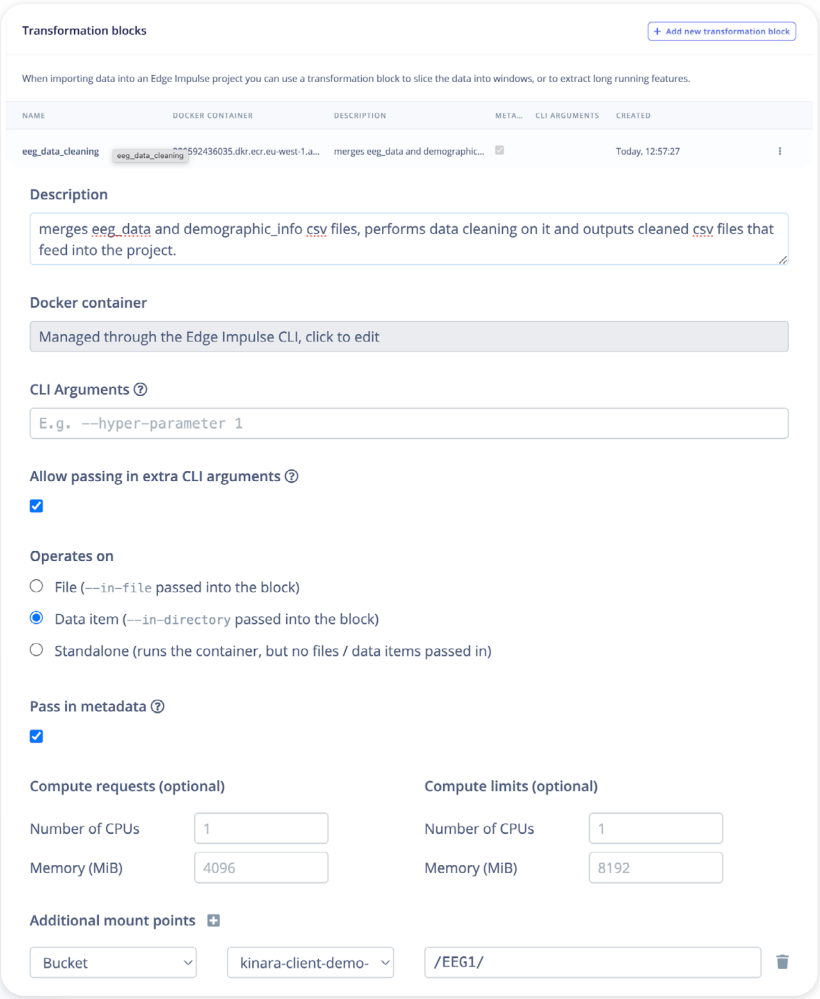
Next, push your new block to EI using command *"edge-impulse-blocks push"*

Example custom transformation block code - [Github]



```
kinarapandya@kinarap2q73gd4j transforms % edge-impulse-blocks init
Edge Impulse Blocks v1.19.2
WARN: You're running an outdated version of the Edge Impulse CLI tools
      Upgrade via `npm update -g edge-impulse-cli`
? In which organization do you want to create this block? Edge Impulse Inc.
Attaching block to organization 'Edge Impulse Inc.'
? Choose a type of block Transformation block
? Choose an option Create a new block
? Enter the name of your block eeg_data_cleaning
? Enter the description of your block merges eeg_data and demographic_info csv files, performs data cleaning on it and outputs a single cleaned csv file.
? What type of data does this block operate on? Data item (--in-directory passed into the block)
? Which buckets do you want to mount into this block (will be mounted under /mnt/s3fs/BUCKET_NAME, you can change these mount points in the Studio)?
? Would you like to download and load the example repository? yes
Template repository fetched!
Your new block 'eeg_data_cleaning' has been created in '/Users/kinarapandya/PycharmProjects/health_solution_guide/transforms'.
When you have finished building your transform block, run 'edge-impulse-blocks push' to update the block in Edge Impulse.
```

Step 2: Navigate to Custom blocks tab to view the pushed block in your Edge Impulse organization



Transformation blocks

When importing data into an Edge Impulse project you can use a transformation block to slice the data into windows, or to extract long running features.

NAME	DOCKER CONTAINER	DESCRIPTION	META...	CLI ARGUMENTS	CREATED
eeg_data_cleaning	...592436035.dkr.ecr.eu-west-1.a...	merges eeg_data and demographic...	✓		Today, 12:57:27

Description

merges eeg_data and demographic_info csv files, performs data cleaning on it and outputs cleaned csv files that feed into the project.

Docker container

Managed through the Edge Impulse CLI, click to edit

CLI Arguments

E.g. --hyper-parameter 1

Allow passing in extra CLI arguments

✓

Operates on

- File (--in-file passed into the block)
- [✓ Data item (--in-directory passed into the block)]
- Standalone (runs the container, but no files / data items passed in)

Pass in metadata

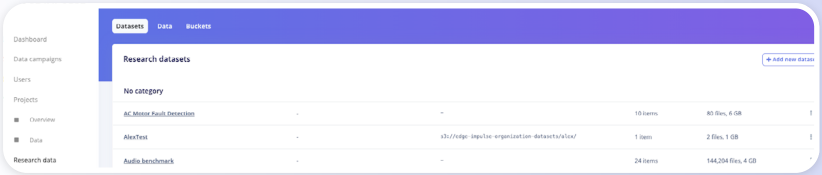
✓

Compute requests (optional)		Compute limits (optional)	
Number of CPUs	1	Number of CPUs	1
Memory (MiB)	4096	Memory (MiB)	8192

Additional mount points

Bucket	kinara-client-demo-	/EEG1/

Step 3: Under your organization, Navigate to “Data” and create new dataset

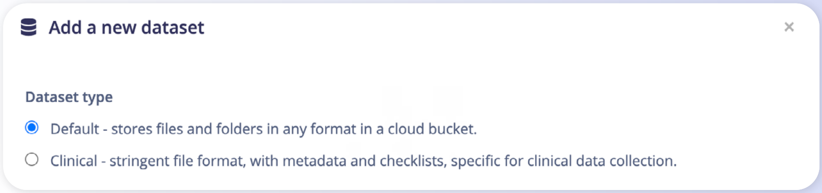


Research datasets

No category

AC Motor Fault Detection	-	-	10 items	80 files, 6 GB
AlexTest	-	s3://edge-impulse-organization-datasets/alex/	1 item	2 files, 1 GB
Audio benchmark	-	-	24 items	144,204 files, 4 GB

Select dataset type as “default”. In future revisions, we will go deeper into “clinical” datasets.



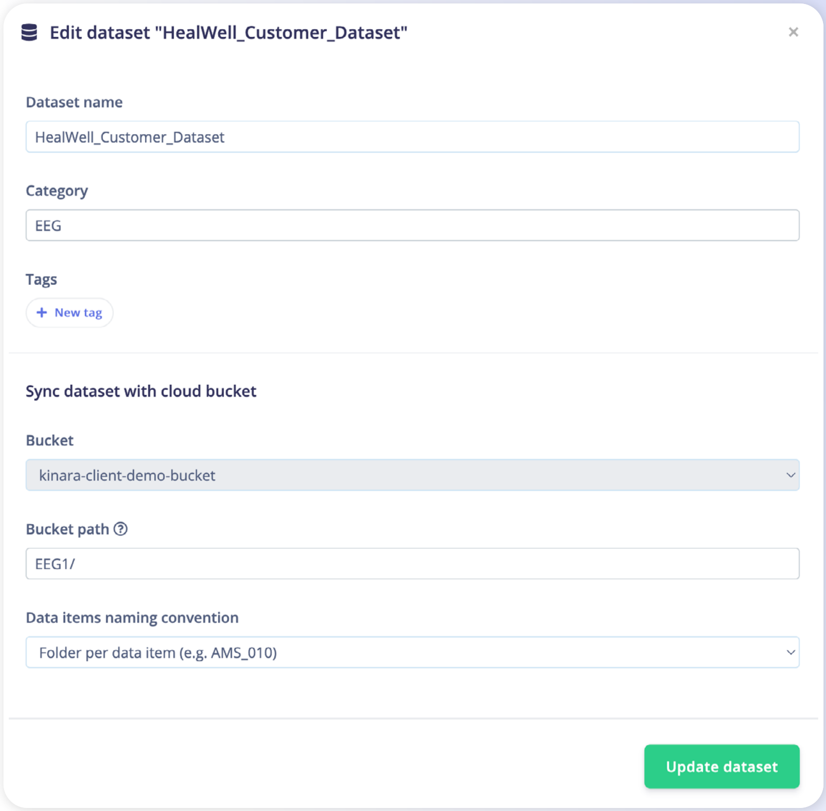
Add a new dataset

Dataset type

(●) Default - stores files and folders in any format in a cloud bucket.

( ) Clinical - stringent file format, with metadata and checklists, specific for clinical data collection.

Step 4: Mount the S3 bucket or mount the Upload portal wherever your raw data resides



Edit dataset "HealWell_Customer_Dataset"

Dataset name: HealWell_Customer_Dataset

Category: EEG

Tags

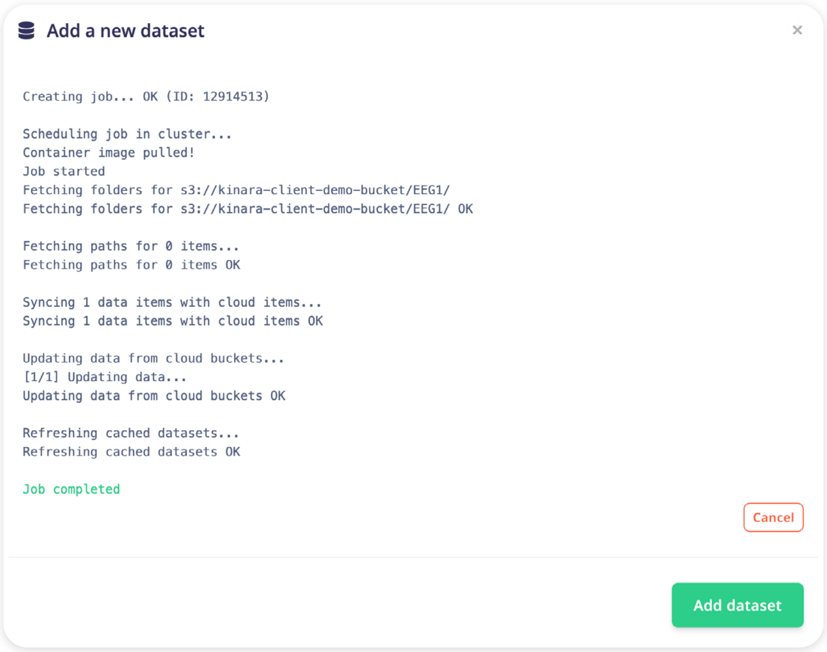
Sync dataset with cloud bucket

Bucket: kinara-client-demo-bucket

Bucket path: EEG1/

Data items naming convention: Folder per data item (e.g. AMS_010)

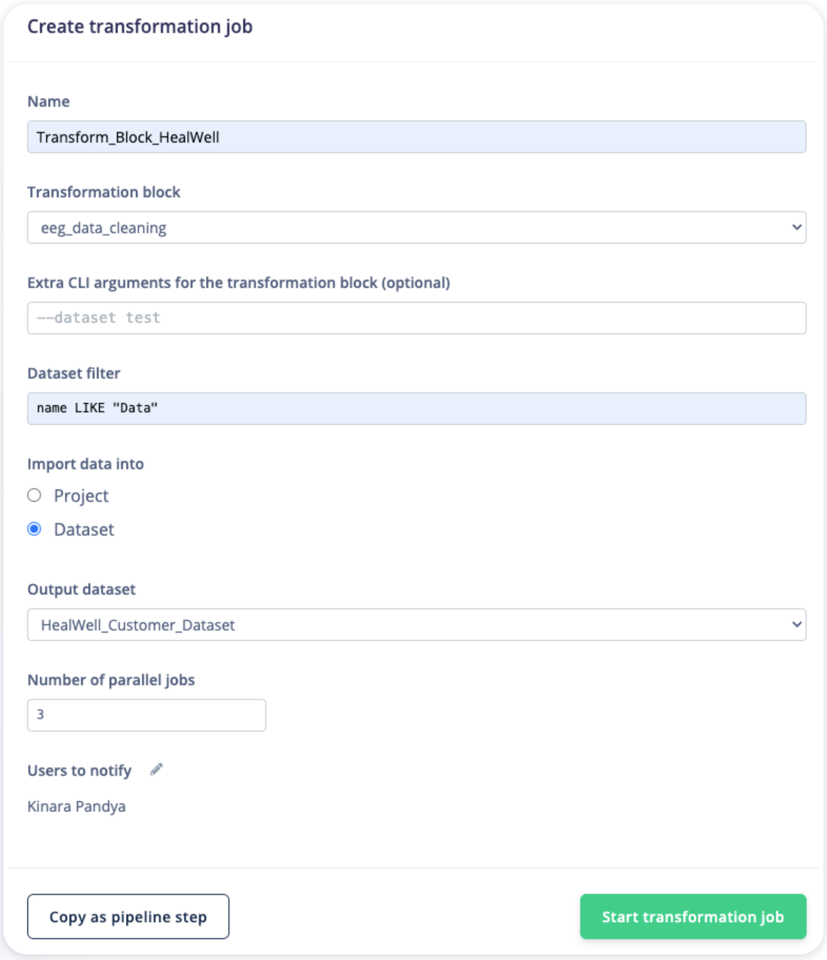
You should be able to see a similar output as shown below. Here, it is fetching data from the s3 bucket.



```
Creating job... OK (ID: 12914513)

Scheduling job in cluster...
Container image pulled!
Job started
Fetching folders for s3://kinara-client-demo-bucket/EEG1/
Fetching folders for s3://kinara-client-demo-bucket/EEG1/ OK

Fetching paths for 0 items...
Fetching paths for 0 items OK

Syncing 1 data items with cloud items...
Syncing 1 data items with cloud items OK

Updating data from cloud buckets...
[1/1] Updating data...
Updating data from cloud buckets OK

Refreshing cached datasets...
Refreshing cached datasets OK

Job completed
```

Step 5: Under your organization, navigate to "Data Transformation" tab and create a job

This step creates a transformation job to save the transformed cleaned data into a new dataset called "HealWell customer dataset" created in the previous step.



Name: Transform_Block_HealWell

Transformation block: eeg_data_cleaning

Extra CLI arguments for the transformation block (optional): --dataset test

Dataset filter: name LIKE "Data"

Import data into: ○ Project ● Dataset

Output dataset: HealWell_Customer_Dataset

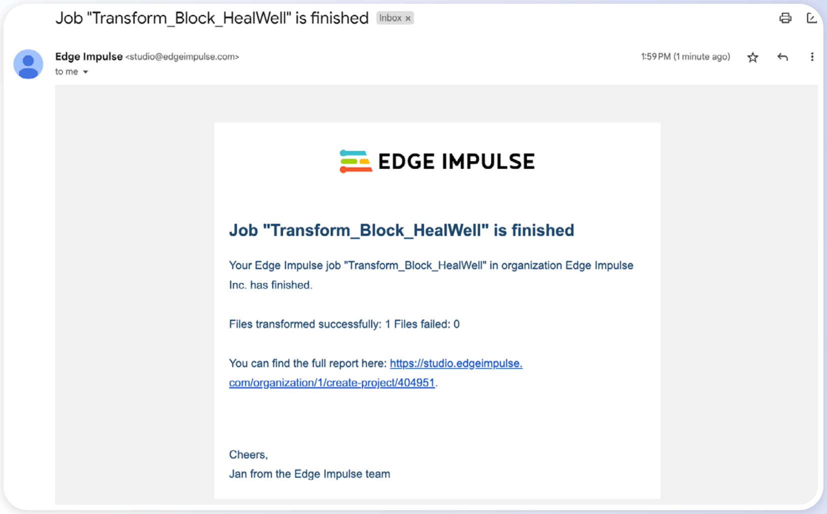
Number of parallel jobs: 3

Users to notify: Kinara Pandya

Next, select "start transformation job". This will make the transformed data available in the Output dataset.

Make sure to click on "Copy as pipeline" and save the code. We will use it later while creating data pipelines in step 10.

Step 6: Receive email with status of the transformation job



Job "Transform_Block_HealWell" is finished Inbox ×

Edge Impulse <studio@edgeimpulse.com>
to me
1:59 PM (1 minute ago)

EDGE IMPULSE

Job "Transform_Block_HealWell" is finished

Your Edge Impulse job "Transform_Block_HealWell" in organization Edge Impulse Inc. has finished.

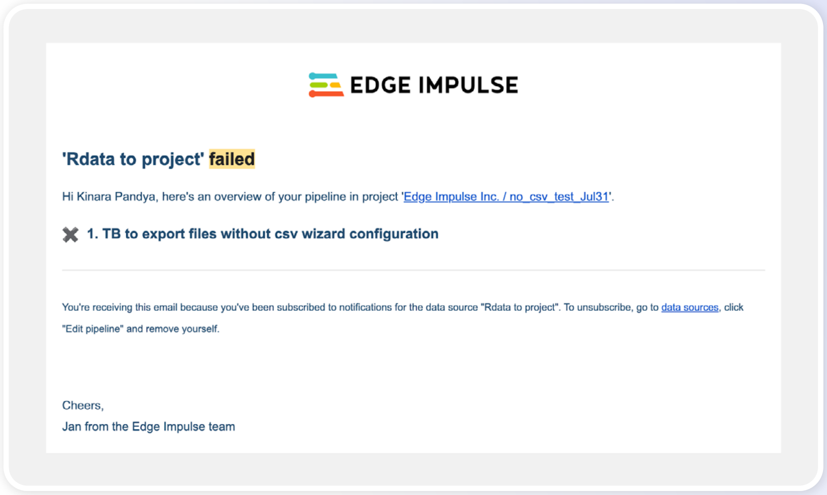
Files transformed successfully: 1 Files failed: 0

You can find the full report here: https://studio.edgeimpulse.com/organization/1/create-project/404951.

Cheers,
Jan from the Edge Impulse team

If the transformation was successful, you will receive an email as shown below.

If the transformation was not successful, you will receive a similar email as shown below.



EDGE IMPULSE

'Rdata to project' failed

Hi Kinara Pandya, here's an overview of your pipeline in project 'Edge Impulse Inc. / no_csv_test_Jul31'.

✖ **1. TB to export files without csv wizard configuration**

You're receiving this email because you've been subscribed to notifications for the data source "Rdata to project". To unsubscribe, go to data sources, click "Edit pipeline" and remove yourself.

Cheers,
Jan from the Edge Impulse team

Step 7: Under your organization, navigate to "Projects" and create new project

Quite often, data scientists and ML engineers want to experiment with a slice of data only. Now that we have clean data available in the dataset from the step above, we can feed this data into a project for feature extraction, visualization, model training and generating a C++ library for deployment on any target hardware.

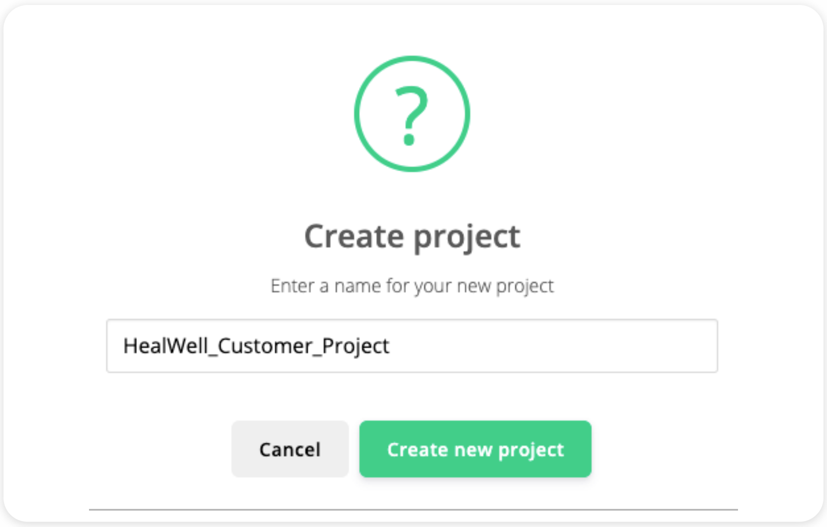
Step 8: Set up Data pipeline #1 (from S3 into project)

Let's automate all the steps until now (i.e fetching data from S3 bucket, running a transformation block every 30 days to clean any new data and feeding it into a dataset and project) using the *"Data pipelines"* feature within EI.

To configure data pipelines, follow the steps below:

In your organization, under *"data pipelines"* tab, "add new pipeline"

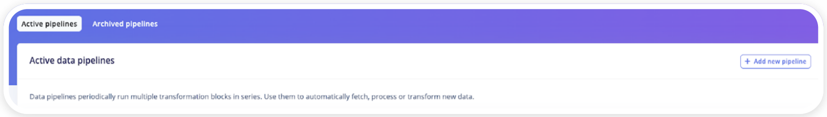
Paste the code saved in step 6 from "Copy as pipeline" step while configuring the new pipeline.



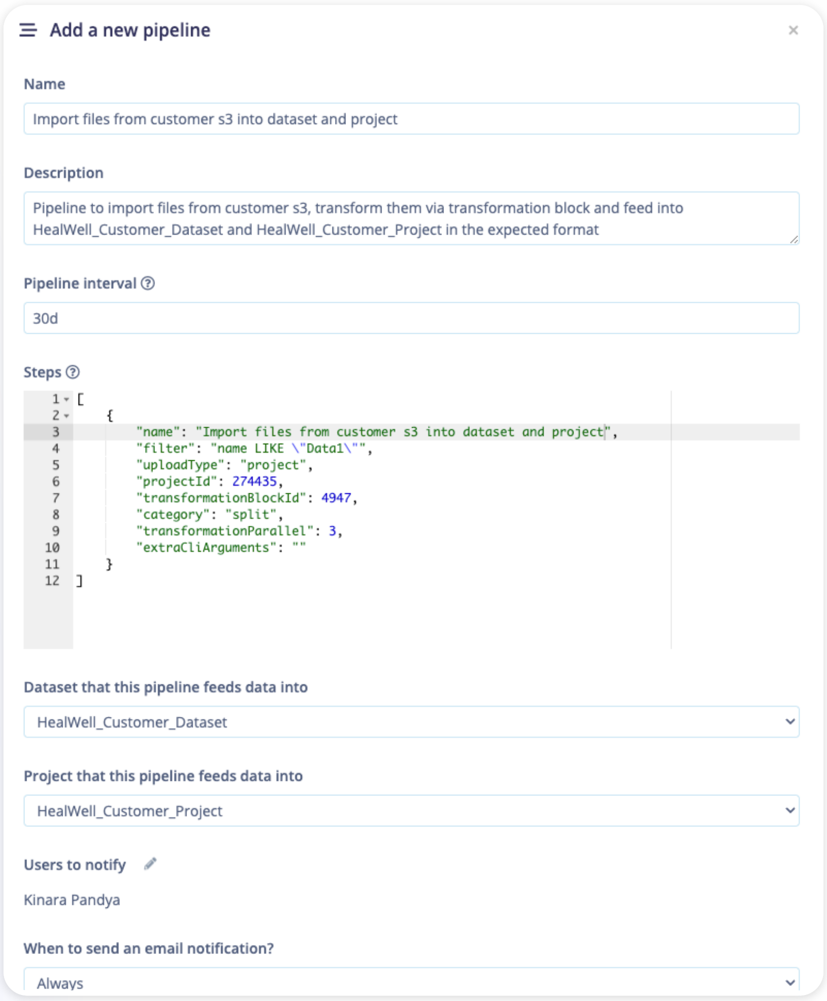
Create project

Enter a name for your new project

HealWell_Customer_Project



Data pipelines periodically run multiple transformation blocks in series. Use them to automatically fetch, process or transform new data.



Add a new pipeline

Name: Import files from customer s3 into dataset and project

Description: Pipeline to import files from customer s3, transform them via transformation block and feed into HealWell_Customer_Dataset and HealWell_Customer_Project in the expected format

Pipeline interval: 30d

Steps

```
[
    {
        "name": "Import files from customer s3 into dataset and project",
        "filter": "name LIKE \"Data1\"",
        "uploadType": "project",
        "projectId": 274435,
        "transformationBlockId": 4947,
        "category": "split",
        "transformationParallel": 3,
        "extraCliArguments": ""
    }
]
```

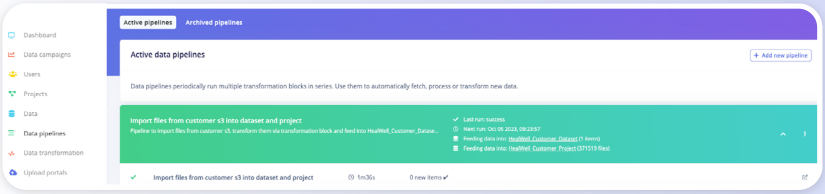
Dataset that this pipeline feeds data into: HealWell_Customer_Dataset

Project that this pipeline feeds data into: HealWell_Customer_Project

Users to notify: Kinara Pandya

When to send an email notification?: Always

Once the pipeline runs successfully, you should see an active pipeline under the "Active pipelines" tab.



Active data pipelines

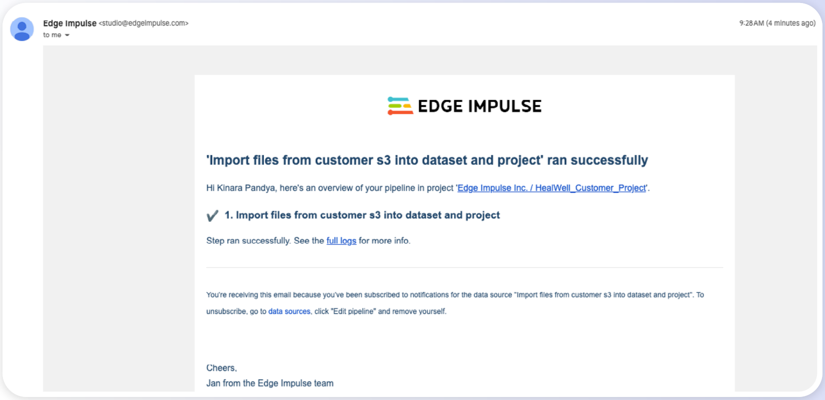
Data pipelines periodically run multiple transformation blocks in series. Use them to automatically fetch, process or transform new data.

Import files from customer s3 into dataset and project
Pipeline to import files from customer s3, transform them via transformation block and feed into HealWell_Customer_Datase...

✓ Last run: success
Next run: Oct 05 2023, 09:23:57
Feeding data into: HealWell_Customer_Dataset (1 items)
Feeding data into: HealWell_Customer_Project (371519 files)

✓ Import files from customer s3 into dataset and project 1m36s 0 new items ✓

Immediately, following completion of the pipeline, you should receive an update on the status of your pipeline via email.



Edge Impulse <studio@edgeimpulse.com>
to me
9:28 AM (4 minutes ago)

EDGE IMPULSE

'Import files from customer s3 into dataset and project' ran successfully

Hi Kinara Pandya, here's an overview of your pipeline in project 'Edge Impulse Inc. / HealWell_Customer_Project'.

✓ 1. Import files from customer s3 into dataset and project

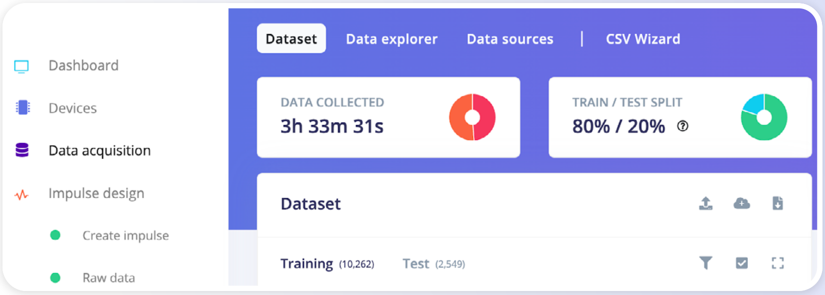
Step ran successfully. See the full logs for more info.

You're receiving this email because you've been subscribed to notifications for the data source "Import files from customer s3 into dataset and project". To unsubscribe, go to data sources, click "Edit pipeline" and remove yourself.

Cheers,
Jan from the Edge Impulse team

Step 9: Navigate to "Projects" tab → "Data acquisition"

Your cleaned data is now populated into *"HealWell_ Customer_Project"* as seen under the *"Data acquisition"* tab.



DATA COLLECTED
3h 33m 31s

TRAIN / TEST SPLIT
80% / 20%

Dataset

Training (10,262) Test (2,549)

3. Feature Engineering and Model Training

Create an impulse

After the data is cleaned, meaningful features need to be extracted from it for the model to learn the most important patterns within the data. In order to achieve this, we need to first set up/create an impulse.

So what is an "Impulse" ?
A complete "Impluse" consists of 3 main building blocks: *input block*, processing block and a *learning block* mimicking the stages of a standard ML workflow of data input, feature extraction and training a model respectively.

The above capabilities of Edge Impulse help users derive insights from their data, which serve as the input for machine learning models.

Input block

The input block indicates the type of input data you are training your model with. For the purpose of this tutorial, we have chosen time series data.

Processing block

Extracting meaningful features from your data is crucial to building small and reliable machine learning models. In many cases, raw data can be extremely high dimensional, with many variables that may not be particularly informative for the task at hand. Feature extraction allows us to transform this raw data into a lower-dimensional space that captures the most important aspects of the data. It helps remove the noise and highlights only the important signals, making the model more reliable. With smaller, less complex models, computation and storage requirements are reduced which is the key for edge devices, which often have limited computational power and memory.

In Edge Impulse this is done through processing blocks.

A **processing block** is basically a feature extractor. It consists of DSP (Digital Signal Processing) operations such as normalization, scaling, flattening etc. that are used to extract features that our model learns on.

For example, to monitor vital signs such as temperature, heart rate, blood glucose, oxygen etc. in real time, a model trained on EI or outside (BYOM) could be deployed to a wearable device such as Polar H10 heart rate sensor, to continuously monitor heart rate. These signals are extracted to reduce noise and feed in the correct features for model training. EI provides a robust built-in **"PPG (Photoplethysmography) to Heart rate Variance (HRV)"** processing block that can be used right away for your digital signal processing needs. If the model detects unusual trends that might indicate a medical emergency, it could automatically alert emergency contacts or healthcare providers.

Another example would be for sleep stage analysis, where data from wearable devices such as smartwatches or smart rings could be used to analyze users' sleep stages, helping users understand their sleep patterns. By detecting different sleep stages (e.g., REM, deep sleep, light sleep) using heart rate data from the *"PPG to HRV"* processing block, your application could provide personalized recommendations to improve sleep quality, such as adjusting bedtime routines or sleep environments.

To recap, previously, we used transformation blocks to clean the messy raw data stored in the S3 bucket. For the purpose of this tutorial, we have not used a processing block but rather just used the clean data as features which will feed into the model for training in the next step.

EON Tuner for model optimization

In machine learning, hyperparameters are the variables that govern the training process itself, such as the learning rate or the number of layers in a neural network. These are not learned from the data during training and must be set beforehand. Finding the optimal set of hyperparameters for a model can be a challenging and time-consuming task. It usually involves a lot of trial and error, and even with the same dataset, different sets of hyperparameters can produce very different results.

The EON Tuner (a proprietary algorithm developed by EI) automates this process by evaluating many candidate model architectures (selected based on your target device and latency requirements) concurrently to help you find the best performing architecture for your application.

It helps you find and select the best embedded machine learning model for your application within the constraints of your target device. It analyzes your input data, potential signal processing blocks, and neural network architectures - and gives you an overview of possible model architectures that will fit your chosen device's latency and memory requirements.

For example, EON Tuner enables sleep monitoring wearables to comprehensively analyze sleep patterns and stages using data from various sensors, such as accelerometers and heart rate monitors. By tailoring the machine learning model to the unique constraints of the wearable sleep monitor, the EON Tuner ensures that users receive personalized insights into their sleep quality, helping them make informed decisions to improve their overall well-being.

Another example of a target device is a small wearable device that continuously records a patient's heart rate data to analyze their cardiac health. However, due to the limited computational resources and power constraints of the wearable, finding the right machine learning model architecture and hyperparameters becomes challenging. The EON Tuner steps in by evaluating various combinations of model architectures and hyperparameters that are suitable for the wearable's hardware limitations.

For time series data (in this tutorial), the tuner would test different sample window sizes and increment amounts.

Depending on the selected dataset category, the EON Tuner considers a variety of **Processing blocks** when evaluating model architectures. It then tests the different parameters and configurations of these processing blocks.

In summary, different model architectures, hyper-parameters, and even data augmentation techniques are evaluated by the EON Tuner. The tuner combines these different neural networks with the processing and input options described above, and then compares the end-to-end performance that balances accuracy, memory usage, and processing speed.

Learning block

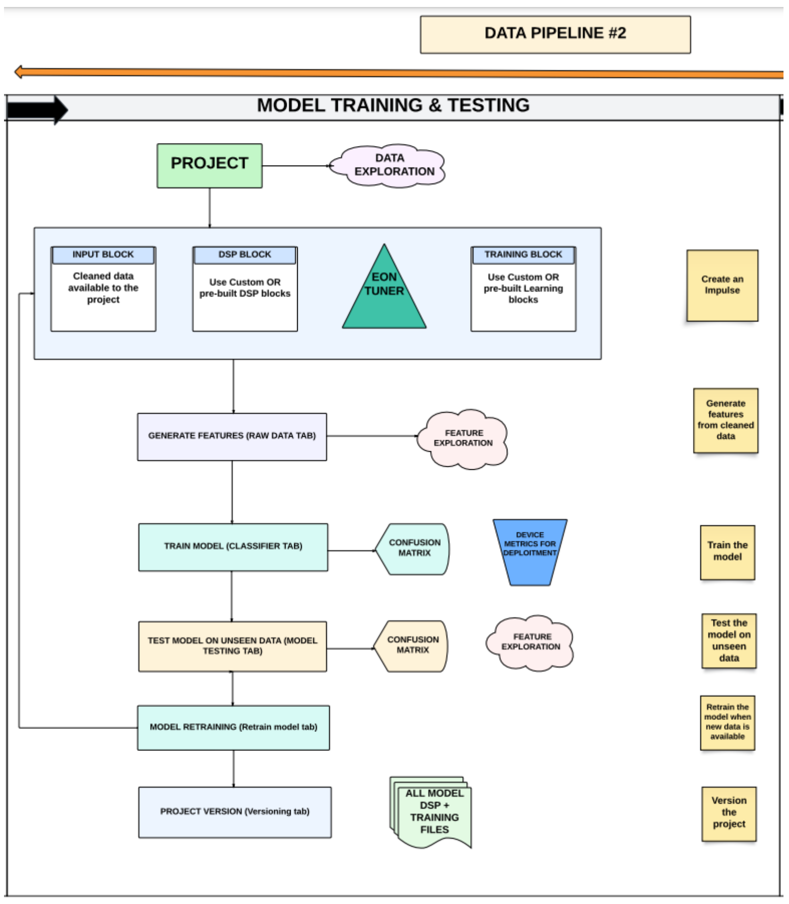
Model training is the phase where your machine learning model learns to recognize patterns, correlations, and relationships in the input data. In a digital health application, a model might learn to predict health outcomes based on patterns in patient data. For example, consider a model trained to predict blood glucose levels for patients with diabetes. The model would learn from historical data, such as glucose levels, insulin doses, age, and dietary information. By recognizing patterns and correlations in this data, the model can predict blood glucose levels in real time. This enables patients to proactively manage their diabetes by making informed decisions about insulin doses and dietary choices.

Another example would be, training a model to classify sleep stages based on data from wearable sensors like accelerometers and heart rate monitors. The model learns from a labeled sleep dataset and corresponding sensor data. As the model processes this data, it learns to identify patterns indicative of different sleep stages such as deep sleep, REM sleep, and light sleep. Once trained, the model can automatically classify sleep stages, providing valuable insights into sleep quality for users.

Essentially, without model training, machine learning wouldn't exist. It's the critical process that turns raw data into a tool for prediction and decision-making.

We achieve this within EI by using the Learning blocks. A learning block is simply a ML model, for e.g a neural network, that is trained to learn on your data. The basic idea is that a neural network classifier will take some input data, and output a probability score that indicates how likely it is that the input data belongs to a particular class.

After adding your processing block from the previous step,follow the steps below and add a learning block (mimics the train your model stage in a typical ML workflow) to make your impulse complete.


DATA PIPELINE #2
MODEL TRAINING & TESTING
PROJECT
DATA EXPLORATION
INPUT BLOCK
Cleaned data available to the project
DSP BLOCK
Use Custom OR pre-built DSP blocks
EON TUNER
TRAINING BLOCK
Use Custom OR pre-built Learning blocks
Create an Impulse
GENERATE FEATURES (RAW DATA TAB)
FEATURE EXPLORATION
Generate features from cleaned data
TRAIN MODEL (CLASSIFIER TAB)
CONFUSION MATRIX
DEVICE METRICS FOR DEPLOYMENT
Train the model
TEST MODEL ON UNSEEN DATA (MODEL TESTING TAB)
CONFUSION MATRIX
FEATURE EXPLORATION
Test the model on unseen data
MODEL RETRAINING (Retrain model tab)
Retrain the model when new data is available
PROJECT VERSION (Versioning tab)
ALL MODEL DSP + TRAINING FILES
Version the project

Step 1: Navigate to “Impulse design” → “Create impulse”

In this example, the "Input" block comes pre-populated from data acquisition. Configure the DSP (digital signal processing) and "Learning blocks". They usually display a star to indicate the most recommended blocks based on your input data. *You may use the build-in options or add custom blocks if the sensor type or model you wish to use is not supported.*

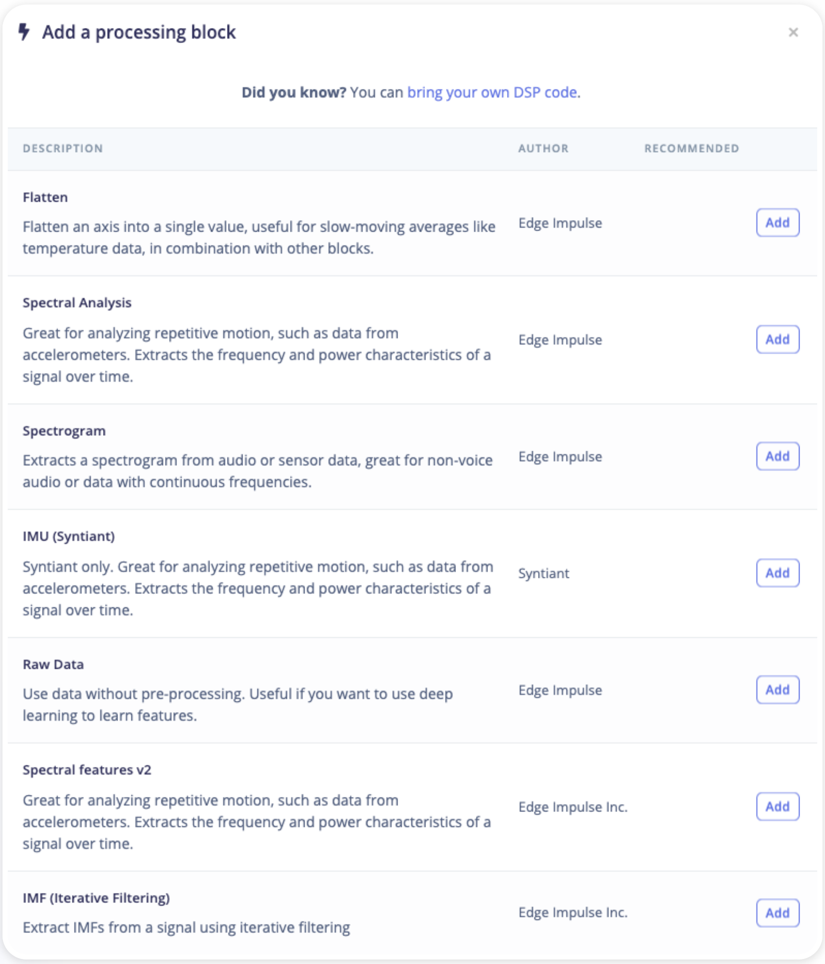
Select "Raw Data" as a *"processing block"* for the purpose of this tutorial.

(Optional) EON Tuner:
Finds the optimal architecture for your model

The EON™ Tuner will evaluate many candidate model architectures (selected based on your target device and latency requirements) concurrently to help you find the best performing architecture for your resource constrained device.
The search process can take up to 6 hours to complete. While the EON Tuner is running you can view the progress on this page at any time.

(optional) Navigate to Project → EON Tuner tab and select the target hardware

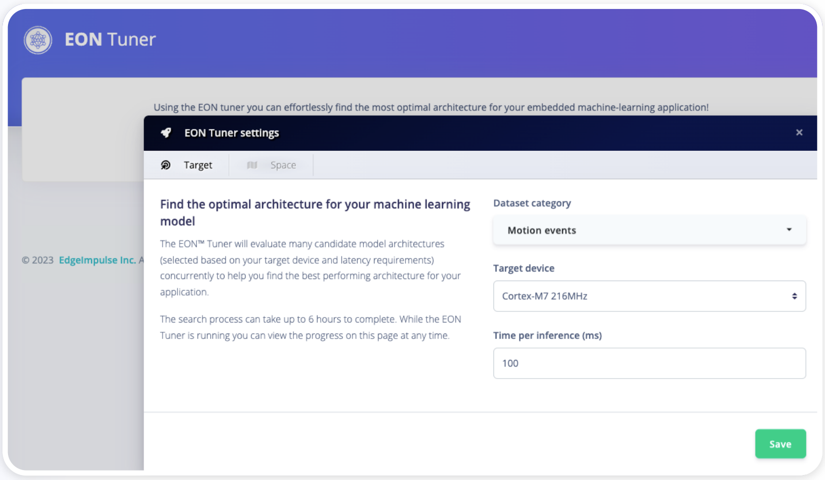
Next, add a learning block. Select the *"Classification"* block for this tutorial to predict if the student was confused or not while watching MOOC (Massive Open Online Courses) videos.



Add a processing block

Did you know? You can bring your own DSP code.

DESCRIPTION	AUTHOR	RECOMMENDED
Flatten Flatten an axis into a single value, useful for slow-moving averages like temperature data, in combination with other blocks.	Edge Impulse	Add
Spectral Analysis Great for analyzing repetitive motion, such as data from accelerometers. Extracts the frequency and power characteristics of a signal over time.	Edge Impulse	Add
Spectrogram Extracts a spectrogram from audio or sensor data, great for non-voice audio or data with continuous frequencies.	Edge Impulse	Add
IMU (Syntiant) Syntiant only. Great for analyzing repetitive motion, such as data from accelerometers. Extracts the frequency and power characteristics of a signal over time.	Syntiant	Add
Raw Data Use data without pre-processing. Useful if you want to use deep learning to learn features.	Edge Impulse	Add
Spectral features v2 Great for analyzing repetitive motion, such as data from accelerometers. Extracts the frequency and power characteristics of a signal over time.	Edge Impulse Inc.	Add
IMF (Iterative Filtering) Extract IMFs from a signal using iterative filtering	Edge Impulse Inc.	Add



EON Tuner

Using the EON tuner you can effortlessly find the most optimal architecture for your embedded machine-learning application!

EON Tuner settings

Target | Space

Find the optimal architecture for your machine learning model

The EON™ Tuner will evaluate many candidate model architectures (selected based on your target device and latency requirements) concurrently to help you find the best performing architecture for your application.

The search process can take up to 6 hours to complete. While the EON Tuner is running you can view the progress on this page at any time.

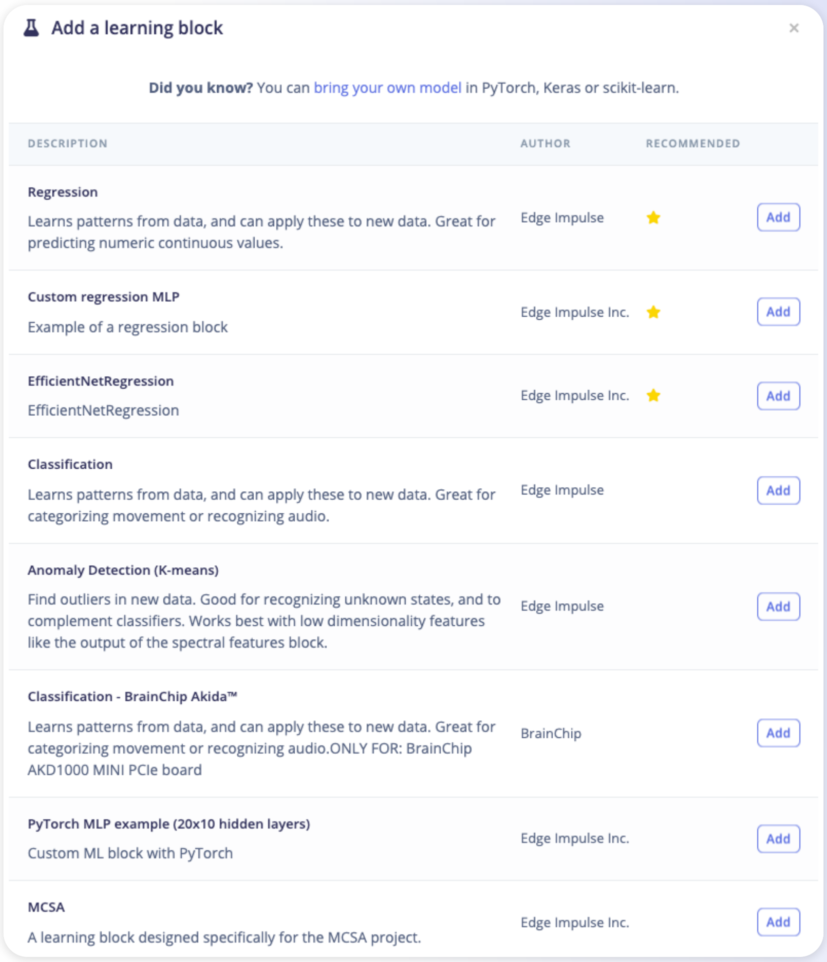
Dataset category: Motion events

Target device: Cortex-M7 216MHz

Time per inference (ms): 100

Save

© 2023 EdgeImpulse Inc.

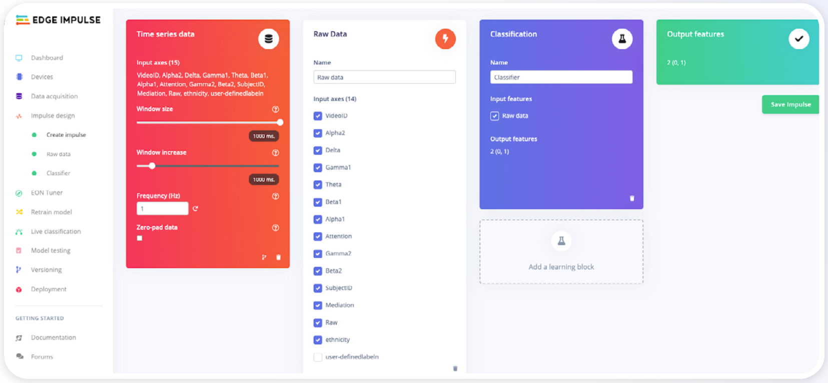


Add a learning block

Did you know? You can bring your own model in PyTorch, Keras or scikit-learn.

DESCRIPTION	AUTHOR	RECOMMENDED	
Regression Learns patterns from data, and can apply these to new data. Great for predicting numeric continuous values.	Edge Impulse	★	Add
Custom regression MLP Example of a regression block	Edge Impulse Inc.	★	Add
EfficientNetRegression EfficientNetRegression	Edge Impulse Inc.	★	Add
Classification Learns patterns from data, and can apply these to new data. Great for categorizing movement or recognizing audio.	Edge Impulse		Add
Anomaly Detection (K-means) Find outliers in new data. Good for recognizing unknown states, and to complement classifiers. Works best with low dimensionality features like the output of the spectral features block.	Edge Impulse		Add
Classification - BrainChip Akida™ Learns patterns from data, and can apply these to new data. Great for categorizing movement or recognizing audio.ONLY FOR: BrainChip AKD1000 MINI PCIe board	BrainChip		Add
PyTorch MLP example (20x10 hidden layers) Custom ML block with PyTorch	Edge Impulse Inc.		Add
MCSA A learning block designed specifically for the MCSA project.	Edge Impulse Inc.		Add

Once the entire flow of the impulse is configured. Save the impulse.



Time series data

Input axes (15)
VideoID, Alpha2, Delta, Gamma1, Theta, Beta1, Alpha1, Attention, Gamma2, Beta2, SubjectID, Mediation, Raw, ethnicity, user-definedlabeln

Window size: 1000 ms.

Window increase: 1000 ms.

Frequency (Hz): 1

Zero-pad data: ☐

Raw Data

Name: Raw data

Input axes (14)
- ✓ VideoID
- ✓ Alpha2
- ✓ Delta
- ✓ Gamma1
- ✓ Theta
- ✓ Beta1
- ✓ Alpha1
- ✓ Attention
- ✓ Gamma2
- ✓ Beta2
- ✓ SubjectID
- ✓ Mediation
- ✓ Raw
- ✓ ethnicity
- ☐ user-definedlabeln

Classification

Name: Classifier

Input features: ✓ Raw data

Output features: 2 (0, 1)

Output features

2 (0, 1)

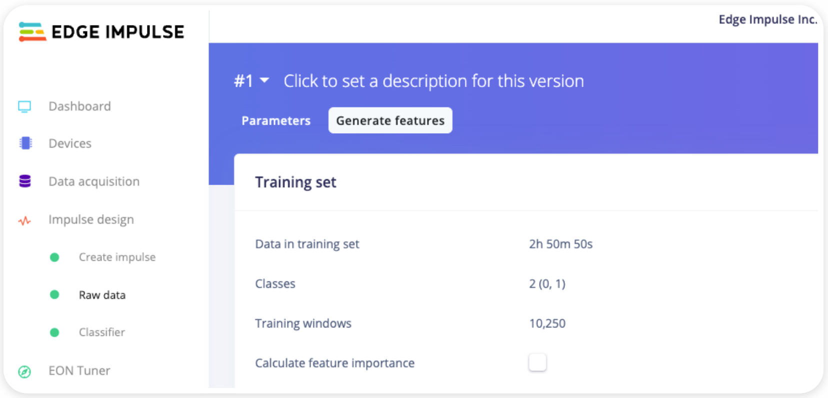
Step 2: Navigate to “Raw data” and “Generate features”

The screen also displays approx. on-device performance on a Cortex-M4 80MHz. This may be changed by navigating to the “Devices” tab.

Step 3: Navigate to “Classifier” and start Training

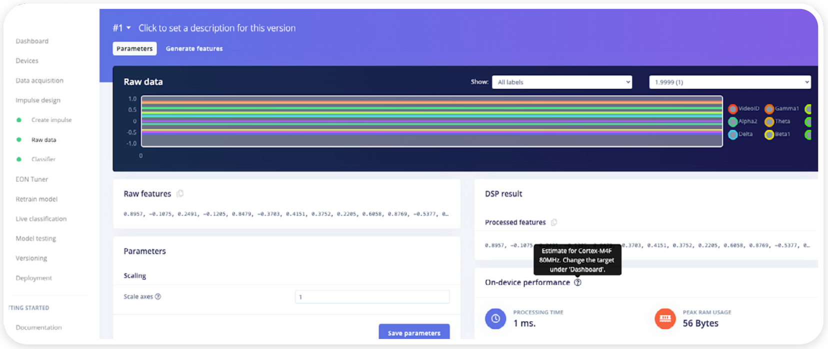
Configure the Neural Network architecture by selecting the “add an extra layer” button. You may “edit” or “delete” the layer/s as shown below.

You may also switch to “Keras (expert) mode” to edit the code.



Training set

Data in training set	2h 50m 50s
Classes	2 (0, 1)
Training windows	10,250
Calculate feature importance	



Raw features

```
0.8957, -0.1075, 0.2491, -0.1205, 0.8479, -0.3703, 0.4151, 0.3752, 0.2205, 0.6058, 0.8769, -0.5377, 0…
```

Parameters

Scaling

Scale axes: 1

DSP result

Processed features

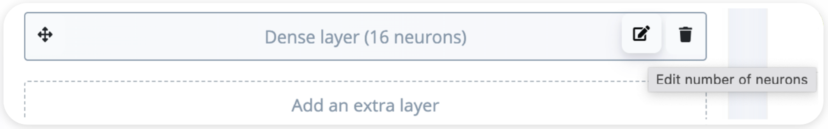
```
0.8957, -0.1075, …0.3703, 0.4151, 0.3752, 0.2205, 0.6058, 0.8769, -0.5377, 0…
```

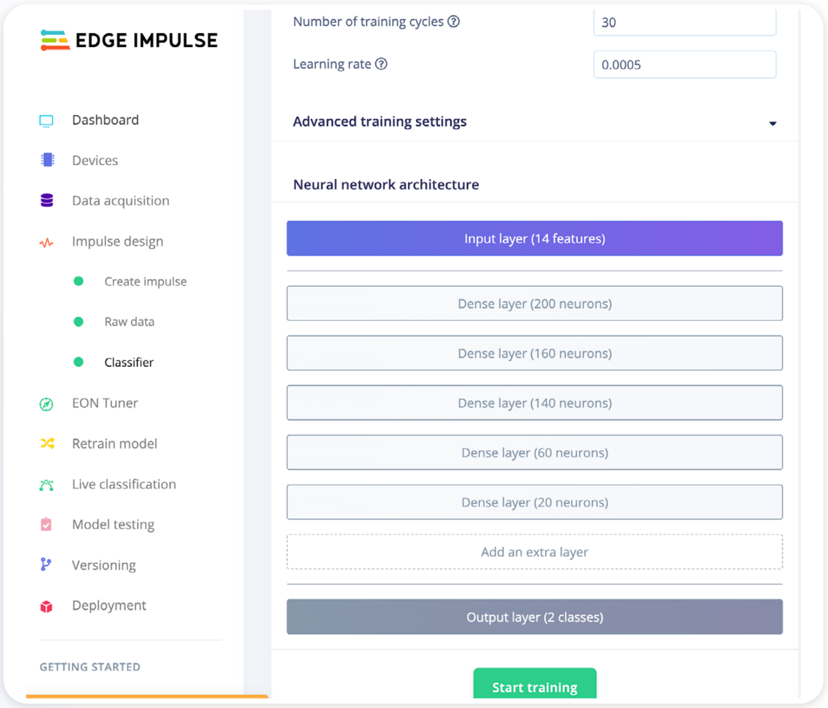
Estimate for Cortex-M4F 80MHz. Change the target under 'Dashboard'.

On-device performance

PROCESSING TIME 1 ms.

PEAK RAM USAGE 56 Bytes





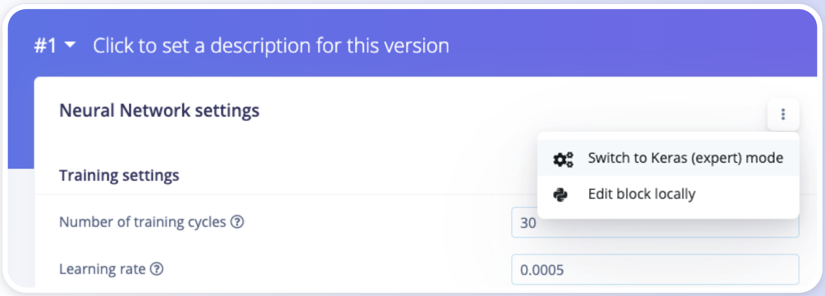
Number of training cycles: 30

Learning rate: 0.0005

Neural network architecture

- Input layer (14 features)
- Dense layer (200 neurons)
- Dense layer (160 neurons)
- Dense layer (140 neurons)
- Dense layer (60 neurons)
- Dense layer (20 neurons)
- Output layer (2 classes)

Upon completion of the training, a confusion matrix along with a visual "data explorer" and "on-device performance" metrics are generated as seen in the screenshot below.



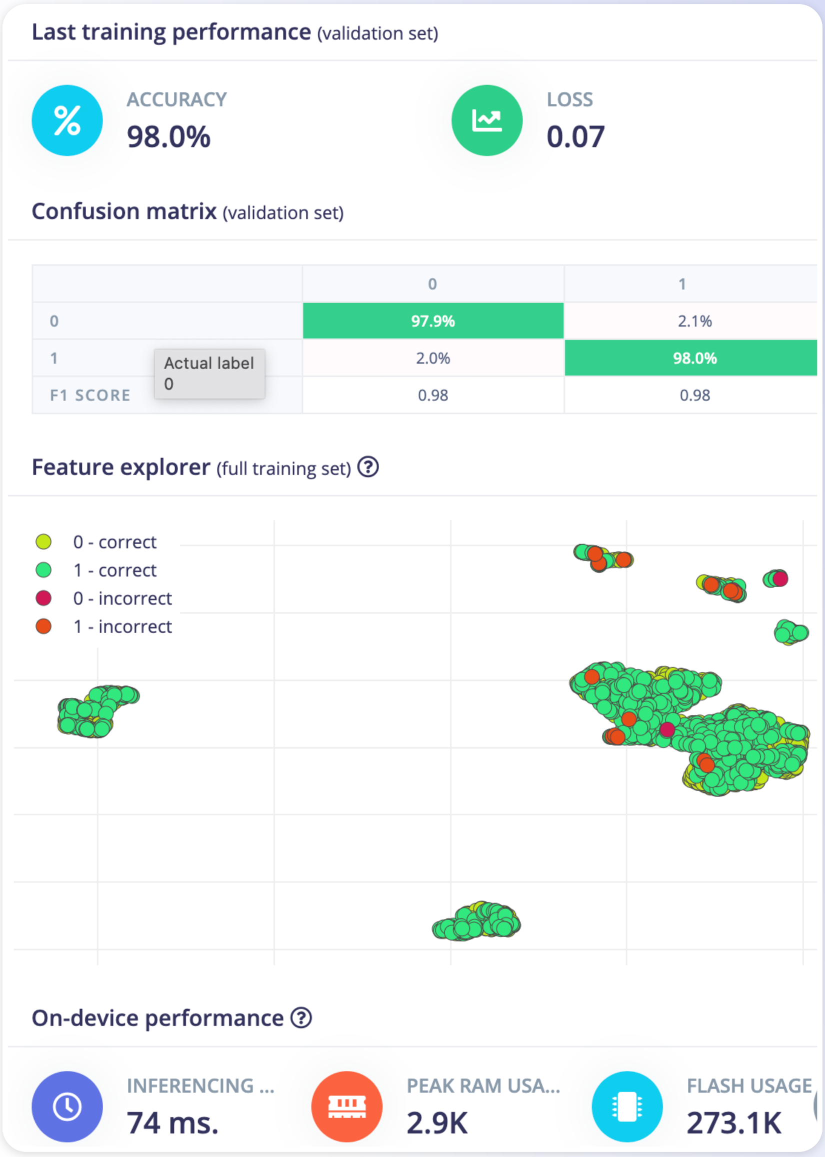
#1 Click to set a description for this version

Neural Network settings

Training settings

Number of training cycles: 30

Learning rate: 0.0005



Last training performance (validation set)

ACCURACY 98.0%

LOSS 0.07

Confusion matrix (validation set)

	0	1
0	97.9%	2.1%
1	2.0%	98.0%
F1 SCORE	0.98	0.98

Feature explorer (full training set)

On-device performance

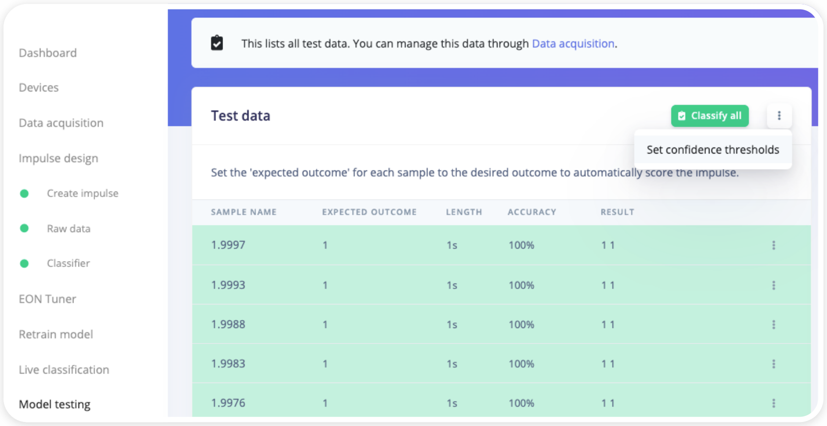
INFERENCING ... 74 ms.

PEAK RAM USA... 2.9K

FLASH USAGE 273.1K

Step 4: Navigate to the “Model testing” tab to test the trained model on unseen data.

Click “classify all to test the trained model on unseen data. You may also *“set confidence thresholds”* depending on how critical your use case is. For e.g detecting a heart attack is critical and thus setting a low confidence threshold in this case would make sense. or “delete” the layer/s as shown below.



This lists all test data. You can manage this data through Data acquisition.

Test data

Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.

SAMPLE NAME	EXPECTED OUTCOME	LENGTH	ACCURACY	RESULT
1.9997	1	1s	100%	1 1
1.9993	1	1s	100%	1 1
1.9988	1	1s	100%	1 1
1.9983	1	1s	100%	1 1
1.9976	1	1s	100%	1 1

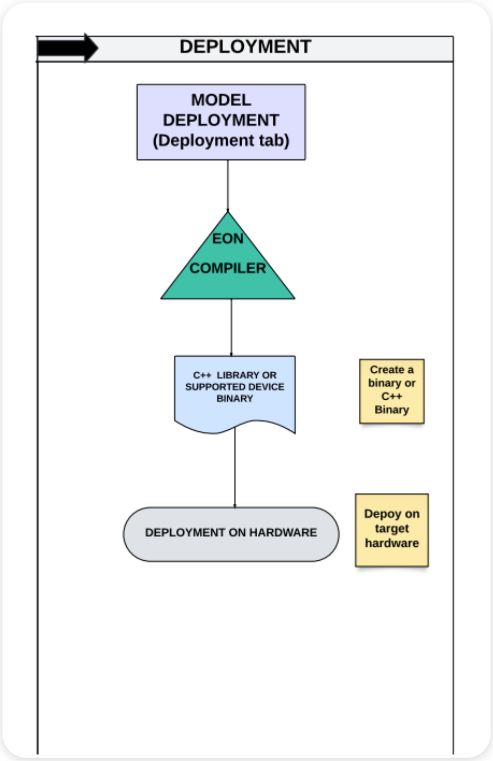
4. Model Deployment

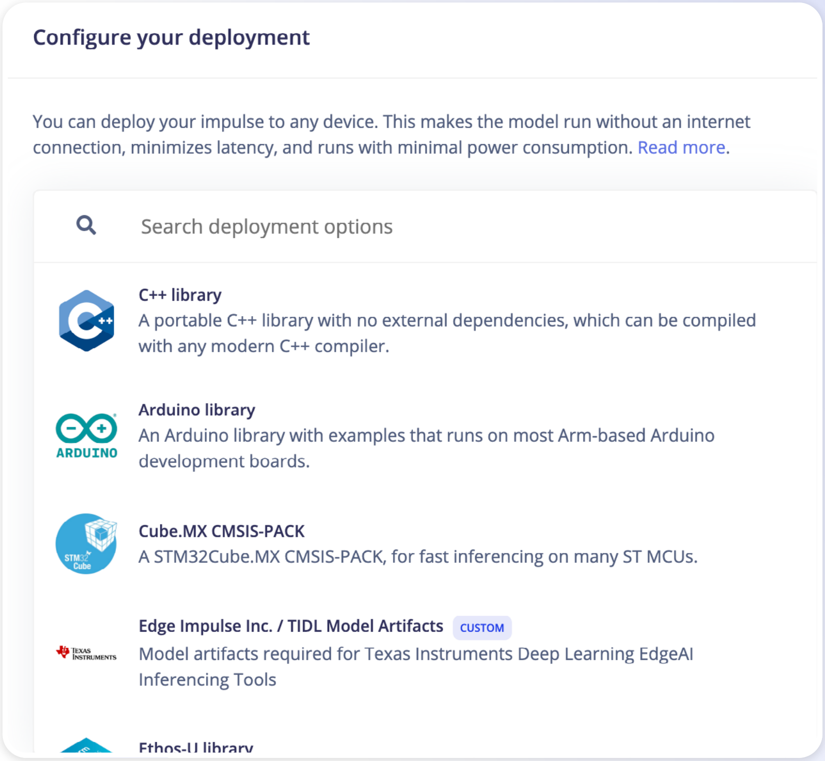
Deployment block

Digital health use-cases often involve a variety of devices like wearables, mobile devices, or specialized medical equipment. Integrating with these devices and deploying the trained models on them in an efficient manner can be challenging.

One of the most powerful features in Edge Impulse are the built-in deployment targets (under **Deployment** in the Studio), which let you create ready-to-go binaries for development boards, or custom libraries for a wide variety of targets that incorporate your trained impulse.

Regardless of whether you are using a fully supported development board or not, you can deploy your impulse to any device. The C++ library and Edge Impulse SDK enable the model to run without an internet connection on the device, minimize latency, and with minimal power consumption.


DEPLOYMENT
MODEL DEPLOYMENT (Deployment tab)
EON COMPILER
C++ LIBRARY OR SUPPORTED DEVICE BINARY
Create a binary or C++ Binary
DEPLOYMENT ON HARDWARE
Depoy on target hardware


Configure your deployment

You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. Read more.

Search deployment options

C++ library
A portable C++ library with no external dependencies, which can be compiled with any modern C++ compiler.

Arduino library
An Arduino library with examples that runs on most Arm-based Arduino development boards.

Cube.MX CMSIS-PACK
A STM32Cube.MX CMSIS-PACK, for fast inferencing on many ST MCUs.

Edge Impulse Inc. / TIDL Model Artifacts CUSTOM
Model artifacts required for Texas Instruments Deep Learning EdgeAI Inferencing Tools

Ethos-U library

EON Compiler (optional)

Many digital health applications need to operate in real-time. For instance, a wearable that monitors heart rate variability for early detection of cardiac issues, or a system that alerts for falls in the elderly, requires immediate response. A slow or lagging model could lead to delayed alerts, potentially risking lives.

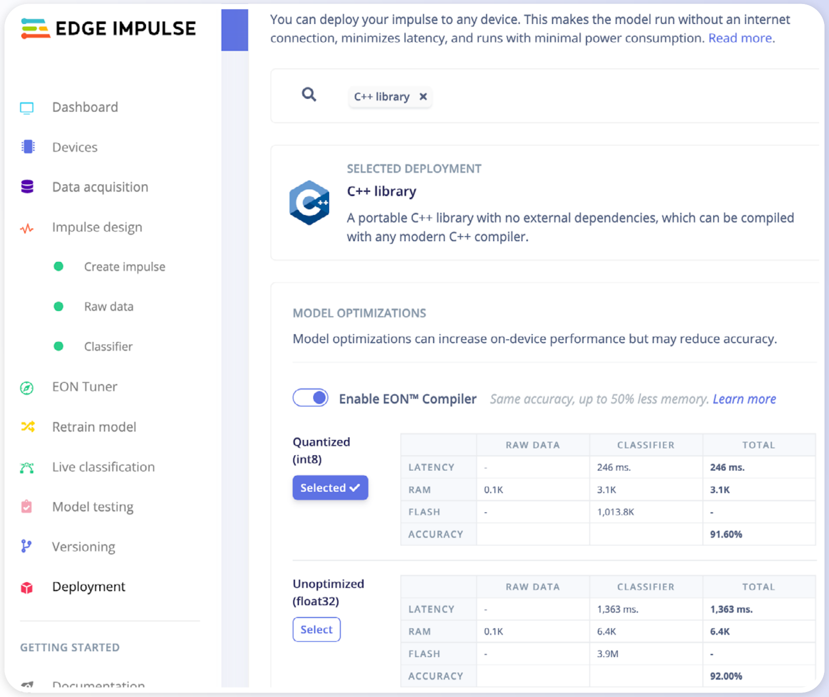
Additionally, the user experience is another important factor to consider. A device that is slow or unresponsive can frustrate users, leading to poor adoption rates. On the other hand, a device with a smooth, responsive user interface, facilitated by well-optimized models, can encourage user engagement. As a result, providing real-time monitoring and diagnosis, while maintaining a high standard of user experience and data privacy is crucial for digital health applications.

To support this goal, EI provides EON Compiler capability which helps you get the best performance from your device. By optimizing machine learning models to better utilize the device's specific hardware characteristics, the EON Compiler ensures your application runs as efficiently as possible.

5. Data Pipelines

Retrain model & versioning

One of the most common pain points in the digital health device space is the challenge of managing the constant influx of new data, drifted data distribution, model overfitting issues and ensuring seamless updates. Data pipelines enable automated data updates, effortlessly incorporating the latest information into your analysis. Moreover, pipeline versioning allows you to track and reproduce your findings, ensuring transparency and traceability.



You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. Read more.

SELECTED DEPLOYMENT

C++ library

A portable C++ library with no external dependencies, which can be compiled with any modern C++ compiler.

MODEL OPTIMIZATIONS

Model optimizations can increase on-device performance but may reduce accuracy.

Enable EON™ Compiler *Same accuracy, up to 50% less memory. Learn more*

Quantized (int8) — Selected ✓

	RAW DATA	CLASSIFIER	TOTAL
LATENCY	-	246 ms.	**246 ms.**
RAM	0.1K	3.1K	**3.1K**
FLASH	-	1,013.8K	-
ACCURACY			**91.60%**

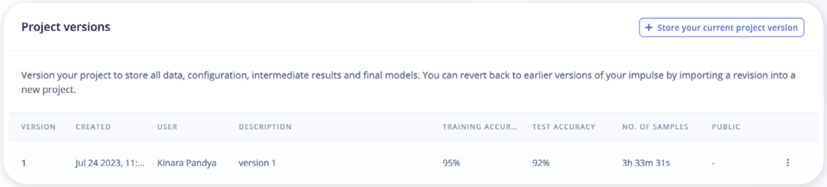
Unoptimized (float32) — Select

	RAW DATA	CLASSIFIER	TOTAL
LATENCY	-	1,363 ms.	**1,363 ms.**
RAM	0.1K	6.4K	**6.4K**
FLASH	-	3.9M	-
ACCURACY			**92.00%**

The Retrain model feature (optional in this tutorial) in Edge Impulse Studio comes handy during such scenarios. It uses already known parameters from your selected DSP and ML blocks then uses them to automatically regenerate new features and retrain the Neural Network model in one single step.

To version your current project, follow the steps below.

Step 1: Navigate to Project → Versioning tab → store your current project version



Project versions

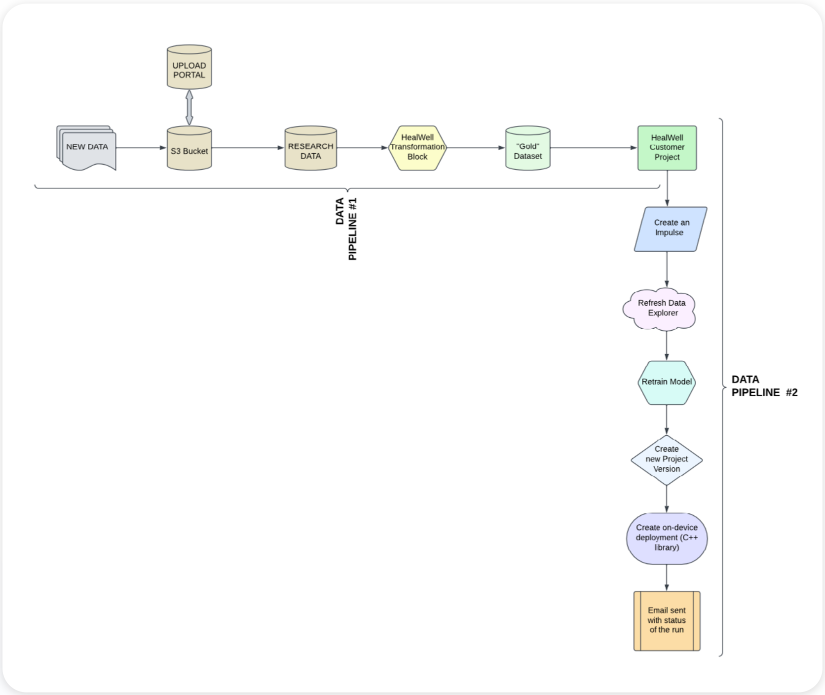
Version your project to store all data, configuration, intermediate results and final models. You can revert back to earlier versions of your impulse by importing a revision into a new project.

VERSION	CREATED	USER	DESCRIPTION	TRAINING ACCUR...	TEST ACCURACY	NO. OF SAMPLES	PUBLIC
1	Jul 24 2023, 11:...	Kinara Pandya	version 1	95%	92%	3h 33m 31s	-

Data Pipelines

Automated data pipelines can fuel machine learning models used for a variety of use cases such as predictive analytics by forecasting patient health outcomes, detecting potential health risks earlier, and guiding preventative care strategies.

In digital health, especially with the rise of wearable technologies and IoT devices, there's a constant stream of health data being produced. Data pipelines can help process this real-time data, allowing for continuous health monitoring and immediate response to critical changes.


UPLOAD PORTAL
NEW DATA
S3 Bucket
RESEARCH DATA
HealWell Transformation Block
"Gold" Dataset
HealWell Customer Project
DATA PIPELINE #1
Create an Impulse
Refresh Data Explorer
Retrain Model
Create new Project Version
Create on-device deployment (C++ library)
Email sent with status of the run
DATA PIPELINE #2

To support the above requirements, EI provides the capability to build data pipelines where you can stack several transformation blocks which then feed transformed data into a dataset or a project.

We already created a data pipeline to fetch data from s3 into the project here. In this section, we will extend the pipeline to retrain the model when new data becomes available.

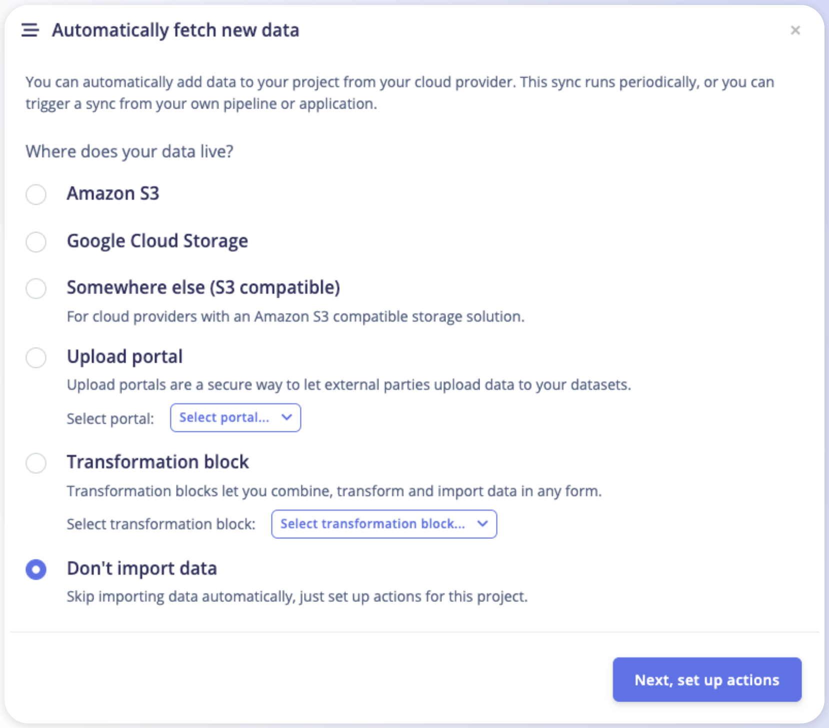
As new data gets added, the pipeline smoothly manages the entire Edge AI platform process from start to finish: it recreates the data explorer, allowing insights into the newly acquired data; it generates and versions the retrained model, keeping track of the model's evolution over time; and also constructs a new C++ library, ensuring that the updated model is efficiently integrated into the target device of your choice. This approach not only ensures the model's adaptability to changing conditions but also optimizes its deployment, ensuring that the target device on which the model is deployed benefits from the most current and relevant insights provided by the retrained model.

Follow the steps below to configure this automated data pipeline.

Step 2: Creating data pipeline #2 (model re-training to deployment)

Navigate to your Project → Data acquisition → Data source → Add a new data source

Select "Don't import data" → "Next, setup actions"



Automatically fetch new data

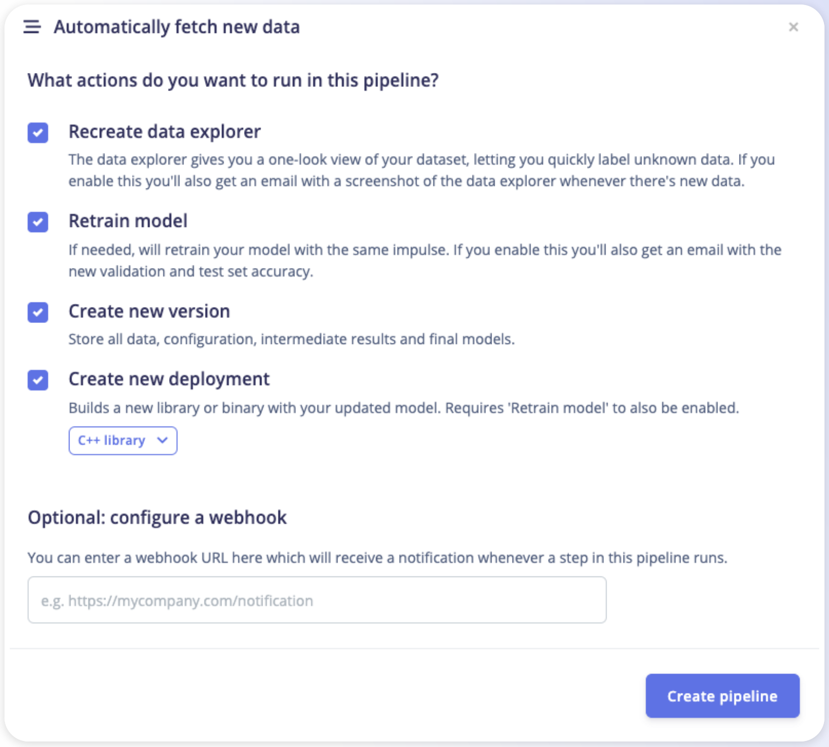
You can automatically add data to your project from your cloud provider. This sync runs periodically, or you can trigger a sync from your own pipeline or application.

Where does your data live?

- ○ Amazon S3
- ○ Google Cloud Storage
- ○ Somewhere else (S3 compatible)
  For cloud providers with an Amazon S3 compatible storage solution.
- ○ Upload portal
  Upload portals are a secure way to let external parties upload data to your datasets.
  Select portal: Select portal...
- ○ Transformation block
  Transformation blocks let you combine, transform and import data in any form.
  Select transformation block: Select transformation block...
- ● Don't import data
  Skip importing data automatically, just set up actions for this project.

Select all actions and "create pipeline"

All pipelines for the project are displayed under the "data sources" tab. Active data pipelines for a sample project are shown below.



Automatically fetch new data

What actions do you want to run in this pipeline?

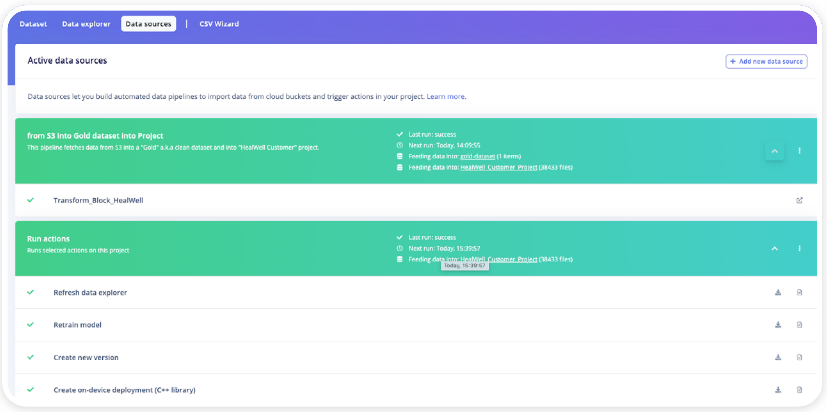
- ✓ Recreate data explorer
  The data explorer gives you a one-look view of your dataset, letting you quickly label unknown data. If you enable this you'll also get an email with a screenshot of the data explorer whenever there's new data.
- ✓ Retrain model
  If needed, will retrain your model with the same impulse. If you enable this you'll also get an email with the new validation and test set accuracy.
- ✓ Create new version
  Store all data, configuration, intermediate results and final models.
- ✓ Create new deployment
  Builds a new library or binary with your updated model. Requires 'Retrain model' to also be enabled.
  C++ library

Optional: configure a webhook

You can enter a webhook URL here which will receive a notification whenever a step in this pipeline runs.

e.g. https://mycompany.com/notification

You can also check the "Versioning" tab for the newly created version of the model.



Dataset | Data explorer | Data sources | CSV Wizard

Active data sources

+ Add new data source

Data sources let you build automated data pipelines to import data from cloud buckets and trigger actions in your project. Learn more.

from S3 into Gold dataset into Project
This pipeline fetches data from S3 into a "Gold" a.k.a clean dataset and into "HealWell Customer" project.

- Last run: success
- Next run: Today, 14:09:55
- Feeding data into: gold-dataset (1 items)
- Feeding data into: HealWell_Customer_Project (38433 files)

✓ Transform_Block_HealWell

Run actions
Runs selected actions on this project

- Last run: success
- Next run: Today, 15:39:57
- Feeding data into: HealWell_Customer_Project (38433 files)

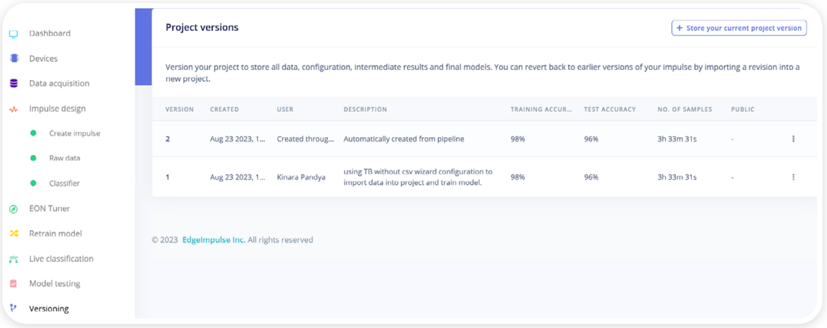
Today, 15:39:57

✓ Refresh data explorer

✓ Retrain model

✓ Create new version

✓ Create on-device deployment (C++ library)



Project versions

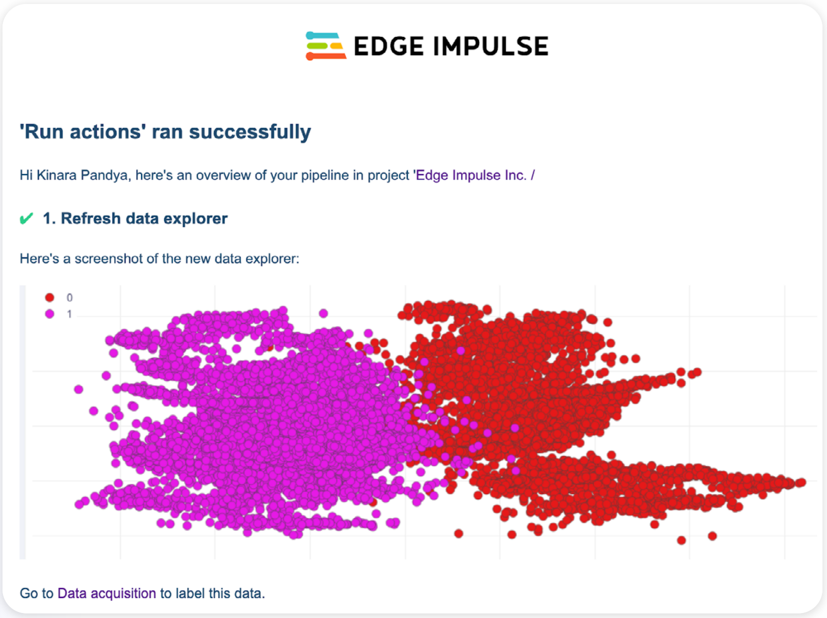
+ Store your current project version

Version your project to store all data, configuration, intermediate results and final models. You can revert back to earlier versions of your impulse by importing a revision into a new project.

VERSION	CREATED	USER	DESCRIPTION	TRAINING ACCUR...	TEST ACCURACY	NO. OF SAMPLES	PUBLIC
2	Aug 23 2023, 1...	Created throug...	Automatically created from pipeline	98%	96%	3h 33m 31s	-
1	Aug 23 2023, 1...	Kinara Pandya	using TB without csv wizard configuration to import data into project and train model.	98%	96%	3h 33m 31s	-

© 2023 EdgeImpulse Inc. All rights reserved

Immediately following completion of the pipeline run, you will receive an email with the status of the run as shown below.



EDGE IMPULSE

'Run actions' ran successfully

Hi Kinara Pandya, here's an overview of your pipeline in project 'Edge Impulse Inc. /

✔ **1. Refresh data explorer**

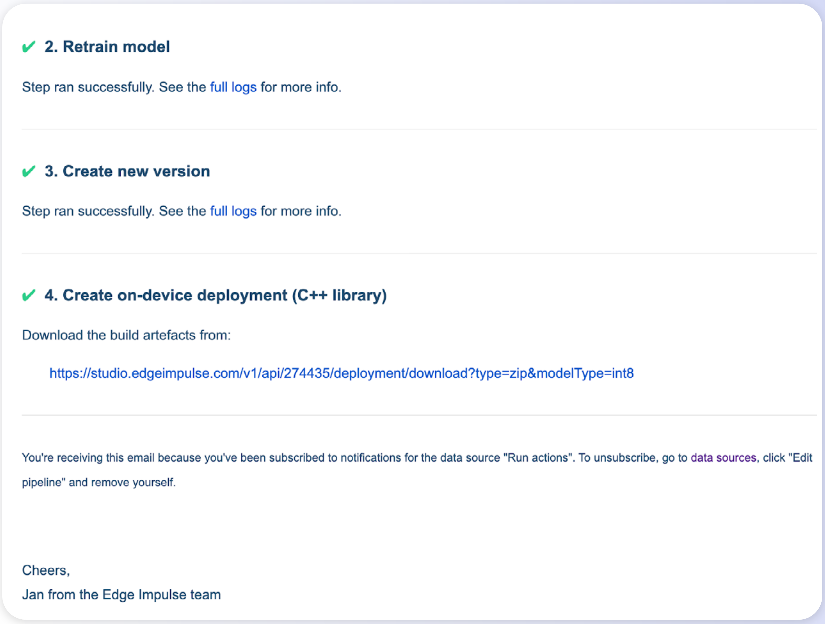
Here's a screenshot of the new data explorer:

Go to Data acquisition to label this data.

Data Campaigns (optional)

To keep the tutorial beginner friendly, this section is optional and for information only. “Data Campaigns” feature is an integral part of monitoring the health of your pipelines and model performance over time.

It allows you to quickly track your experiments as the model’s development progresses. It is an overview of your pipelines where you can easily extract useful information from your datasets and correlate those metrics with your model performances.


✔ 2. Retrain model

Step ran successfully. See the full logs for more info.

✔ 3. Create new version

Step ran successfully. See the full logs for more info.

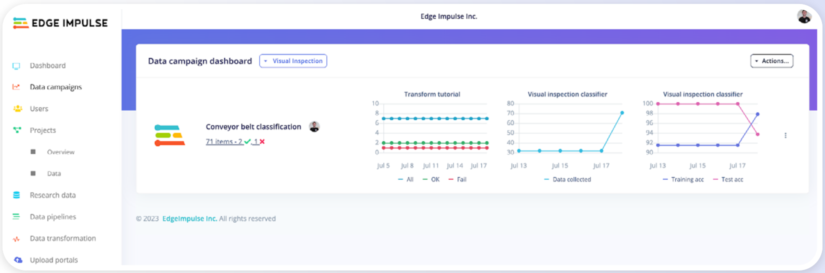
✔ 4. Create on-device deployment (C++ library)

Download the build artefacts from:

https://studio.edgeimpulse.com/v1/api/274435/deployment/download?type=zip&modelType=int8

You're receiving this email because you've been subscribed to notifications for the data source "Run actions". To unsubscribe, go to data sources, click "Edit pipeline" and remove yourself.

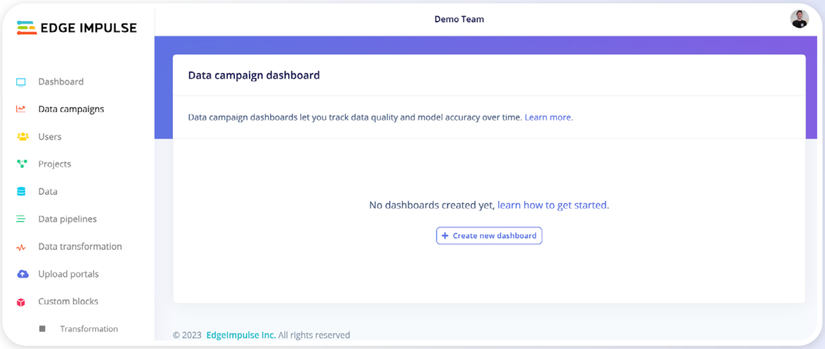
Cheers,
Jan from the Edge Impulse team



Step 1: Setting up your dashboard

(Data campaigns overview)

To get started, navigate to the Data campaigns tab in your organization:



Data campaign dashboard

Data campaign dashboards let you track data quality and model accuracy over time. Learn more.

No dashboards created yet, learn how to get started.

+ Create new dashboard

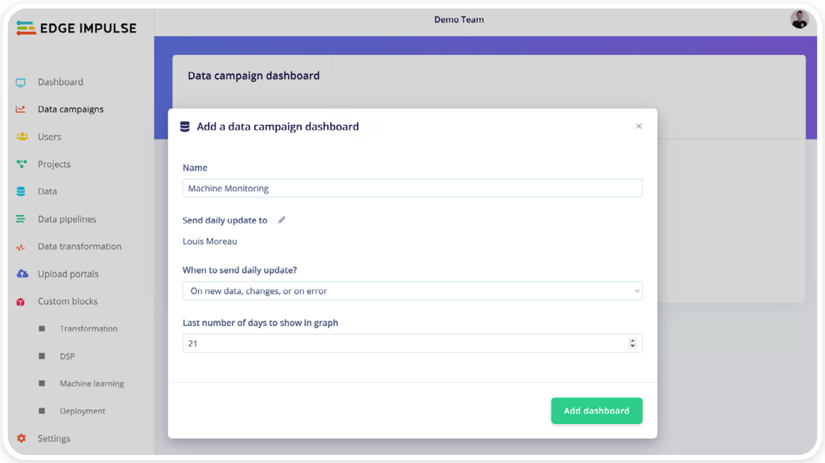
© 2023 EdgeImpulse Inc. All rights reserved

(Data campaigns)

Click on **+ Create new dashboard.**

Give your dashboard a name, and select one or more collaborators to receive the daily updates by email. If you don't want to be spammed, you can select when you want to receive these updates, either *Always, On new data, changes or on error, or Never*. Finally, set the last number of days shown in the graphs.

Add data campaign dashboard



Add a data campaign dashboard

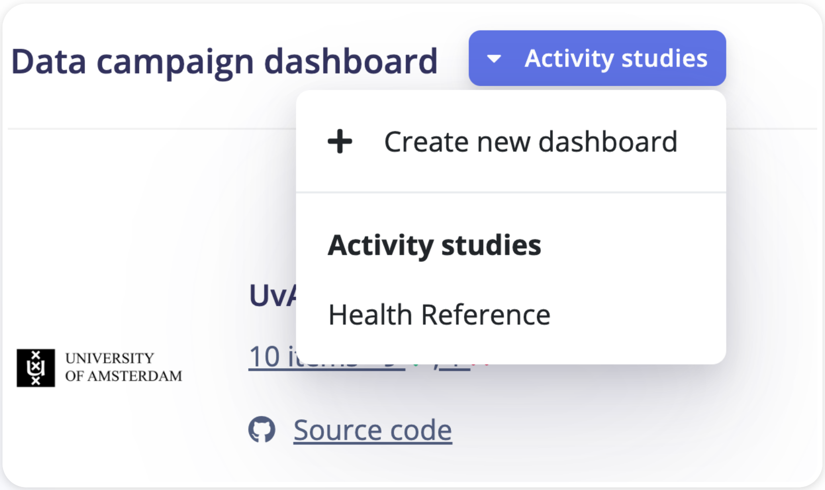
Name: Machine Monitoring

Send daily update to: Louis Moreau

When to send daily update?: On new data, changes, or on error

Last number of days to show in graph: 21

You can create as many dashboards as needed, simply click on **+ Create a new dashboard** from the dropdown available under your current dashboard:

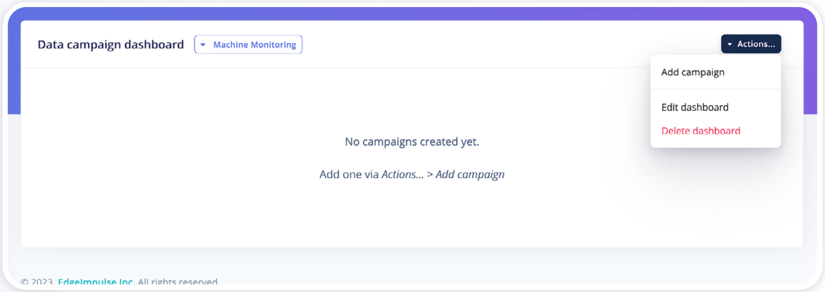


If you want to delete a dashboard, Click on **Actions… → Delete dashboard**

Step 2: Setting up your campaign

Once your dashboard is created, you can add your custom campaigns. It's where you will specify which metrics are important to you and your use case. Click on **Actions… → Add campaign**

Add data campaign



No campaigns created yet.

Add one via *Actions... > Add campaign*

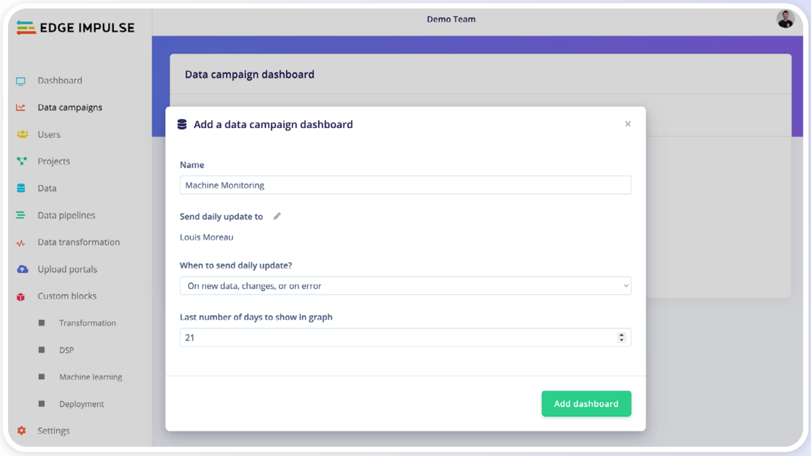
© 2023 EdgeImpulse Inc. All rights reserved.

Step 3: Fill the form to create your campaign

Name: Name of your data campaign.
Description: Description of your data campaign.
Campaign coordinators: Add the collaborators that are engaged with this campaign
Datasets: Select the datasets you want to visualize in your campaign.
You can add several datasets.
Projects: Select the projects you want to visualize in your campaign.
You can add several projects.
Pipelines: Select the pipeline that is associated with your campaign. *Note that this is for reference only, it is currently not displayed in your campaign*
Links: Select between the link type you need. Options are Github, Spreadsheet, Text Document, Code repository, List or Folder. Add a name and the link. This place is useful for other collaborators to have all the needed information about your project, gathered in one place under your campaign.

Additional queries to track:
These queries are data filters that need to be written in the SQL WHERE format. For example metadata->age >= 18` will return the data samples from adult patients.

Save your data campaign and it will be added to your dashboard.



Add a data campaign dashboard

Name: Machine Monitoring

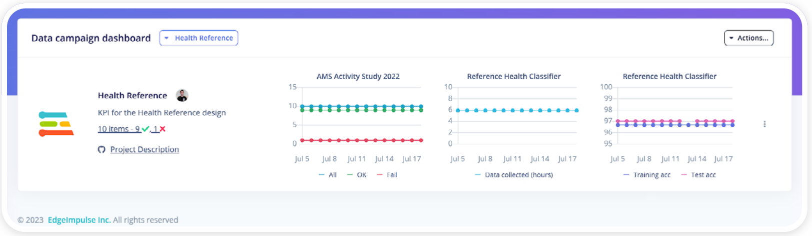
Send daily update to: Louis Moreau

When to send daily update?: On new data, changes, or on error

Last number of days to show in graph: 21

Create or edit data campaign

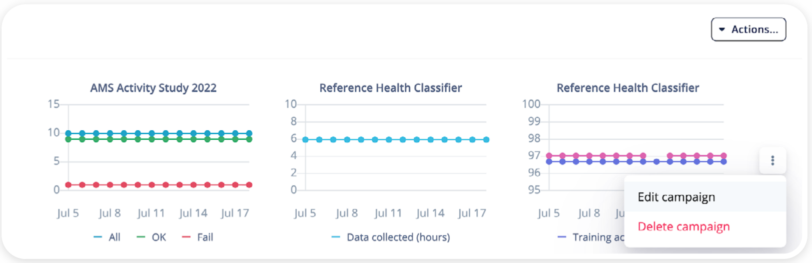
This dashboard shows the metrics' progress from the [Health reference design data.]



Create or edit data campaign

If you want to edit or delete your campaign, click on the "⋮" button on the right side of your campaign:

[Data Campaigns]

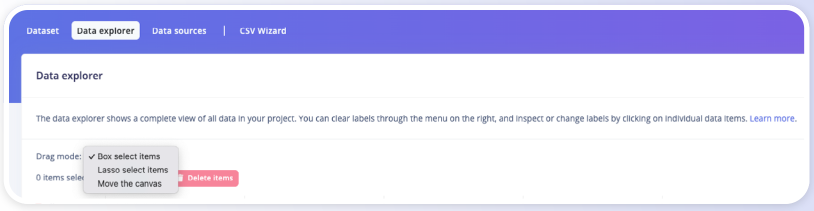


Edit campaign

6. Data Visualization and Exploration Capabilities

As your organization expands, the need for efficiency and scalability becomes ever more critical. Traditional data processing methods struggle to keep pace with the growing volume and complexity of data. *Edge Impulse rises to the challenge by offering dataset and feature exploration graphs, scalable architectures, distributed computing, and parallel processing capabilities.* With these powerful tools, you can analyze vast datasets in a fraction of the time.

Step 1: Navigate to Project → Data acquisition → Data explorer to visualize the dataset

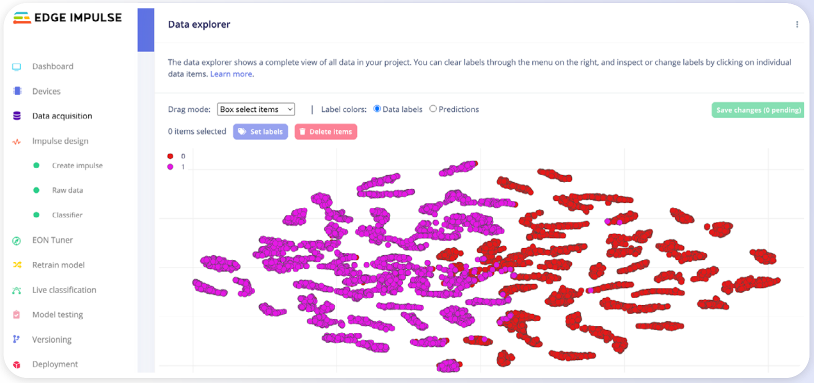


The data explorer shows a complete view of all data in your project. You can clear labels through the menu on the right, and inspect or change labels by clicking on individual data items. Learn more.

Drag mode: ✓ Box select items

0 items sele

In the following screenshot you see the cleaned dataset with a pretty good separation between the labels 0 (not confused) and 1(confused).

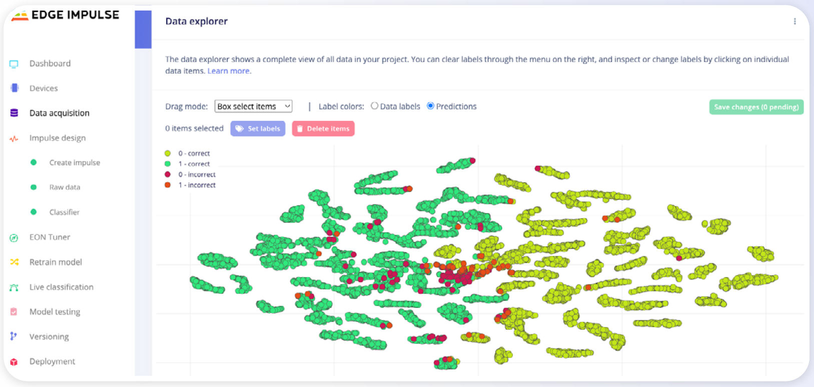


The data explorer shows a complete view of all data in your project. You can clear labels through the menu on the right, and inspect or change labels by clicking on individual data items. Learn more.

Drag mode: Box select items | Label colors: [✓ Data labels] Predictions

0 items selected

In the screenshot below, you see the correct and incorrect predictions made by the model for the two labels.



The data explorer shows a complete view of all data in your project. You can clear labels through the menu on the right, and inspect or change labels by clicking on individual data items. Learn more.

Drag mode: Box select items | Label colors: Data labels [✓ Predictions]

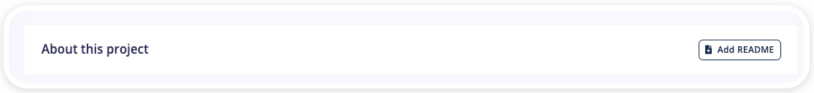
0 items selected

7. Collaboration Capabilities

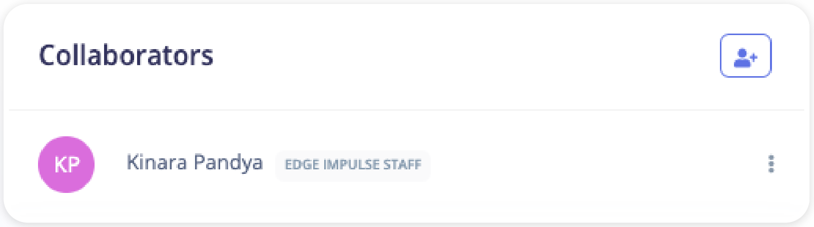
Collaboration and reproducibility are essential pillars of impactful product development and maintenance. *Edge Impulse* offers a unified environment where both embedded and ML teams can work together. It supports deployment of trained models directly onto resource-constrained embedded devices. This feature ensures that the ML models are integrated smoothly into the embedded system, eliminating potential integration challenges and fostering collaboration between the two teams.

Additionally, by documenting the origin of the dataset, data processing steps and transformations within your project's README, *Edge Impulse* fosters reproducible engineering practices, allowing your peers to replicate and validate your findings.

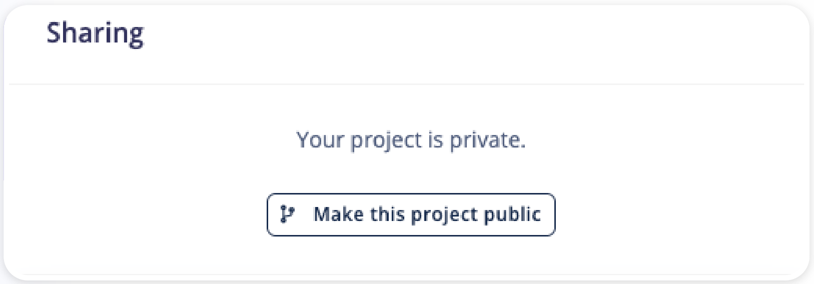
Step 1: Navigate to Project → Dashboard → "Add Readme" and update the project information



Step 2: From the dashboard, add any collaborators you wish to add to the project



From the Dashboard, you may also make the project public (optional).



8. Security and Compliance

Data security and compliance are paramount in the digital health domain, where sensitive PHI *(Private Human information)* must be safeguarded. By allowing for on-device machine learning, Edge Impulse enables data to be processed without the need for transmission, improving data security and privacy. EI is designed to incorporate stringent security measures such as encryption, SOC2 regulatory compliance such as HIPAA, GDPR, access controls, and anonymization techniques.

Please note that our plan is to be HIPAA certified by Q2 2024.

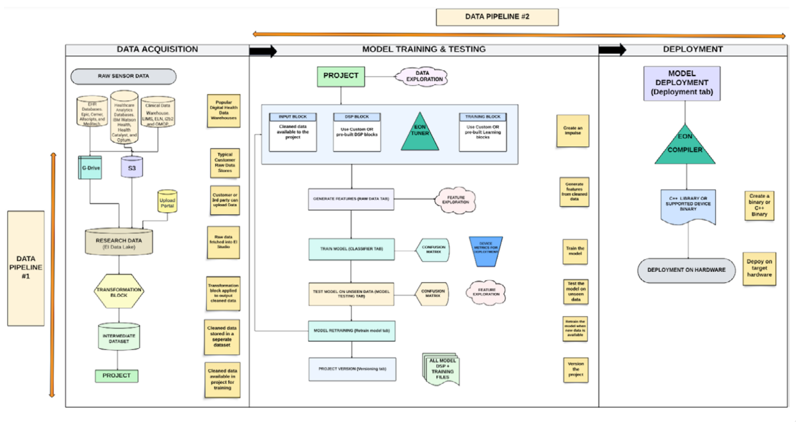
Security and Compliance

Additional Resources

The Edge Impulse Python SDK is a library to help you develop machine learning (ML) applications for edge and Internet of Things (IoT) devices. While the Edge Impulse Studio is a great interface for guiding you through the process of collecting data and training a model, the **edgeimpulse** Python SDK allows you to programmatically Bring Your Own Model (BYOM), developed and trained on any platform.

Edge Impulse Python SDK

Summary


DATA PIPELINE #2
DATA ACQUISITION
MODEL TRAINING & TESTING
DEPLOYMENT
DATA PIPELINE #1
RAW SENSOR DATA
EHR Databases. Epic, Cerner, Allscripts, and Meditech.
Healthcare Analytics Databases. IBM Watson Health, Health Catalyst, and Optum.
Clinical Data Warehouse. LIMS, ELN, i2b2 and OMOP.
Popular Digital Health Data Warehouses
G-Drive
S3
Typical Customer Raw Data Stores
Upload Portal
Customer or 3rd party can upload Data
RESEARCH DATA (EI Data Lake)
Raw data fetched into EI Studio
TRANSFORMATION BLOCK
Transformation block applied to output cleaned data
INTERMEDIATE DATASET
Cleaned data stored in a seperate dataset
PROJECT
Cleaned data available in project for training
PROJECT
DATA EXPLORATION
INPUT BLOCK
Cleaned data available to the project
DSP BLOCK
Use Custom OR pre-built DSP blocks
EON TUNER
TRAINING BLOCK
Use Custom OR pre-built Learning blocks
Create an impulse
GENERATE FEATURES (RAW DATA TAB)
FEATURE EXPLORATION
Generate features from cleaned data
TRAIN MODEL (CLASSIFIER TAB)
CONFUSION MATRIX
DEVICE METRICS FOR DEPLOYMENT
Train the model
TEST MODEL ON UNSEEN DATA (MODEL TESTING TAB)
CONFUSION MATRIX
FEATURE EXPLORATION
Test the model on unseen data
MODEL RETRAINING (Retrain model tab)
Retrain the model when new data is available
PROJECT VERSION (Versioning tab)
ALL MODEL DSP + TRAINING FILES
Version the project
MODEL DEPLOYMENT (Deployment tab)
EON COMPILER
C++ LIBRARY OR SUPPORTED DEVICE BINARY
Create a binary or C++ Binary
DEPLOYMENT ON HARDWARE
Depoy on target hardware

Edge Impulse's components are designed to address the different stages of a typical machine learning (ML) pipeline. The arrangement of these components reflects the common process of data collection, data cleaning/ transformation, feature extraction, model training, testing, and deployment.

By mirroring the standard stages of the ML pipeline, Edge Impulse ensures users have all the tools they need at each stage. This integrated approach enables anyone who is a beginner or an expert to seamlessly build and deploy ML models.

The "glue" that connects all these components together in Edge Impulse is its unified software platform. Because of this centralized interface, users can manage each stage of the machine learning workflow right from data ingestion to model deployment. All the components are organized in a cohesive manner/ tightly integrated on the platform's user interface. For example, data gathered and processed in one stage of the workflow is seamlessly available for the next, and so on. Additionally, Edge Impulse provides APIs and SDKs for integrating its platform with other tools and systems. This enables users to customize their workflows and use external tools when necessary.

The real power of Edge Impulse lies in its modular and flexible architecture. You can modify and iterate over your impulse design, data, and model as many times as you want until you achieve a satisfactory result.

Moreover, by incorporating collaboration tools and a focus on security and compliance, Edge Impulse addresses key considerations for teams working in the digital health industry.

About the Author

Kinara Pandya is a seasoned Machine Learning Solutions Engineer with a background in Computer Science, Data Science, and Biotechnology. With experience spanning digital health solutions and the life sciences industry, Kinara has shared insights on next-gen digital health technologies, including “Advancing Health with Edge AI”, at prominent conferences like HLTH in Las Vegas. Additionally, her expertise is showcased in her contributions to a report published on **MedRxiv**.