

tinyML[®] EMEA

Enabling Ultra-low Power Machine Learning at the Edge

June 26 - 28, 2023



www.tinyML.org

```
pip install edgeimpulse
```

A programmatic approach to automate your MLOps Pipelines



Louis Moreau

Developer Relations Lead Engineer, Edge Impulse

June 27, 2023

Introduction

We started our journey by building tools to ease AI for embedded engineers

EDGE IMPULSE

- Dashboard
- Devices
- Data sources
- Data acquisition
- Impulse design
 - Create impulse
 - Image
 - NN Classifier
- EON Tuner
- Retrain model
- Live classification
- Model testing
- Performance calibration
- Versioning
- Deployment

GETTING STARTED

- Documentation
- Forums

Deploy your impulse

You can deploy your impulse to any device. This makes the model run without an internet connection, minimizing consumption. [Read more.](#)

Create library

Turn your impulse into optimized source code that you can run on any device.



C++ library



Arduino library



WebAssembly



TensorRT library



Tensai Flow library

brainchip
MetaTF Model

Meta TF Model

BETA



Simplicity Studio Component



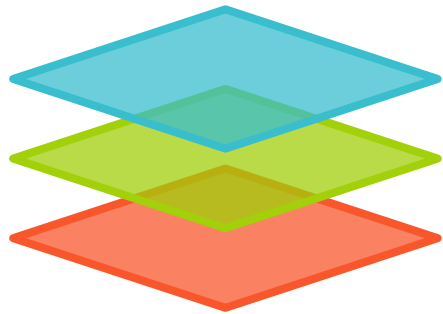
OpenMV library

Now we also build tools for domain experts to deploy models on edge devices...





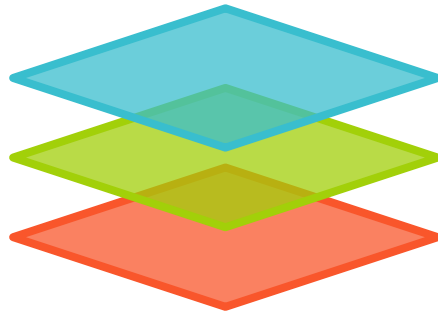
EDGE IMPULSE



Python SDK



EDGE IMPULSE



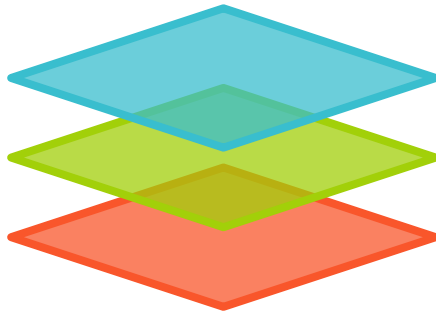
Python SDK

`profile()`

```
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms
```



EDGE IMPULSE



Python SDK

`profile()`

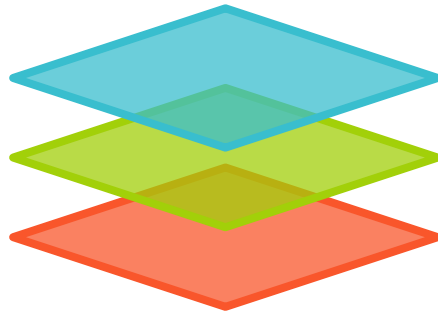
```
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms
```

`deploy()`





EDGE IMPULSE



Python SDK

`profile()`

```
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms
```

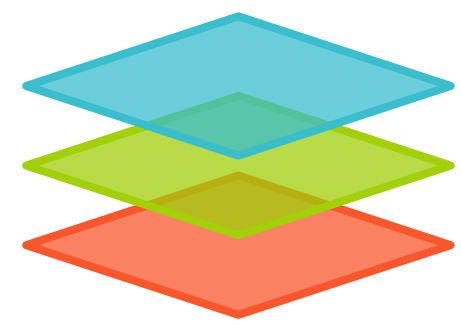
`deploy()`




TensorFlow

 Keras

 **EDGE IMPULSE**



 Python SDK

`profile()`

```
target: cortex-m4f-80mhz  
RAM: 39.1 kB  
flash: 37.6 kB  
latency: 145 ms
```

`deploy()`

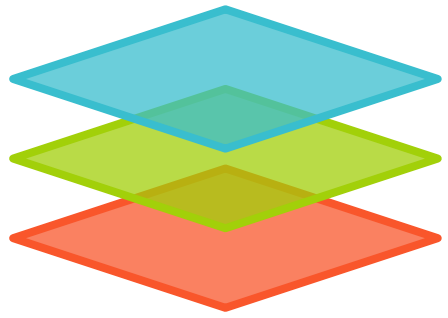



TensorFlow

 Keras


ONNX

 **EDGE IMPULSE**



 Python SDK

`profile()`

```
target: cortex-m4f-80mhz  
RAM: 39.1 kB  
flash: 37.6 kB  
latency: 145 ms
```

`deploy()`




TensorFlow

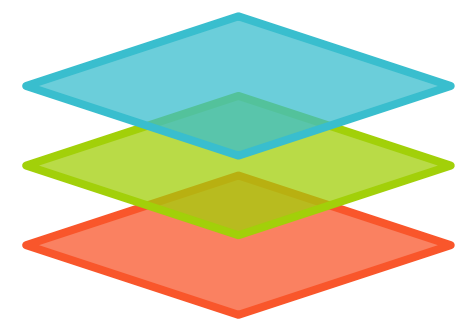
 Keras


ONNX

 PyTorch


APACHE
Spark™

 **EDGE IMPULSE**



 Python SDK

`profile()`

```
target: cortex-m4f-80mhz
RAM: 39.1 kB
flash: 37.6 kB
latency: 145 ms
```

`deploy()`



Edge Impulse Python SDK

How it works?





+ Code + Text

Connect ▾



▼ Configure the Edge Impulse Python SDK

```
!pip install edgeimpulse
```

```
[ ] import edgeimpulse as ei  
  
    ei.API_KEY = api_key
```

▼ Profile

```
[ ] profile = ei.model.profile(model="./my_model/",  
                               device='cortex-m4f-80mhz')  
  
    print(profile.summary())
```

▼ Deploy

```
[ ] deploy = ei.model.deploy(model="./my_model/",  
                              model_output_type=ei.model.output_type.Classification(),  
                              deploy_target="zip",  
                              output_directory=".")
```



+ Code + Text

Connect



▼ Configure the Edge Impulse Python SDK

```
[ ] !pip install edgeimpulse
```

```
▶ import edgeimpulse as ei  
  
ei.API_KEY = api_key
```



▼ Profile

```
[ ] profile = ei.model.profile(model="./my_model/",  
                               device='cortex-m4f-80mhz')  
  
print(profile.summary())
```

▼ Deploy

```
[ ] deploy = ei.model.deploy(model="./my_model/",  
                              model_output_type=ei.model.output_type.Classification(),  
                              deploy_target="zip",  
                              output_directory="./")
```



+ Code + Text

Connect



▼ Configure the Edge Impulse Python SDK

```
[ ] !pip install edgeimpulse
```

```
[ ] import edgeimpulse as ei

    ei.API_KEY = api_key
```

▼ Profile

```
▶ profile = ei.model.profile(model="./my_model/",
                             device='cortex-m4f-80mhz')

print(profile.summary())
```



▼ Deploy

```
[ ] deploy = ei.model.deploy(model="./my_model/",
                              model_output_type=ei.model.output_type.Classification(),
                              deploy_target="zip",
                              output_directory=".")
```

Example - Profile

```
m [43] profile = ei.model.profile(model="saved_model_float32.zip",  
                                device='cortex-m7-216mhz')  
  
print(profile.summary())
```

Target results for float32:

=====

```
{  
  "device": "cortex-m7-216mhz",  
  "tfliteFileSizeBytes": 863312,  
  "isSupportedOnMcu": true,  
  "memory": {  
    "tflite": {  
      "ram": 399257,  
      "rom": 927576,  
      "arenaSize": 398905  
    },  
    "eon": {  
      "ram": 328776,  
      "rom": 882432  
    }  
  },  
  "timePerInferenceMs": 75  
}
```




+ Code + Text

Connect ▾



▼ Configure the Edge Impulse Python SDK

```
[ ] !pip install edgeimpulse
```

```
[ ] import edgeimpulse as ei

    ei.API_KEY = api_key
```

▼ Profile

```
[ ] profile = ei.model.profile(model="./my_model/",
                               device='cortex-m4f-80mhz')

    print(profile.summary())
```

▼ Deploy

```
▶ deploy = ei.model.deploy(model="./my_model/",
                            model_output_type=ei.model.output_type.Classification(),
                            deploy_target="zip",
                            output_directory=".")
```



Example - Quantize & deploy

▼ Deploy

```
[35] # Generate the representative data for the quantized model
import glob, cv2
import numpy as np
X_data = []
files = glob.glob ("test-set/*.jpg")
for myFile in files:
    image = cv2.imread (myFile)
    resized = cv2.resize(image, (96,96), interpolation = cv2.INTER_AREA)
    X_data.append (resized)

print('X_data shape:', np.array(X_data).shape)
```

```
X_data shape: (36, 96, 96, 3)
```

Example - Quantize & deploy

✓
54s

```
[42] # Set model information, such as your list of labels
deploy_filename = "generated_cpp.zip"
labels = ['cotton stem', 'epidermis onion', 'housefly legs', 'unknown', 'wood stem']
model_output_type = ei.model.output_type.Classification(labels=labels)

# Create C++ library with trained model
deploy_bytes = None
try:

    deploy_bytes = ei.model.deploy(model="saved_model.zip",
                                   model_output_type=model_output_type,
                                   deploy_model_type="int8",
                                   representative_data_for_quantization=np.array(X_data, dtype="float32"),
                                   engine="tflite-eon",
                                   deploy_target="zip",
                                   output_directory=".")

except Exception as e:
    print(f"Could not deploy: {e}")

# Write the downloaded raw bytes to a file
if deploy_bytes:
    with open(deploy_filename, 'wb') as f:
        f.write(deploy_bytes)
```

✓
0s

```
[44] !ls

generated_cpp.zip  sample_data  saved_model_float32.zip  test-set
```

Need more?

The Python SDK is built on top of the [Edge Impulse Python API bindings](#), the `edgeimpulse-api` package.

These are Python wrappers for all of the [web API](#) calls available to interact with Edge Impulse projects programmatically (i.e. without needing to use the Studio graphical interface).

Need more?

```
[ ] python -m pip install edgeimpulse-api
```

```
[ ] from edgeimpulse_api import Configuration, ApiClient, ProjectsApi

# Settings
host = "https://studio.edgeimpulse.com/v1"
api_key = "ei_dae2..."

# Create a client object that can connect to our project
config = Configuration(host=host, api_key={"ApiKeyAuthentication": api_key})
client = ApiClient(config)

# Get info about the project
projects = ProjectsApi(client)
project_list = projects.list_projects()
print(project_list.projects[0])
```

Edge Impulse Python SDK

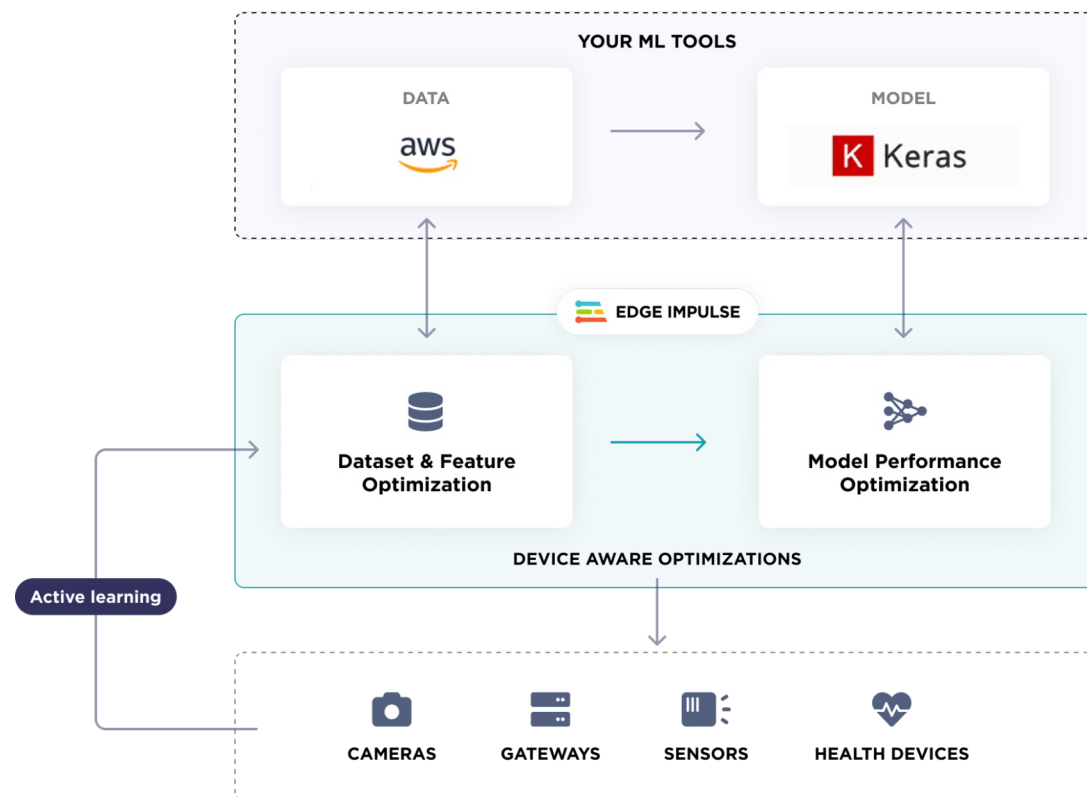
**Integrations and
automation**



Integrations & automation

Designed to help ML practitioners with every stage of their workflow

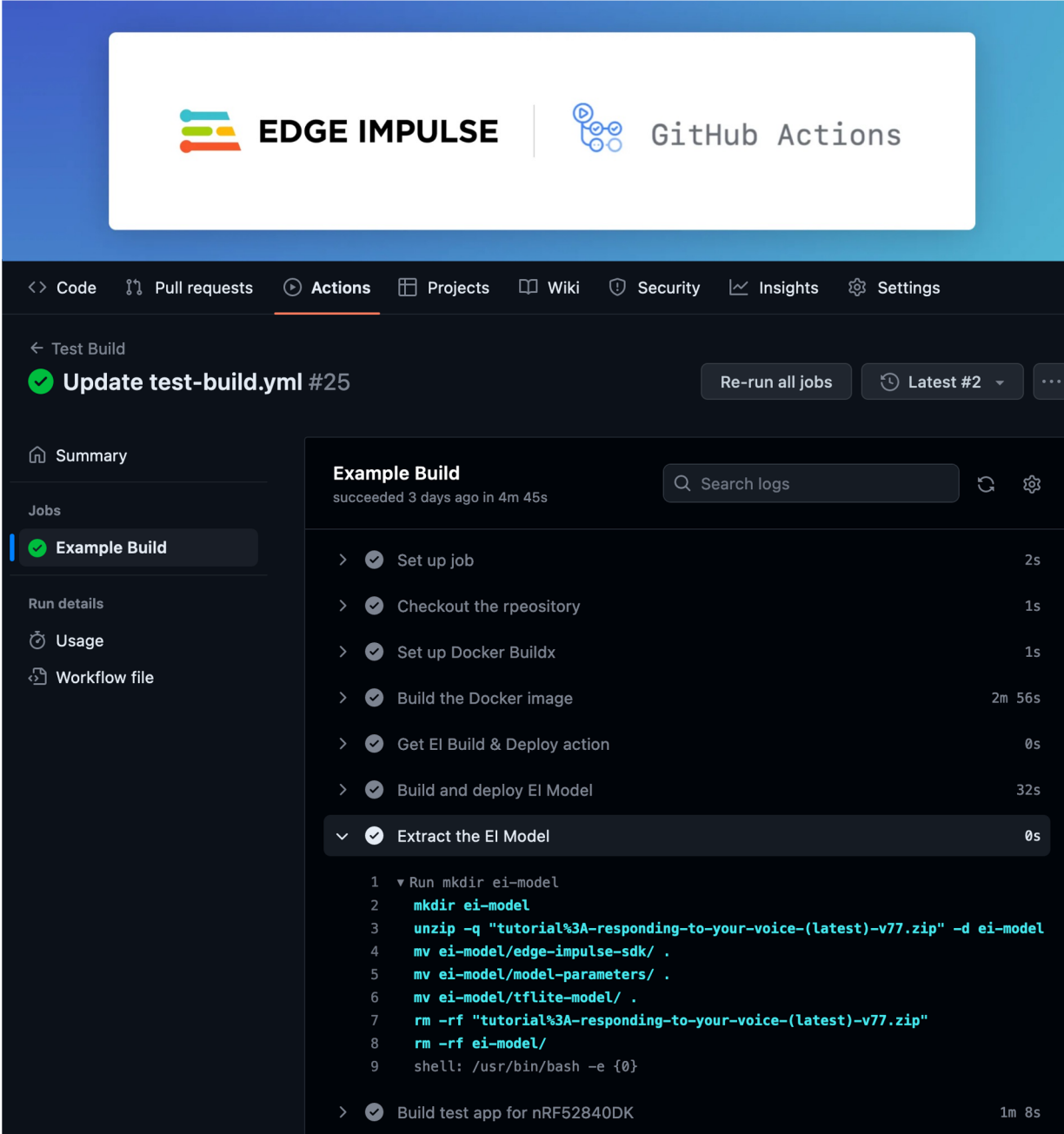
- Integrates easily in **existing ML workflows**
- Unlocks **feature engineering** and fasten feedback loops
- Empowers **model optimization** and deployment to any device



Integrations & automation

CI/CD pipelines

- CI/CD is one of the critical factors for delivering fully tested and up-to-date software or firmware.
- We developed a **GitHub Action** to easily profile, build and deploy your Edge Impulse model



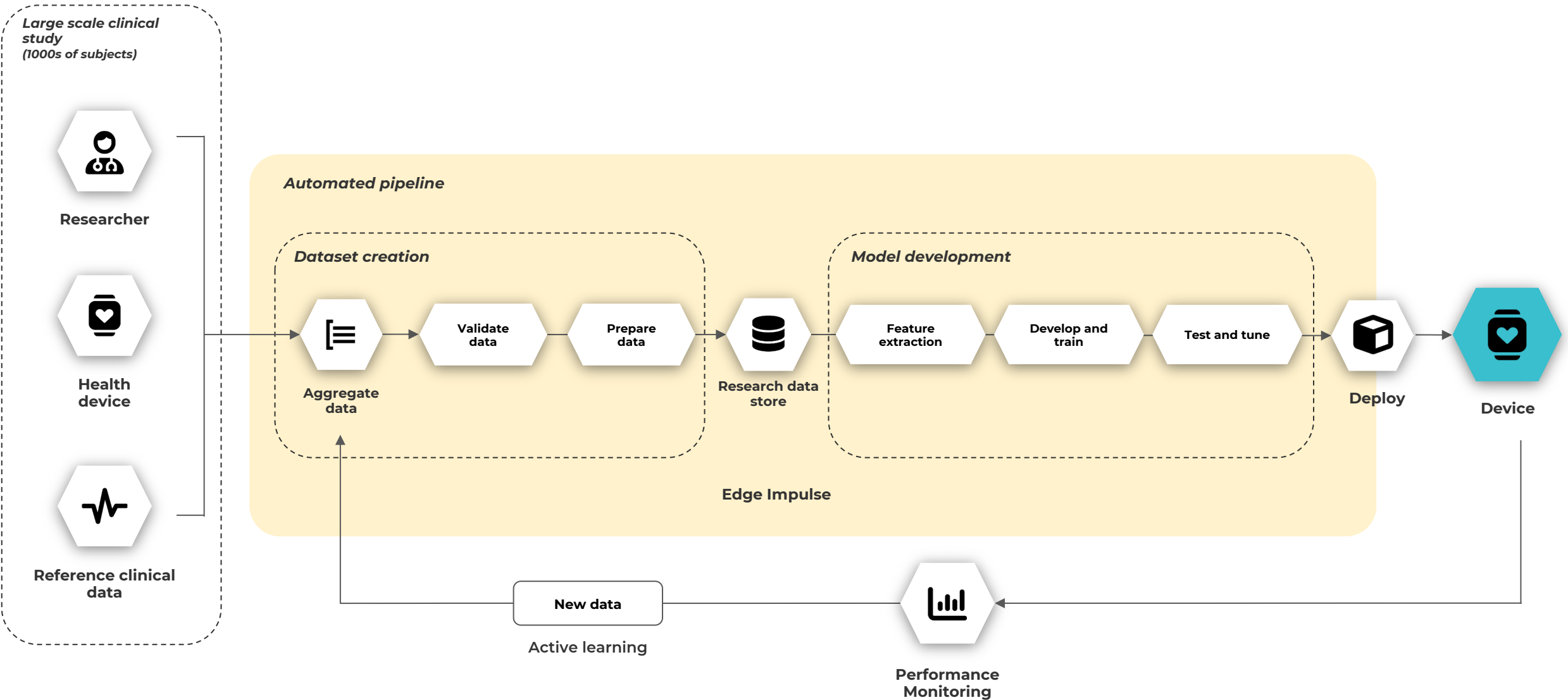
The screenshot displays the GitHub Actions interface for a workflow named 'Test Build'. The workflow is currently running, and the selected job is 'Update test-build.yml #25'. The interface shows a list of jobs and a detailed view of the 'Example Build' job, which has succeeded 3 days ago in 4m 45s. The job steps are as follows:

- Set up job (2s)
- Checkout the repository (1s)
- Set up Docker Buildx (1s)
- Build the Docker image (2m 56s)
- Get EI Build & Deploy action (0s)
- Build and deploy EI Model (32s)
- Extract the EI Model (0s)
- Build test app for nRF52840DK (1m 8s)

The 'Extract the EI Model' step is expanded, showing the following shell commands:

```
1 Run mkdir ei-model
2 mkdir ei-model
3 unzip -q "tutorial%3A-responding-to-your-voice-(latest)-v77.zip" -d ei-model
4 mv ei-model/edge-impulse-sdk/ .
5 mv ei-model/model-parameters/ .
6 mv ei-model/tflite-model/ .
7 rm -rf "tutorial%3A-responding-to-your-voice-(latest)-v77.zip"
8 rm -rf ei-model/
9 shell: /usr/bin/bash -e {0}
```


Health ML automation example



Edge Impulse Python SDK

Recap



Convert Python Models into Optimized C++

- Profile on-device performance of any trained model
- Analyze the impact of architectural decisions
- Generate optimized C++ libraries
- Deploy to edge devices



Copyright Notice

This presentation in this publication was presented as a tinyML® EMEA Innovation Forum. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org