Tech Talks LIVE Schedule – Presentation will begin shortly

tech talks

Wireless Connectivity Tech Talks

Thursday, July 15th	**Get to Know OpenThread Resources and Examples**
Thursday, August 26th	**Understand the Benefits of Wi-SUN for Long Range Industrial Applications**
Wednesday, September 29th	**Learn to use Machine Learning for Predicative Maintenance**

Recording and slides will be posted to:
www.silabs.com/training

We will begin ir 3:00

Speaker

水谷章成（Aki Mizutani）

Sr. FAE, Japan

SILICON LABS

tech talks

WELCOME

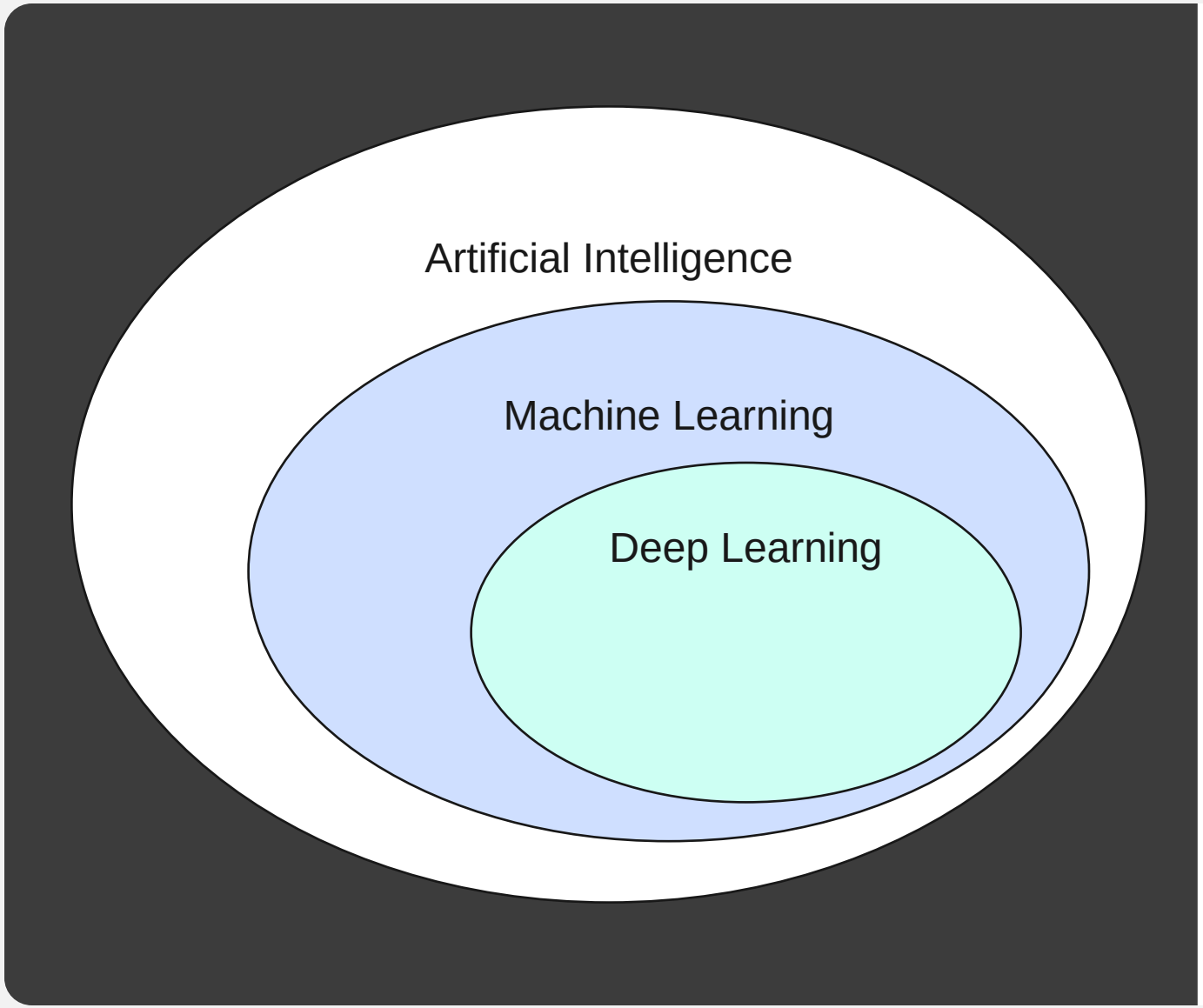
Learn to add Speech Recognition with Machine Learning

Aki Mizutani

Agenda

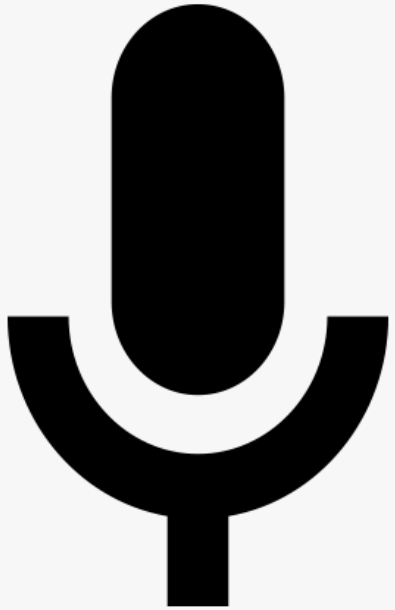
- A brief introduction to Machine Learning on Embedded Devices
- Introduction to Edge Impulse and a high level overview on how to use their platform to create a neural network
- Demonstration: “wireless gecko” running on a Thunderboard Sense 2
- Wrap-up and Q&A

Artificial Intelligence, Machine Learning, and Deep Learning



- **Artificial Intelligence: Broadly defined as the effort to automate intellectual tasks normally performed by humans**
- **Machine Learning: A system which outputs predictions based on previous observations**
- **Deep Learning: A subset of Machine Learning in which the learning algorithmic architecture is based on approixmations of how a human brain might work.**

Embedded Machine Learning Use Cases



AUDIO

Wake Word / Key Phrase Detection

Glass Breaking

Intrusion

SENSING

Motion Sensing

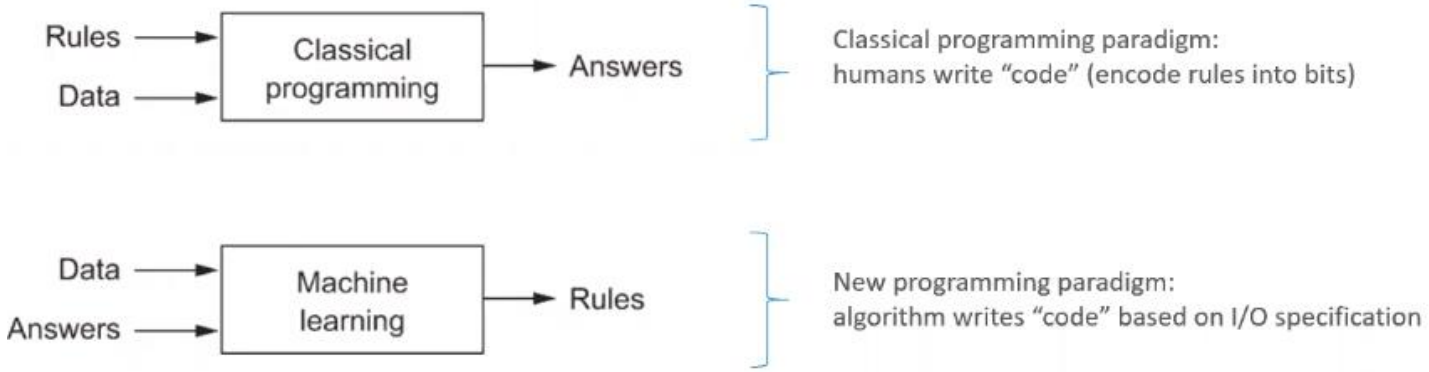
Intelligent Sensor Fusion

INDUSTRIAL

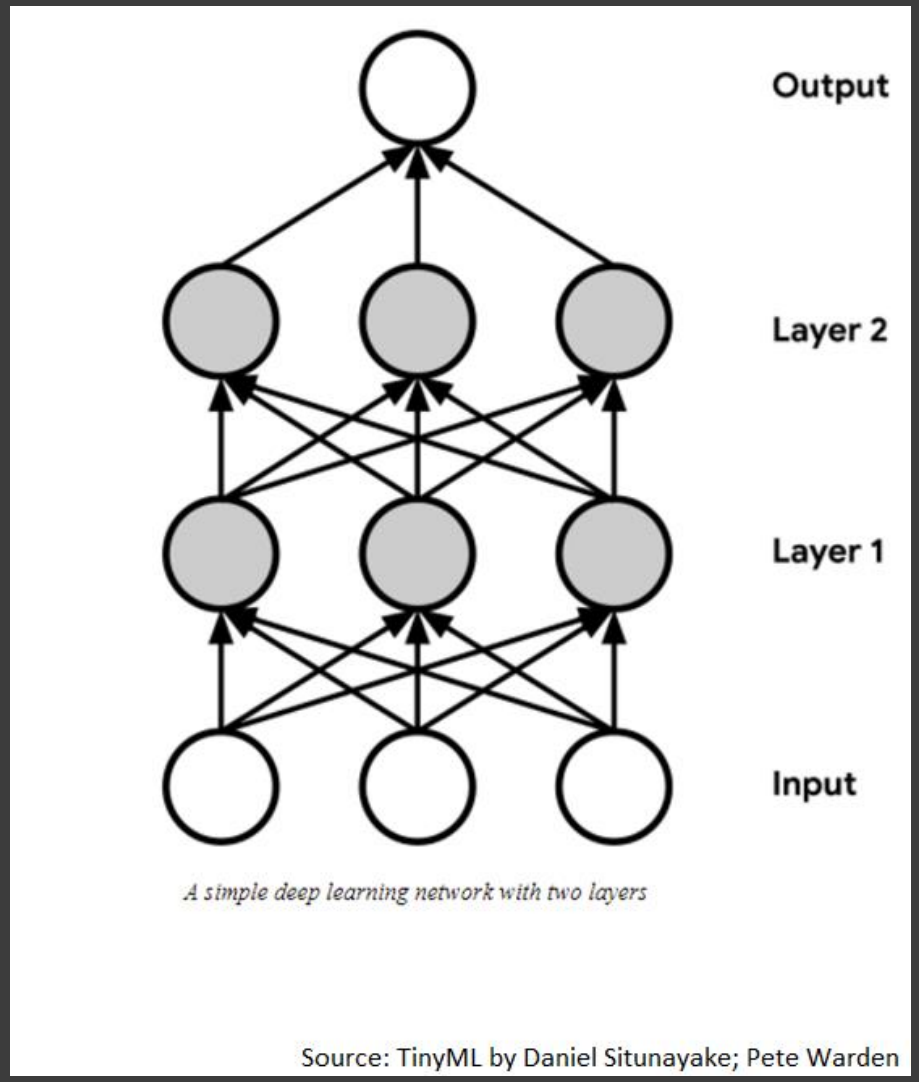
Predictive Maintenance

What is Machine Learning?

- **Definition - Technique for using computers to predict things based on past observations**
- **Collect data – paired input/output**
- **Design Model**
  - Create a computer program that analyzes that data, learns patterns
  - Computer uses the analysis to predict future states from new input
- **Training: Machine Learning model is trained rather than explicitly programmed with rules**
  - Model is presented with many examples of input/output sets
- **Inference: Model finds statistical structure in these examples and produces rules for automating the task**



Train the Model

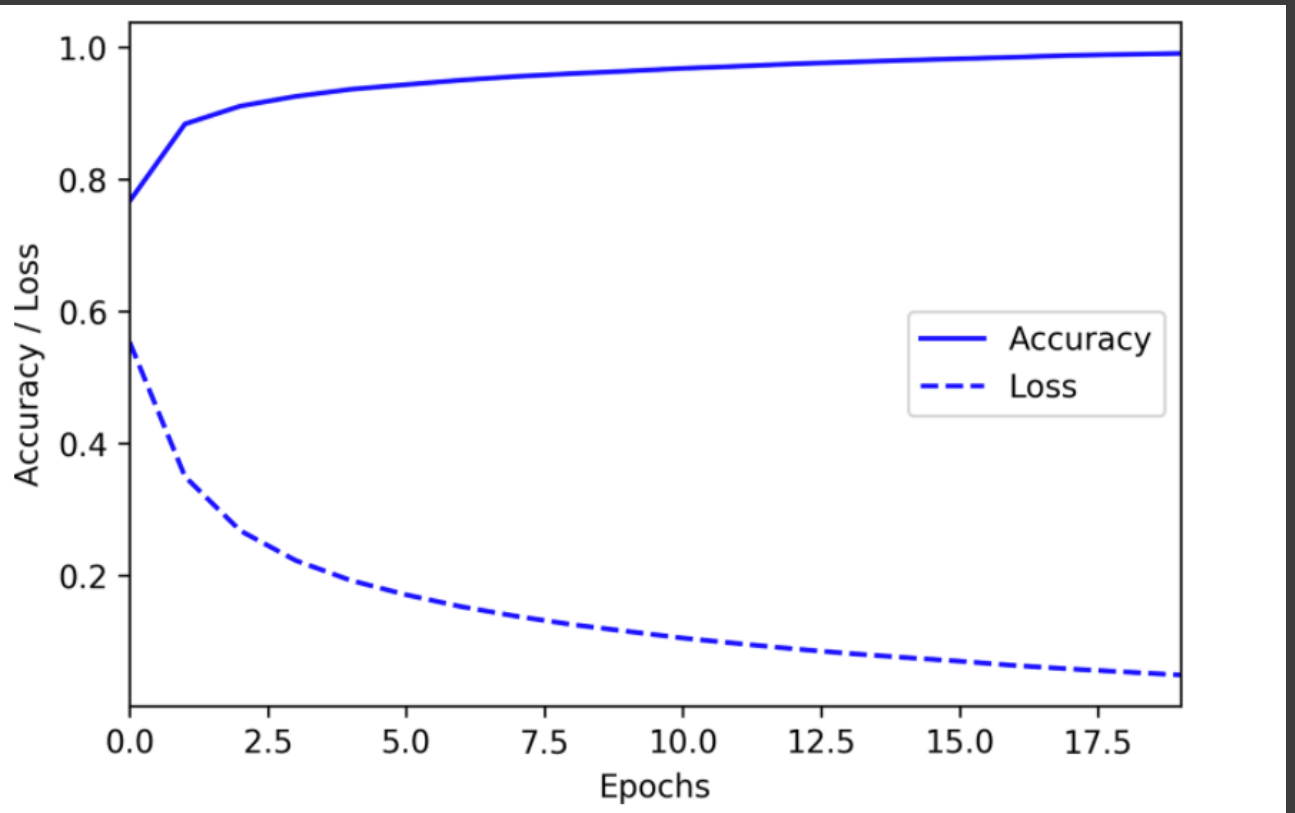


A simple deep learning network with two layers

Source: TinyML by Daniel Situnayake; Pete Warden

- **Selecting Data**
  - Train your model only using information relevant to solving the problem
- **Collecting Data**
  - The more, the more varied, the better!
  - 80% for training; 20% for testing
- **Labelling Data**
  - Needed to classify the data
- **The idea is to feed training data through a model and make small adjustments**
  - Weights and biases of each node are initialized with random values
- **Training is done in iterations (or epochs) via an algorithm called backpropagation**
  - Weights and biases are adjusted iteratively
  - Stops once the model's performance stops improving: The model "converges"

Training the Model Cont'd – Accuracy and Loss



A graph showing model convergence during training

Source: TinyML by Daniel Situnayake; Pete Warden

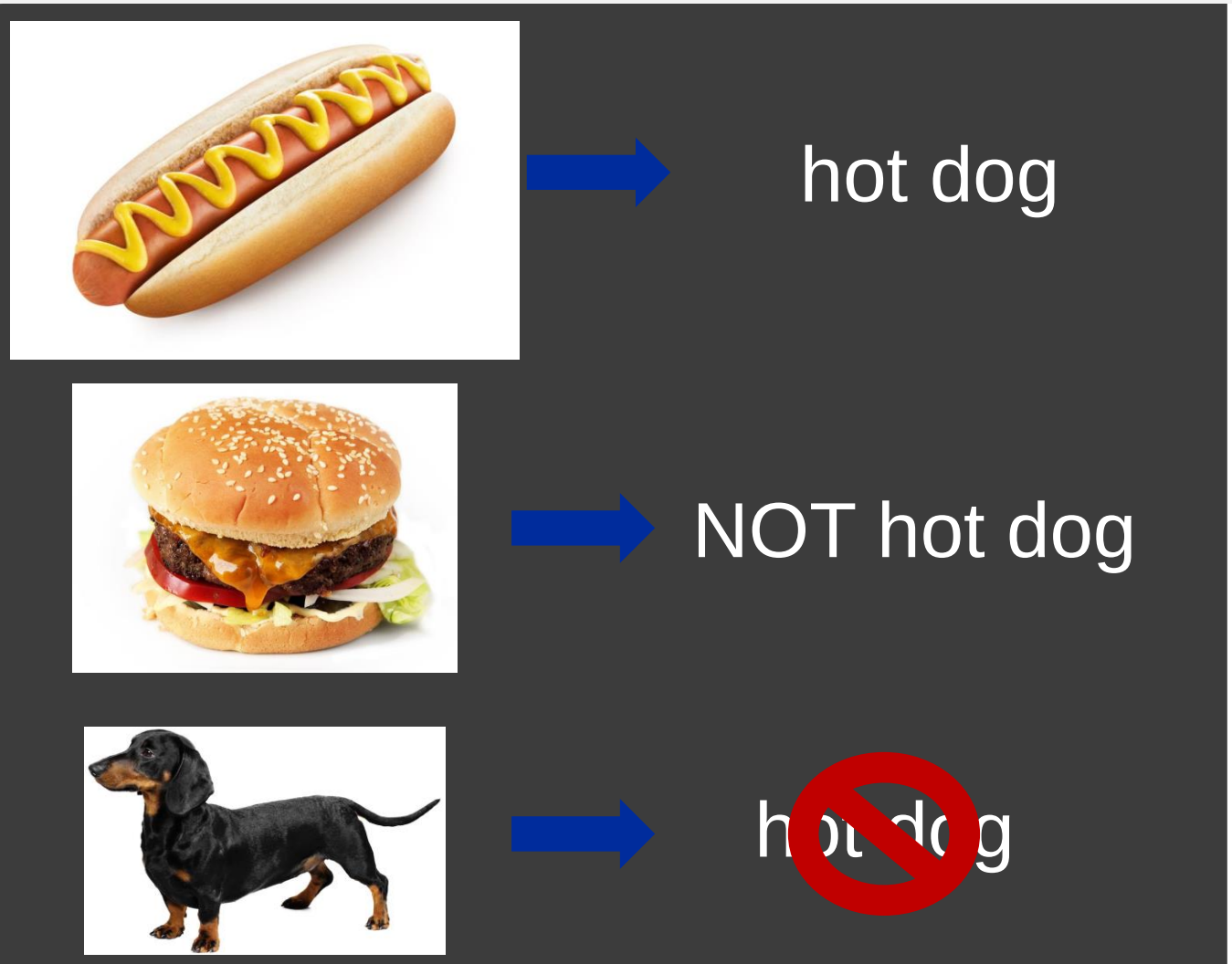
- **How to determine if a model has "converged"?**
  - Loss and Accuracy
- **Loss: gives a numerical estimate of how far the model is from producing the expected answers**
- **Accuracy: percentage of time that the model chooses the correct prediction**
- **Perfect Model: Loss of 0.0 and Accuracy of 100%**
- **As training progresses, accuracy increases and loss is reduced until the model no longer improves**

Training Data Sets

Hot Dog

Not Hot Dog (Unknown)

Testing and Inferencing

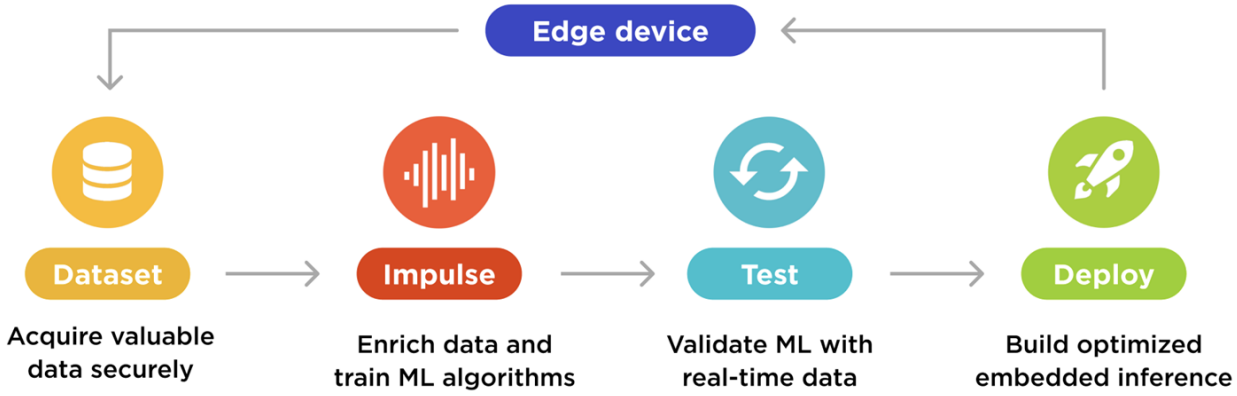


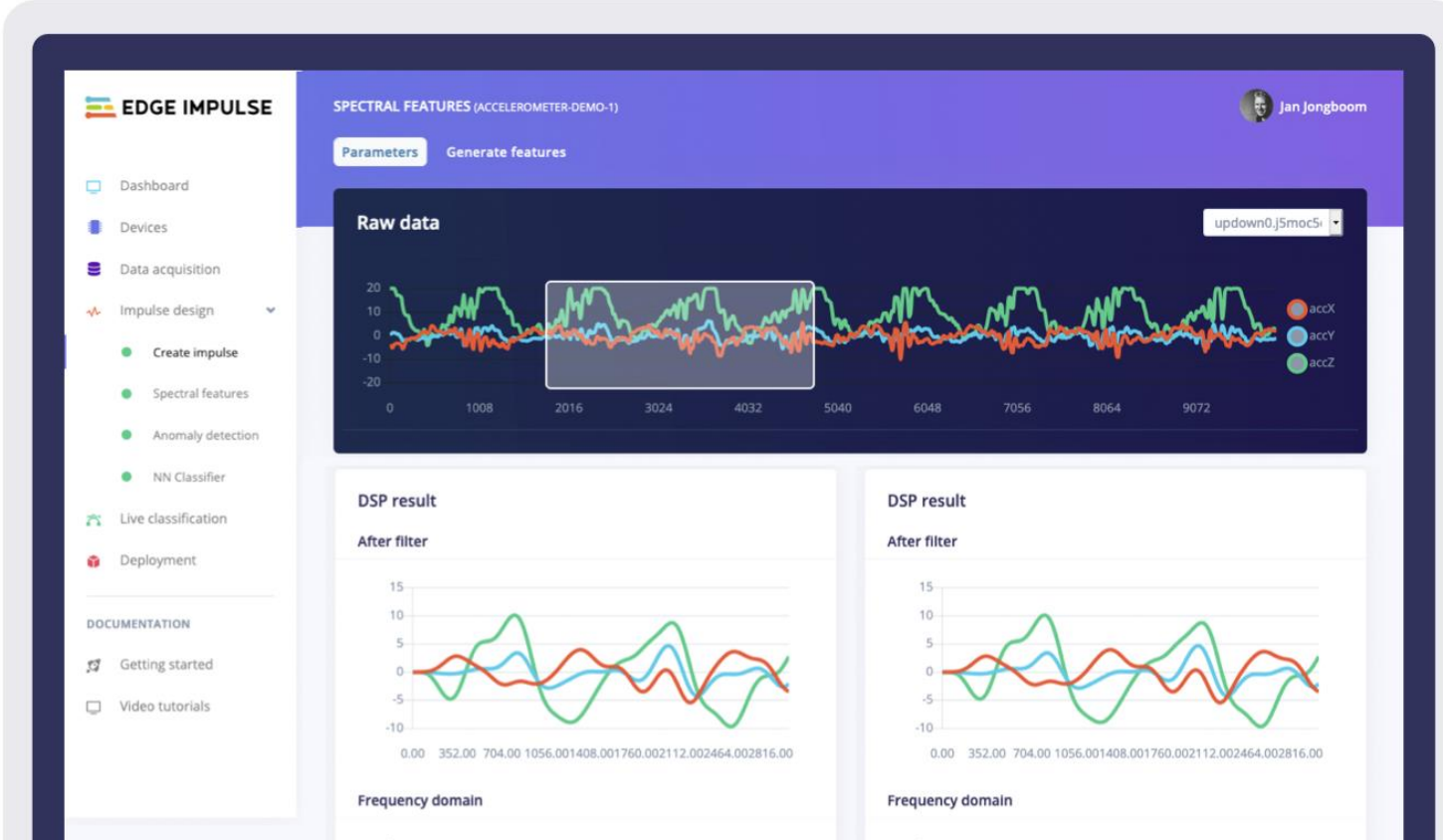
- **Reserve 20% of your data set for testing**
- **A working model will won't be perfect, but should return acceptable results a high percentage of the time**
- **Poor results?**
  - Check the input dataset for miscategorized data
  - Provide more input data
  - 30+ minutes of voice samples not uncommon for typical voice recognition

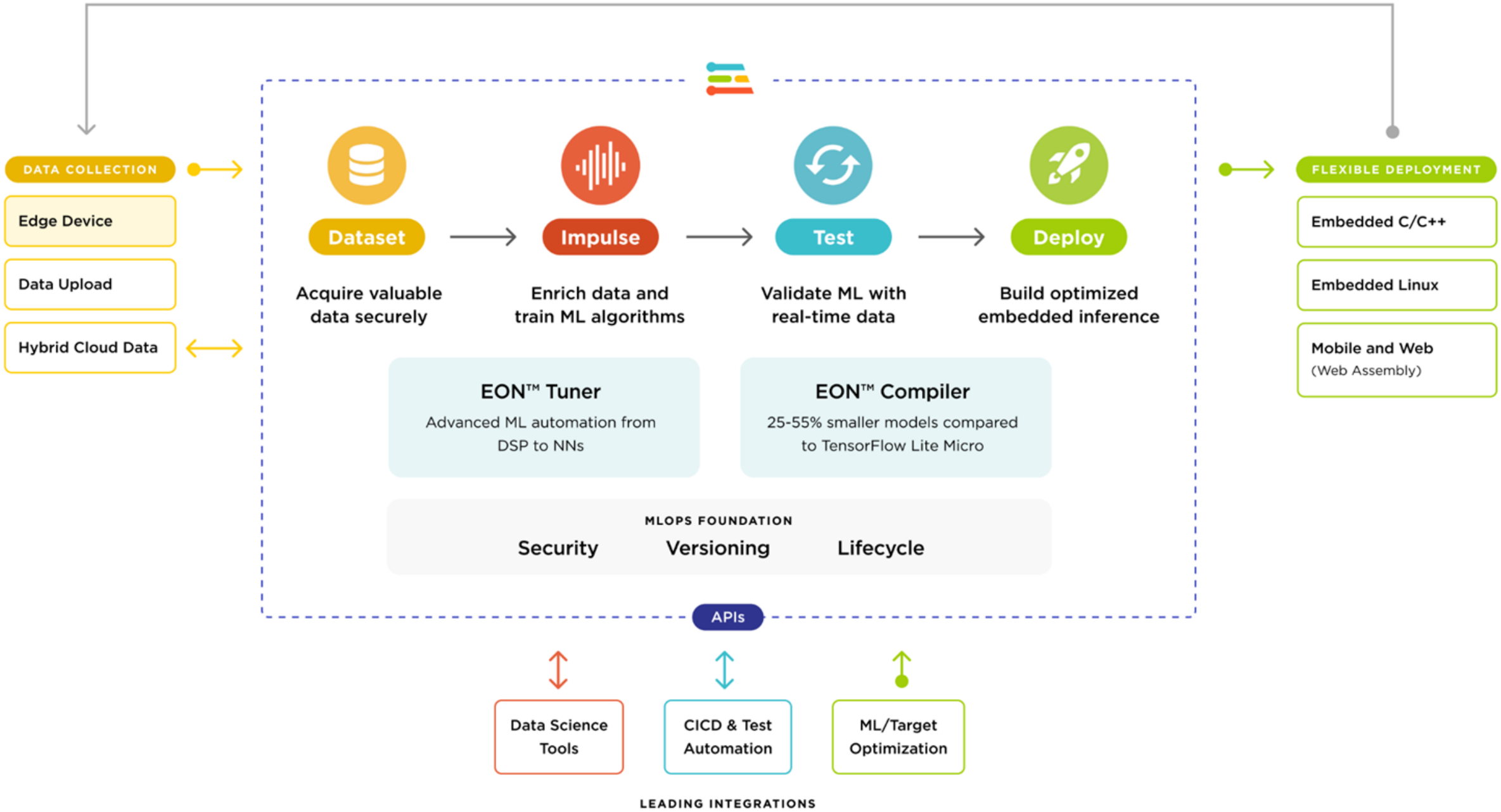
EDGE IMPULSE

The leading embedded ML platform

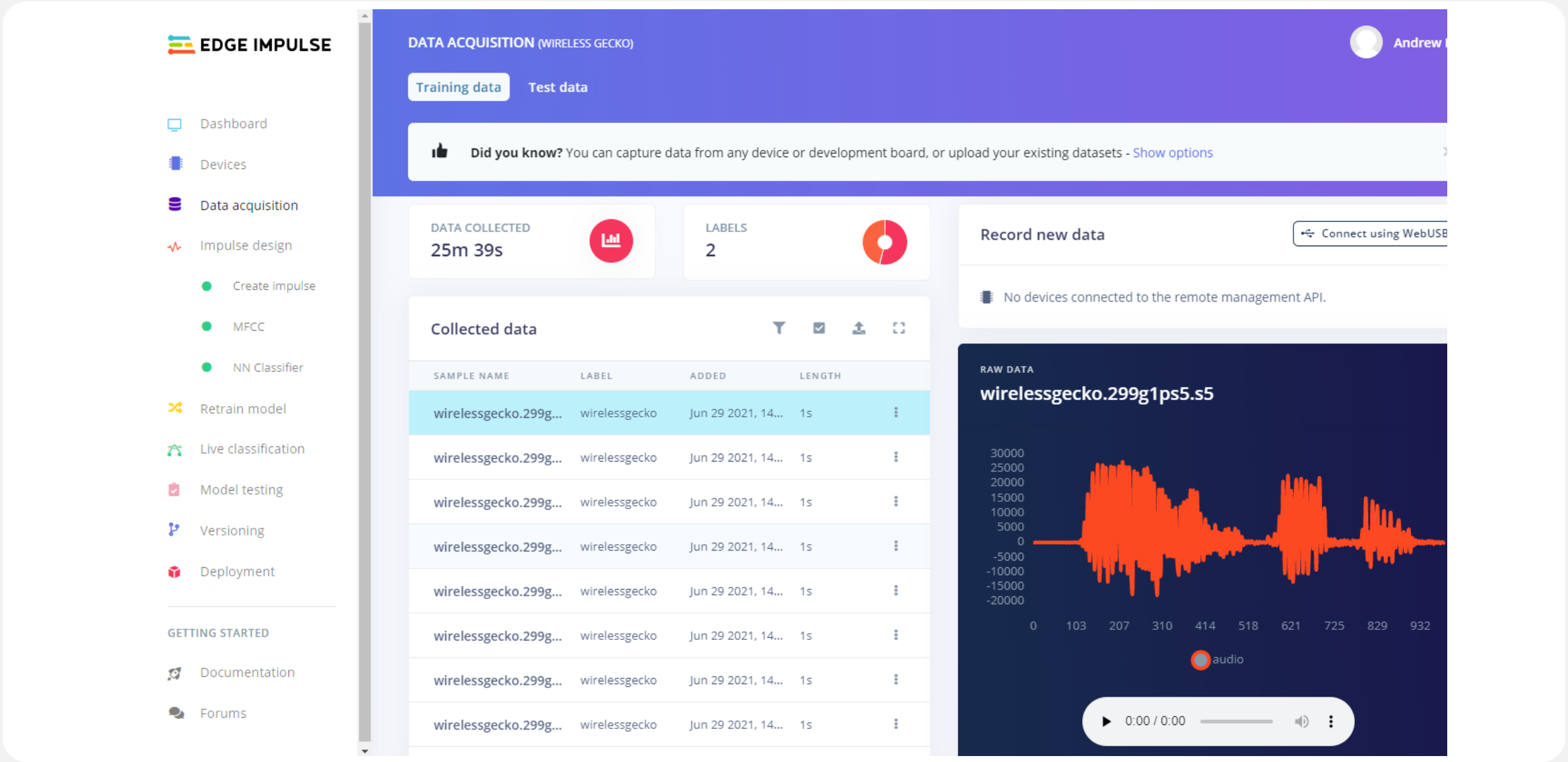
Learn more at:
http://edgeimpulse.com



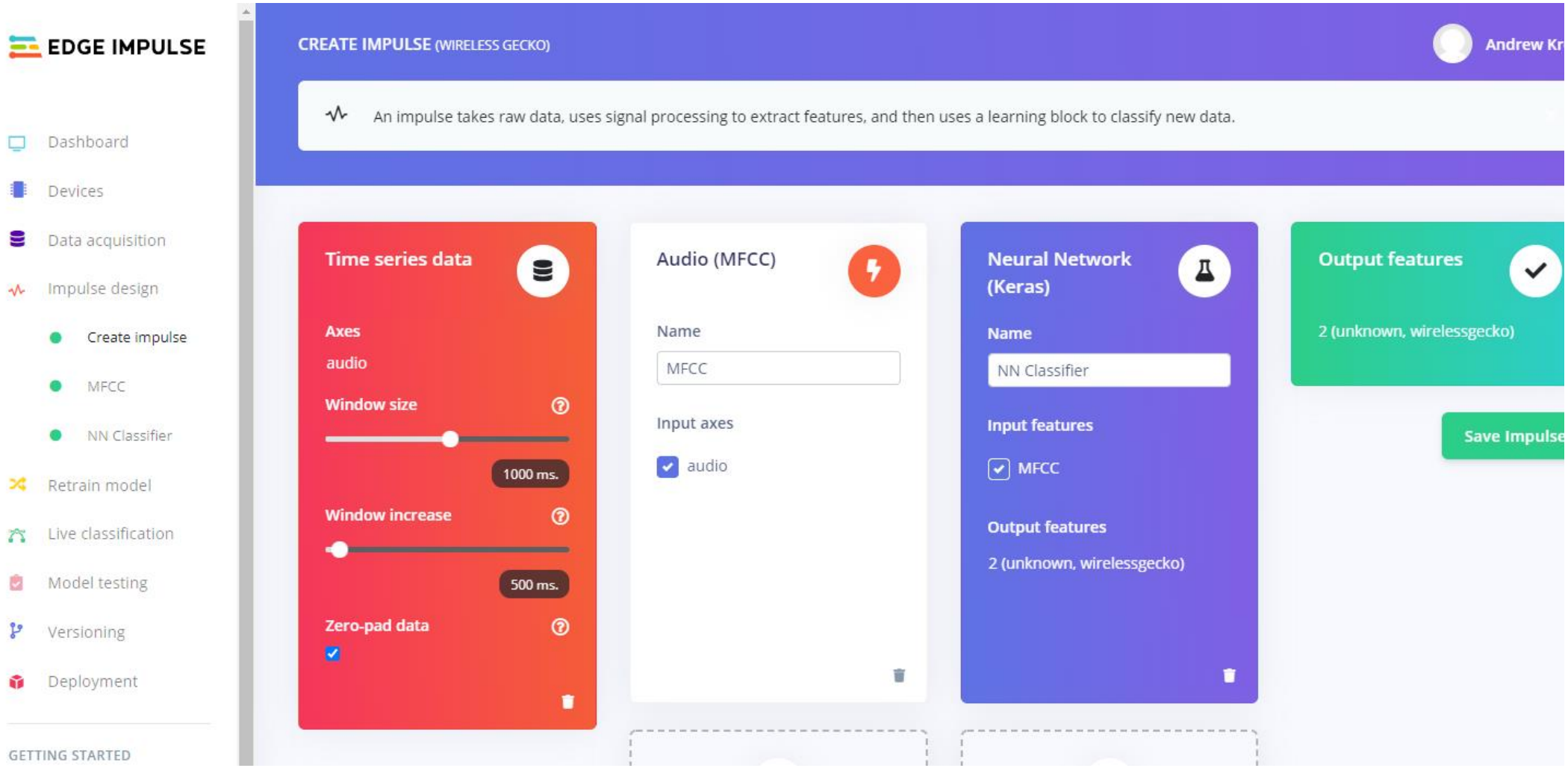



DATA COLLECTION
Edge Device
Data Upload
Hybrid Cloud Data
Dataset
Acquire valuable data securely
Impulse
Enrich data and train ML algorithms
Test
Validate ML with real-time data
Deploy
Build optimized embedded inference
EON™ Tuner
Advanced ML automation from DSP to NNs
EON™ Compiler
25-55% smaller models compared to TensorFlow Lite Micro
MLOPS FOUNDATION
Security
Versioning
Lifecycle
APIs
Data Science Tools
CICD & Test Automation
ML/Target Optimization
LEADING INTEGRATIONS
FLEXIBLE DEPLOYMENT
Embedded C/C++
Embedded Linux
Mobile and Web
(Web Assembly)

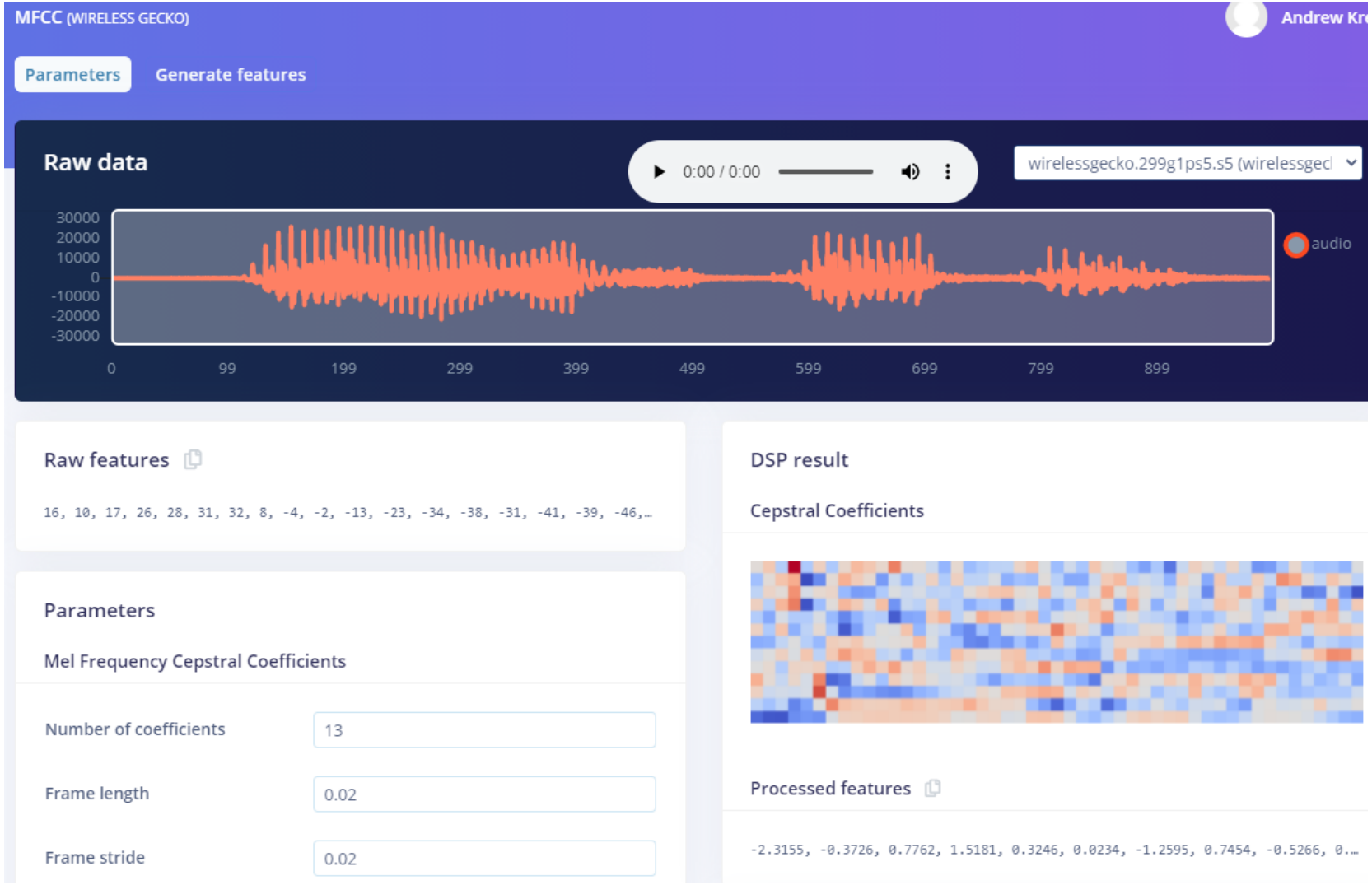
Closeup: Data Acquisition and Audio Sampling



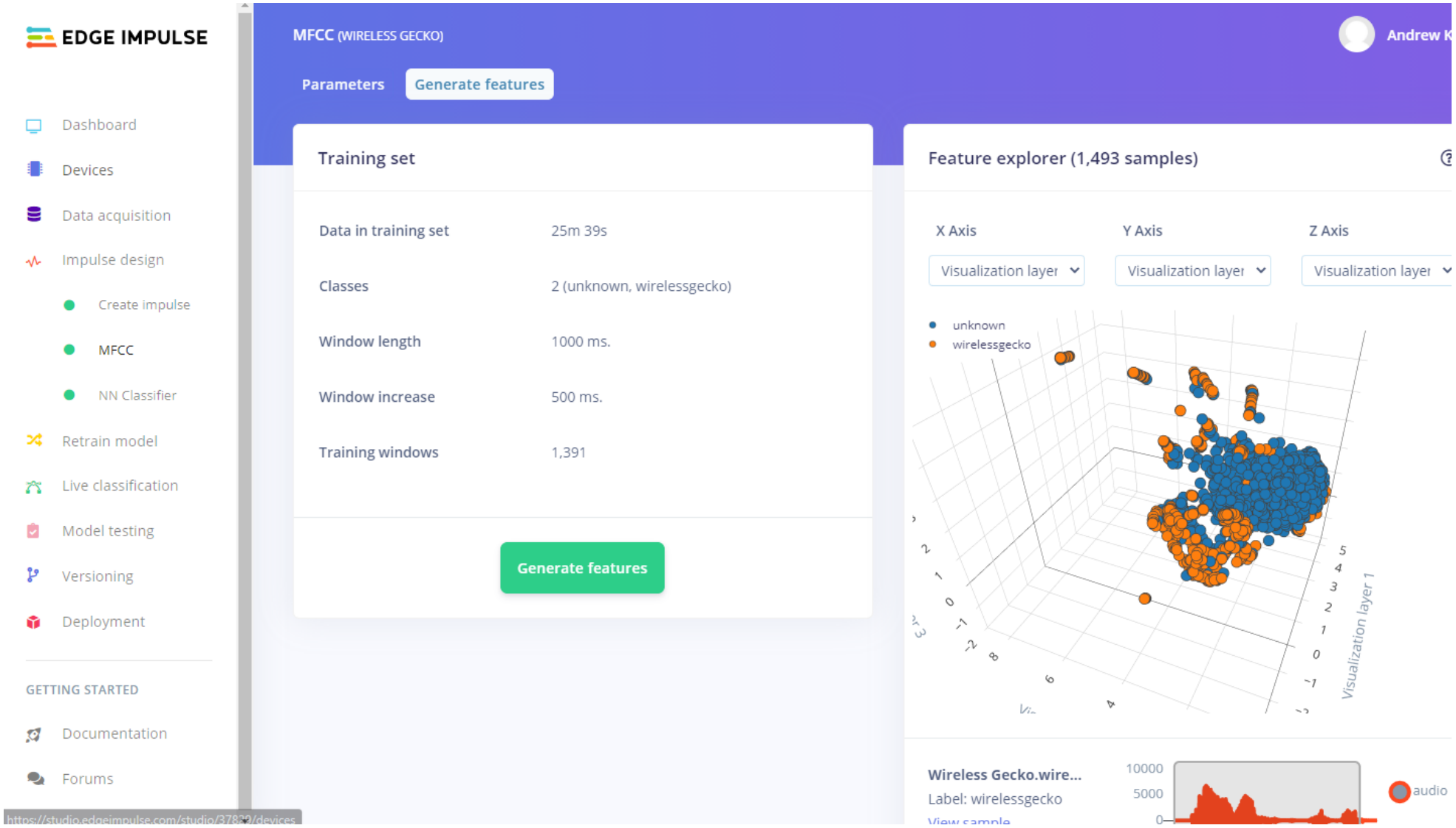
Closeup: Design the Impulse



Closeup: MFCC and Feature Extraction



Closeup: MFCC and Feature Extraction



Closeup: Training the NN

EDGE IMPULSE

- Dashboard
- Devices
- Data acquisition
- Impulse design
  - Create impulse
  - MFCC
  - NN Classifier
- Retrain model
- Live classification
- Model testing
- Versioning
- Deployment

GETTING STARTED

- Documentation
- Forums

Neural Network settings

Training settings

Number of training cycles: 100

Learning rate: 0.005

Minimum confidence rating: 0.60

Audio training options

Data augmentation: ☑

Add noise: None | Low | High

Mask time bands: None | Low | High

Mask frequency bands: None | Low | High

Warp time axis: ☐

Neural network architecture

Architecture presets: 1D Convolutional (Default) | 2D Convolutional

Input layer (650 features)

Reshape layer (13 columns)

1D conv / pool layer (8 neurons, 3 kernel size, 1 layer)

Dropout (rate 0.25)

Training output

Model

Model version: Quantized (int8)

Last training performance (validation set)

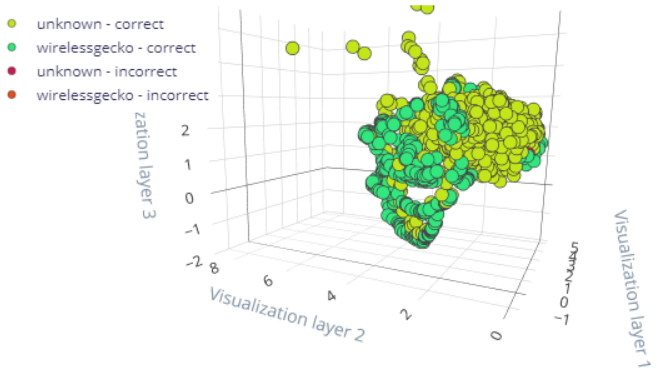
ACCURACY 99.0%

LOSS 0.03

Confusion matrix (validation set)

	UNKNOWN	WIRELESSGECKO
UNKNOWN	98.8%	1.2%
WIRELESSGECKO	0.8%	99.2%
F1 SCORE	0.99	0.99

Feature explorer (full training set)



On-device performance

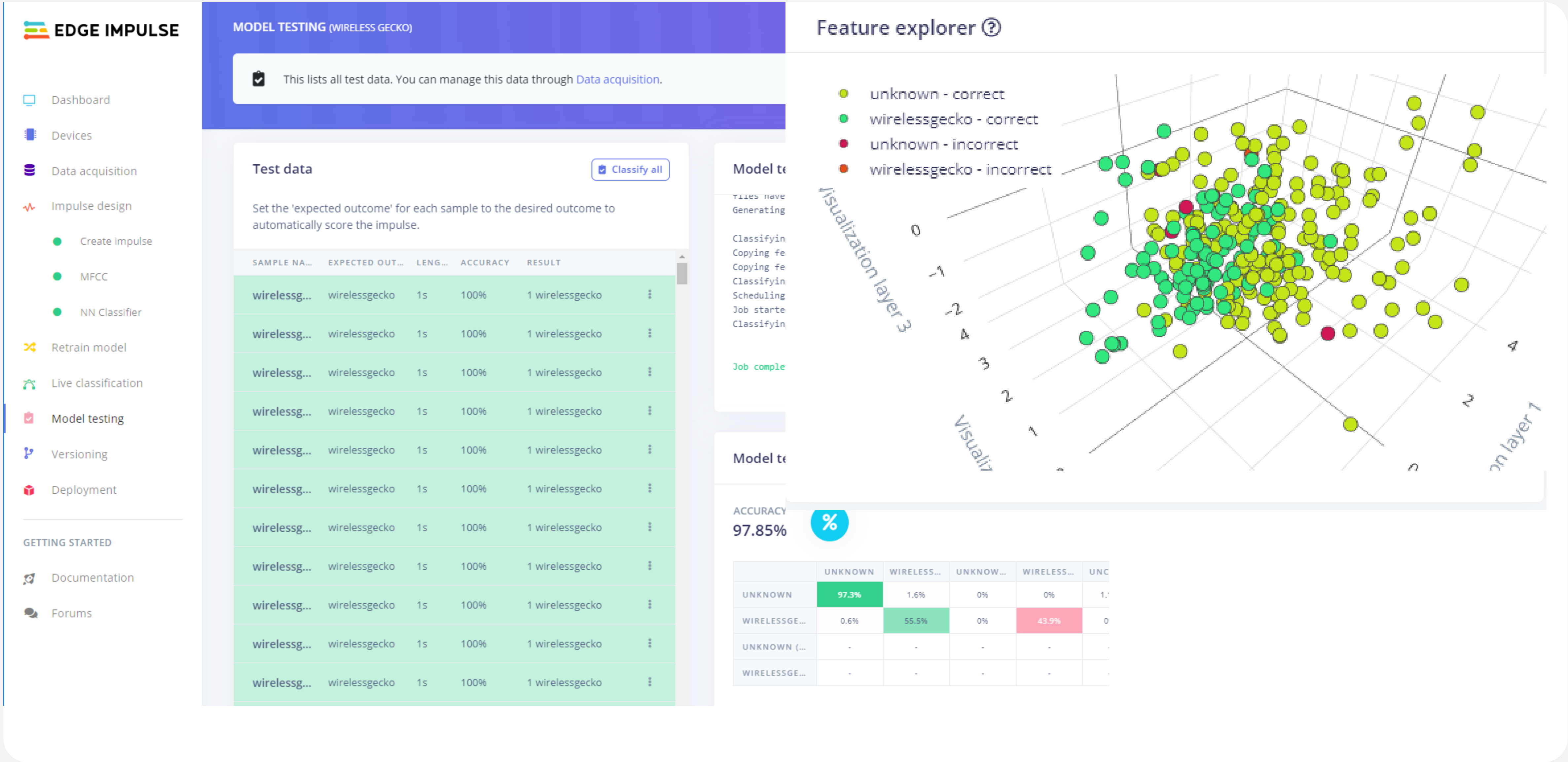
INFERENCING ... 8 ms.

PEAK RAM US... 4.4K

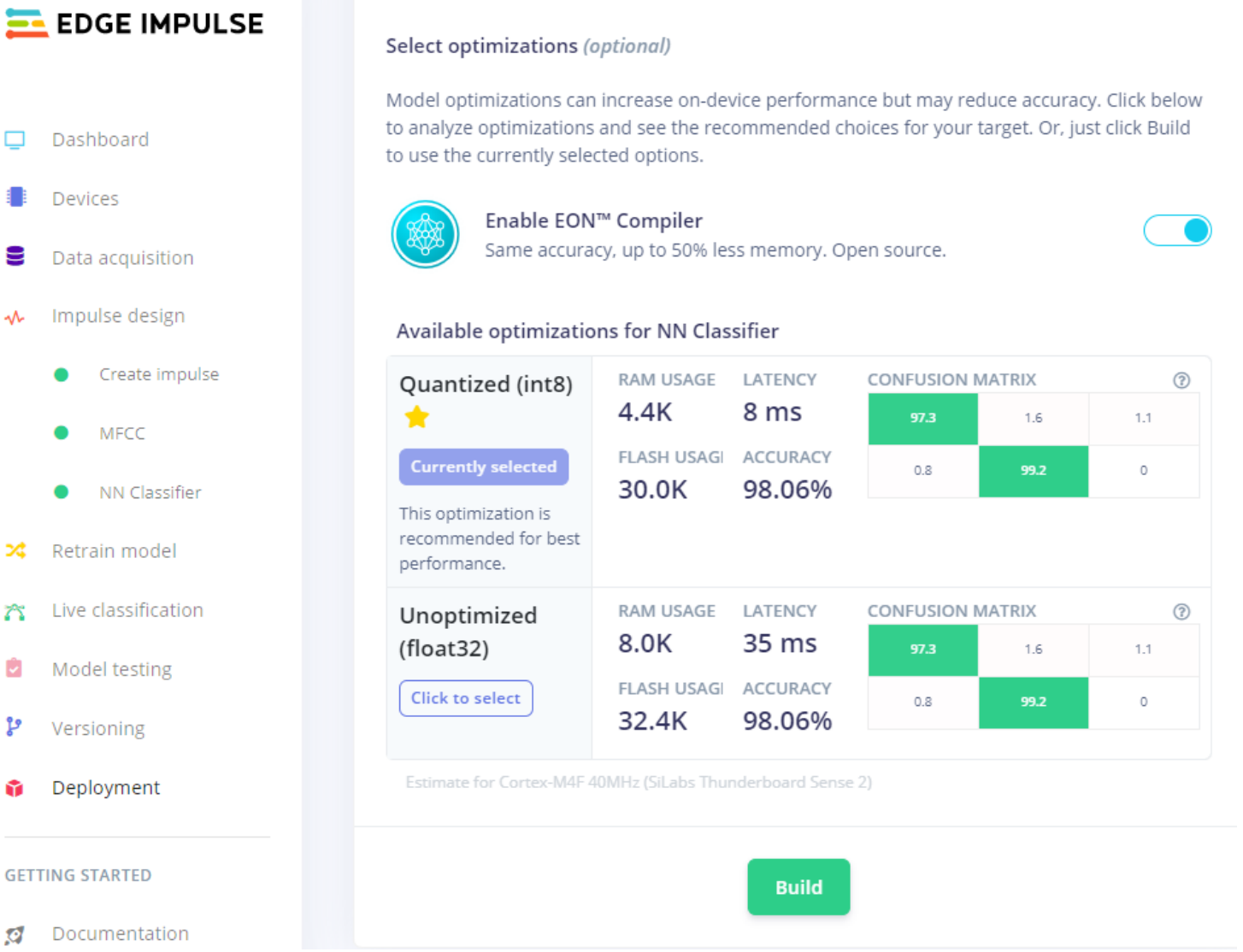
FLASH USAGE 30.0K

SILICON LABS

Initial Test Results



Deployment



DEMO

Dashboard - TechTalk_Demo1 - E

studio.edgeimpulse.com/studio/49459

EDGE IMPULSE

Project info | Keys | Export

Aki

- Dashboard
- Devices
- Data acquisition
- Impulse design
  - Create impulse
- EON Tuner
- Retrain model
- Live classification
- Model testing
- Versioning
- Deployment

GETTING STARTED

- Documentation
- Forums

Aki / TechTalk_Demo1

This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

Creating your first impulse (34% complete)

Acquire data

Every Machine Learning project starts with data. You can capture data from a development board or your phone, or import data you already collected.

LET'S COLLECT SOME DATA

Design an impulse

Teach the model to interpret previously unseen data, based on historical data. Use this to categorize new data, or to find anomalies in sensor readings.

GETTING STARTED: CONTINUOUS MOTION RECOGNITION

GETTING STARTED: RESPONDING TO YOUR VOICE

GETTING STARTED: ADDING SIGHT TO YOUR SENSORS

Deploy

Package the complete impulse up, from signal processing code to trained model, and deploy it on your device. This ensures that the impulse runs with

Sharing

Your project is private.

Make this project public

Summary

DEVICES CONNECTED
11

DATA COLLECTED
27m 35s

Collaborators

nd Prompt

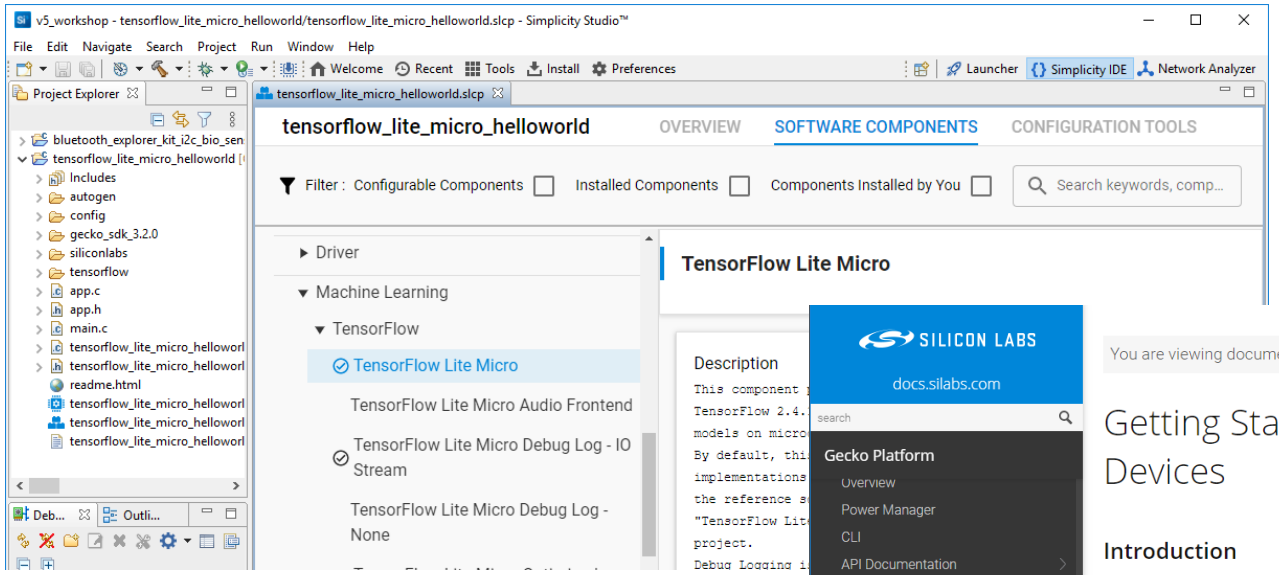
Type here to search

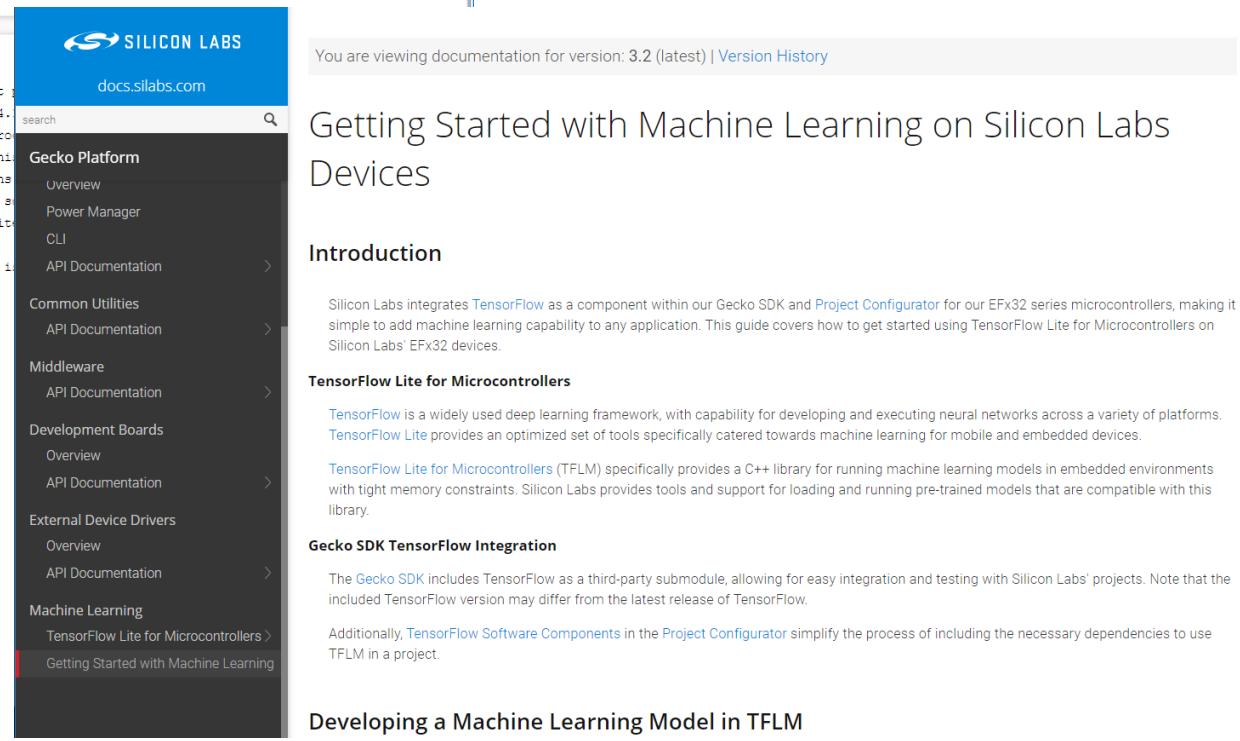
Testing on Hardware: Thunderboard Sense 2 – SLTB004A

Thunderboard Sense 2

- **Wireless SoC with multi-protocol radio**
- **Direct support with Edge Impulse**
- **ARM® Cortex® M4 core with 256 kB RAM and 1024 kB Flash**
- **Broad Range of Sensors**
  - 6-axis Inertial Sensor ICM-20648
  - Digital Microphone ICS-43434
  - Pressure Sensor BMP280
  - Indoor Air Quality and Gas Sensor CCS811
  - Relative Humidity and Temperature Sensor Si7021
  - UV and Ambient Light Sensor Si1133
  - Hall-effect Sensor Si7210

TensorFlow Lite for Microcontrollers





You are viewing documentation for version: 3.2 (latest) | Version History

Getting Started with Machine Learning on Silicon Labs Devices

Introduction

Silicon Labs integrates TensorFlow as a component within our Gecko SDK and Project Configurator for our EFx32 series microcontrollers, making it simple to add machine learning capability to any application. This guide covers how to get started using TensorFlow Lite for Microcontrollers on Silicon Labs' EFx32 devices.

TensorFlow Lite for Microcontrollers

TensorFlow is a widely used deep learning framework, with capability for developing and executing neural networks across a variety of platforms. TensorFlow Lite provides an optimized set of tools specifically catered towards machine learning for mobile and embedded devices.

TensorFlow Lite for Microcontrollers (TFLM) specifically provides a C++ library for running machine learning models in embedded environments with tight memory constraints. Silicon Labs provides tools and support for loading and running pre-trained models that are compatible with this library.

Gecko SDK TensorFlow Integration

The Gecko SDK includes TensorFlow as a third-party submodule, allowing for easy integration and testing with Silicon Labs' projects. Note that the included TensorFlow version may differ from the latest release of TensorFlow.

Additionally, TensorFlow Software Components in the Project Configurator simplify the process of including the necessary dependencies to use TFLM in a project.

Developing a Machine Learning Model in TFLM

- **Watch On-demand**

https://workswith.silabs.com/agenda

- **Take Survey and Get a FREE Bluetooth Explorer Kit:**

SILICON LABS

Q&A

tech talks

Facebook	Twitter	Community

SILICON LABS

tech talks

THANK YOU

Recording and slides will be posted to:
www.silabs.com/training