# Intelliquench:

## Machine learning for real-time monitoring of superconducting magnets


*Duc Hoang (Rhodes College/Fermilab)*

*with Nhan Tran, Stoyan Stoynev, Steven Krave, Vittorio Marinozzi, Cristian Boffo.*

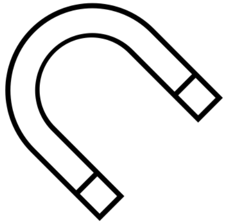


***Accelerator Physics and Technology Seminar — September 22, 2020***

# About me

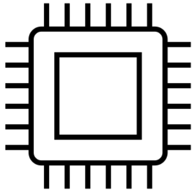
- Undergraduate student at Rhodes College.
- Project was done during the SIST internship program this summer at Fermilab.

# Outline



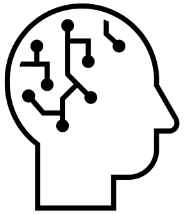
Superconducting magnets and quenches

Questions



Hardware setup and data acquisition

Questions



Anomaly detection using machine learning

Questions

## Superconducting magnets and quenches

*Questions*

## Hardware setup and data acquisition

*Questions*

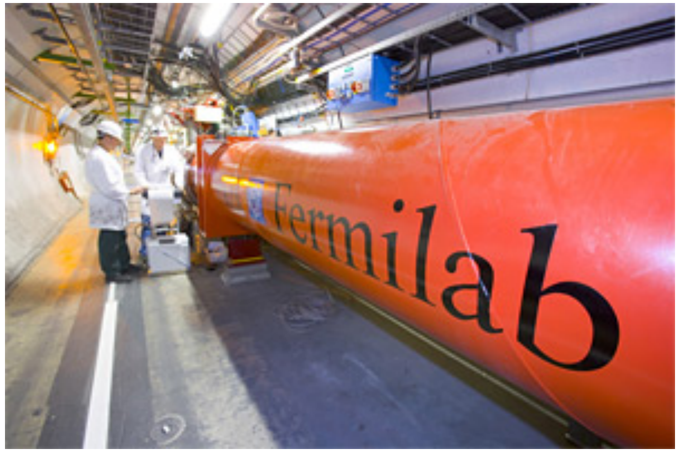
## Anomaly detection using machine learning

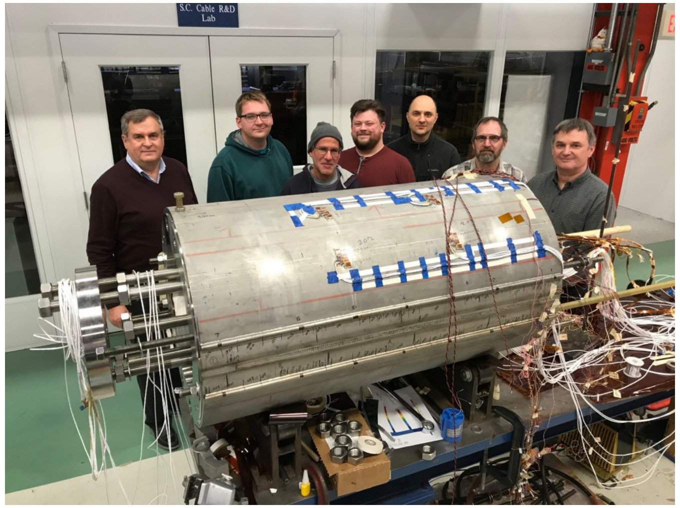
*Questions*

# Superconducting magnets

- Superconducting magnets are used widely in particle accelerators for **manipulation of particle beams**.

- Can carry large current with **no electrical resistance**. Can generate very **intense magnetic field.**

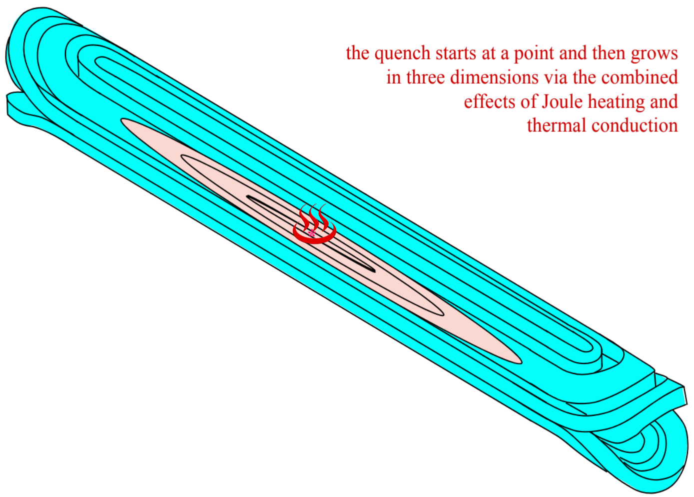


# Quenches in superconducting magnets

- In order to maintain superconductivity, superconducting magnets typically operate **at or below liquid helium temperature.**
- Due to several reasons (mechanical imperfections, conductor motions, …), **a specific spot** in the magnet may **heat up**.
- This can eventually cause the whole magnet to **become resistive.** And with huge amount of current pumping through, it can be **catastrophic**.

*Growth of the resistive zone*




Wilson et al. Superconducting magnets for accelerators.

# Quenches are dangerous

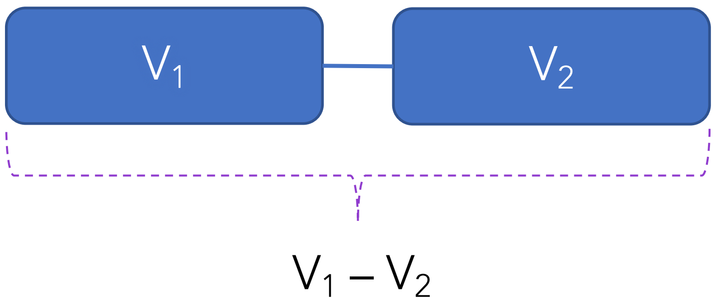
In 2008, magnet quench occurred in 100 magnets at the LHC at CERN, leading to a loss of approximately **six tons** of liquid helium.

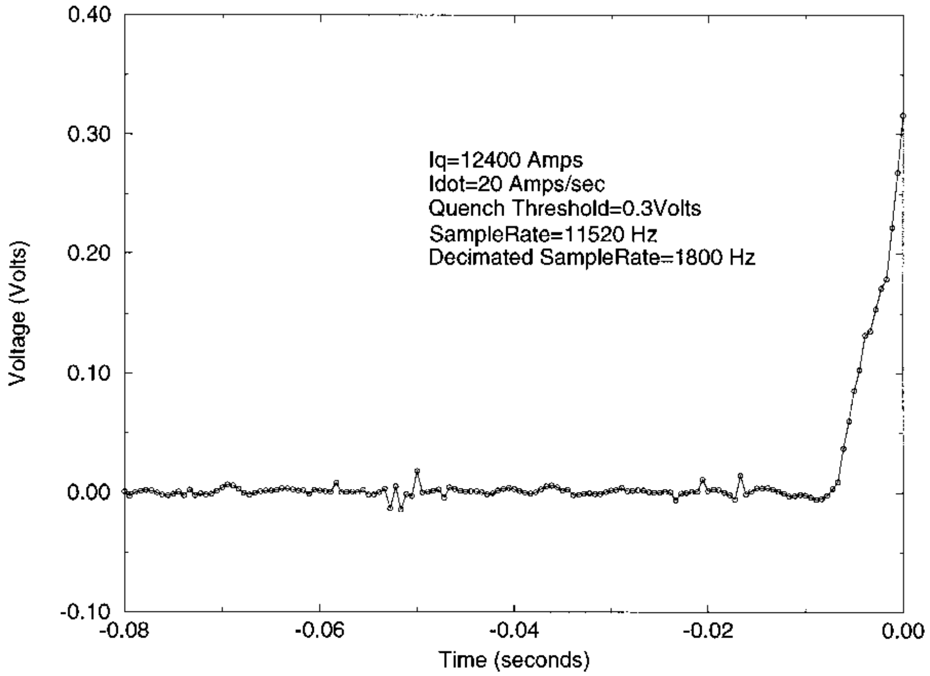
The escaping vapor expanded with explosive force, damaging a total of **53 superconducting magnets** (each costs several millions dollar.)

https://en.wikipedia.org/wiki/Large_Hadron_Collider#Quench_incident

# Current quench protection systems

- A typical system detects a **difference in voltage** between **two halves** of the magnet.
- Discharge the magnet by dissipating energy either on to the magnets' own **thermal mass** or externally through a **dump resistor**.
- System responses after quench happens in about ~**few ms to tens of seconds**.




D.F. Orris et al

See more: R. Denz , Steckert and A. Skoczen

# Challenges

- Quenches are not well-understood: we don't know whether "quench precursors" exist, and in what kind of data.
- Underlying physics is still not well-understood. We don't know the exact details of what's happening yet.
- Quench protection systems only detect the quench ***after*** it happens, thus requiring very short response time O(*ms*).

Superconducting magnets and quenches

Questions

Hardware setup and data acquisition

Questions

Anomaly detection using machine learning

Questions

Superconducting magnets and quenches

Questions

Hardware setup and data acquisition

Questions

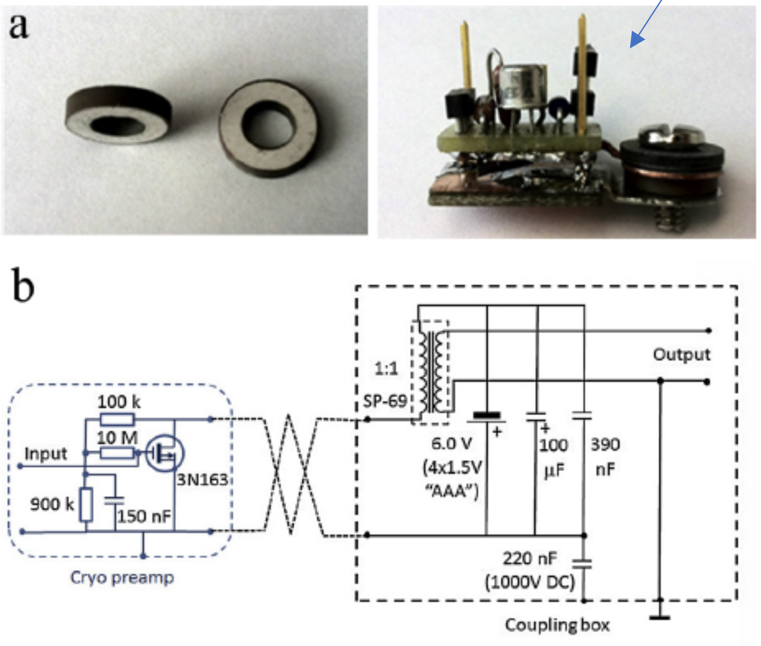
Anomaly detection using machine learning

Questions

# Acoustic sensors for magnet diagnostics

- Non-invasive technique which allows **better localization** of mechanical events.
- Highly advantageous for preventing **magnet integrity** from being jeopardized by voltage taps.
- Has been **widely adapted** for non-destructive evaluation of mechanical stability in various settings.
- Only **high-rate data** set available for us.

Sensor with integrated cryogenic preamplifier.



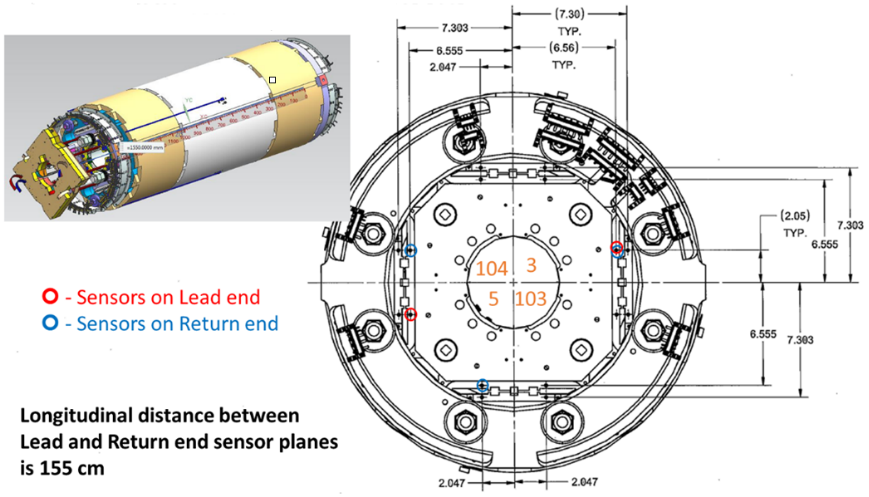

See [M. Marchevsky et al]()

*Data are courtesy of the LARP, AUP and HL-LHC teams and Fermilab US-MDP team.*

# Specific setups in two different magnets

## MQXFS1d (5 sensors)

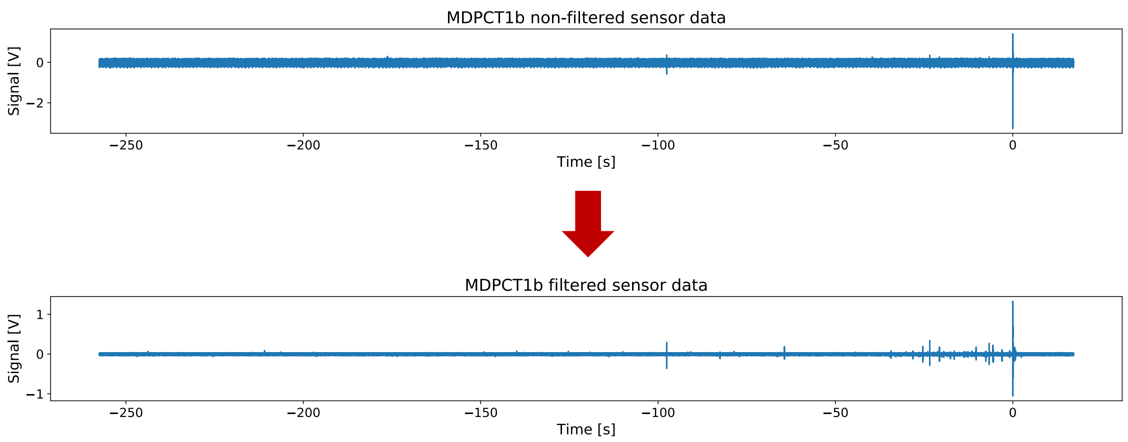
- **5 acoustic sensors** attached to two sides of magnet.
- Sampling rate of **100kHz**




**Longitudinal distance between Lead and Return end sensor planes is 155 cm**

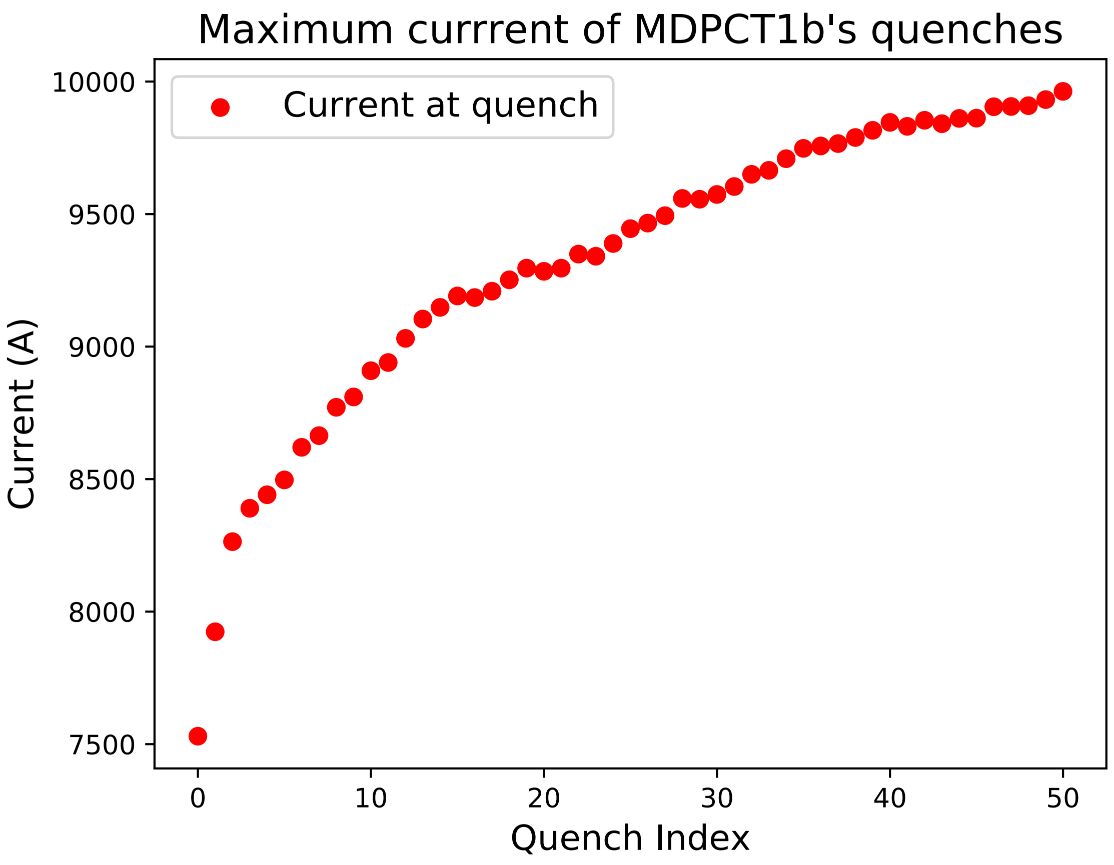
## MDPCT1b (2 sensors)

- **2 acoustic sensors** attached to two sides of magnet.
- Sampling rate of **1MHz** but was **filtered and down-sampled to 100kHz** due to noise.



# Obtain quench data from magnet training

- Superconducting magnets must be "**trained**" to operate at high current.
- During the training process, magnet is repeatedly ran at a low current, and then a slightly increased current until a **natural quench happens**.
- Typically, magnet is quenched until quench's current reaches **plateau**.




*Magnet condition will also change continuously*

Superconducting magnets and quenches

Questions

Hardware setup and data acquisition

Questions

Anomaly detection using machine learning

Questions

# *Pause*

Intelliquench:

Machine learning for real-time monitoring of superconducting magnets

# Pause

## *Why machine learning?*

- We **have yet to** exactly **pin down the characterizations** of "quench precursors." And they might have different features among each other.

  👉 Impossible to have a one-for-all standard method.

- Magnet training quenches **are certainly due** to some release of energy that induces the quench, and this released energy **can be detected in several ways**.

  👉Not only voltage, but also optical data, acoustic data, electromagnetic data, etc.

- Machine learning techniques are ideal **to analyze and correlate** this **large amount of different types of data** in order to find anomalies and **detect quench precursors**.

  👍If properly trained, ML algorithms can find anomalies invisible to human eyes, and can be faster compare to traditional methods.

# Pause

## *Why machine learning?*

- We **have yet to** exactly **pin down the characterizations** of "quench precursors." And they might have different features among each other.

  👉 Impossible to have a one-for-all standard method.

- Magnet training quenches **are certainly due** to some release of energy that induces the quench, and this released energy **can be detected in several ways**.

  👉Not only voltage, but also, acoustic data, electromagnetic data, optical data, etc.

- Machine learning techniques are ideal **to analyze and correlate** this **large amount of different types of data** in order to find anomalies and **detect quench precursors**.

  👍If properly trained, ML algorithms can find anomalies invisible to human eyes, and can be faster compare to traditional methods.

# Pause

## *Why machine learning?*

- We **have yet to** exactly **pin down the characterizations** of "quench precursors." And they might have different features among each other.

  👉 Impossible to have a one-for-all standard method.

- Magnet training quenches **are certainly due** to some release of energy that induces the quench, and this released energy **can be detected in several ways**.

  👉Not only voltage, but also acoustic data, electromagnetic data, optical data, etc.

- Machine learning techniques are ideal **to analyze and correlate** this **large amount of different types of data** in order to find anomalies and **detect quench precursors**.

  👍If properly trained, ML algorithms can find anomalies invisible to human eyes, and can be faster compare to traditional methods.

# *Pause*

*Why real-time?*

Data pouring in at **high rate** (100kHz and higher in the future), since many events are much clearer or at all visible then.

Also, the earlier we know the quench, the better.

Superconducting magnets and quenches

*Questions*

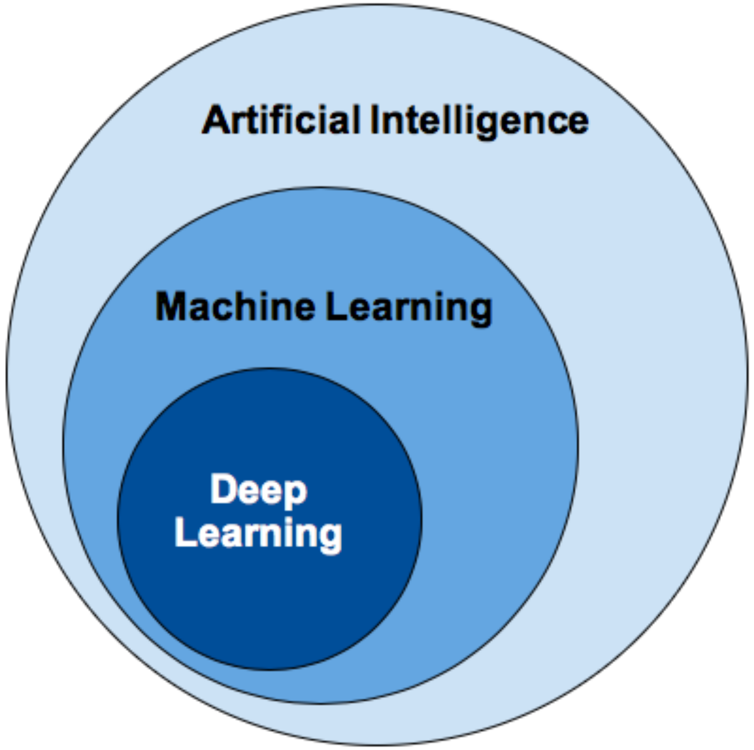
Hardware setup and data acquisition

*Questions*

Anomaly detection using machine learning

*Questions*

# Machine Learning Overview

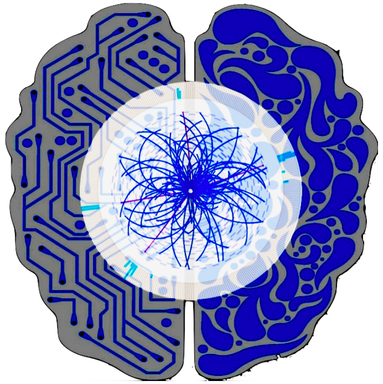


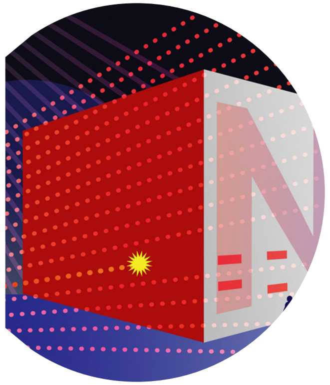

*"Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so."*

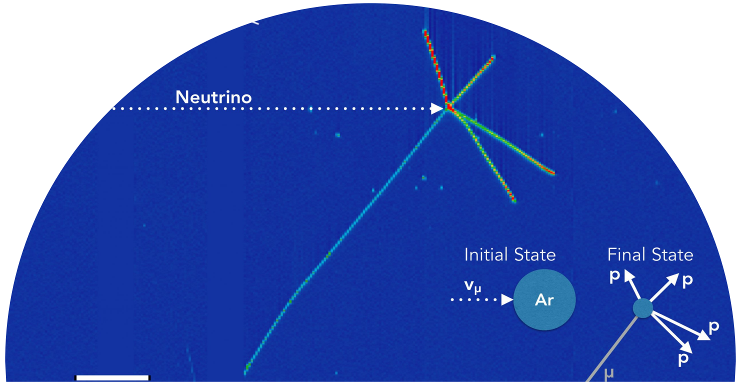
https://en.wikipedia.org/wiki/Machine_learning

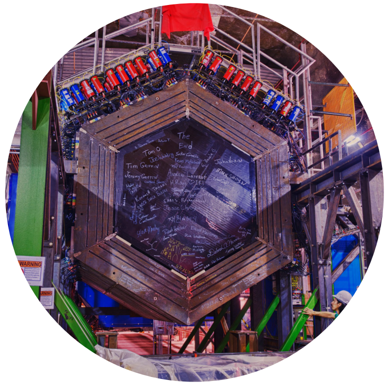
*Generic non-linear function approximation tools!*

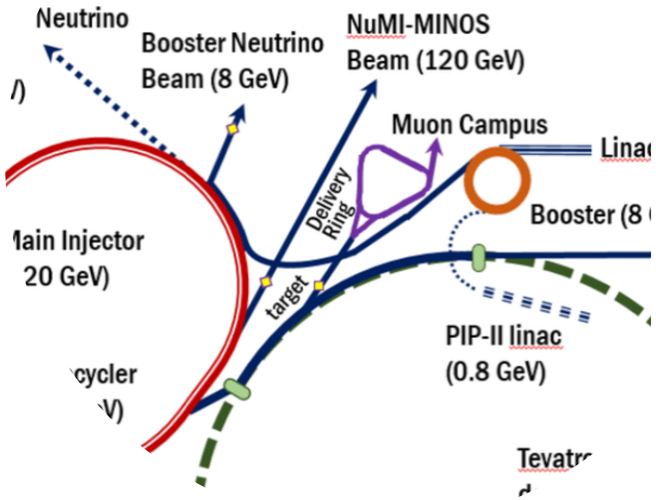
# Machine Learning @Fermilab










# How are we approaching this problem?

The problem: we don't know what we are looking for.  So need to rely on unsupervised learning.

## Unsupervised learning ⭐

- Data set with **no pre-existing labels**.
- Example: How many classes of images are in my data?
- Anomaly detection: Identify commonalities in the data and detect anomalous data points.

## Supervised learning

- Data set with **pre-existing labels**
- Example: Cats & Dogs Classifier

# What are needed?

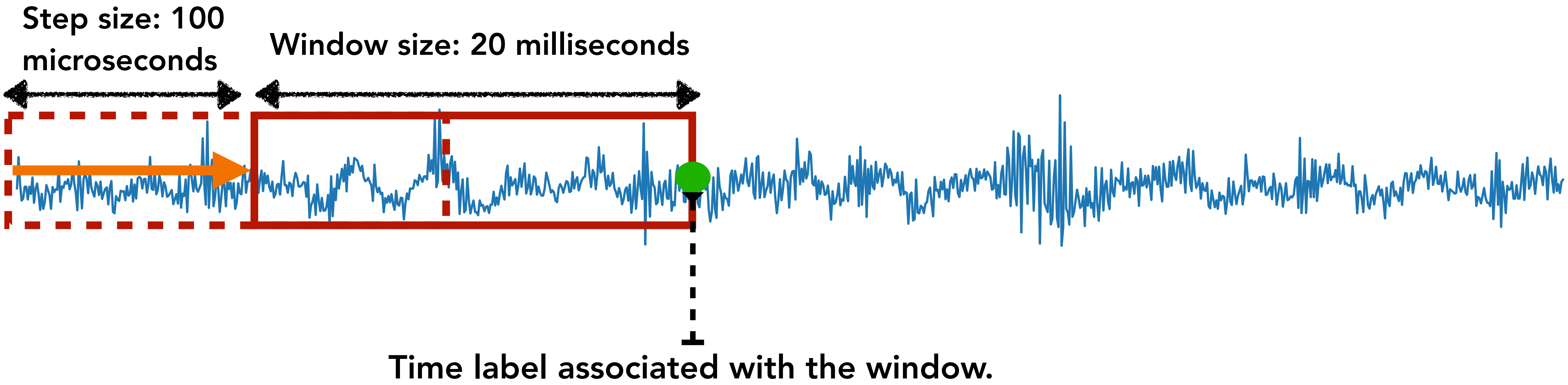
1. Input features (Ingredients).
2. Type of algorithm to process the inputs (Recipe).
3. How to deploy the algorithm in real-time (Serve the food).

# What are needed?

1. Input features (Ingredients).
    1. Statistical features.
    2. Raw signal data.
2. Type of algorithm to process the inputs (Recipe).
3. How to deploy the algorithm in real-time (Serve the food).

# Statistical features calculation (ingredients)

*Inspired by: LANL Earthquake prediction challenge.*




*In each window, calculate 2 features: standard deviation, and mean of the amplitude.*
*Also features are multiplied across sensors to look for "coherent" spikes.*

# Multiply inputs across sensors

*SD = Standard Deviation*

Sensor 1 – SD

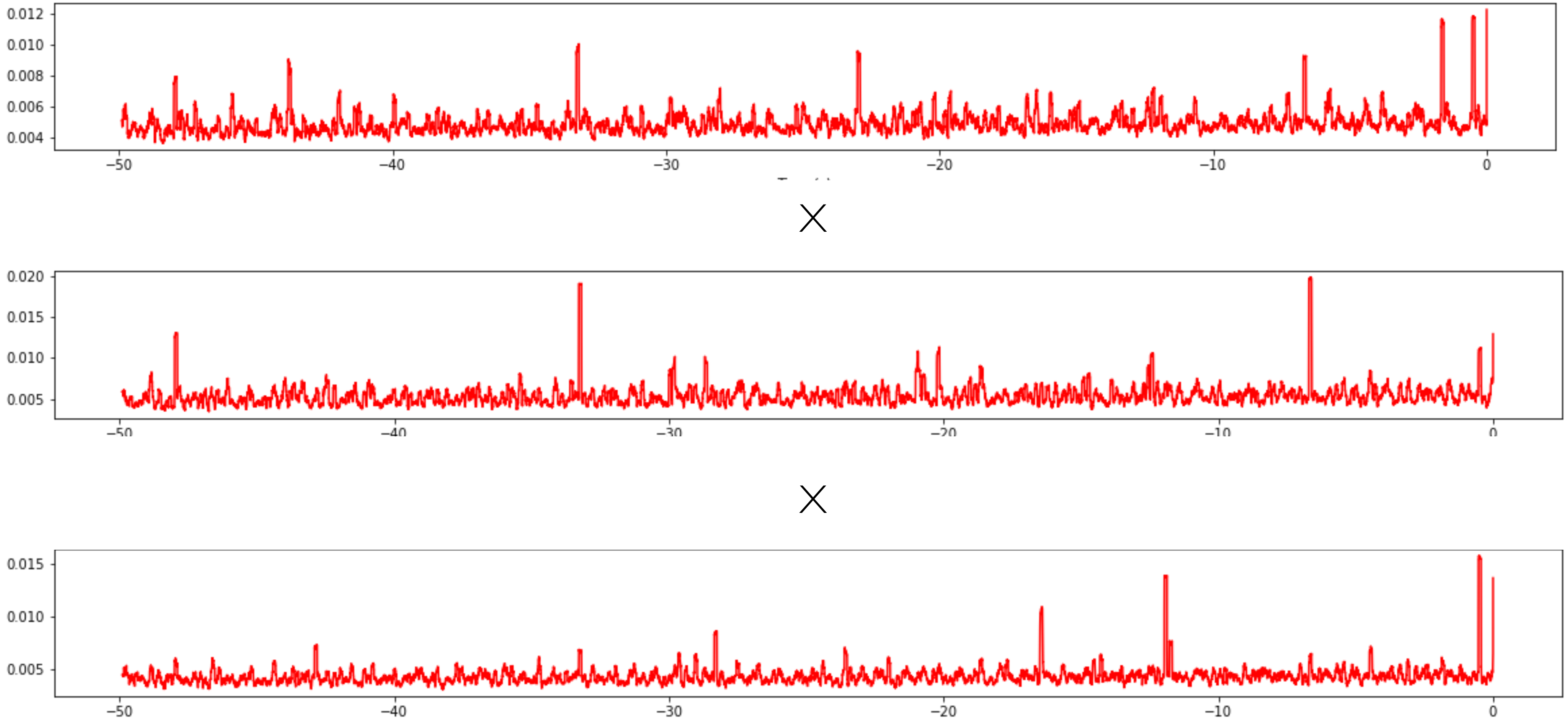
×

Sensor 2 – SD

×

Sensor 3 – SD

Etc.




Time [s]

- *12 inputs in total if 5 sensors.*
- *6 inputs in total if 2 sensors.*

# Scaling inputs

After calculations, every inputs are scaled to between (0,1) so that they can be treated equally in the algorithm when training the model.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

*Use the same scaling factors when making predictions.*

# What are needed?

1. Input features (Ingredients).
    1. Statistical features.
    2. Raw signal data.
2. Type of algorithm to process the inputs (Recipe).
    1. Fully-Connected Deep Neural Network (Statistical features)
    2. Convolutional Neural Network (Raw signal data)
3. How to deploy the algorithm in real-time (Serve the food).

# Exploring two classes of deep learning algorithms.

Exploratory

## Fully-Connected Deep Neural Network ⭐

- Input: **statistical features** calculated using "rolling window."
- Less computationally intensive
- Easier to design

## Convolutional Neural Network

- Input: **raw signals.** 👉 Don't need to worry about expert features.
- More computationally intensive.
- Harder to design

# Exploring two classes of deep learning algorithms.

Exploratory

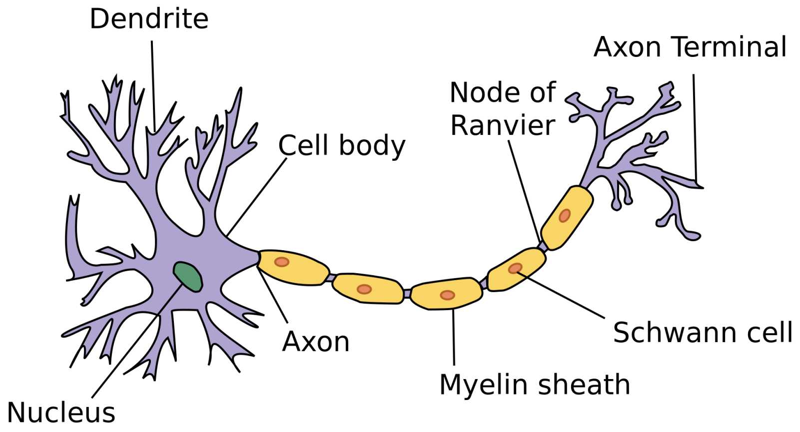
## Fully-Connected Deep Neural Network ⭐

- Input: **statistical features** calculated using "rolling window."
- Less computationally intensive
- Easier to design

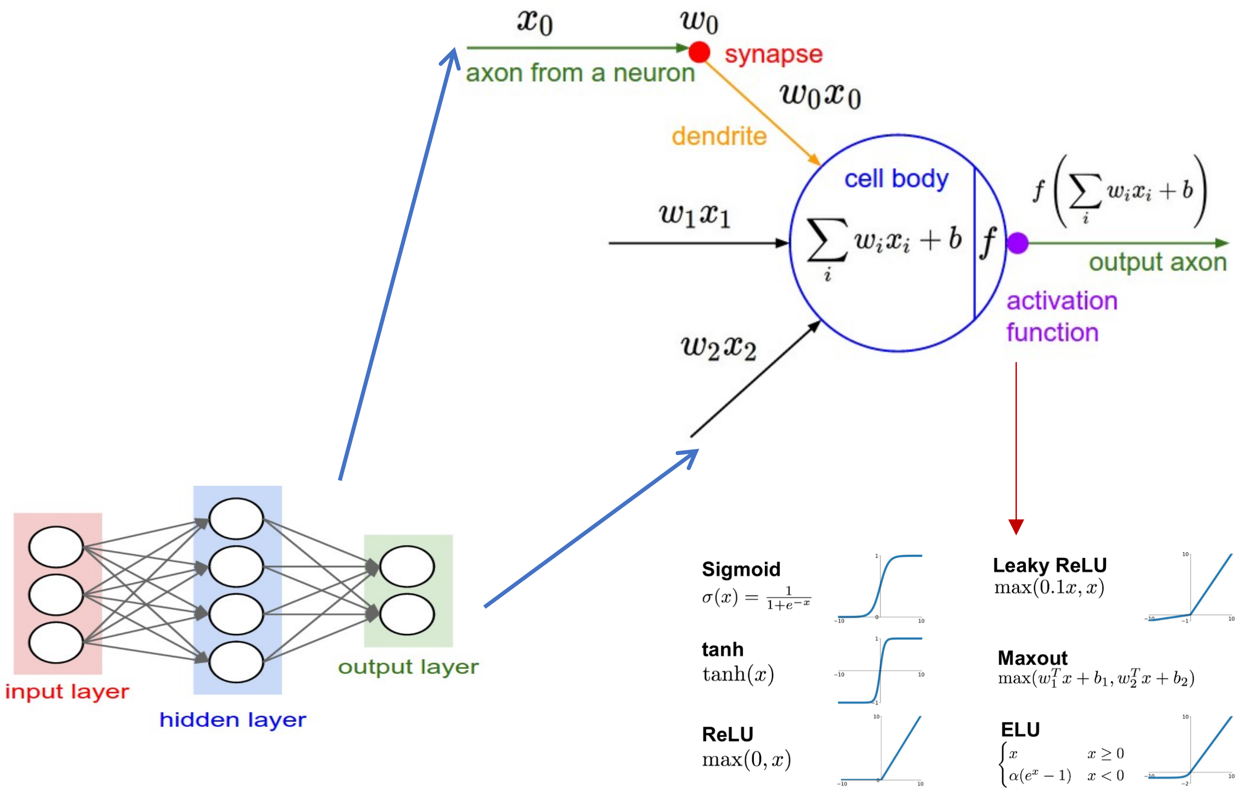
## Convolutional Neural Network

- Input: raw signals. 👉 Don't need to worry about expert features.
- More computationally intensive.
- Harder to design

# Deep Neural Network (DNN)



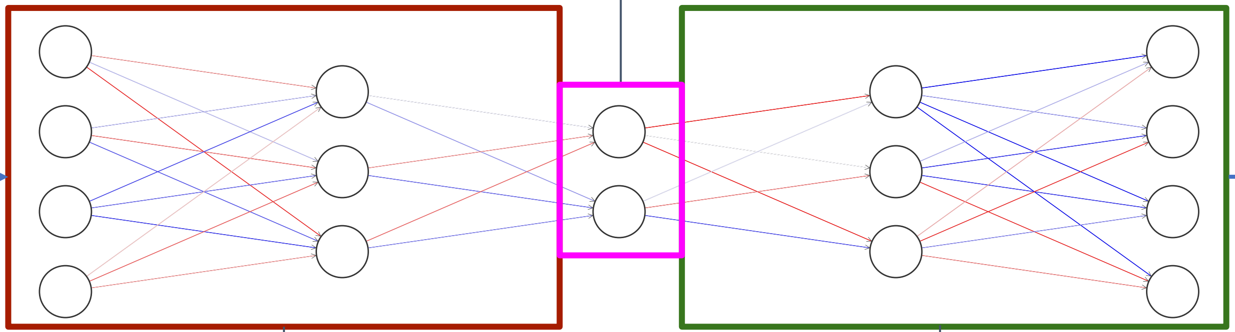

- Each **input** is multiplied by a weight.
- Weighted values are summed, Bias is added.
- Non-linear activation function is applied
- Trained by varying the parameters to minimize a loss function (quantifies how many mistakes the network makes)



# Auto-encoder DNN (Recipe)

Compressed info
(latent space)

Statistical features



Input Layer $\in \mathbb{R}^4$ | Hidden Layer $\in \mathbb{R}^3$ | Hidden Layer $\in \mathbb{R}^2$ | Hidden Layer $\in \mathbb{R}^3$ | Output Layer $\in \mathbb{R}^4$

Encoder

Decoder

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \hat{y}^{(i)})^2}$$

Root mean square error (RMS)

Large error? Anomaly

# Exploring two classes of deep learning algorithms.

Exploratory

| Fully-Connected Deep Neural Network ⭐ | Convolutional Neural Network |
|---|---|
| • Input: statistical features calculated using "rolling window." | • Input: raw signals. 👉 Don't need to worry about expert features. |
| • Less computationally intensive | • More computationally intensive. |
| • Easier to design | • Harder to design |

# Convolutional Neural Network (CNN)

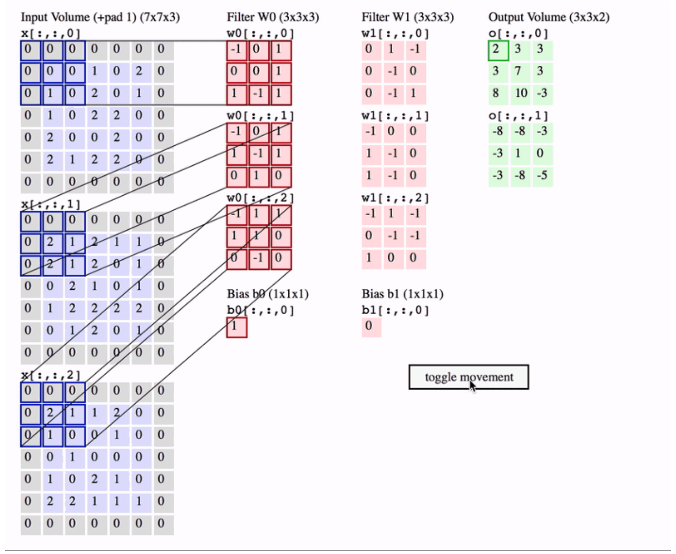
Exploratory

## Convolution Operation (Conv)

Red

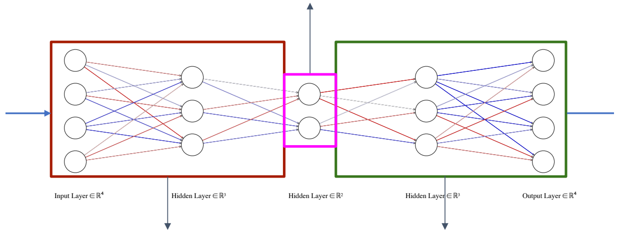
Blue

Green





*Similar idea to DNN, but using raw signals, instead of calculated features*

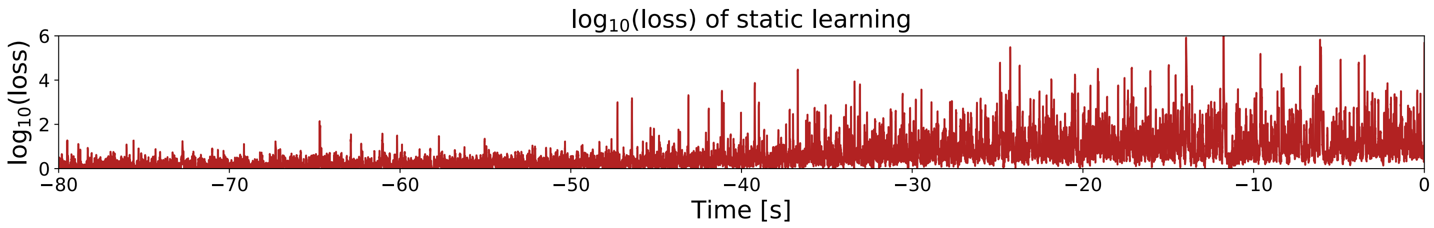


# What are needed?

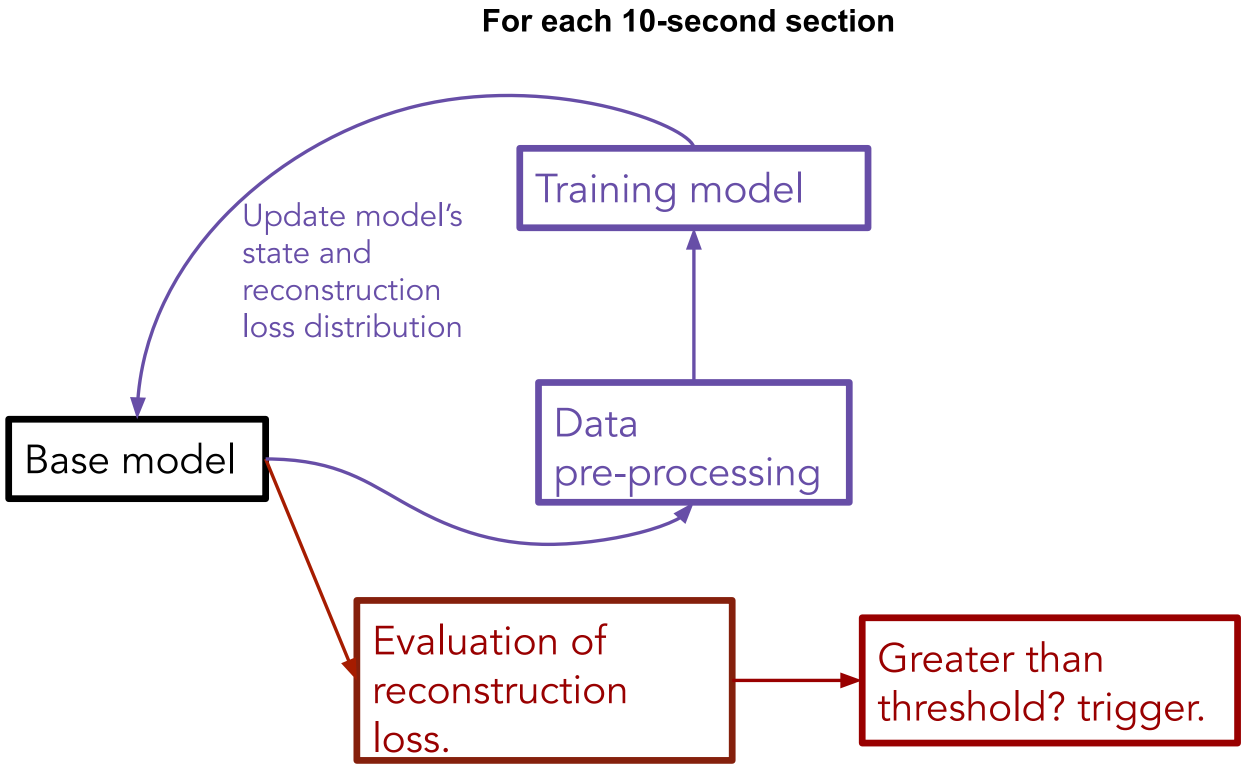
1. Input features (Ingredients).
   1. Statistical features.
   2. Raw signal data.
2. Type of algorithm to process the inputs (Recipe).
   1. Fully-Connected Deep Neural Network
   2. Convolutional Neural Network
3. How to deploy the algorithm in real-time (Serve the food).
   1. Static Learning ❌
   2. Dynamic Learning ✅

# “Static” learning

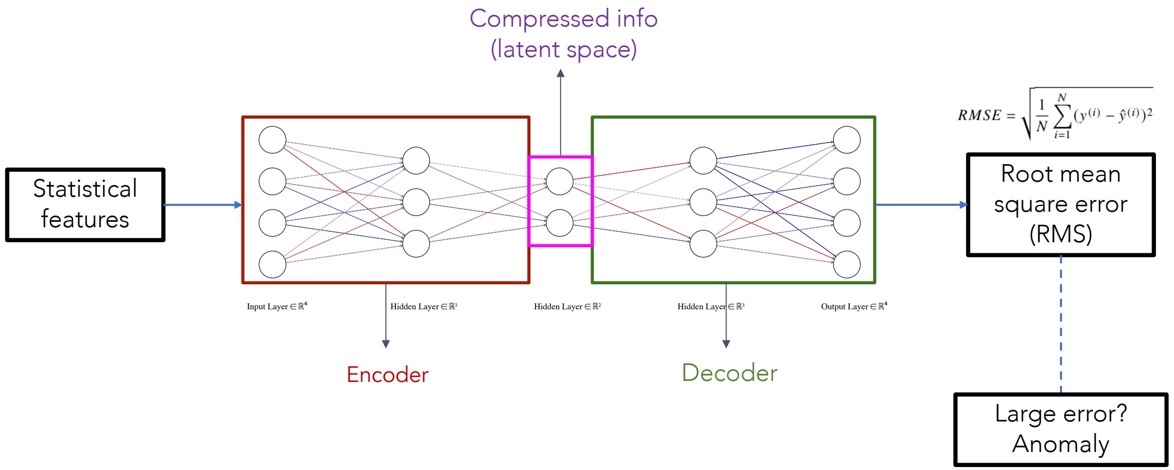
- Typically, a model is trained on a specific section of normal data, and then used for prediction.
- However, this is not applicable for this scenario, since there is an increasing noise level as we go to higher current.



# Dynamic learning

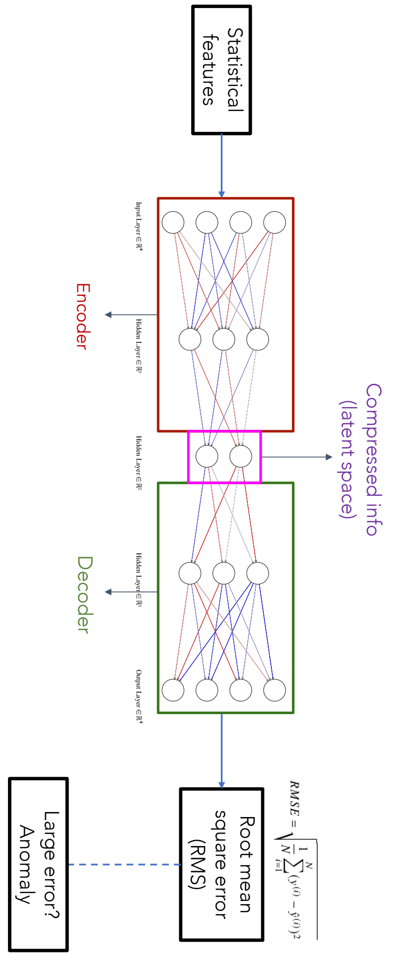


# Experiment with auto-encoder DNN



# Network architecture

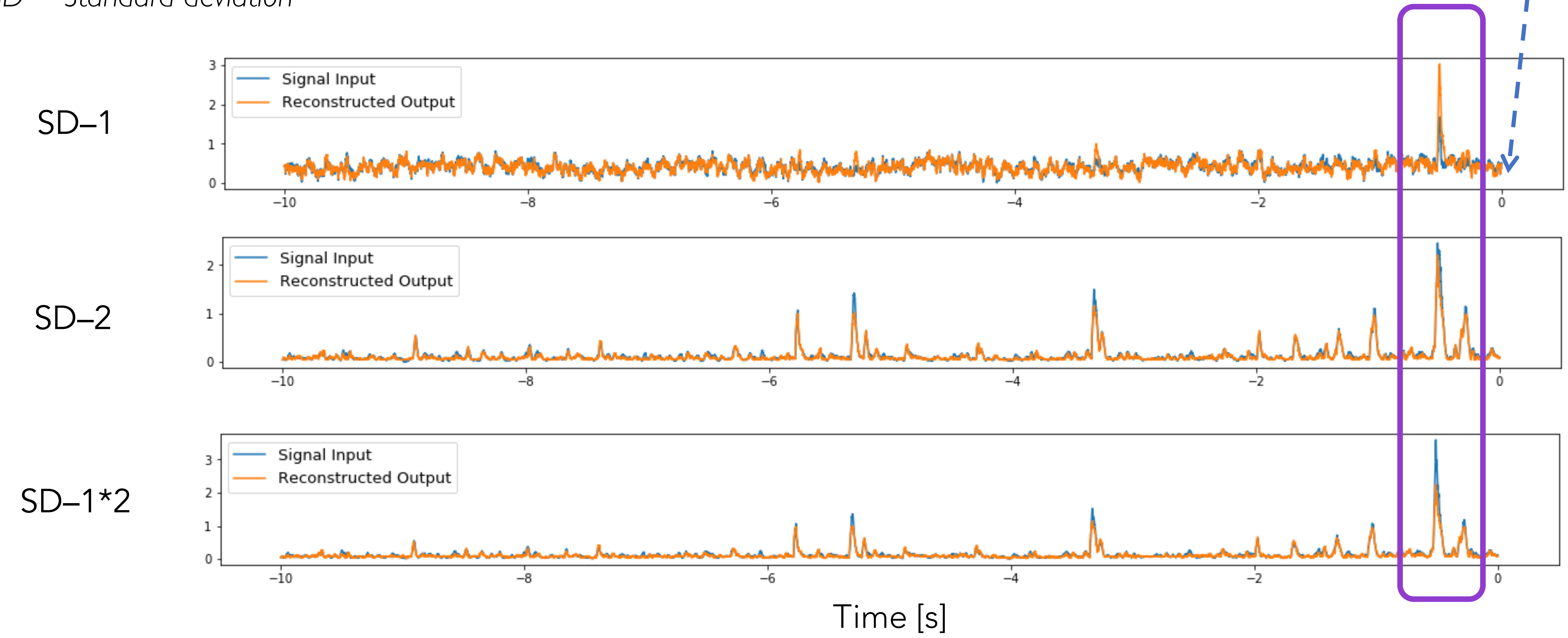
| | MQXFS1d (5 sensors) | MDPCT1b (2 sensors) |
|---|---|---|
| | 12 inputs | 6 inputs |
| Encoder | 6 nodes<br>activation: ELU | 4 nodes<br>activation: ELU |
| Compressed representation<br>(Latent space) | 3 nodes<br>activation: ELU | 2 nodes<br>activation: ELU |
| Decoder | 6 nodes<br>activation: ELU | 4 nodes<br>activation: ELU |
| Reconstructed Output | 12 outputs | 6 outputs |



# Anomalous event visualization

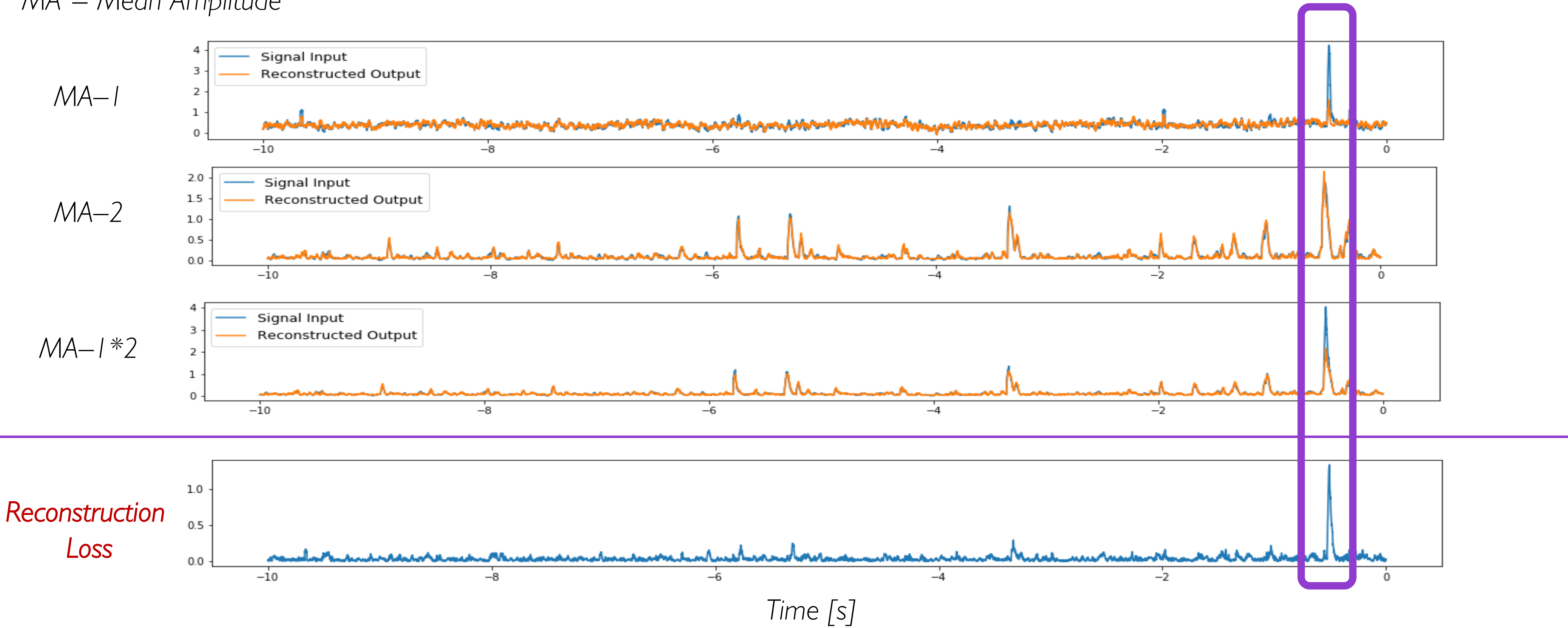
*SD = Standard deviation*

0 is quench detection time

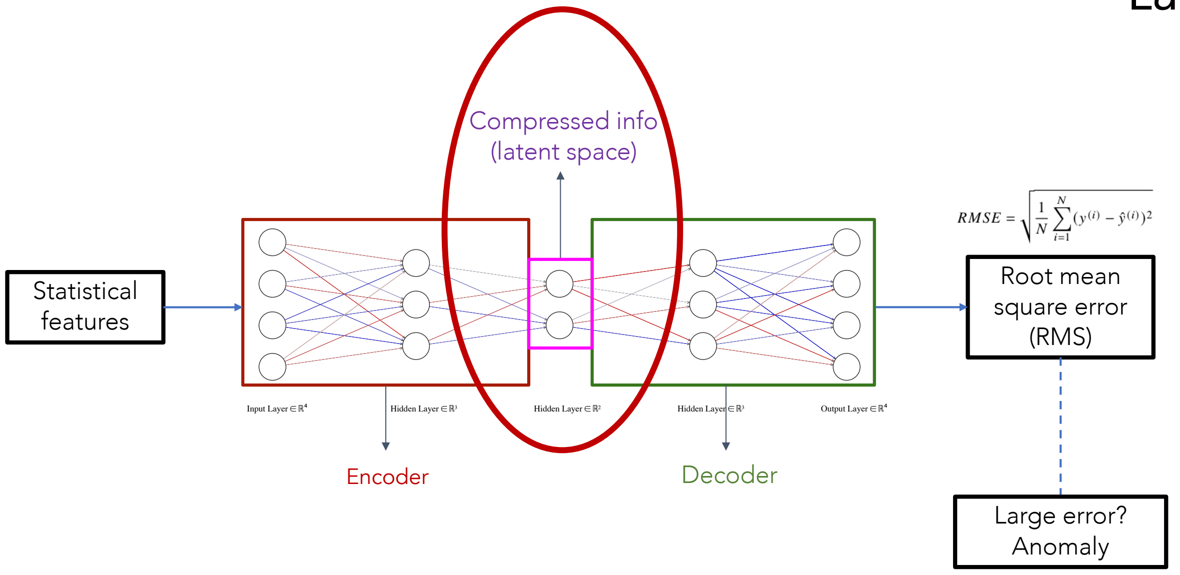


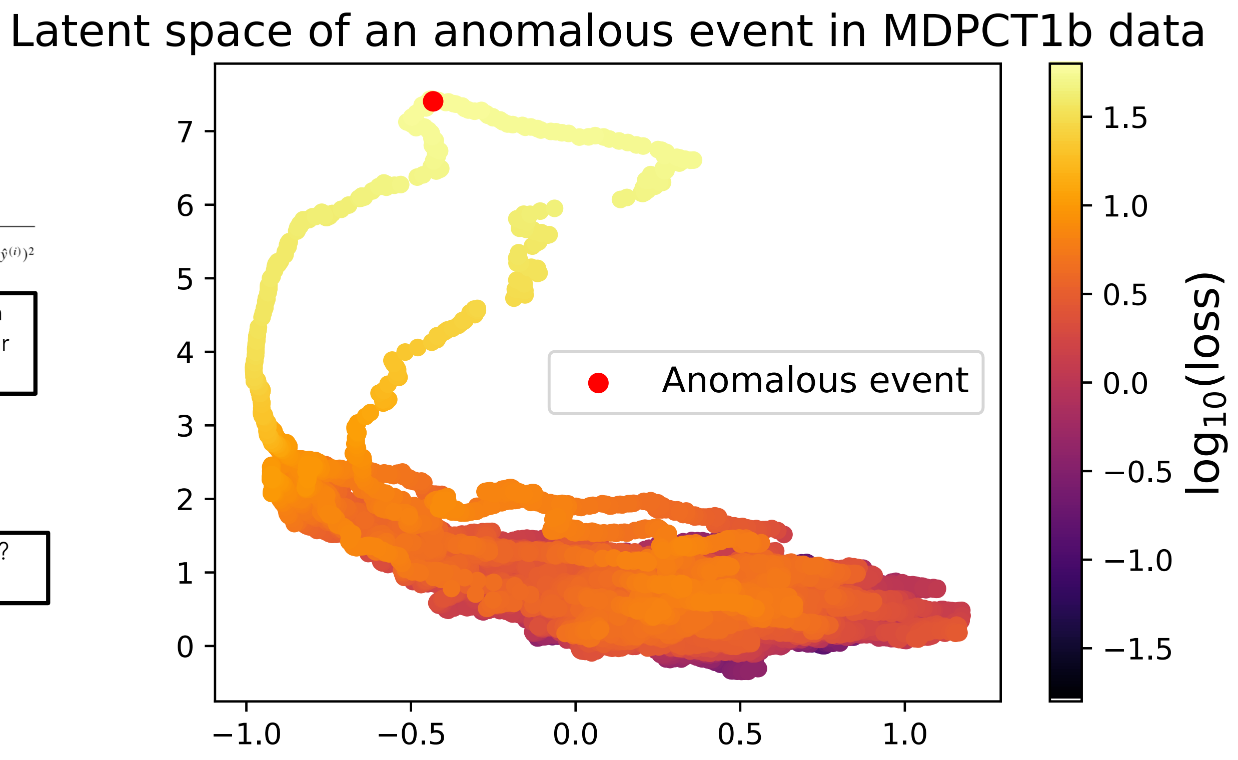
# Anomalous event visualization – 2

*MA = Mean Amplitude*



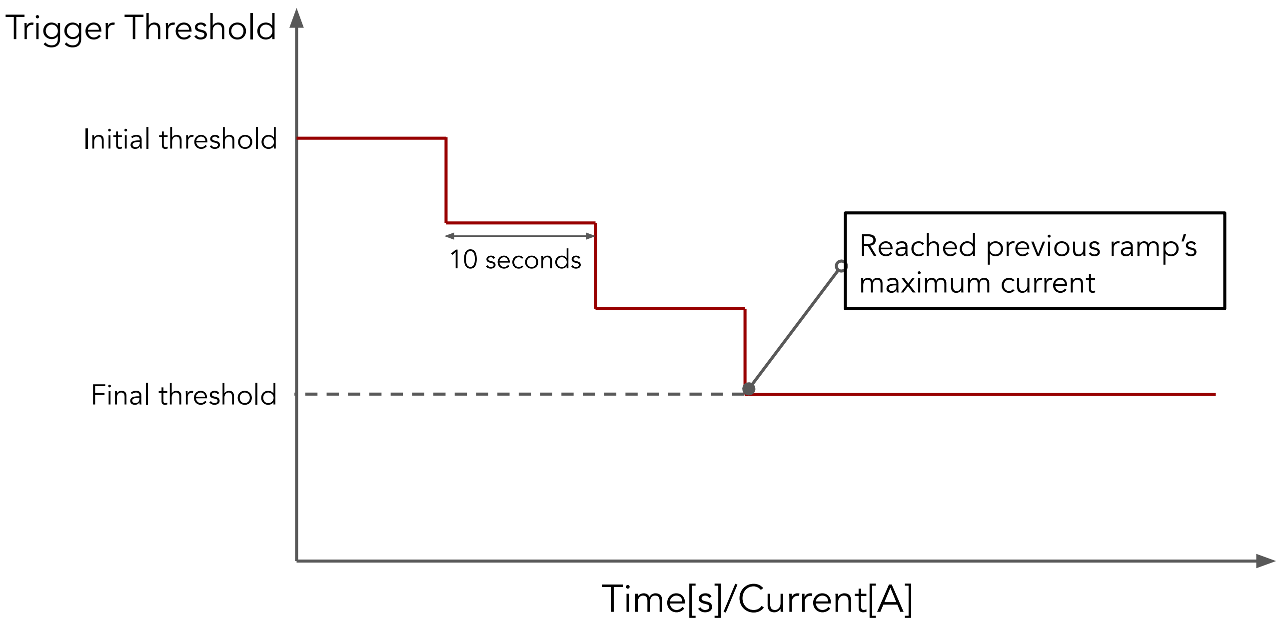
# Visualize latent space



# Threshold setting for registering anomalies

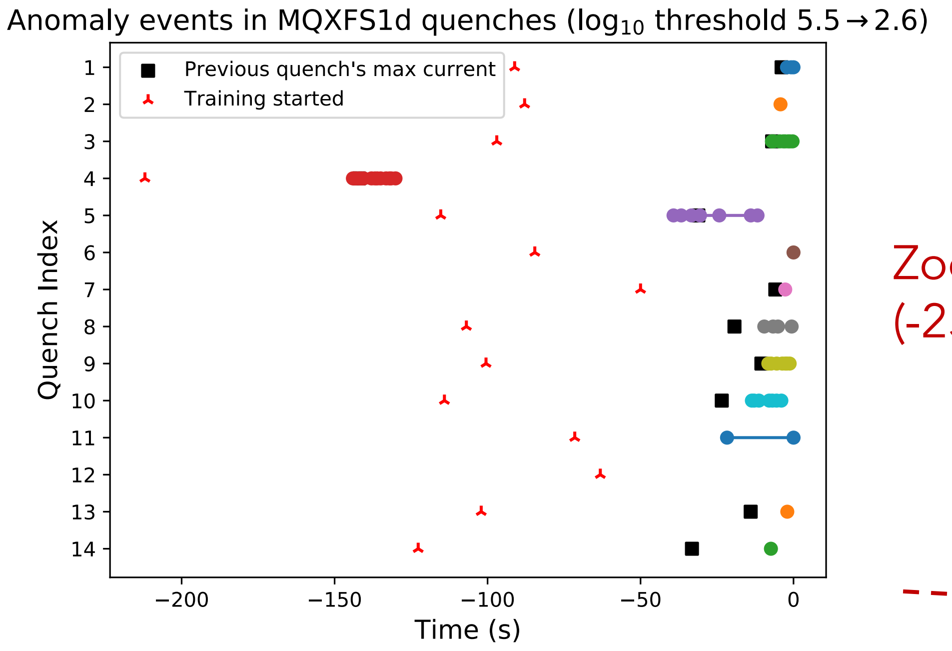
- Quenches at different currents may have different characteristics.
- Events must be more "anomalous" at lower current. Prevent false positives.



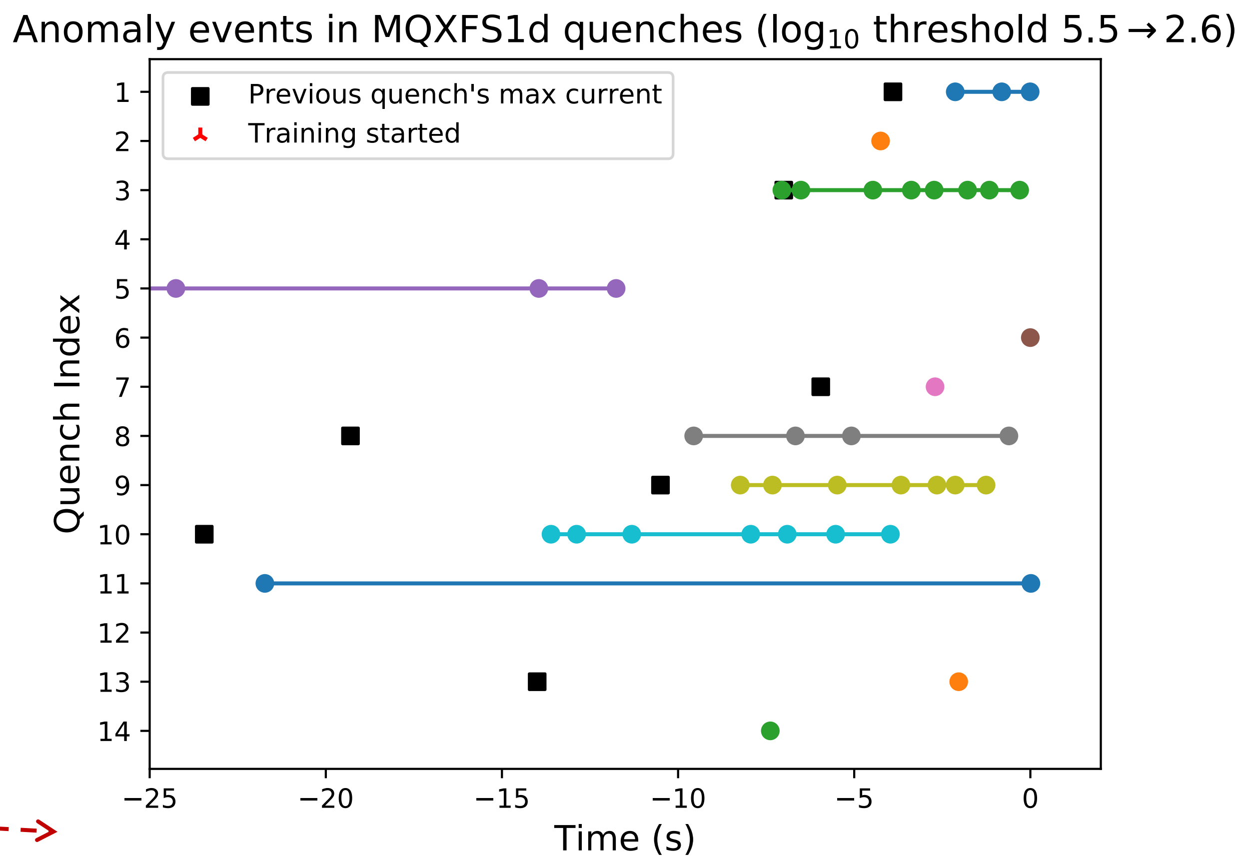
# Define Procedure

- Objective: Define a set procedure on known data and use it on unseen data to see what happens.
- Procedure: Start **dynamic learning** when current reached **9/10 max current of previous quench.**
  - Operating under magnet training condition.
  - Quench in next ramp wouldn't drop too much.
- Tried this out on MQXFS1d (5 sensors) first, and a **randomized experiment** with MDPCT1b (2 sensors).

# Initial observations in MQXFS1d (5 sensors) data

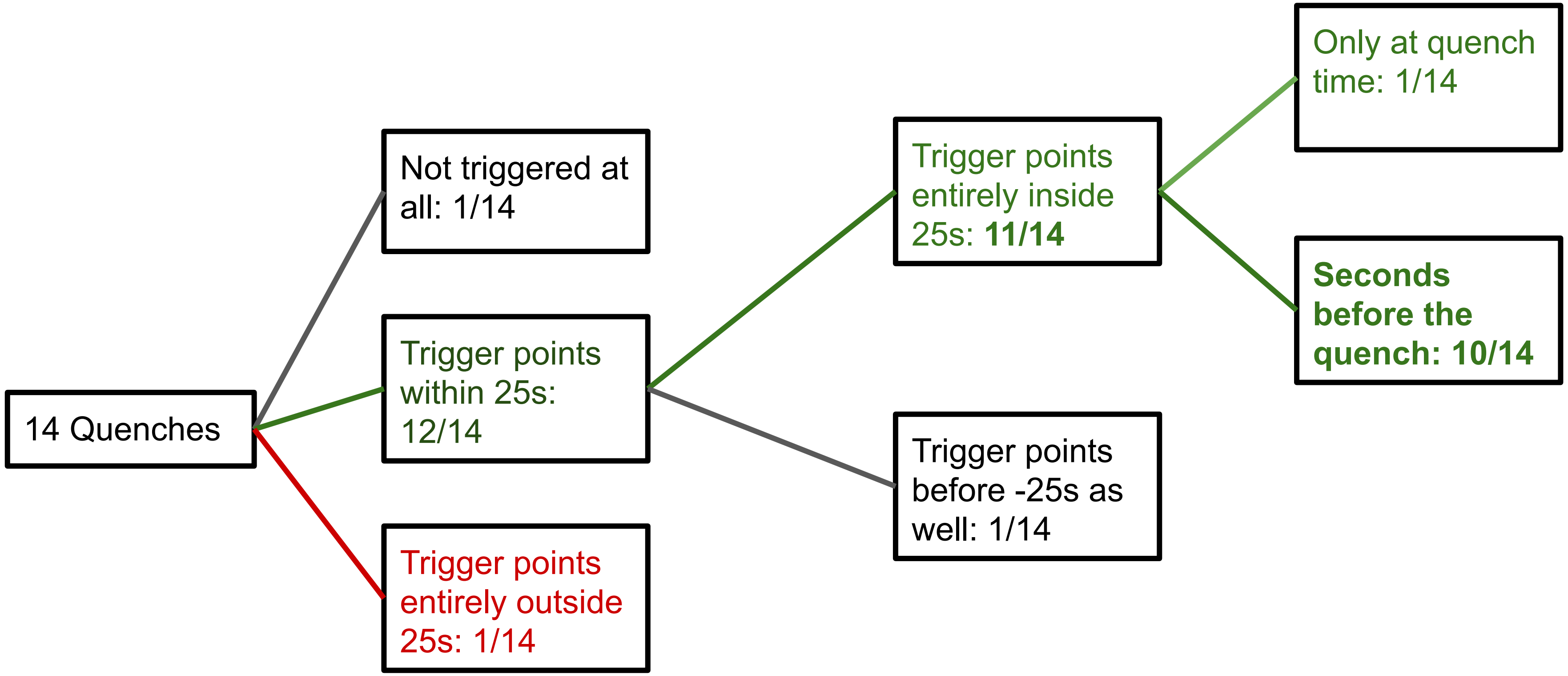



Zoomed in (-25,0)s



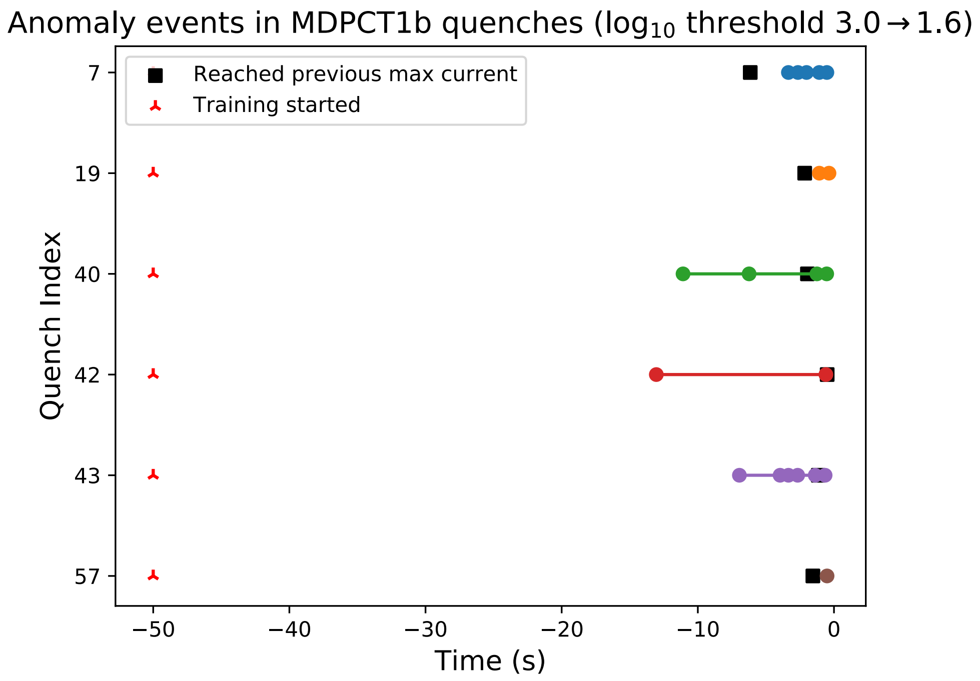

*Found interesting events near quench!*

# Categorize anomalies in quenches



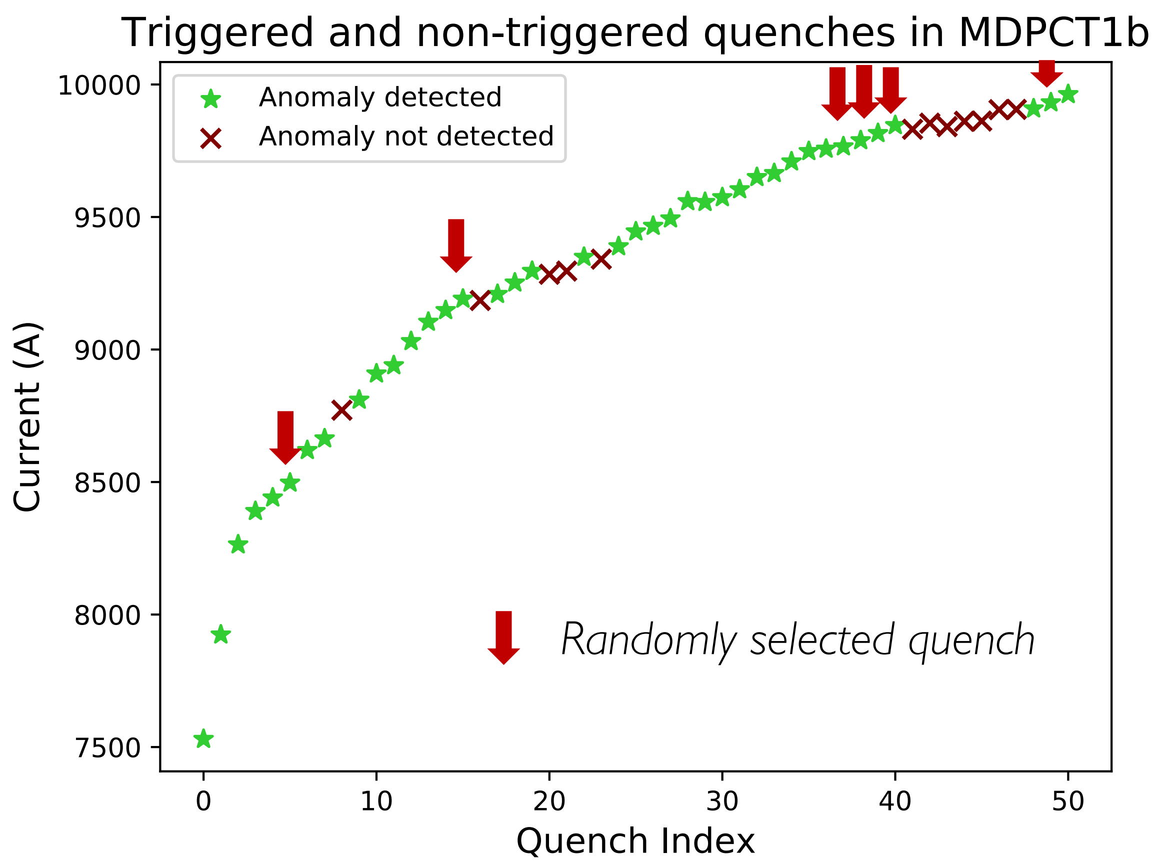
# Experiment on MDPCT1b (2 sensors) data

- Randomly picked ~10% of the quenches to tune the initial and final thresholds. (Started from 3.0 and decrease until 1.6)



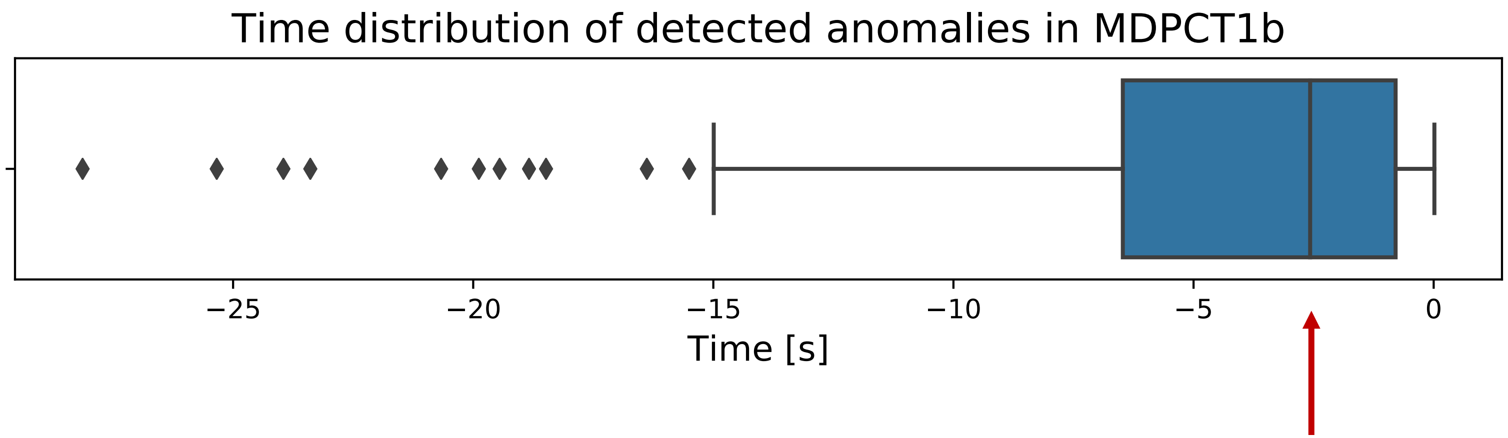
# Anomalous events on all quenches in MDPCT1b (2 sensors)

- Simulation of a real-time system.
- Triggered on 77% of the quenches.



# Anomalies time distribution in MDPCT1b (2 sensors)

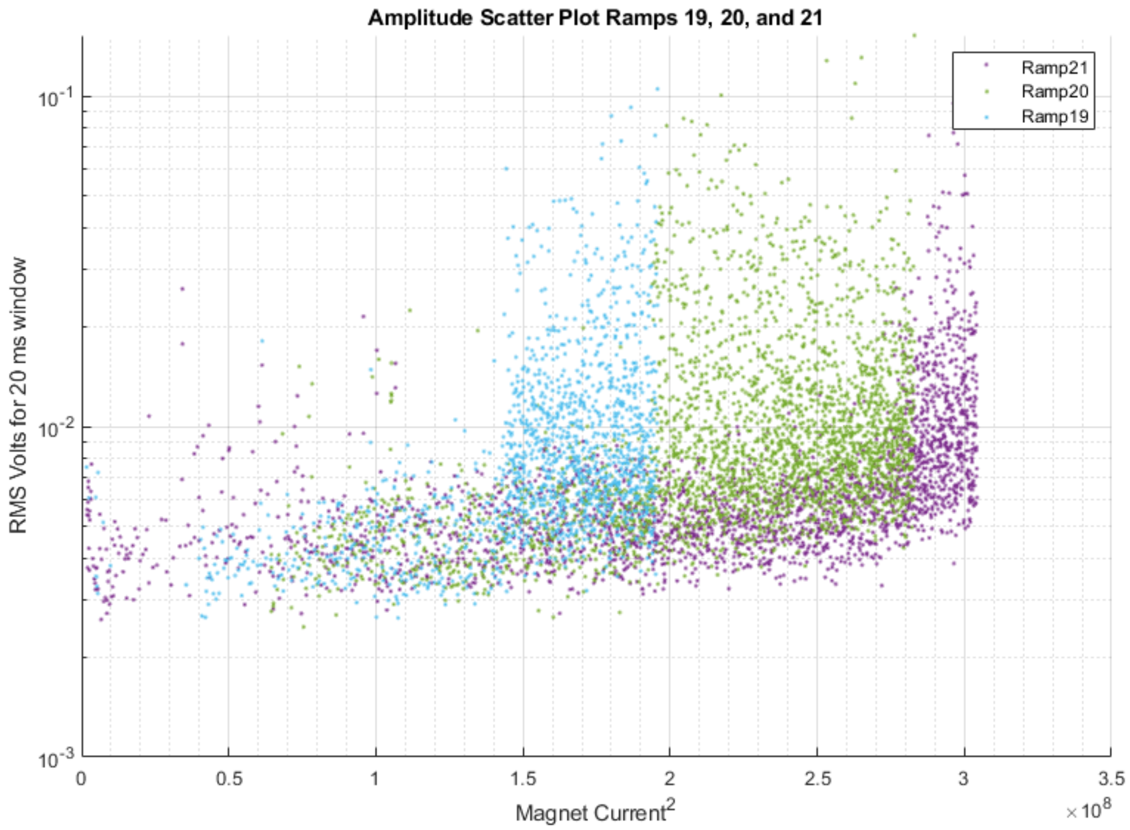
- Nearly all the detected events fall within 15 seconds.




Median is around 2.5 seconds before quench

# False Positives?

- Kaiser effects can interfere with triggering, since it introduces noises into the system.
- However, the system is still able to detect anomalies in quenches that should not have Kaiser effects. Or anomalies before Kaiser effects should happen.
- Future plans include better characterizations of the events not related to Kaiser effects.




*Kaiser effect: acoustic signals become much noisier if the current reaches the previous quench's current.*

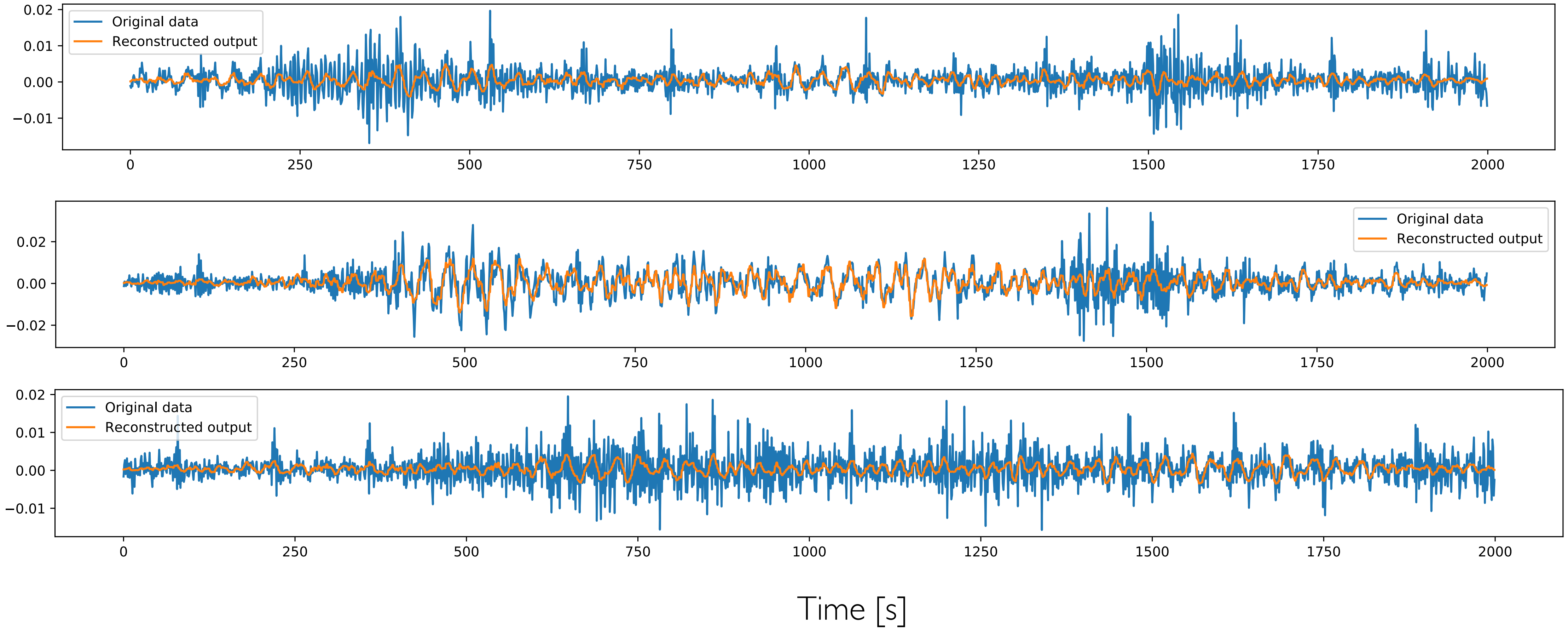
# Convolutional Neural Network

Exploratory

# CNN for reconstruction of raw signals

Exploratory

Different segments of training data



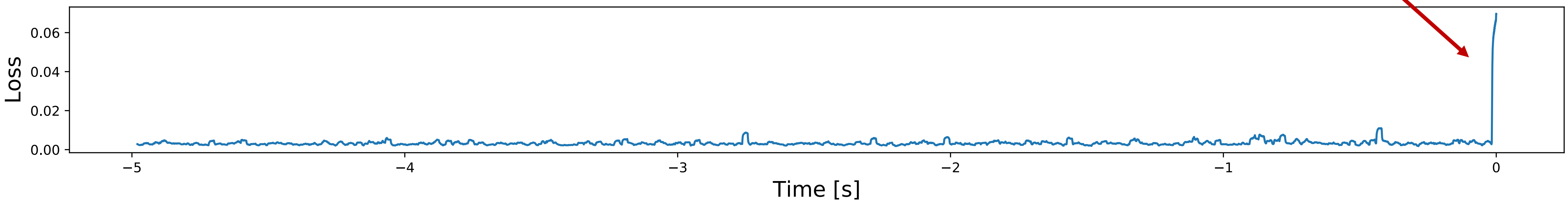
# CNN anomaly

Exploratory

- No extensive experiment yet, but promising as an alternative to DNN (no need to manually pick features).

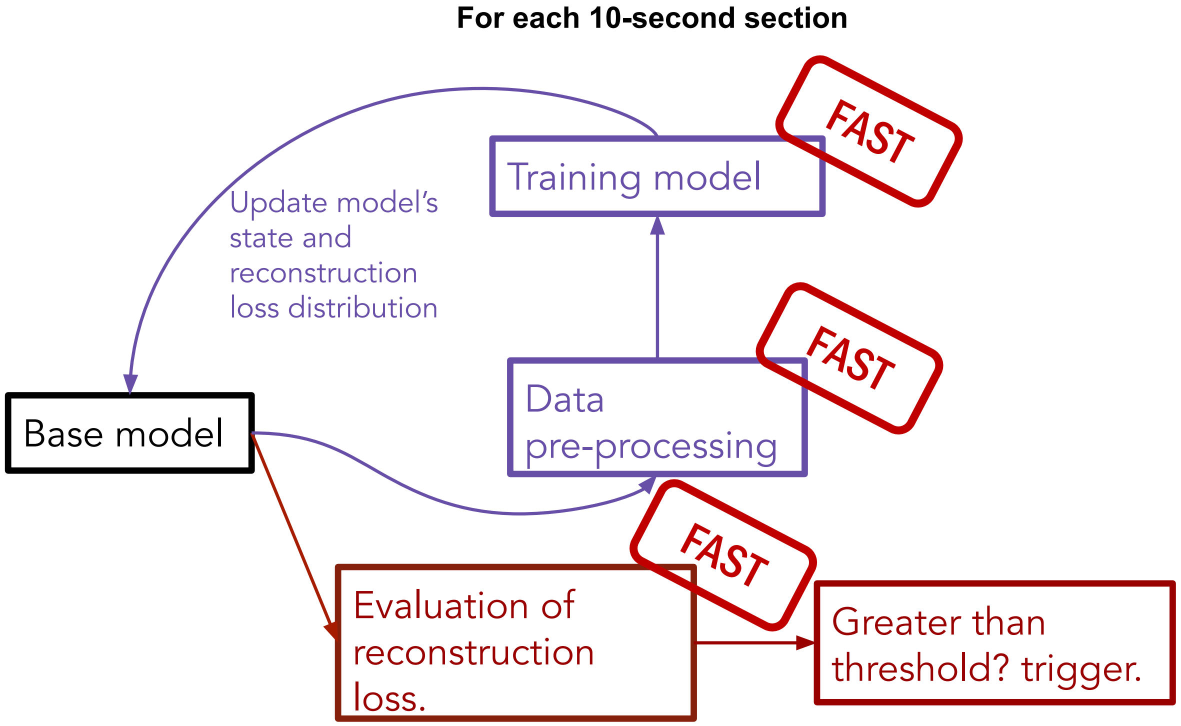
Can already recognize the same thing as DNN.

CNN's reconstruction loss near quench (Quench Index 22 – 5 sensors).



# Hardware implementation

- With increasing data rate, need to perform all steps as fast as possible to being a real "dynamic/online" system.



# Inference FAST

hls4ml

- The hls4ml software package, developed by collaborators at FNAL and else where, can be used to implement **deep learning models for fast inference on FPGAs.**

| Model | Latency (cycles) | Interval (cycles) | DSP ($10^3$) | LUT ($10^3$) | FF ($10^3$) | BRAM (Mb) |
|---|---|---|---|---|---|---|
| MQXFS1d | 21 (**105ns**) | 1 | 0.72 [13%] | 16.7 [2%] | 39.2 [5%] | 0.018 [0%] |
| MDPCT1b | 20 (**100ns**) | 1 | 0.26 [4%] | 14.8 [1%] | 5.9 [1%] | 0.01125 [0%] |

# Data pre-processing and model training

FAST

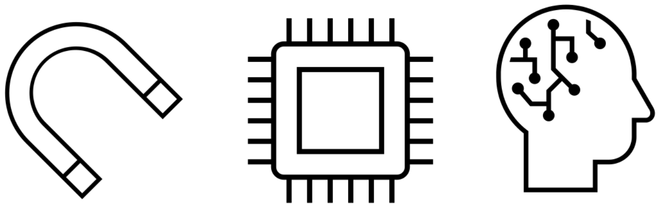
- Fast data pre-processing can be done by implementing them on FPGAs. But this depends on the complexity of the process.

- There are no worked out solution for fast training, but potential solution might include putting a GPU on site. 👉 This also depends on the model's complexity.

# Summary

- Investigated deep learning anomaly detection techniques for clues of "quench precursors."
- Point to interesting anomalous events seconds before the quench in acoustic data. Used these events to "predict" 77% of quenches in a randomized experiment within 15 seconds.
- Realistic dynamic learning workflow for real-time processing of streaming data.

# Outlook



- Study and understand the detected events and their relationship with the quench.

- Improve data taking process, expand to many more input variables, and create curated datasets for benchmarking.

- Can use the same ideas for monitoring of several other types of operations at Fermilab. And potentially applicable to high temperature superconductors.

Superconducting magnets and quenches

*Questions*

Hardware setup and data acquisition

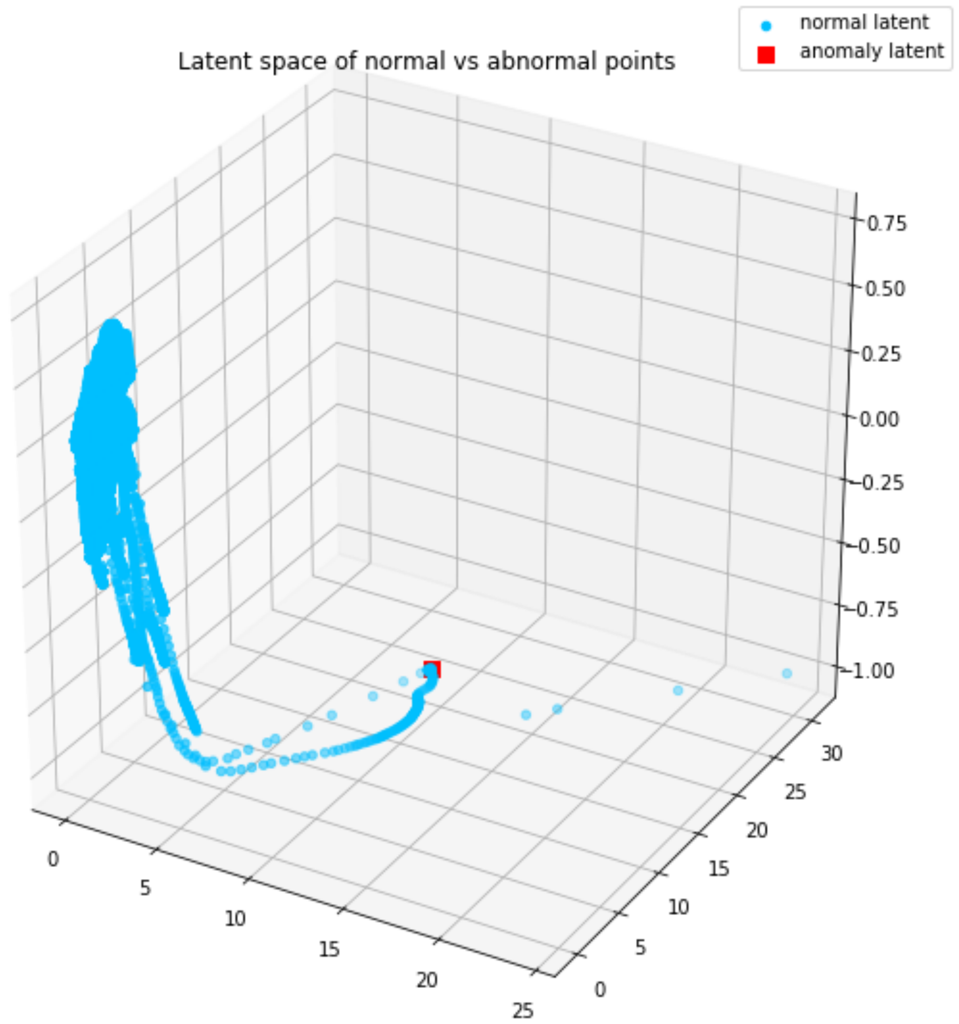
*Questions*

Anomaly detection using machine learning

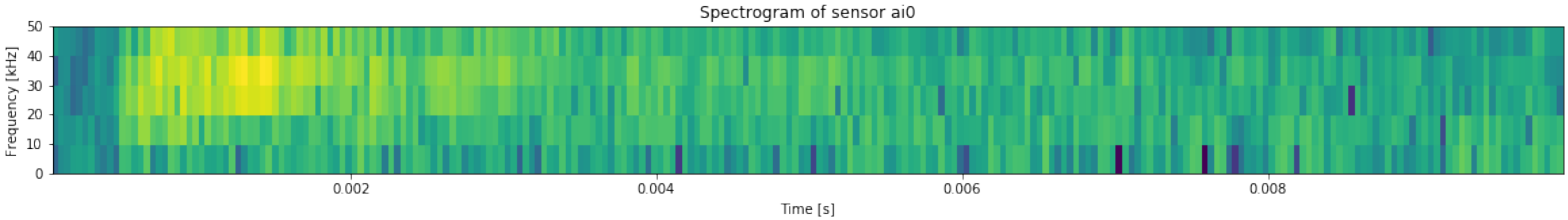
*Questions*

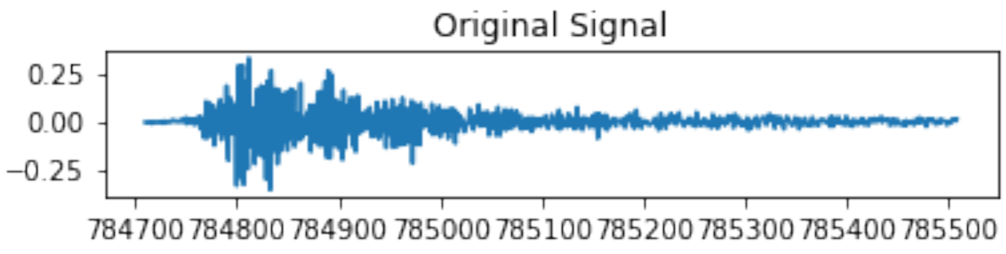
# Backups

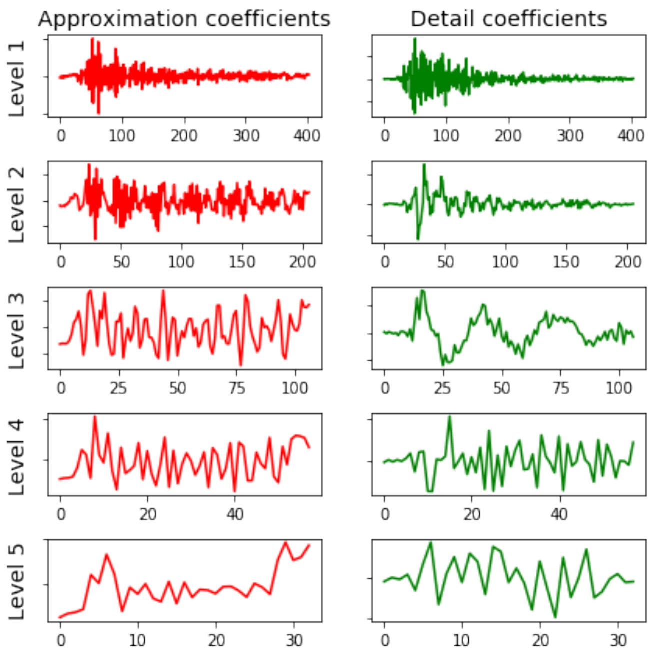
# 3D latent space



# Event characterization

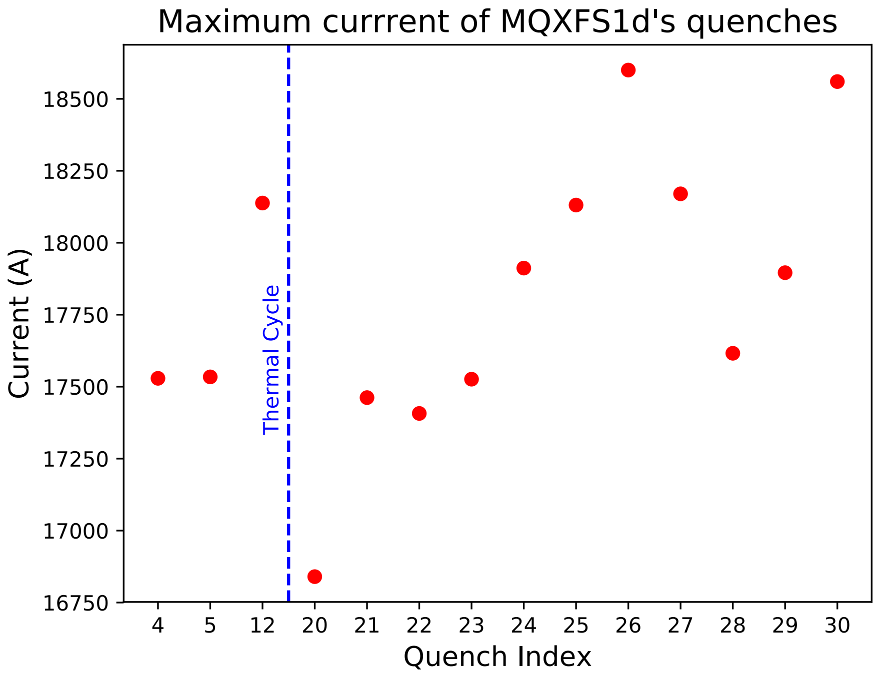


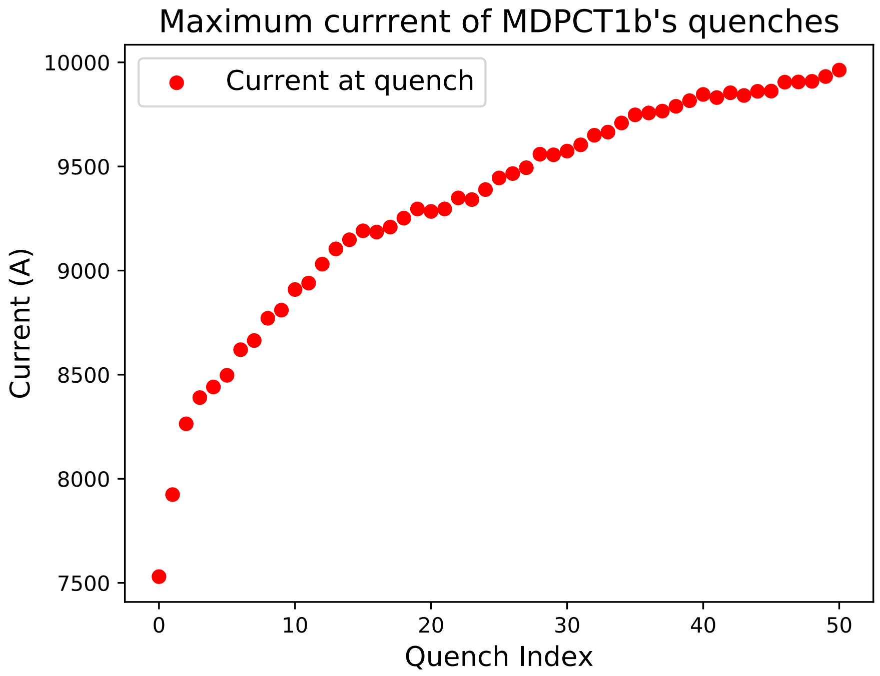
# Quench data in two magnets

## MQXFS1d (5 sensors)




*Not totally consecutive quench*

## MDPCT1b (2 sensors)



# General setup

- We attached several acoustic sensors around the magnet