A Real-time tool for anomaly detection in Advanced Virgo's Auxiliary channels

Luca Negri - Utrecht University
l.negri@uu.nl

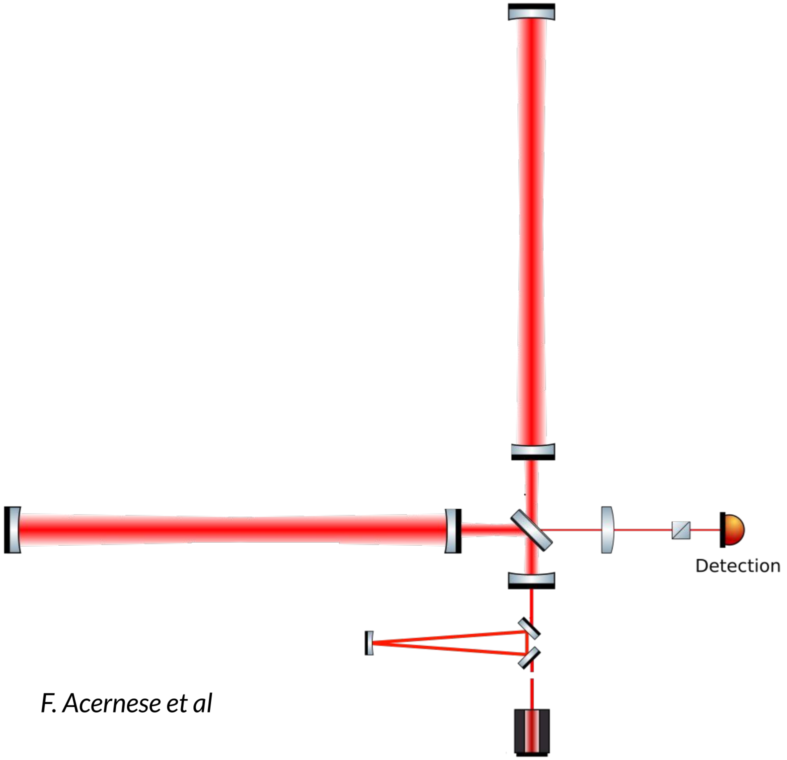
EuCAIFCon - April 30 2024

Luca Negri, Alberto Gennai, Massimiliano Razzano

What are auxiliary channels?

What are Auxiliary channels?

$$h(t) = \frac{\Delta L}{L}$$

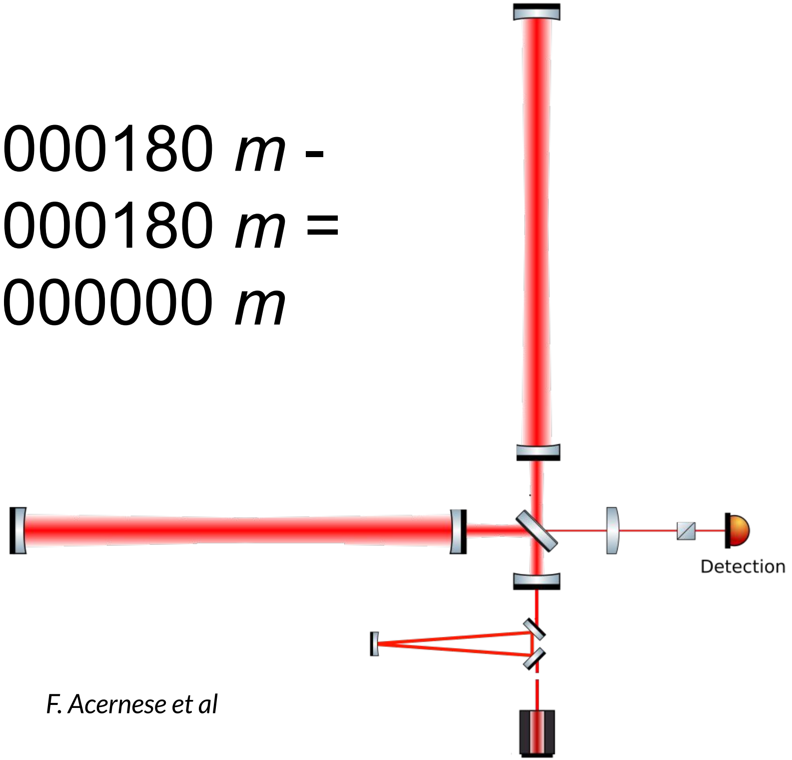


F. Acernese et al

What are Auxiliary channels?

$$h(t) = \frac{\Delta L}{L}$$

3000.000180 *m* -
3000.000180 *m* =
0.000000 *m*



F. Acernese et al

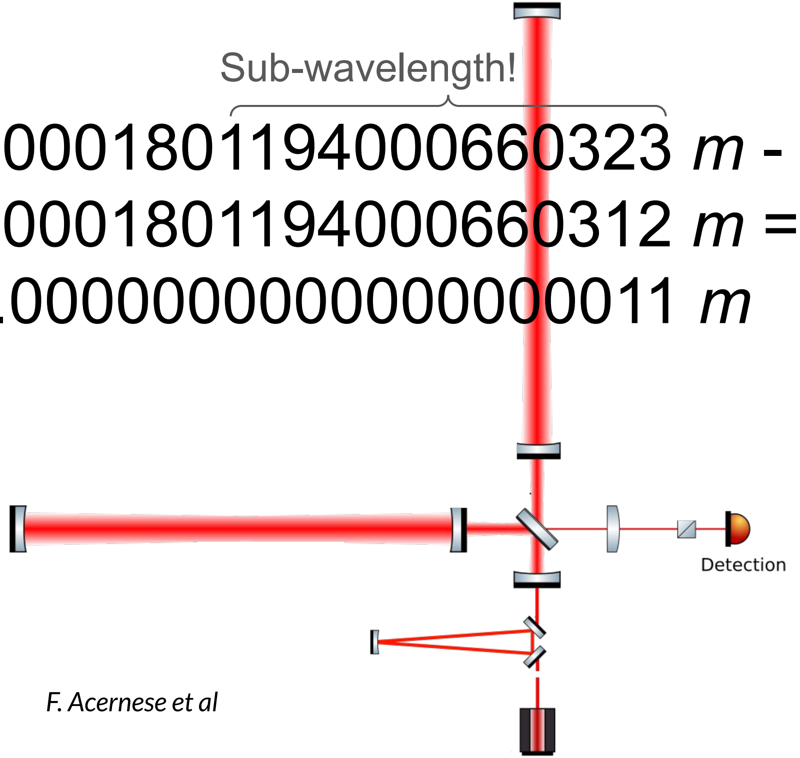
What are Auxiliary channels?

$$h(t) = \frac{\Delta L}{L}$$

$$h(t) \sim 10^{-21}$$

Sub-wavelength!

3000.0001801194000660323 *m* -
3000.0001801194000660312 *m* =
0.0000000000000000011 *m*

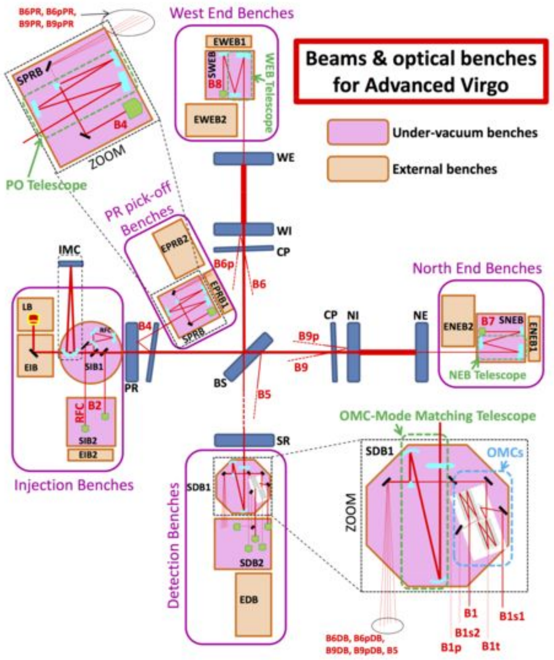


F. Acernese et al

What are Auxiliary channels?

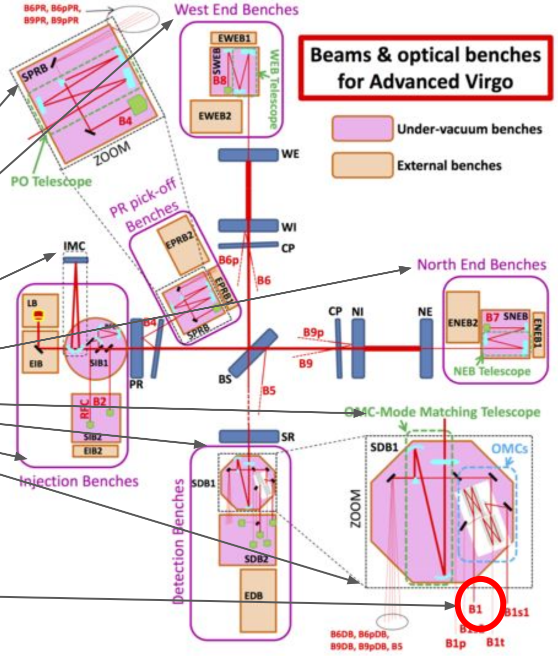
Virgo is a very complex instrument!

- Thermal control
- Ultra high vacuum
- Seismic attenuation
- Environmental monitoring
- Laser stability
- ~ 7 optical cavities kept at resonance
- Feedback control loops
- Input and output mode cleaners
- Frequency dependent light squeezing (dark magic)
- …



What are Auxiliary channels?

Auxiliary channels constantly control and monitor the instrument and its surroundings.



Auxiliary channels

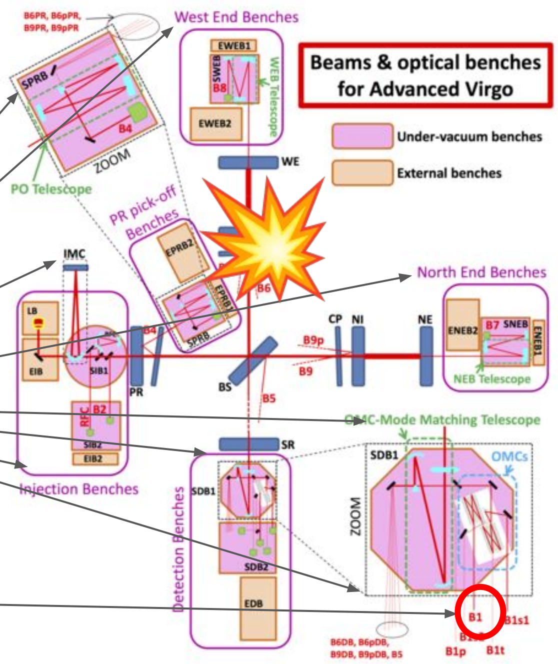
□ 10^5

$h(t)$

~1

What are Auxiliary channels?

Auxiliary channels constantly control and monitor the instrument and its surroundings.



Auxiliary channels

□ 10^5

…but how do we know when something goes wrong?

$h(t)$

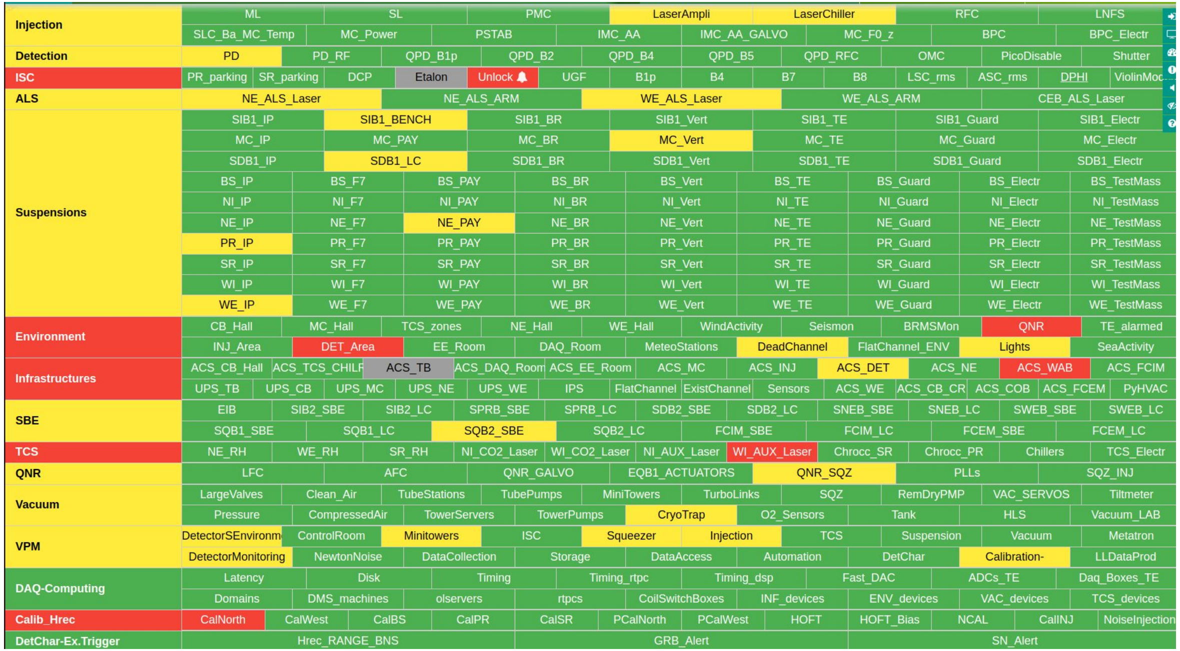
~1

How are we monitoring this complex system?

Tools currently in use, like the Data Monitoring System (DMS) or Omicron, are based on linear algorithms,

But …

- The instrument has many nonlinear behaviours
- They need constant manual retuning by experts of the instrument



Screenshot of the DMS

Non-linearity …

Large amounts of data …

Data with high dimensionality …

Can machine learning help?

Can we build a DMS-like anomaly detection tool based on AI?

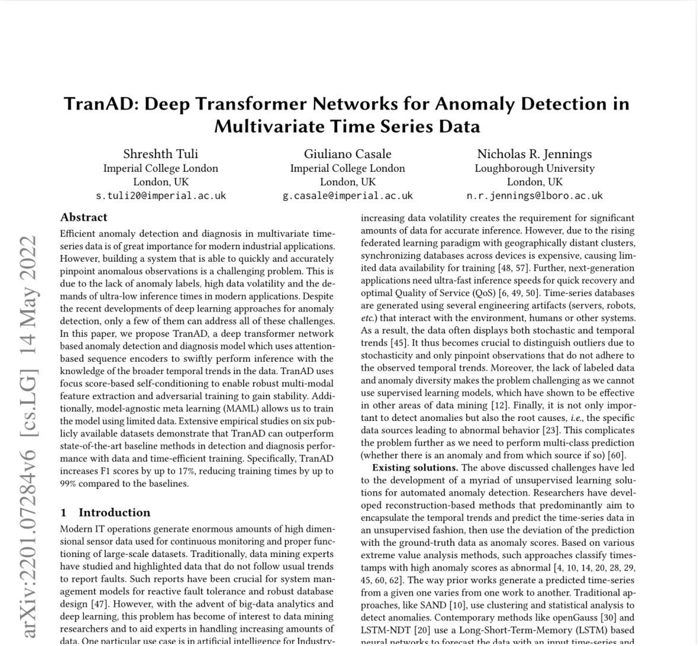
We want an algorithm capable of detecting abnormal behaviour in real-time, in order to swiftly notify the instrument operators when & where something is wrong.

It must be:

- Unsupervised
- Multi-channel
- Work on minimal assumptions
- Flexible
- Computationally cheap

We landed on the TranAD architecture by S. Tuli

et al (arxiv:2201.07284)



TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data

Shreshth Tuli
Imperial College London
London, UK
s.tuli20@imperial.ac.uk

Giuliano Casale
Imperial College London
London, UK
g.casale@imperial.ac.uk

Nicholas R. Jennings
Loughborough University
London, UK
n.r.jennings@lboro.ac.uk

arXiv:2201.07284v6 [cs.LG] 14 May 2022

Abstract

Efficient anomaly detection and diagnosis in multivariate time-series data is of great importance for modern industrial applications. However, building a system that is able to quickly and accurately pinpoint anomalous observations is a challenging problem. This is due to the lack of anomaly labels, high data volatility and the demands of ultra-low inference times in modern applications. Despite the recent developments of deep learning approaches for anomaly detection, only a few of them can address all of these challenges. In this paper, we propose TranAD, a deep transformer network based anomaly detection and diagnosis model which uses attention-based sequence encoders to swiftly perform inference with the knowledge of the broader temporal trends in the data. TranAD uses focus score-based self-conditioning to enable robust multi-modal feature extraction and adversarial training to gain stability. Additionally, model-agnostic meta learning (MAML) allows us to train the model using limited data. Extensive empirical studies on six publicly available datasets demonstrate that TranAD can outperform state-of-the-art baseline methods in detection and diagnosis performance with data and time-efficient training. Specifically, TranAD increases F1 scores by up to 17%, reducing training times by up to 99% compared to the baselines.

1 Introduction

Modern IT operations generate enormous amounts of high dimensional sensor data used for continuous monitoring and proper functioning of large-scale datasets. Traditionally, data mining experts have studied and highlighted data that do not follow usual trends to report faults. Such reports have been crucial for system management models for reactive fault tolerance and robust database design [47]. However, with the advent of big-data analytics and deep learning, this problem has become of interest to data mining researchers and to aid experts in handling increasing amounts of data. One particular use case is in artificial intelligence for Industry-

increasing data volatility creates the requirement for significant amounts of data for accurate inference. However, due to the rising federated learning paradigm with geographically distant clusters, synchronizing databases across devices is expensive, causing limited data availability for training [48, 57]. Further, next-generation applications need ultra-fast inference speeds for quick recovery and optimal Quality of Service (QoS) [6, 49, 50]. Time-series databases are generated using several engineering artifacts (servers, robots, *etc.*) that interact with the environment, humans or other systems. As a result, the data often displays both stochastic and temporal trends [45]. It thus becomes crucial to distinguish outliers due to stochasticity and only pinpoint observations that do not adhere to the observed temporal trends. Moreover, the lack of labeled data and anomaly diversity makes the problem challenging as we cannot use supervised learning models, which have shown to be effective in other areas of data mining [12]. Finally, it is not only important to detect anomalies but also the root causes, *i.e.*, the specific data sources leading to abnormal behavior [23]. This complicates the problem further as we need to perform multi-class prediction (whether there is an anomaly and from which source if so) [60].

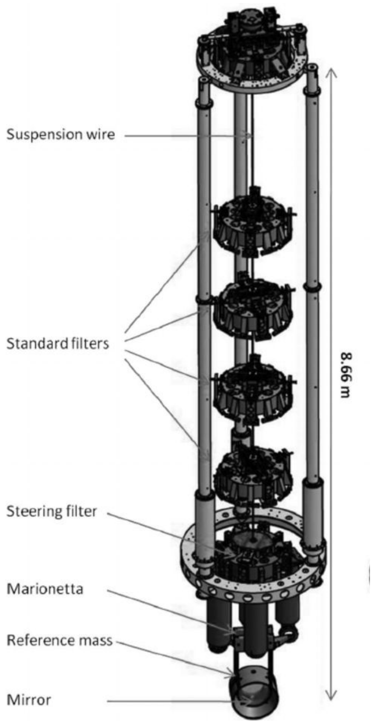
Existing solutions. The above discussed challenges have led to the development of a myriad of unsupervised learning solutions for automated anomaly detection. Researchers have developed reconstruction-based methods that predominantly aim to encapsulate the temporal trends and predict the time-series data in an unsupervised fashion, then use the deviation of the prediction with the ground-truth data as anomaly scores. Based on various extreme value analysis methods, such approaches classify timestamps with high anomaly scores as abnormal [4, 10, 14, 20, 28, 29, 45, 60, 62]. The way prior works generate a predicted time-series from a given one varies from one work to another. Traditional approaches, like SAND [10], use clustering and statistical analysis to detect anomalies. Contemporary methods like openGauss [30] and LSTM-NDT [20] use a Long-Short-Term-Memory (LSTM) based neural networks to forecast the data with an input time-series and

Dataset and Methods

Dataset: the SuperAttenuators

- Mirror suspension in Advanced Virgo
- Achieves in-band passive attenuation of 10 orders of magnitude!
- Offers a platform for the actuators and other instruments
- There are 10 of them in AdV
- Monitored by ~ 600 sensors
- This system is well understood (physically)
- Has a known response to ground motion.

We considered only 4 SATs (BS, PR, NI, WI) that are located in the same building

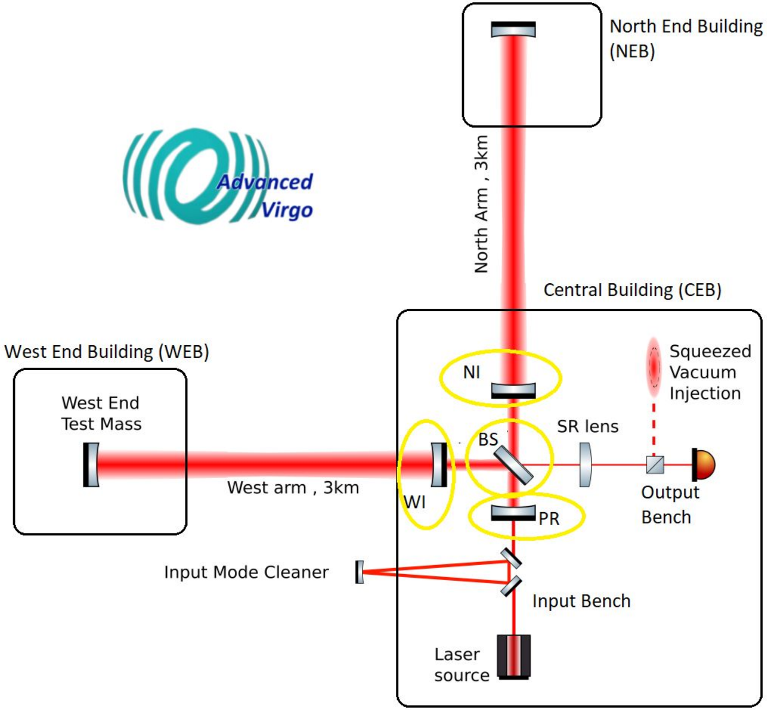


V. Boschi et al

Dataset Challenges

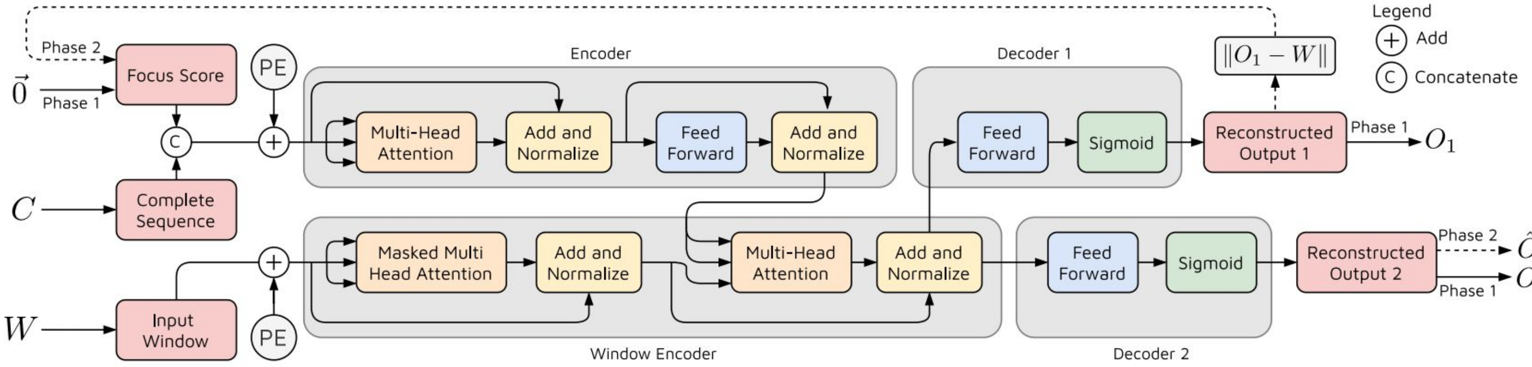
- Data is very heterogeneous
- White noise dominates
- Large dynamic range
- High sample rate (500 Hz)
- Many different operation conditions

Anomalies come in many different shape and sizes we may not even be aware of!



F. Acernese et al

ML architecture : TranAD (*S. Tuli et al*)



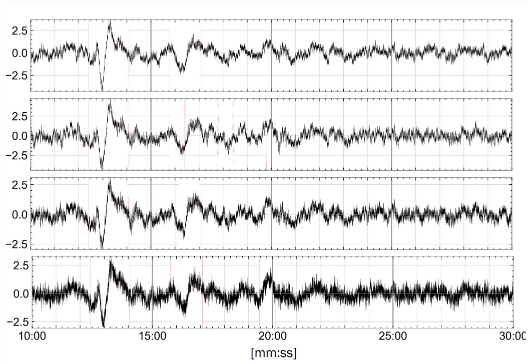
- Transformer based encoder & 2 decoders architecture
- Network tries to reconstruct the signal
- Training in 2 phases :
    - 1st phase: Both decoders try and minimize reconstruction loss
    - 2nd phase: Naughty decoder maximizes reconstruction error, while also having access to the loss of the good decoder in phase 1 (Focus score)

$$\min_{\text{Decoder1}} \max_{\text{Decoder2}} \|\hat{O}_2 - W\|_2.$$

Adversarial training allows the algorithm to focus on small deviations.

The architecture also allows for inference of anomalies at both short and long timescales

Basic inference workflow



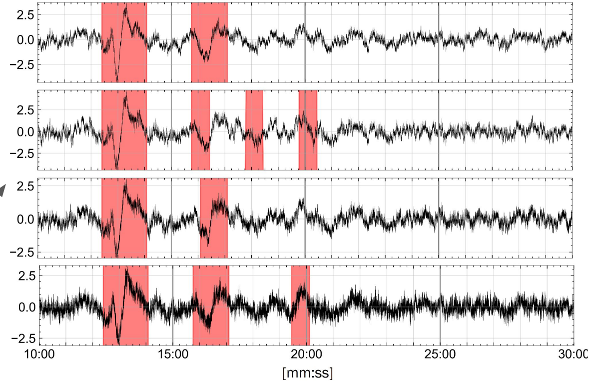
Downsampling to 500 Hz

Normalization: High-pass filter +

$$x_t \leftarrow \frac{x_t - \min(\mathcal{T})}{\max(\mathcal{T}) - \min(\mathcal{T}) + \epsilon'},$$

Training on dataset labelled as anomaly-free by experts

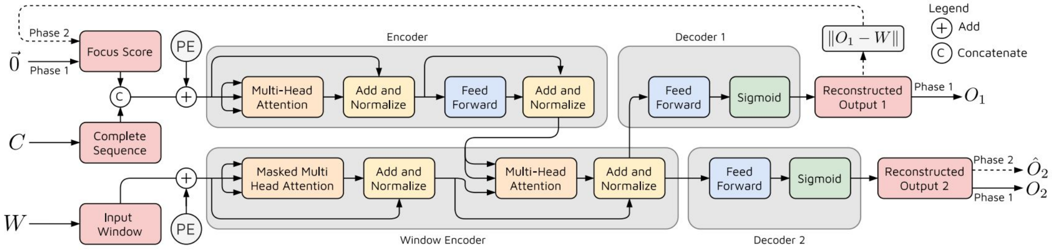
Anomaly threshold computed on training loss based on desired FAR



Reconstruction error used as anomaly score!

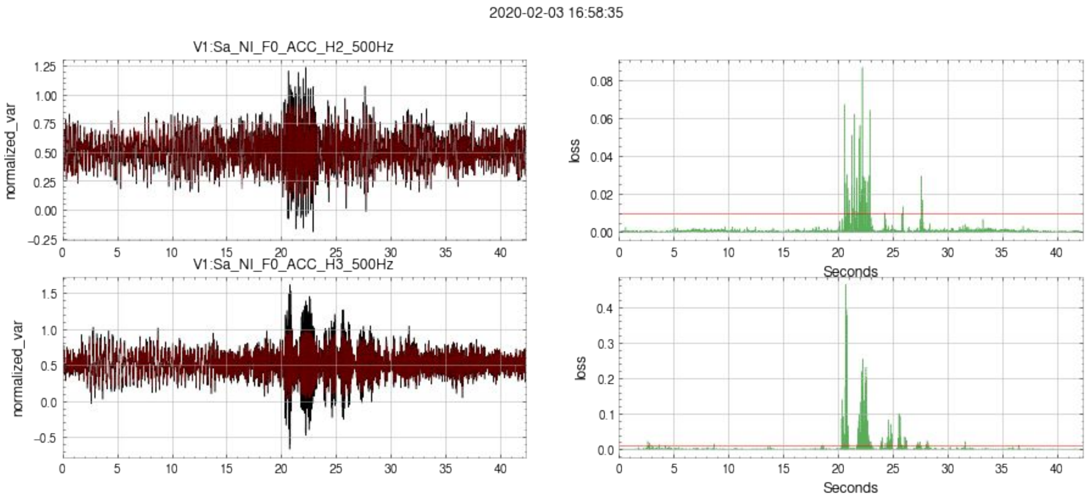
Compression

Reconstruction

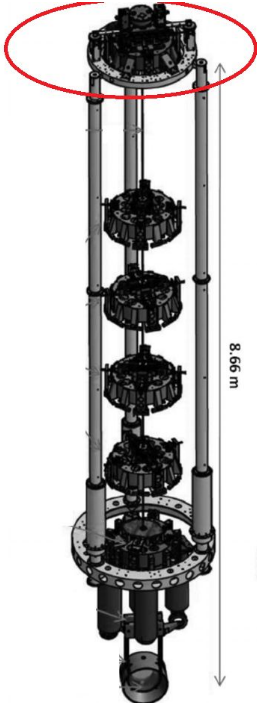


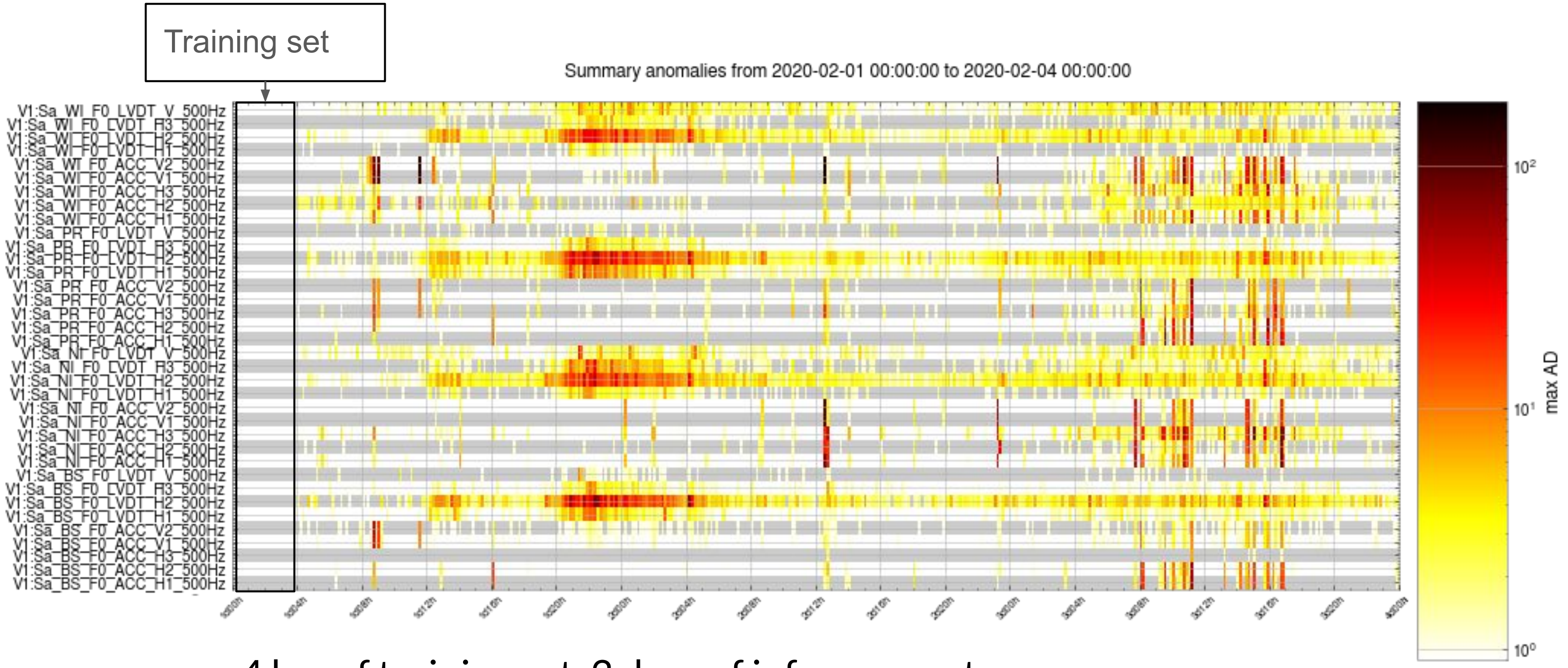
S. Tuli et al

Results


2020-02-03 16:58:35
V1:Sa_NI_F0_ACC_H2_500Hz
normalized_var
loss
Seconds
V1:Sa_NI_F0_ACC_H3_500Hz
normalized_var
loss
Seconds

Results: Upper part of the 4 SATs





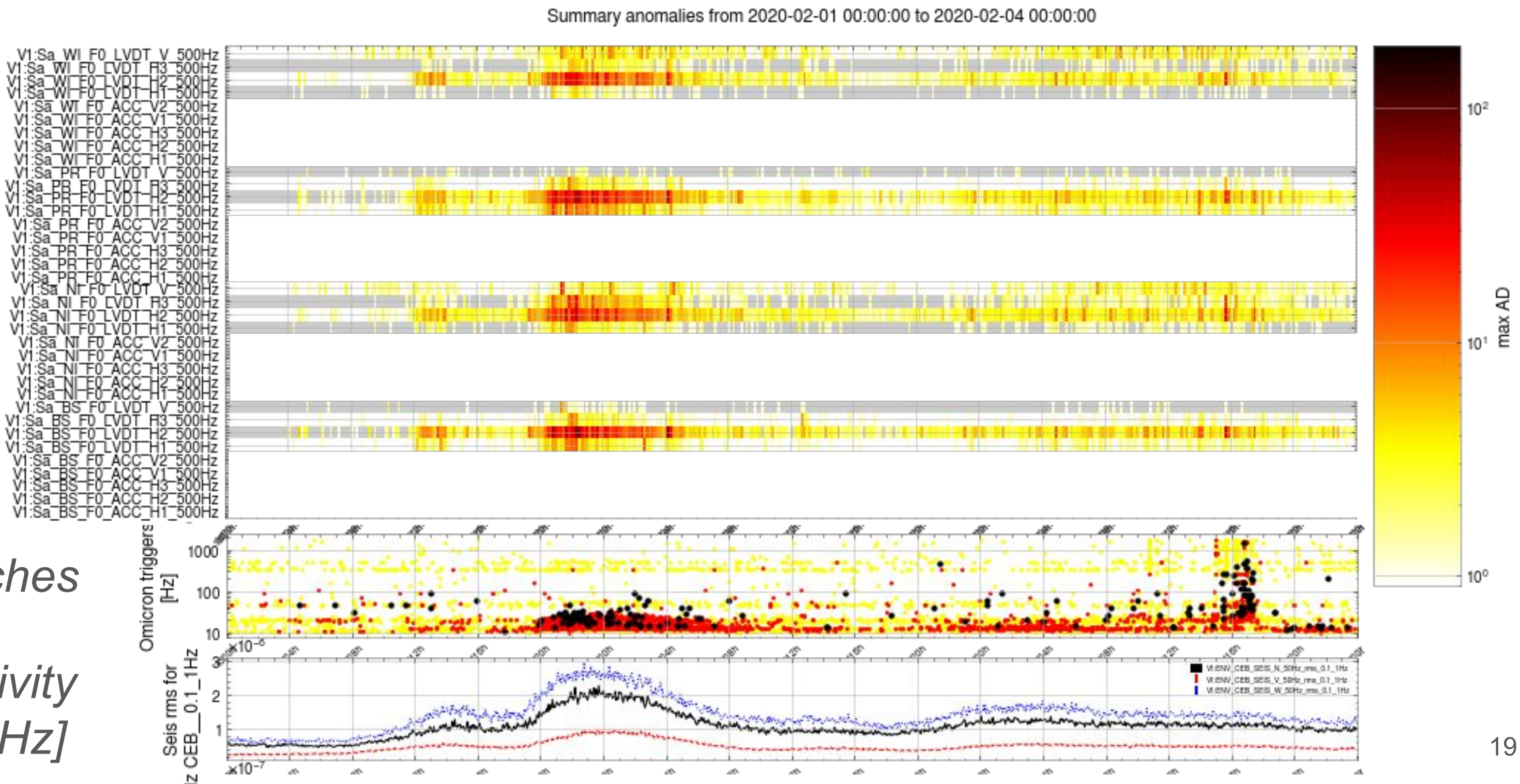
- 4 hrs of training set, 3 days of inference set
- 36 channels @ 500 Hz
- Total network has ~ 1.5e5 parameters
- 20 seconds of data -> 3 seconds of inference (on CPU!)

Results: Upper part of the 4 SATs : LVDTs

LVDTs mostly monitor low frequencies

h(t) glitches

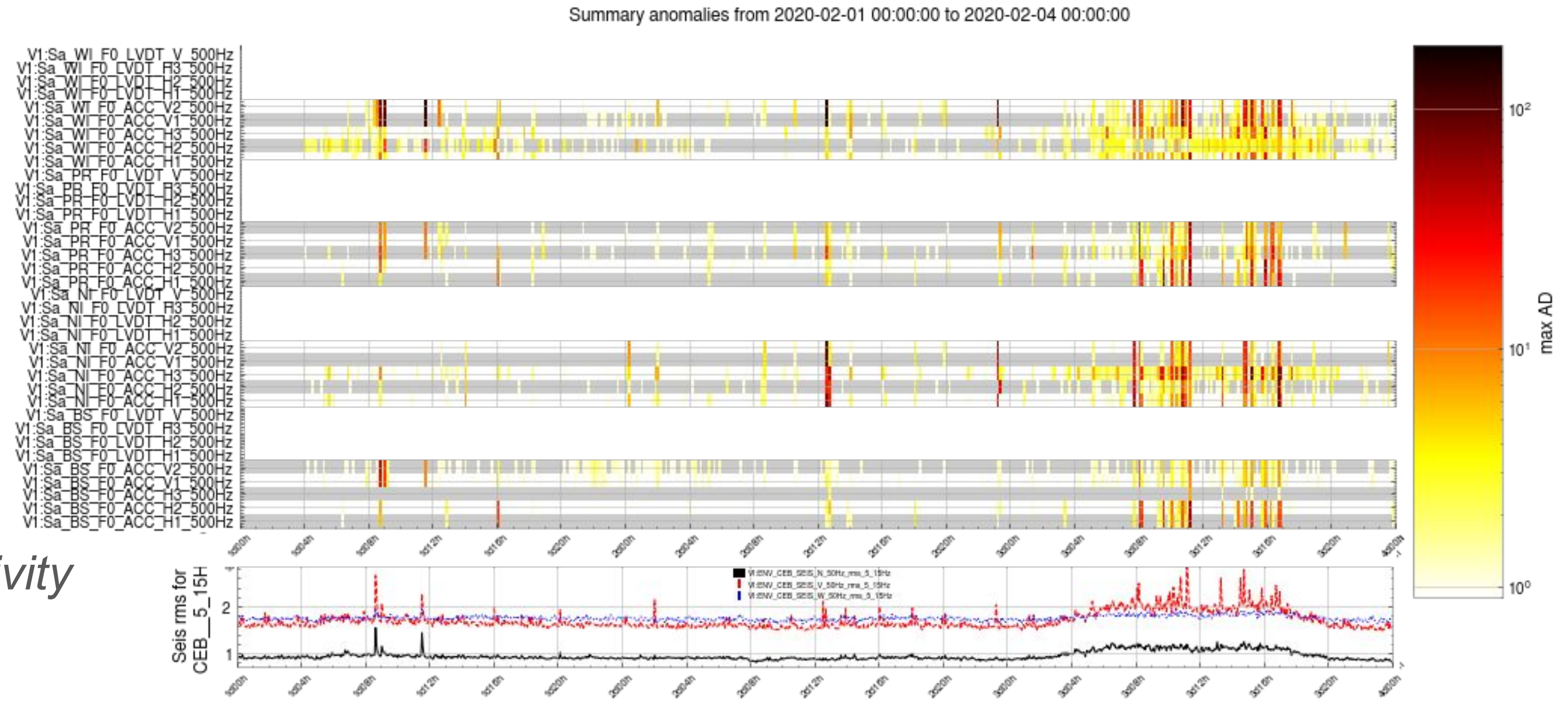
Sea activity [0.1 - 1 Hz]



Results: Upper part of the 4 SATs : Accelerometers

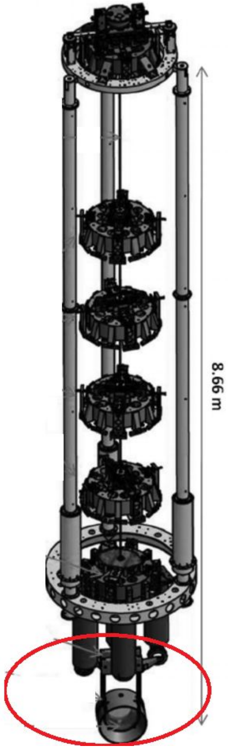
ACCs mostly monitor high frequencies

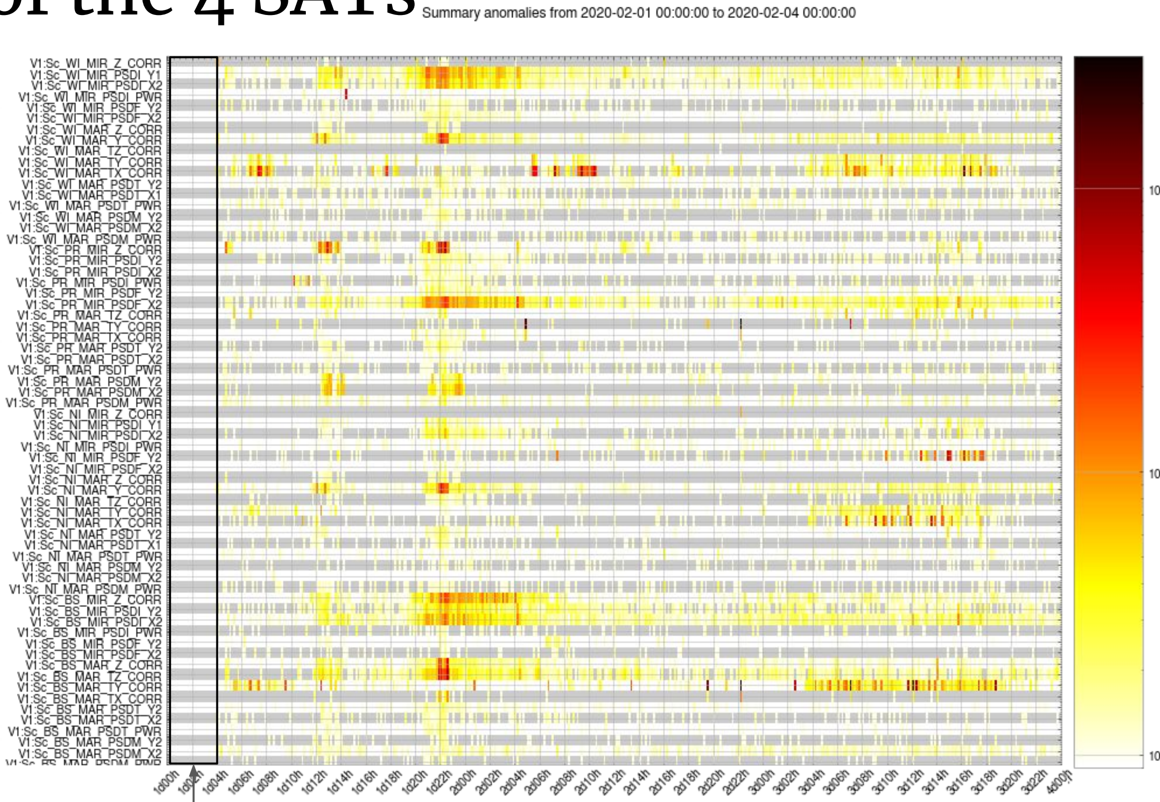
Human activity [5 - 15 Hz]



Results: Lower part of the 4 SATs

- 4 hrs of training set, 3 days of inference set
- 65 channels @ 10 kHz, downsampled to 500Hz
- Total network has ~ 3.5e5 parameters
- 20 seconds of data -> 5 seconds of inference (on CPU!)





Training set

But, can it run in real time?

Test run with non-ideal computing hardware (CPUs)

For inference on 100 seconds of data (65 channels @ 10 kHz)

- Data handling : ~ 15.5 s (mainly download time)
- Inference time : ~ 17.5 s

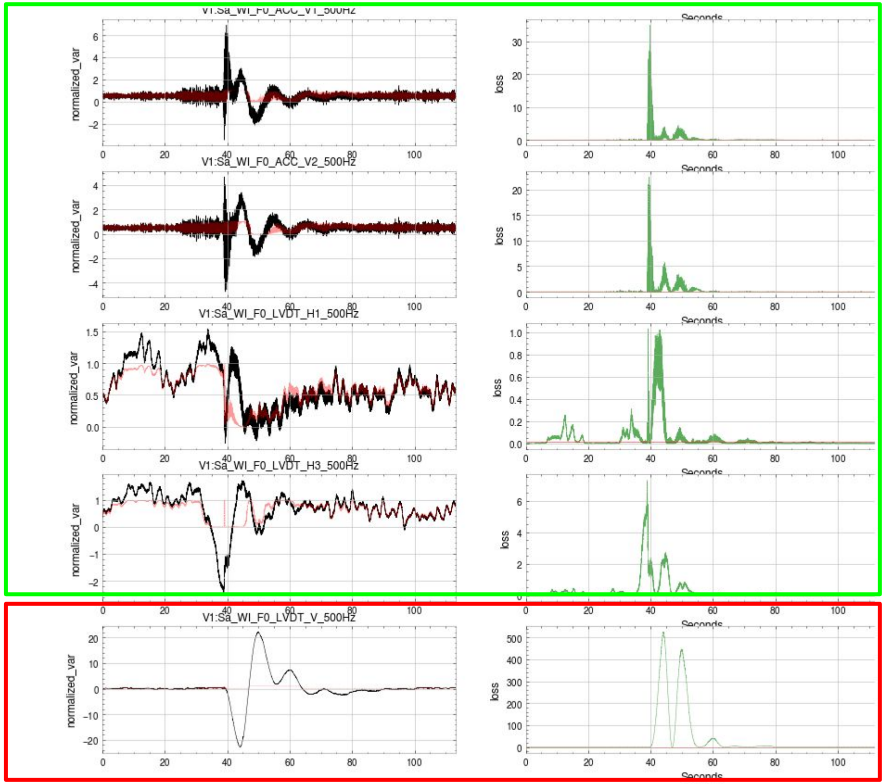
Total time = ~ 33 s < 100 s

Results: first use cases!

- WI during an unlock: we expect to find anomalies.
- And in fact we find them!
- … but the vertical WI LVDT has the highest anomaly score. Too high …
- SuperAttenuator experts are now looking into this

Ok-ish

Suspicious…



Conclusions

Algorithm shows promising results, it is capable of performing real-time anomaly detection with a decently low FAR.

But there is still a long way to go.

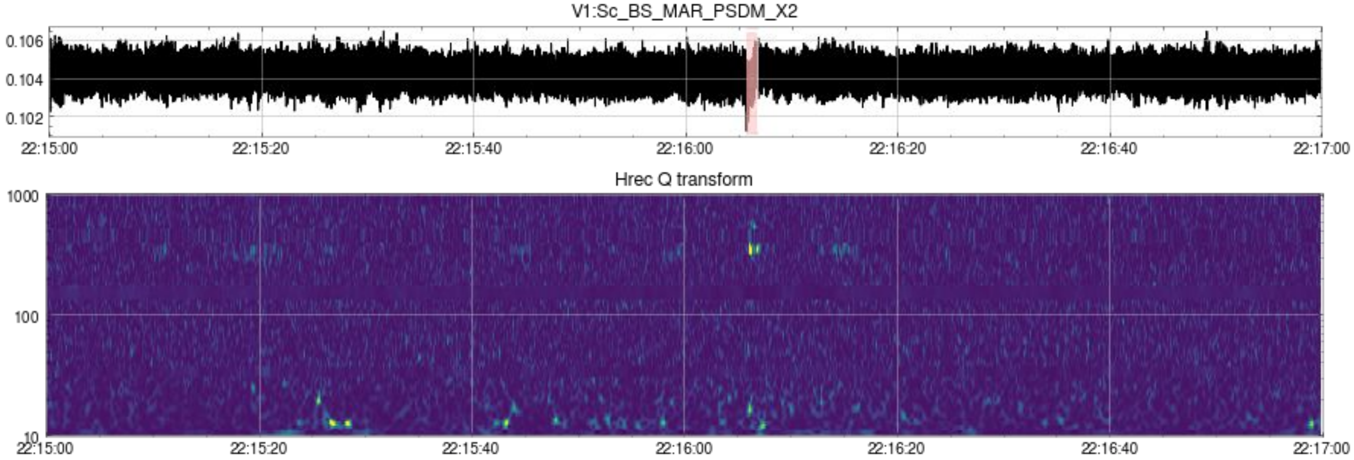
- Can this setup actually deal with many more channels?
- How much speedup can we gain with better hardware? (GPUs)
- How can we deal with an ever-changing instrument? (Periodic retraining)
- Can we implement in the algorithm some previous knowledge of the system? (physical nature of the system)

A paper is on the way.

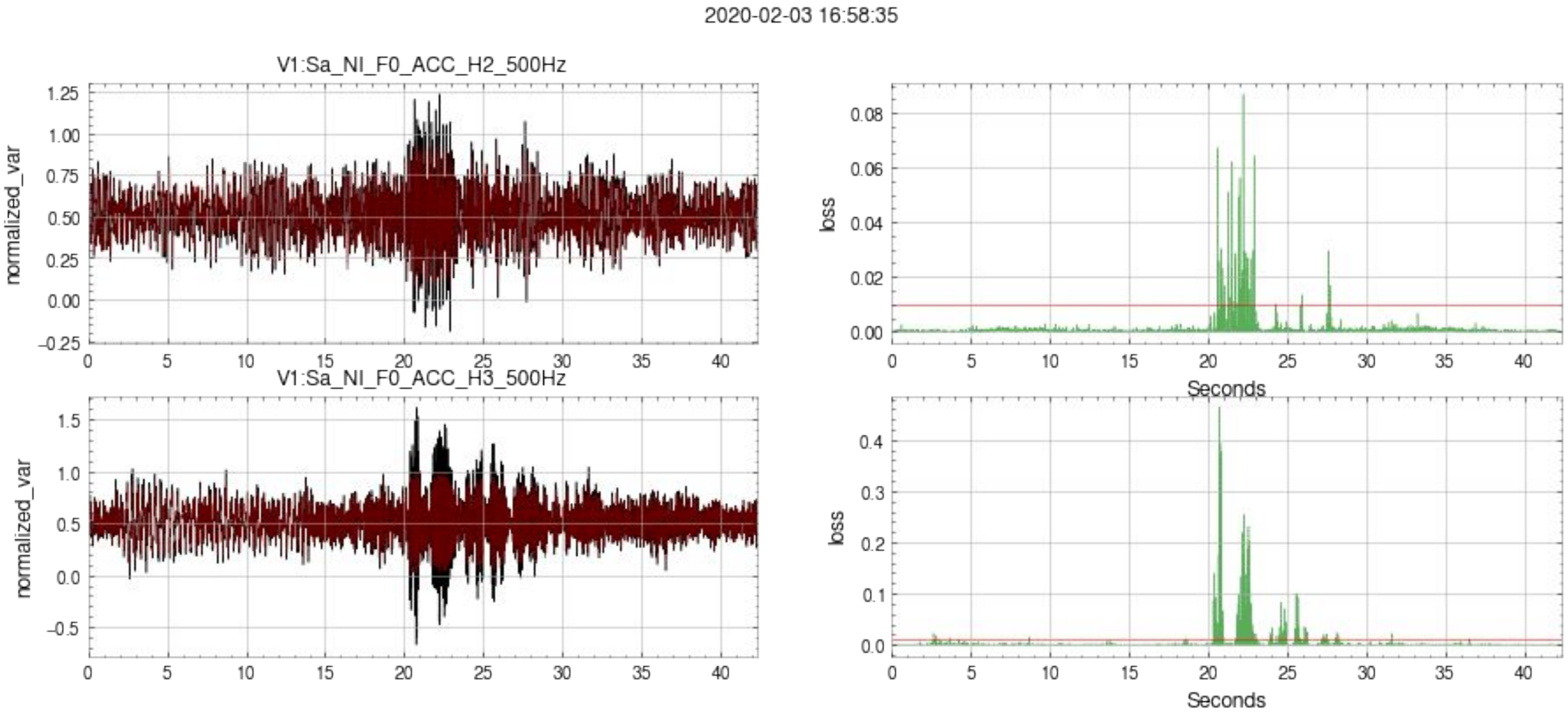
Thanks for your attention!

Backup: Veto channels?

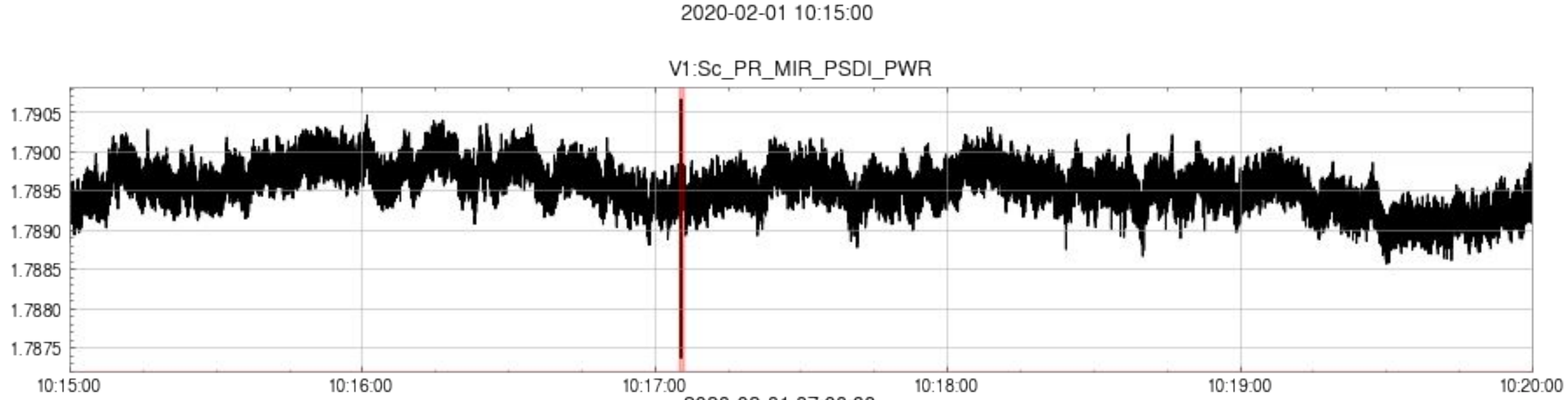
There were a few examples of glitch + anomaly correspondence

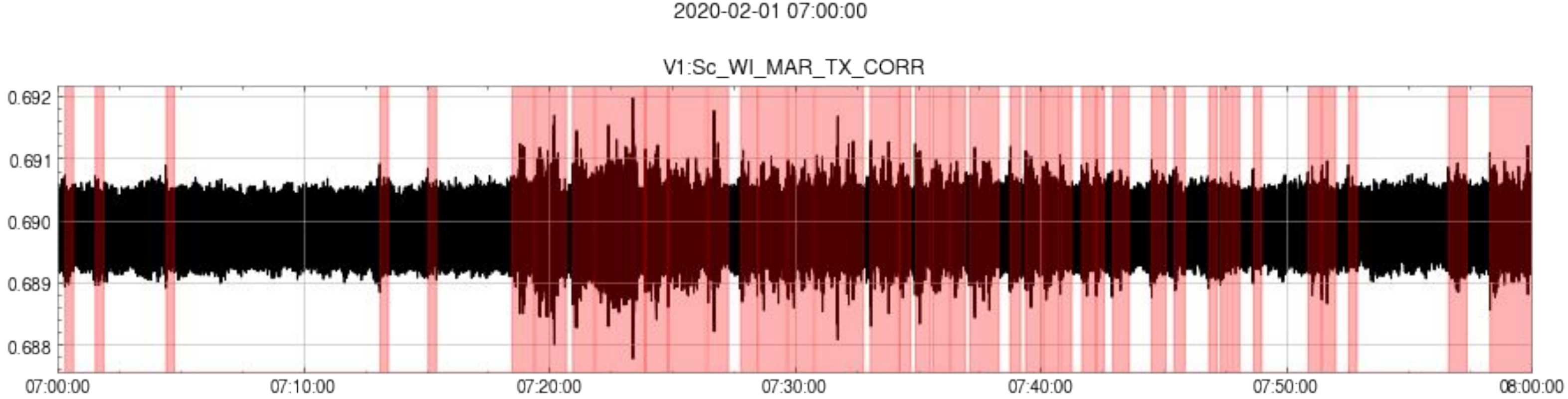


Backup: Bunch of anomalies



Backup: Bunch of anomalies





Backup: Inference run but longer