

# CIVIL-426

# Assignment 1

## Debrief

Chenghao XU  
Amaury WEI  
Arthur BIZZI

30.10.2025

# Overview of the Assignment 1

## Part 1 – Feature-based Anomaly Detection (65%)

Q1 [10%] Q2 [15%] Q3 [5%] Q4 [45%]

Q5 [10%] Q6 [10%] Q7 [5%]

**Average: 66.59%**

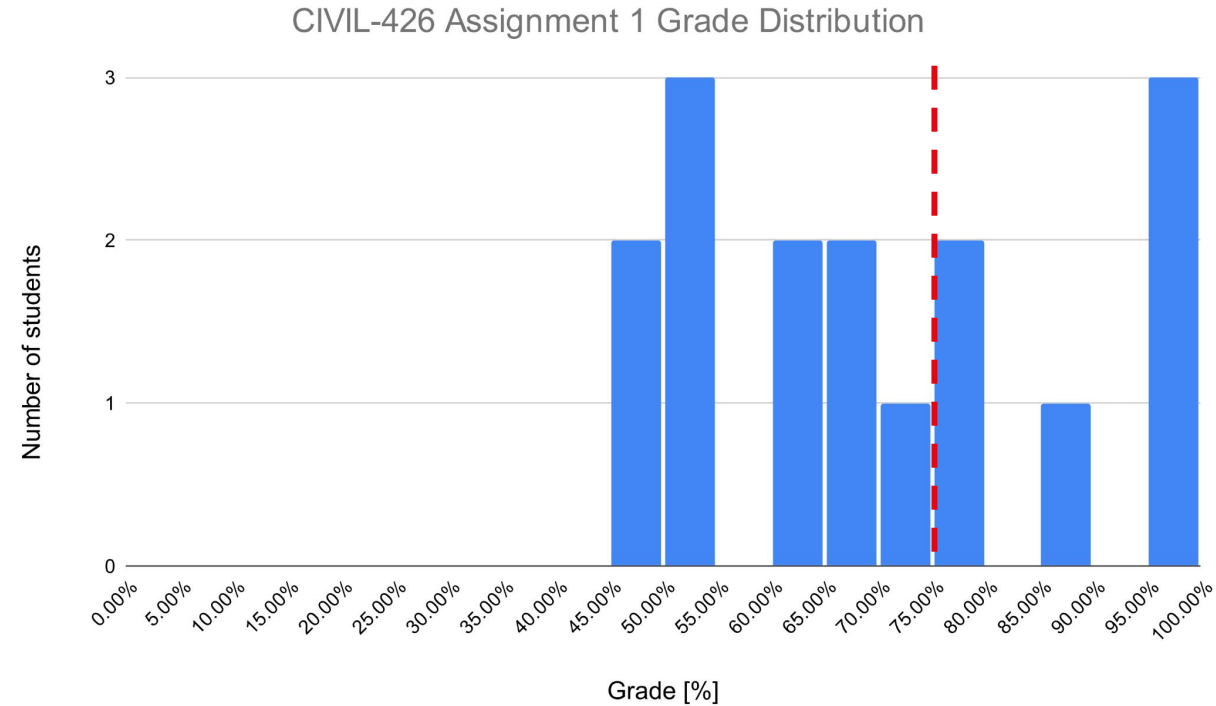
## Part 2 – One-class Anomaly Detection (35%)

Q1 [25%] Q2 [25%] Q3 [25%] Q4 [25%]

Bonus [10%]

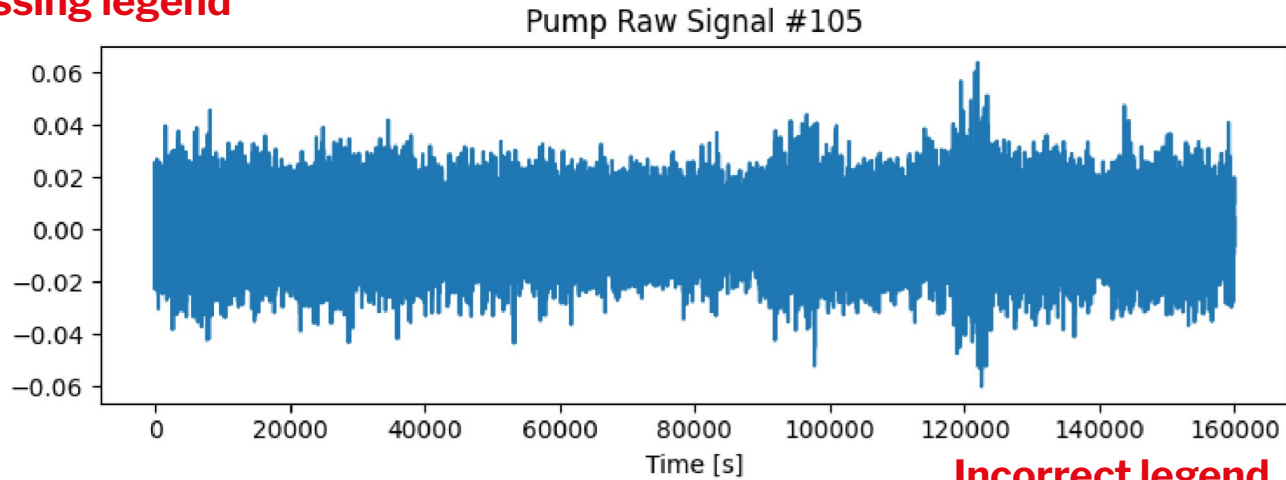
**Average: 77.91%**

**Assignment 1 Overall : 70.55%**



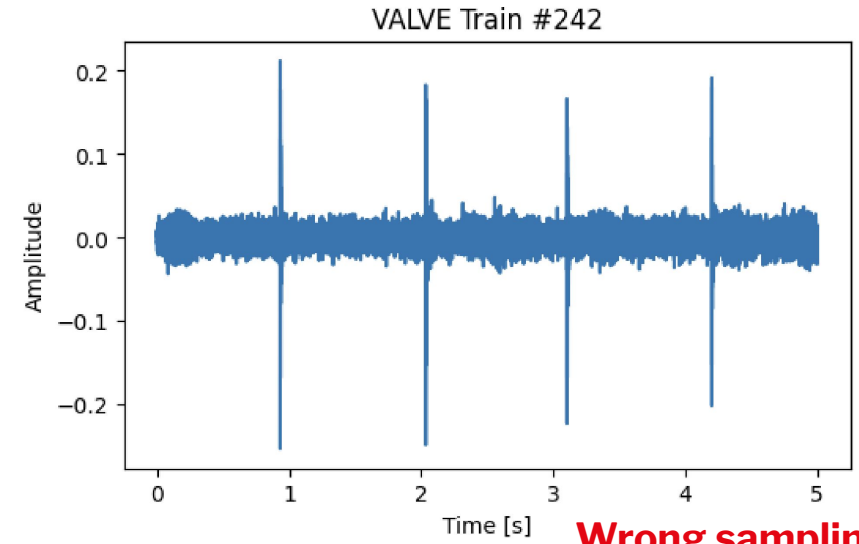
## ✘ Plot quality (Q1)

Missing legend



Incorrect legend

## ✘ Data loading (Q1)

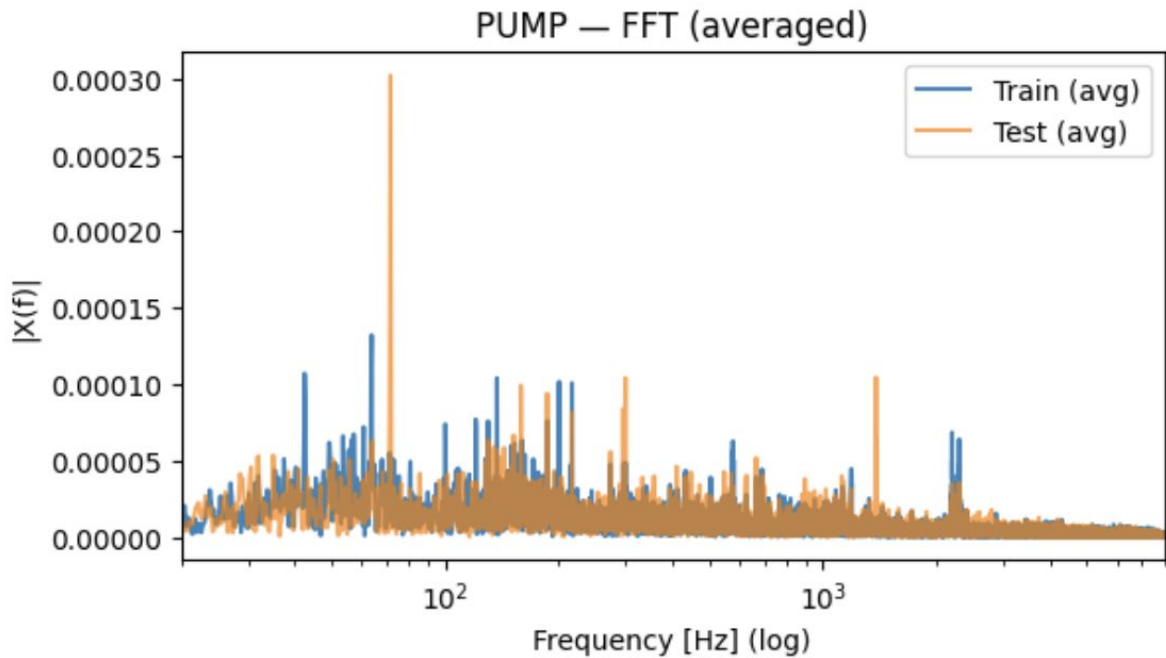


Wrong sampling frequency

## ✘ Not discussing the noise content (Q1)

## ✘ Analysis of differences between train/test sets (Q2)

Good initial ideas, but misleading analyses ...

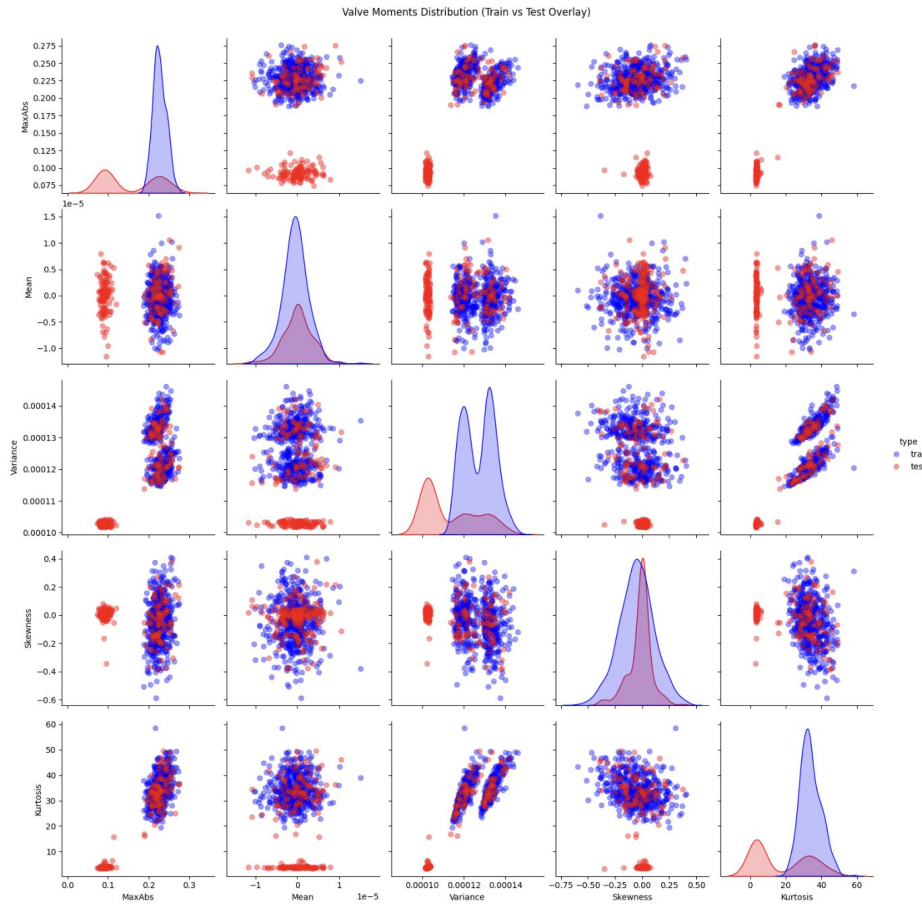


Average training set = 👍

Average a bimodal test set = 👎

=> instead, compare test samples against the training set average

**Computation and Feature Analysis (Q3)**



**Distinct colors**

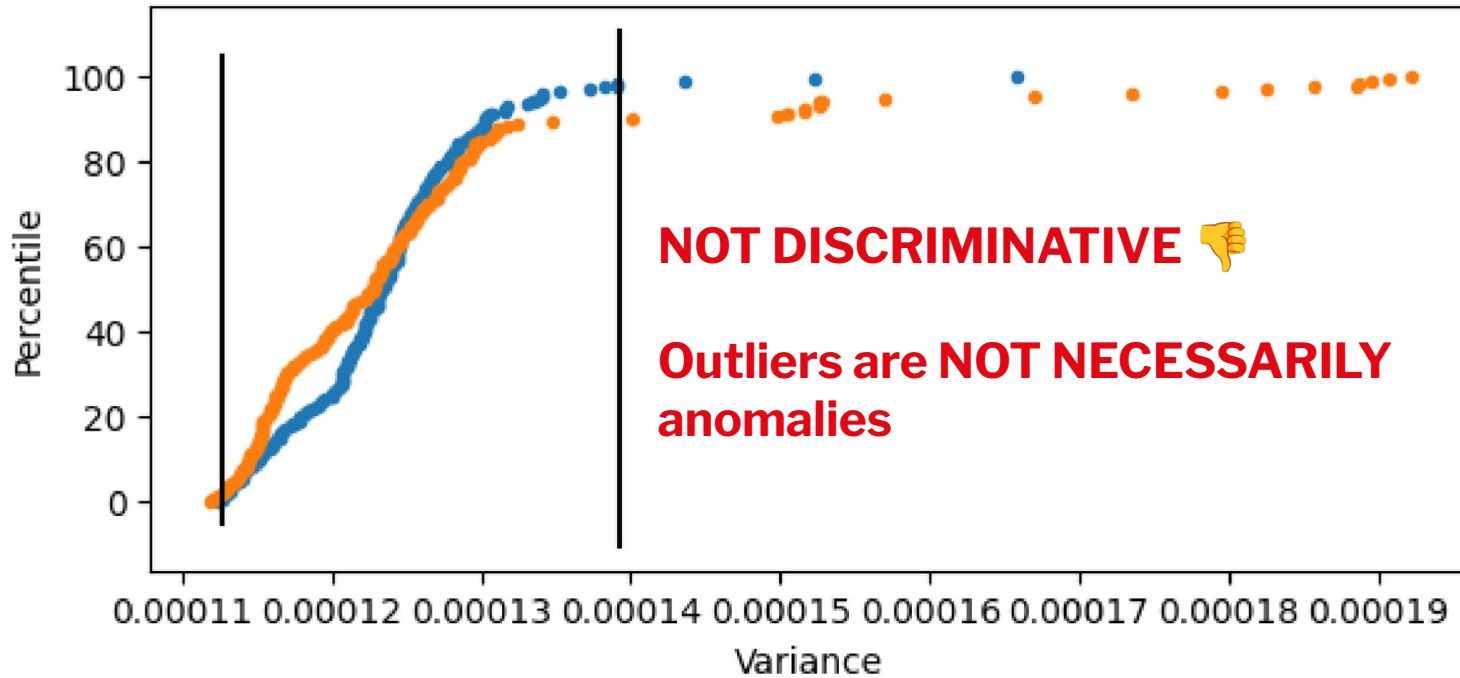
**Overlaying training and testing sets**

**Good discussions**



## ✘ Discriminative Features (Q4 - pump)

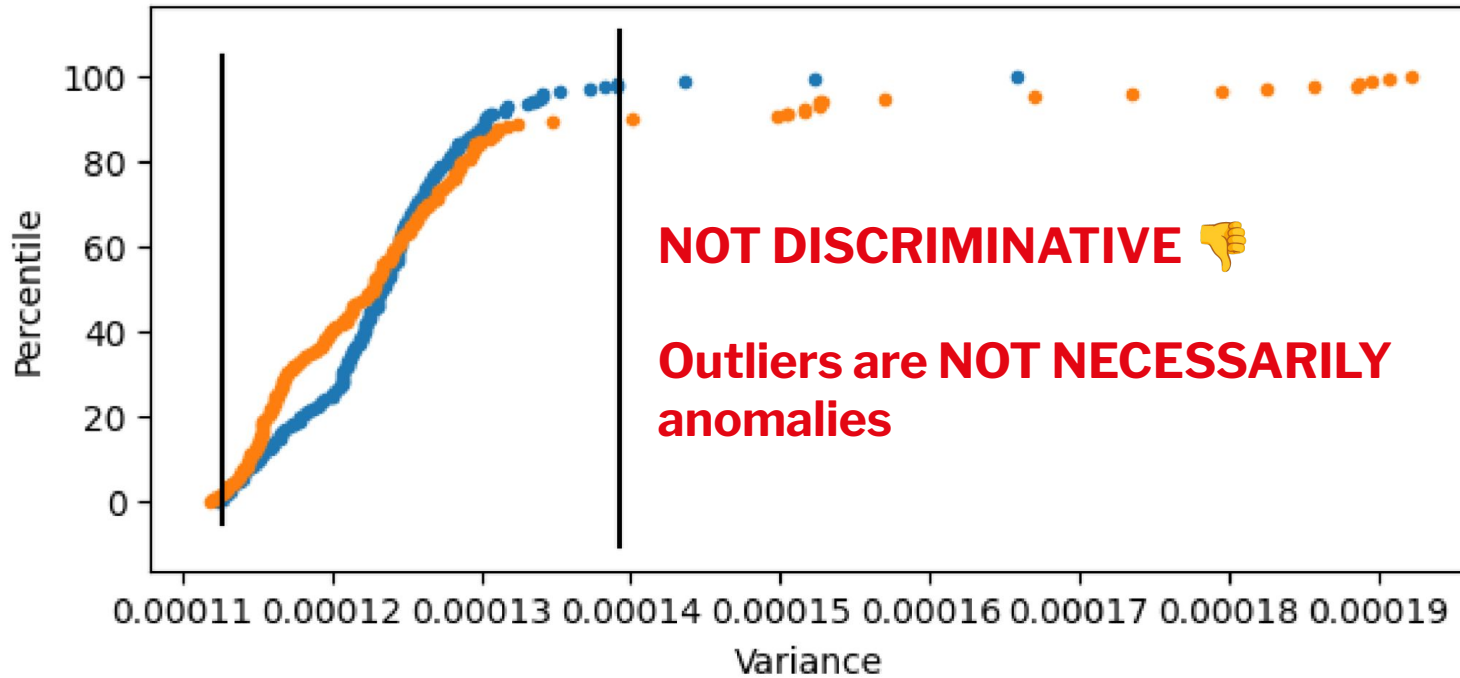
Too many groups simply stuck with the basic statistical features



## ✘ Lack of Venn diagram

## ✘ Discriminative Features (Q4 - pump)

Too many groups simply stucked with the basic statistical features



## ✘ Lack of Venn diagrams

All the code was already given in Part 0, just re-use it



# Feedback on Part 1

**Q5: There are many limitations, but two had to be mentioned at least implicitly:**

-  1. **Thresholds are finicky and tuned manually**
- 2. **Not robust with regards to data/domain shift**

**Q6: There are many ways to do this, but these should involve at least one of:**

-  1. **Retraining/Recalibration**
- 2. **Domain adaptation/feature normalization**

**Q7: Many options here; good job! These should involve:**



- 1. Unsupervised methods (Autoencoders, OCSVM...)**
- 2. Synthetic Fault Injection**

**Overall: Very good results! If you lost some points you likely forgot a detail, did not provide justification, or very obviously used an LLM.**



## **THRESHOLD SELECTION RATIONALE (FOR OVERLEAF)**

*Robust z-scores:*  $z = (f - \text{median})/\text{MAD}$ ; trigger on  $|z| > k$ .

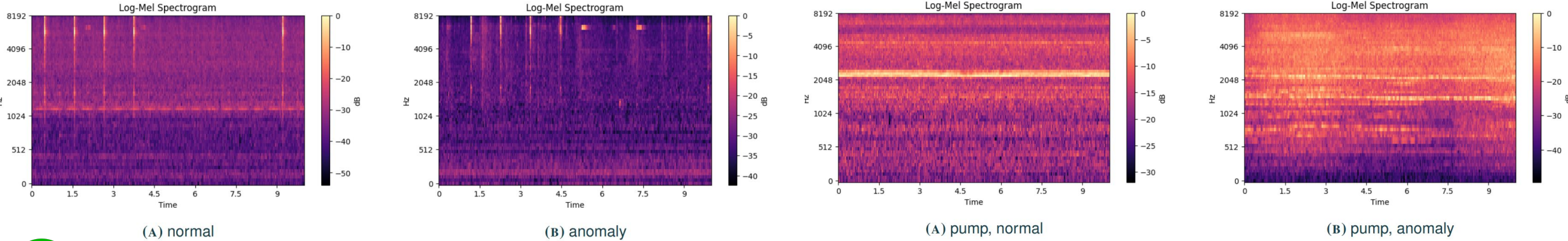


*Ensembles:* pump = OR with co-flag ( $\geq 2$  of {Var, Kurt, MaxAbs, (freq) Skew}); valve = majority/AND over {Kurt, Var, MaxAbs} (impulsive events), monitor (freq) Mean only after level normalization.

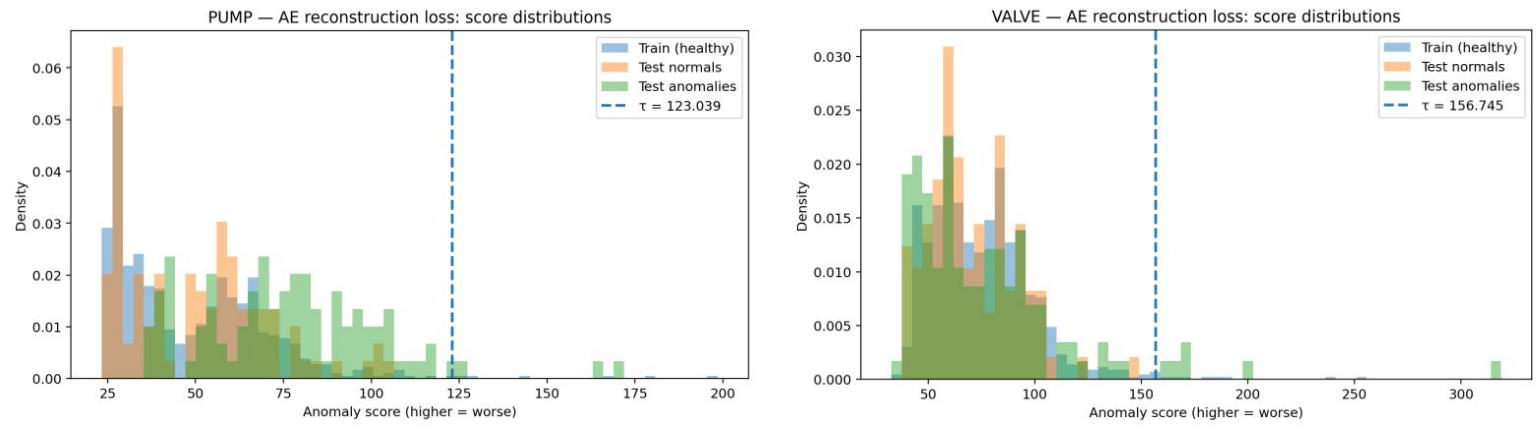
*Slow updates:* EWMA/rolling quantiles to track benign drift without chasing anomalies.

## ✓ Visualization and analysis on Mel-Spectrogram features (Q1)

It is important that you can distinguish the anomalies **first**.



## ✓ Why AutoEncoder can be used for anomaly detection? (Q1)



What is the assumption here?

✗ “Higher AUC scores means that the AutoEncoder reconstructs abnormal data fairly well”

## ✔ How to write a good analysis? (Q1)

**Data/Feature:** How does the data look like? How do the features look like?

**Model:** How do you design your model? What model did you use?

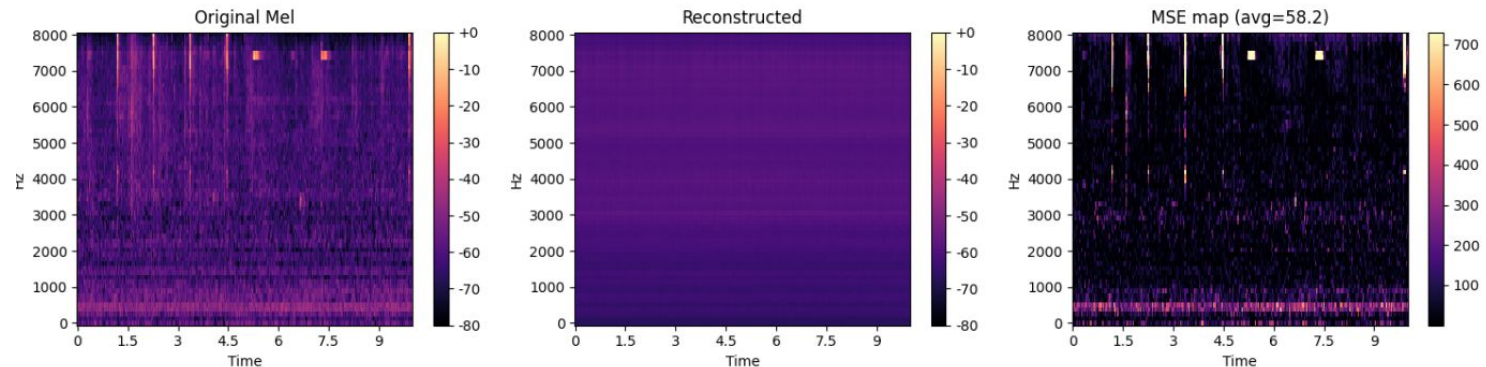
**Performance:** How does this model perform? Prove it using your metrics.

**Reasoning:** Why does this model work/fail?

*“It is noteworthy that the held-out ROC-AUC (0.805) exceeds the validation AUC (0.755). This is consistent with finite-sample effects: the larger evaluation set (n = 160) yields a smoother and often slightly higher AUC estimate, while the smaller validation set produces a noisier ROC.”*

*From a signal perspective, these outcomes align with the error distributions induced by the autoencoder. The model is trained on “stationary, narrow-band + low floor” log-Mel patches typical of a healthy pump. Anomalous recordings exhibit broader bands, intermittent bursts, and elevated broadband energy; these deviations are not reconstructed well, so their residuals shift upward. This separation is exactly what the ROC captures.”*

**Autoencoder is not directly trained for classification.**



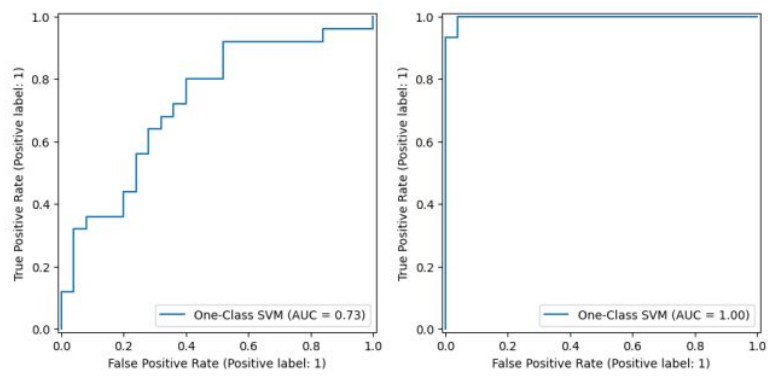
✔ Compare with Residual-based method or different systems (Q2)

✘ Only provide plots without any text

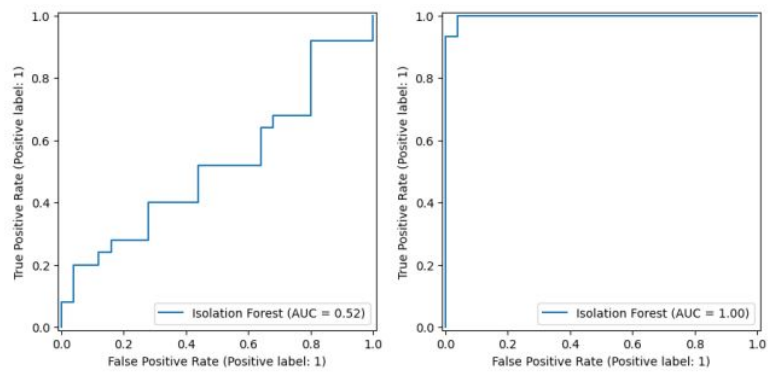
✘ Insufficient analysis

QUESTION 2

From the trained AutoEncoder, use the bottleneck features to train both (1) a One-Class SVM and (2) an Isolation Forest, and report the corresponding AUC scores.



AUC of 1-class SVM training






AUC of Isolation forest training

*“The results for the pump dataset are less satisfactory. As the AutoEncoder features reconstruct healthy and abnormal spectrograms with a similar level of precision, both machine learning algorithms struggle to differentiate between them. The ROC for one-class SVM, displayed, has an AUC score of 0.49, meaning it is no better than random choice. Isolation forest performs slightly better with AUC score of 0.62. The somewhat linear shape of the curve indicates that the ratio between false and true positives remains constant....”*

✘ Unnecessary text

*“The autoencoder training function needed to be completed. To accomplish this, the back-propagation of the results of each optimization loop needed to be added. This is essential as the back-propagation used the gradient descent to compute the optimal weights and biases of each neuron. **Finally, the autoencoder was trained on the training data set and was then tested on the testing data set...**”*

- 
**Well done on comparing PCA with bottleneck features (Q3)**
- 
**Missing final conclusion for each system (Q4)**
- 
**Fail to match your result with your experiment (Q4)**

<b>Machine</b>	Accuracy	TPR	FPR	F1-score
Pump	0.865	0.850	0.120	0.863
Valve	0.626	0.462	0.180	0.573

**Table 2.1**

Evaluation metrics of the anomaly detection models for different machines



1. Describe how the faults are combined and for what reason.
2. Tables and/or graphs to illustrate!
3. Results were highly dependent on the approach
  - a. If your model was already good, not much difference
  - b. A small improvement was possible for some implementations



4. Extreme differences were either due to an error or... suspicious.....
5. Code should back it up



# Questions