Weekly Report

DQM-DC
Patomporn (Jab)
28 June 2019

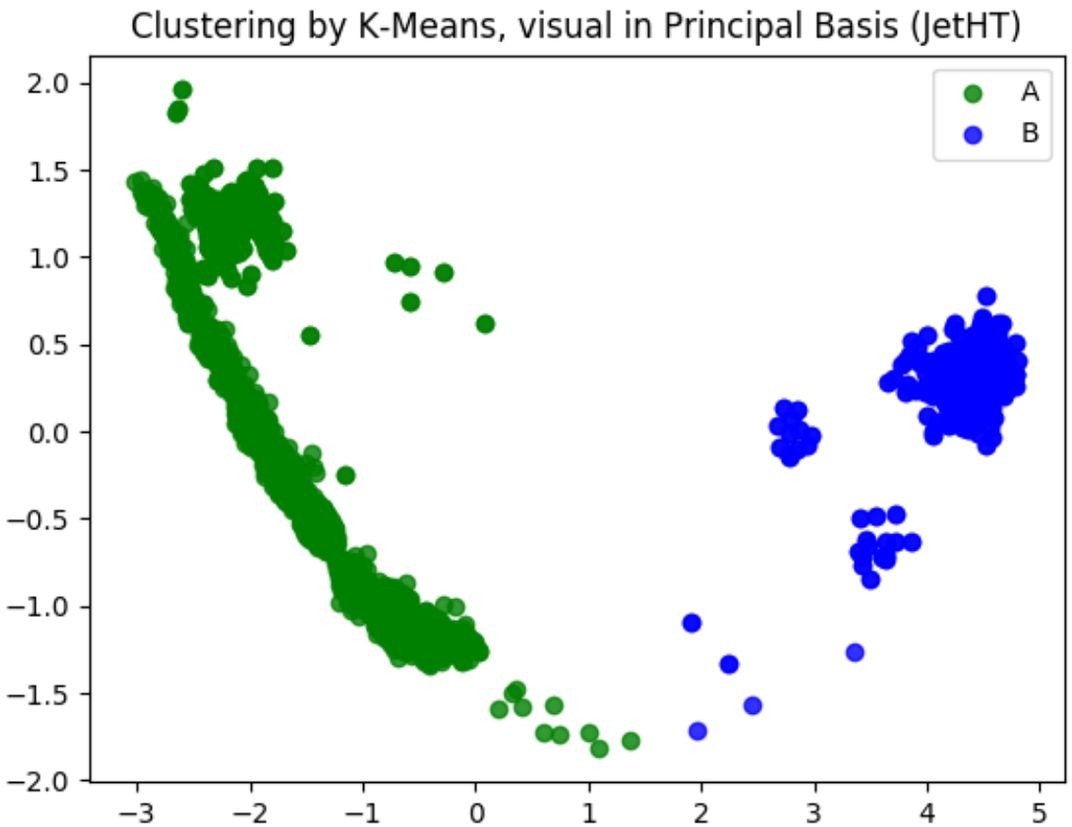
Outline

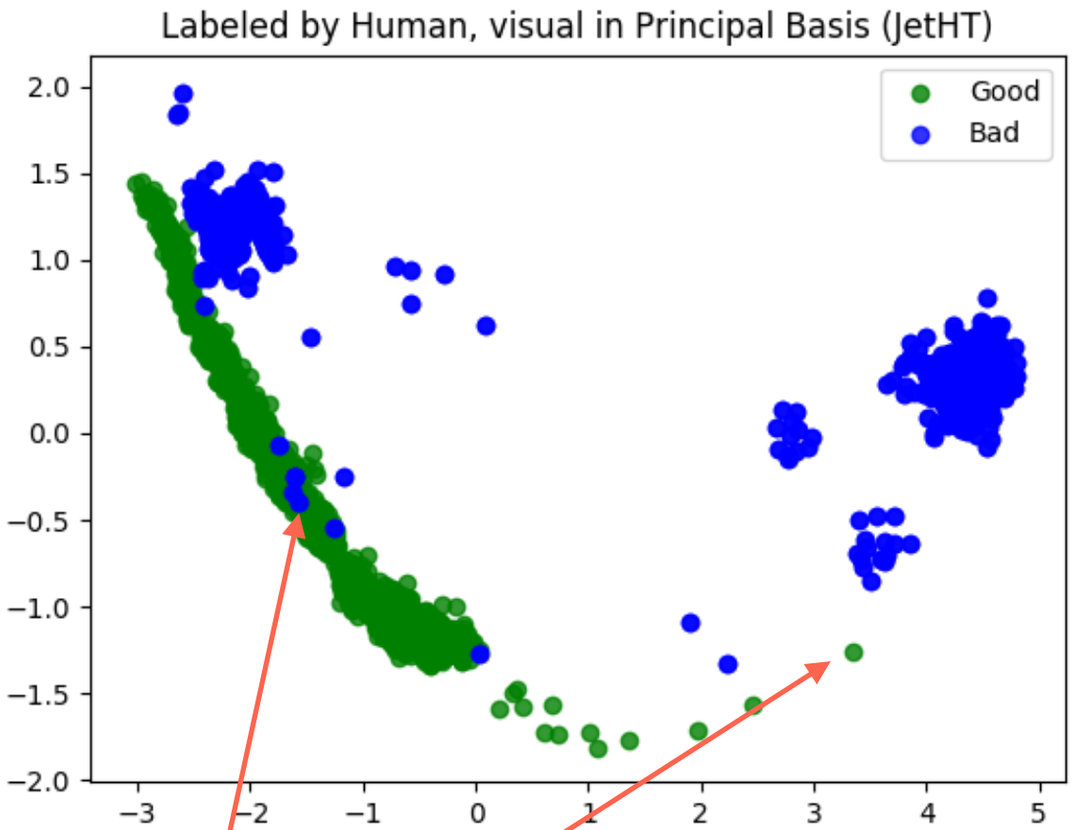
- Autoencoder
    - Clustering in reduced data features
    - Performance
        - Sampling from testing datasets
- ML (no neural-net)
    - Isolation Forest
    - Schölkopf’s One-Class SVM
- Let’s find the cutoff

Take a look again

MinMaxScalar

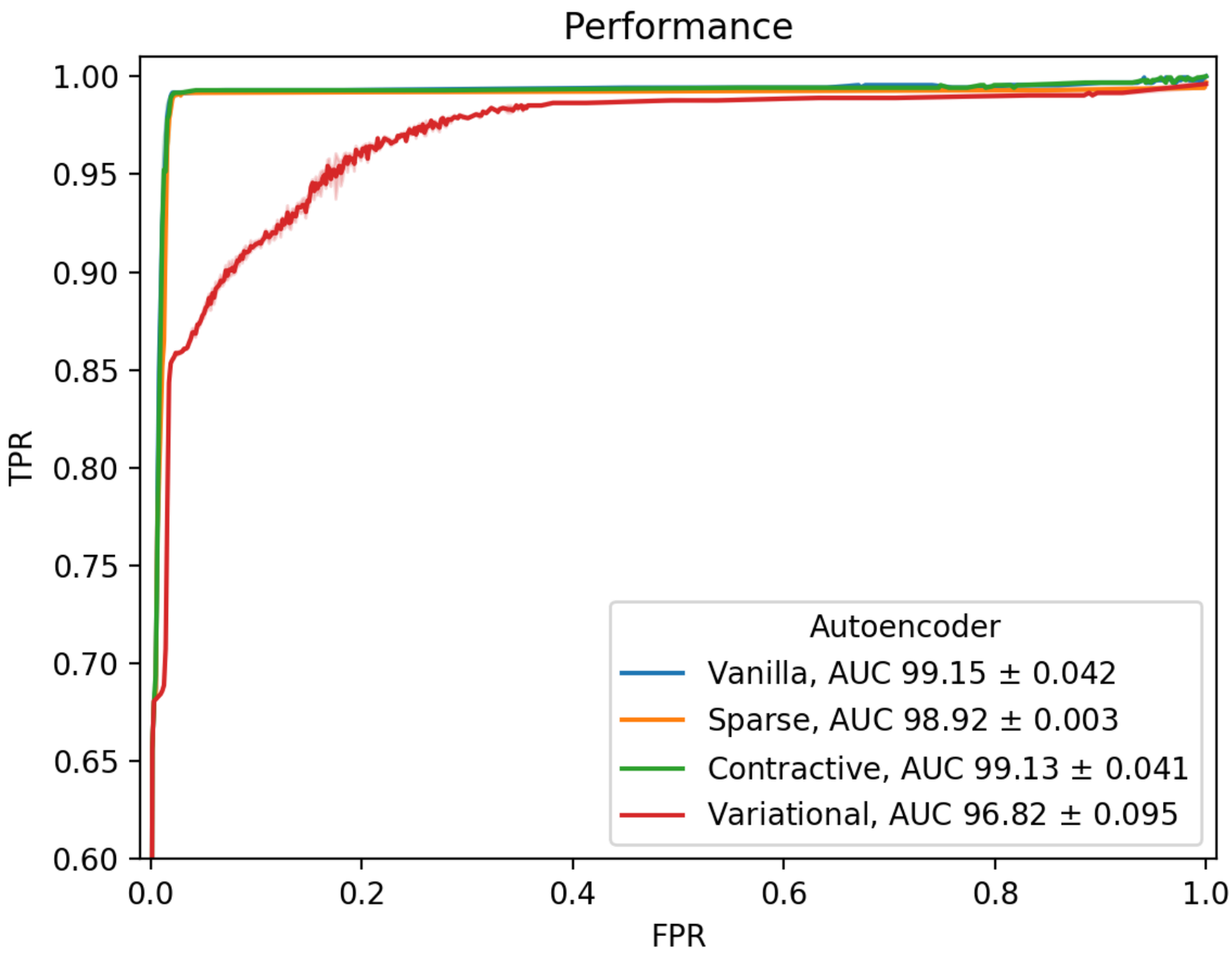
“ If the scattering looks more separately, It might possible to simply apply cutoff by SVM..”



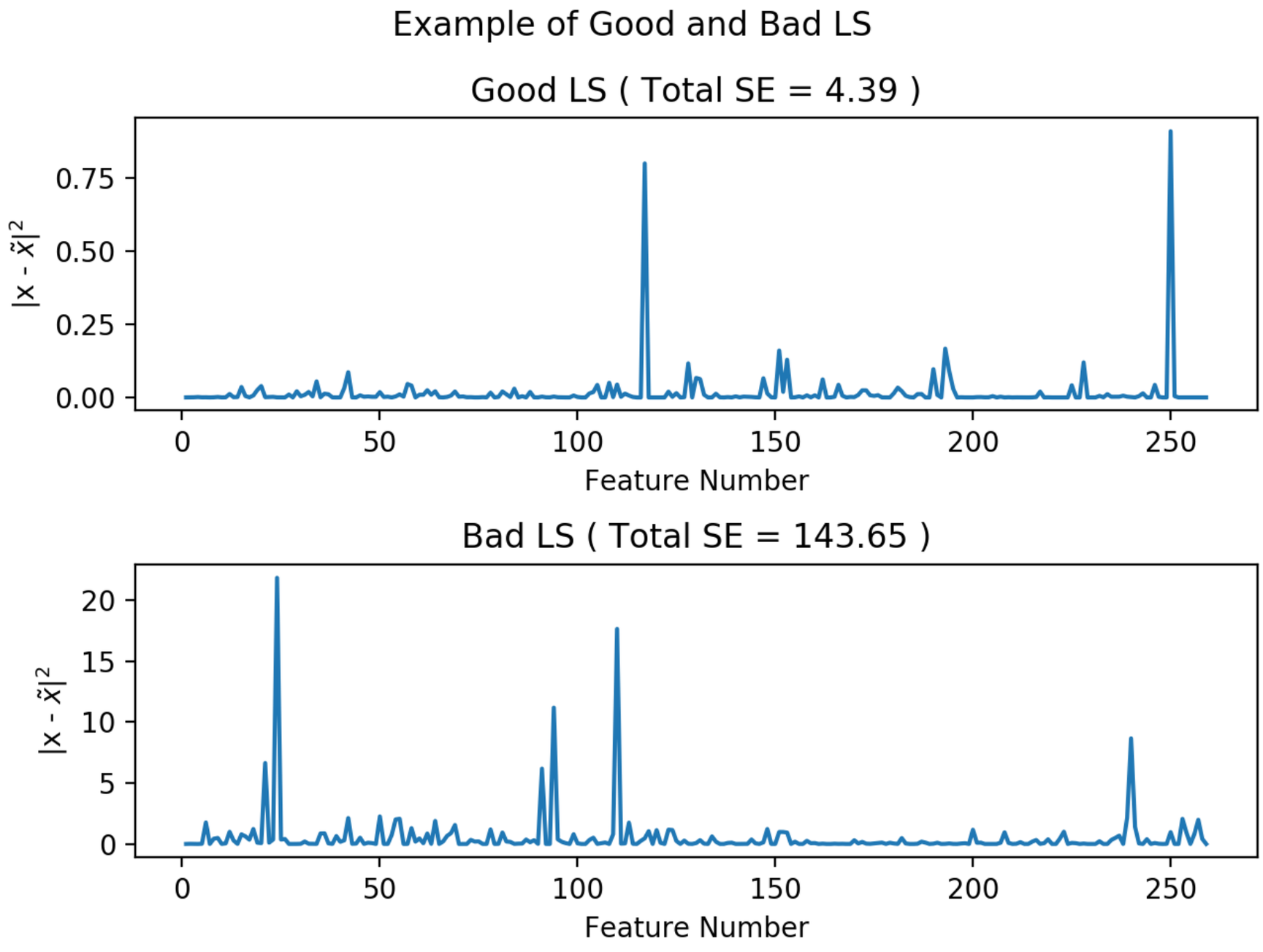


Contamination from wrong label ?

Results


Performance
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
TPR
0.0
0.2
0.4
0.6
0.8
1.0
FPR
Autoencoder
Vanilla, AUC 99.15 ± 0.042
Sparse, AUC 98.92 ± 0.003
Contractive, AUC 99.13 ± 0.041
Variational, AUC 96.82 ± 0.095

Example from Vanilla



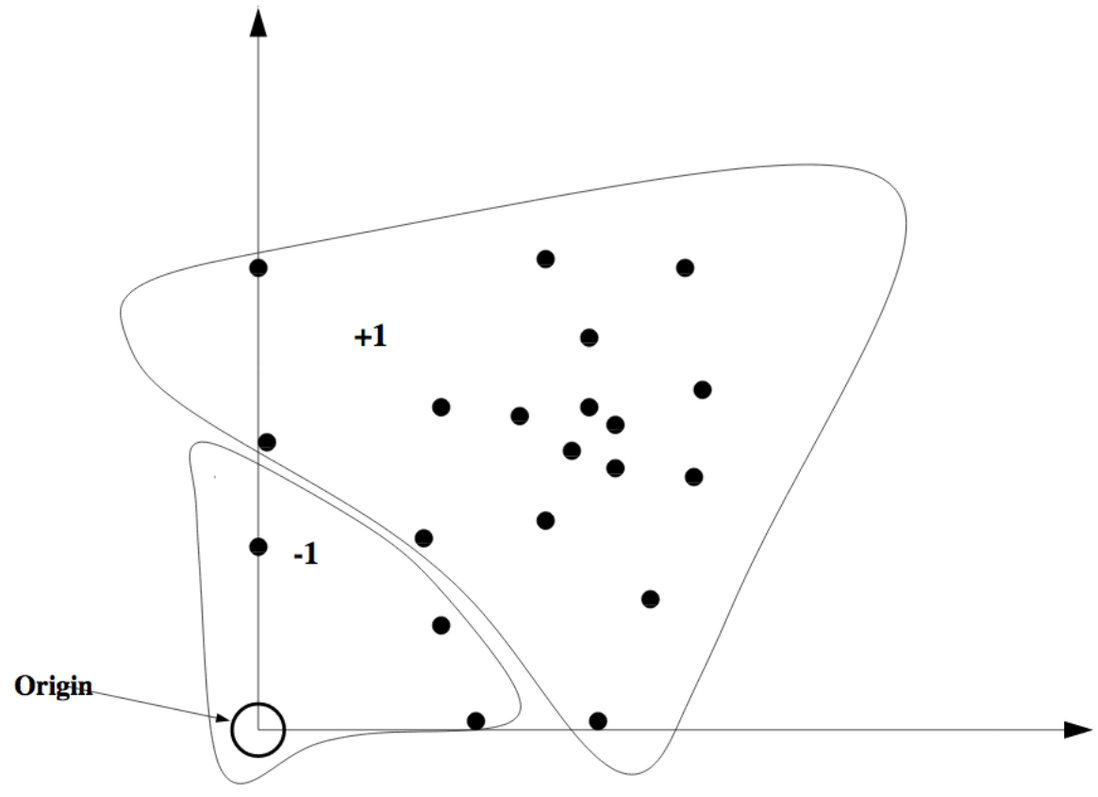
Isolation Forest

- Ensemble Forest from tree by subsampling (Ψ)
  - Iteratively picking up features and random value to contract the node (equivalent to step fn.)
- Anomaly score likely to be average depth of the instance over forest

$$s(x, \Psi) \equiv e^{-<h(x)>/c(\Psi)}$$

- Where
  - $h(x)$ is the depth in tree h
  - $c(\Psi)$ normalization factor growing as log2(Ψ) from branching

Schölkopf's One-Class SVM



- Minimize (Soft margin)

$$\frac{||w||^2}{2} + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i - \rho$$

- Under

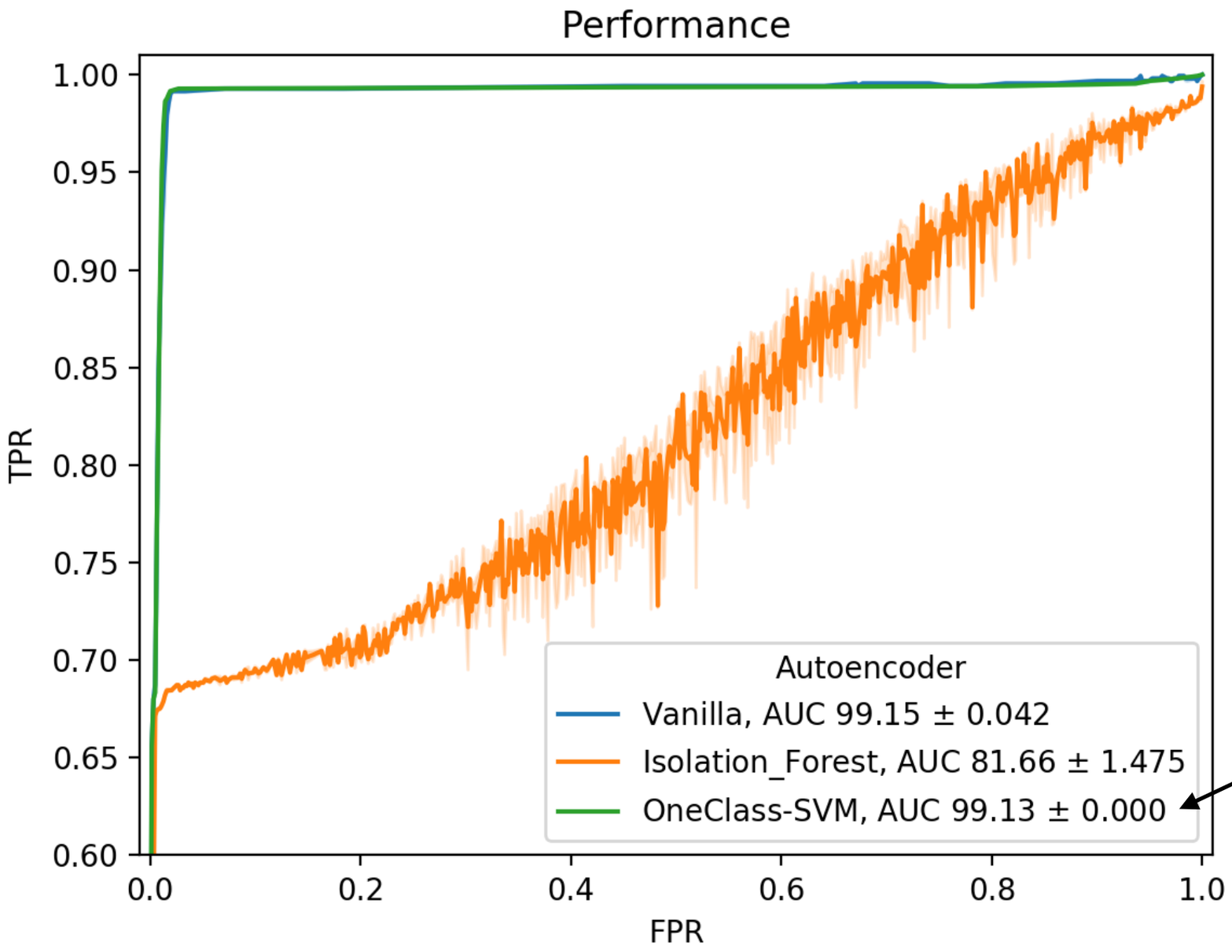
$$w \cdot \Phi(x_i) \geqslant \rho - \xi_i; \xi_i \geqslant 0$$

- **Kernel:** Gaussian Base Radial function (GBF)

- Determine by tangent distant from data point to hyperplane

[1] http://www.jmlr.org/papers/volume2/manevitz01a/manevitz01a.pdf
[2] http://papers.nips.cc/paper/1723-support-vector-method-for-novelty-detection.pdf

Results



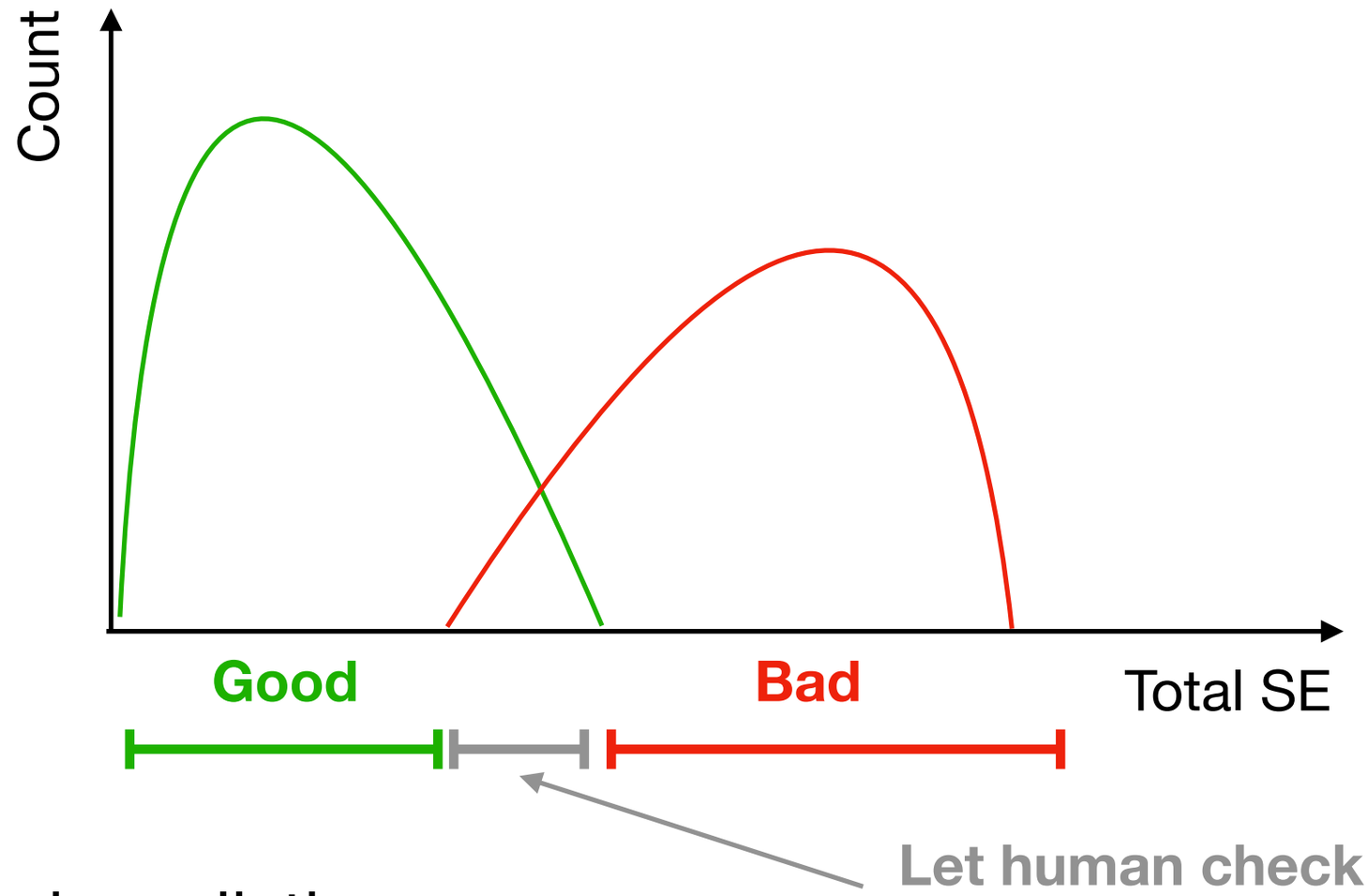
This guy
has no randomness
Then we mightn't surprise

Under configuration

- Isolation Forest: tree = 200, sampling_size = 512
- OneClass-SVM: nu=0.1, gamma=0.1(inverse gaussian width)

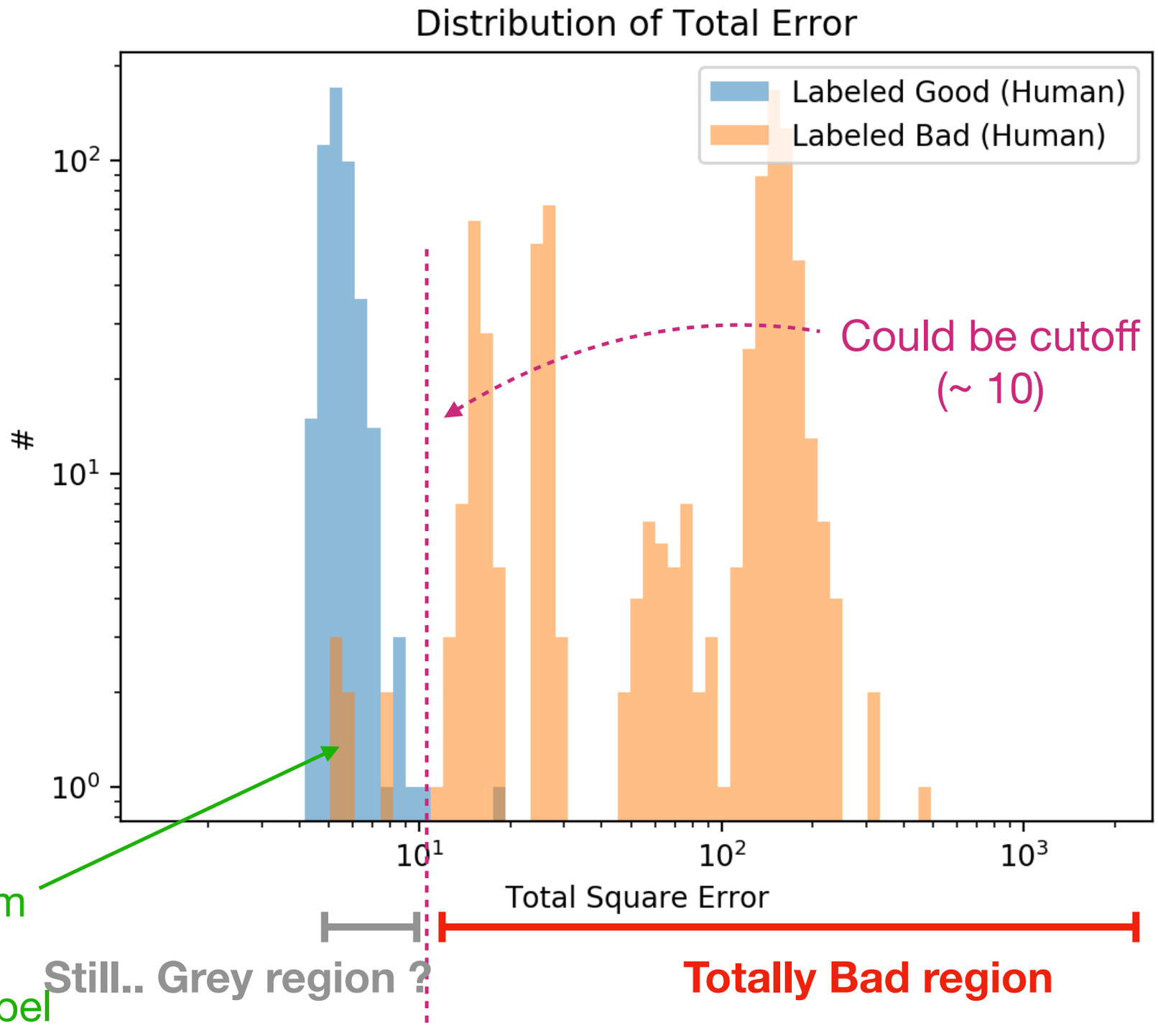
Find the cutoff

- **Expect** the total square error (SE) distribution to be like



- Next slide is realistic..

SE Distribution from Vanilla



Could be Either Mistake from model Or Human label (When mark run#)

Then where exactly could be green zone ?

Grey could be good if contamination ~ 1.5% is acceptable

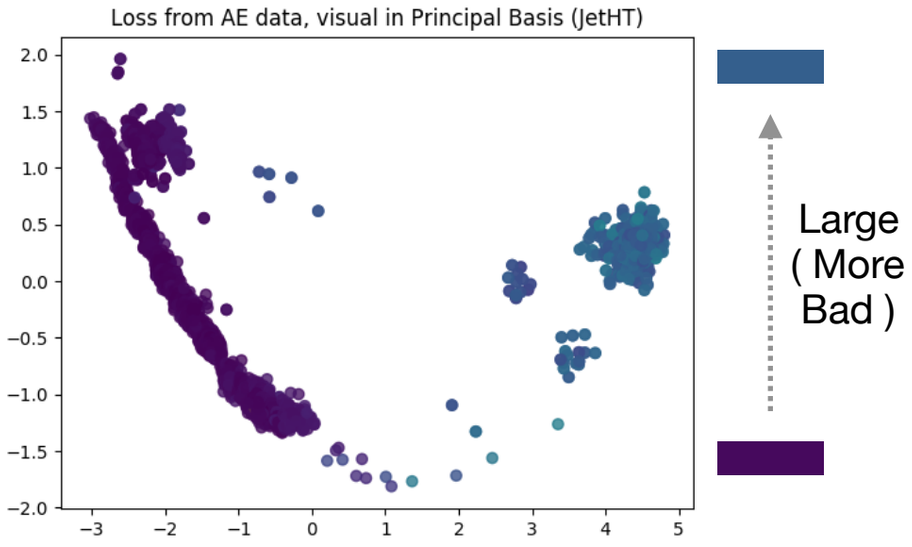
Table of unexpected LS
(Bad human label falling into good)

#RUN	#LS	Investigation
281689	3	**Human correct (ES-DAQ, HLT BAD)**
282037	458	**Human correct (HCAL)**
282923	18	**Human correct (HCAL)**
282923	31	**Human correct (HCAL)**
282923	87	**Human correct (HCAL)**
282924	1	**Human correct (HCAL FED)**
283358	244	**Human correct (HCAL)**
283416	48	**Human correct (ECAL)**

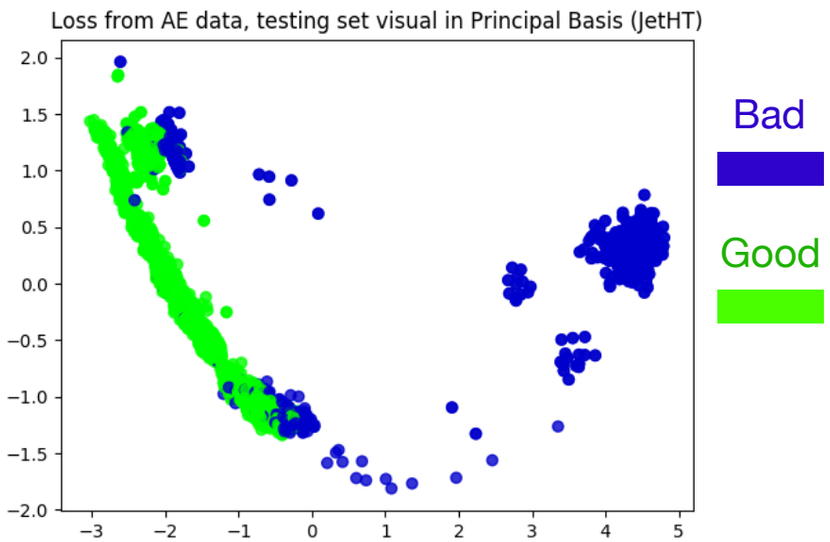
Extended Investigation

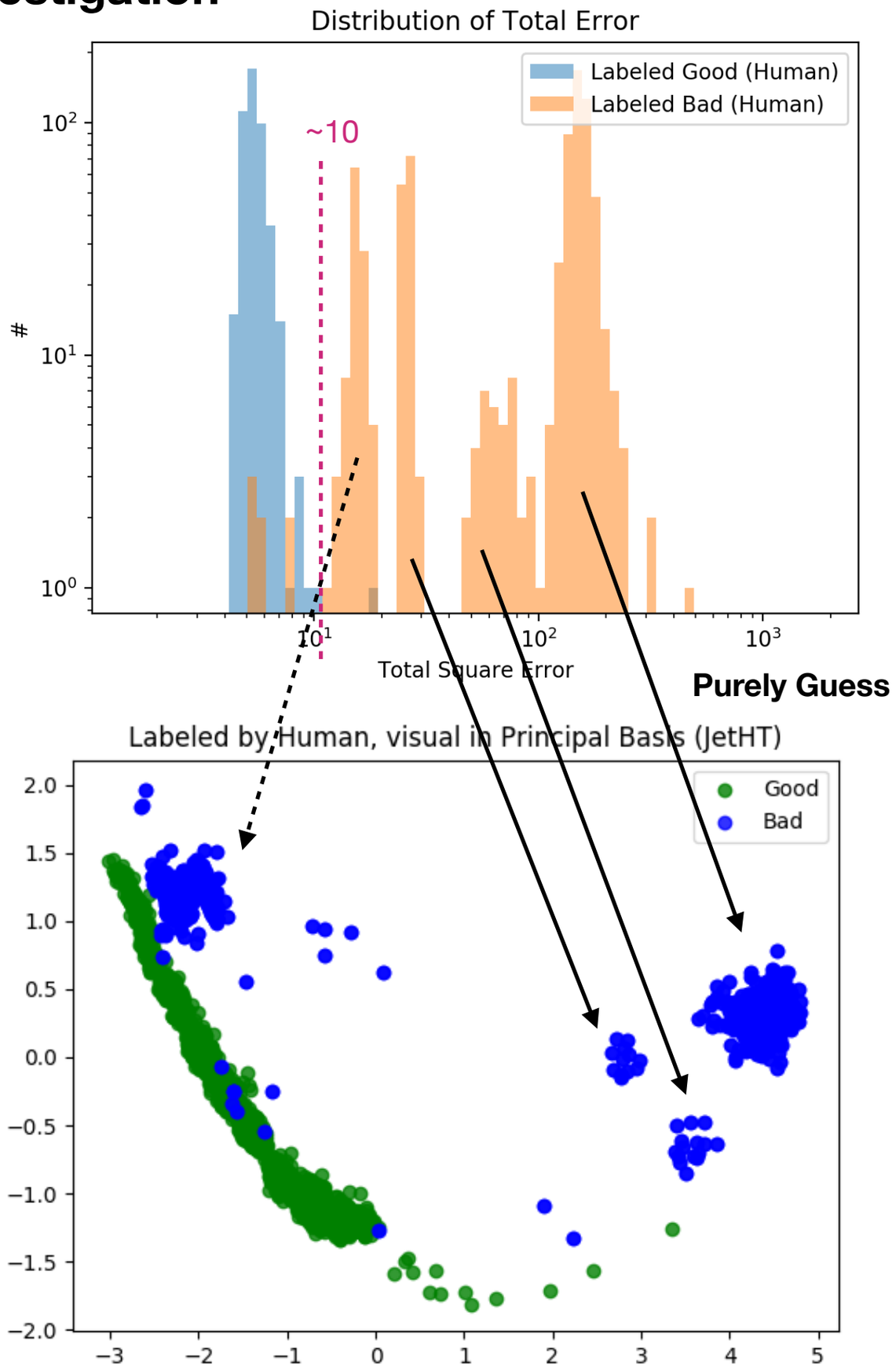
Loss value from Model

Loss as color shading

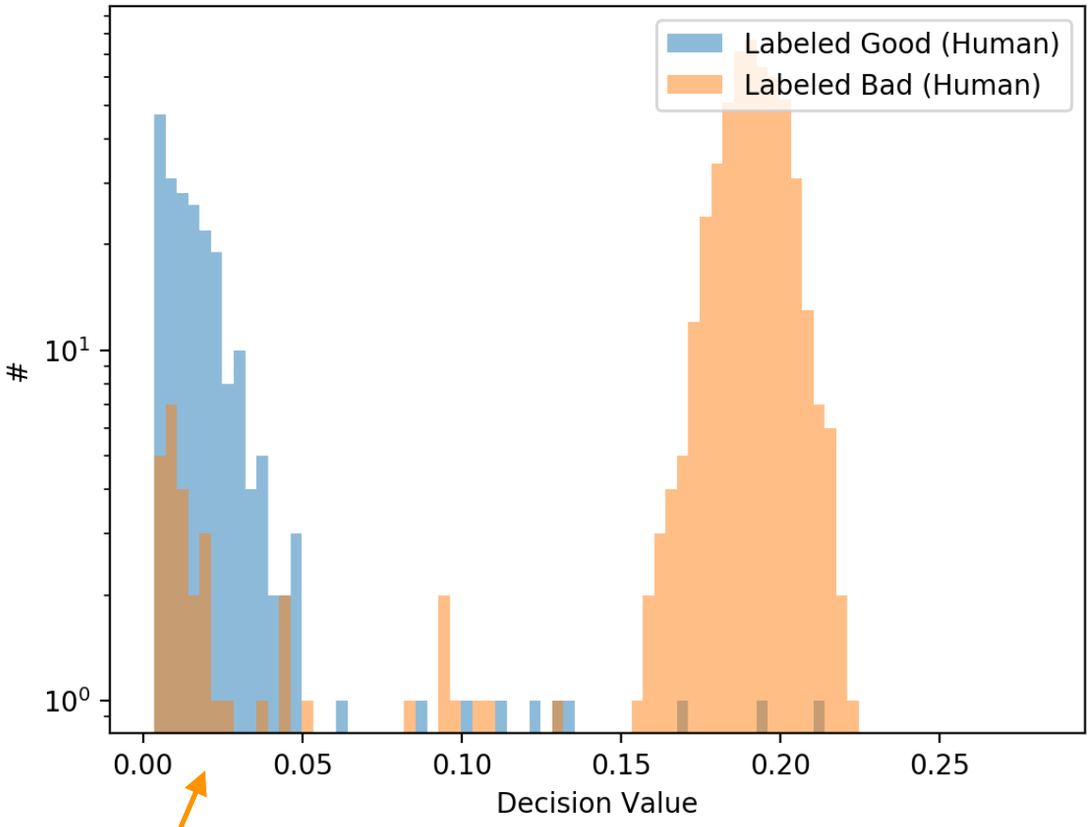


Applying cutoff (MSE > 10.0 is bad)



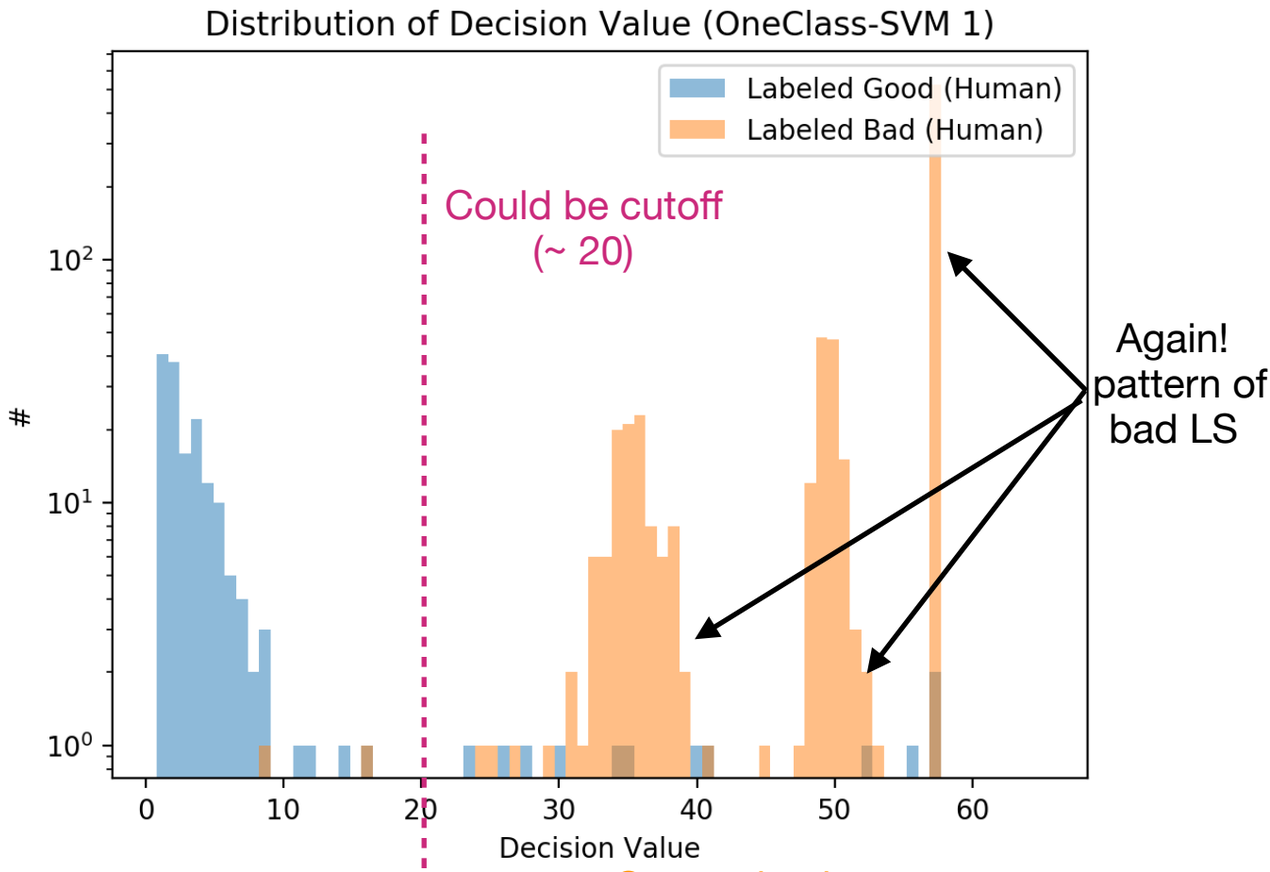


Decision Value from ML



Lack of efficiency to distinguish

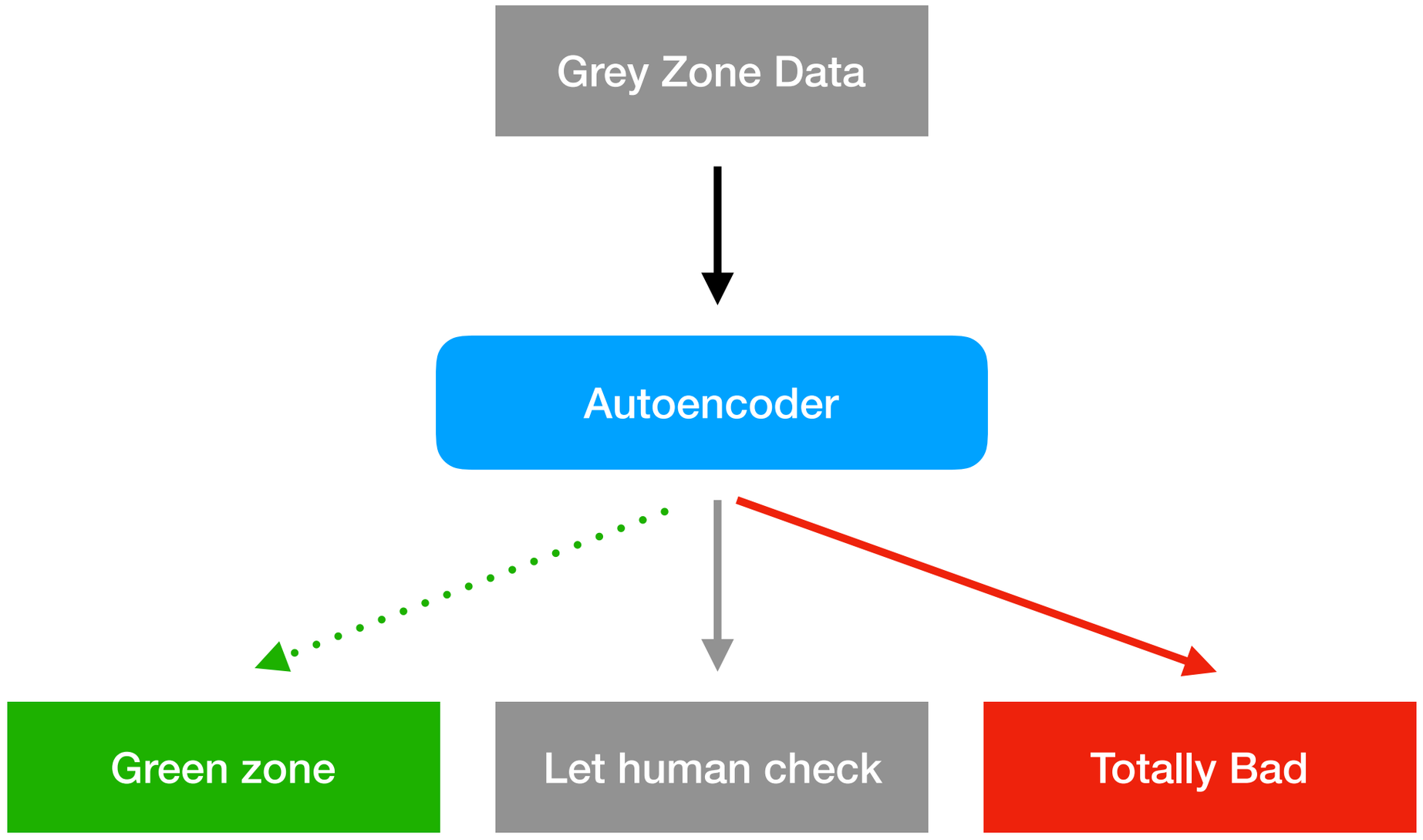
Even more distinguishable than Vanilla



The contamination LS will be investigate next week

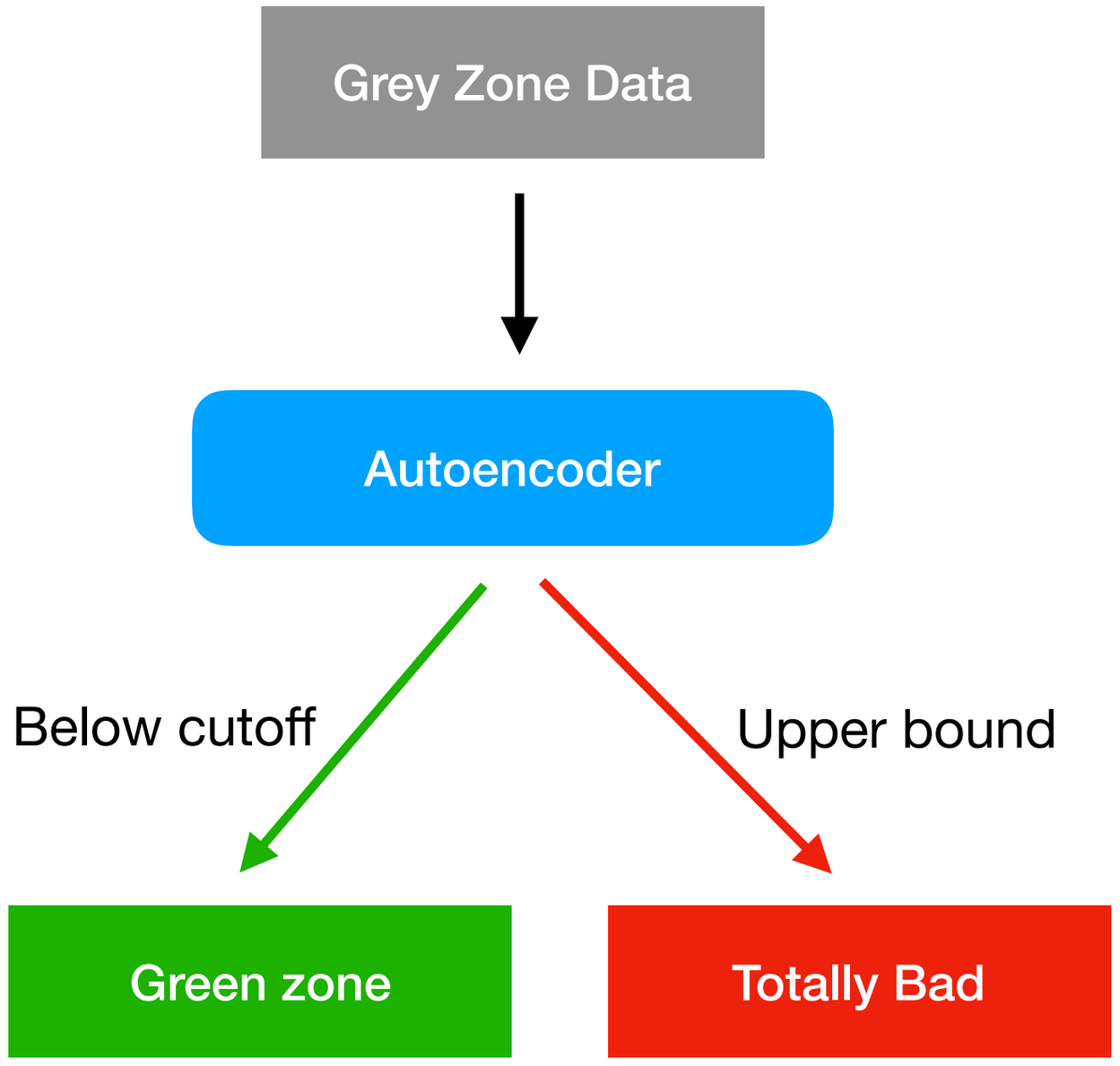
We suppose to have 2 options

Option 1



“Current result seems not promising to pick this option”
(Grey zone in the distribution is not obvious at all)

Option 2



“ If ~1.5% of contamination is acceptable ”

Future work

- Investigate contamination LS in OneClass-SVM
- Waiting for datasets 2018 and reprocess 4 channels

- Random Idea: If we could provided labeled LS feedback on the fly
  - **Reinforcement Learning**
  - **Pros** Model itself also growing up and could facing with a new configuration
  - **Cons** require feedback architecture which definitely increase our work