ANOMALY DETECTION
-- ACTIVE LEARNING PROJECT

Project Director : Yuri Balasanov

Project Author : Harper Xiang

2020.09

PROJECT OVERVIEW

This project is to detect Anomalies from un-labeled data using Active Learning techniques. Detecting Anomalies is binary classification, but as the target variable is assumed unknown, the detecting process has to be Unsupervised Learning. The goal to be achieved is labeling as many anomalies with as less human labeling effort.

- WHAT IS GIVEN
  - Limited Budget of Human Labeling
  - Super Imbalanced Data (Anomalies= 0.17%)
- WHAT TO ACHIEVE
  - Labeling Most Anomalies (80%)
  - Saving Budget

SOLUTION OVERVIEW

The solution is constructed upon three pillars

- **Unsupervised Learning**

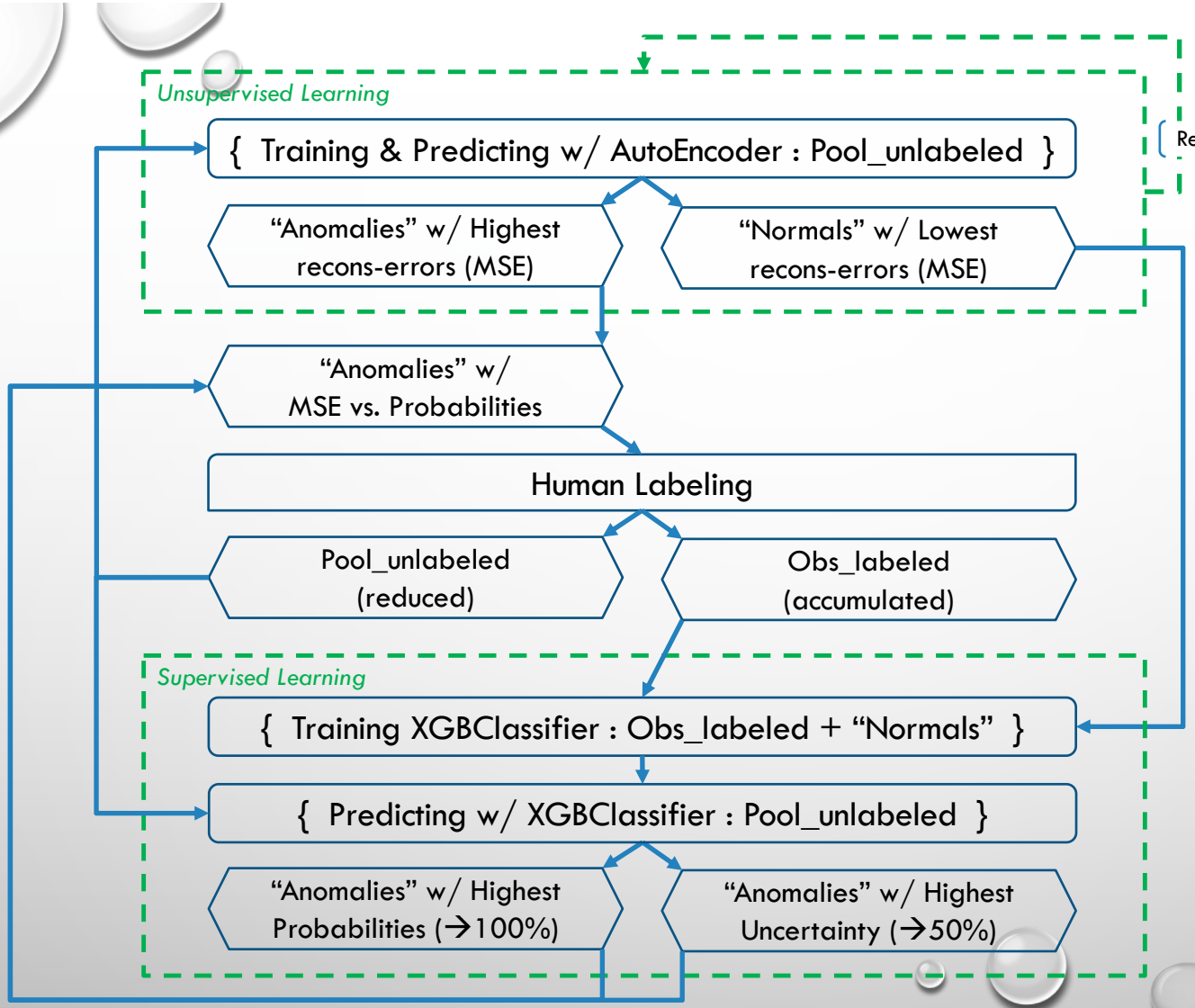
  With the large amount of un-labeled data, Neural Network is available for Unsupervised Learning. Autoencoder models are adopted to detect anomalies by reconstructing X and checking the reconstruction errors.

- **Supervised Learning**

  As human labeling involved, Supervised Learning is available. But with the tiny volume of labeled data, only GLM or Tree models are possible. XGBClassifier and Logistic models are both used for specific purposes in this project.

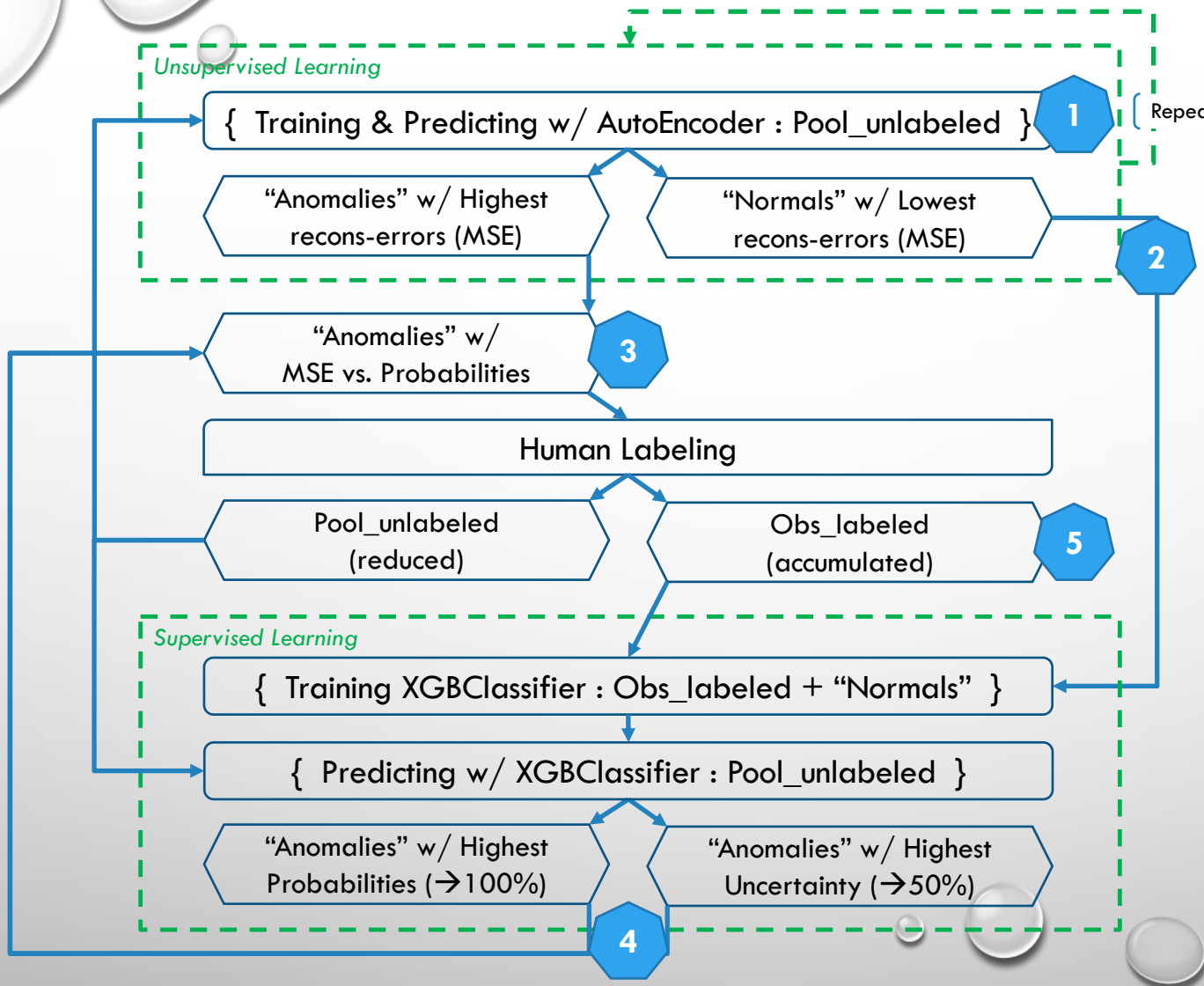
- **Integrating the two parts**

  This is the pivot of the solution. Neither Unsupervised nor Supervised Learning can solely resolve the challenges. Appropriately combining the two techniques may successfully implement Active Learning to complete the task.



Universal Solution Structure

The whole structure iterates over and over until a stop point is reached



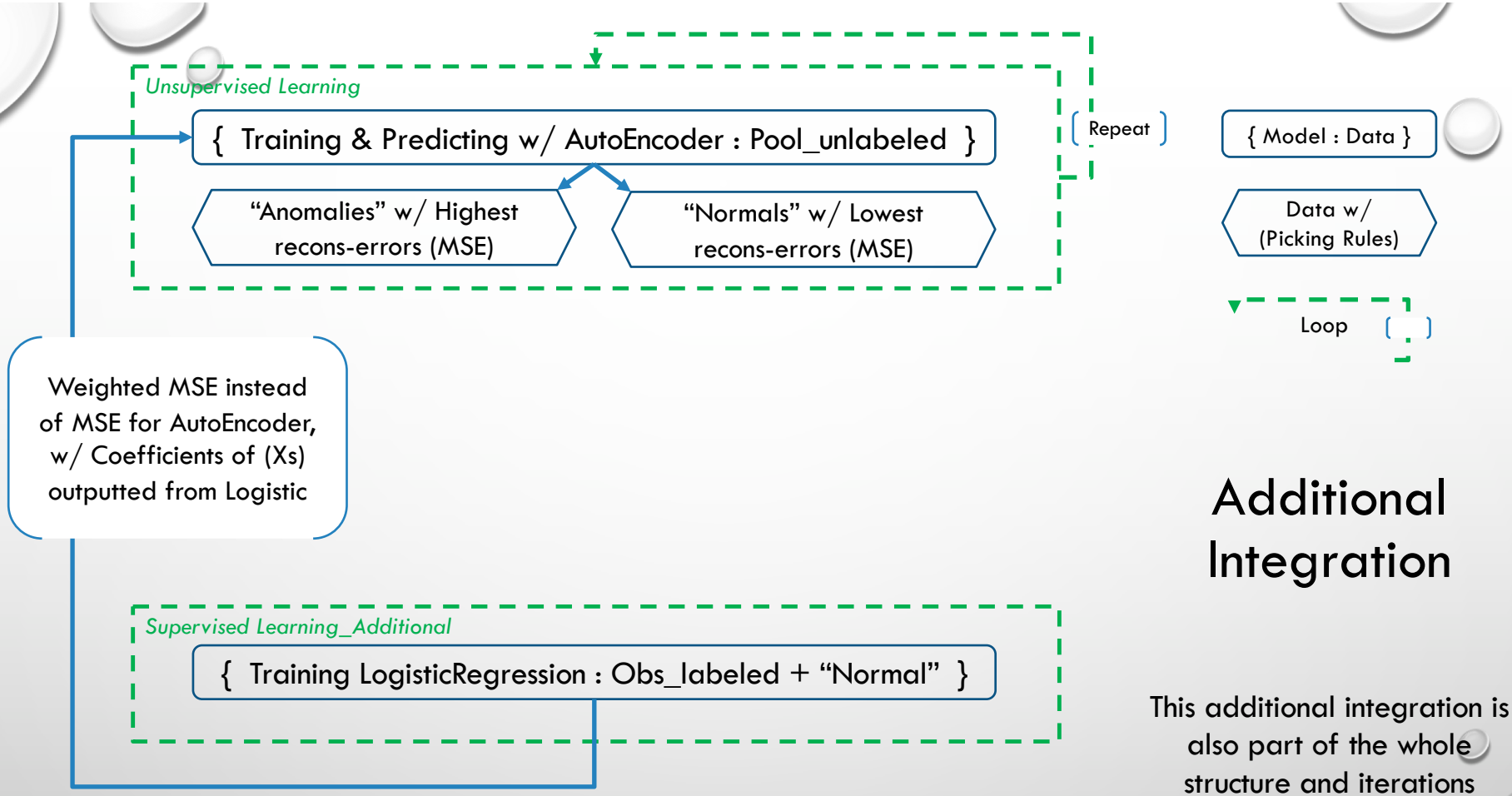
{ Model : Data }

Data w/ (Picking Rules)

Loop

...with Tunings based on Data

Tunings are done according to the feedbacks of Anomaly detection/recognition



Additional Integration

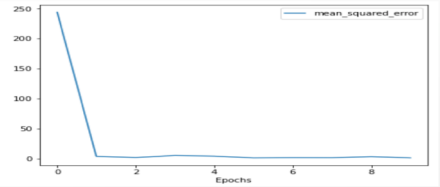
This additional integration is also part of the whole structure and iterations

1 Tuning Unsupervised Learning Model

- AutoEncoder Hyper-parameters
  - epoch = 10
  - regularizer_rate = 0.05
  - dropout_rate = 0, or < 0.001
- Metrics
  - MSE is the metrics for AutoEncoder tuning
  - However, there's no benchmark to achieve with MSE, and it is not true in our case that the lower MSE the better. Instead, the result of Anomaly recognition efficiency is the final criterion.

- Module Tests

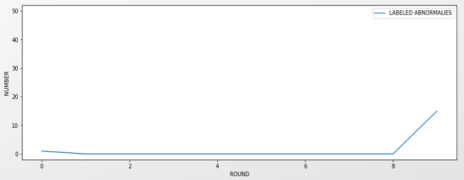
MSE declines for two and more epochs



- Solution Trial_01

dropout_rate = 0.05

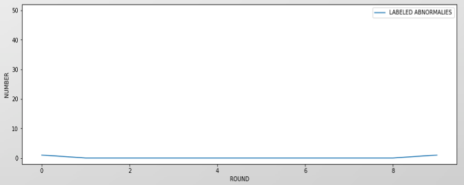
MSE good, but the final result of Anomaly recognition poor



- Solution Trial_02

dropout_rate = 0.01

Similar to Trial_01



1 Tuning Unsupervised Learning Model *(cont.)*

- Stabilization with Repetition
  - To stabilize the Outputs ("Anomalies", "Normals") of Unsupervised Learning, the optional loop is added
  - In each turn of the loop, a new AutoEncoder is created and trained, and then its Outputs are generated
  - After the loop has finished, Outputs from all turns together vote the top overlapped observations
  - Turns of repetition can be tuned, with the hyper-parameters together, to improve the degree of overlapping

- Solution Trial_02

Overlap degree is low, and the final Anomaly recognition result is poor

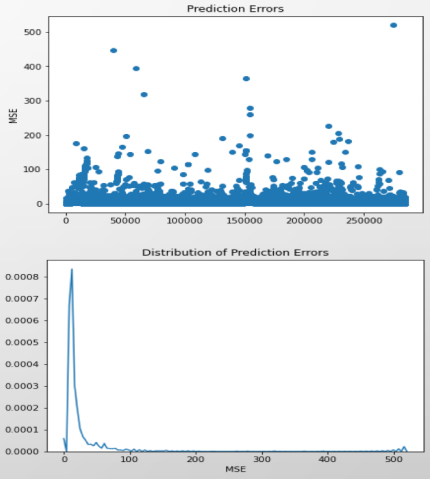
- Solution Trial_03

Overlap degree is higher (but far below 100% to keep some randomness) with :
n_repeat = 20, and
dropout_rate = 0.001 or 0,
and the Anomaly recognition result is better

2 "Normal" Selection by Utilizing Data Imbalance

- Majority Assumption
  - Given the imbalanced data, it is much safer for Unsupervised Learning to predict the majority "Normals" than the minority "Anomalies"
  - Hence, the observations with the lowest MSE are assumed "Normal" and sent as labeled data to Supervised Learning
- Imbalance Estimation
  - In a real world case where the balance of data is unknown, extra estimation steps need to be done in the Module Tests of Unsupervised Learning before the Solution Trials. Sampling by different MSE levels and some consumption of human labeling budget may be needed.
  - How and how much "Normals" are extracted depend on the estimated balance condition

- Module Tests MSE analysis in Unsupervised Learning

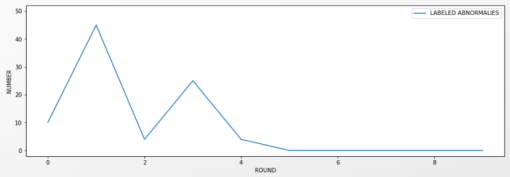


3 Tuning of "Anomaly" Selection

- Motives
  - Two Integration points combining Unsupervised and Supervised Learning in the solution structure:
    1) select "Normal" by utilizing the data imbalance (the last page)
    2) select "Anomaly" with info from both parts
  - In Anomaly selections with only MSE, the Anomaly recognition results decay too faster than expected
- Objectives
  - "Anomalies" selected by MSE (Unsupervised Learning)
  - "Anomalies" selected by predicted Probabilities (Supervised Learning)
  - How to combine them depends again on the Anomaly recognition results

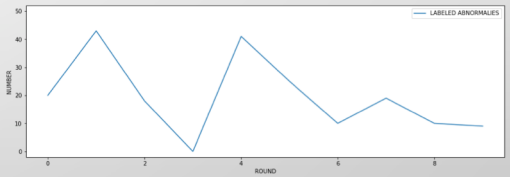
- Solution Trial_03

Anomaly recognition decays fast with selection by only MSE



- Solution Trial_06

Anomaly recognition is not stable with selection by only Probabilities

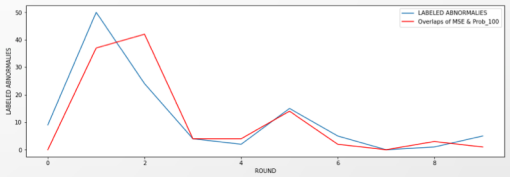


3 Tuning of "Anomaly" Selection *(cont.)*

- Primary-party Approach
  - One Primary party (Unsupervised or Supervised Learning) :
    1) First select Overlapped observations (red curves in the plots on the right) from both parties
    2) Then fill the vacant positions from the Primary party
  - According to Trial_04&07, Supervised Learning actually plays the key role in Anomaly recognition
- Alternative Approach
  - Weighted Selection (e.g. 40% from Unsupervised, 60% from Supervised Learning)
  - If the Anomaly recognition result curves resonate with neither Overlap curves no matter which party as Primary, Weighted Selection could be considered instead of Primary-party

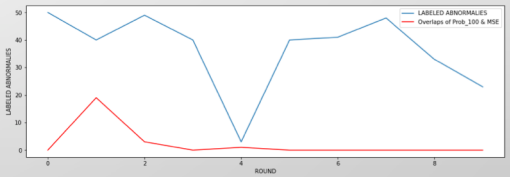
- Solution Trial_04

Anomaly recognition curve resonates with Overlap curve when MSE as primary



- Solution Trial_07

Anomaly recognition curve NOT resonates with Overlap curve when Probs as primary

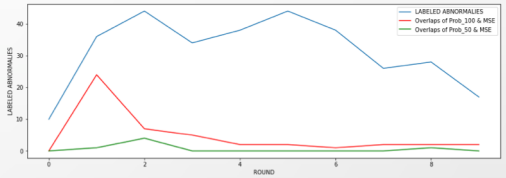


4 Tuning of Supervised Learning Outputs

- What to tune
  - Tuning the Supervised Learning model does not make sense because the feed data vary in each iteration round
  - Rather, tuning what to output by the model. As the outputs play the key role in Anomaly selection (shown in the last page)
- What to output
  - Two types of observations
    1) with Highest predicted Anomaly Probabilities (→100%)
    2) with Highest predicted Uncertainty Probabilities (→50%)
  - Improve the Correct Anomaly recognition rate (over all labeled data) by adjusting the weights in Anomaly selection of the two types of observations (weighing more on Prob_100)
  - Prob_50 observations help in diversity and stabilization

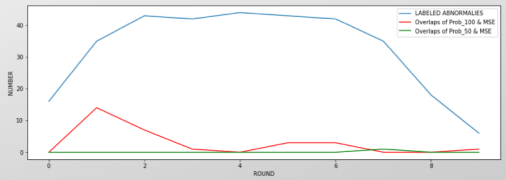
- Solution Trial_08

Correct Anomaly recognition : 60%
with half Prob_100 & half Prob_50



- Solution Trial_10

Correct Anomaly recognition : 70%
with 80% Prob_100 & 20% Prob_50

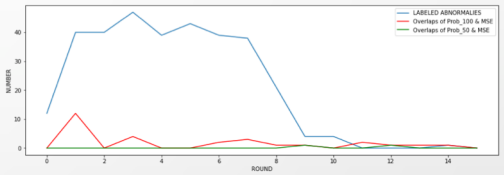


5 Tuning of Stop Point

- Motives
  - Correct Anomaly recognition rate (Detection Efficiency) can be improved with all the approaches, but the Maximum number of recognized Anomalies (Detection Limit) is almost no way to change
  - In real world cases, the Detection Limit is unknown until it is known; the solution should stop as early to save human labeling budget
  - Meanwhile, all available approaches should be taken advantage of before the stop to detect Anomalies as many as possible
- Mechanisms
  - When Detection Efficiency decays (lower than a threshold in two consecutive rounds), switch the Primary-party in Anomaly selection
  - When switches repeat (the 3rd time), stop the iteration as the Detection Limit has been reached

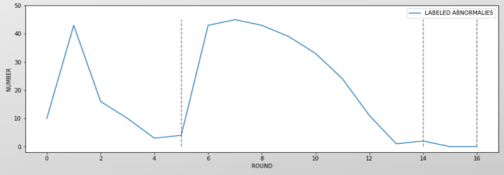
- Solution Trial_11

Once Detection Limit reached, waste of human labeling budget begins



- Solution Trial_14

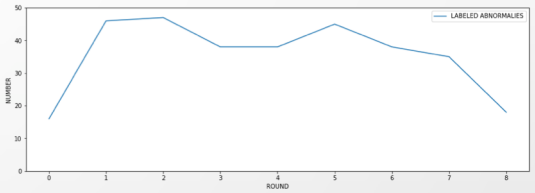
Solution performance with stop point in the worst situation



The Best Performance : Task Accomplished

- Tuning
  - Unsupervised Learning model hyper-parameters fit
  - Unsupervised Learning outputs stabilization with repetition
  - "Normal" selection by utilizing data imbalance
  - "Anomaly" selection Primary-party: Supervised Learning
  - Supervised Learning outputs: Prob_100 (80%), Prob_50 (20%)
  - Additional Integration: Weighted MSE for Unsupervised Learning
  - Stop criterion set as Detection Limit (80%)

- Solution Trial_15 Champion Trial



- Performance

	Trial_15 (best)	Trial_14 (worst)
Iteration Rounds	9	17
Anomaly Detection rate (over total Anomalies)	82%	82%
Detection Efficiency (over all labeled data)	71%	38%

- Performance of a common case should fall into the range between these two

Data Validation

- Train Classifiers with the Labeled Data
  - XGBClassifier model
  - Data

    1) 450 human labeled + 450 “Normal” (from Trial_15)

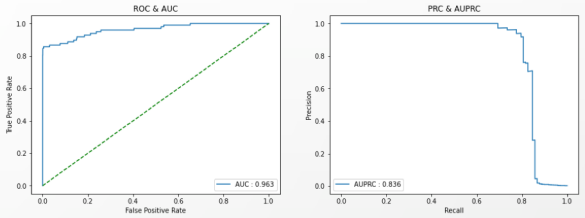
    2) 850 human labeled + 850 “Normal” (from Trial_14)

    3) All original full-labeled Train dataset (227K)
- Test Classifiers with the reserved Test Data
  - Test Dataset :

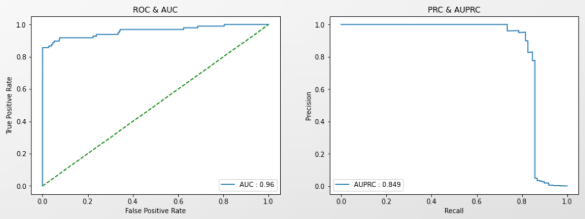
    20% of the whole data

    Stratified split
  - Test results have no significant difference between the three cases

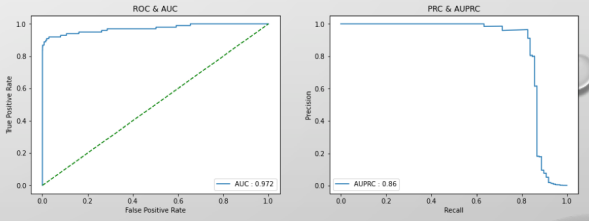
- with Data 1)



- with Data 2)



- with Data 3)



FUTURE IMPROVEMENT

This project explores Anomaly detection using Active Learning techniques, especially the approaches integrating Unsupervised and Supervised Learning. For future improvement, further approaches include :

- **Data Imbalance Estimation**

  When the balance status of data is unknown, it needs to be estimated by Unsupervised Learning and carefully subsampling for human labeling. This is a necessary step in Module Tests for better understanding how to run the "Normal" selection in the solution.

- **Supervised Learning Model Selection**

  The simple XGBClassifier model is adopted for Supervised Learning, as it performs well for the data in this case. Several alternative models could be tried simultaneously in each iteration round to vote the most representative observations (as the repetition of AutoEncoder does), if none of them prevails in Anomaly recognition.