Online Diagnosis of Performance Variations Using Machine Learning

Efe Şencan
25083
Sabanci University
Computer Science and Engineering
June, 8, 2020 – September, 4, 2020

Motivation

- Increasing size of HPCs.
- Performance variation due to shared resource contention as well as software and hardware-related problems.
  - Memory leaks
  - Orphan processes leftover from previous jobs
  - Reduced CPU frequency due to hardware problems
  - Shared recource contention
- These problems can cause failures and inefficiencies.

Tuncer et al., 2019, TPDS

Peaclab Team’s Framework

- Automatically diagnoses performance anomalies at runtime using machine learning
- Identifies 98% of the injected anomalies with negligible overhead.
- They utilized synthetic anomalies.

Challenges

- The data collected by real-world HPC monitoring tools does not contain ample labeled data.
- Creating the ground truth values by the experts are very time consuming.
- It becomes infeasible to obtain labels as the data size increases.

Objective of the project

- Design and implement scalable methods such as such as:
  - Semi-supervised learning
  - Unsupervised learning
  - Other statistical methods

- Achieve adequate amount of classification performance with less labeled data.

Methods & Tools

- Utilized **Python** as a programming langıuage and **Jupyter Notebook** as a development environment.

- **Semi-supervised learning**
  - Self-learnign
  - Cluster then label
  - Graph-based methods
- **Active Learning**
  - Different query strategies:
    - Margin Sampling
    - Entropy Sampling
    - Uncertainty Sampling

“Zhu et al., 2009, Goldberg, Morgan & Claypool Publishers”
“Sun et al., 2010, ICMLC”

Experimental Methodology

- Dataset: HPAS Dataset
  - Memleak
  - Memeater
  - Cpuoccupy
  - Cachecopy
  - Membw

- Applied sliding window (window_size = 45) and feature extraction to capture the characteristic of time series data.

- We run the experiments at SCC(Shared Computing Cluster).

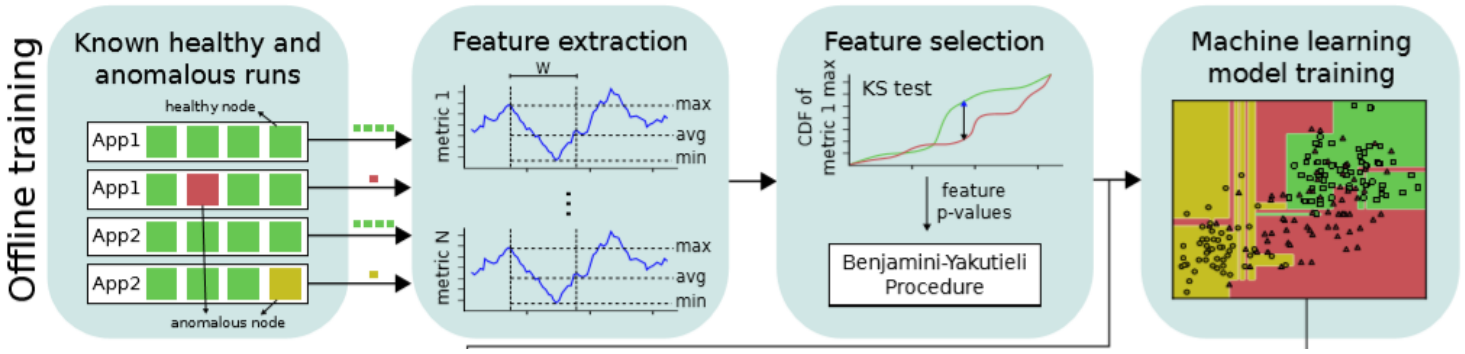
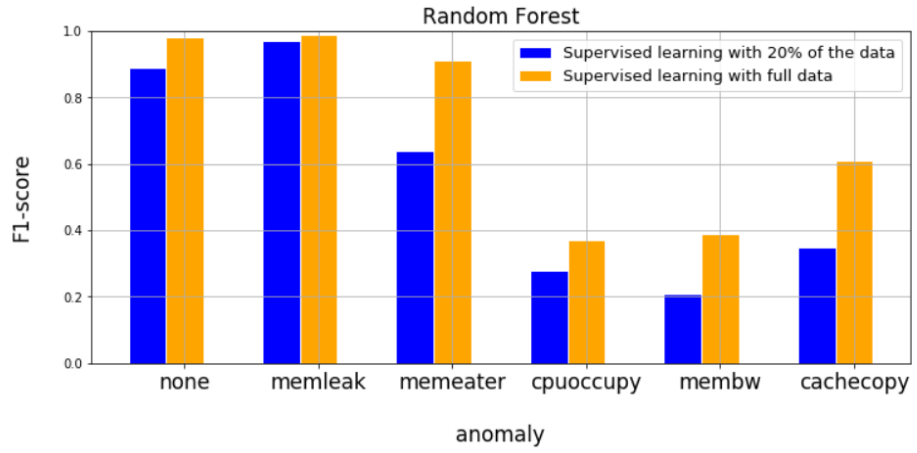
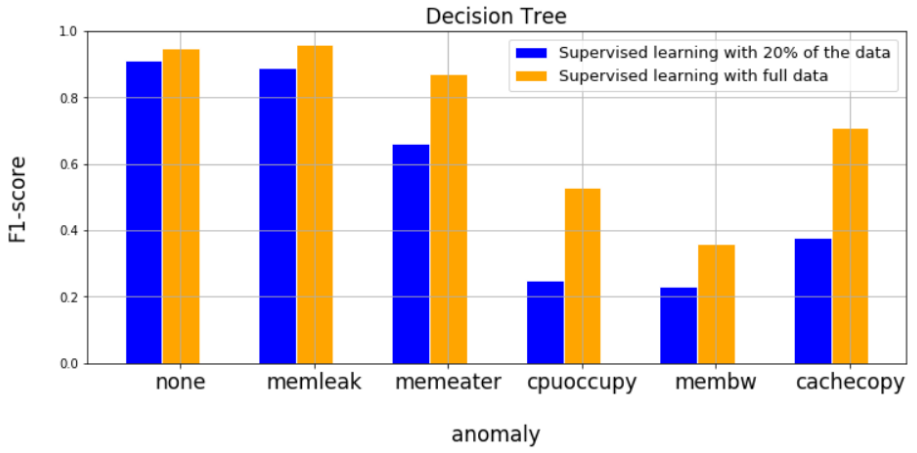


Figure taken from [TPDS, Tuncer et. al, 2018]

Experimental Methodology

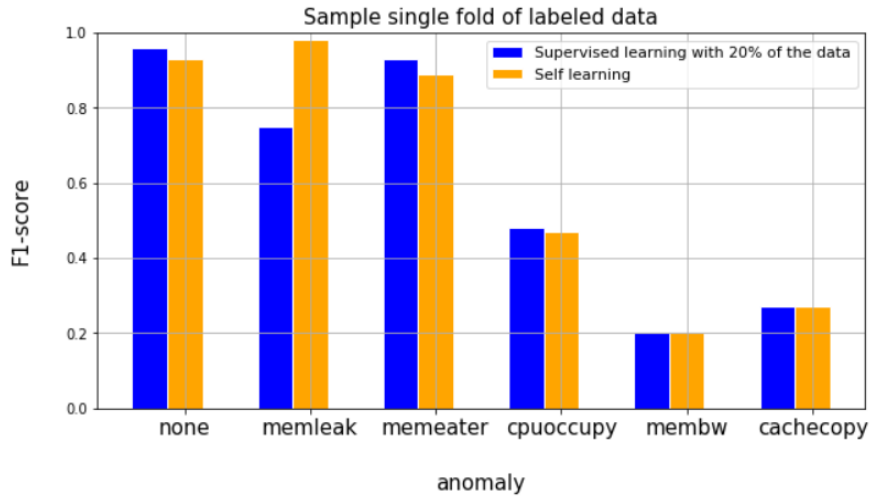
- We splitted the dataset into disjoint train (80%) and test (20%) sets
- Evaluated the performance of the model using supervised learning with 3-fold cross validation using the whole data
    - **Baseline:** To mimic the scarcely labeled data scenario, we dropped the 80% of the labels in the train set while preserving the app-anomaly distribution in the train set.
        - Trained our model with the scarcely labeled train set.
    - Compared f1-scores of the anomalies based on our model's test set performance.

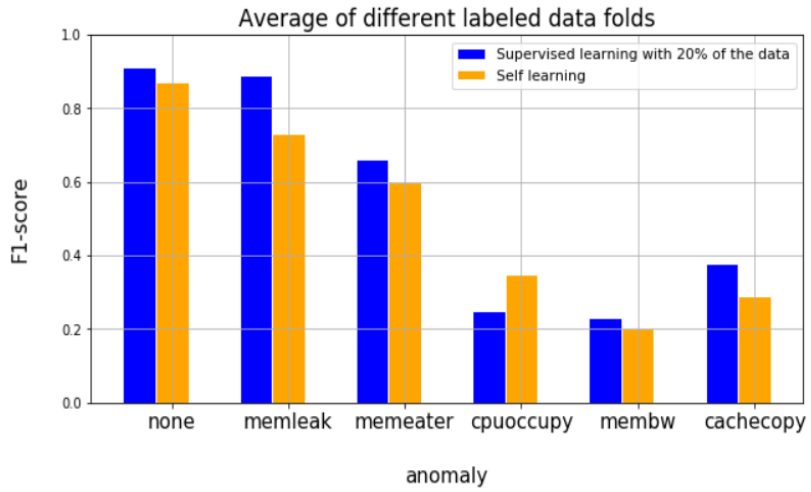




Self-Learning

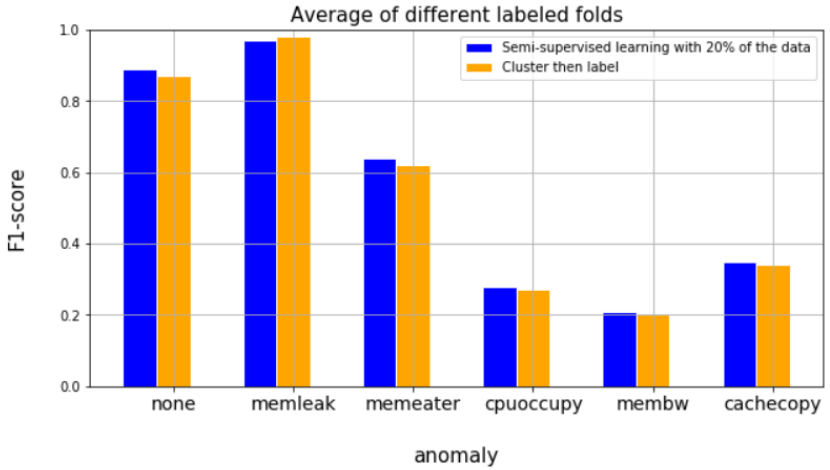
- How does self-learning work?
  - Train the classifier f with labeled data, and apply f on unlabeled data.
  - If the predictions are greater than the probabilistic threshold, then we label that instance and remove it from the unlabeled set.
  - Keep that process until the convergence criteria is met.
- Experiment methodology:
  - Used 20% of the labeled data (slide 8) as a train data and evaluated the model's performance using the same test set (slide 8).





Cluster then Label

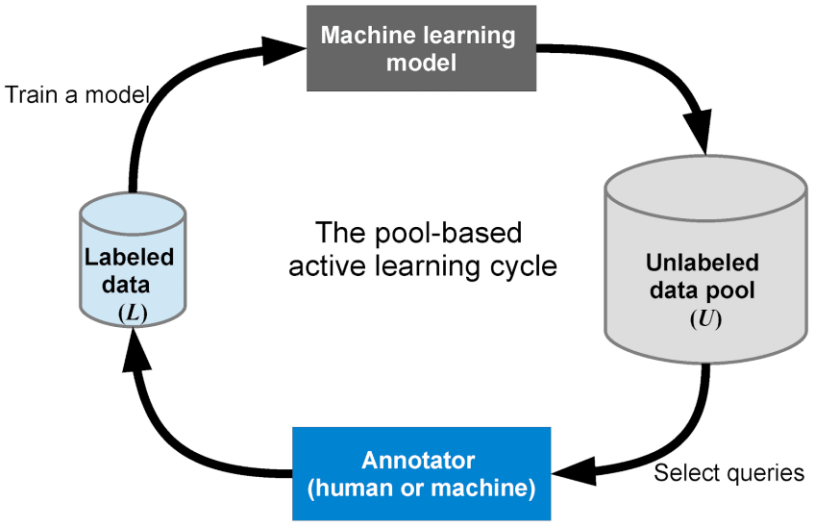
- How does cluster-then-label work?
  - Cluster both labeled and unlabeled data with a clustering algorithm C.
  - For each cluster, predict the unlabeled instances by using the labeled instances in that cluster with a classifier f.
  - If there is not any labeled instance in that cluster, then the whole labeled data in the dataset (20% of the train data in our case) is used to predict these unlabeled instances for that cluster.
- Experiment methodology:
  - Used the same train (contains both labeled and unlabeled data) and test set (slide 8).
  - Utilized k-means clustering.
  - Used decision tree as a cluster classifier.

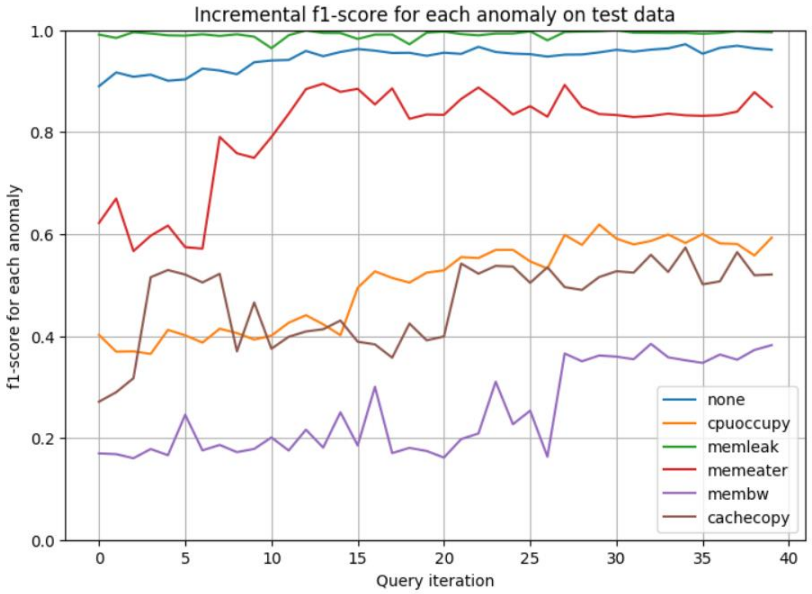


"Zhu et al., 2009, Goldberg, Morgan & Claypool Publishers"

Active Learning

- Useful when there are lots of unlabeled data points and querying the label is costly.
- It can be helpful to increase the classification accuracy with limited number of queries.
- Margin sampling as a query strategy is used for the experimental purposes.

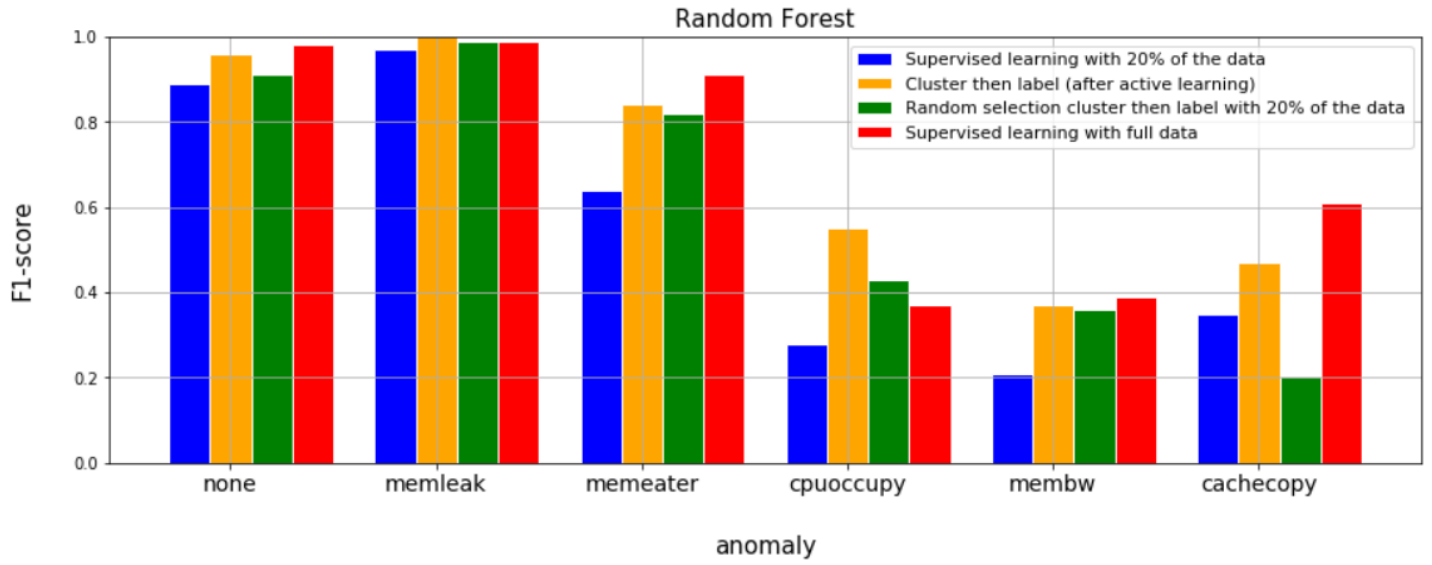




"Yang et. al, Visually-Enabled Active Deep Learning for (Geo) Text and Image Classification: A Review", 2018

Performance after Active Learning

- We queried 40 labels (8% of the train set) from the unlabeled pool with margin sampling strategy and extended our labeled dataset.
- We used the newly created (after active learning) labeled instances for the cluster-then label method and compared the performances.



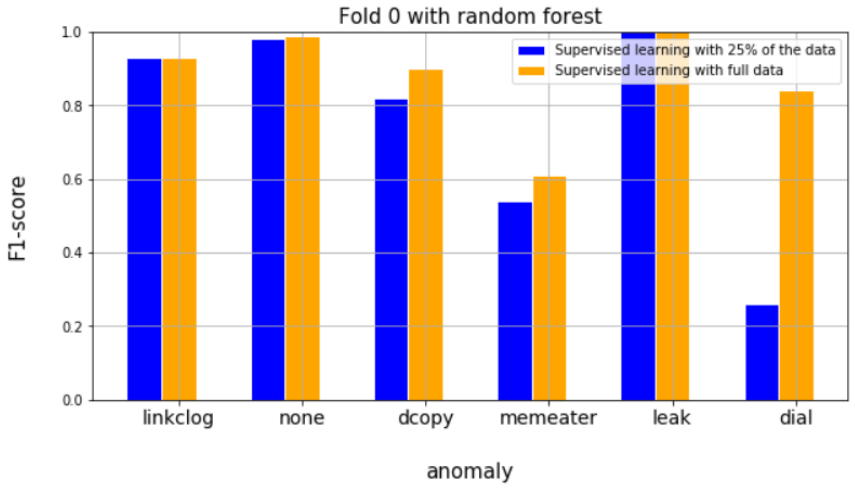
TPDS Experiment

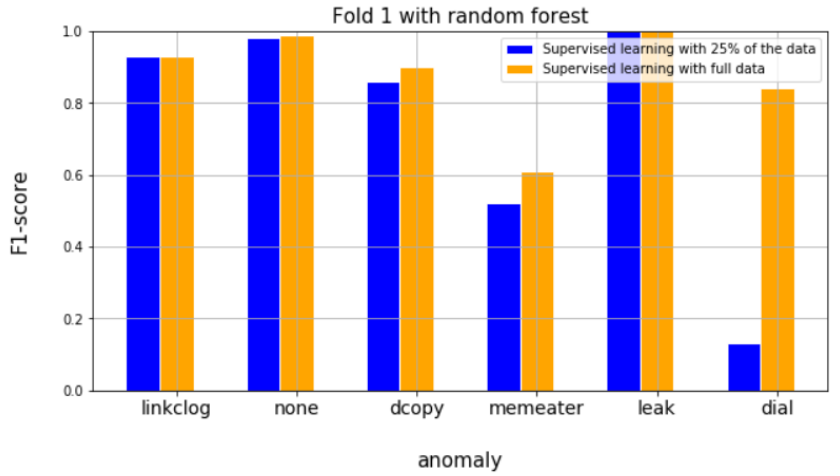
- 10.8 GB of data was selected from the whole TPDS data.
  - Train data (~ 72%), Test data (~ 28%).

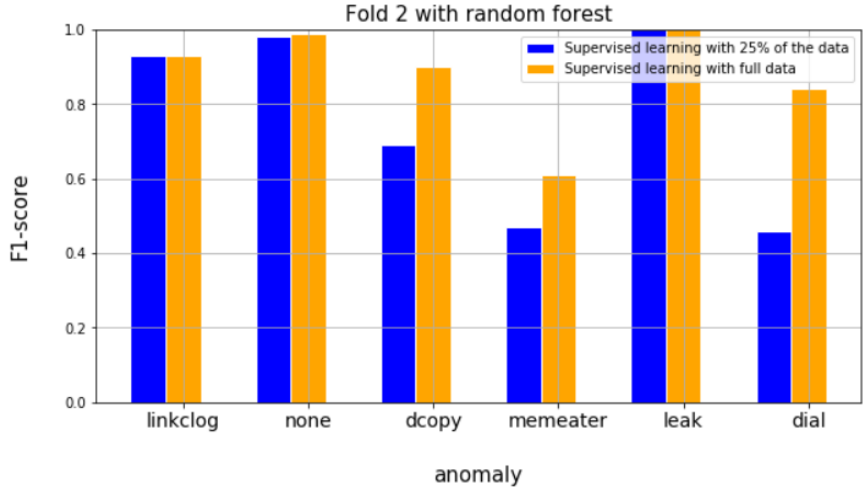
- Applied feature selection and feature extraction to train and test data.

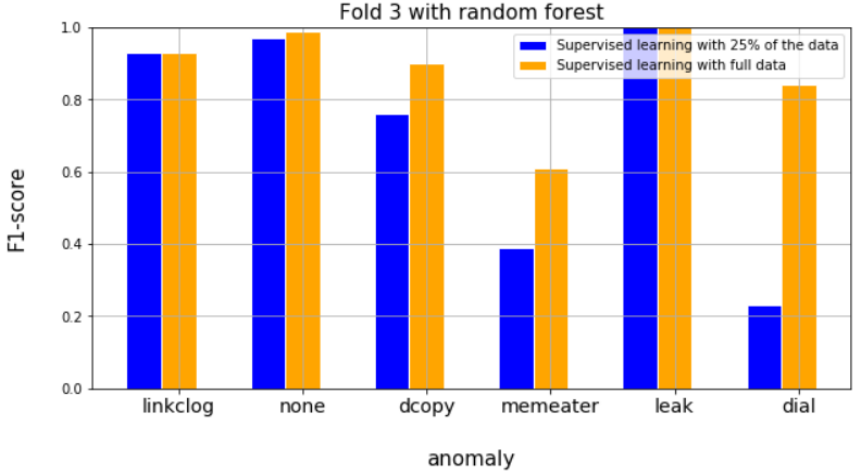
- To mimic the scarcely labeled scenario:
  - Applied stratified k-fold with two different cases:
    - Dropped 75% of the labels from train data (4 different folds)

Comparison of Baseline with Full Data Performance

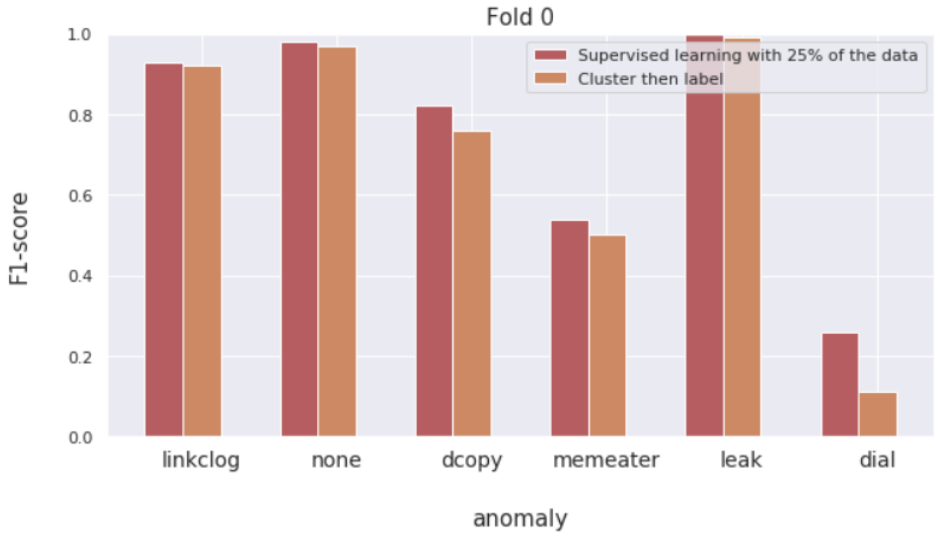


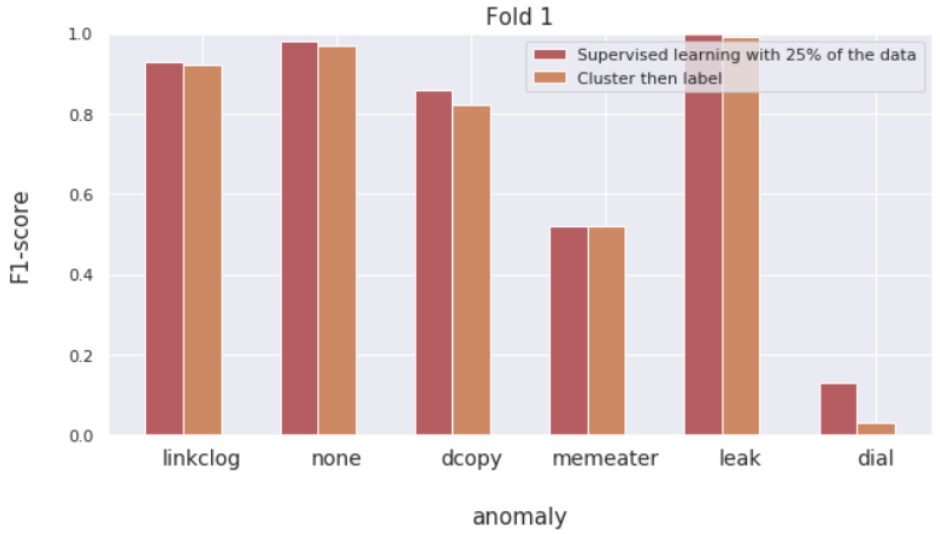


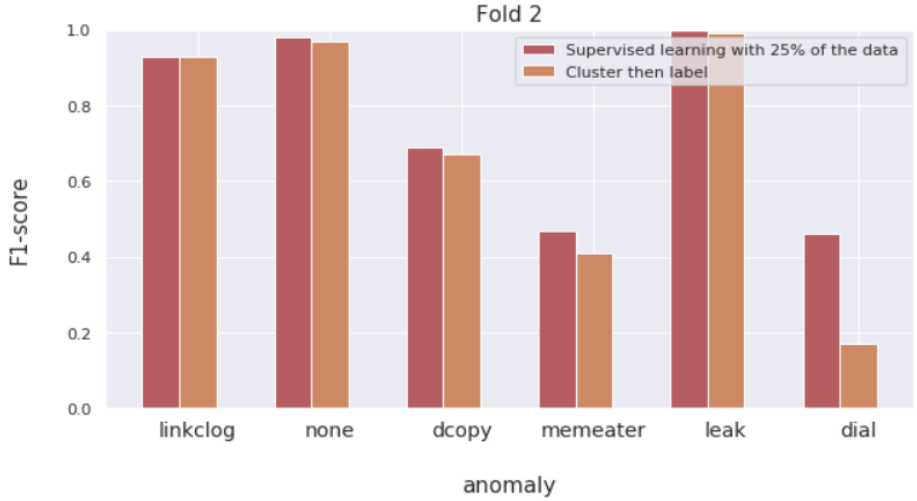


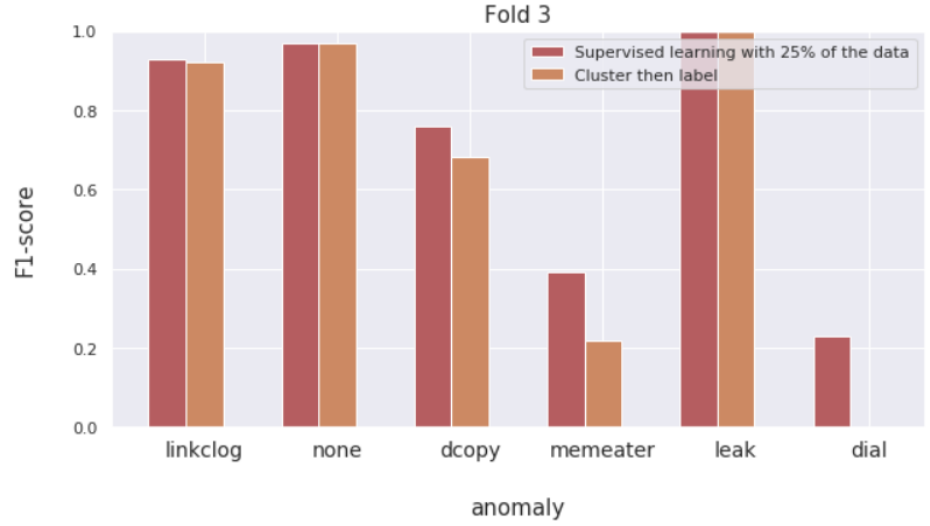


Cluster then label Results

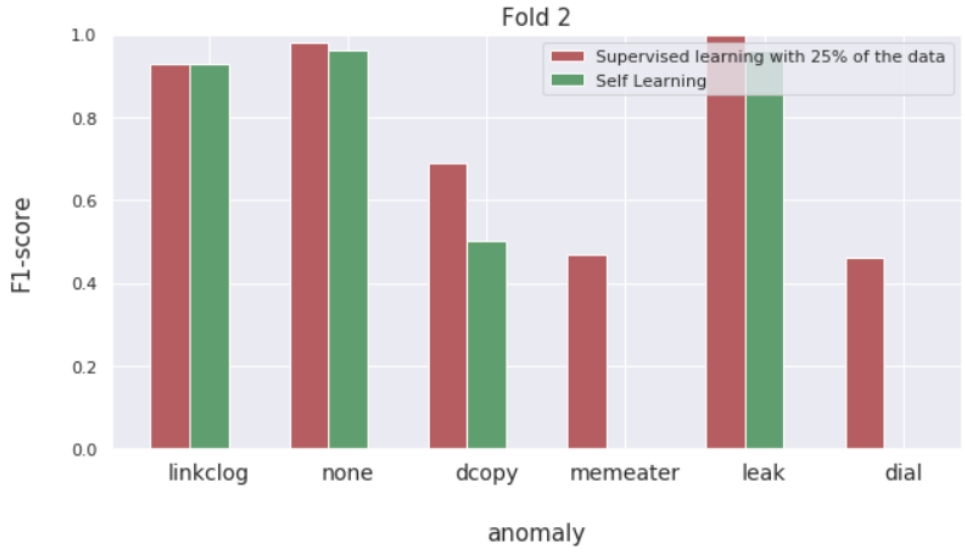




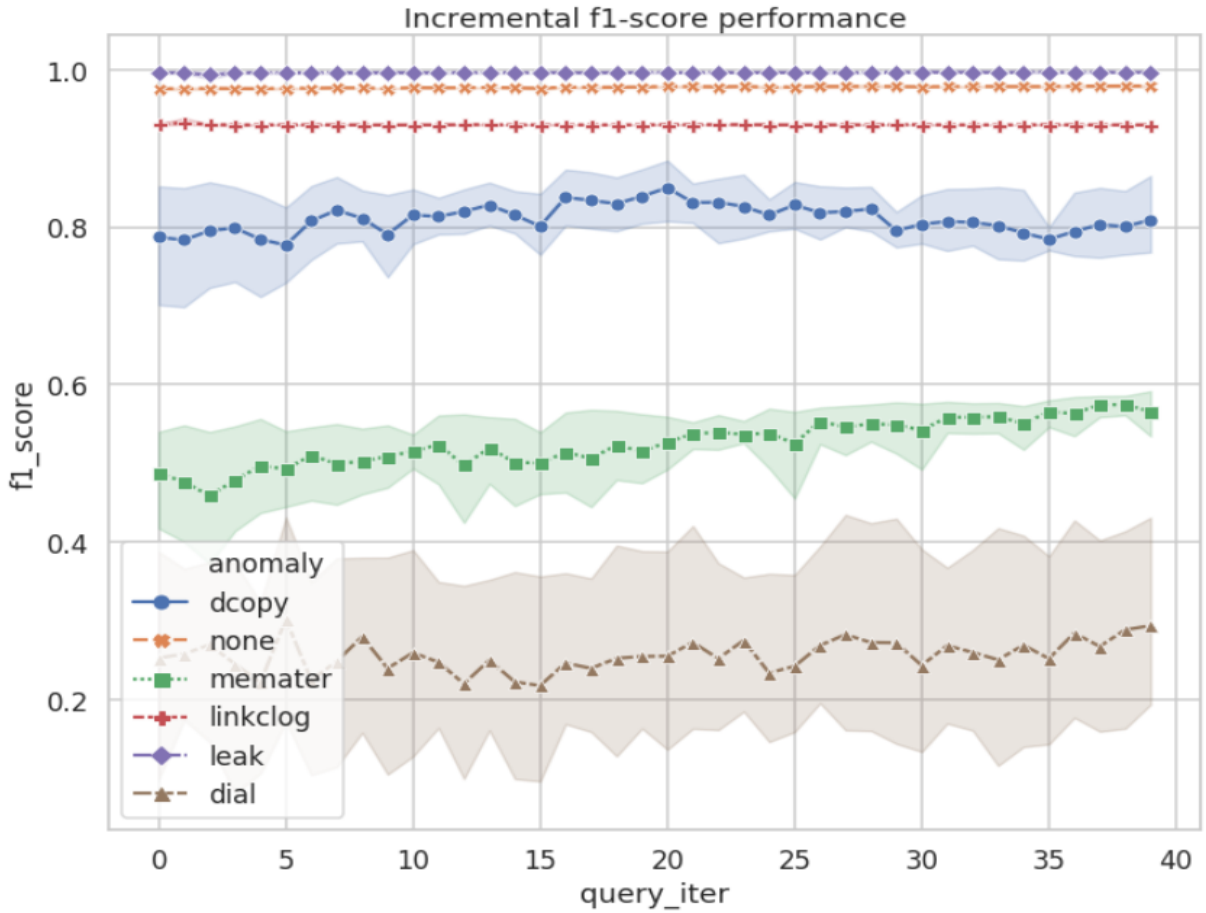




Self-Learning Results



Active Learning Results



Outcomes & Deliverables

- Designed and implemented several semi-supervised learning algorithms such as self-learning, cluster then label and graph-based methods.
- Utilized active learning to detect important applications working on HPC Sytems for anomaly diagnosis.
- Analyzed the performance of different machine learning models with different percentage of the labeled data.
- Combined active learning with semi-supervised methods and evaluated the performance of the models.

What Did We Learn?

- Tree algorithms as a base learner in self-learning does not always produce better results because of the poor probability estimation at the leaves.
- Performance of the cluster then label depends on the clustering performance as well as the how the data fits to the clustering assumption.
- Active learning can be useful for determining the instance to be queried.
- Active learning does not increase the f1-score of all anomaly types in a single iteration, the information of one application-anomaly pair can lead to performance decrease for others.

Tanha et al., “Semi-supervised self-training for decision tree classifiers”, *International Journal of Machine Learning and Cybernetics* volume, 2015