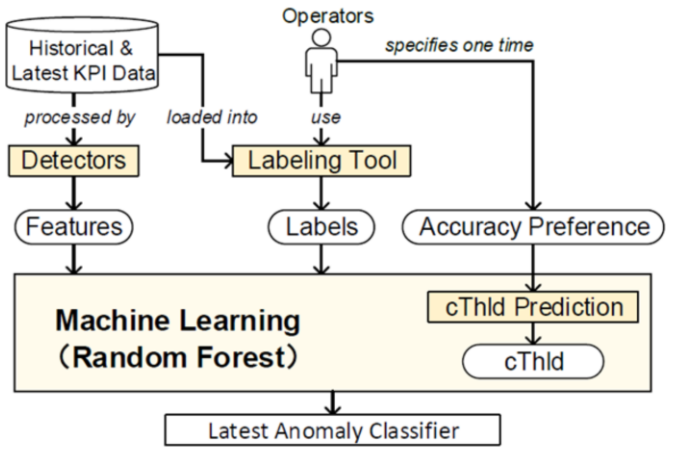
Design Overview

Training a classifier



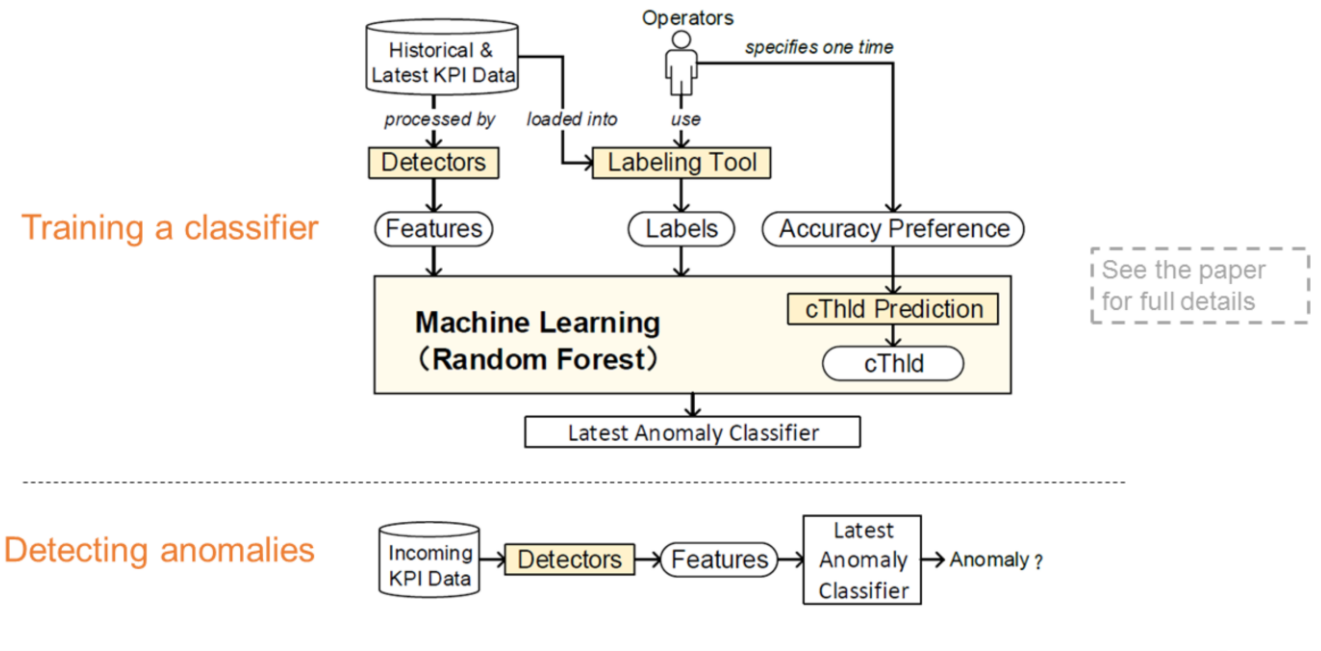
See the paper for full details

This is the design overview of Opprentice.

First we train the anomaly classifier. Give the KPI dataset, dozens of detectors extract the features, and operators provide the anomaly labels. The features and labels are used to train a random forest classifier. Besides, operators specify their accuracy preference in the form of precision and recall. It is used to adjust the classification threshold of the random forest.

The classifier is incrementally re-trained with the latest data. For example, we can do it once a week.

Design Overview



For the detecting process, the same detectors extract the features of the incoming data. Based on these features, the latest classifier will decide the class of each data point, anomaly or not.

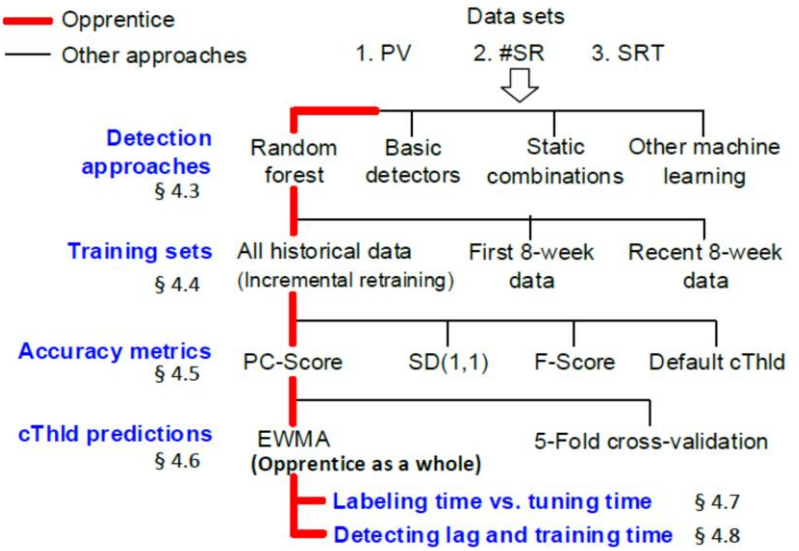
Please read our paper for the details of each component.

Outline

- Background and Motivation
- Key Ideas
- **Results**
- Conclusion

I will show you the evaluation results of Opprentice

Evaluation

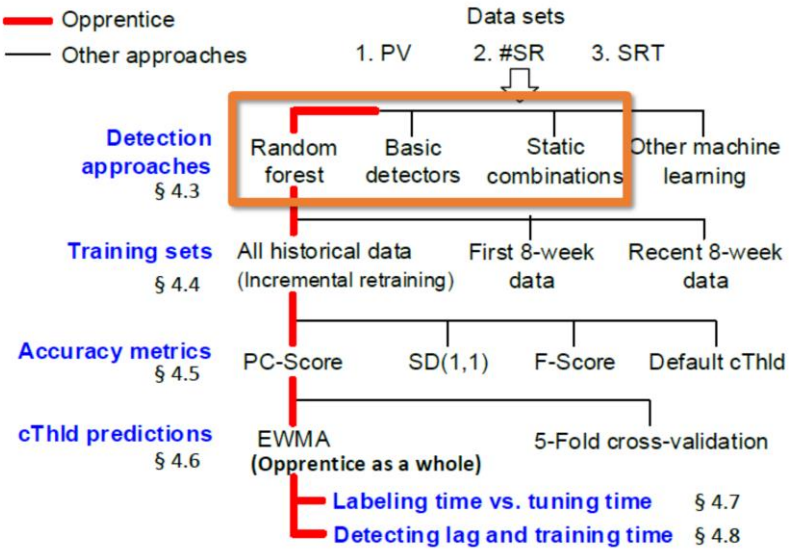


This is our evaluation flow. We use three representative KPIs from Baidu. They are labeled by the operators. We compare different components of Opprentice with other methods.

With all these four parts combined together, we get Opprentice as a whole here.

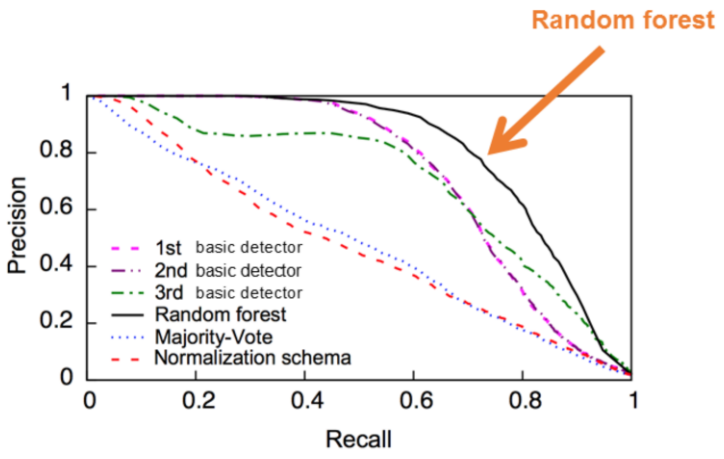
We also evaluate the labeling time, the online detecting lag and the offline training time of Opprentice.

Evaluation



Let's firs look at the comparison between the random forest and the basic detectors and two static methods for combining different basic detectors.

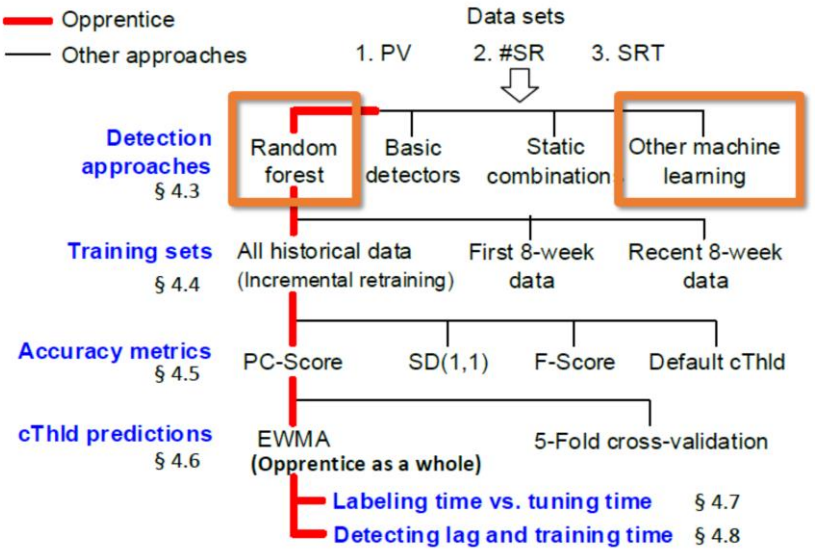
Random forests vs. Basic Detectors and Static Combinations



This figure shows the precision-recall curves, or PR curves on the PV dataset. The results of other two KPIs are similar.

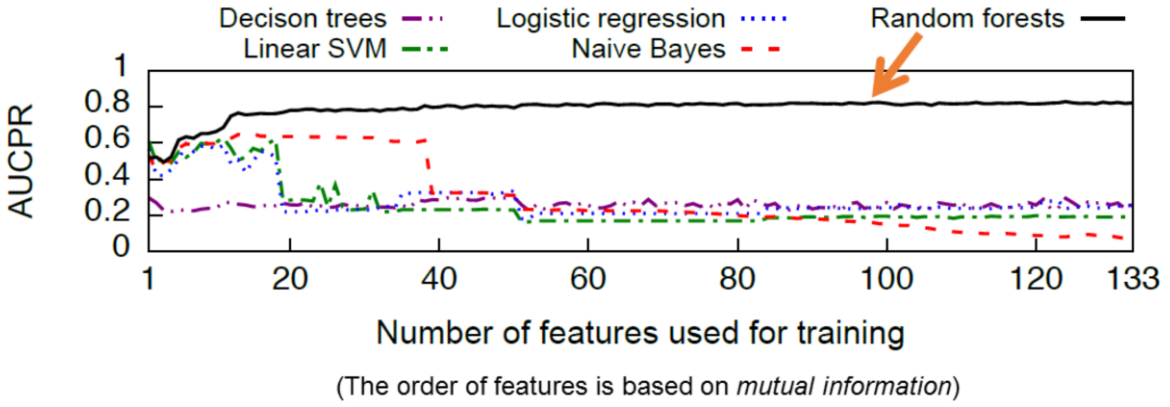
Empirically, the area under the PR curve is larger, the detection approach is better. We see that random forest outperforms other detection approaches, including the top-3 best basic detectors, and two static combination methods (majority-vote and normalization schema).

Evaluation



We also compare the random forest with other machine learning algorithms.

Random Forests vs. Other Learning Algorithms

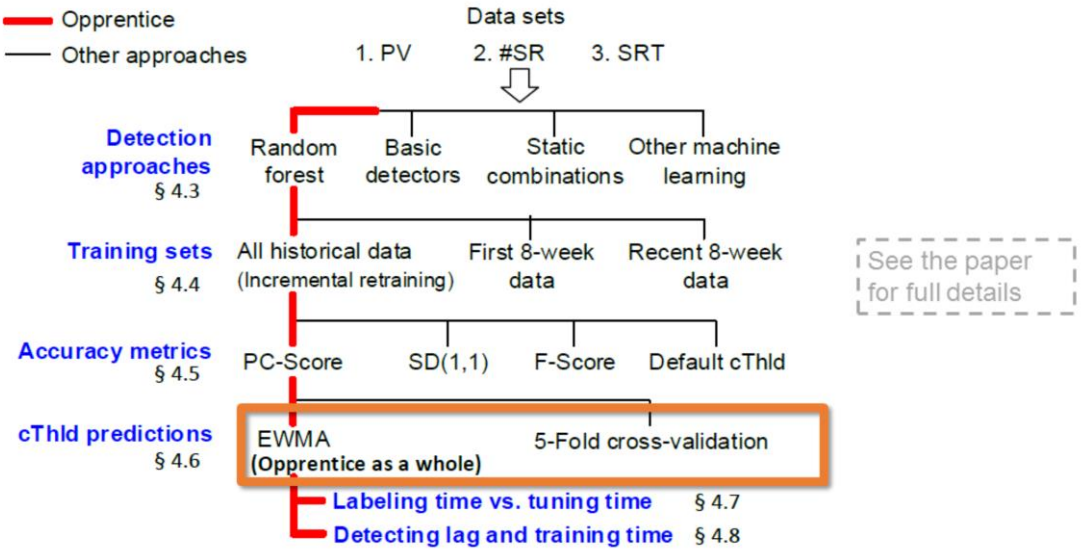


(The order of features is based on *mutual information*)

AUCPR means the area under the PR-curve, and it is the larger the better. In order to see how different machine learning algorithms perform when faced with irrelevant and redundant features, we rank the 133 features based on mutual information, a typical feature selection method, and add the features one by one.

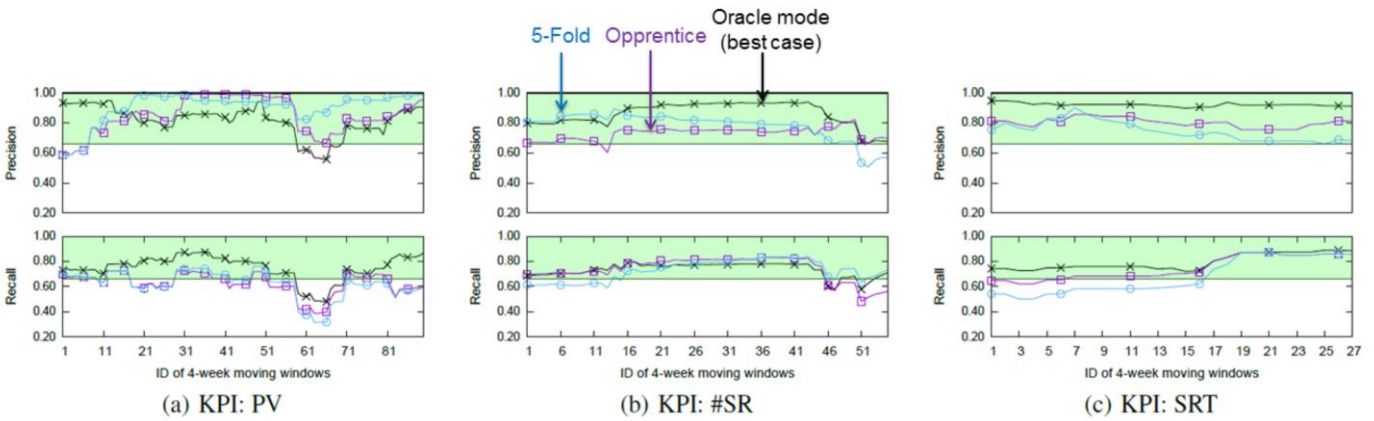
We see that the random forest is almost not affected when we use more features, on the other hand, the performance of other learning algorithms decreases a lot. The result shows that the random forest is more robust to irrelevant and redundant features

Evaluation



I will just skip other detailed results, and show the performance of Opprentice as a whole. In this step, we compare the EWMA used by Opprentice to predict the classification threshold with another method 5-fold cross-validation.

Opprentice as a whole



(a) KPI: PV (b) KPI: #SR (c) KPI: SRT

Opprentice achieves

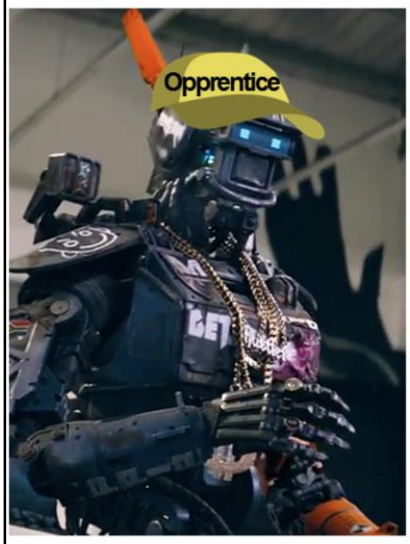
40% **23%** **110%**

more points inside the preference regions than 5-Fold cross-validation

The green regions represent the preference of precision and recall given by the operator. Overall, we see that for the three KPIs, Opprentice can satisfy or approximate the preference most of the time.

When compared with 5-fold cross-validation, Opprentice achieves more points in those preference regions.

Conclusion



- Opprentice is an **automatic** and **accurate** machine learning framework for KPI anomaly detection

~~Defining anomalies~~ ~~Selecting detectors~~ ~~Tuning detectors~~

- Opprentice **bridges the gap** in applying complex detectors in practice
- The idea of Opprentice
  i.e., **using machine learning to model the domain knowledge**

could be a very promising way to automate other service managements

In summary

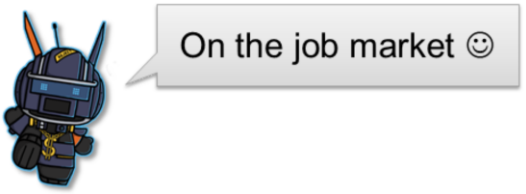
First, #Read#

Second, Opprentice bridges the gap in applying complex detectors from literatures in practice

Third, #Read# , Such as diagnosing root causes

Thank you

liudp10@mails.tsinghua.edu.cn



So, that's all for Opprentice, and I'm happy to take questions now

Thank you.

Backup

Questions

- What if anomalies are frequent?
  - First, based on our experience in Baidu, anomalies are not that many, or someone will be fired
  - If it is the case, we do not have to label all the anomalies, machine learning is good at dealing with noisy data
  - But, it is more promising to test the performance of Opprentice with different sizes of noisy data. We consider it as future work, after all, right now we do not find frequent anomalies as a common case
- What if there are many KPI curves?
  - Different KPIs are operated by different operators, and the KPIs per operators are not that many
  - The anomalous behaviors of different KPIs can be similar, such as the page views of different ISPs. So we can reuse the classifier for those similar KPIs (we need to do the normalization for feature extraction)
  - This could be a great extension for Opprentice. We can explore this direction in the future.

Questions

- If operators do not know anomalies? Or some anomalies may not be visually identified from the curve?
  - First, the anomaly we focus on in this work is the anomalous behavior on single KPI curve (Not troubleshooting)
  - Our goal is detect anomalies confirmed by operators. After all, operators are the users of the detection system. If the anomalies they do not agree with, they do not know how to start investigation and may just ignore it.
- Precision and Recall are not very high
  - Because in the evaluation, we use the performance of fixed window size, 4 weeks. But anomalies are rare (actually, less than 4% for some weeks). For example, missing just a few could decrease recall a lot
  - But, in this case, low recall does not mean we miss a lot of anomalies, the absolute number is small too, so the performance is not that bad

Questions

- Why random forests work better than others
  - There are a lot of irrelevant and redundant features
  - The two properties of random forests: ensemble and random (talk offline)
- What about network traffic data
  - Opprentice can deal with traffic data, it is similar with the PVs of Baidu (seasonality)
  - We apply Opprentice to a traffic data in our journal paper
- What do you mean by complex detectors
  - They take advantage of some more complicated techniques, such as wavelet analysis. It takes more time for operators to learn and understand such detectors
  - They have more parameters, or their parameters are not intuitive to tune
- Is Opprentice deployed?
  - We have build a prototype of Opprentice, and evaluate it with real data from Baidu
  - We are now working on deploying Opprentice in Baidu to detect PVs of different products