# Erratic Server Behavior Detection Using Machine Learning on Basic Monitoring Metrics

Given a history of a distributed application workload and a cluster of servers running said application, it is possible to detect an erratic server by analyzing only the os-level metric data.

# Goals

1) Gather a dataset

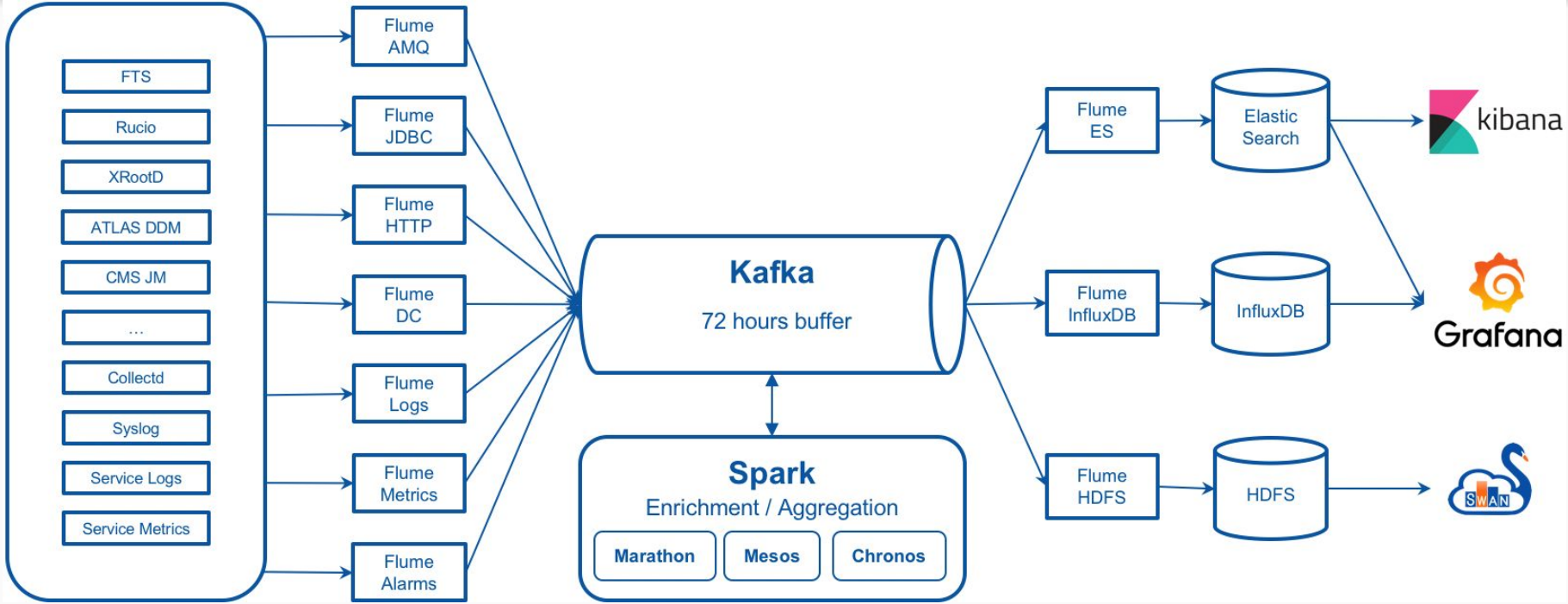
2) Train a machine learning model on the gathered dataset that is able to recognize an erratic node

# 1) Gather a Dataset

# Dataset properties

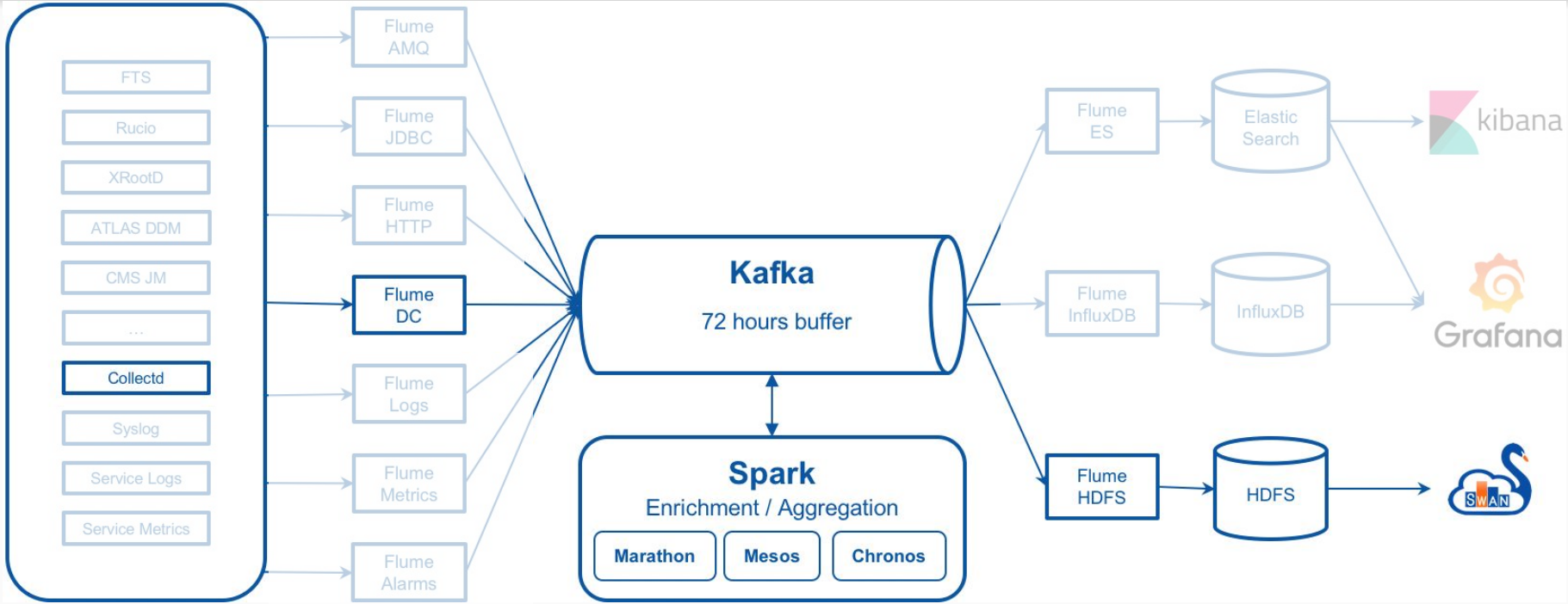
1) OS metrics from a cluster of equal servers running a load balanced distributed application
2) Metrics can be correlated
3) Capture numerous anomalies

# MONIT




http://monitdocs.web.cern.ch/monitdocs/overview/architecture.html

# MONIT




http://monitdocs.web.cern.ch/monitdocs/overview/architecture.html

# Topic: collectd-raw-*


```
{ "metadata": {...},

  "data": { …

       "host": "monit-kafkax-dev-b3b87d619b.cern.ch",

       "plugin": "cpu",

       "type_instance": "nice",

       "value": 0.120220690839613,

       "time": 1578144918.605,

         }

}
```

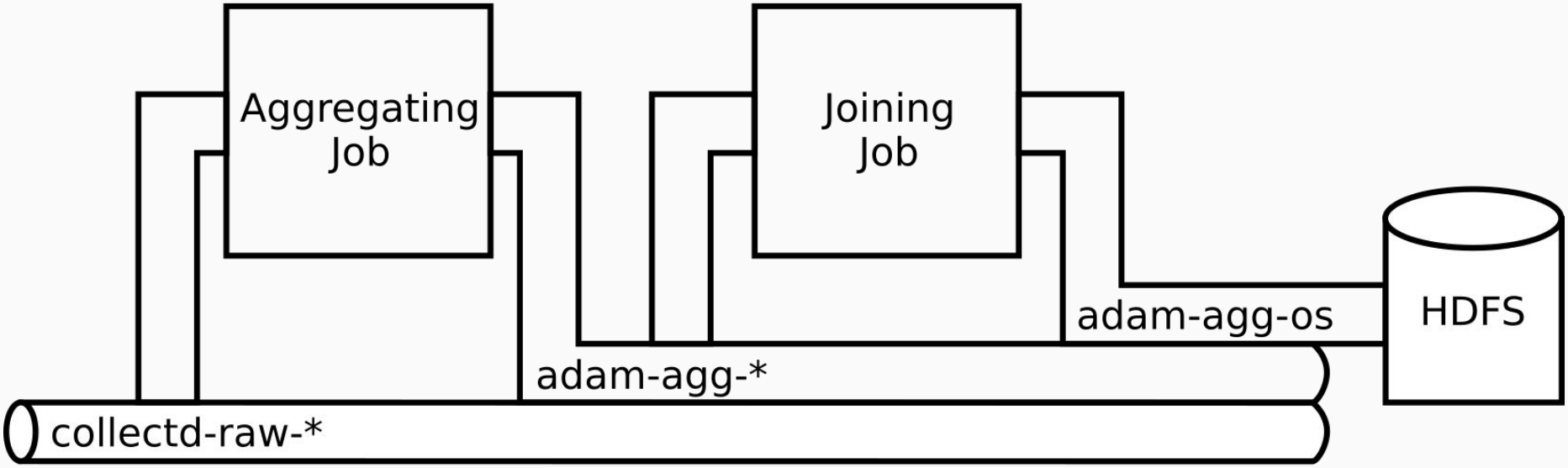
# Topic: adam-agg-*


```
{ "metadata": {...},
  "data": { …
       "host": "monit-kafkax-dev-b3b87d619b.cern.ch",
       "plugin": "cpu",
       "type_instance": "nice",
       "value": 0.120220690839613,
       "window": {
             "start": "2018-05-17T09:00:00.000Z",
             "end": "2018-05-17T09:20:00.000Z" },
         }
}
```

# Topic: adam-agg-os


```
{ "metadata": {...},
  "data": { …
       "host": "monit-kafkax-dev-b3b87d619b.cern.ch",
       "window": {
             "start": "2018-05-17T09:00:00.000Z",
             "end": "2018-05-17T09:20:00.000Z" },
       "cpu_idle": 30.462260570062956,
       "cpu_nice": 0.120220690839613,
       "free_memory": 163472588.8, …
         }
}
```

# Gathering Pipeline in MONIT



# Gathering Pipeline in MONIT


- ~25 CPU
- >110GB RAM
- 20k files in HDFS checkpoint
- 13MB/s compressed input


- ~2 CPU
- ~10GB RAM
- 50k files in HDFS checkpoint

# 1b) Anomalies

# 1b) Create Anomalies

# Anomalies

| Anomaly Type | Count |
| --- | --- |
| Service Stop | 6 |
| Memory Over-use | 2 |
| CPU Over-use | 8 |
| Combined (CPU+Mem) | 10 |

# Dataset

- 26 anomalies spread over 19 days
- Each sequence preceded by >3 days of stable operation

- 10 and later 36 collectd metrics

# 2) Data Analytics

# Data Pre-processing

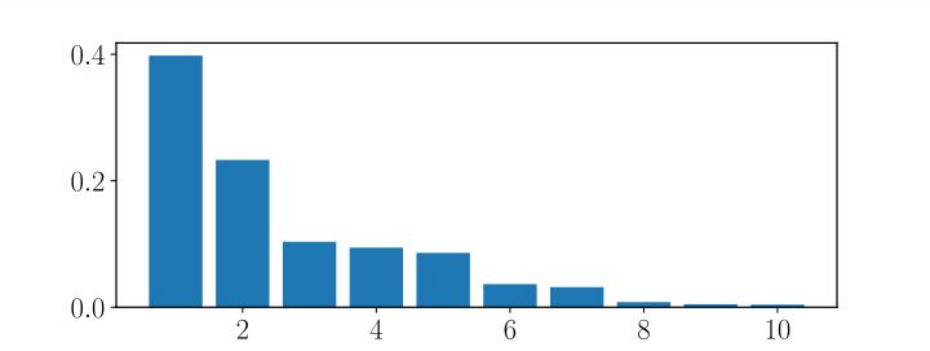
Even 10 metrics might impact performance

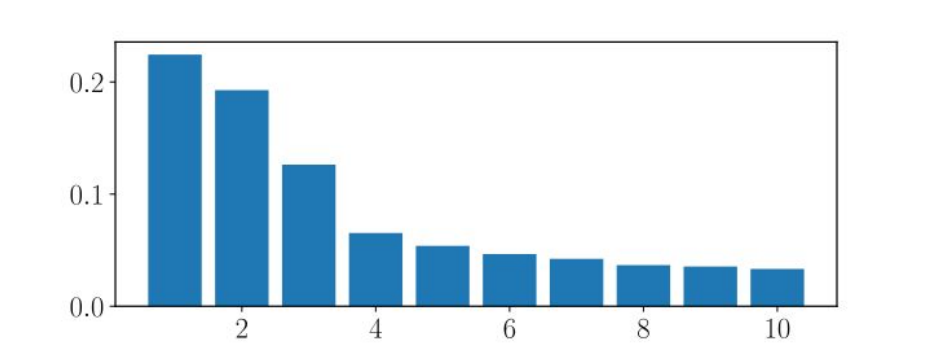
Principal Component Analysis (PCA) was used for dimensionality reduction

# Data Pre-processing

Even 10 metrics might impact performance

Principal Component Analysis (PCA) was used for dimensionality reduction

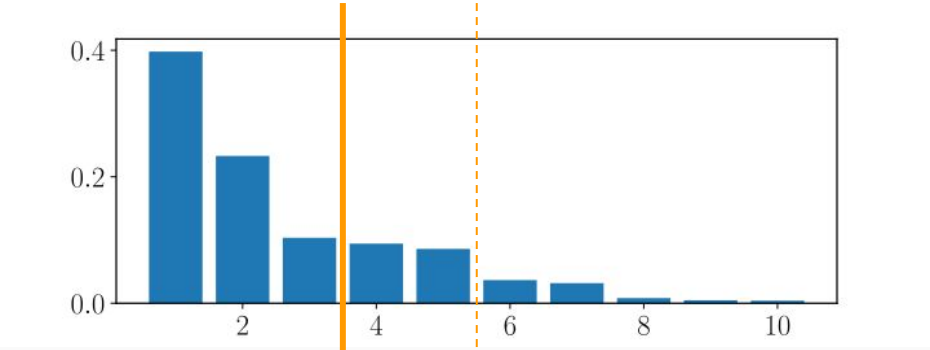


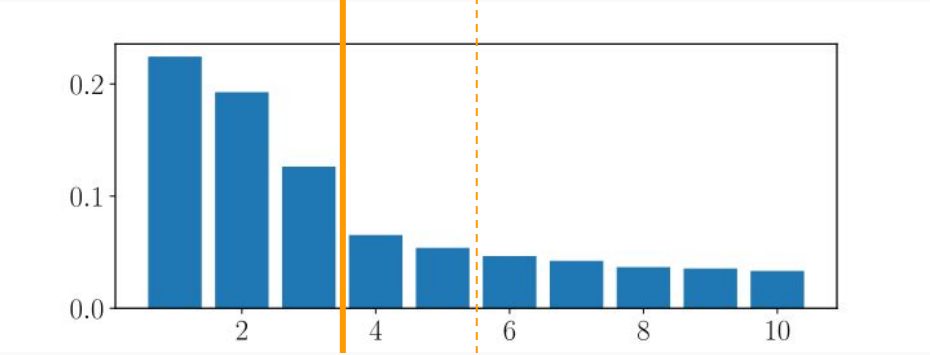



# Data Pre-processing

Even 10 metrics might impact performance

Principal Component Analysis (PCA) was used for dimensionality reduction



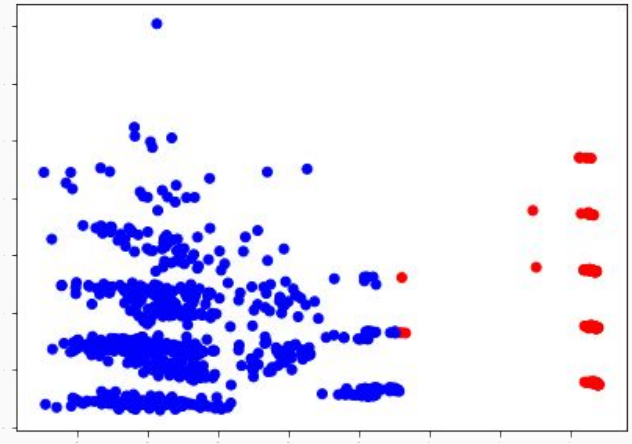
# 2a) Unsupervised Data Analytics

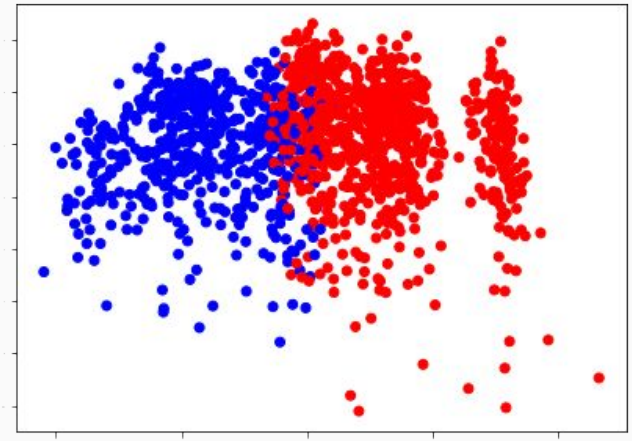
# Clustering

Clustering - Group data points together based on their similarity

# Clustering

Clustering - Group data points together based on their similarity





# Novelty Detection

Novelty Detection - Algorithm tries to find data points that don’t fit with the majority of the dataset

# Novelty Detection

Novelty Detection - Algorithm tries to find data points that don't fit with the majority of the dataset

| Model | Precision | Recall |
|---|---|---|
| Isolation Forest without PCA | 0.79 | 0.37 |
| Isolation Forest 3D PCA | 0.72 | 0.32 |
| One Class SVN without PCA | 0.29 | 0.16 |
| One Class SVN 5D PCA | 0.43 | 0.19 |

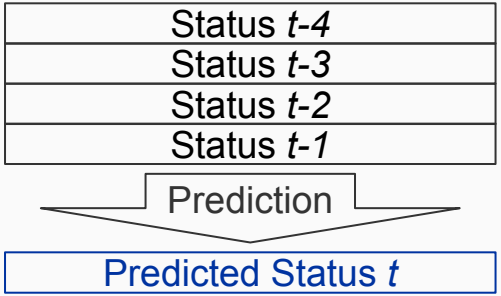
Precision = True_positives / Positives

Recall = True_positives / Anomaly Count

# 2b) Supervised Data Analytics

# Supervised Learning

Supervised Learning - Algorithm learns
$f: X \rightarrow y$
function from provided learning data



# Supervised Learning

Supervised Learning - Algorithm learns
$f: X \rightarrow y$
function from provided learning data

| | |
|---|---|
| Status *t-4* | Cluster Avg. Status *t-4* |
| Status *t-3* | Cluster Avg. Status *t-3* |
| Status *t-2* | Cluster Avg. Status *t-2* |
| Status *t-1* | Cluster Avg. Status *t-1* |

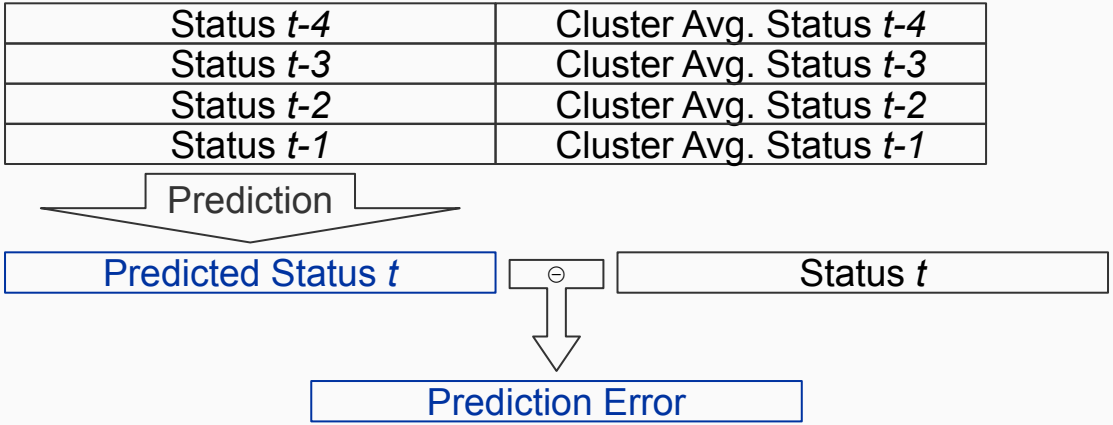

Predicted Status *t*

# Tested Algorithms

- Baseline
  - status *t* ← status *t-1*
- Linear Regression
  - status *t* ← linear combination of inputs
- Random Forest Regressor
  - status *t* ← combined output of many decision trees

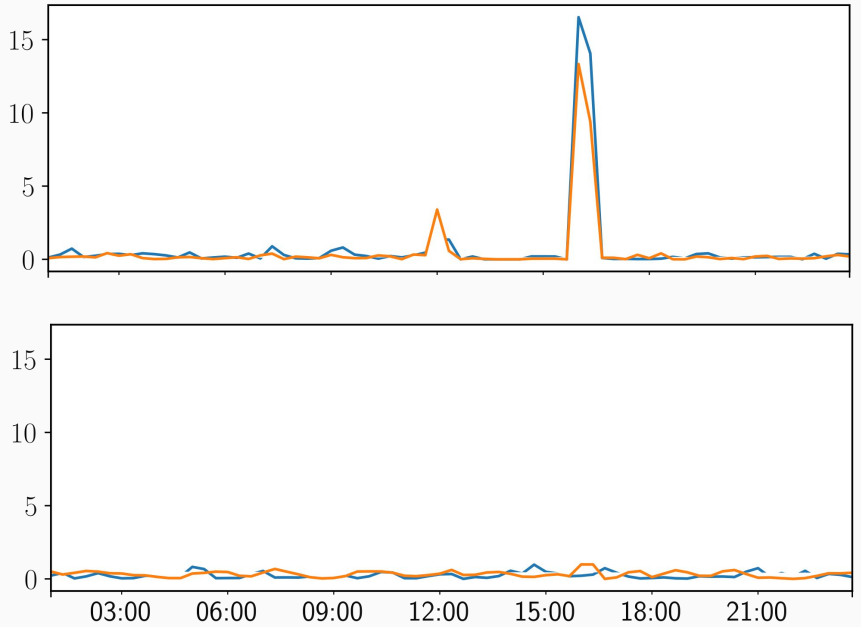
# Supervised Learning

Supervised Learning - Algorithm learns
$f: X \rightarrow y$
function from provided learning data



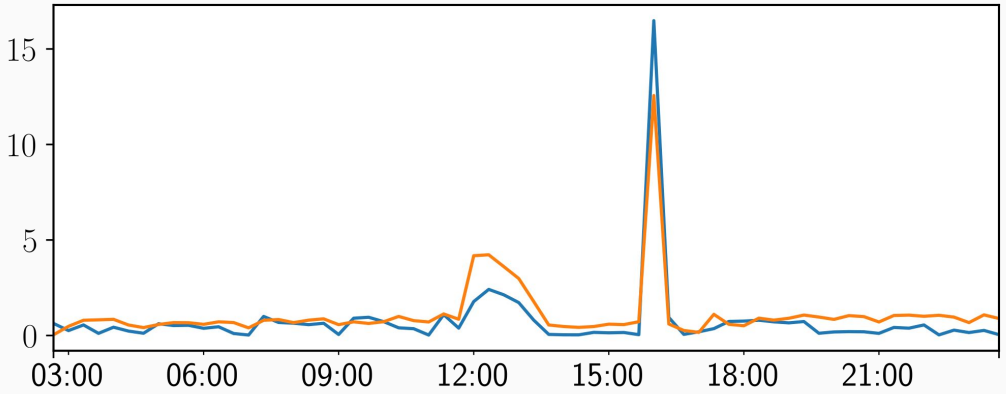
# Analysing Prediction Error

Baseline algorithm on a day with a service stop anomaly:
anomalous vs random host



# Analysing Prediction Error

Random Forest algorithm on a day with a service stop anomaly: anomalous host



# Anomalies from Error

anomaly(n) = `True` if *error(n) > mean_error + 3∗std_dev*
= `False` otherwise

# Analysing Prediction Error

Random Forest algorithm on a day with a service stop anomaly

| prediction time | hostname | anomaly |
|---|---|---|
| 12:00:00 | anomalous.cern.ch | TRUE |
| 12:20:00 | anomalous.cern.ch | TRUE |
| 12:40:00 | anomalous.cern.ch | TRUE |
| 13:00:00 | anomalous.cern.ch | TRUE |
| 16:00:00 | host01.cern.ch | FALSE |
| 16:00:00 | host09.cern.ch | FALSE |
| 16:00:00 | anomalous.cern.ch | TRUE |

# Analysing Prediction Error

Random Forest algorithm on a day with a service stop anomaly

| prediction time | hostname | anomaly |
|---|---|---|
| 12:00:00 | anomalous.cern.ch | TRUE |
| 12:20:00 | anomalous.cern.ch | TRUE |
| 12:40:00 | anomalous.cern.ch | TRUE |
| 13:00:00 | anomalous.cern.ch | TRUE |
| 16:00:00 | host01.cern.ch | FALSE |
| 16:00:00 | host09.cern.ch | FALSE |
| 16:00:00 | anomalous.cern.ch | TRUE |

# Improving precision

Count the anomaly only if the model has noticed it twice in a row

# Improving precision

Count the anomaly only if the model has noticed it twice in a row (also lower the error threshold)

| prediction time | hostname | anomaly |
|---|---|---|
| 12:00:00 | anomalous.cern.ch | TRUE |
| 12:20:00 | anomalous.cern.ch | TRUE |
| 12:40:00 | anomalous.cern.ch | TRUE |
| 13:00:00 | anomalous.cern.ch | TRUE |
| 16:00:00 | host01.cern.ch | FALSE |
| 16:00:00 | host09.cern.ch | FALSE |
| 16:00:00 | anomalous.cern.ch | TRUE |

# Improving precision

Count the anomaly only if the model has noticed it twice in a row (also lower the error threshold)

| model | recall |
|---|---|
| Random Forest without PCA | **0.78** |
| Random Forest 3D PCA | 0.7 |
| Linear Regression 3D PCA | 0.56 |
| BASELINE | 0.48 |

# Stats per Metric

Random Forest without PCA dimensionality reduction

| model | precision | recall |
|---|---|---|
| cpu idle | 1 | 0.85 |
| running processes | 1 | 0.73 |
| load | 0.99 | 0.68 |
| cpu system | 0.64 | 0.46 |
| cpu steal | 0 | 0 |
| stopped processes | 0 | 0 |
| used memory | 0 | 0 |
| cpu iowait | 0 | 0 |
| cpu interrupt | 0 | 0 |
| cpu nice | 0 | 0 |

# Future

Move to Production

- Automatize training
- Anomaly detection on streaming data
- Improving the stability of the streaming jobs

# Questions?

Special thanks to Luca Magnoni and the whole IT-CM-MM section at CERN

# Backup Slides

| id | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FRF | 0 | 0 | 0 | 0 | 0 | 0 | 0 | NaN | 9 | NaN | 0 | 0 | NaN | NaN | NaN | NaN | 0 | 6 | 7 | 7 | 6 | 7 | 6 | 6 | 7 | 6 | 6 |
| RF | 0 | 0 | 0 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 0 | 3 | 3 | 2 | NaN | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| LR | 0 | 0 | 0 | 0 | 12 | 12 | 0 | 1 | 2 | NaN | NaN | NaN | 12 | NaN | NaN | NaN | NaN | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12 | 12 |
| bas | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 0 | NaN | NaN | NaN | NaN | 1 | 0 | NaN | NaN | NaN | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |