Unsupervised Machine Learning Techniques for Anomaly Detection with Multi-Source Data

author: Filippo Pacinelli
supervisor: Prof.ssa Elisabetta Ronchieri

Department of Statistics, University Of Bologna

Second Cycle Degree in Statistical Sciences

II Academic Session, Year 2021/2022

October 25, 2022

Problem

This work deals with anomaly detection in the Data Center of INFN-CNAF Institute.

$+1000$ machines $\rightarrow$ 11 *log services* $\rightarrow$ *avg* $1M$ *message/machine*

$+1000$ machines $\rightarrow$ 18 *monit metrics* $\rightarrow$ *avg* $300K$ *values/metric*

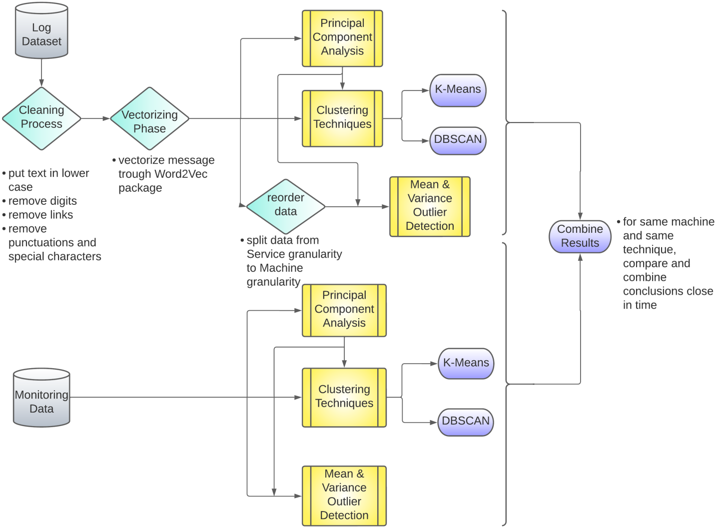
Main challenges:

- different types of data (textual and numerical)
- completely unsupervised task
- thousands of machines to analyze, therefore some automatic mechanism was necessary

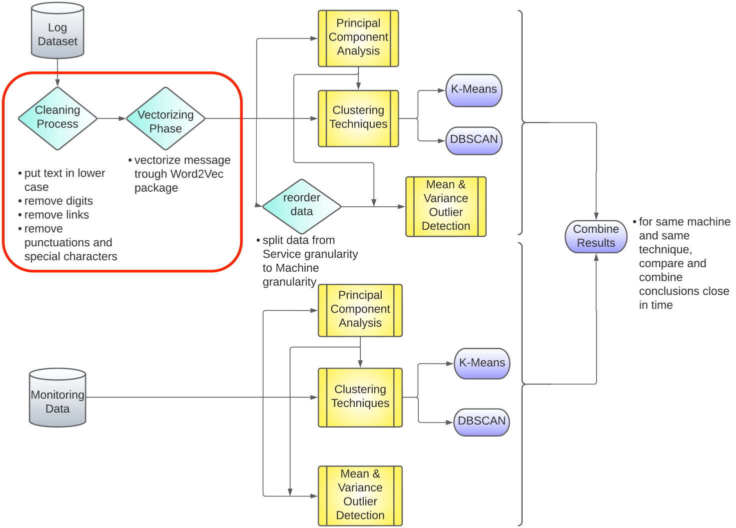
Data Available:

- **Log Data**: log messages of softwares running on the machines of the data center
- **Monitoring Data**: numerical sequences representing metrics to check the health status of machines.

Process Adopted



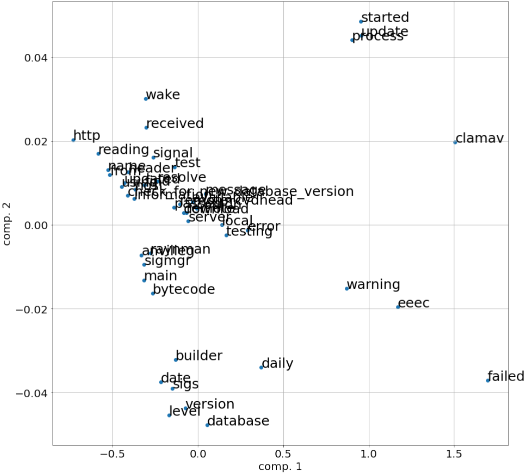
Process Adopted



PCA For Dimensionality Reduction

PCA was used to:

- reduce dimensionality
- as an anomaly detection method itself trough decomposition and reconstruction



Dimensionality Reduction, 2 components

PCA And Reconstruction Error

Algorithm 1 PCA Anomaly Detection with Reconstruction Error

```
Reduce dimensionality trough PCA
Define threshold τ
Using only the n principal components obtained, obtain an approximation of initial data trough transition matrix
Compute Reconstruction Error: R.E.
for i in R.E. do:
    if i ≥ τ then:
        i ← anomaly
    else:
        i ← regular
```

Rationale: Reconstruction error is larger for uncommon, so less observed observations

DBSCAN Clustering

Density based clustering method which constructs clusters from highly populated area of observations, close at most a value ε from each other

Log Data

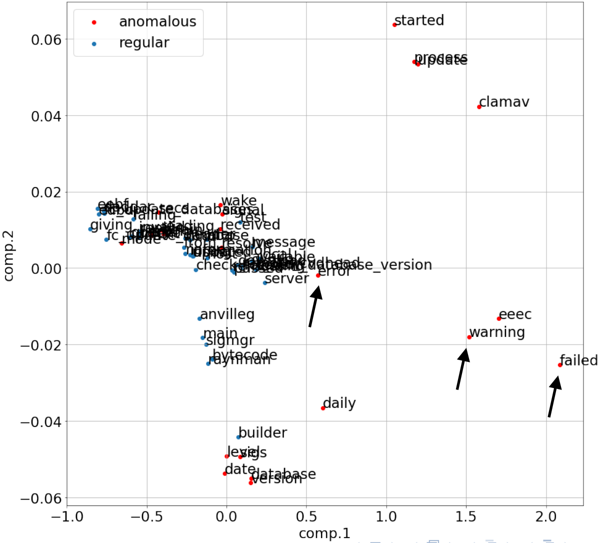
- ε obtained based on overall distances between word-vectors and parameter tuning
- Rationale: non-anomalous words are expected to be more and to concentrate closer between each other rather than potentially anomalous words.

Monitoring Data

- ε obtained by means of elbow curve and and parameter tuning
- same rationale

DBSCAN Clustering

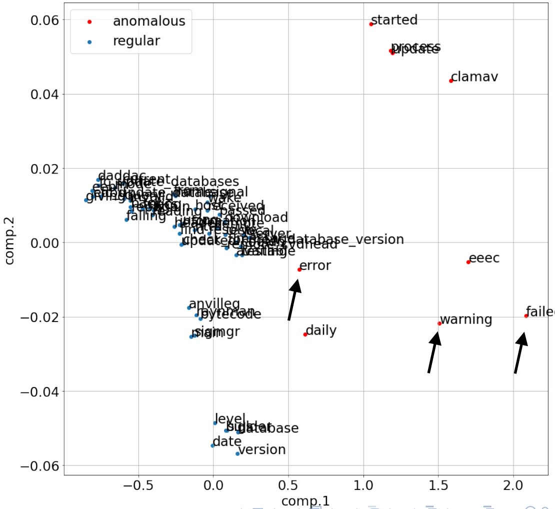
Specifically, serious anomalies correctly detected



K-Means Clustering

- clustering method based on centroids
- number of clusters obtained with parameter tuning
- rationale: expected large distances between anomalies and non-anomalies.

Same example, in particular more serious anomalous words correctly detected



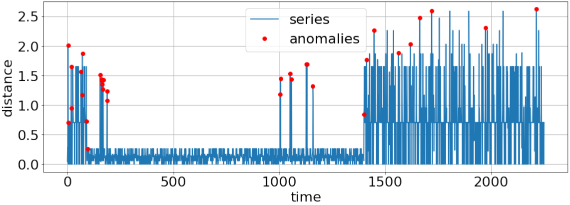
Mean & Variance Outlier Detection

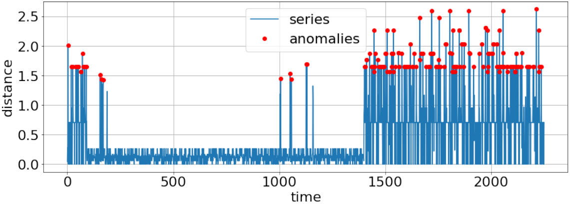
Algorithm 2 Mean & Variance Outlier Detection

get message-vectors by averaging all the word-vectors $\in$ message
get d = distances between consecutive message-vectors
if sliding windows = True **then**:
    d = d[sliding window]
else:
    d = d
end if
define threshold $\tau = \mu_d + k\sigma_d$ *
for i *in* **d do**
    if $i \geq \tau$ **then**:
        $i \leftarrow$ *anomaly*
    else:
        $i \leftarrow$ *regular*

Mean and Variance Outlier Detection

Anomaly Detection with sliding windows (above example) is more robust to change in patterns with respect to the algorithm without sliding windows





windows size = 20 and **tolerance** = 2 standard deviations on machine cloud-ctrl01

Validating Results

Log Data:

continuous anomaly score $\in [0, 1]$ given to log messages. Applied after PCA, DBSCAN and K-Means algorithms results.

- **PROS**: more robust and safe (e.g. misclassified words have less impact).
- **CONS**: less precise and no more binary, intuitive classification.

Monitoring Data:

Mann-Whitney U Test to check for significant differences between anomalous and non-anomalous classified observations, for every metric

- **PROS**: It checks for significant difference between potential anomalous and non-anomalous samples
- **CONS**: It doesn't provide information about the quality of classification

Anomaly Score Example

daily database available for update (local version: 26052, remote version: 26053)

$\Big\downarrow$ clean message

'daily', 'database', 'available', 'for', 'update', 'local', 'version', 'remote', 'version'

$$1 + 0 + 0 + 0 + 1 + 0 + 0 + 0 + 0 = 2/9 = 0.222$$

Process to compute the anomaly score of a message

Algorithm	dim. reduction	n. comps	n. clusters	window size	Epsilon	tolerance
DBSCAN	TRUE, FALSE	2, 3, 10, 20	*variable*	*N.A.*	0.05%, 0.1%, 0.25%, 0.5%	*N.A.*
K-Means	TRUE, FALSE	2, 3, 10, 20	2, 3	*N.A.*	*N.A.*	*N.A.*
PCA Decomposition & Reconstruction	*implicit*	2, 3, 10, 20, 30	*N.A.*	*N.A.*	*N.A.*	0.75, 0.85, 0.95
Time-Series Outlier Detection	*N.A.*	*N.A.*	*N.A.*	0, 10, 30, 60 (0 means no window partition)	*N.A.*	2, 3

Algorithms and Hyperparameters Overview

Conclusion & Future Developments

Conclusion

- this work is a step forward anomaly detection mapping in a data center
- different types of data were successfully combined together to build more robust conclusion
- automatic implementation was also the key to deal with huge amount of data

Future Developments

- hopefully in the future also some semi-supervised or even fully supervised techniques might be employed
- feed together in a model textual and numerical data
- identify different types of anomalies and not only a binary classification (the anomaly score is a step forward in this regard)

Thank you!