
CERN

Elasticsearch Anomaly detection

Elasticsearch Service in Numbers

- 36 clusters
- O(190) entry points/use cases
- 3 major ES version 5.x, 6.x, 7.x
- ~250 data nodes, ~180 master and search nodes
- ~400 TB SSD space
- 400000 indices, 400 Billion documents
- Up to 8 Mio accesses / h
- Use cases from all over the place

What is the problem we want to solve ?

- The issue
  - **Complexity** of the service makes it difficult to spot issues in time
  - **Elasticsearch health** itself is often not terribly helpful
    - Most complains while the cluster is perfectly green, such as poor performance, read-only indices, full disks, misbehaving users, ...
    - When it changes color it's often too late
    - When it's yellow it does not mean that the cluster is down
  - There are **plenty of internal metrics** and logs which can give hints for problems
    - Internal monitoring with many dashboards and plots of metrics for investigation
  - **Relevant information is often hidden**
- Extract the currently relevant information, and display it on a single and simple dashboard
  - Something that tells us what to look at first
  - Something that looks for **Anomalies**

So … what is an *Anomaly* ?

Something that deviates from what is standard, normal, or expected.

Source: google.com

So … what is an *Anomaly* ?

Something that deviates from what is standard, normal, or expected.

Source: google.com

- Not necessarily bad
- Not necessarily a problem
- Possibly interesting

… what is *standard, normal, or expected*?

Something that we have seen in the past

… what is *standard, normal, or expected*?

Something that we have seen in the past

- Compare the current status to the past
- That's what we do when we look at our dashboards: we compare the current status with the past
- No need for a PhD for that …
- Is that something we can teach our computers to do for us ?

… what is *standard, normal, or expected*?

Something that we have seen in the past

- Compare the current status to the past
- That's what we do when we look at our dashboards: we compare the current status with the past
- No need for a PhD to do that …
- Is that something we can teach our computers to do for us ?

… what is *standard, normal, or expected*?

Something that we have seen in the past

- Compare the current status to the past
- That's what we do when we look at our dashboards: we compare the current status with the past
- No need for a PhD to do that
- Is that something we can teach our computers to do for us ?

… what is *standard, normal, or expected*?

Something that we have seen in the past

- Compare the current status to the past
- That's what we do when we look at our dashboards: we compare the current status with the past
- No need for a PhD to do that …
- Is that something **we can teach our computers to do** for us ?

YES, we can !

Source: Obama

Elasticsearch Anomaly detection in practice

History:

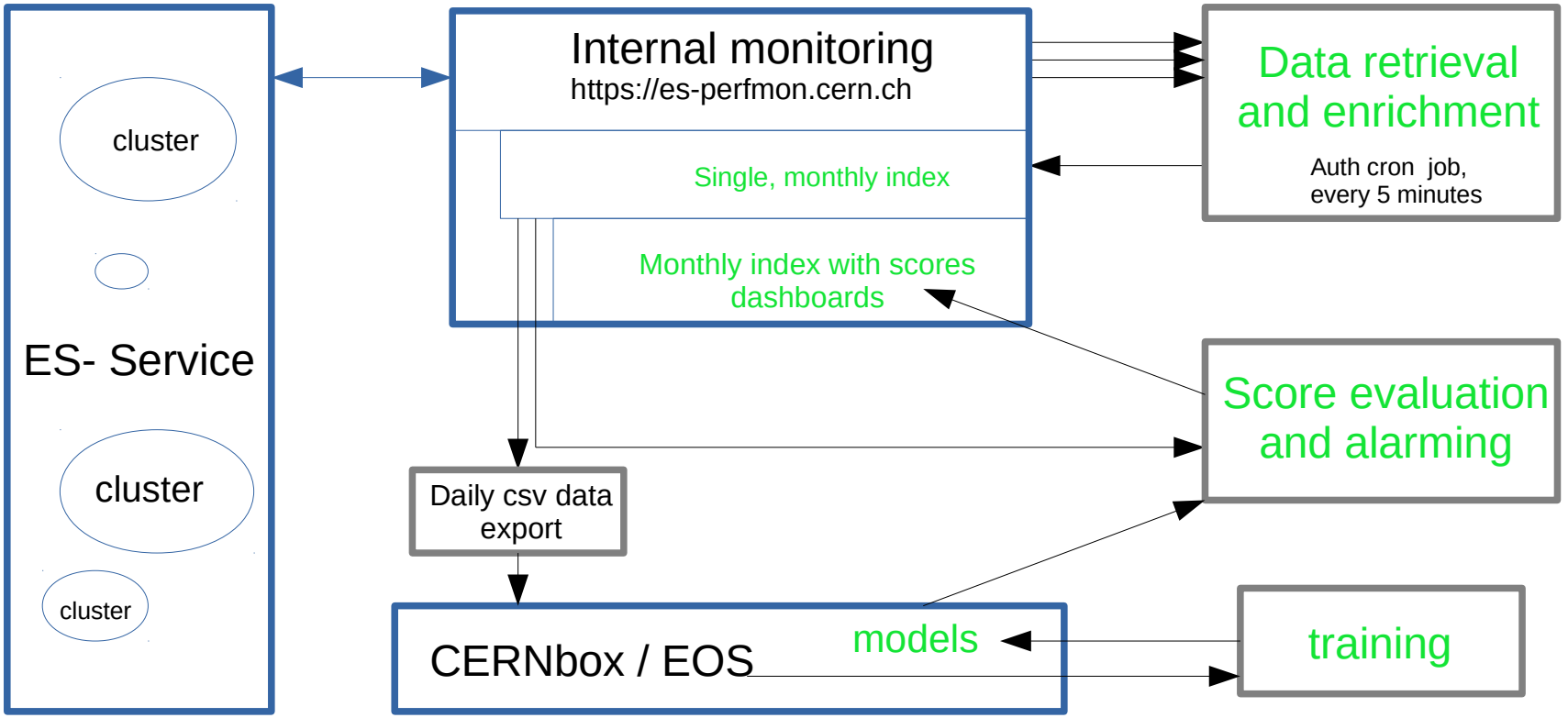
- Technical student Jose Alonso, 2017
  https://indico.cern.ch/event/687877
- Summer student Jennifer Anderson, summer 2019
  https://indico.cern.ch/event/836289/
- Myself … eg. this talk (see references at the end of the talk)

Disclaimer: Not going into too many technical details here.

The short of it

- Data retrieval:
  - Sample metrics every 5 minutes on all nodes and clusters of the service
  - Use of service metrics, node health information, logfile entries
- Analysis
  - Time series anomaly detection using a one layer **LSTM network** and a **moving average** approach
  - Use 288 previous time steps (so 1 day of data), predict the next sample and compare with reality
- The mean squared error is used as anomaly score

Mechanics


cluster
ES- Service
cluster
cluster
Internal monitoring
https://es-perfmon.cern.ch
Single, monthly index
Monthly index with scores
dashboards
Data retrieval
and enrichment
Auth cron job,
every 5 minutes
Score evaluation
and alarming
Daily csv data
export
CERNbox / EOS
models
training

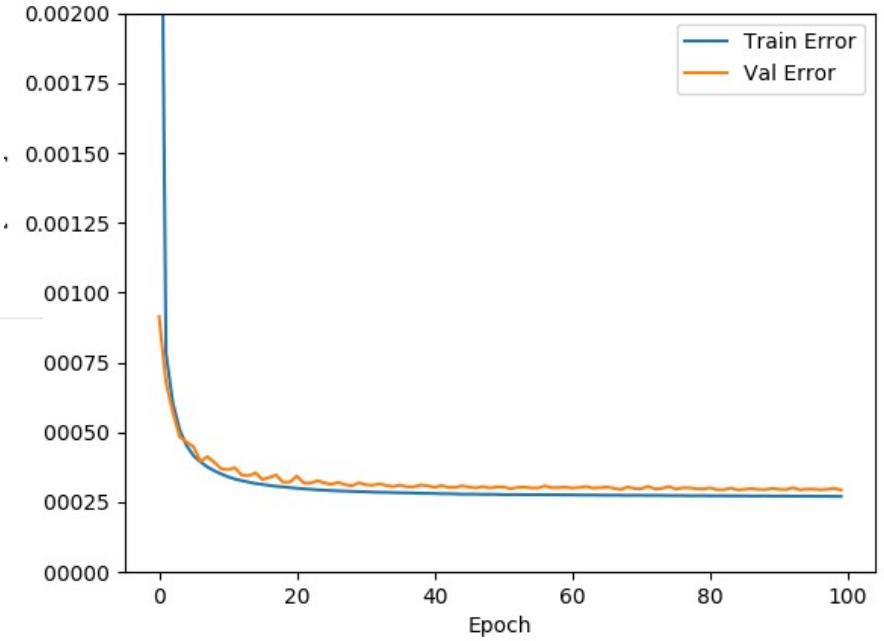
NNW model

(Time-)steps: 288 (1 day of data)
Training size/validation size = 4/1
Training data from **all clusters**

Model layout

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 148)               175824
=================================================================
Total params: 175,824
Trainable params: 175,824
Non-trainable params: 0
```

Mean squared error



Remarks:

- Using Keras/Tensorflow 2.1.0
- Python3
- CuDNNLstm implementation used for training
- converted into standard LSTMs after training

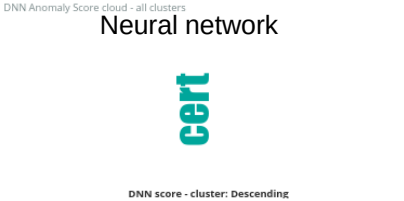
- Training time: ~3-4min/epoch (TESLA V100, RTX-2080 GPU)
- About 612k distinct time samples for training and validation

Moving average

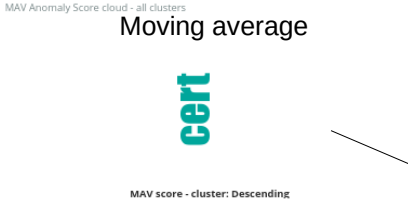
- Moving average calculation
  - For each feature take the average over the past N-1 samples as a prediction for the current value
  - Uses exactly the same pre-processing pipeline as for the ML data
  - Calculate the anomaly score between predicted and actual data
- Non ML, simple and fast
  - No need for any training
- Monitors different things than the network based approach
  - Looks only at a single cluster only
  - Complements the Network approach
  - Differences most visible in the classification step rather than in the score itself

Monitoring dashboards

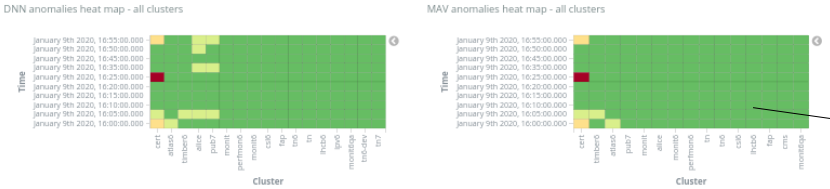
Neural network



Moving average

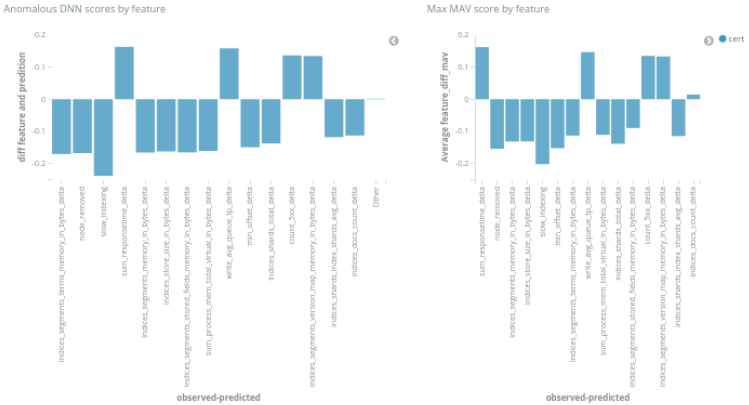


Cloud: Clusters with highest scores



1h history, ordered by cluster score

Scores above a threshold



Delta for features with highest scores, for classification

Upgrade of CERT cluster OTG0054074

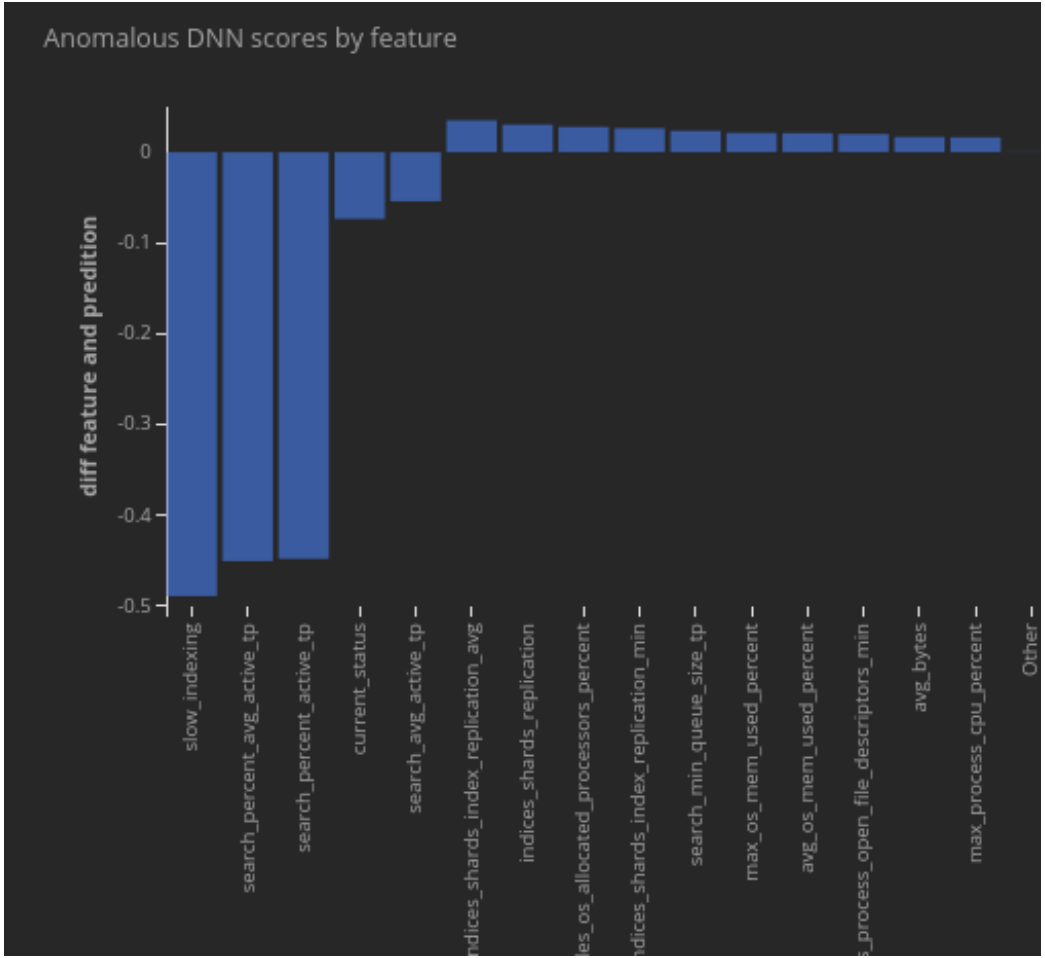
Classification example

Observation:
- Anomaly in once cluster (monit6) at 10am
- OK recovered after 5min
- Confirmed by MAV approach

Feature values observed – predicted,
Ordered by the absolute value of the difference.

Several things are happening here:
- The cluster is slow
- There is a search ongoing
- The status changed (yellow to green)

Diagnostics:
A heavy search and recent indexing activity



Summary

- Released Anomaly detection for the ES service
  - Based on ML and moving average approach
  - Very few false positives
  - Some false negatives, being addressed as they are seen
  - Proven useful in daily operation
- Todo:
  - Code cleanup, optimisation and beautification ...
- Side effects
  - Feedback for GPU resources in batch
  - Steep learning curve

media.ccc.de

Search...

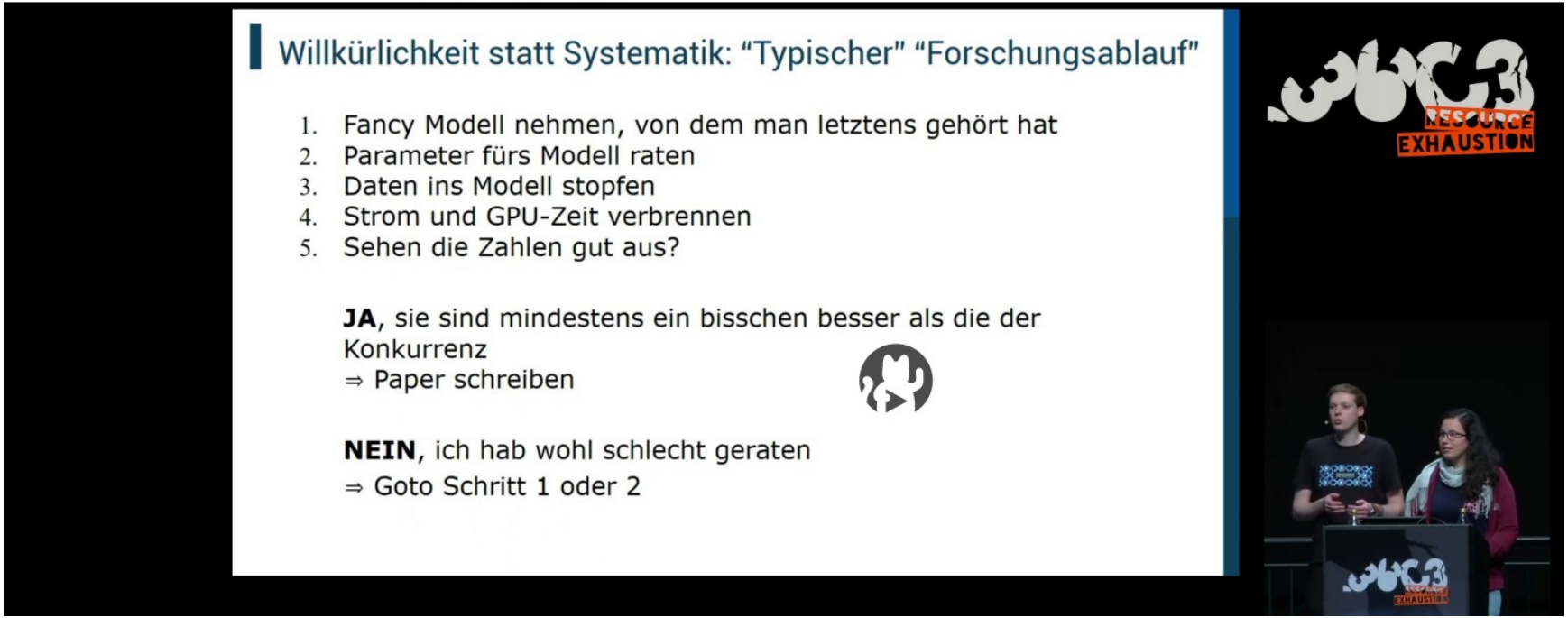
browse › congress › 2019 › event

https://media.ccc.de/v/36c3-11006-der_deep_learning_hype

Der Deep Learning Hype

Wie lange kann es so weitergehen?

Nadja Geisler and Benjamin Hättasch



References

- Previous presentations
  - Original approach: https://indico.cern.ch/event/687877
  - New approach: https://indico.cern.ch/event/830947/
  - LSTMs: https://indico.cern.ch/event/836289/
  - Update: https://indico.cern.ch/event/851356/
  - Final approach: https://indico.cern.ch/event/879832/
- Code
  - https://gitlab.cern.ch/it-elasticsearch-project/es-ml

Questions and discussion