Anomaly Detection for Predictive Maintenance in Data Centers Using Autoencoders

Dr. Marco Lorusso [1]

[1] National Institute for Nuclear Physics - CNAF Division

19th March 2025

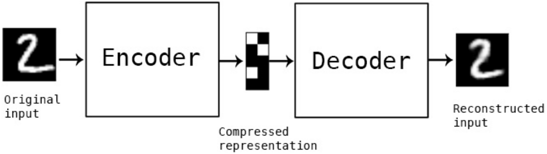
Who am I?

- Phd Student in Physics at University of Bologna, Italy;
- Focused on Software and Computing:
  - Application of Artificial Neural Networks for HEP
  - Optimization of ANNs for low-latency applications;
- Recently started a Post-Doc position at CNAF:
  - Computing division of the Italian Institute for Nuclear Physics (INFN)

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Autoencoders

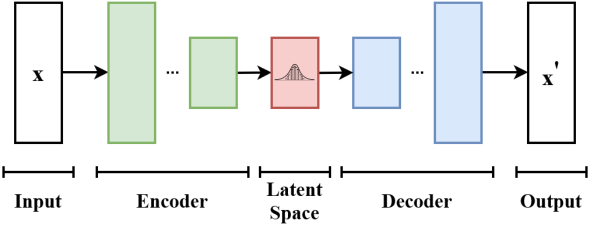
- Autoencoder (AE): a NN that is trained to attempt to copy its input to its output;
- Made up of two parts: an Encoder and a Decoder that produces a reconstruction of the input.



- Designed to be unable to learn to copy perfectly but only approximately, prioritizing useful properties of the data;
  - Usually done by constraining the latent space to have a smaller dimension than the input (Undercomplete);
- Widely used for dimensionality reduction, feature learning and generative models.

Making them Variational

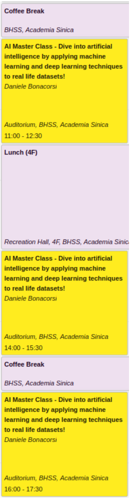
- AE that aims to learn a **compact**, **continuous** latent representation of data;



- **Goal**: learn encoder and decoder parameters such that new samples can be generated from the learned latent space through the decoder;
- VAE uses a probabilistic encoder that maps the input to a probability distribution over the latent space.

AI Masterclass

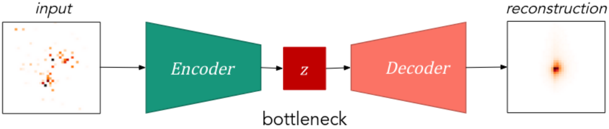
- This is just a niche example of the vast zoo of AI models;
- During this conference a Masterclass on AI is being held:
  - Hope you didn't miss it if you want to know what there is behind AI models and how they work!
- Today: Wednesday 19/03 (or 03/19);
- From 11:00 to 17:30 (11AM to 5:30PM).
- Lecturer: Prof. Daniele Bonacorsi (Full Professor @ University of Bologna;
  - Tutors: Dr. Marco Lorusso, Dr. Luca Giommi and Dr. Simone Gasperini.



Anomaly Detection with AEs

Auto-Encoders (AEs) are neural network models trained to **reconstruct** their **inputs**:

- The **encoder compress** the input into a smaller latent space.
- The **decoder reconstructs** from the compressed representation.



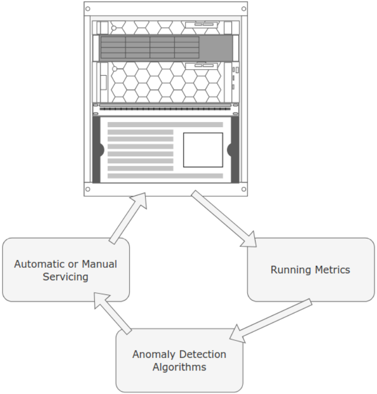
- Idea: **background** samples are associated to **low** anomaly scores;
- Anomalies can be either erroneous, rare or interesting events;
- This approach is **self-supervised**, requiring only background samples in the data;
- **No assumption** on the type of signal.

AD in HEP

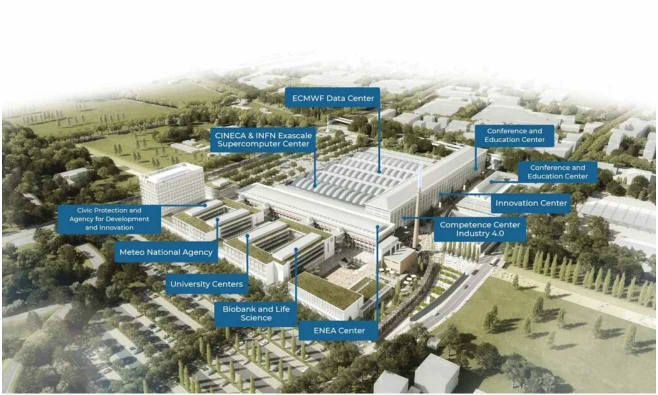
- **Machine Learning** strongly considered to process the **increased** amount of **data** in next phases of LHC $\rightarrow$ major focus on **Artificial Neural Networks**;
- An important candidate for using ML is the research for **Beyond Standard Model** events:
  - Networks like **Autoencoders** are **unbiased** algorithm which can **select** events based on their degree of **abnormality**, **without** theoretical **priors**;
- Trigger **latency and energy constraints** are quite **unique** $\rightarrow$ need for **specific software** development and strategies to deploy ANN **efficiently** on the **hardware** available on-site, like **FPGAs**.
- Need to **optimize** and **compress** these kind of algorithm to make them **suitable** for **trigger** environments.
- Topic of my PhD thesis and of a talk tomorrow at 2PM.

Moving to Datacenters

- The idea is to use Autoencoders to perform Anomaly Detection using sensor data from a datacenter:
  - This could lead to a system able to pick up a malfunction before major or even any disruption of the service
- Particularly interesting in Bologna after the construction of the Tecnopolo!



Bologna's Tecnopolo



- It houses Leonardo, the 9th SuperComputer in the top 500 (Nov. 2024), with 241.20 PFlop/s!;
- But also the new and upgraded CNAF Datacenter.

The Use Case

Bucket Name	Data frequency	Retention Policy
one_week	5 minutes	1 week
one_month	15 minutes	1 month
six_month	2 hours	6 months
all_data	3 hours	inf

- Service under study is WebDAV: it enables to access and manage files remotely via HTTP;
- To maintain seamless operation:
  - → essential to detect anomalies indicating potential service disruptions.

- ML to analyze system metrics—such as CPU usage, memory consumption, and network traffic—using a Time Series dataset;
- Data extracted from an InfluxDB v2 database

Metrics Available

- General CPU utilization metrics;

cpu.ctxt
cpu.total.system
cpu.total.user
cpu.procs_blocked
cpu.total.nice
cpu.total.idle
cpu.total.iowait
cpu.procs_running

- IO operations metrics;

iostat.avg-cpu.pct-idle
iostat.avg-cpu.pct-system
iostat.avg-cpu.pct-user
iostat.avg-cpu.pct-iowait
interface.eth0.rxBytes

- Used/Free/Swap memory metrics;

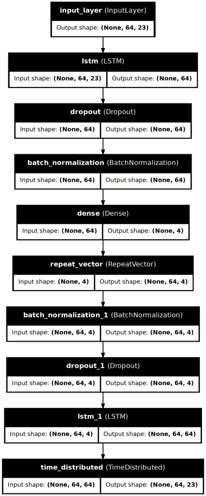
memory.available
memory.used
memory.cached
memory.buffers
memory.free
memory.used
WOBuffersCache
memory.free
WOBuffersCache

Identifying Anomalies

- Searched for ‘None‘ values in the dataset as Anomalies;
- Three main families of metrics with holes in the data:
  - Two dealing with comunication with the machines;
  - One concerning JVM metrics.
- In the 4 machines studied (*one_month* bucket) these sum up to:
  - 30 anomalies;
  - 36 anomalies;
  - 34 anomalies;
  - 42 anomalies.

The Neural Network

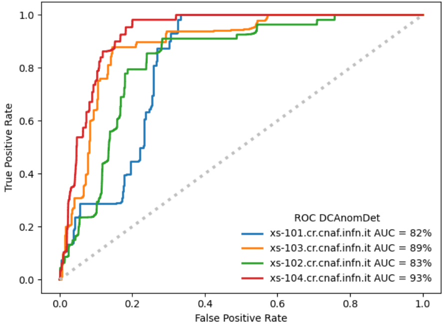
- To test the idea, an AE was built:
  - Using Long-Short Term Memory layers to deal with Timeseries;
  - Dropouts and Normalization Layers to increase generalization;
  - The Model infers a reconstruction of the input series;
  - RMSE as Loss function to measure the difference between input and output
- In total $\approx$ 42000 trainable parameters.



Anomaly Detection Results

- Model trained using timeseries w/o anomalies coming from 4 different machines;
- Test series with both anomalies and normal entries;
- Acceptable TPR;
- **But** still too sensible to False positives.

	AUC	TPR @ FPR = 0.2
xs-101	81.8%	0.45
xs-102	83.0%	0.80
xs-103	89.0%	0.88
xs-104	92.9%	0.95

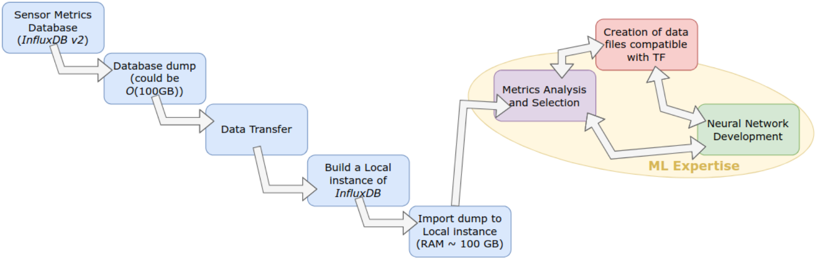


Challenges in Anomaly Detection with ML for Data Centers

- **Limited Availability of Anomaly Data**
  - Data centers already available to the public **should** have few anomalies.
- **Insufficient Retained Data**
  - With *one_week* or *one_month* retention policy there is not enough data to build a complex model with a good false positive rate (FPR);
  - Still valid for six months of data or more, compounded by the two or more hours timestep that is too sparse to be useful in online applications;
- **Need for a Realistic Machine Learning Approach**
  - Handling denser data could be very challenging but maybe necessary for realistic ML applications;
  - Suggested solution: Create a temporary database for training purposes.

Going from Datacenters to ML development

- Big computing infrastructures use very complex and useful database management systems for handling sensor data;
- However making data coming from these systems usable with ML frameworks (e.g. TensorFlow) is not so straightforward.



Conclusions

Key Takeaways:

- **Autoencoders** provide a powerful tool for **unsupervised** anomaly detection;
- Can be applied in **High-Energy Physics** (HEP) and **Data Centers** to detect rare or erroneous events;
- Still a sub-optimal **false positives** rate, due to **data scarcity**.

Future Directions:

- Bridge the gap between ML developers and dense highly structured data;
- Develop strategies for handling **denser, high-frequency data** for more intense ML development;
- Improve model robustness by incorporating additional **feature engineering**;
- Create more complex and generalized NNs (like VAEs to be able to generate other samples).

Thank you