ADMON
Anomaly Detection for MONIT data

ML Workshop

Nikolay Tsvetkov

10.03.2023

Anomaly Detection

- **ML based technique for detecting data pattern anomalies**
- **Very useful for monitoring data**
  - Applicable on time-series metrics and/or logs
  - Allow correlation of different datasets
  - Help to identify misbehaviours
  - Decrease the reaction time

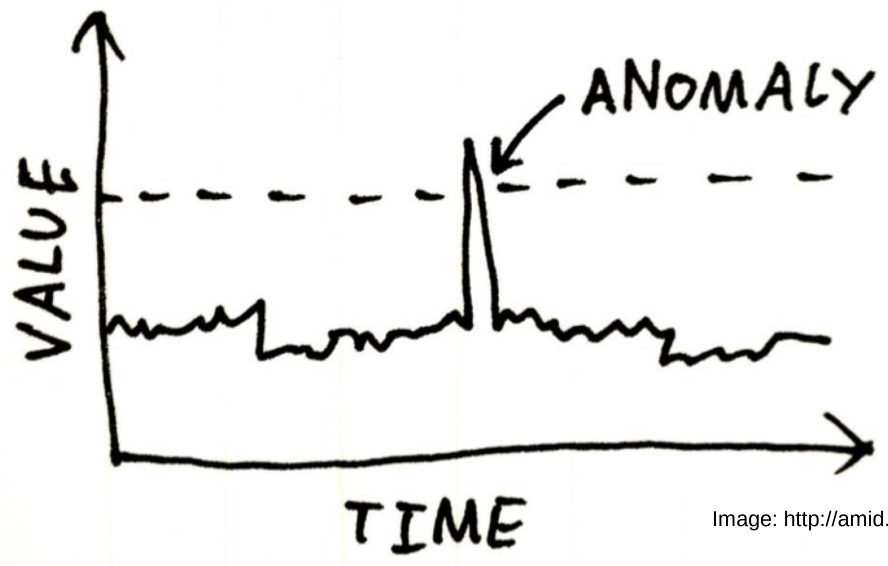


Image: http://amid.fish

Leads to prevented failures and improved reliability!

ADMON Motivation and Objectives

Improve the monitoring experience

- Allow users and Service Managers to detect and prevent outages as early as possible
- Improve the performance and stability of services by improving their monitoring

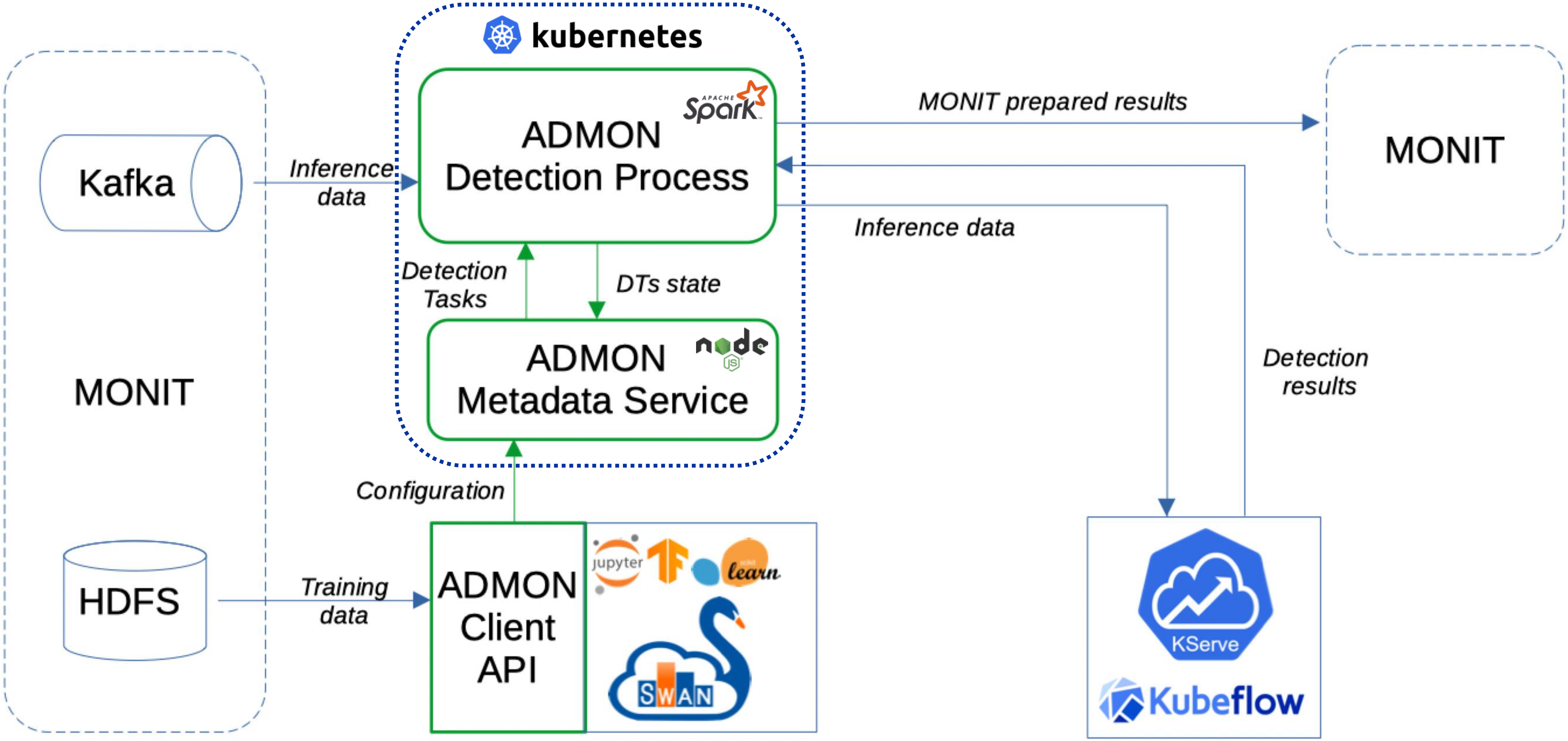
Make AD widely accessible

- Provide common infrastructure for processing AD models for IT Monitoring (MONIT) data
- Simplify the access to fresh IT Monitoring data and generate results on recent events
- Export the AD results to the IT Monitoring infrastructure

Consolidate ongoing work and efforts within IT / WLCG

- Integrate already available tools and services provided within CERN IT
- Reduce the overhead of building and maintaining custom ML infrastructure and tools
- Set ground for sharing know-how on already developed and proven algorithms and models

Architecture



Input Data Processing

1. Filter, Select, Rename

E.g.: filter for type 'msg'

time	col a	host	type
0	2	b	msg
1	1	a	msg
3	3	a	msg
3	4	b	msg
5	2	a	msg
6	4	b	msg
7	6	a	msg

time	col b	host	type
0	3	a	msg
1	7	b	msg
2	3	a	msg
4	3	b	msg
5	9	b	msg
7	3	a	msg
7	5	b	msg

2. Join over time window

E.g.: Window of size 5 with average

time	col a	host
0	2	a
0	3	b
5	4	a
5	4	b

time	col b	host
0	3	a
0	5	b
5	3	a
5	7	b

time	col a	col b	host
0	2	3	a
0	3	5	b
5	4	3	a
5	4	7	b

3. JSON payload

E.g.: Payload of host "a"

```
{
  metadata: {
    host: a
  },
  data: [
    {
      time: 0,
      col a: 2,
      col b: 3
    }, {
      time: 5,
      col a: 4,
      col b: 3
    }
  ]
}
```

ADMON API

1.Create *SourceConfig*

```python
from admonapi import SourceConfig

sc_gled = SourceConfig(
    "collectd", "raw", "monitoring",
    select=["timestamp", "value", "host"],
    filter_expression="topic=='xrootd_raw_gled'",
    rename={"value": "gled_value"}
)
sc_alice = SourceConfig(
    "collectd", "raw", "monitoring",
    select=["timestamp", "value", "host"],
    filter_expression="topic=='xrootd_raw_alice'",
    rename={"value": "alice_value"}
)
```

2.Create *InputDataConfig*

```python
from admonapi import InputDataConfig

input_data_config = InputDataConfig(
    source_configs = [sc_gled, sc_alice],
    agg_interval_seconds = 300,
    agg_method = "avg",
    group_by = ["host"]
)
```

3.Load data from HDFS

```python
from admonapi import DataSource

data_source = DataSource(spark, input_data_config)
data_frame = data_source.read_hdfs(start_timestamp, end_timestamp)
```

Result document


```json
{
    {
  "metadata": {
    "host": "monit-kafkay-1182c933d6.cern.ch",
    "agg_interval_seconds": "300",
    "agg_method": "avg"
  },
  "data": [
    {"timestamp": 1652169600000, "gled_value": 1404320.02, "alice_value": 16440.78},
    {"timestamp": 1652169900000, "gled_value": 1189512.86, "alice_value":    35.42},
    {"timestamp": 1652170200000, "gled_value":   11370.35, "alice_value":    42.97},
    {"timestamp": 1652170500000, "gled_value":   10336.88, "alice_value":    35.98},
    {"timestamp": 1652170800000, "gled_value":   11548.64, "alice_value":  1297.48},
    {"timestamp": 1652171100000, "gled_value":   11200.05, "alice_value":    33.63},
    {"timestamp": 1652171400000, "gled_value":   11147.68, "alice_value":    48.03},
    {"timestamp": 1652171700000, "gled_value":   13309.60, "alice_value":    33.78},
    {"timestamp": 1652172000000, "gled_value":   12769.24, "alice_value":    48.07},
    {"timestamp": 1652172300000, "gled_value":   13392.86, "alice_value":    33.79},
    {"timestamp": 1652172600000, "gled_value":   13040.87, "alice_value":    47.94},
    {"timestamp": 1652172900000, "gled_value":   14044.85, "alice_value":    33.49},
  ]
}
```


Users can use these data for developing their model

- The same schema will be received for model inference
- Further transformation functions can be applied by the user

ADMON API

4.Create *Project* and *DetectionEntity*

```
project = Project.create(
    title="XRootD Anomaly Detection",
    project_url="https://admon.docs.cern.ch",
    description="Anomaly detection on XRootD data based on correlation.",
    is_private=False,
    egroup="admon-dev"
)

de = DetectionEntity.create(
    project=project,
    title="XRootD Anomaly Detection for 1 hour intervals",
    interval_minutes=60,
    sliding_interval_minutes=15,
    input_data_config=idc,
    inference_model="xrootd-model",
    inference_namespace="admon-dev",
    monit_producer="admon",
    monit_label: {"admon_entity": "xrootd_with_join"}
)
```

ADMON Docs:
(https://admon.docs.cern.ch/)

Final step: Create *InferenceService* with *Transformer* in Kubeflow

- Build Docker image containing your transformation functions
- Train and store prediction model in S3

Project summary

- **Simplifies feature engineering and developing AD models**
  - Integrates the MONIT HDFS storage through Python API in SWAN
  - Provides aggregation of multiple data sources into a single dataset
- **Automates the model inferencing using the provided configuration**
  - Removes the effort of developing and maintaining own ML pipelines
  - Based on standard IT tools (SWAN, Kubeflow, IT Monitoring)
- **Applies on fresh MONIT data and sends results back to MONIT**
  - Allows earlier detection of potential problems
- **Scalable infrastructure able to cover more load in case of demand**
  - Spark based process running in Kubernetes cluster
- **Standard API allows sharing configurations between Service Managers**
- **Project has been completed and ready-to-use infrastructure is archived**

Thank you !

Q & A

home.cern

Application: United Nations Satellite Centre (UNOSAT)

Edoardo Nemni, Data Scientist

Taoyuan Liu, ML Trainee

10/03/2023, CERN IT Machine Learning Infrastructure Workshop

unitar
United Nations Institute for Training and Research

UNOSAT
United Nations Satellite Centre

Background of damage assessment from satellite imageries

- Accurate information about the extent of building damage is essential for **humanitarian relief** and **disaster response**
- **Application**: urban planning, population and growth estimation, damage assessment, etc…
- Multi-temporal (pre- and post-) high-resolution satellite images can be used but there are complex challenges:

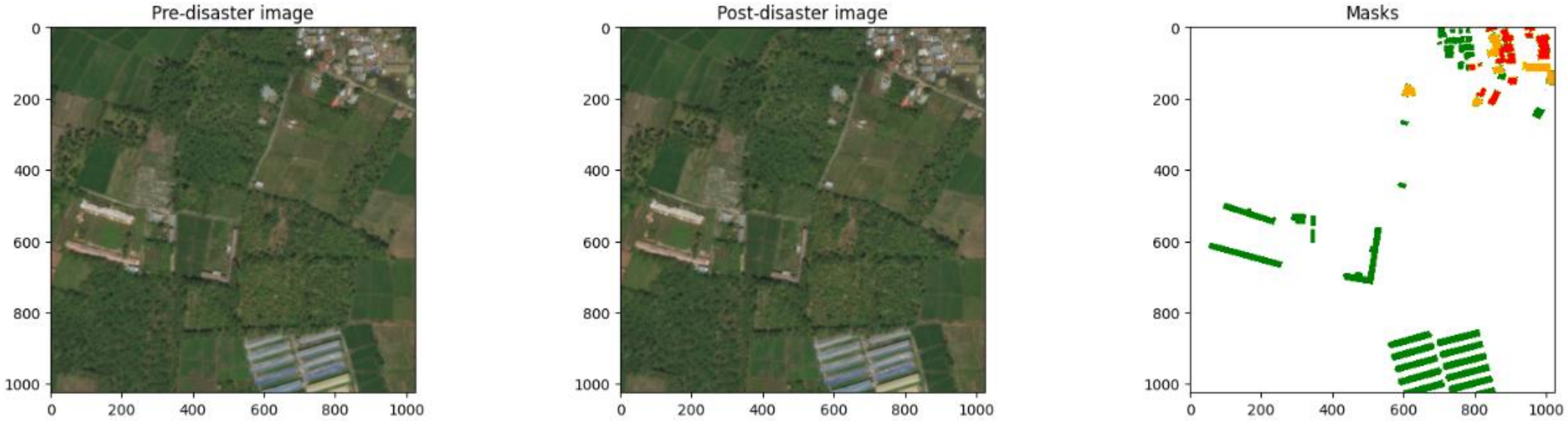
Earthquake

Flood

Tsunami

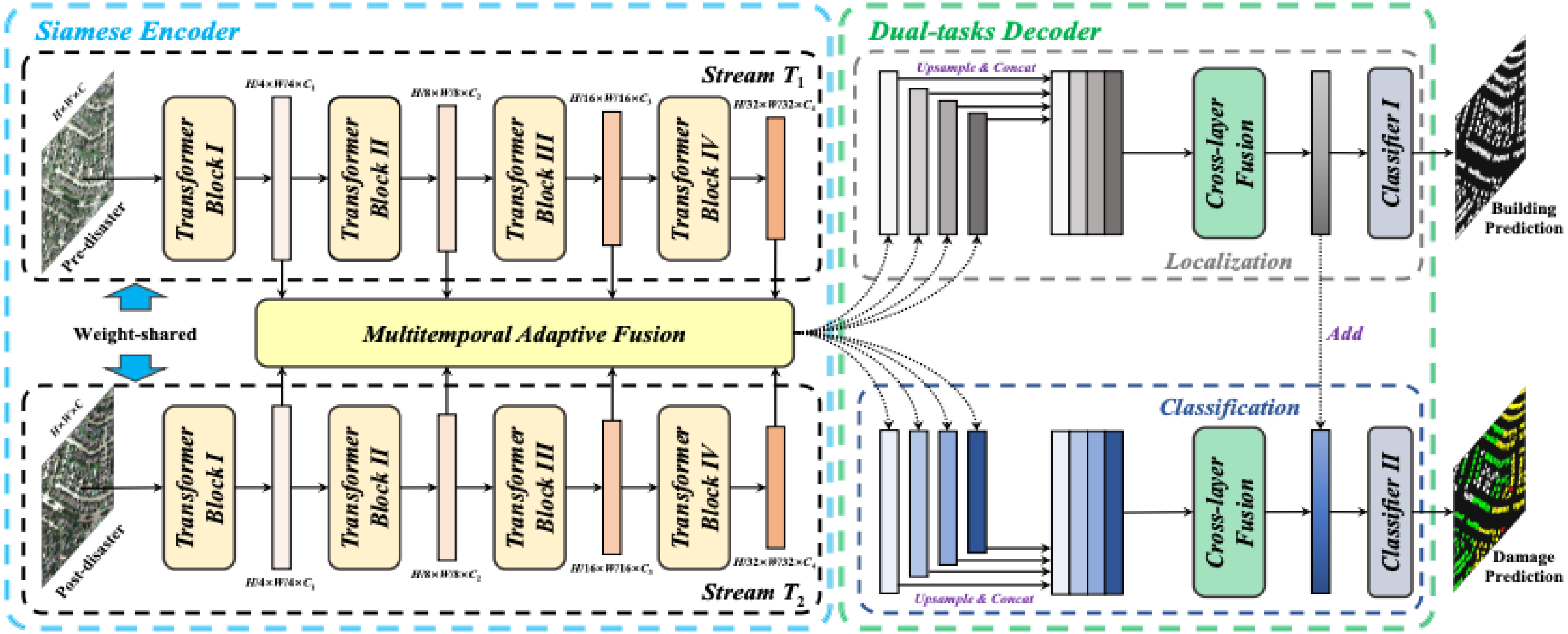
Wildfire

Data

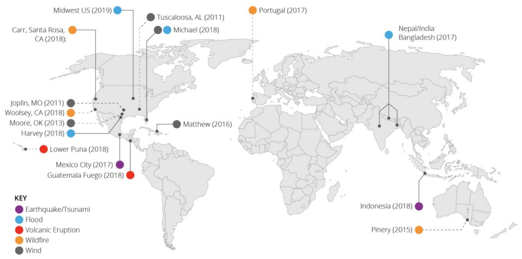


Score	Label	Visual Description of the Structure
0	No damage	Undisturbed. No sign of water, structural damage, shingle damage, or burn marks.
1	Minor damage	Building partially burnt, water surrounding the structure, volcanic flow nearby, roof elements missing, or visible cracks.
2	Major damage	Partial wall or roof collapse, encroaching volcanic flow, or the structure is surrounded by water or mud.
3	Destroyed	Structure is scorched, completely collapsed, partially or completely covered with water or mud, or no longer present.

UNOSAT DamFormer Architecture

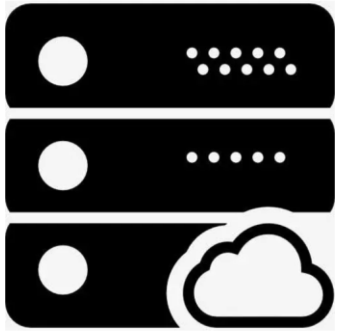


Training



xBD Dataset

Deploying



CERN Server

https://arxiv.org/pdf/2201.10953.pdf

Requirement

- Batch Size: 8
- GPU memory usage: 28 GB
- GPU required: 2*T4 or 1* V100s
- Data transfer worker: 4
- Training time required per epoch: 1 hour
- Epochs required for global best convergence: 100+

ml.cern.ch positive experience

CERN openlab UNOSAT

- High availability: Kubernetes supports high availability and self-healing. If a container or node fails, the system can automatically restart or migrate containers to keep the application available.

- Elastic scaling: Using a Kubernetes cluster makes it easy to scale compute resources for training tasks to meet training needs of different sizes without having to manually manage resources.

- Unified management: Using a Kubernetes cluster allows you to unify the management of different types of containers and applications, thereby improving management efficiency and reducing complexity.

ml.cern.ch challenges

- **Availability** of ml.cern.ch
- Max **idle time** < 24h
- **Network communication problems**: Network communication problems can arise when the cluster suddenly loses connection to /eos, leading to data read errors and program interruptions. To mitigate this issue, we have implemented a try/except method to read data that allows for a buffer margin in the event of a reading failure.
- **Pods communication problems**: For a multi-pods tasks, if a pod experiences an error, it can cause all other pods to pause and wait for the faulty pod to reconnect and resume training. However, the current distributed training initialization method of NCCL+:/env (the default method used in the CERN cluster) can cause the error pod to be unable to determine its own rank number after reconnection. This is because the rank number is randomly assigned during initialization. To address this issue, we suggest recording the rank number of initialized pods or modifying the initialization method in the code.

@UNOSAT

@UNITAR.unosat

/UNOSAT

unitar
United Nations Institute
for Training and Research

UNOSAT
United Nations
Satellite Centre

UNOSAT, United Nations Institute for Training and Research (UNITAR)
7 bis, Avenue de la Paix, CH-1202 Geneva 2, Switzerland

T +41 022 917 4720
E unosat@unitar.org
www.unosat.org

Cloud Anomaly Detection

D. Giordano (CERN)

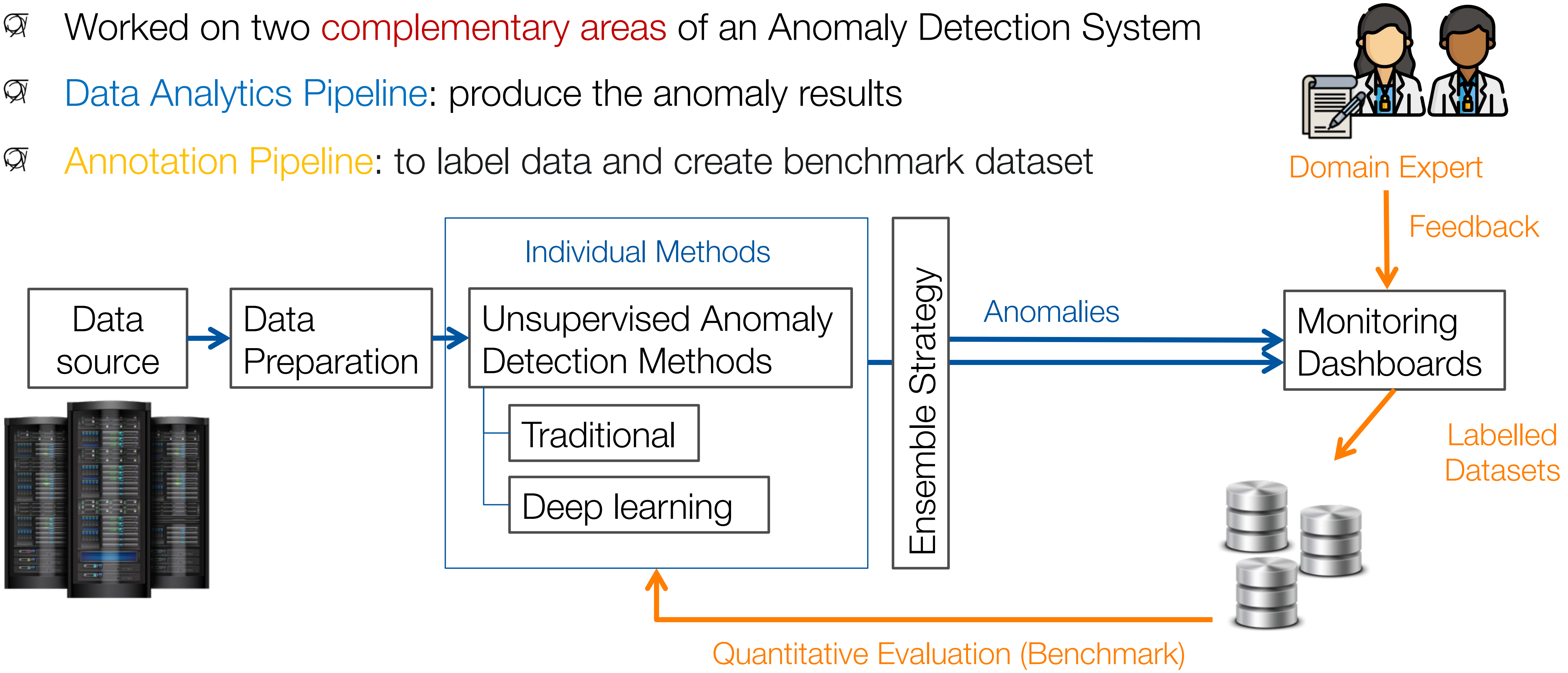
Objective of the Project

- Reliably detect anomalies in the CERN Cloud using *time series* monitoring data
  - Evaluate different algorithms suitable for the Cloud case
  - Use unsupervised techniques: lack of labelled datasets rules out supervised approaches
- Provide Cloud service managers with an Anomaly Detection System
  - Implement a production pipeline
- Project executed during 2020/2022

  The Anomaly Detection System is in production since then

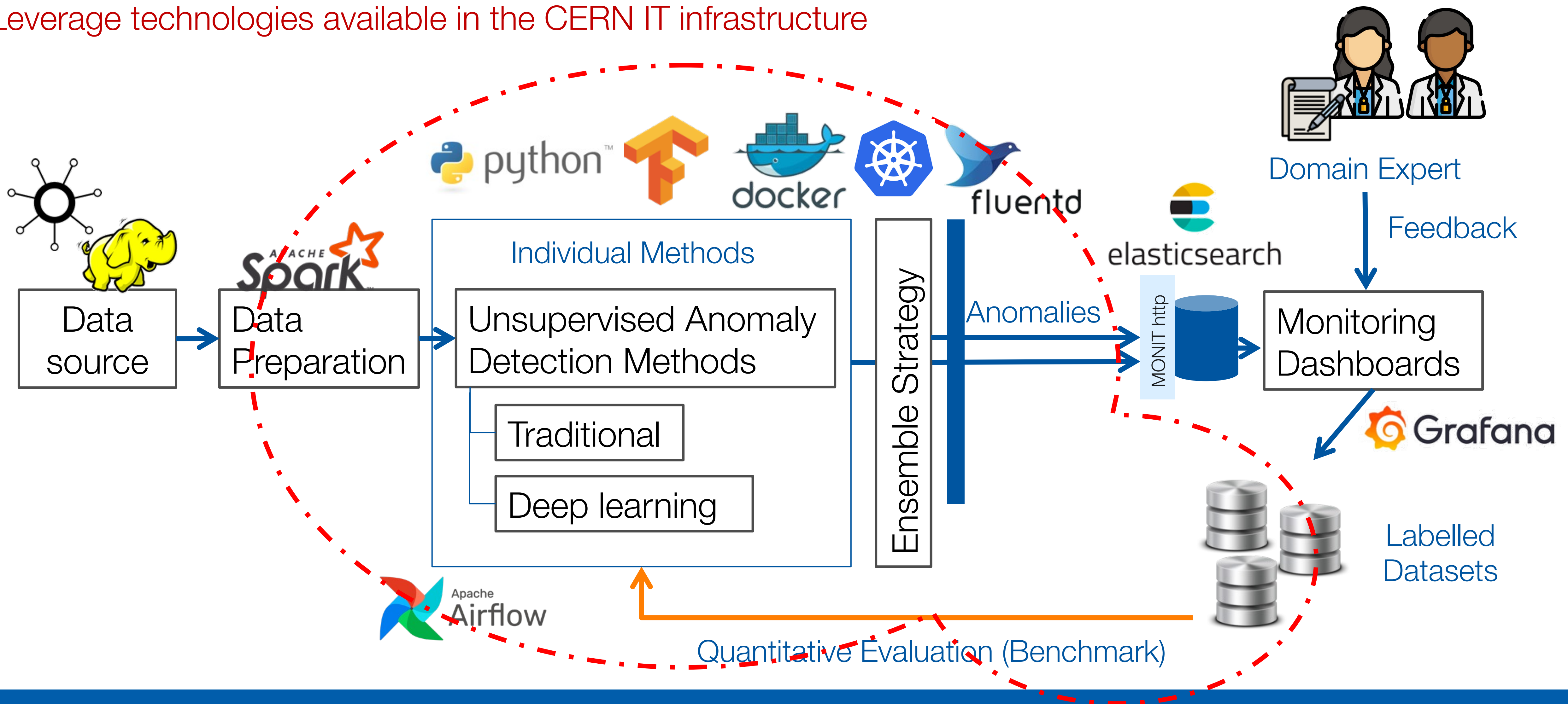
Anomaly Detection System: Design Concept

- Worked on two complementary areas of an Anomaly Detection System
- Data Analytics Pipeline: produce the anomaly results
- Annotation Pipeline: to label data and create benchmark dataset

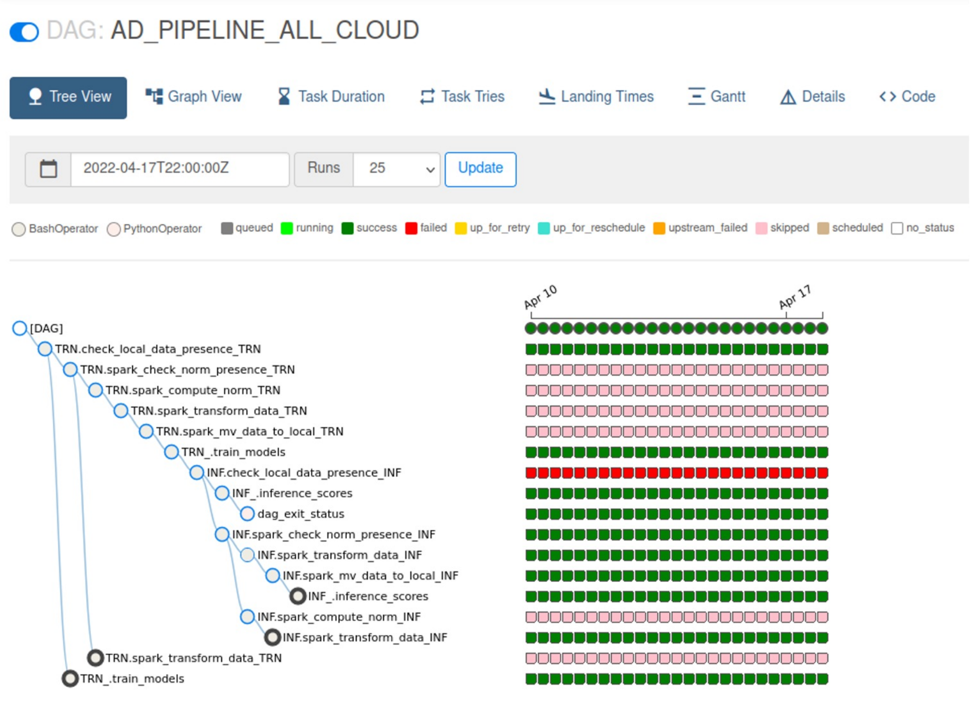


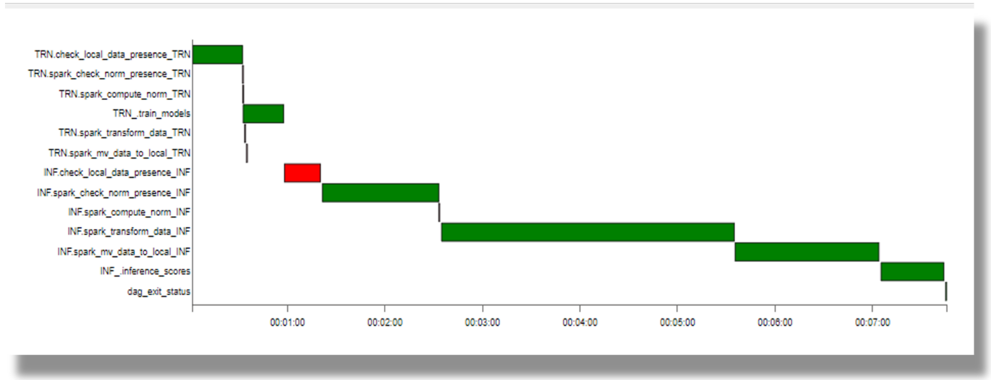
Anomaly Detection System: Technologies

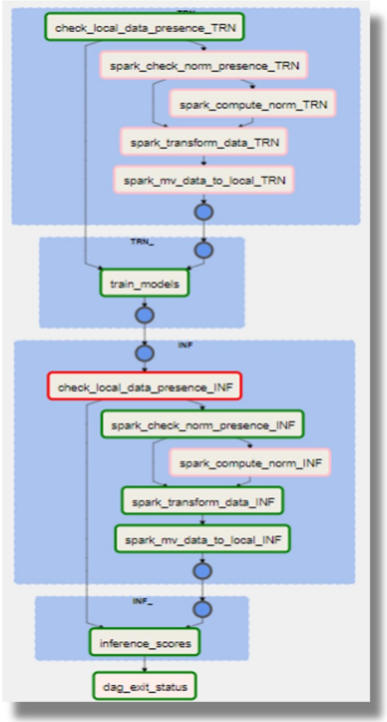
Leverage technologies available in the CERN IT infrastructure



Why Airflow?







Convenient for its easy deployment, scheduling and monitoring

- DAGs to declare Training / Inference stages
- Airflow running in Docker containers
  - Images built by GitLab CI/CD
  - Orchestrated by Docker Compose
  - Optional creation of local DEV Elasticsearch container
- At the time of the project Kubeflow @CERN did not include the whole functionalities we needed
  - eg. Spark-Kubeflow integration
  - Today we would probably start from Kubeflow

Considerations / Challenges

A key aspect for a rapid progress of the work has been

- Integration and portability of tools: ability to develop, test, run using the same approaches
  - The developed code could run in notebooks, Gitlab CI/CD, production deployment
- Modularity of the ML libraries to easily include new algorithms

Challenges

- Size and quality of the annotated datasets is vital
  - Need tools to automatically include users' flags into the labelling task

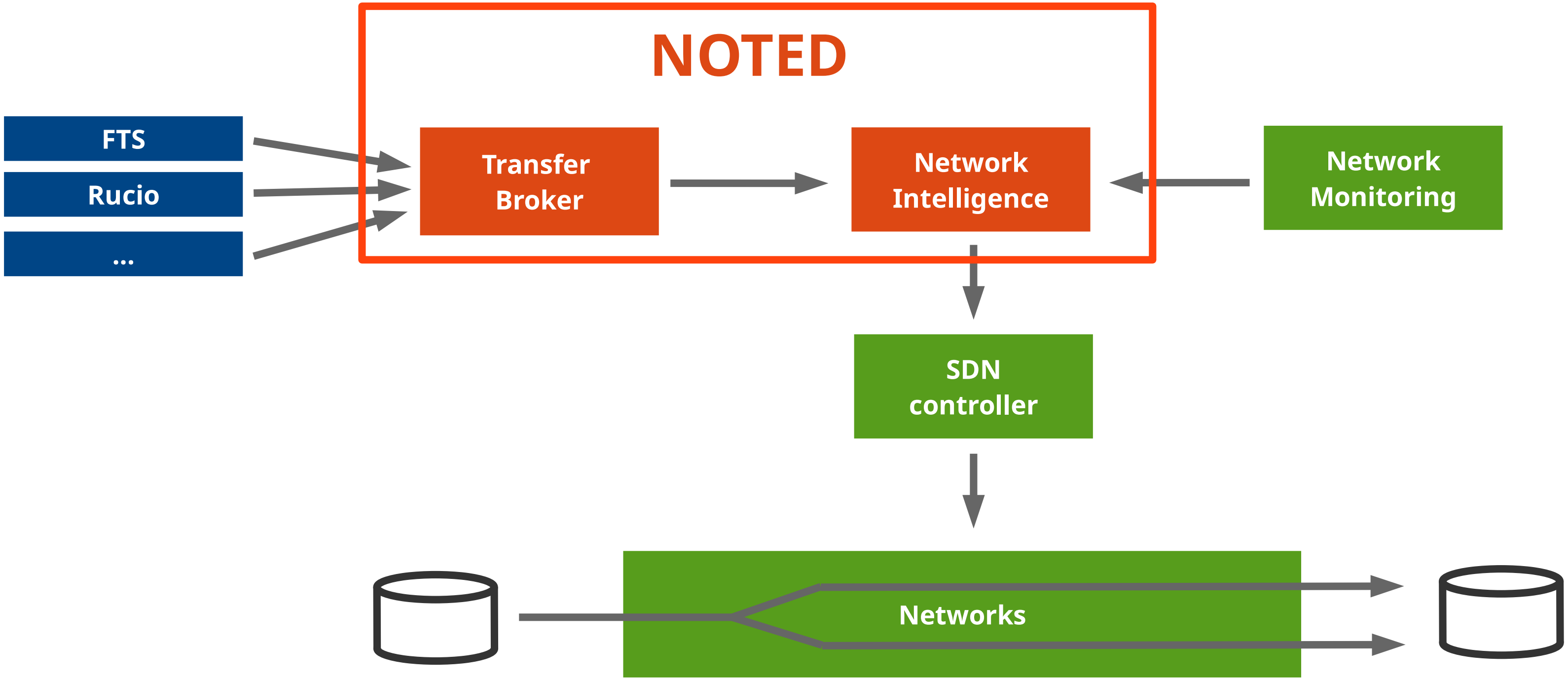
www.cern.ch

NOTED and NL

CERN IT Machine Learning Infrastructure Workshop
10th March 2023
Carmen Misa Moreira and Edoardo Martelli

Network Optimized Transfers of Experimental Data

NOTED: framework that dynamically improves network performances for **large, on-going, long-lasting** data transfers



Data Transfers

- The current NOTED implementation works only with FTS

- NOTED queries FTS via the CERN MONIT Infrastructure

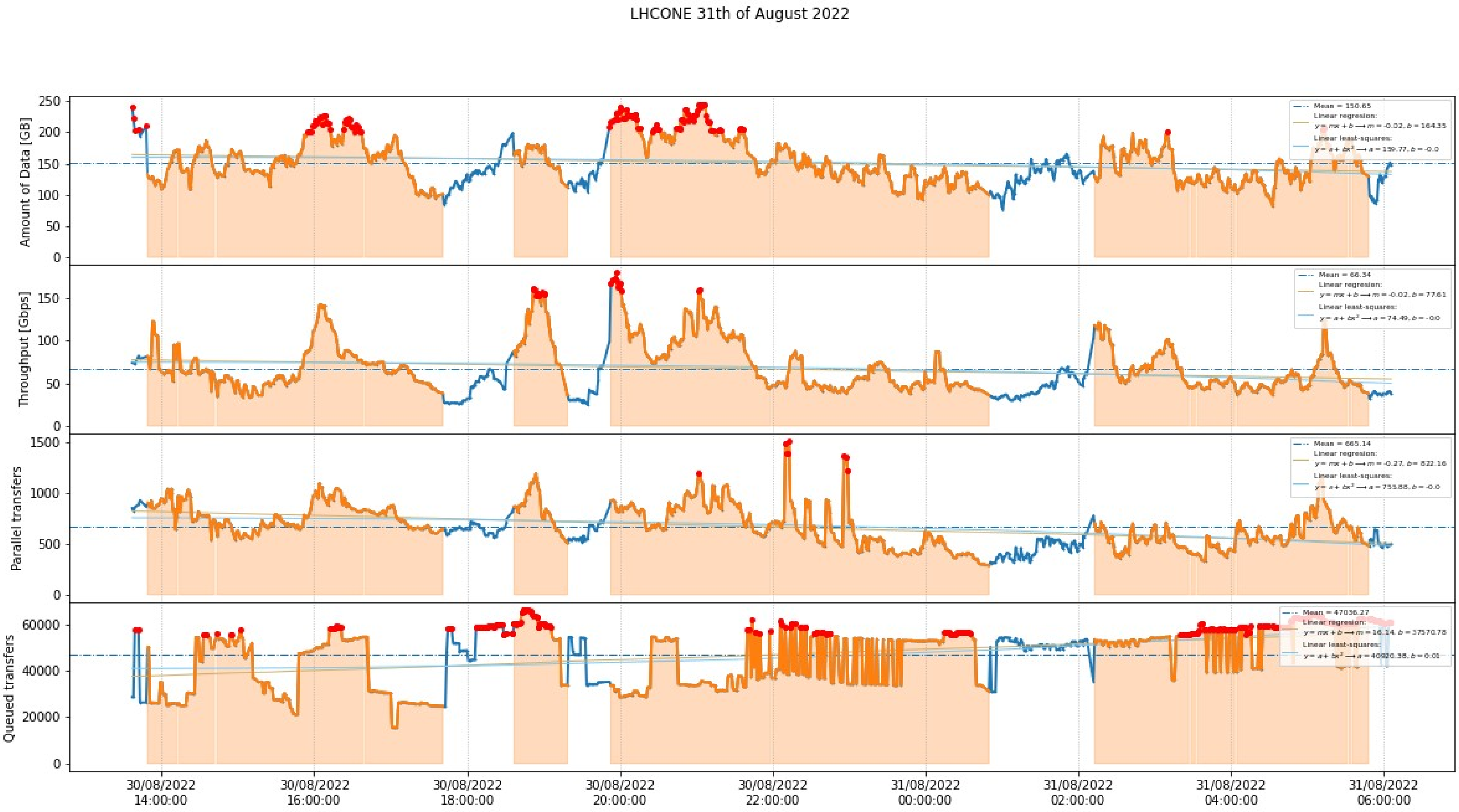
- Relevant parameters collected:
  - **{source se, dest se}**: source and destination endpoints involved in the transfer
  - **{throughput, filesize avg}**: throughput [bytes/s] and filesize [bytes] of the transfer
  - **{active count, success rate}**: number of TCP parallel flows and successful rate of the transfer
  - **{submitted count, connections}**: number of transfers in the queue and maximum number of transfers that can be held

Machine learning

Machine Learning LSTM has been tested to better estimate the duration and the size of the transfers

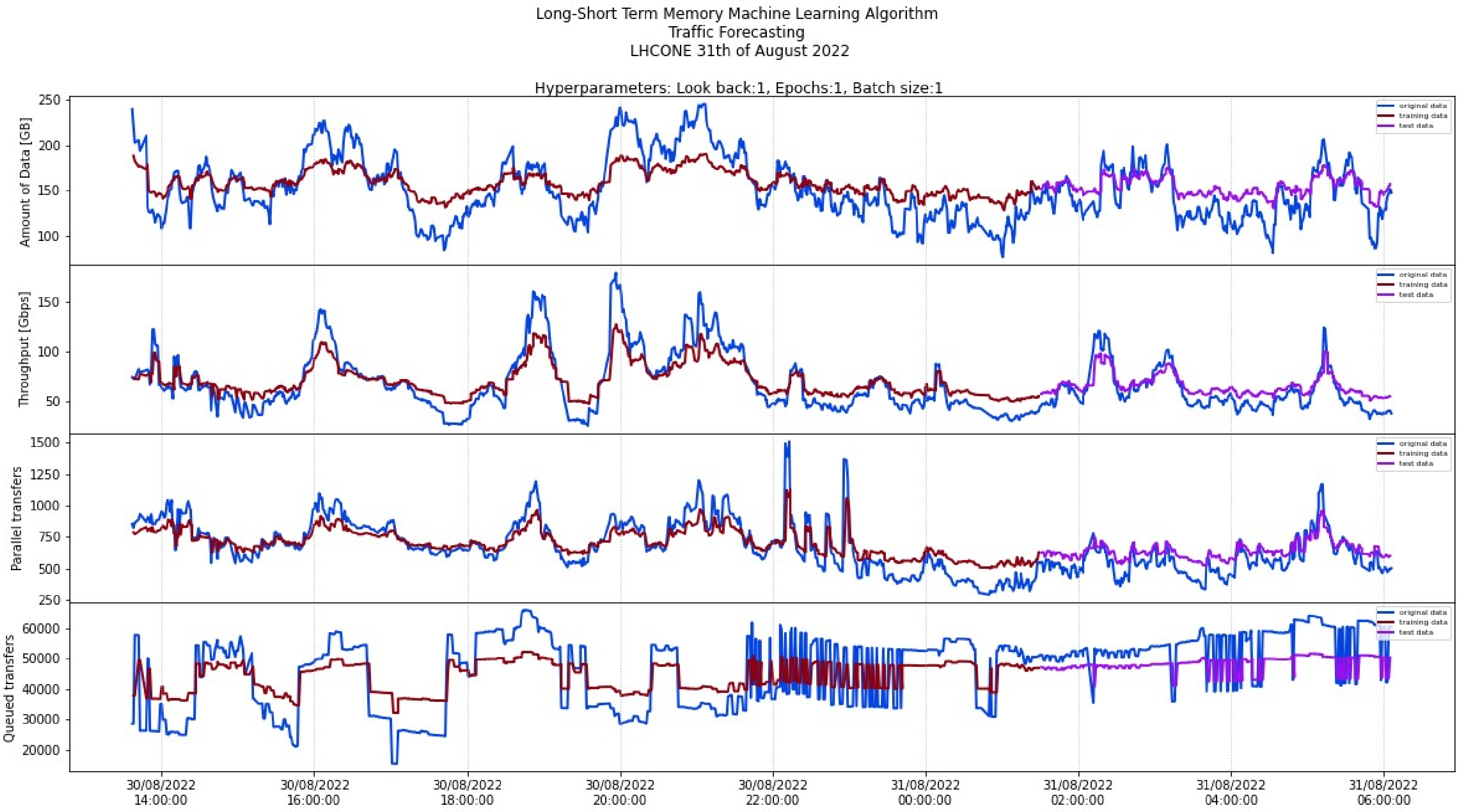
Work in progress

“Plain” NOTED in actions



Orange area: NOTED triggered network action

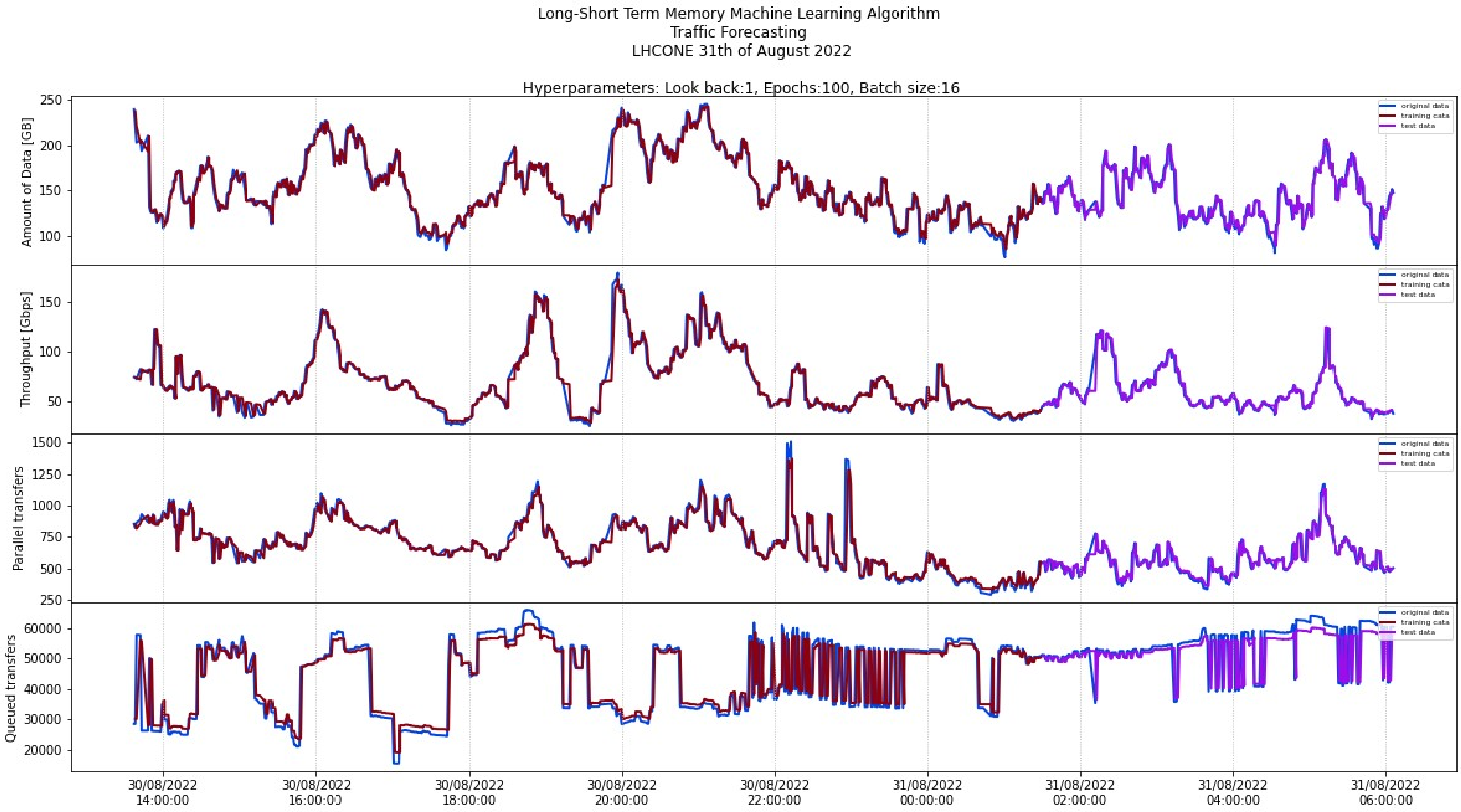
Traffic forecast with LSTM



Real data
Training data
Predicted data

This model is not well fitted to the data

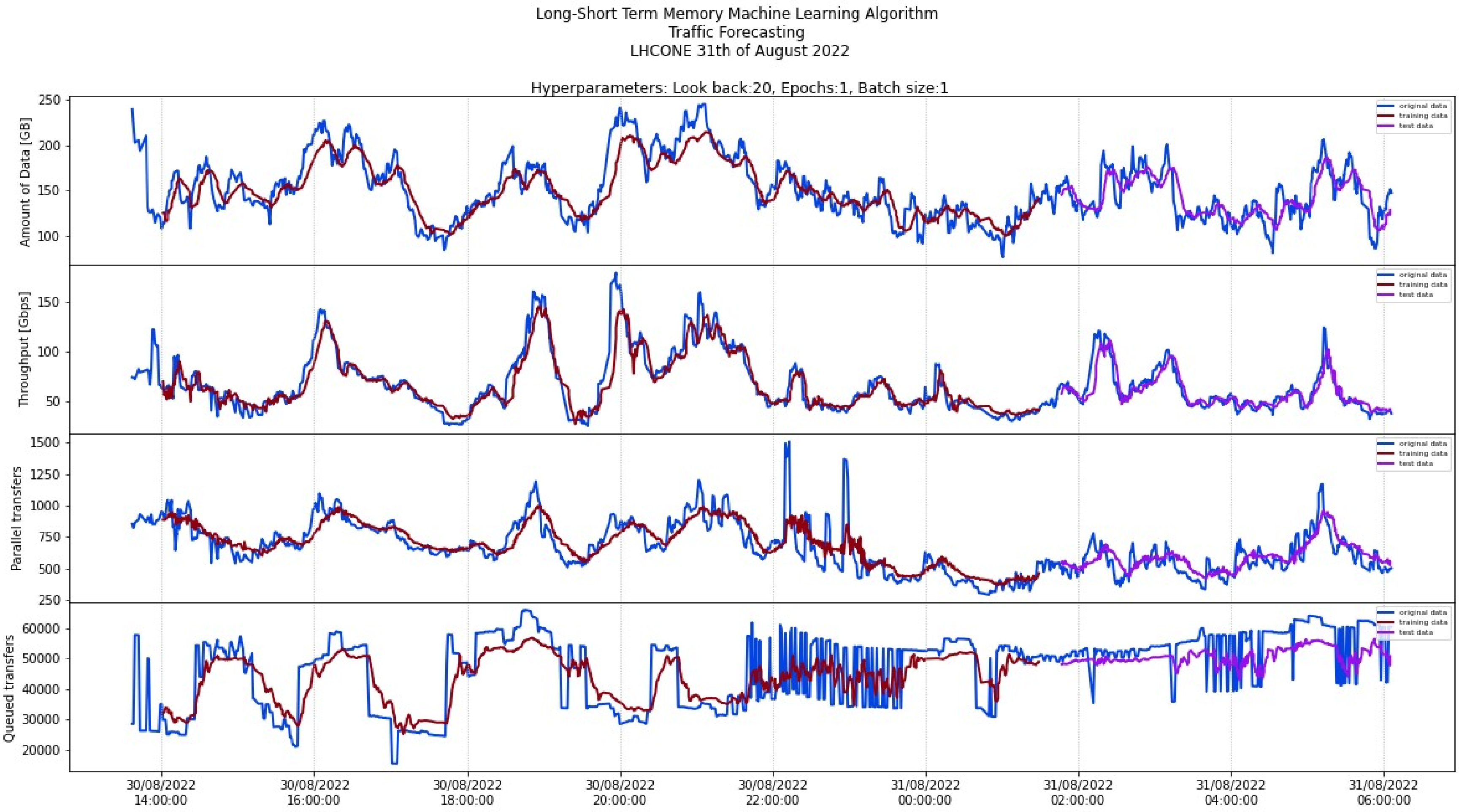
Traffic forecast with LSTM



With increased Epoch and Batch size, the model fits very well

Real data
Training data
Predicted data

Traffic forecast with LSTM



Real data
Training data
Predicted data

With increased loopback (20), the model fits well even with epoch 1 and batch 1

Layers of the LSTM network

The LSTM network has:

- a visible layer with 1 input,
- a hidden layer with 4 LSTM blocks or neurons,
- an output layer that makes a single value prediction

The sigmoid activation function is used for the LSTM blocks

Execution details

Look back: 1 Epochs: 1 Batch size: 1

```
CPU times: user 3.45 s, sys: 120 ms, total: 3.57 s
Peak memory: 711.70 MiB
Train Score: 15.77 RMSE
Test Score:  11.37 RMSE
Length of train dataset: 821
Length of test dataset: 353
```

Look back: 1 Epochs: 100 Batch size: 16

```
CPU times: user 16.3 s, sys: 578 ms, total: 16.8 s
Peak memory: 816.57 MiB
Train Score: 6.96 RMSE
Test Score:  5.59 RMSE
Length of train dataset: 821
Length of test dataset: 353
```

Look back: 20 Epochs: 1 Batch size: 1

```
CPU times: user 5.57 s, sys: 138 ms, total: 5.7 s
Peak memory: 823.27 MiB
Train Score: 11.83 RMSE
Test Score:  9.22 RMSE
Length of train dataset: 821
Length of test dataset: 353
```

Future research

Use autoencoders and transformers

Make predictions in real time

Questions?

edoardo.martelli@cern.ch
carmen.misa@cern.ch

CERN openlab

Foundation Model

Renato Cardoso, Sofia Vallecorsa

Work realized in collaboration with IBM

Foundation Models

- A model trained on broad data and adaptable to a range of different downstream tasks, zero-shot, few-shot learning.
- Foundation Models concepts:
  - self/semi-supervised learning + transfer learning but at scale:
    - Billions of parameters and gigabytes of data
    - Large and diverse datasets → powerful representations
- Examples:
  - BERT (340M params.), GPT-2, GPT-3 (175B params.) – Generative language models
  - CLIP – Language-Image pre-training
  - DALL-E, DALL-E 2, Imagen – Text to Image models
  - GATO – Sequence to sequence model

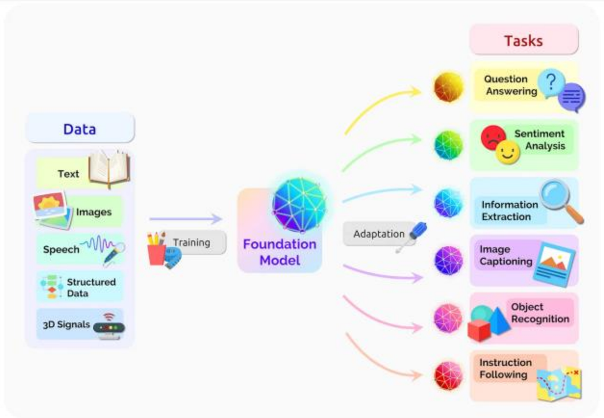


Image obtained from:
On the Opportunities and Risks of Foundation Models

- Stanford CRFM (2021) : On the Opportunities and Risks of Foundation Models [arxiv.2108.07258]

Foundation Models

Why use Foundation Models:

- ML is computational expensive
  - Train once. Then, adapt to new detector geometries, quickly.
- Transformers as building block in foundation models:
  - A generalized architecture without any inductive bias
  - Model long-range dependencies (Attention mechanism)
  - Permutation invariant
  - [arXiv:1706.03762]

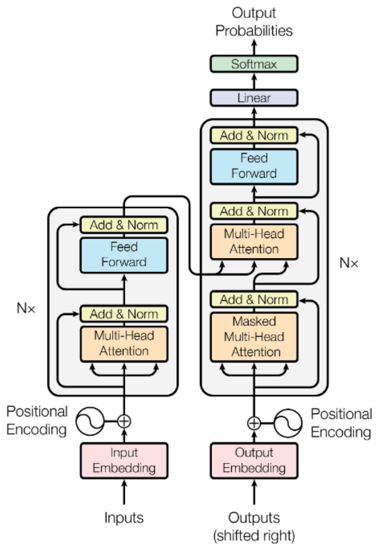


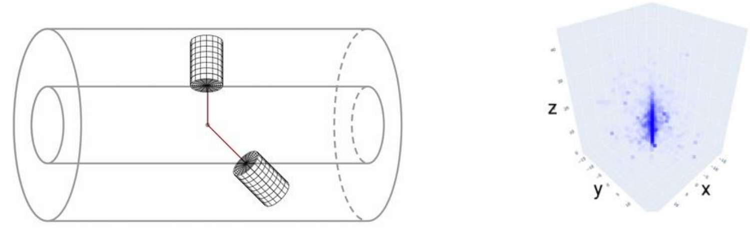
Figure 1: The Transformer - model architecture.

Our Objective:

- Foundation model trained on MC data to perform different physics related tasks
  - Simulations - one lengthy training, then fast adaptation to different detector geometries
  - Reconstruction - one base model adaptable to different tasks (particle identification, regression on phys. variables, etc.)
- Understand how foundation model concept apply to our use case:
  - Understand the minimal scale of the model for reaching meaningful results (No need to reach BERT / GPT-3 scale)

Work done

Dataset: High Granularity Electromagnetic Calorimeter Shower Images

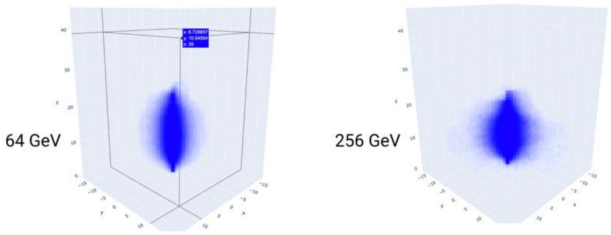


Dataset

Our first task Foundation model for fast and accurate calorimetry simulation

Single dataset training multiple model architectures:

- Vision Transformer (ViT) based architecture [arXiv:2010.11929]
  - Masked Model
- VAE-like learning model with transformers
- Graph neural network
- VQ-VAE model [arXiv:1711.00937]
- DDPM model [arXiv:2006.11239]
- Other tests:
  - Preprocessing
  - Sinkhorn Loss
  - Regression Loss
  - Etc.



Results Obtained from ViT based architecture model

Infrastructure

Why do we need computational infrastructure for this project:

- Models with a high number of parameters
  - High parallelizable but take time to train
- Multiple test being realized simultaneously
  - Multiple people working in the same project
  - Optimization of a single model takes a lot of time with minimal resources
- Memory requirements
  - Big models not only take time to train they need GPUs with a high amount of memory

CERN openlab

AN INTERDISCIPLINARY DIGITAL TWIN ENGINE FOR SCIENCE

CERN IT Machine Learning Infrastructure Workshop

Matteo Bunino, Kalliopi Tsolaki, Alexander Zöchbauer, Maria Girone,
Alberto Di Meglio, Sofia Vallecorsa, CERN-IT-GOV-INN

Funded by the European Union

The interTwin project is funded by the European Union - Grant Agreement Number 101058386

Motivation and Objective

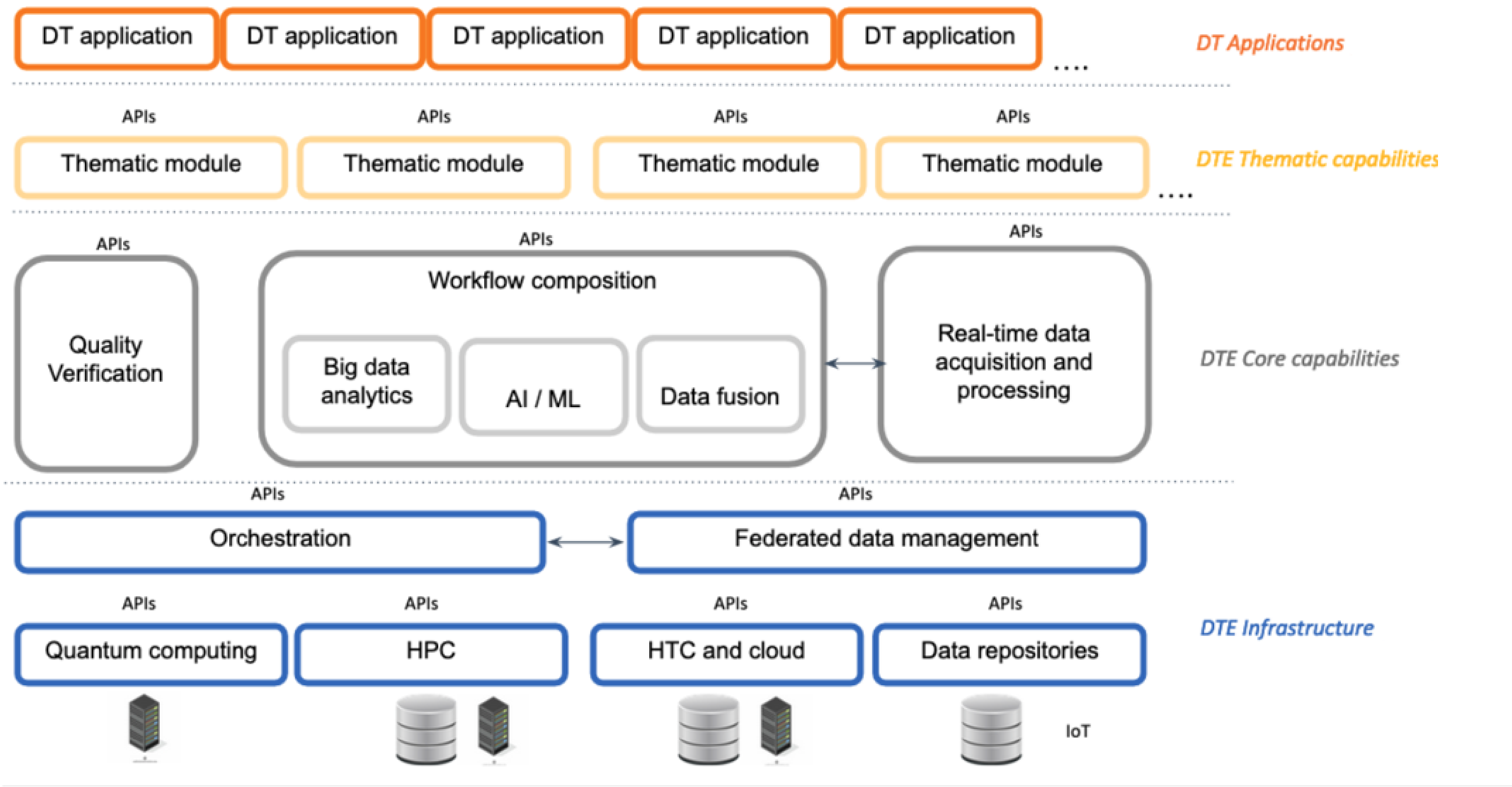
- **What is it**
  - EC project for co-designing and implementing the prototype of a Digital Twin Engine (DTE) covering a variety of scientific topics, from HEP to radio astronomy and from Lattice QCD to climate extreme events projection

- **Objective**
  - Development of general-purpose and automated DT workflows to relieve scientists from low-level engineering problems when working with DT applications

7 DT use cases

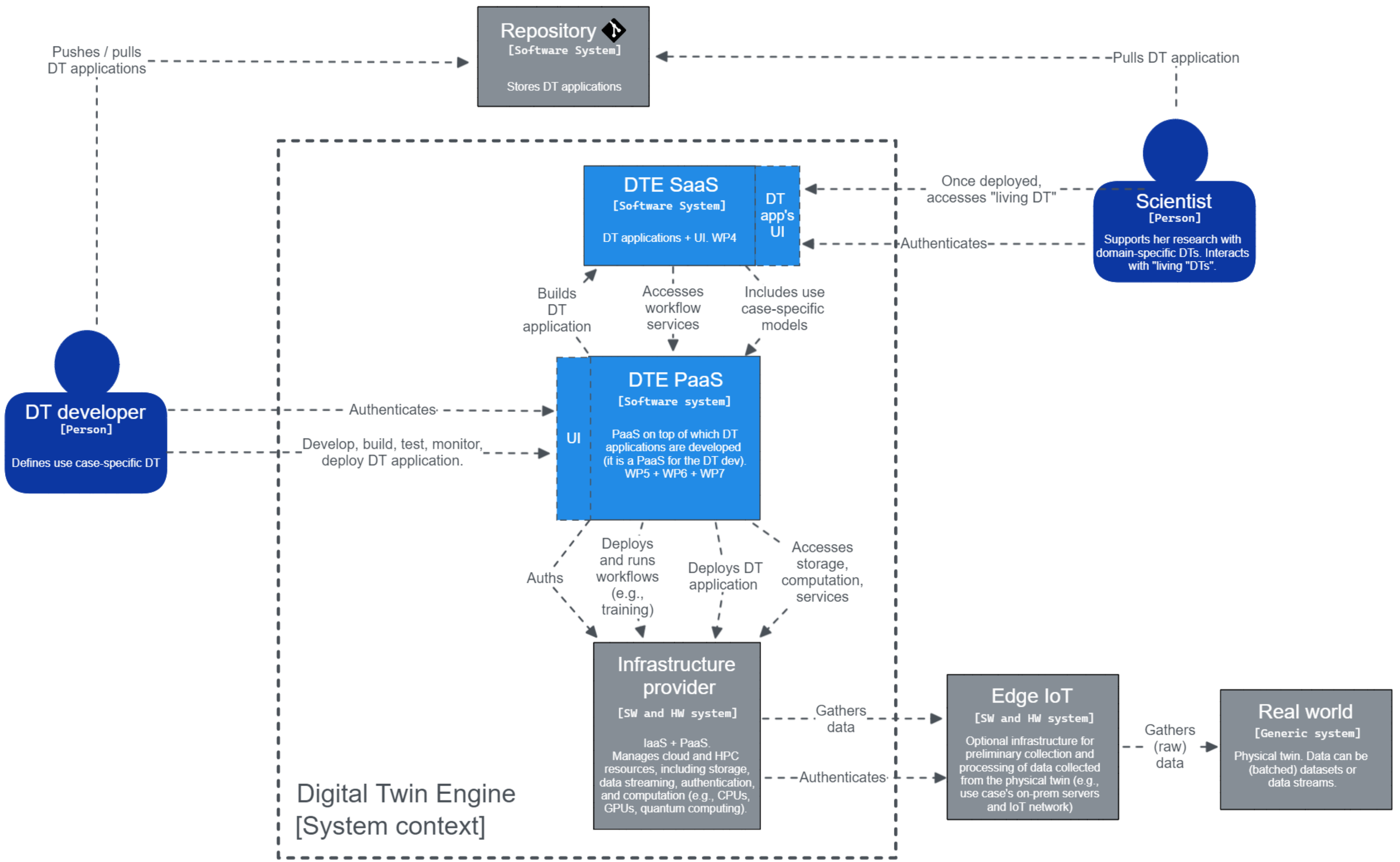
The partners

Implementation



interTwin Digital Twin Engine conceptual model

Implementation – cont'd



Project requirements

Consolidated requirements from ALL use cases concerning physics and environment domain:

- **Storage I/O**: Cloud, File-based, Object-based, HPC centers
- **Data volume**: Range from 10s GB to TBs
- **Data formats**:
  - Physics: Binary, text, ROOT, HDF5,
  - Climate: NetCDF, CSV, GetTIFF
- **Computing:** CPU, **GPU**, **HPC**, HTC, MPI infrastructure
- **OS and execution framework:** Linux, Containers (Docker, Singularity)
- **Big data processing:** Apache Spark, OpenEO
- **Workflow composition/engines**: **Apache Airflow**, OSCAR, Kubeflow, K8s.
- **Machine Learning:** Tensorflow, PyTorch, **distributed ML (e.g., Horovod)**, **MLOps** (e.g., Kubeflow)
- **Real-time data acquisition and processing:** **Streaming** platforms (e.g., Apache Kafka), off-line/online pre-processing
- **Software stack:** Geant4, ROOT, C/C++, Python, R, Jupyter Notebooks, openEO
- **Visualization:** Visualization frameworks (not specified, except Tensorboard)

Back up

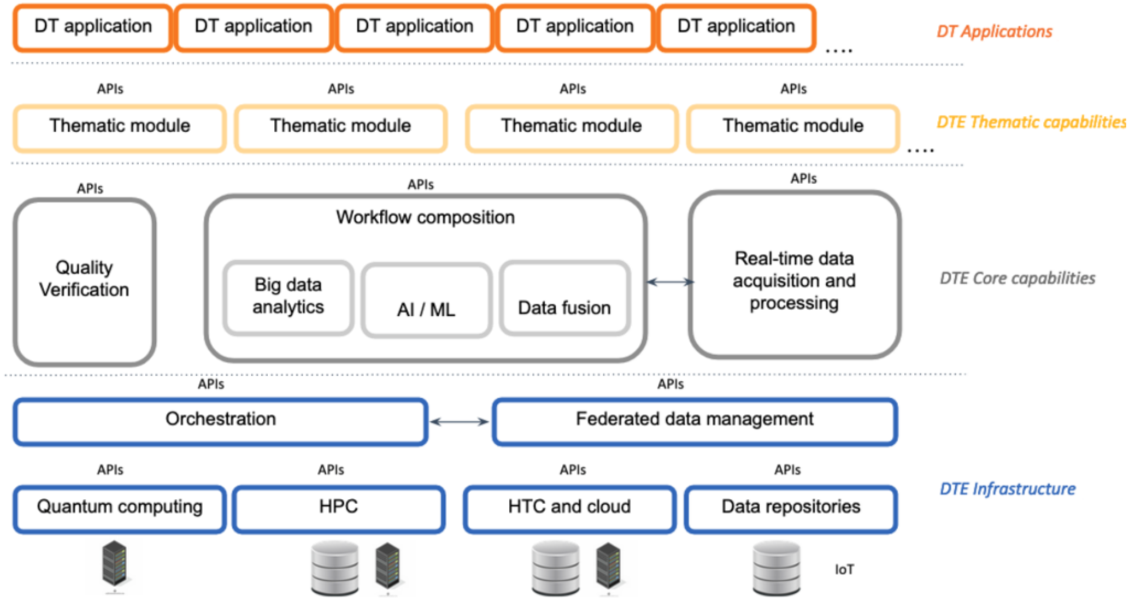
Funded by the European Union

The interTwin project is funded by the European Union - Grant Agreement Number 101058386

CERN activities

- **CERN involvement**
  - Technical co-design and validation of the use cases
    - Detector simulation
  - DTE Infrastructure
    - Federated data infrastructure
  - DTE Core Modules
    - AI workflow and method lifecycle
  - DTE Thematic Modules
    - Fast simulation with GAN

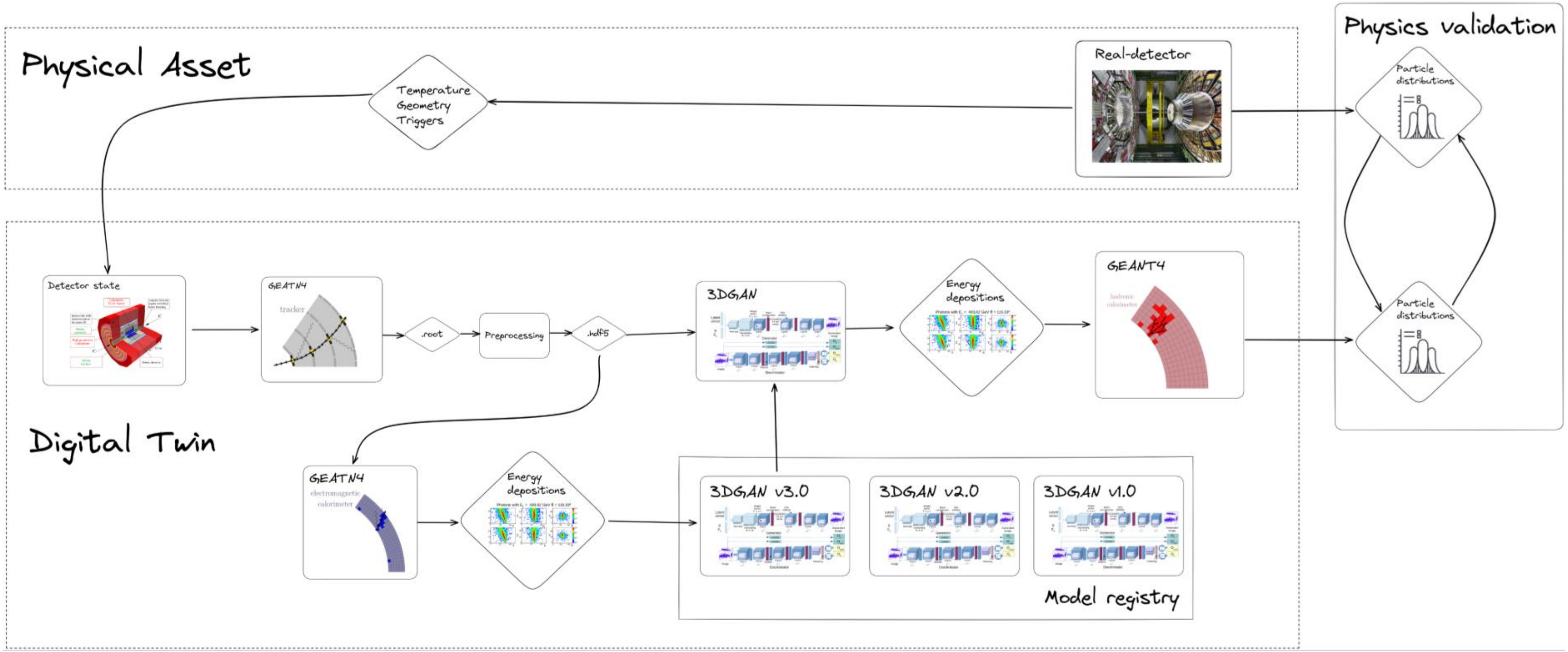
- **Activities**
  - *Analyze use cases requirements*
  - *Co-design a DT model for CERN use case with other use cases*
  - *Develop fast detector simulation exploiting GAN based model*
  - *Develop unified MLOps workflow for data-driven DT models*



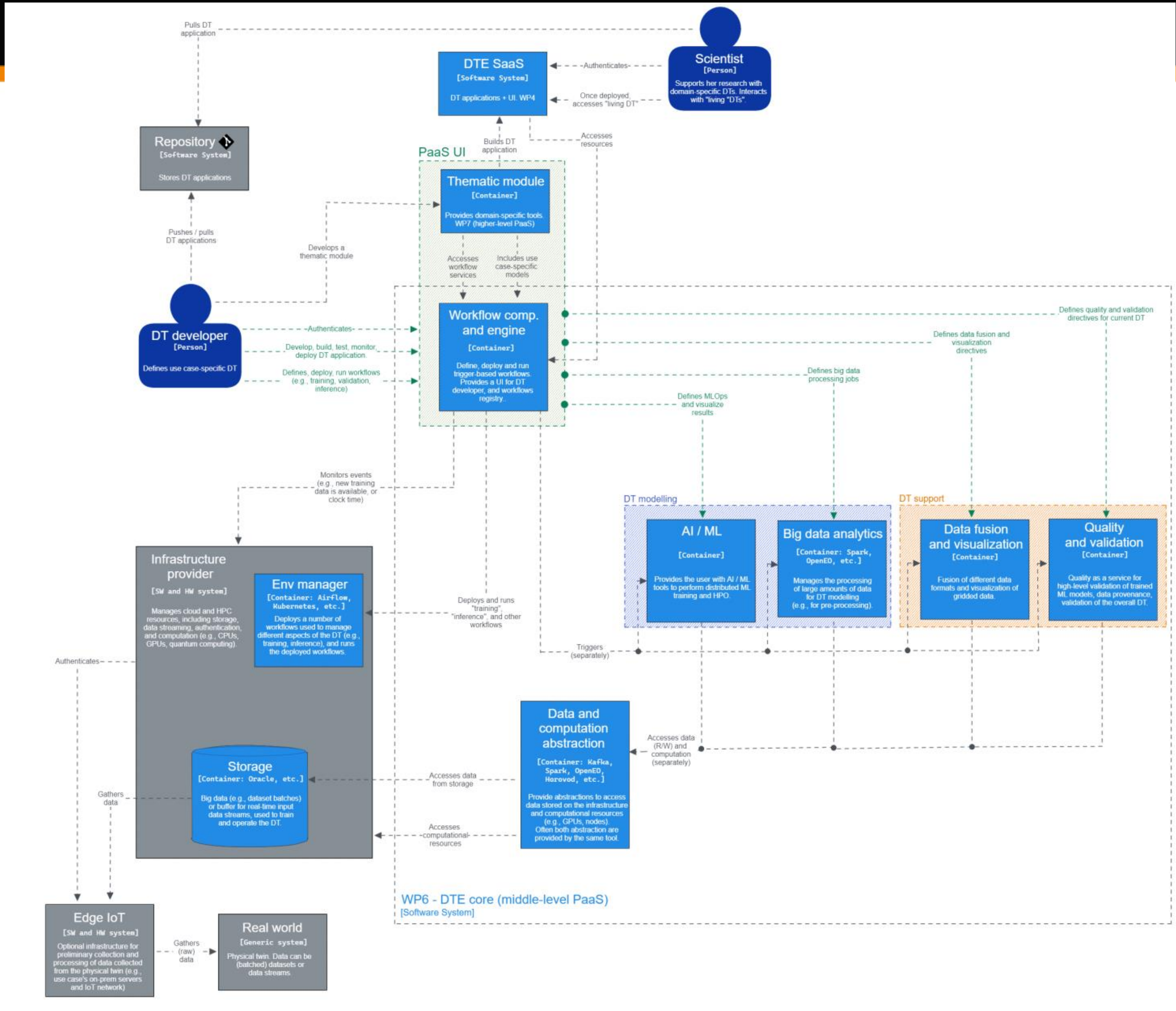
interTwin Digital Twin Engine conceptual model

Possible Future DT applications at CERN

- **Online-ML for Detectors:** adapt in real-time to property changes of detector configuration in geometry, temperature, trigger thresholds
- **Detector Prototyping:** build a DT of a testbench detector and test it on conditions that can't be recreated in the lab easily

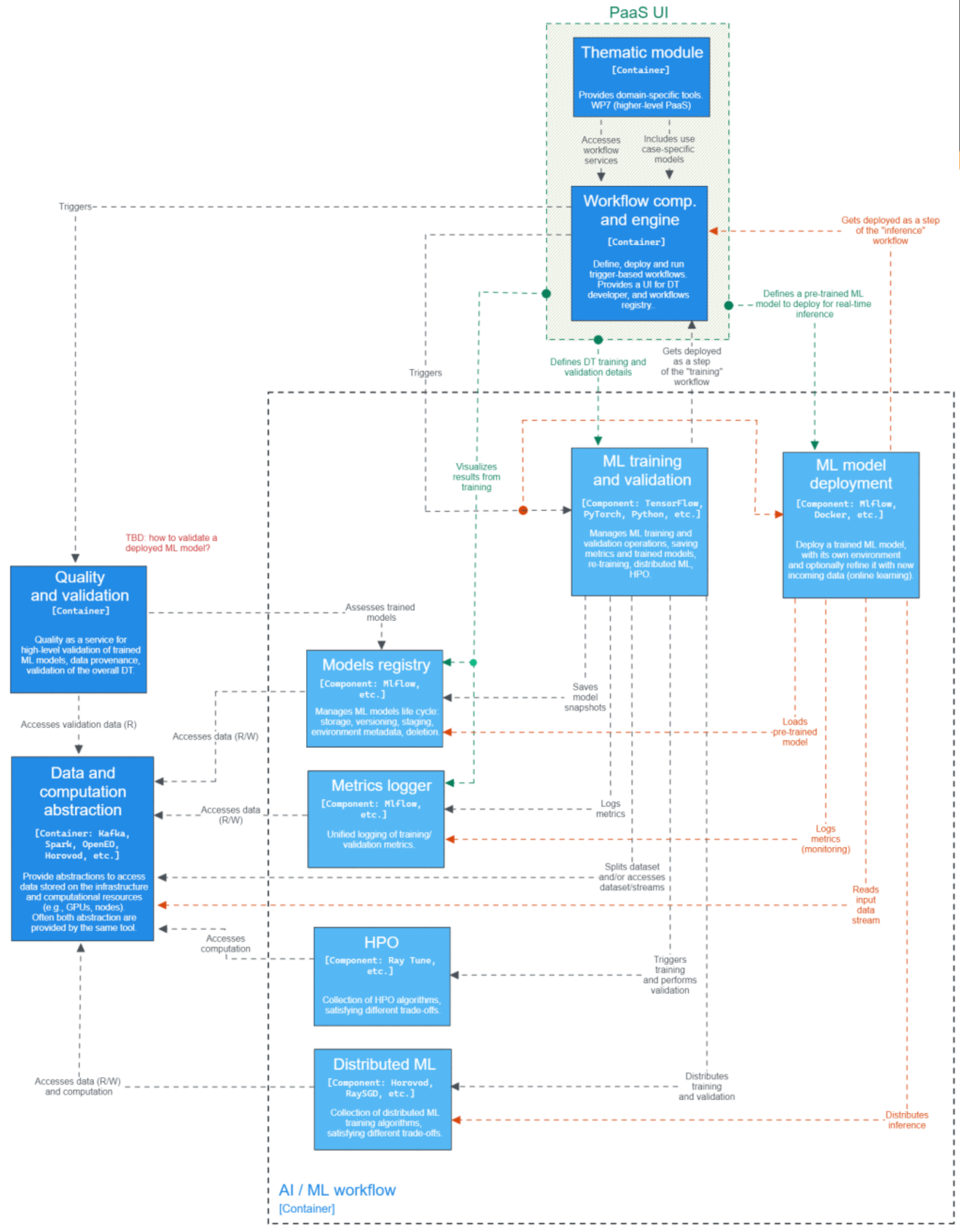


DTE Composition



[Container] DTE core (WP6)

ML/AI module composition



[Component] DTE - AI / ML workflow (T6.5)

REANA and ML

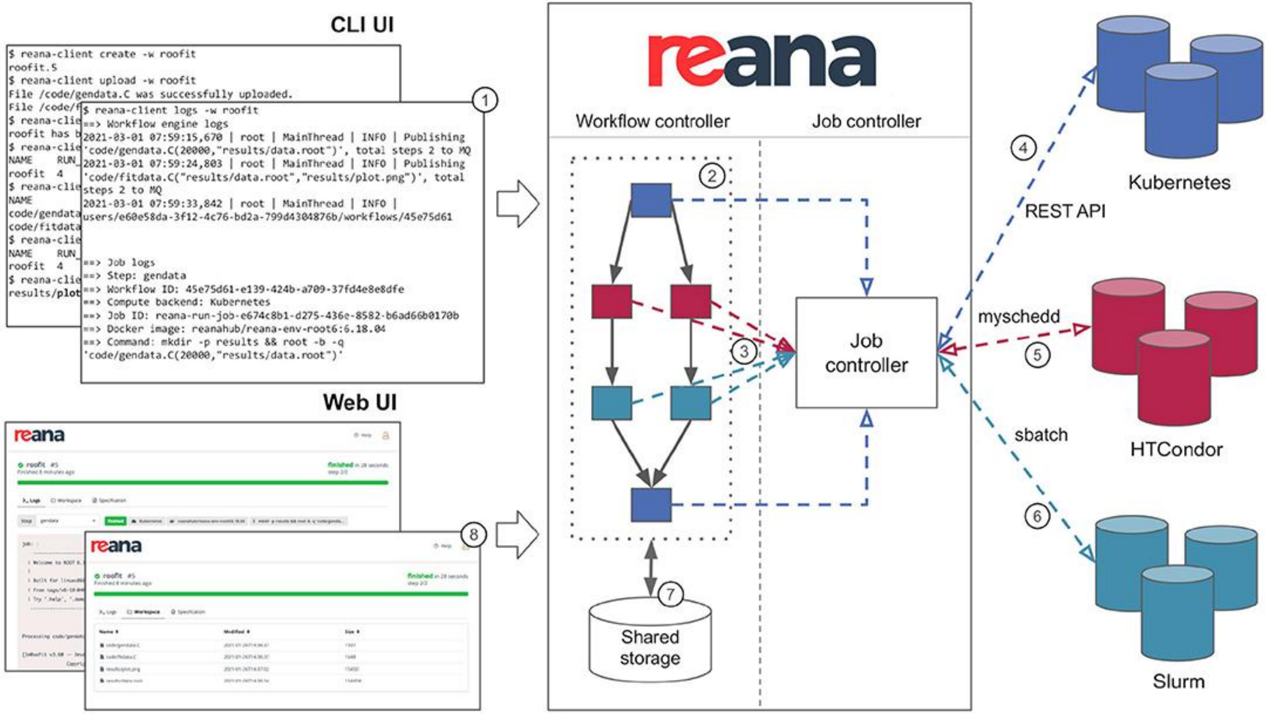
Tibor Simko
IT-PW

CERN IT Machine Learning Infrastructure Workshop, March 10th 2023

https://indico.cern.ch/event/1253881

REANA Reusable Analysis platform

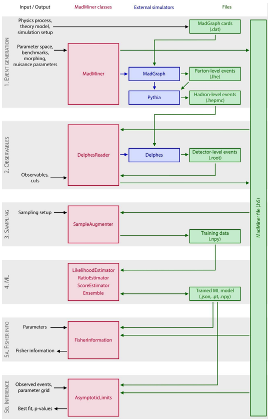
https://www.reana.io

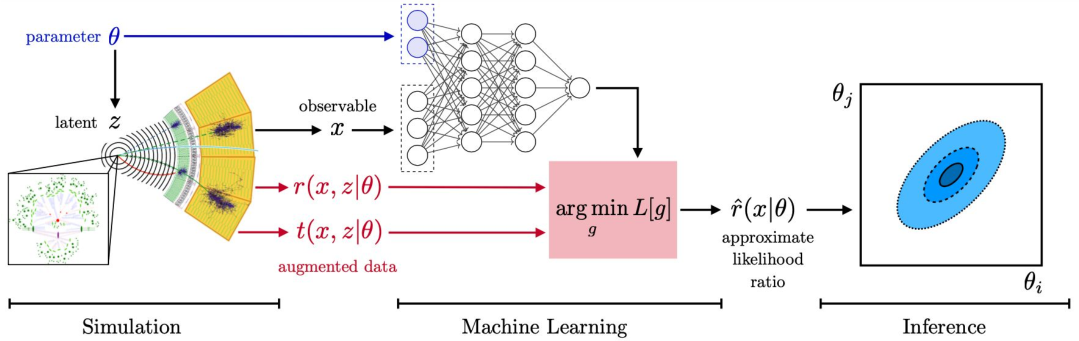


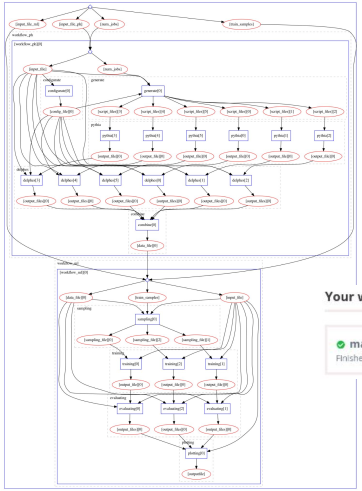
Running declarative containerised computational workflows

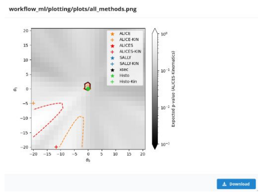
ML use cases on REANA 1/2

Pheno-level analyses embedded into Python ML ecosystem (and optionally MLFlow)





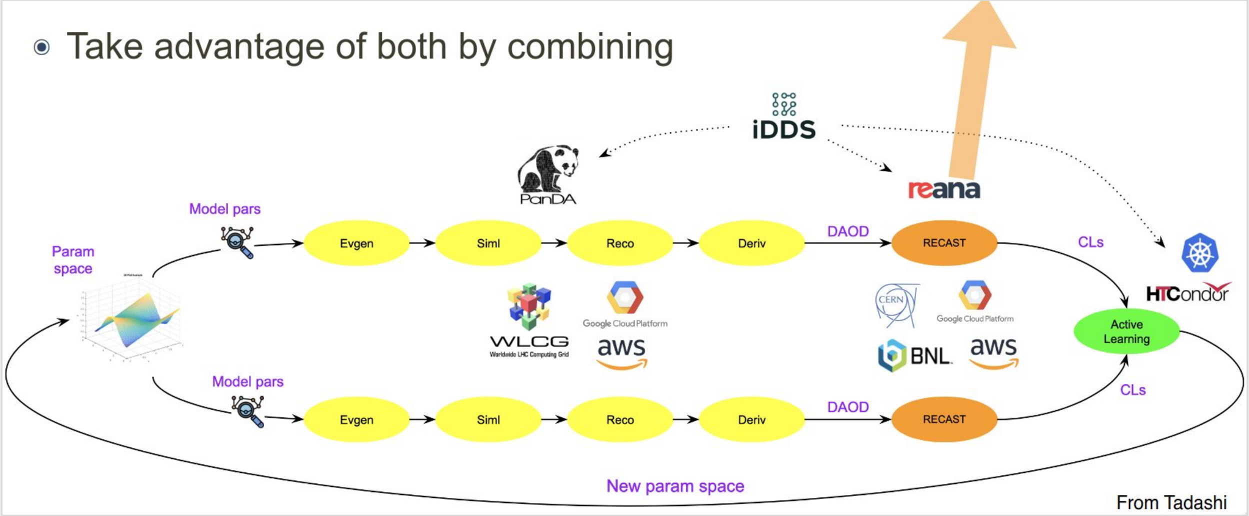




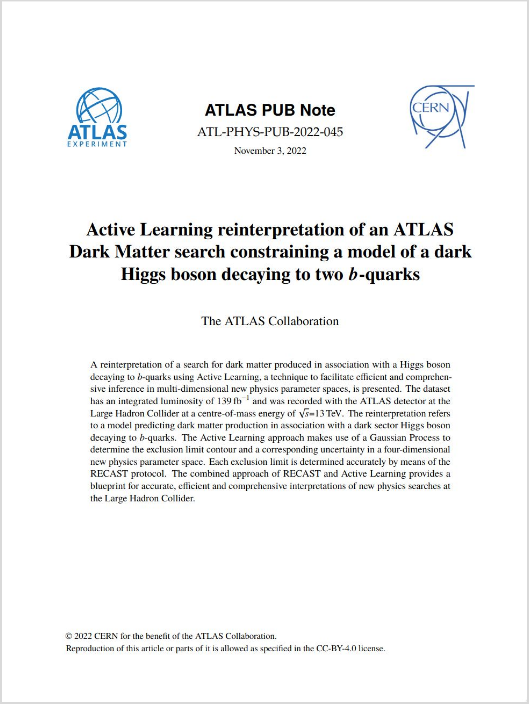
"MadMiner: Machine learning-based inference for particle physics", J. Brehmer, F. Kling, I. Espejo, K. Cranmer, [arXiv:1907.10621](https://arxiv.org/abs/1907.10621).

- Running ML based workflows

ML use cases on REANA 2/2



“REANA / PanDA integration for Active Learning”, W.Guan, T.Maeno, C.Weber, T.Wenaus, R.Zhang, https://indico.cern.ch/event/1134581.



ATLAS PUB Note

ATL-PHYS-PUB-2022-045

November 3, 2022

Active Learning reinterpretation of an ATLAS Dark Matter search constraining a model of a dark Higgs boson decaying to two b-quarks

The ATLAS Collaboration

A reinterpretation of a search for dark matter produced in association with a Higgs boson decaying to b-quarks using Active Learning, a technique to facilitate efficient and comprehensive inference in multi-dimensional new physics parameter spaces, is presented. The dataset has an integrated luminosity of 139 fb^{-1} and was recorded with the ATLAS detector at the Large Hadron Collider at a centre-of-mass energy of $\sqrt{s}$=13 TeV. The reinterpretation refers to a model predicting dark matter production in association with a dark sector Higgs boson decaying to b-quarks. The Active Learning approach makes use of a Gaussian Process to determine the exclusion limit contour and a corresponding uncertainty in a four-dimensional new physics parameter space. Each exclusion limit is determined accurately by means of the RECAST protocol. The combined approach of RECAST and Active Learning provides a blueprint for accurate, efficient and comprehensive interpretations of new physics searches at the Large Hadron Collider.

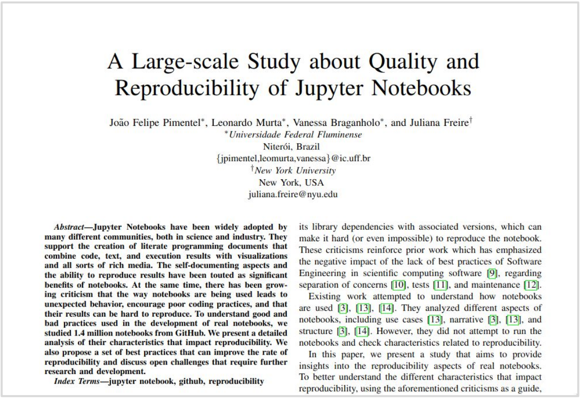
© 2022 CERN for the benefit of the ATLAS Collaboration.
Reproduction of this article or parts of it is allowed as specified in the CC-BY-4.0 license.

ATL-PHYS-PUB-2022-045

- Running workflows as part of a bigger data processing chain (whole physics analysis from MC generation to new physics discovery)

Possible areas of interest

- Capturing the knowledge behind data analyses
  → *preserve to reuse*
- Computational reproducibility
  → *run outside the original context*
- Running workflows at scale
  → *10k workflows for ATLAS pMSSM searches*
- “Continuous analyses”
  → *Gitlab-REANA bridge*
- Interplay between notebooks and workflows
  → *interactive vs batch*



A Large-scale Study about Quality and Reproducibility of Jupyter Notebooks

João Felipe Pimentel*, Leonardo Murta*, Vanessa Braganholo*, and Juliana Freire†
*Universidade Federal Fluminense
Niterói, Brazil
{jpimentel,leomurta,vanessa}@ic.uff.br
†New York University
New York, USA
juliana.freire@nyu.edu

***Abstract*—Jupyter Notebooks have been widely adopted by many different communities, both in science and industry. They support the creation of literate programming documents that combine code, text, and execution results with visualizations and all sorts of rich media. The self-documenting aspects and the ability to reproduce results have been touted as significant benefits of notebooks. At the same time, there has been growing criticism that the way notebooks are being used leads to unexpected behavior, encourage poor coding practices, and that their results can be hard to reproduce. To understand good and bad practices used in the development of real notebooks, we studied 1.4 million notebooks from GitHub. We present a detailed analysis of their characteristics that impact reproducibility. We also propose a set of best practices that can improve the rate of reproducibility and discuss open challenges that require further research and development.**

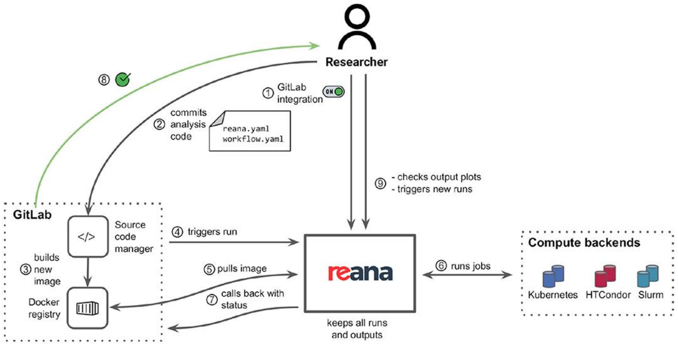
***Index Terms*—jupyter notebook, github, reproducibility**

its library dependencies with associated versions, which can make it hard (or even impossible) to reproduce the notebook. These criticisms reinforce prior work which has emphasized the negative impact of the lack of best practices of Software Engineering in scientific computing software [9], regarding separation of concerns [10], tests [11], and maintenance [12].

Existing work attempted to understand how notebooks are used [3], [13], [14]. They analyzed different aspects of notebooks, including use cases [13], narrative [3], [13], and structure [3], [14]. However, they did not attempt to run the notebooks and check characteristics related to reproducibility.

In this paper, we present a study that aims to provide insights into the reproducibility aspects of real notebooks. To better understand the different characteristics that impact reproducibility, using the aforementioned criticisms as a guide,

“...only 4.03% produced the same results”
DOI 10.1109/MSR.2019.00077



GitLab-REANA bridge