Anomaly Detection in the CMSWEB Services

CERN SUMMER STUDENT 2024

QUAID-I-AZAM UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

NASIR HUSSAIN

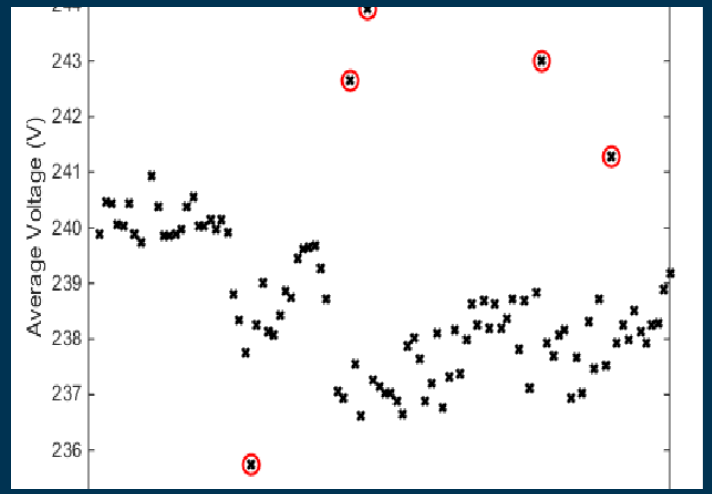
SUPERVISOR: DR. MUHAMMAD IMRAN

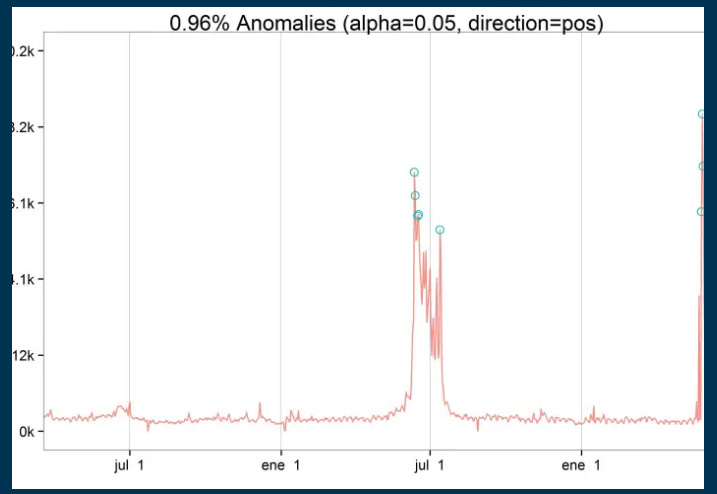
Project Overview

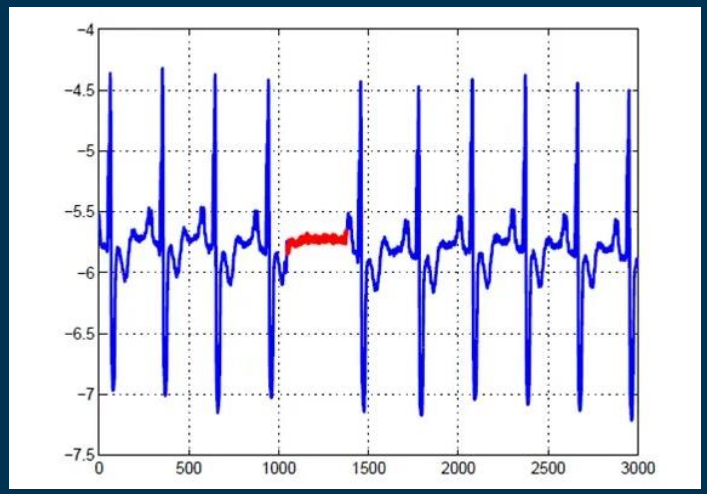
- **Objective:**
  Develop a robust application for anomaly detection within the CMSWEB cluster, leveraging machine/deep learning to ensure the reliability and security of critical web services.
- **Key Components:**
- **Infrastructure:**
  - Built on Kubernetes (k8s) technology.
  - Hosts over 24 critical web services (e.g., DBS, DAS, CRAB, WMarchive, WMCore).
- **Approach:**
  - Utilize machine/deep learning techniques to analyze service parameters.
  - Continuously monitor for deviations from expected behavior.
- **Functionality:**
  - Identify and discern anomalies indicative of security breaches or performance issues.
  - Record detected anomalies and generate alerts.
- **Alerting Mechanism:**
  - Intelligent routing of alerts to relevant service developers or administrators.
  - Proactive issue resolution to minimize disruptions.
- **Outcome:**
  Enhance the reliability and security of the CMSWEB cluster, ensuring seamless operation of critical services through timely anomaly detection and response.

Understanding Anomalies

An anomaly is an irregularity or deviation from the expected behavior in a system, which may indicate potential issues such as performance degradation, security breaches, or operational failures.







Point Anomalies

A point anomaly is where a single datapoint stands out from the expected pattern, range, or norm. In other words, the datapoint is unexpected.

Contextual Anomalies

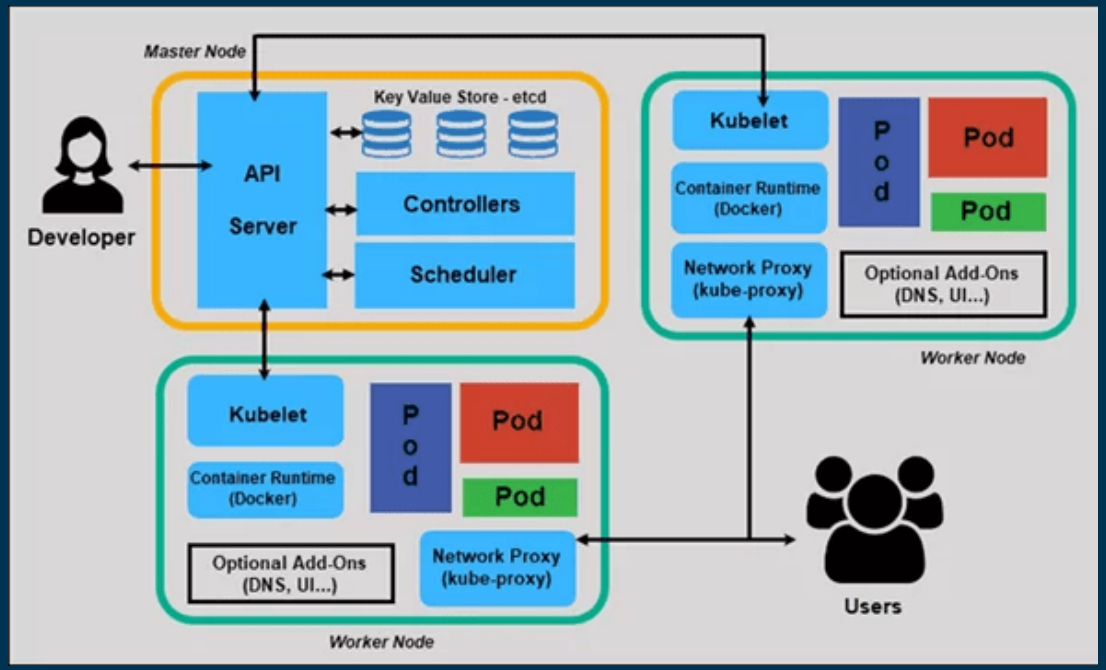
Instead of looking at specific datapoints or groups of data, an algorithm looking for contextual anomalies will be interested in unexpected results that come from what appears to be normal activity.

Collective Anomalies

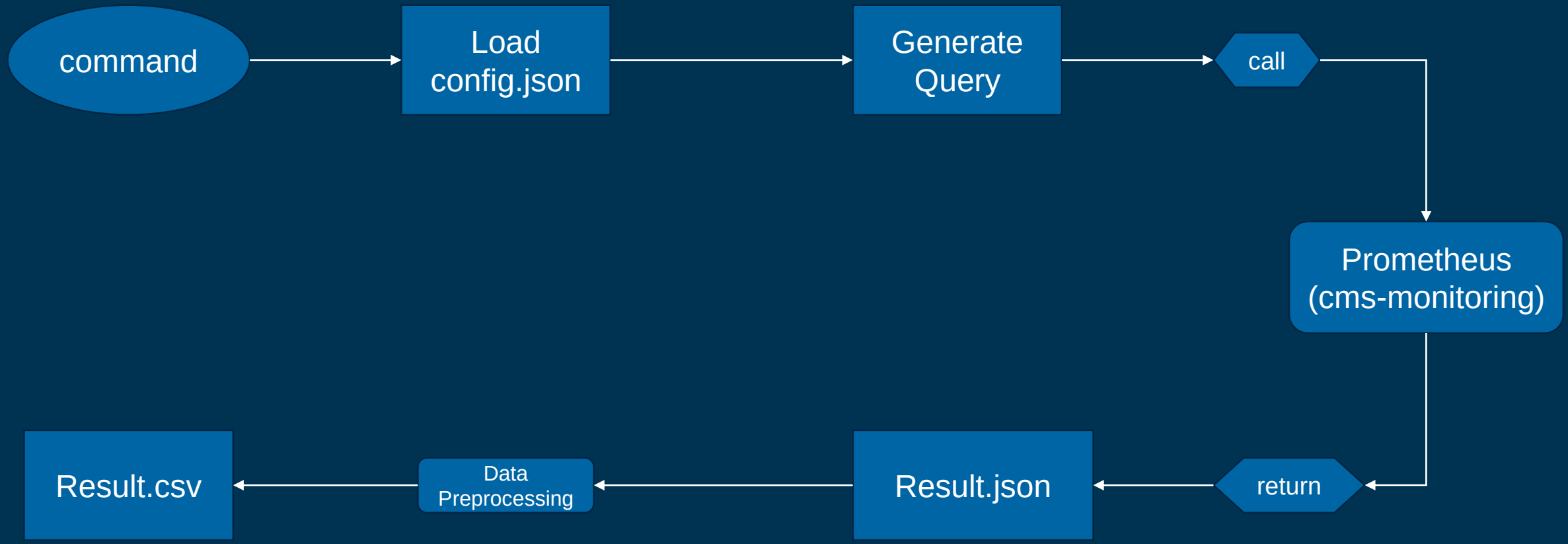
A collective anomaly occurs where single datapoints looked at in isolation appear normal. When you look at a group of these datapoints, however, unexpected patterns, behaviors, or results become

Overview of Kubernetes (k8s)

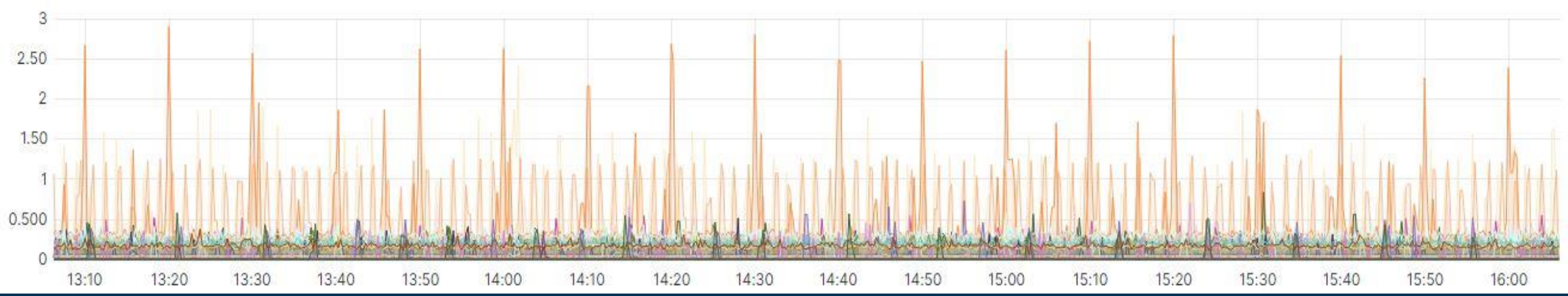
- **Master Node**
- The master node is the control plane of a Kubernetes cluster, responsible for managing the worker nodes and orchestrating the deployment and scaling of applications.
- **Worker Node**
- Worker nodes listen to the API Server for new work assignments; they execute the work assignments and then report the results to the Kubernetes Master node.
- **Pods**
- Pods are the smallest deployable units in Kubernetes, encapsulating one or more containers that share storage and network resources. They ensure that the containers within them operate together and can communicate easily.
- **Namespace**
- Namespaces provide a way to divide cluster resources between multiple users or applications, creating virtual clusters within a physical cluster. This helps in organizing resources and managing access control effectively.



Data Collection



Data Collection - Parameters



Filtration

- Env = k8s-prod
- App = kube-eagle
- Namespace & Container

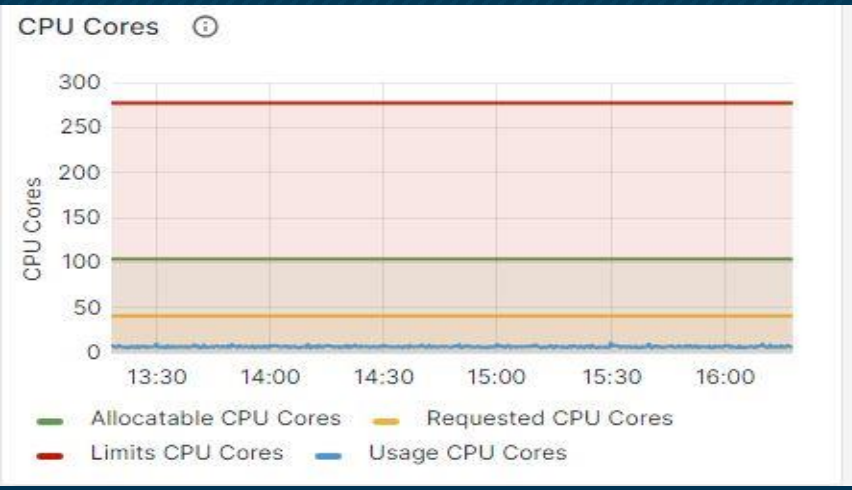
Metrics

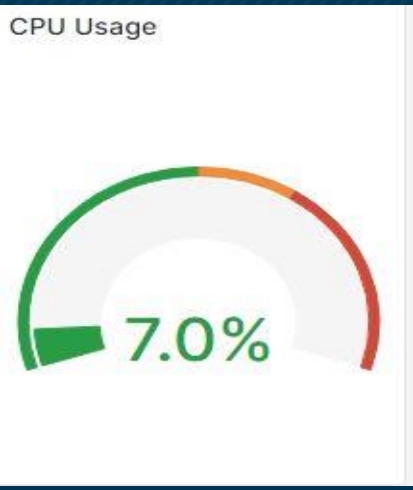
- CPU Usage
- Memory Usage
- Requests

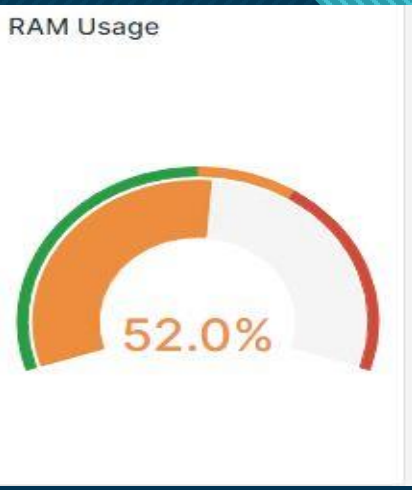
Time

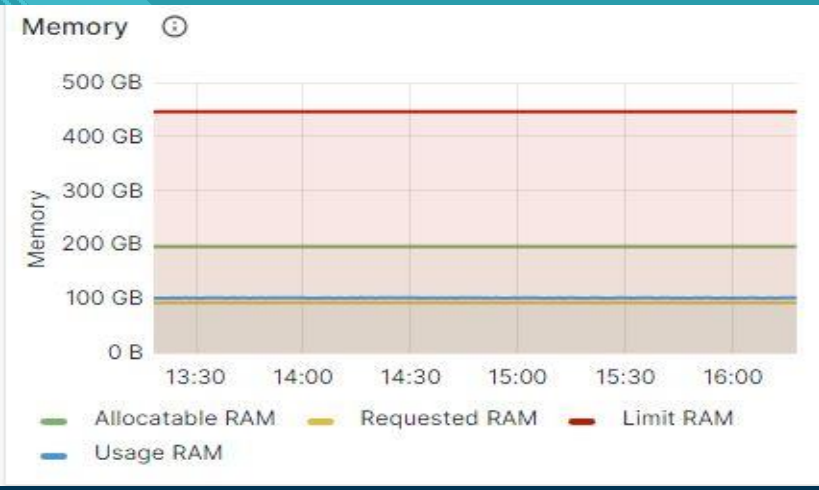
- **Start = 1718398800**
- **End = 1720990800**
- **Step = 15s**

Data Collection - Insight & Optimize Code



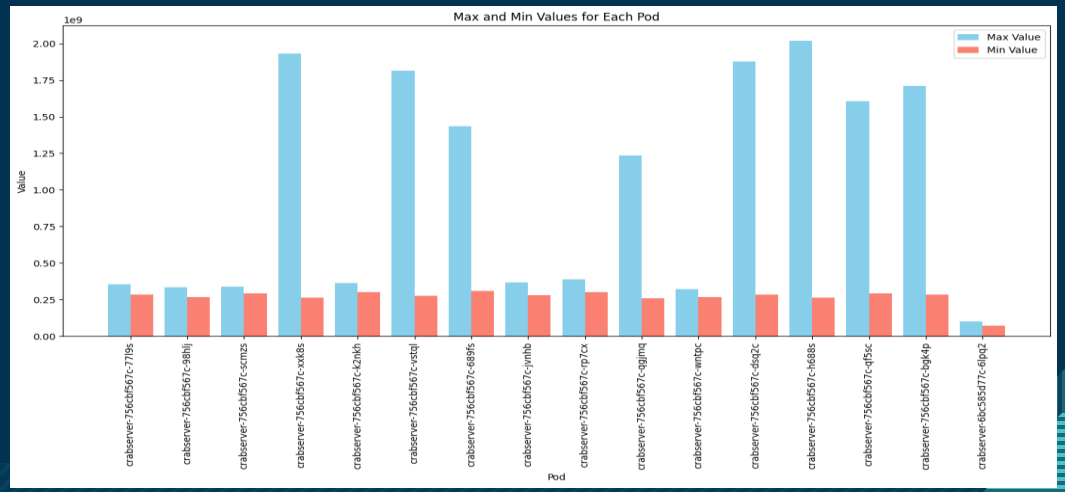






- python3 dataCollectorCurl.py --config config.json --start 2024-06-15T00:00:00Z --end 2024-07-15T00:00:00Z --step 15s --output_dir outputCSV
- config.json

```
        "prometheus_url": "http://cms-monitoring-ha1.cern.ch:30428/prometheus/api/v1/query_range",
        "metrices": [
    "eagle_pod_container_resource_usage_memory_bytes",
    "eagle_pod_container_resource_usage_cpu_cores"
  ],
```



LSTM Autoencoder

Training Phase:
- Trained on time series data representing normal operational behavior from CMSWEB services.
- Learns to encode and decode sequences, minimizing reconstruction errors on normal data.

Encoding:
- Time series data is encoded into a lower-dimensional representation, capturing essential features.

Decoding:
- Encoded data is decoded back to its original time series form.

Reconstruction Error:
- The difference between the original and reconstructed time series is calculated.
- Low error indicates normal behavior; high error suggests an anomaly.

Anomaly Detection:
- During operation, the system reconstructs new time series data.
- Anomalies are detected when reconstruction errors exceed a predefined threshold.

Goal

- AELSTM
- AEFC
- AECNN
- FORCNN

- ❑ Performance Evaluation and Computational Efficiency
- ❑ Detecting anomalies on live data
- ❑ Alert Mechanism
- ❑ Deployment

Thank You!

Q/A

nasir.hussain@cern.ch