IceCube - Neutrinos in Deep Ice Final Presentation

Group 12

Michael Saxer (saxermi1)

Andrés Mock (mockand1)

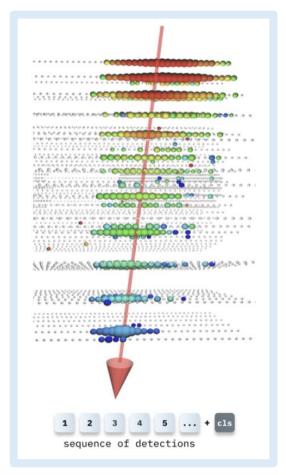
Mike Gasser (gassemik)

Fabian Stepinski (stepifab)

Machine Learning Operations FS 2025

## Purpose/Task of the ML System

- Neutrinos are abundant particles in our universe emitted by violent astrophysical events
- Hard to dected since they rarely interact with matter
- The <u>IceCube Neutrino Observatory</u>
  - A detector array spanning a cubic kilometer of Antarctic ice
  - Detects the directions of neutrino traces using photon detections (3D Unit Vector)
  - Enables determination of the neutrinos' origin, thereby helping locate astrophysical events
- Disclaimer: Our aim is NOT to optimize ML performance, but to explore how the IceCube data processing might be structured



3D Unit Vector

### Overview of Components/Pipeline/System Setup

### **Data Emitter**

#### [Raw IceCube Dataset]

#### **Emitter Container**

- Batches and Serialized Events
- Pushes to Redis Queue
- Send Heartbeat

#### **Redis Container (Message Broker)**

Stored Serialized Event Batches

### Processing / Model / Monitoring

#### **Consumer Container**

- Pops from Redis
- Preprocesses
- Calls Model
- Sends Heartbeat

#### MLflow Tracker

#### Model Server Container

- REST API (/predict)
- Model Inference
- Health Check (/health)
- Sends Hearbeat



Inference Logging (Consumer)

#### **Prometheus Container**

- Scrapes/Metrics Endpoints on all Containers

#### **Grafana Container**

- Visualizes Heartbeat, Throughput, inference Metrics
- Alerts if Heartbeat is missing or high latency

## Key Features of the MLOps Pipeline [1/2]

Tool	Raison d'être
GiHub Actions	<ul> <li>Automatic Checks of Code Quality</li> <li>Automatic requirments.txt check and update (Experimental)</li> <li>Automatic Container Build and upload to GHCR</li> </ul>
Docker	<ul> <li>Easy to deploy</li> <li>Ensures reproducibility</li> <li>Allows containerization of each component</li> </ul>
Redis Community Edition	<ul> <li>Easy to use and performant que system</li> <li>Relatively fast deployment</li> </ul>
MongoDB	Used to track Events
Promotheus	<ul> <li>Scrapes metrics (total requests, latency, etc.) from the different containers (including heartbeat)</li> <li>Can define alert rules</li> </ul>
Grafana	<ul> <li>Dashboard build upon Promotheus</li> <li>Displays Real Time Metrics</li> </ul>
Flask	Server used to serve the model via REST API

# Key Features of the MLOps Pipeline [2/2]



**GitHub Actions** 



**MongoDB for tracking Evenents** 



Grafana Dashboard with Real Time Insights



Promotheus (Eg. Automatic Alerts



**«Behind the Scenes» Info Flow** 

### Sources

- <u>Title Slide Picture</u> [Slide 1]
- 3D Unit Vector [Slide 3]
- Ice Pattern [Slide 3]