RICH AI - MDS Team Proposal

Nico Van den Hooff, Mukund Iyer, Rakesh Pandey, Shiva Jena

May 13, 2022

Contents

1 Executive Summary

- *TODO: Mukund to add exec summary*

2 Introduction

- *TODO: Mukund to add exec intro*

3 Data Science Techniques

- *TODO: Change "Figure X" with dynamic figure captions in markdown*

3.1 Data

3.1.1 Data Generation process

The RICH AI dataset was created as part of the NA62 experiment at CERN in Switzerland in 2016, 2017, 2018, and 2021. In order to generate data, several experiment "runs" are performed. For each run, the experiment configuration is fixed and then the following steps are performed:

1. A beam rich in kaon particles is delivered in "bursts" every four or five seconds into the beam and detector set up as shown in Figure X.
2. During a burst, several particle decays occur. Each particle decay has an individual "event" ID associated with it.
3. The product of the decay is accelerated through a chamber of neon gas in the RICH detector and produces/emits a cone of light. The RICH detector is shown in Figure X.

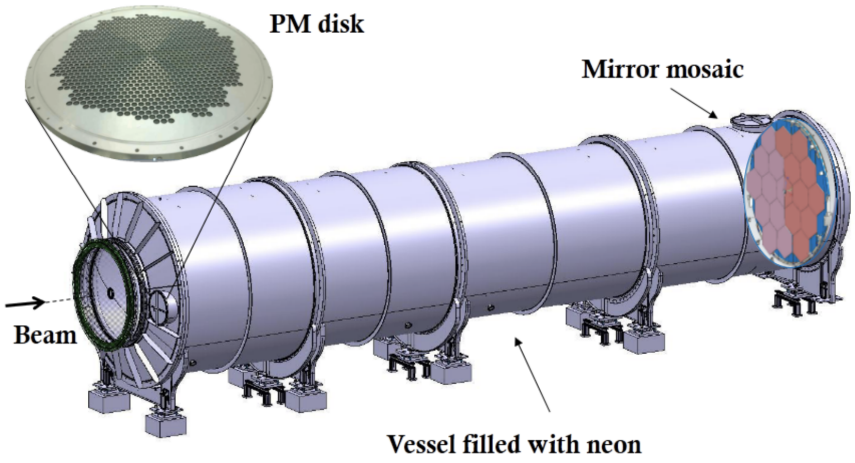

Figure 1: RICH AI beam and detector set up.

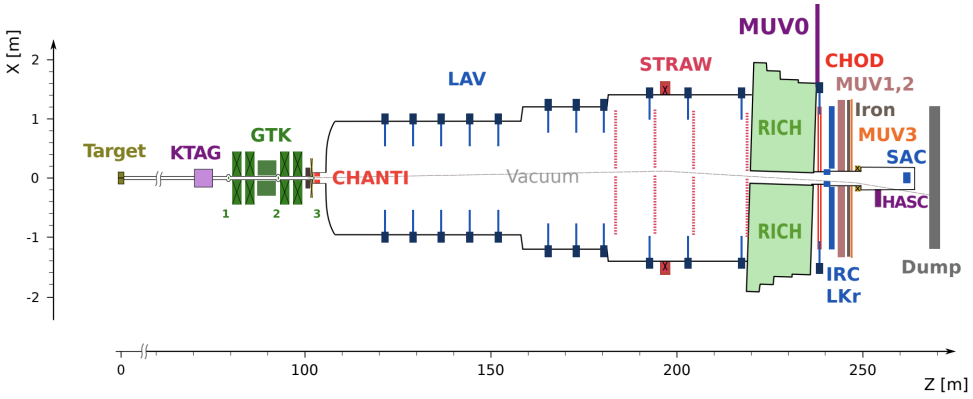

Figure 2: RICH AI beam and detector set up.

4. The cone of light is reflected by a mosaic of mirrors onto an array of photomultiplier tubes ("PM"). In an ideal situation, the cone of light forms a "ring" on the PM array.
5. Each individual PM in the array records whether or not it was hit by light, as well as the time of arrival for each hit of light.
6. The CHOD detector shown in Figure X records the time that the particle decay occurs.

3.1.2 RICH AI dataset

Our project will be utilising the 2018 data generated as part of the NA62 experiment. The 2018 dataset contains information on approximately 11 million particle decays and is stored in HDF5 format. Each particle decay contains the following features:

- Decay label (pion, muon, positron)
- PMT data
  - Hit locations (x, y)
  - Hit times
- Particle momentum
- CHOD time
- Features from TRIUMFs maximum likelihood fit:
  - Ring radius
  - Ring centre (x, y)
  - Ring likelihood
- Metadata (Run ID, Burst ID, Event ID etc.)

The total number of decays for each class is as follows:

- 10,281,301 muon decays
- 1,169,556 pion decays
- 113,112 positron decays

3.2 Machine Learning

3.2.1 Task

The task of our project is to build a machine learning model that can accurately classify an individual particle decay event as a pion, muon, or positron. The rarest and most interesting decay is a pion, whereas muon and positron decays are more common and less interesting. We note that the rarity of pion decays is demonstrated by the imbalanced dataset.

3.2.2 TRIUMF's Current Approach

TRIUMF currently employs an analytical approach without the use of machine learning to classify particle decays. Specifically, maximum likelihood estimation is used to fit a ring to the PM hit data, and the ring radius is used to classify the particle decay. With this methodology, TRIUMF achieves the following results:

1. Pion efficiency of 95% (true positive rate for pion classification)
2. Misclassification of a pion as a muon 1% of the time.

3.2.3 Goals

In building our machine learning model we have the following goals:

1. Achieve a pion efficiency of at least 95% or greater.
2. Reduce pion/muon misclassification by 10x to 0.01% of the time.

3.2.4 Metrics

In building our machine learning model, we will attempt to maximize pion efficiency, which is defined as:

$$\text{Pion efficiency} = \frac{\text{total number of pions detected}}{\text{total number of pions in dataset}}$$

In comparing multiple machine learning models, we will utilize receiver operating characteristic ("ROC") curves.

3.2.5 Baseline model

We have decided to use a gradient boosted tree as our baseline model. Specifically, we will try both XGBoost and LightGBM. These models will be trained on raw feature data, and the one that performs better will be selected as the baseline mode to use in assessing the performance of our more complex models.

3.2.6 Deep learning models

In building a more complicated machine learning model we will employ deep learning. We have identified two potential models that may work well in achieving our task and goals:

1. PointNet or PointNet++
2. Graph Convolutional Neural Networks ("Graph CNN")

The first model architecture is called PointNet, which was originally designed to work with point cloud coordinate data (x, y, z). We will first try out the "vanilla" version of PointNet, and if needed we will also implement the more complex updated architecture called PointNet++.

The second model architecture is called a Graph CNN. Depending on the performance of our PointNet model, we may or may not implement this architecture for our task.

3.2.7 Implementation

The baseline model will be implemented with the python libraries for XGBoost or LightGBM, respectively. The deep learning models will be built with PyTorch and PyTorch Geometric.

4 Communication and timeline

4.1 Communication between TRIUMF and MDS

The following modes of communication exist for communication between the two teams:

- TRIUMF/MDS slack channel
- Thursday Zoom meetings between TRIUMF and MDS
- Email

4.2 Overall Project Timeline

The total time allocated for our capstone project is eight weeks. We propose the following high level project timeline:

Week 1 (May 2 to May 6)

- MDS hackathon
- Initial meetings with TRIUMF to learn about data generation process, machine learning options and helper scripts provided
- Brainstorming problem solutions
- Prepare and complete proposal presentation to MDS faculty

Week 2 (May 9 to May 13)

- Perform exploratory data analysis
- Create input data pipeline for machine learning models

- Begin implementation of machine learning models
- Prepare and submit formal written proposal to TRIUMF

Week 3, 4, 5 (May 16 to June 3)

- Build machine learning models

Week 6 (June 13 to June 17)

- Evaluate models on test set against target metrics
- If needed, further iterate and improve models
- Modularize code
- Create final deliverable python package

Week 7 (June 20 to 24)

- Document code base and steps required to run in production
- Write final project report

Week 8 (June 27 to June 30)

- Submit final product to TRIUMF and perform handover session

5 References

- *TODO: References (bibliography)*