ICECUBE - NEUTRINOS IN DEEP ICE

A MINI PROJECT REPORT

Submitted by

BODDUKURI JESHWANTH 211420104042

PRIYADHARSHAN S 211420104205

SURAJ R 211420104275

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University,Chennai)

MAY 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University,Chennai)

BONAFIDE CERTIFICATE

Certified that this project report ―IceCube - Neutrinos in Deep Ice‖ is the bonafide work of **BODDUKURI JESHWANTH 211420104042, PRIYADHARSHAN S 211420104205, SURAJ R 211420104275** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.L.JABASHEELA,M.E.,Ph.D .,	**Mrs. C. JACKULIN M.E.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	ASSISTANT PROFEESOR
DEPARTMENT OF CSE,	DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,	PANIMALAR ENGINEERING COLLEGE,
NAZARETHPET,	NAZARETHPET,
POONAMALLEE,	POONAMALLEE,
CHENNAI.	CHENNAI.

Certified that the above candidate(s) were examined in the Mini ProjectViva-Voce Examination held on..........................

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kindwords and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt**. **C. VIJAYARAJESWARI , Dr. C . SAKTHIKUMAR , M.E. , Ph.D.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our **Principal Dr. K. Mani, M.E., Ph.D.** for histimely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department , **Dr. L.JABASHEELA, M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my Project Guide **Mrs. C. JACKULIN M.E.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

NAME OF THE STUDENTS

BODDUKURI JESHWANTH (211420104042)

PRIYADHARSHAN S (211420104205)

SURAJ R (211420104275)

TABLE OF CONTENTS

ABSTRACT

The IceCube Neutrino Observatory, located at the South Pole, has the ability to detect high-energy neutrinos that pass through the Earth. However, to fully understand the origin of these neutrinos, it is crucial to determine their direction of arrival. This is achieved by reconstructing the direction of the rged particles produced by the interaction of the neutrino with the ice surrounding the detector. The reconstruction process involves using the timing and location of the light produced by the charged particles, known as Cherenkov radiation, to determine the direction of the neutrino. This method has been highly successful in reconstructing the direction of neutrinos from astrophysical sources such as supernovae and blazars. Recent developments in reconstruction algorithms have significantly improved the accuracy and precision of the direction reconstruction. Additionally, the addition of new detection modules to the IceCube detector has increased its sensitivity to lower energy neutrinos, expanding the range of neutrino sources that can be studied. The reconstructed directions of neutrinos detected by IceCube have provided valuable insights into the high-energy universe, including the identification of the first high-energy neutrino source, a blazar in 2018. Continued efforts to improve the reconstruction algorithms and expand the capabilities of the detector will further enhance our understanding of the most energetic phenomena in the universe.

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The IceCube Neutrino Observatory, located at the South Pole, is a cutting-edge research facility that is dedicated to studying high-energy neutrinos. Neutrinos are subatomic particles that are extremely difficult to detect because they interact very weakly with matter. However, they provide a unique window into the most energetic phenomena in the universe, such as supernovae, black holes, and gamma-ray bursts.One of the key challenges in studying neutrinos is determining their direction of arrival. To address this challenge, the IceCube detector uses a process called direction reconstruction. This involves detecting the charged particles produced by the interaction of the neutrino with the ice surrounding the detector and using this information to determine the direction of the neutrino.Recent developments in reconstruction algorithms have significantly improved the accuracy and precision of the direction reconstruction process. This has enabled researchers to identify the first high-energy neutrino source, a blazar, in 2018. Additionally, the addition of new detection modules to the IceCube detector has increased its sensitivity to lower energy neutrinos, expanding the range of neutrino sources that can be studied.The reconstructed directions of neutrinos detected by IceCube have provided valuable insights into the high-energy universe. By studying the properties of these neutrinos, researchers can learn about the sources that produced them and the physical processes that govern their interactions. This information can help us to better understand some of the most fundamental questions in astrophysics, such as the nature of dark matter and the origins of cosmic rays.In conclusion, the IceCube Neutrino Observatory is an essential tool for studying the most energetic phenomena in the universe. By improving the direction reconstruction process and expanding the capabilities of the detector, researchers will continue to make groundbreaking discoveries about the nature of the universe and its most extreme environments.

1.2 PROBLEM DEFINITION

- The problem addressed by the IceCube Neutrino Observatory is the difficulty in detecting and studying high-energy neutrinos, which are subatomic particles that interact weakly with matter but provide a unique window into the most energetic phenomena in the universe. The key challenge is determining their direction of arrival, which is crucial for identifying their sources and understanding their interactions.
- The IceCube detector uses a process called direction reconstruction to infer the direction of the neutrino from the direction of the charged particles produced by their interaction with the surrounding medium, but accuracy and precision are limited by various factors.

CHAPTER 2

LITERATURE SURVEY

Title : “IceCube collaboration , The IceCube Data Acquistion System :

Signal Capture, Digitization and Timestamping “ by M.Ackermann (2009)

Description : This survey paper gives information about the ability of IceCube to identify point sources of neutrinos, should they exist, depends on the pointing resolution of reconstructed muon trajectories.

Title : “IceCube Collaboration,The Fundamental interactions of Neutrino”

by Lake Louise (2008)

Description : IceCube is a cubic kilometer neutrino telescope under construction at the South Pole. The primary goal is to discover astrophysical sources of high energy neutrinos. We describe the detector and present results on atmospheric muon neutrinos from 2006 data collected with nine detector strings.

Title : “ The design and performance of IceCube DeepCore ” by Astroparticle physists (2012)

Description : In this paper we describe the design and initial performance of DeepCore.The features of DeepCore will increase IceCube's sensitivity to neutrinos from WIMP dark matter annihilations, atmospheric neutrino oscillations, galactic supernova neutrinos, and point sources of neutrinos in the

Title : “Energy Reconstruction Methods in the IceCube Neutrino Telescope ”

By Jinska Telescope Research centre (2014)

Description : Accurate measurement of neutrino energies is essential to many of the scientific goals of large-volume neutrino telescopes. The fundamental observable in such detectors is the Cherenkov light produced by the transit through a medium of charged particles created in neutrino interactions.

Title : “ IceCube Collaboration, Measurement of Atmospheric Neutrino Oscillations with IceCube DeepCore” by Joshua Highnit (2018)

Description: We present a measurement of the atmospheric neutrino oscillation parameters using three years of data from the IceCube Neutrino Observatory. The DeepCore infill array in the center of IceCube enables detection and reconstruction of neutrinos produced by the interaction of cosmic rays in the Earth's atmosphere

Title : “IceCube collaboration, Measurement of Atmospheric Tau Neutrino Apperance with DeepCore Ice ” by M.G.Aartsen (2019)

Description : We present a measurement of atmospheric tau neutrino appearance from oscillations with three years of data from the DeepCore subarray of the IceCube Neutrino Observatory. This analysis uses atmospheric neutrinos from the full sky with reconstructed energies between 5.6 and 56 GeV to search for a statistical excess of cascadelike neutrino events

Title : “AMANDA Collaboration, Muon track reconstruction and data selection techniques ” by AMANDA Nuclear Instrumentation Centre (2004)

Description : The primary goal of this detector is to discover astrophysical sources of high-energy neutrinos. A high-energy muon neutrino coming through the earth from the Northern Hemisphere can be identified by the secondary muon moving upward through the detector.

Title : “The IceCube Upgrade – Design and Science Goals ” by PoS ICRC (2019)

Description : The IceCube Neutrino Observatory at the geographic South Pole has reached a number of milestones in the field of neutrino astrophysics. The achievements of IceCube include the discovery of a high-energy astrophysical neutrino flux, and the temporal and directional correlation of neutrinos with a flaring blazar.

Title : “ A next-generation optical sensor for IceCube-Gen2 ” by PoS ICRC (2021)

Description : The electronics design must meet the space constraints posed by the mechanical design as well as the power consumption and cost considerations from over 10,000 optical modules being deployed. This contribution presents forward-looking solutions to these design considerations. Prototype modules will be installed and integrated in the IceCube Upgrade.

Title : “IceCube collaboration, Using convolutional neural networks to reconstruct energy of GeV scale IceCube neutrinos,” by Jessie Micallef (2021)

Description : Sensitivity to oscillation parameters, dependent on the distance traveled over the neutrino energy (L/E), is limited in IceCube by the resolution of the arrival angle (which determines L) and energy (E). Event reconstruction improvements can therefore directly lead to advancements in oscillation results. This work uses a Convolutional Neural Network (CNN) to reconstruct the energy of 10-GeV scale neutrino events in IceCube, providing results with competitive resolutions and faster runtimes than previous likelihood-based methods.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

- Authors mention *RETRO algorithm*, used by IceCube team, which is quite sophisticated, but requires up to 40 seconds of CPU time for proper reconstruction of just one event.
- Considering the number of the events, it's impossible to use this algorithm at the observatory (due to amount of computational infrastructure needed) or analyse real time data.
- CNNs don't work particularly well with low-energy events.
- This approach lead to serious degradation of information for low-energy events

3.2 PROPOSED SYSTEM: GRAPH NEURAL NETWORK

- By adopting graphs as the input data structure, the idea of convolution generalizes from the application of filters on the rigid structure of grids to abstract mathematical operators that utilize the interconnection of nodes in its computation.
- This allows such models to naturally incorporate an irregular geometry directly in the edge specification of the graph, without imposing artificial constraints.

3.3 DEVELOPMENT ENVIRONMENT

SOFTWARE REQUIREMENT

1. TENSORFLOW
2. PYTORCH
3. PLOTLY
4. MATPLOTLIB
5. PYTHON
6. SCIKIT-LEARN
7. PANDAS & NUMPY
8. PYTORCH-LIGHTENING
9. REACT
10. DJANGO
11. HTML & CSS

HARDWARE REQUIREMENT

i. Processor: Minimum 4 GHz

ii. Memory (RAM): 8 GB

iii. Hard Drive: 128 GB

iv. GPU (Optional)

CHAPTER 4

SYSTEM DESIGN

4.1 UML DIAGRAMS

4.1.1 USE CASE DIAGRAM

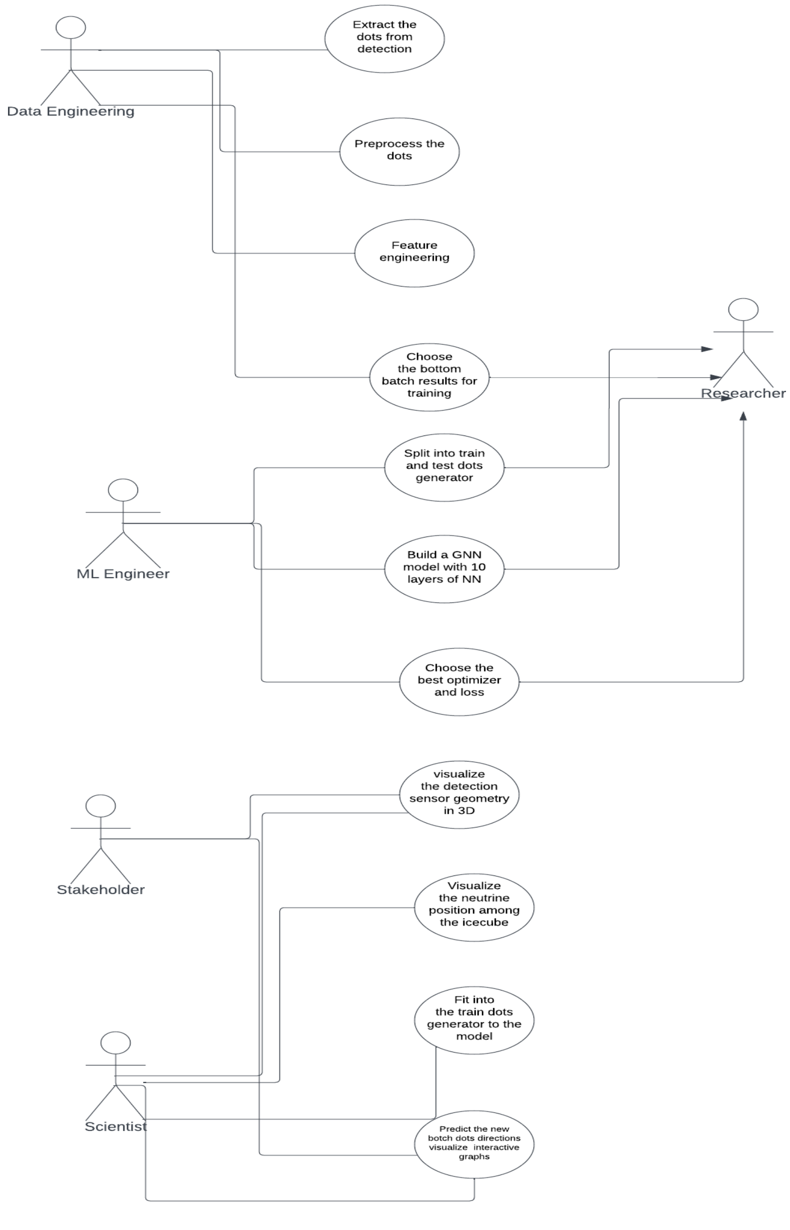

Fig 4.1.1Use case diagram for Neutrino direction prediction Using GNN

This use case diagram refers to activities done within users and their corresponding use cases in the Icecube Neutrino detection.

4.1.2 CLASS DIAGRAM

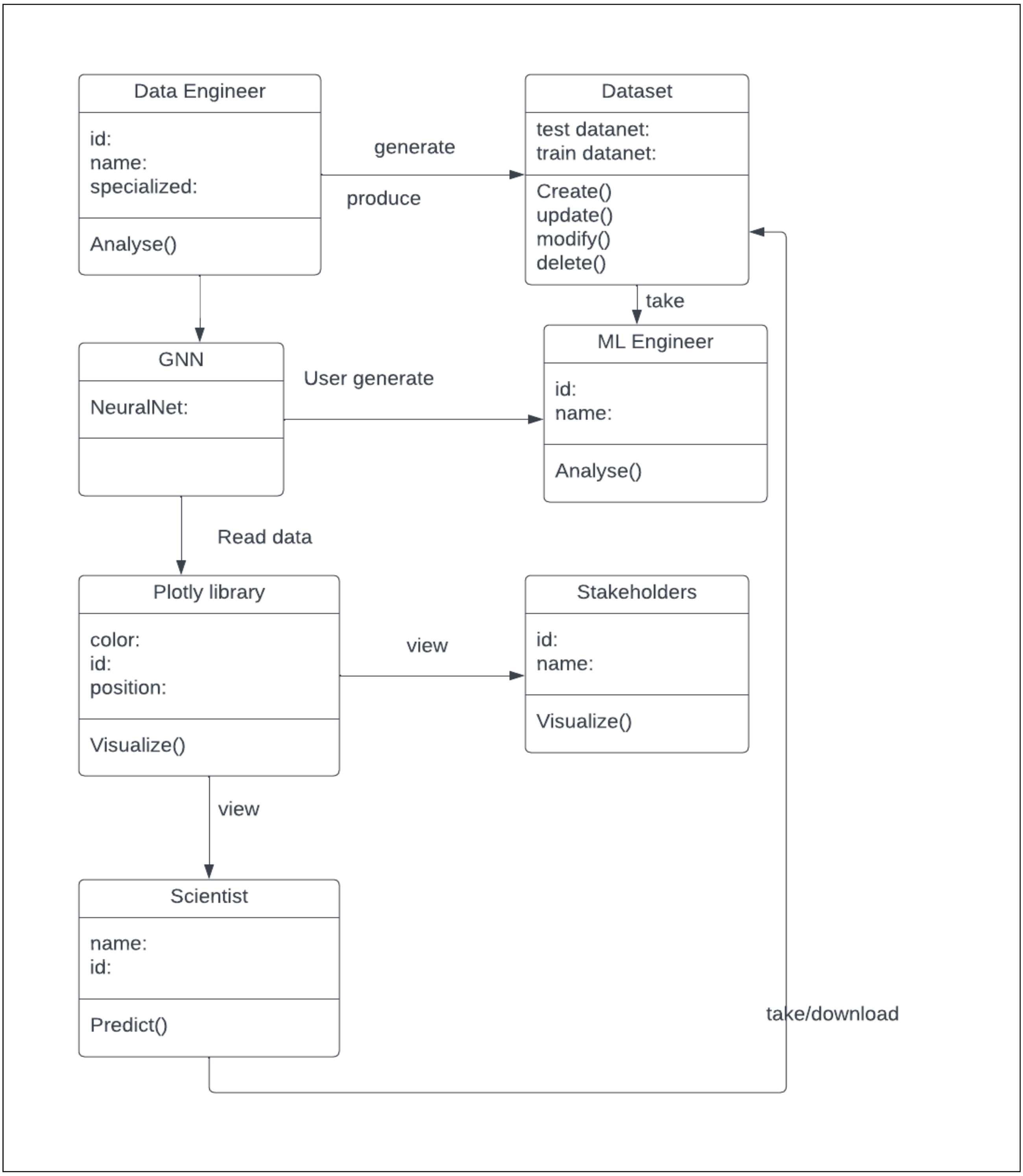

Fig 4.1.2 Class diagram for Neutrino direction prediction Using GNN

The class diagram for Neutrino direction prediction using GNN gives classes for datasets and neural network.

4.1.3 SEQUENCE DIAGRAM

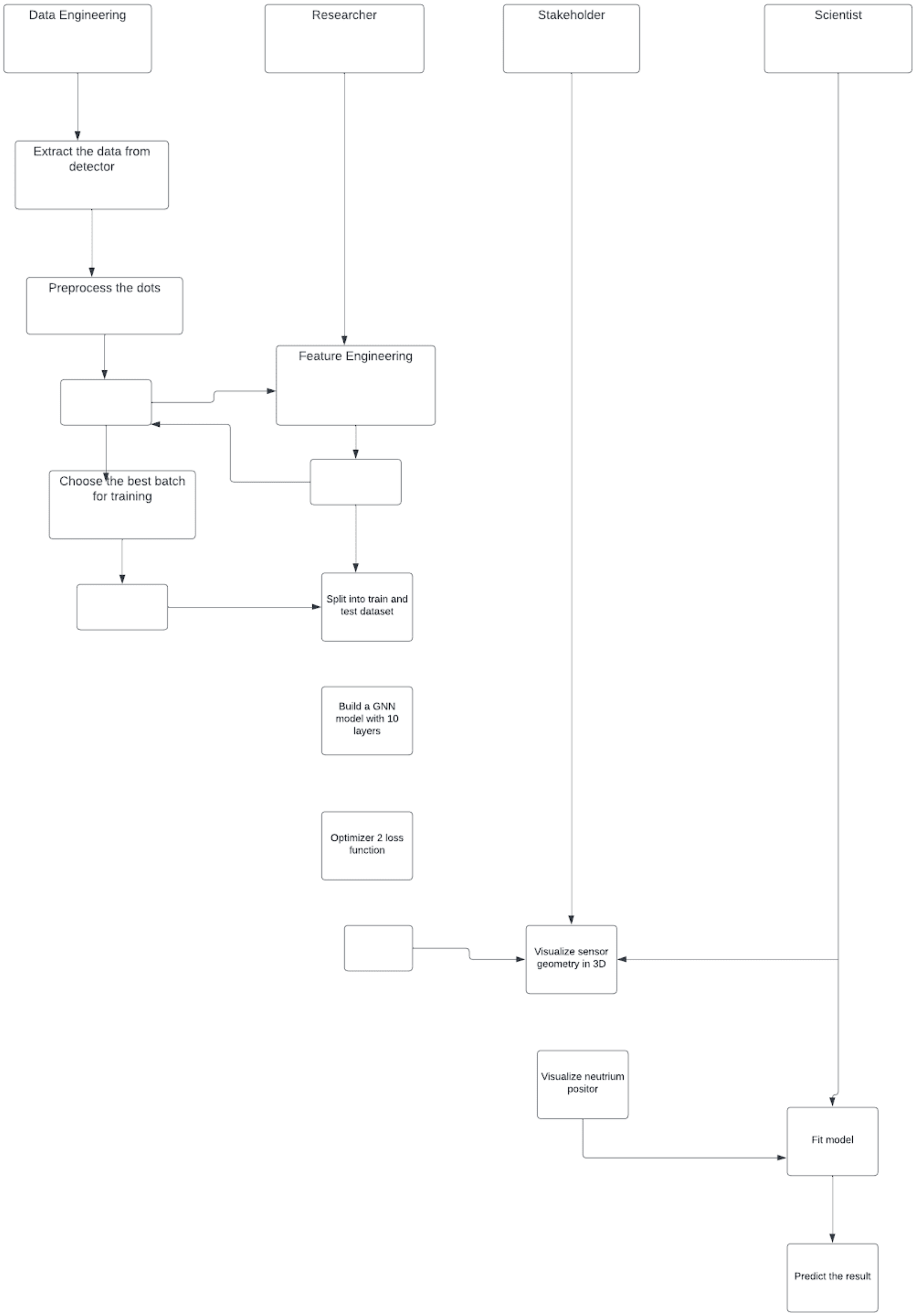

Fig 4.1.3 Sequence case diagram for Neutrino direction prediction Using GNN

This sequence case diagram refers to activities done within users and their corresponding use cases in the Icecube Neutrino detection.

4.1.4 ACTIVITY DIAGRAM:

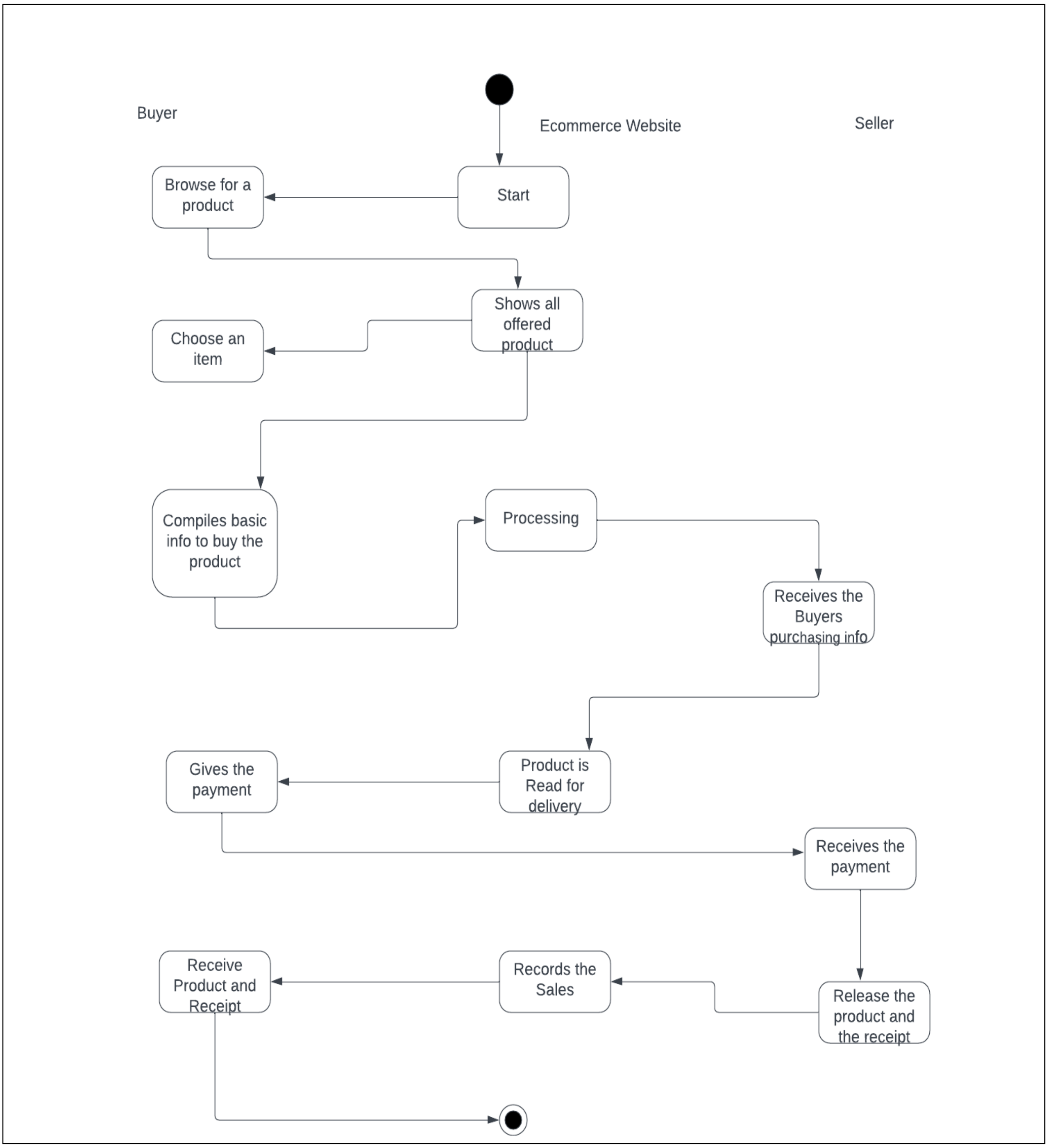

Fig 4.1.4 Activity diagram for Neutrino direction prediction Using GNN

This activity diagram refers to activities done within users and their corresponding use cases in the Icecube Neutrino detection.

4.2 DATAFLOW DIAGRAM

4.2.1 DATAFLOW DIAGRAM LEVEL 0

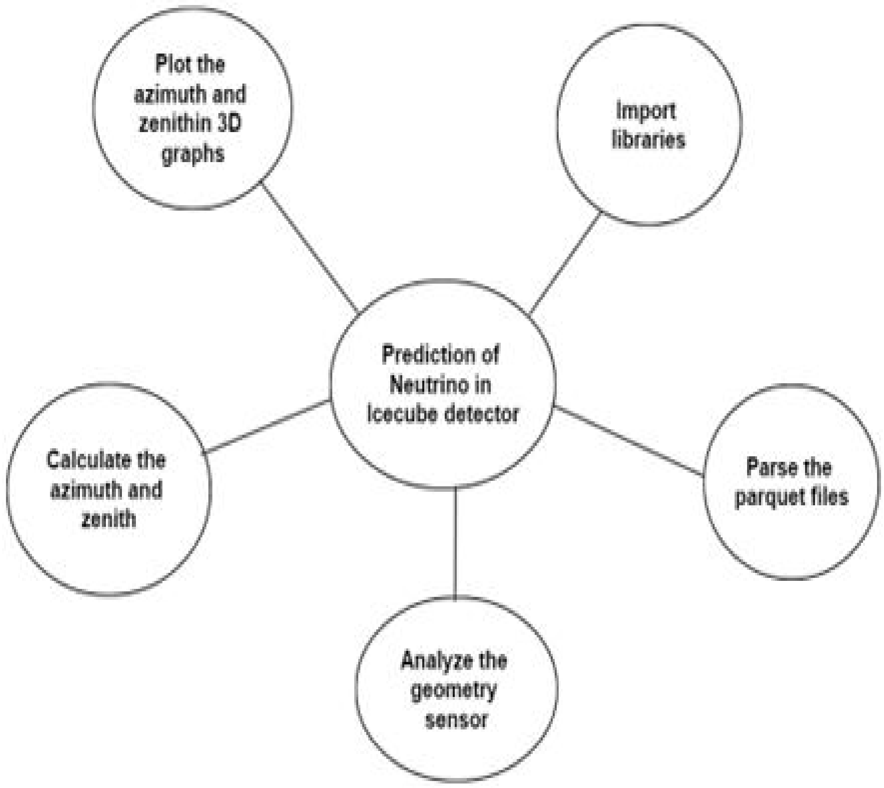

Fig 4.2.1 Data flow diagram level 0

The zero level of data flow diagram of Icecube Neutrino detection which the users can communicate with. DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its entities.

4.2.2 DATAFLOW DIAGRAM LEVEL 1

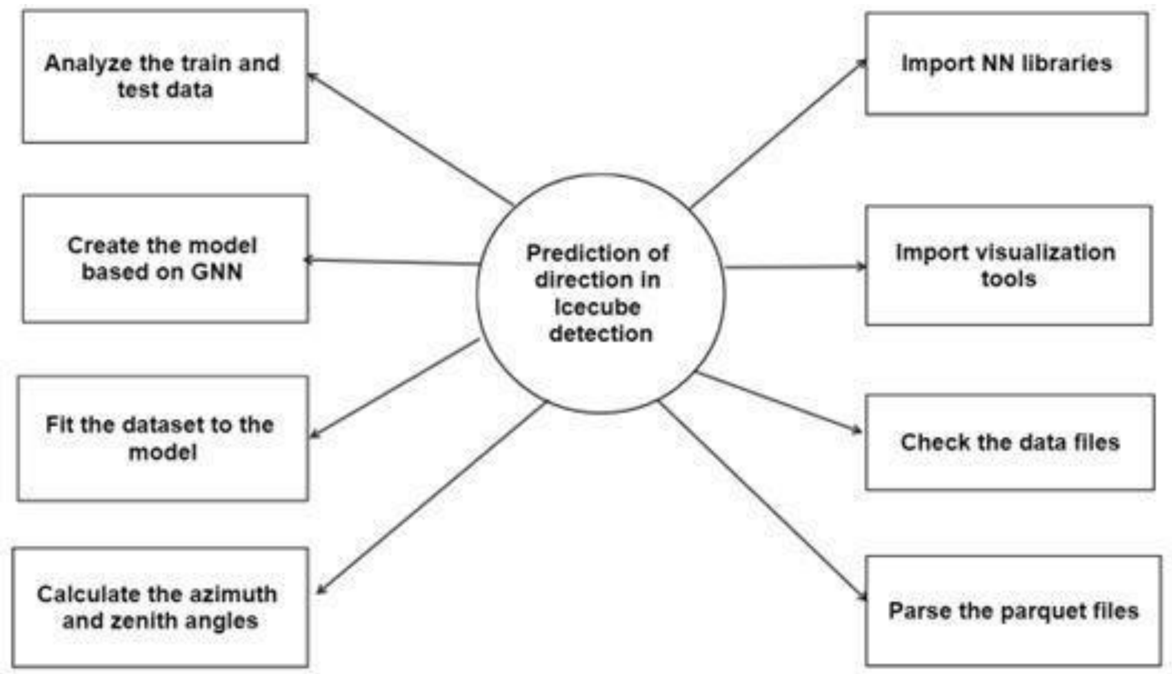

Fig 4.2.2 Data flow diagram level 1

The first level of data flow diagram of Icecube Neutrino detection which the users can communicate with. DFD Level 1 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its entities.

4.2.3 DATA FLOW DIAGRAM LEVEL 2

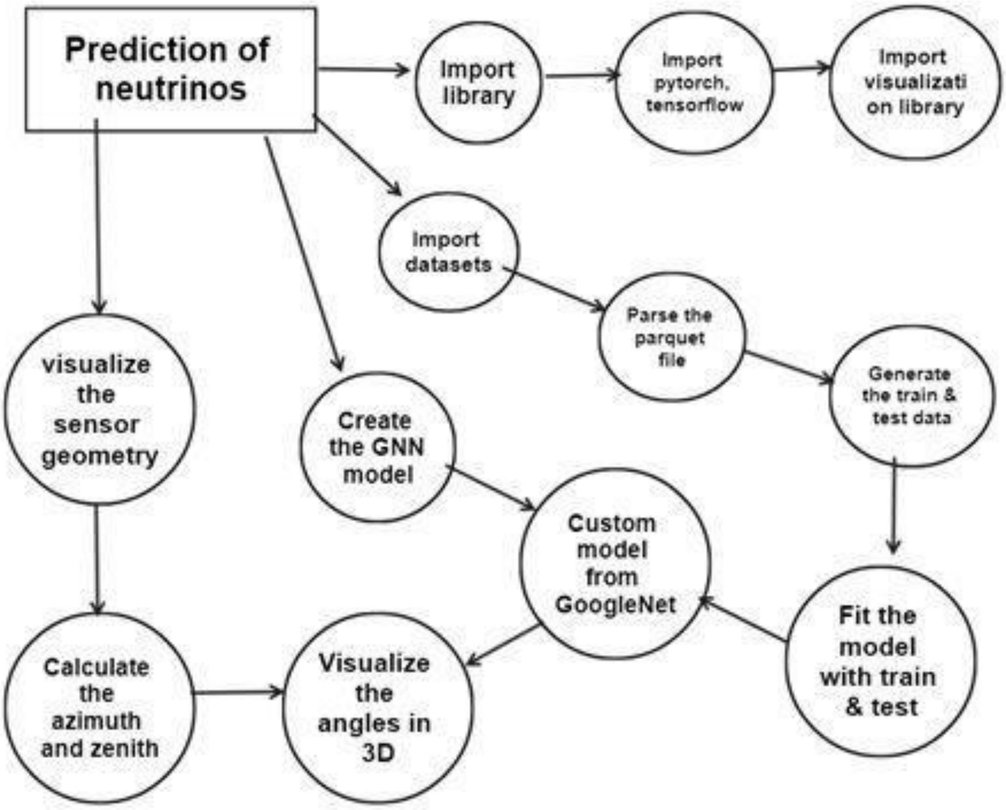

Fig 4.2.3 Data flow diagram level 2

The second level of data flow diagram of Icecube Neutrino detection which the users can communicate with. DFD Level 2 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its entities.

CHAPTER 5

SYSTEM ARCHITECTURE

5.1 ARCHITECTURE OVERVIEW

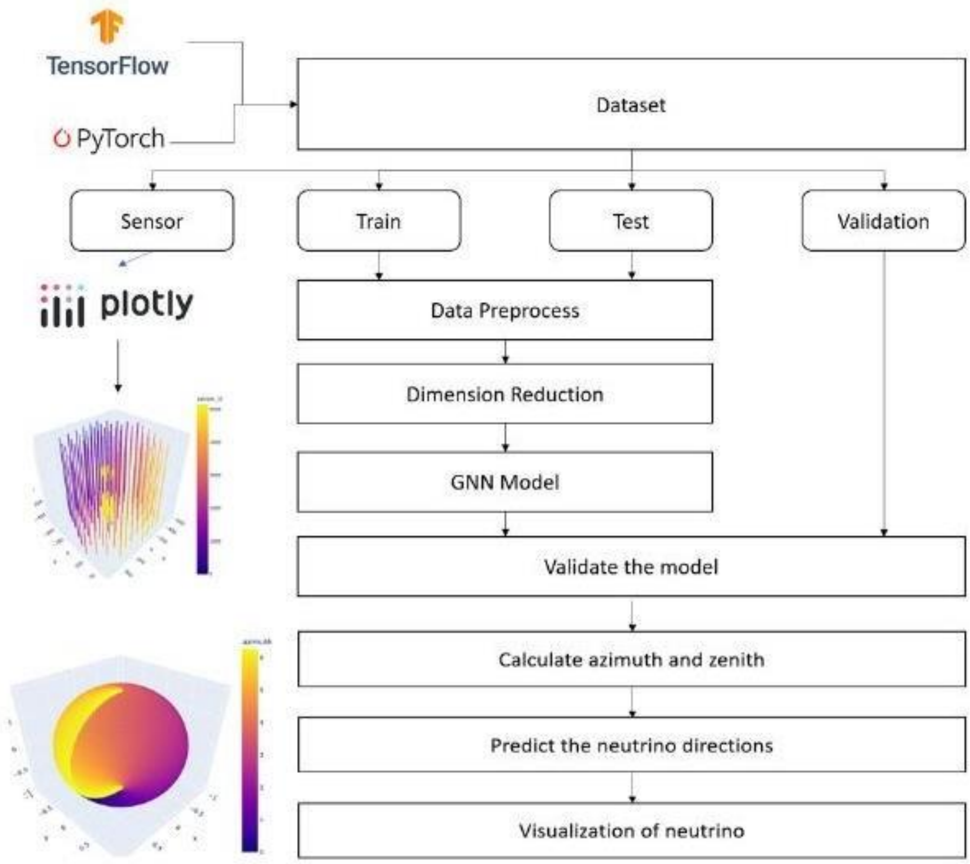

Fig 5.1 Architecture diagram for IceCube Neutrino prediction

By adopting graphs as the input data structure, the idea of convolution generalizes from the application of filters on the rigid structure of grids to abstract mathematical operators that utilize the interconnection of nodes in its computation.

5.2 MODULE DESCRIPTION

IceCube Neutrino prediction consists of 6 modules:

They are

5.2.1 Importing Libraries

5.2.2 Extracting Datasets

5.2.3 Pre-process the dataset

5.2.4 Build a GNN model

5.2.5 Event Reconstruction

5.2.6 Neutrino angles prediction

Importing Libraries

In this project we are using PyTorch and Tensorflow as our ML and Deep Learning library for model building. Pandas and Numpy are used to parse and preprocess the parquet files.

Extracting Datasets

Initially we read the parquet file from the local computer and parse those parquet files and convert them into pandas dataframe. After that we preprocess the parquet file using pandas and numpy.

Pre-process the dataset

Pandas: This is a powerful data manipulation library that can be used to clean, transform, and reshape data.

Numpy: This is a numerical computing library that can be used for mathematical operations on data, such as statistical analysis, linear algebra, and matrix manipulation.

Scikit-learn: This is a machine learning library that provides a variety of pre-processing modules such as scaling, normalization, and encoding of categorical variables.

TensorFlow: This is a deep learning library that can be used for pre-processing of image, audio, and text data.

Dimension reduction

Principal Component Analysis (PCA)

Autoencoders

Linear Discriminant Analysis (LDA)

Independent Component Analysis (ICA)

Build a GNN model

Graph Neural Networks (GNNs) are a powerful approach for modeling structured data, such as social networks, molecules, and recommendation systems. GNNs operate on graph-structured data by recursively aggregating information from a node's neighbors, enabling the learning of rich representations of the underlying graph structure. Building GNN models involves selecting an appropriate architecture, defining a message passing scheme, and optimizing the model param

Event Reconstruction

Event reconstruction is the process of analyzing raw data from detectors to extract meaningful information about physical processes. It involves identifying signals, reconstructing particle trajectories, measuring their energies and momenta, and inferring underlying physics. Its goal is to understand the fundamental laws of nature.

Neutrino angles prediction

Neutrino angle prediction involves using machine learning techniques to analyze neutrino data and predict the direction from which the neutrinos originated. This is achieved by combining information from multiple neutrino detectors and applying advanced algorithms to accurately estimate the angle of incidence.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 DATA PRE-PROCESS

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import torch
import glob

train_meta_path = "D:/Neutrinos/train_meta.parquet"
test_meta_path = "D:/Neutrinos/test_meta.parquet"
meta_train = pd.read_parquet(train_meta_path, engine='fastparquet')
meta_test = pd.read_parquet(test_meta_path, engine='fastparquet')

meta_train[["first_pulse_index", "last_pulse_index"]].plot.hist(bins=100,
alpha=0.5, figsize=(8, 3))
plt.xlabel('time')
meta_train["delay_pulse_index"] = meta_train["last_pulse_index"] -
meta_train["first_pulse_index"]
plt.hist(meta_train["delay_pulse_index"], bins=100, log=True)
plt.xlabel = "duration”
```

```
plt.ylabel = "count_cases"
plt.grid()

test = pd.read_parquet("D:/Neutrinos/test/batch_661.parquet")
print("Number of events in the test set: ", len(test))
print("unique events: ", len(test.index.unique()))
test.head()

import math
def coord(azimuth, zenith):
    x = math.cos(azimuth) * math.sin(zenith)
    y = math.sin(azimuth) * math.sin(zenith)
    z = math.cos(zenith)
    return dict(x=x, y=y, z=z, azimuth=azimuth, zenith=zenith)


res_data = []
for azimuth in np.linspace(0, 2*math.pi, num=100):
    for zenith in np.linspace(0, math.pi, num=100):
        res_data.append(coord(azimuth, zenith))


angles_df = pd.DataFrame(res_data)
```

```
angles_df.head()


# preprocessing.py
def prepare_sensors():
    sensors = pd.read_csv(INPUT_PATH / "sensor_geometry.csv").astype(
        {
            "sensor_id": np.int16,
            "x": np.float32,
            "y": np.float32,
            "z": np.float32,
        }
    )
    sensors["string"] = 0
    sensors["qe"] = 1

    for i in range(len(sensors) // 60):
        start, end = i * 60, (i * 60) + 60
        sensors.loc[start:end, "string"] = i

        # High Quantum Efficiency in the lower 50 DOMs -
https://arxiv.org/pdf/2209.03042.pdf (Figure 1)
        if i in range(78, 86):
```

```
        start_veto, end_veto = i * 60, (i * 60) + 10
        start_core, end_core = end_veto + 1, (i * 60) + 60
        sensors.loc[start_core:end_core, "qe"] = 1.35

    # https://github.com/graphnet-team/graphnet/blob/b2bad25528652587ab0cdb7cf2335ee254cfa2db/src/graphnet/models/detector/icecube.py#L33-L41
    # Assume that "rde" (relative dom efficiency) is equivalent to QE
    sensors["x"] /= 500
    sensors["y"] /= 500
    sensors["z"] /= 500
    sensors["qe"] -= 1.25
    sensors["qe"] /= 0.25

    return sensors
```

6.2 BUILDING GNN MODEL

```
def calculate_distance_matrix(xyz_coords: Tensor) -> Tensor:

    diff = xyz_coords.unsqueeze(dim=2) - xyz_coords.T.unsqueeze(dim=0)
    return torch.sqrt(torch.sum(diff**2, dim=1))

class EuclideanGraphBuilder(nn.Module):


    def__init__(
        self,
        sigma: float,
        threshold: float = 0.0,
        columns: List[int] = None,
    ):
        # Base class constructor
        super().__init__()

        # Check(s)
        if columns is None:
            columns = [0, 1, 2]
```

```
        # Member variable(s)
        self._sigma = sigma
        self._threshold = threshold
        self._columns = columns

    def forward(self, data: Data) -> Data:

        # Constructs the adjacency matrix from the raw, DOM-level data and
        # returns this matrix
        xyz_coords = data.x[:, self._columns]

        # Construct block-diagonal matrix indicating whether pulses belong to
        # the same event in the batch
        batch_mask = data.batch.unsqueeze(dim=0) ==
data.batch.unsqueeze(dim=1)

        distance_matrix = calculate_distance_matrix(xyz_coords)
        affinity_matrix = torch.exp(
            -0.5 * distance_matrix**2 / self._sigma**2
        )

        # Use softmax to normalise all adjacencies to one for each node
```

```
        exp_row_sums = torch.exp(affinity_matrix).sum(axis=1)
        weighted_adj_matrix = torch.exp(
            affinity_matrix
        ) / exp_row_sums.unsqueeze(dim=1)

        # Only include edges with weights that exceed the chosen threshold (and
        # are part of the same event)
        sources, targets = torch.where(
            (weighted_adj_matrix > self._threshold) & (batch_mask)
        )
        edge_weights = weighted_adj_matrix[sources, targets]

        data.edge_index = torch.stack((sources, targets))
        data.edge_weight = edge_weights

        return data


class DenseDynBlock(nn.Module):
    """
    Dense Dynamic graph convolution block
    """
```

```
    def __init__(self, in_channels, out_channels=64, sigma=0.5):
        super(DenseDynBlock, self).__init__()
        self.GraphBuilder = EuclideanGraphBuilder(sigma=sigma)
        self.gnn = SAGEConv(in_channels, out_channels)

    def forward(self, data):
        data1 = self.GraphBuilder(data)
        x, edge_index, batch = data1.x, data1.edge_index, data1.batch
        x = self.gnn(x, edge_index)
        data1.x = torch.cat((x, data.x), 1)
        return data1
```

MODEL: GRAPH NEURAL NETWORK

```
class MyGNN(nn.Module):
    """
    Dynamic graph convolution layer
    """
    def __init__(self, in_channels, hidden_channels, out_channels, n_blocks):
        super().__init__()
        self.n_blocks = n_blocks
        self.head = SAGEConv(in_channels, hidden_channels)
        c_growth  = hidden_channels
```

```
        self.gnn = nn.Sequential(*[DenseDynBlock(hidden_channels+i*c_growth, c_growth)
                                   for i in range(n_blocks-1)])
        fusion_dims = int(hidden_channels * self.n_blocks + c_growth * ((1 + self.n_blocks - 1) * (self.n_blocks - 1) / 2))
        self.linear = nn.Linear(fusion_dims, out_channels)


    def forward(self, data):
        x, edge_index, batch = data.x, data.edge_index, data.batch
        data.x = self.head(x, edge_index)
        feats = [data.x]
        for i in range(self.n_blocks-1):
            data = self.gnn[i](data)
            feats.append(data.x)
        feats = torch.cat(feats, 1)
        x = pyg_nn.global_mean_pool(feats, data.batch)
        out = F.relu(self.linear(x))
        return out
```

CHAPTER 7

REPORTS

7.1 VISUALIZATION CHARTS

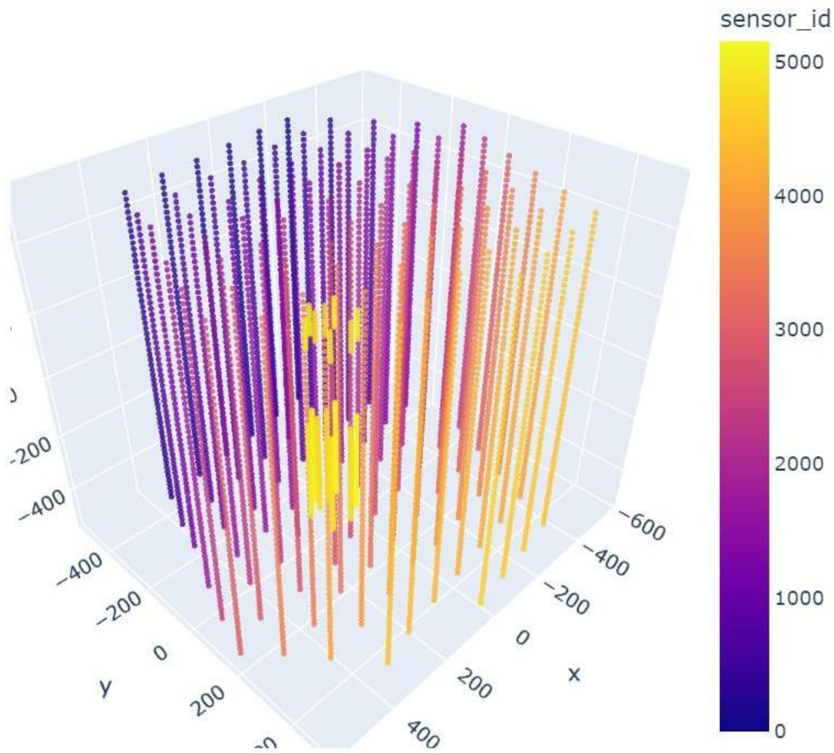

Fig. 7.1.1 3D model of IceCube detector sensor position

x, y, and z positions for each of the 5160 IceCube sensors. The row index corresponds to the sensor_idx feature of pulses. The x, y, and z coordinates are in units of meters, with the origin at the center of the IceCube detector. The coordinate system is right-handed, and the z-axis points upwards when standing at the South Pole.

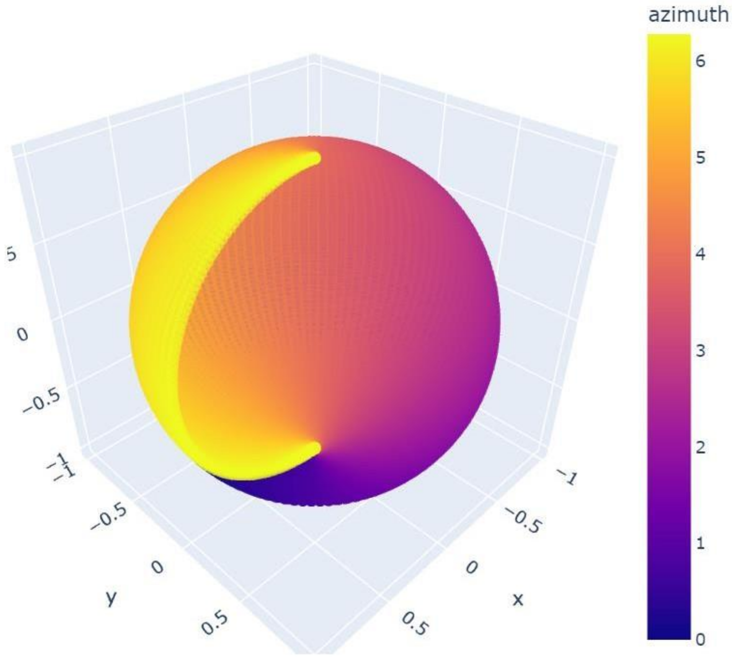

Fig. 7.1.2 Represents the azimuth angle in the centre of Icecube detector

Theory: The x, y, and z coordinates are in units of meters, with the origin at the center of the IceCube detector. The coordinate system is right-handed, and the z-axis points upwards when standing at the South Pole. You can convert from these coordinates to azimuth and zenith with the following formulas (here the vector (x,y,z) is normalized)

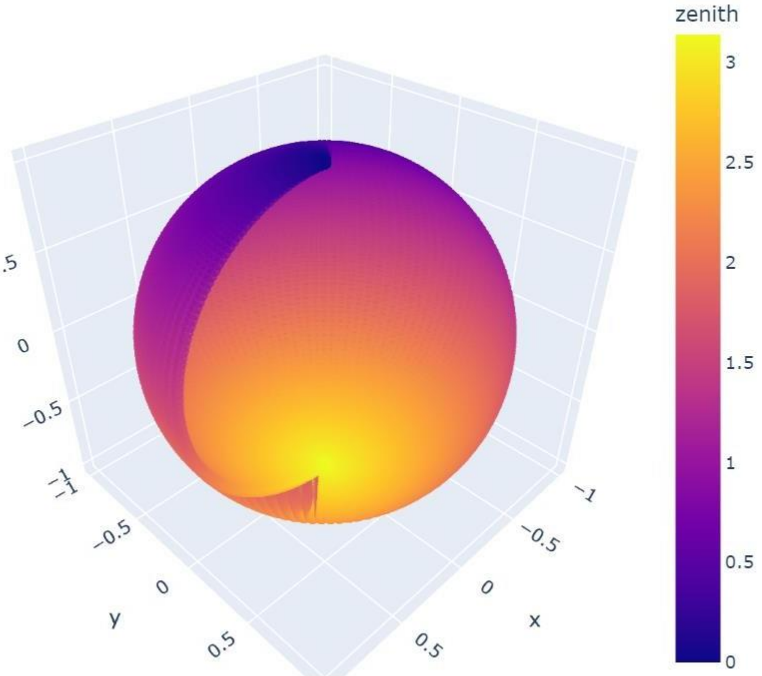

Fig. 7.1.3 Represents the zenith angle in the centre of Icecube detector

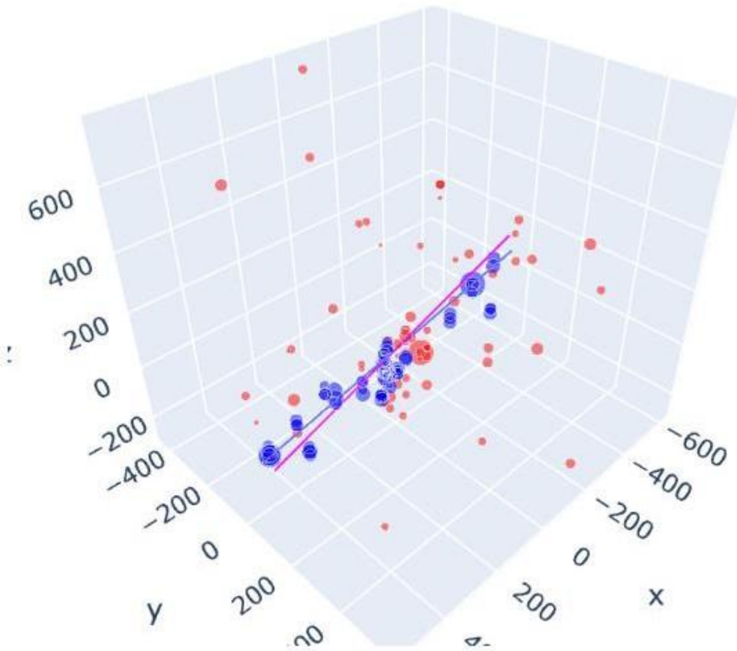

Fig 7.1.4 IceCube-Neutrino Path Detection with 3D Projection

It describes the Path detection of the particular neutrino on the IceCube detector.

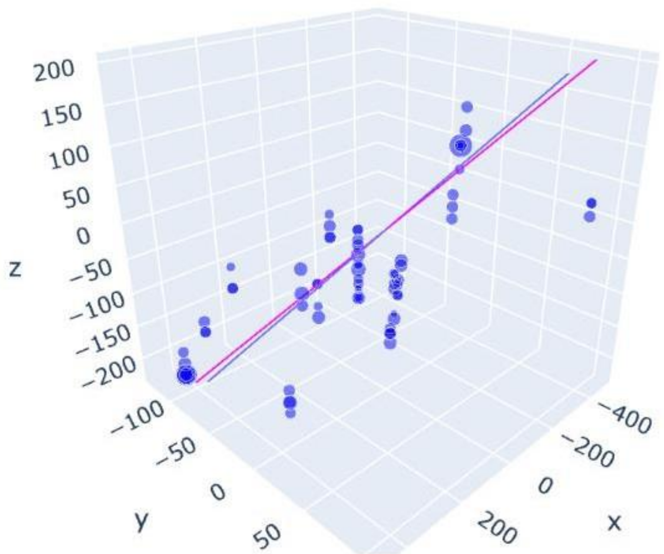

Fig 7.1.5 Plotting event # 100867570

Magenta line is predicted trajectory / Blue line is true trajectory

7.2 FINAL RESULTS

7.2.1 Linear Regression

event_id	azimuth	zenith
2092	3.177299	1.423469
7344	3.16187	1.553881
9482	3.153506	1.566424

Table 7.2.1 azimuth and zenith angles for the 3 events using Linear Regression

Linear regression models the relationship between dependent and independent variables by fitting a straight line to data points.

7.2.2 Edge Conv. And transformer interface

event_id	azimuth	zenith
2092	0.041696	1.280855
7344	3.454236	2.499653
9482	4.708986	1.556324

Table 7.2.2 azimuth and zenith angles for the 3 events using Edge Conv. And transformer interface

The Edge Convolutional layer and Transformer layer both take a graph as input, but use different methods to process it.

7.2.3 TensorFlow LSTM Model Inference

event_id	azimuth	zenith
2092	1.862817	1.432417
7344	3.329414	2.44187
9482	4.646359	1.568192

Table 7.2.3 azimuth and zenith angles for the 3 events using TensorFlow LSTM

To infer with a TensorFlow LSTM model, input sequences are fed to the model, which outputs predicted values for each time step.

7.2.4 GraphNeT Baseline

event_id	azimuth	zenith
2092	0.481458	1.527498
7344	3.497879	2.469359
9482	4.570243	1.538434

Table 7.2.4 azimuth and zenith angles for the 3 events using GraphNET Baseline

GraphNet is a baseline model for graph classification, using Graph Convolutional Networks (GCNs) to learn graph representations.

CHAPTER 8

8.1 CONCLUSION

- We propose a GNN-based reconstruction algorithm for IceCube events named dynedge, that can be applied to any IceCube event. We have selected simulated low-energy data used for studies of atmospheric neutrino oscillations in IceCube as our dataset, and in this energy range, we have benchmarked dynedge against the state-of-the-art reconstruction and classification algorithms as a proof of concept.
- Dynedge offers substantial improvements to both T /C and ν/μ classifications in the entire low energy range. In the energy range 1 GeV–30 GeV, relevant to standard atmospheric $\nu\mu$–νr oscillation studies, dynedge exhibits a 15–20% improvement in reconstruction of energy, zenith, direction, and interaction vertex.

8.2 FUTURE ENHANCEMENTS

- In the energy range 1 GeV–30 GeV, relevant to standard atmospheric $\nu\mu$–νr oscillation studies, dynedge exhibits a 15–20% improvement in reconstruction of energy, zenith, direction, and interaction vertex.
- This worsening effect is ascribed to lower statistics in the training set at these energies and a disproportional amount of cascades compared to more energetic (i.e. longer) tracks. Future studies will focus on improving performance in this region. dynedge and retro are both robust against systematic uncertainties in DOM optical efficiency and angular acceptance, as well as the scattering and absorption properties of the bulk ice. dynedge is also robust against random perturbations to inputs such as DOM position, timing, and charge.

- Based on tests of reconstruction speed, dynedge could reconstruct events online at the South Pole. dynedge is also flexible enough to be compatible with the planned IceCube Upgrade featuring new DOM types on new strings, as well as other neutrino detectors with arbitrary geometries.
- This feature may make dynedge particularly useful for the next generation of large neutrino detectors, such as KM3Net and the proposed IceCube Gen2. dynedge is implemented using GraphNeT.

CHAPTER 9

APPENDICES

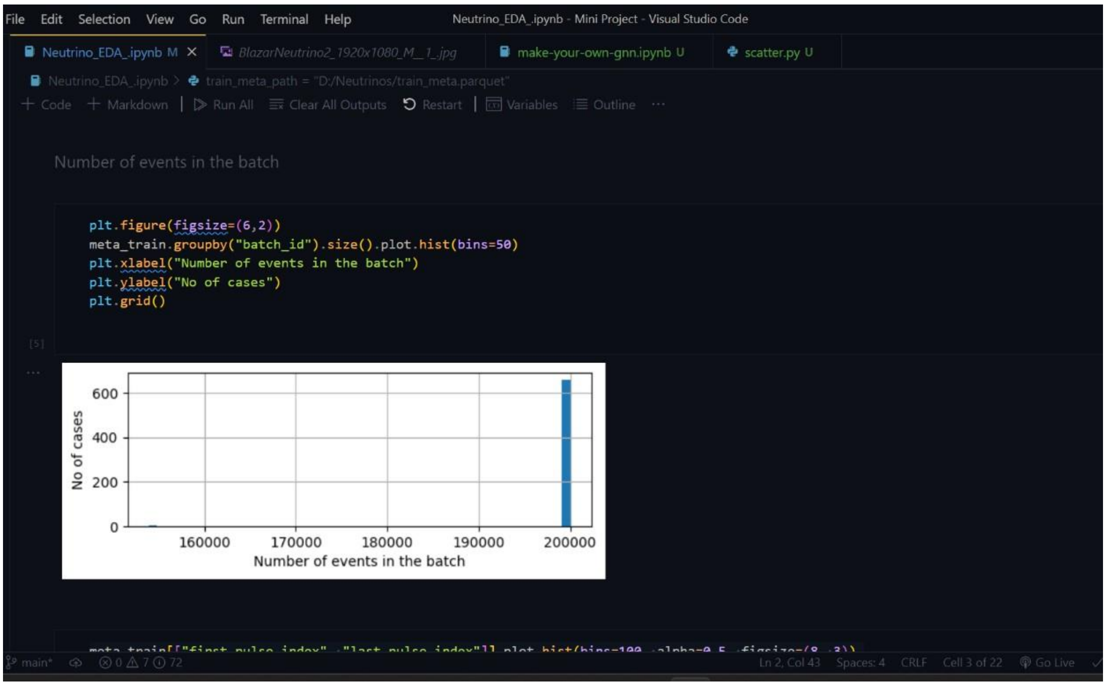

Fig. 9.1 Number of events in the batch

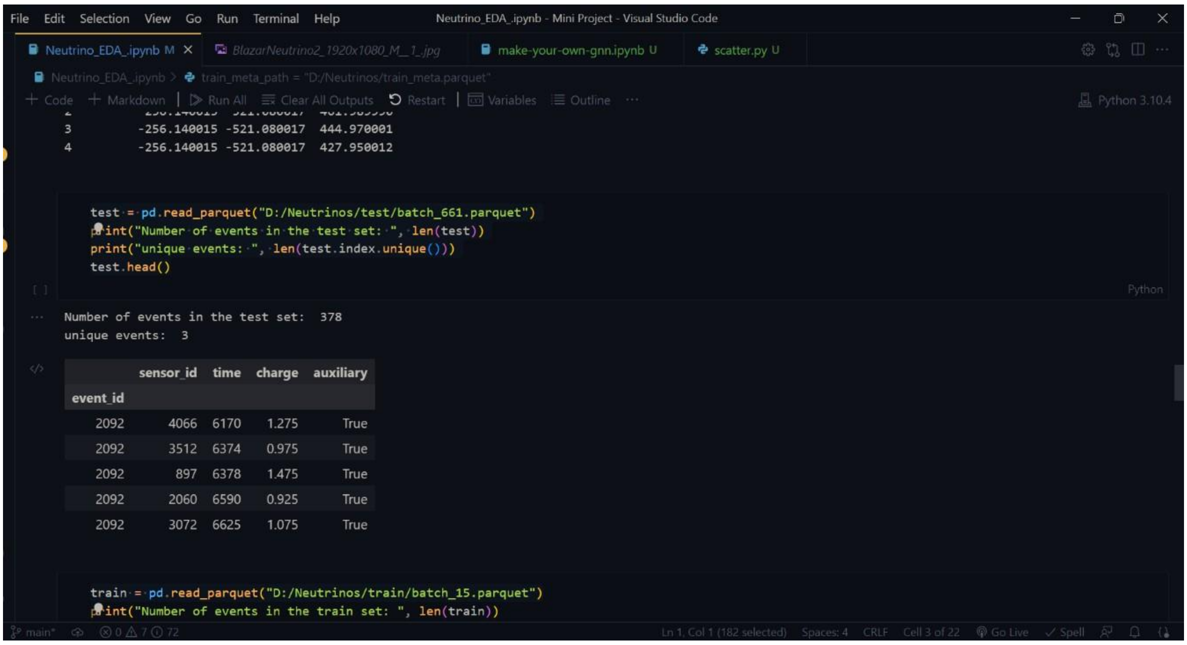

Fig. 9.2 Sensor, Time, Charge, auxiliary events

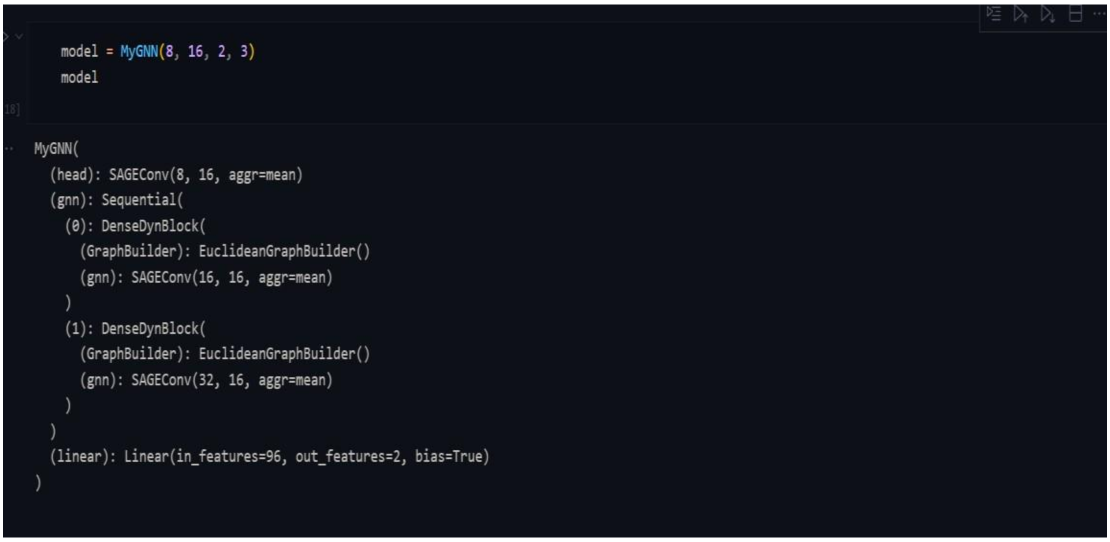

Fig. 9.3 Graph Neural Network Model

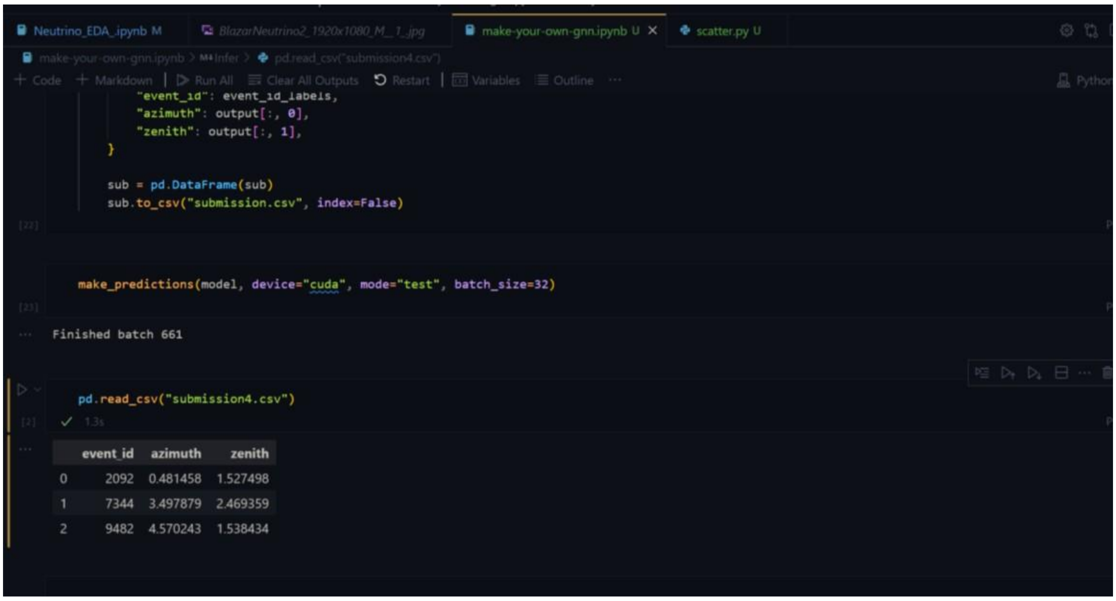

Fig. 9.4 Graph Neural Network prediction of azimuth and zenith angles of events

CHAPTER 10

REFERENCES

1. IceCube collaboration, The IceCube Data Acquisition System: Signal Capture, Digitization, and Timestamping, Nucl. Instrum. Meth. A 601 (2009) 294 [arXiv:0810.4930].
2. IceCube collaboration, Neutrino Physics with the IceCube Detector, in proceedings of the 23rd Lake Louise Winter Institute: Fundamental Interactions, Lake Louise, Alberta, Canada, 18–23 February 2008, pp. 283–287 [arXiv:0806.1717].
3. IceCube collaboration, The Design and Performance of IceCube DeepCore, Astropart. Phys. 35 (2012) 615 [arXiv:1109.6096].
4. F. Halzen and S.R. Klein, IceCube: An Instrument for Neutrino Astronomy, Rev. Sci. Instrum. 81 (2010) 081101 [arXiv:1007.1247].
5. IceCube collaboration, Low energy event reconstruction in IceCube DeepCore, Eur. Phys. J. C 82 (2022) 807 [arXiv:2203.02303].
6. IceCube collaboration, The IceCube Neutrino Observatory: Instrumentation and Online Systems, 2017 JINST 12 P03012 [arXiv:1612.05093].
7. Particle Data Group collaboration, Review of Particle Physics, PTEP 2020 (2020) 083C01.
8. IceCube collaboration, Energy Reconstruction Methods in the IceCube Neutrino Telescope, 2014 JINST 9 P03009 [arXiv:1311.4767].
9. IceCube collaboration, Measurement of Atmospheric Neutrino Oscillations at 6–56 GeV with IceCube DeepCore, Phys. Rev. Lett. 120 (2018) 071801 [arXiv:1707.07081].

10. IceCube collaboration, Measurement of Atmospheric Tau Neutrino Appearance with IceCube DeepCore, Phys. Rev. D 99 (2019) 032007 [arXiv:1901.05366].
11. IceCube collaboration, Measurement of South Pole ice transparency with the IceCube LED calibration system, Nucl. Instrum. Meth. A 711 (2013) 73 [arXiv:1301.5361].
12. AMANDA collaboration, Muon track reconstruction and data selection techniques in AMANDA, Nucl. Instrum. Meth. A 524 (2004) 169 [astro-ph/0407044].
13. IceCube collaboration, The IceCube Upgrade — Design and Science Goals, PoS ICRC2019 (2021) 1031 [arXiv:1908.09441].
14. IceCube-Gen2 collaboration, A next-generation optical sensor for IceCube-Gen2, PoS ICRC2021 (2021) 1062 [arXiv:2107.08837].
15. T. Hastie, R. Tibshirani and J. Friedman, Neural networks, in The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer New York, New York, NY, U.S.A. (2009), pp. 389–416.
16. R. Abbasi et al., A Convolutional Neural Network based Cascade Reconstruction for the IceCube Neutrino Observatory, 2021 JINST 16 P07041 [arXiv:2101.11589].
17. IceCube collaboration, Using convolutional neural networks to reconstruct energy of GeV scale IceCube neutrinos, 2021 JINST 16 C09019 [arXiv:2109.08152].
18. IceCube collaboration, Direction reconstruction using a CNN for GeV-scale neutrinos in IceCube, 2021 JINST 16 C11001.
19. F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner and G. Monfardini, The graph neural network model, IEEE Trans. Neural Networks 20 (2009) 61.
20. N. Choma, F. Monti, L. Gerhardt, T. Palczewski, Z. Ronaghi, P. Prabhat et al., Graph neural networks for icecube signal classification