

Vrije Universiteit Amsterdam



Universiteit van Amsterdam



Master Thesis

Title of the Thesis

Author: student name (student number)

1st supervisor: supervisor name

daily supervisor: supervisor name (company, if applicable)

2nd reader: supervisor name

*A thesis submitted in fulfillment of the requirements for
the joint UvA-VU Master of Science degree in Computer Science*

September 23, 2020

"I am the master of my fate, I am the captain of my soul"

from Invictus, by William Ernest Henley

Abstract

Here goes the abstract of this thesis.

Contents

List of Figures	iii
List of Tables	v
1 Introduction	1
1.1 Situation of Concern	2
1.1.1 State of the Art	3
1.1.2 User Requirements	3
1.2 Research Question	3
2 Data Preparation	5
2.1 Preparation of <i>events</i> dataset	5
2.2 Preparation of <i>noise</i> dataset	5
2.3 Preparation of <i>main</i> dataset	6
3 Data Exploration	7
3.1 Descriptive Statistics	7
3.2 Verification of Bias	8
3.3 Exploration of Interesting Timeslices	8
4 Hit Correlation	13
4.1 Pattern Matrix Replacement	13
4.2 Creation of the Pattern Matrix Dataset	14
4.3 Model Description	14
4.4 Model Training	14
4.4.1 Experiments with Dataset	15
4.4.2 Experiments with Optimizers	15
4.5 Model Evaluation	15

CONTENTS

Bibliography	17
---------------------	-----------

List of Figures

1.1	Artist's impression of the ARCA detector <i>source: https://www.km3net.org</i>	1
1.2	An optical detector (DOM) <i>source: https://www.km3net.org</i>	2
3.1	Correlation matrix of features	9
3.2	Distribution of <code>label</code> column	10
3.3	Verification of Bias	10
3.4	Distribution of event hits per timeslice	11
3.5	Distribution of Timeslice 615	11

LIST OF FIGURES

List of Tables

3.1 Description of columns	8
3.2 Descriptive statistics	9

LIST OF TABLES

1

Introduction

The KM3NeT or the Cubic Kilometer Neutrino Telescope is currently being constructed at the bottom of the Mediterranean Sea. The goal of this telescope is two fold: first is to study high energy neutrinos originating from celestial events such as birth of a neutrino star, supernovae, etc. and second, to study the properties of the neutrino particles produced in the Earth's atmosphere (1). The first goal will be realized with the KM3NeT/ARCA (Astroparticle Research with Cosmics in the Abyss) telescope and the second with KM3NeT/ORCA (Oscillation Research with Cosmics in the Abyss) (1). In this paper, we talk exclusively about KM3NeT/ARCA.

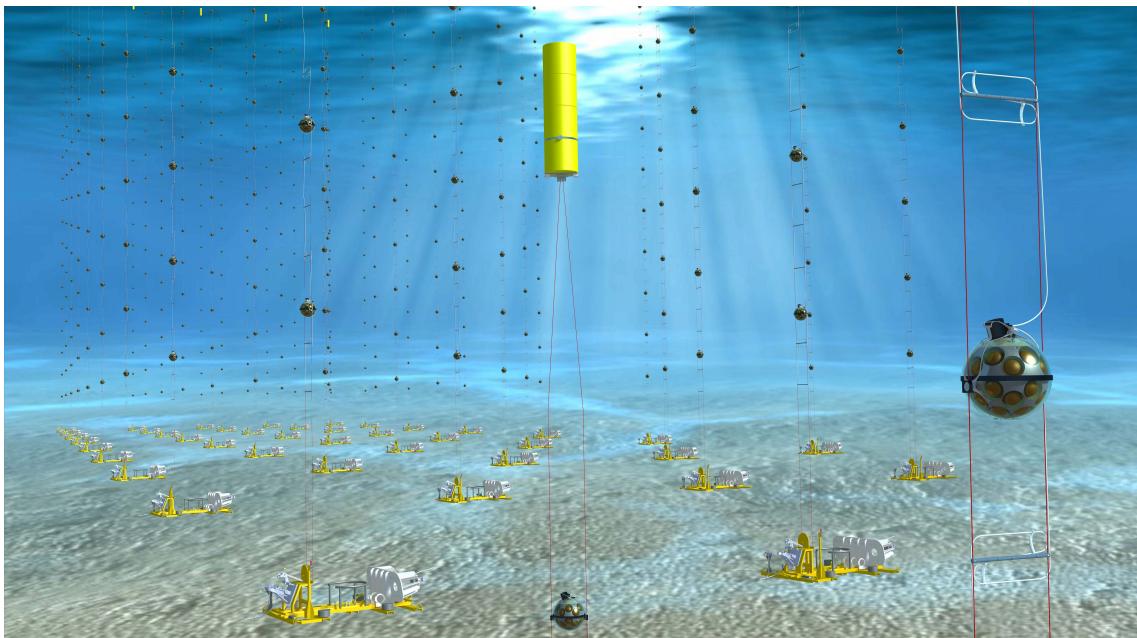


Figure 1.1: Artist's impression of the ARCA detector *source: <https://www.km3net.org>*

1. INTRODUCTION

The ARCA telescope comprises of two “blocks” with a total volume of 1km^3 . Each block consists of 115 spherical detector units (referred to as DOMs henceforth) and each DOM consists of 31 Photo Multiplier Tubes (PMTs) in various spacial arrangement. Figure ?? shows an artist’s impression of ARCA, figure ?? depicts a DOM along with the PMTs inside it.



Figure 1.2: An optical detector (DOM) *source: <https://www.km3net.org>*

The PMTs are sensitive to light or photons, the analog signal for all hits above a certain threshold are digitized. This datapoint consists of a timestamp and the spatial orientation of the DOM (ie. x,y,z coordinates). The digital signals from all PMTs are arranged in 100ms “time slices” and sent to the on-shore facility for further processing (2).

1.1 Situation of Concern

When the high energy neutrino particles interact with surrounding matter, produce an electron and a photon, this phenomenon is known as the Cherenkov Radiation or Cherenkov Light (3). This phenomenon is utilized in the KM3NeT telescopes to detect high energy neutrinos ie. the Cherenkov Light is detected by the PMTs. Unfortunately, there are several sources of noise (in this case, the noise is other light sources), bioluminescence and decay of Potassium 40 (^{40}K) and atmospheric Muons being the primary sources (4).

1.1.1 State of the Art

Due to the high level of noise, data is generated at an extremely high rate of 25GB/sec (1). Due to this high data rate, it must be filtered and selectively stored for further analysis. The state of the art for this task are known as “Event Trigger” algorithms (1, 2). The existing event trigger algorithms namely $L1$ and $L2$ have limitations and can be improved upon (5).

There is a tradeoff between performance and quality of the event trigger algorithms to be realized here. The state of the art $L1$ and $L2$ algorithms are very performant with the ability to filter data in real time however their quality of filtration can be improved. Thus new event trigger algorithms must consider this trade off.

Efforts have already been made to improve the existing event trigger algorithms. (5) proposed and implemented a GPU powered pipeline which utilizes correlation and graph community detection to identify time slices that may contain neutrino hits whilst (4) suggests an alternate using convolutional neural networks.

here is an alternative to introduce related work chapter Next chapter discusses the existing body of work on the matter. Emphasis is put on the work done by (**author?**) since this project seeks to directly improve upon their work.

1.1.2 User Requirements

The primary users of the ARCA are researchers who want to study high energy particles from outer space. The stakeholders are all member institutes involved in the project and by extension all scientists from these institutes who will be working with the data collected.

The requirements of the primary users (and stakeholders) with respect to the data acquisition pipeline are as follows: 1. The accuracy with which time slices are filtered out has to be extremely high. Time slices which are deemed important by event trigger algorithms are stored for further analysis and research. Failure to store time slices containing important data can lead to loss of important data and thus a poor quality of research. 2. The filtration of noise must be highly accurate. The event trigger algorithms must be able to eliminate majority of noise to prevent storage of unnecessary and potentially useless data. 3. The state of the art event trigger algorithms are able to process data in real time. The proposed alternative ideally should maintain or improve upon it’s predecessor’s performance else provide a good trade off with data quality.

1. INTRODUCTION

1.2 Research Question

this section needs more work, iterate as and when possible This report intends to improve upon the GPU pipeline proposed by (5), specifically this project wishes to answer the following research questions:

RQ1. Can the existing GPU pipeline be improved using neural networks (NNs)?

Improvement may be achieved by reducing the processing time of the pipeline or improving the accuracy of identifying important timeslices. This project focuses on achieving improvement via accuracy. The task of validating the runtime performance of the methods proposed in this paper is left to a separate project.

In order to answer **RQ1**, the following sub questions are formulated.

RQ2. Can the *Pattern Matrix Criterion* be replaced with a Multi Layer Perceptron?

(author?) proposed a novel trigger criterion which given a pair of points, can quantify the level of correlation amongst the points with an accuracy of 80%. The first phase of this project focuses on achieving better accuracy to identify “causally related” points using a Multi Layer Perceptron (henceforth referred to as MLP).

RQ3. Can the *Graph Community Detection* step be replaced with a Graph Convolutional Neural Network?

The output of the Pattern Matrix Criterion is fed into the Constant Pots clustering algorithm. The algorithm operates on the principles of Graph Community Detection and groups the input into communities of events hits and noise hits. The second phase of this project focuses on achieving a better accuracy for identifying event hits in a given timeslice using Graph Convolutional Neural Networks (henceforth referred to as GCN).

2

Data Preparation

At the time of undertaking this project, the KM3NeT Neutrino Telescope was still under construction, thus simulated data provided by Nikhef was used for the project. The data itself was split onto two parts namely *events* and *noise*, both of which came from different sources and in different formats.

2.1 Preparation of *events* dataset

The *events* dataset was provided as a *HDF5* (Hierarchical Data Format) with a size of 42MB consisting of the `/data/mc_hits` and `/data/mc_info` tables. For the purposes of this project, the two tables were combined such that each row in the `mc_hits` table contains it's corresponding 'event_id' from the `mc_info` table. A `label` column was added containing a value of '1' and the resulting table (henceforth referred to as the *events* dataset) was saved as a CSV file for future use.

2.2 Preparation of *noise* dataset

The *noise* data was generated using a Python library written and maintained by Nikhef, `k40gen`. `k40gen.Generators(21341, 1245, [7000., 700., 70., 0.])` was used to create an instance of a generator where the first two arguments are random seeds followed by a list of rates at which single, double, triple and quadruple hits should be generated. The generator instance is then passed into `k40gen.generate_40()` method which returns a (4, n) array containing (time, dom_id, pmt_id, tot). The position coordinates (ie. x, y, z coordinates) for each datapoint was provided in a *positions.detx* file which was parsed using the Numpy Python package (6) and added to the *noise* array. The Python library

2. DATA PREPARATION

Pandas (7) was used to convert the array into a (n, 4) dataframe. A `label` column was added containing a value of '0' and the dataframe was saved as a 3.9GB CSV file.

2.3 Preparation of *main* dataset

To create the *main* dataset for the project, the *events* and *noise* datasets were combined. Both datasets were read into memory as Pandas dataframes and their columns were renamed consistently. The two dataframes were concatenated and sorted based on the `time` column. Rows with a negative `time` value were dropped along with columns which are not relevant to this project. The `time` column was discretized into 15000ns bins and the resulting values were added to the `timeslice` column. The resulting dataframe was saved as a 1.9GB CSV file.

3

Data Exploration

The *main* dataset (generated as per the steps outlined in Chapter ??) was explored using statistical analysis and visualizations to observe any patterns and "local trends" that may be present. The following chapter presents the analysis that were done and the observations made.

Note that a random sample of only 10% of the data was taken for the following visualizations. This is because it is difficult to draw reasonable conclusions from the plots due to the high number of data points when the entire dataset is used.

3.1 Descriptive Statistics

Table ?? presents the descriptive statistics of the *main* dataset. The dataset consists of 7 columns and roughly 4.5 million rows, Table ?? provides more information on the columns on the dataset. The dataset does not contain any `nan` or `null` values except for the `event_id` column where rows containing noise hits are not associated with any event.

Next, the correlations amongst features is checked, the "Pearson" correlation is used. Figure ?? represents the correlation matrix of all features. No significant correlations are observed between `pos_x`, `pos_y`, `pos_z` and `time` which indicates that ML models may not be able to learn anything from the dataset without the aid of feature engineering.

The distribution of the `label` column is presented in Figure ?? . A severe class imbalance is noted between events and noise hits. To be precise, the dataset contains 489906 instances of events compared to over 4.5 million instances of noise. An effective strategy to handle the class imbalance will need to be devised during training of models to prevent the model from overfitting.

3. DATA EXPLORATION

Table 3.1: Description of columns

Column	Data type	Unit	Description
pos_x,	float	meters (m)	The position within the detector where the hit was detected, they represent the x,y,z coordinates of the hit respectively.
pos_y,			
pos_z			
time	float	nano seconds (ns)	The time at which the hit was detected.
label	int	NA	The type of hit, '0' represents noise and '1' represents a neutrino hit
event_id	int	NA	The id of the event to which the hit is related to. The id itself does not have any meaning, it is simply used to identify hits that originated from the same event.
timeslice	int	NA	The id of the timeslice to which the hit belongs. The id itself does not have any meaning, it is simply used to group hits into discrete bins.

3.2 Verification of Bias

The *events* dataset is synthetically generated using simulations. As such, it is likely that the event hits in each timeslice may occur at a specific time such as at the beginning, middle or end of the timeslice. Having such a pattern in the dataset may bias the model since it may learn this pattern and thus fail to generalize. If this pattern does exist in the dataset, corrective measures need to be taken such that the event hits in each timeslice are uniformly distributed.

To verify the existence of such patterns in the dataset, the mean `time` of event hits across all events was visualized. Figure ?? depicts a scatter plot of mean `time` across events. A uniform distribution is noted with no visible patterns. Thus no bias exists in the dataset and is deemed suitable for further analysis.

3.3 Exploration of Interesting Timeslices

Figure ?? represents to total number of event hits per timeslice. The dataset is discretized into 6759 timeslices of which 2783 timeslices contain only noise hits. This is corroborated by Figure ?? which presents a long tail distribution where many timeslices contain few to

3.3 Exploration of Interesting Timeslices

	pos_x	pos_y	pos_z	time	label	event_id	timeslice
pos_x	1.00	-0.01	-0.00	-0.00	0.00	-0.02	-0.00
pos_y	-0.01	1.00	0.00	0.00	-0.00	0.00	0.00
pos_z	-0.00	0.00	1.00	-0.00	0.00	-0.01	-0.00
time	-0.00	0.00	-0.00	1.00	-0.00	-0.01	1.00
label	0.00	-0.00	0.00	-0.00	1.00	nan	-0.00
event_id	-0.02	0.00	-0.01	-0.01	nan	1.00	-0.01
timeslice	-0.00	0.00	-0.00	1.00	-0.00	-0.01	1.00

Figure 3.1: Correlation matrix of features

Table 3.2: Descriptive statistics

	pos_x	pos_y	pos_z	time	label	event_id	time slice
count	4.58e+7	4.58e+7	4.58e+7	4.58e+7	4.58e+7	489906	4.58e+7
mean	1.16e-02	-1.59e-02	1.17e+02	5.00e+07	1.06e-02	2862.00	3.33e+03
std	5.12e+01	6.22e+01	4.86e+01	2.89e+07	1.02e-01	1667.61	1.92e+03
min	-9.46e+01	-1.15e+02	3.77e+01	0.00e+00	0.00e+00	0.00	0.00e+00
25%	-4.50e+01	-5.79e+01	7.40e+01	2.50e+07	0.00e+00	1392.25	1.66e+03
50%	1.30e+00	-4.18e+00	1.21e+02	5.00e+07	0.00e+00	2887.00	3.33000e+03
75%	4.04e+01	4.85e+01	1.60e+02	7.50e+07	0.00e+00	4304.75	5.00000e+03
max	9.62e+01	1.05e+02	1.96e+02	1.01e+08	1.00e+00	5734.00	6.77e+03

no event hits and few timeslices contain a high number of event hits.

Figure ?? depicts a scatter plot of *timeslice 615* which contains the largest number of event hits. It is observed that event hits occur close to each other in space and time (represented by the yellow, blue and green points) whilst background hits are uniformly distributed in space and time (represented by the purple points).

3. DATA EXPLORATION

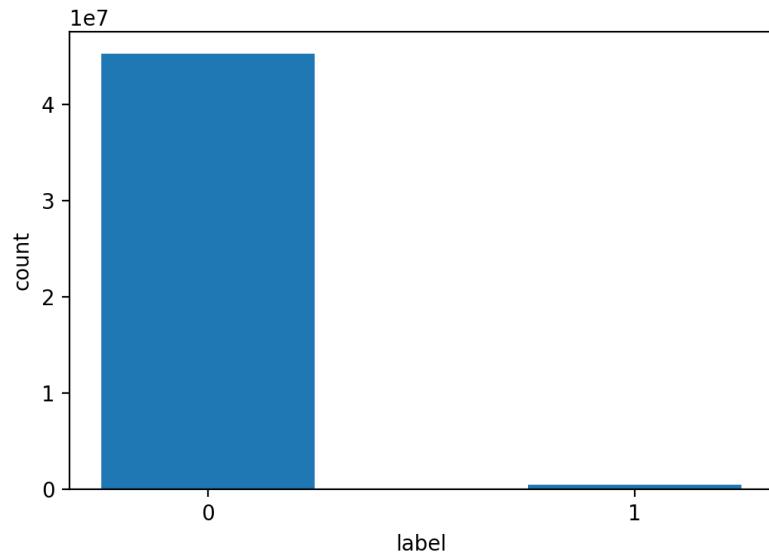


Figure 3.2: Distribution of `label` column

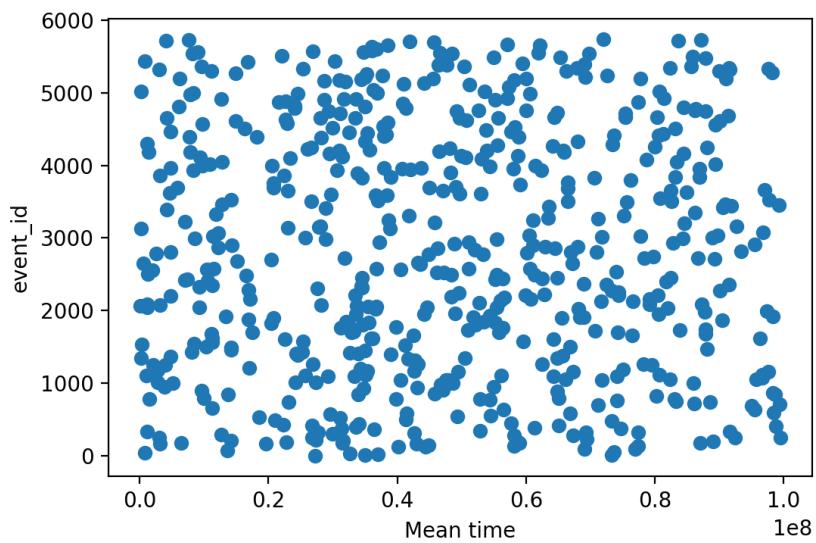


Figure 3.3: Verification of Bias

3.3 Exploration of Interesting Timeslices

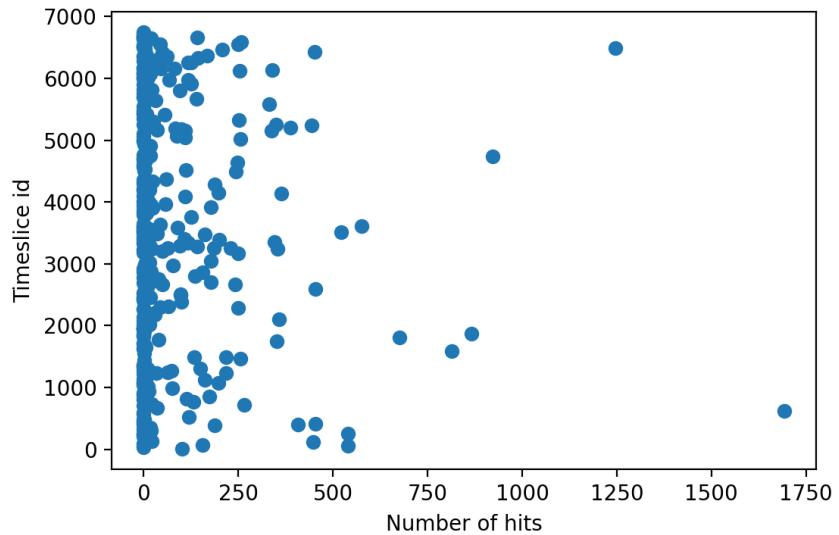


Figure 3.4: Distribution of event hits per timeslice

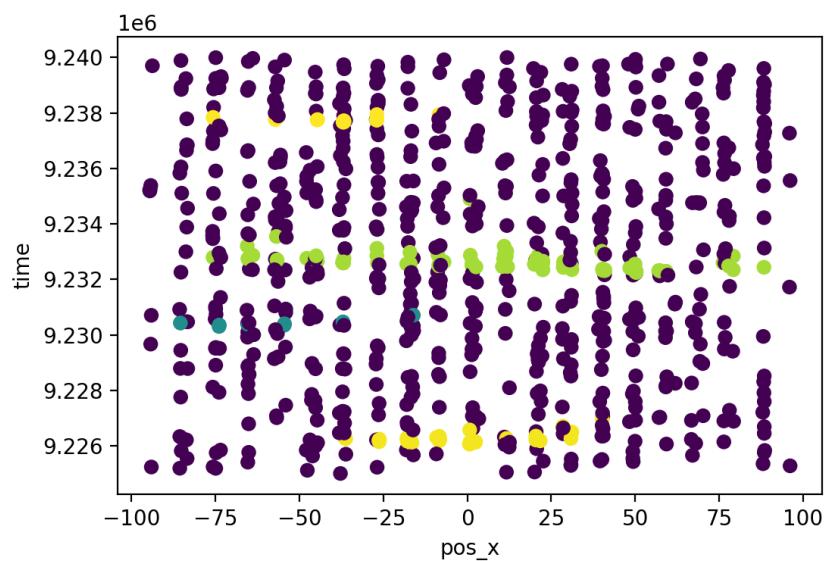


Figure 3.5: Distribution of Timeslice 615

3. DATA EXPLORATION

4

Hit Correlation

This chapter presents the replacement created using a Multi Layered Perceptron (MLP) for the *Hit Correlation Step* of the pipeline proposed by (5) (herein referred to as *The Karas Pipeline*).

The first step of The Karas pipeline is the Hit Correlation step. In this step, pairs of hits (from events and noise) are considered and the correlation along two axes namely space and time are considered. Karas proposes *The Pattern Matrix Criterion*, a novel event trigger algorithm by creating a correlation criterion based on the probability that a given space and time difference occurs between two event hits. From domain knowledge, the evaluation is limited to 100m and 300ns for space and time differences respectively. The algorithm is evaluated with a dataset containing 130 event hits and 5000 noise hits and scores in the range of 0.3 - 0.375 is reported for the recall, precision and F1 metrics.

4.1 Pattern Matrix Replacement

An alternative to the Pattern Matrix Criterion is created using a Multi-Layered Perceptron (MLP) which given a pair of points can classify them as causally related or unrelated. The training of such a neural network, requires two additional sub tasks summarized below and elaborated upon in Section ??.

1. **Preparation of dataset:** Given the main dataset, a new dataset (henceforth referred to as the *pattern matrix* dataset) is created such that each row contains the **x**, **y**, **z** and **time** features for all unique pairs of points.
2. **Creation of labels:** Using the *mc_info* table, labels are created for the dataset. Hits originating from the same event (ie. with the same **event_id**) are given a label

4. HIT CORRELATION

of 1 whilst all other pairs are given a label of 0.

4.2 Creation of the Pattern Matrix Dataset

A sample of the main dataset, ie. all data from the timeslice group 615 were chosen for the creation of the pattern matrix dataset since this timeslice contained the highest number of event hits compared to other timeslice. This is useful since the dataset is highly skewed and using a timeslice with lesser number of event hits will proliferate the skewed nature of the data in the pattern matrix dataset once created. The model is required to classify related and unrelated hits which can be done by observing the space and time difference between the given points. Since this phenomenon is consistent across the entire dataset, training using the timeslice 615 subsample does not introduce any bias into the model.

Figure ?? summarizes the pattern matrix dataset creation process. With an input data of shape $(n, 4)$ (n rows and 4 columns representing the `x`, `y`, `z`, and `time`), an output data of shape $(\sum_{k=n-1}^1 k, 9)$ is obtained. We note that there is a XYZ rise in the number of rows since each row (representing a single hit) is paired with the remaining unique rows. The output dataset consists of 9 columns due to the presence of `x`, `y`, `z` and `time` columns of two hits plus the label column.

4.3 Model Description

The expectation of the model is to identify if two given points are causally related to each other or not. As revealed through data exploration in Chapter ??, hits originating from neutrino events occur close to each other in space and time. Thus, The expectation from the model is to learn this phenomenon by training over pairs of points and classify unseen data as related or unrelated.

Since this is a binary classification task, the *Binary Cross Entropy Loss (BCELoss)* was selected as the loss function since it has been established as the standard loss function for binary classification tasks (?). As the BCELoss function expects an input in the range of $[0, 1]$, the *Sigmoid* activation function was chosen for the output layer. The *ReLU* activation was chosen for the hidden layers due to its XYZ properties as supported by many literate (?). The model architecture consists of an input layer, two hidden layers and an output layer. Figure ?? summarizes the model parameters and architecture.

4.4 Model Training

Several experiments were conducted in order to obtain the final model to replace the *pattern matrix* algorithm of The Karas pipeline.

Experiments were carried out in two alternative paths and their findings were ultimately combined to obtain the best model. The final results obtained from each category of experiments are summarized below. For further details, the corresponding section for the experiment categories follow.

1. **Experiments with dataset:** 50% random sample with equal number of samples for each class produced the best result.
2. **Experiment with optimizers:** *Adam optimizer* with a *learning rate* of 0.001 produced the best result.

4.4.1 Experiments with Dataset

In these experiments, variants of the data shape (ie. its length and breadth) were manipulated. Two breadth variants were considered, 1. the **Original Pattern Matrix** dataset of shape (n, 9) and 2. the **Delta pattern Matrix** dataset of shape (n, 5) where the difference in space and time of the points were taken.

Since the dataset is highly skewed, the majority class was *under sampled* for each breadth variant and four length variants were considered, 1. **10%** 2. **25%** 3. **50%** and 4. **75%** random sample of the dataset. Overall, diminishing rewards were observed as the length of the dataset was increased. The **50%** random sample of the **Delta pattern matrix** produced the best results.

4.4.2 Experiments with Optimizers

The *Stochastic Gradient Descent (SGD)* and the *Adam* optimizers were considered and several experiments with conducted by varying the *learning rate* parameter. The **50% delta pattern matrix** dataset was used for these produced the best results in the previous class of experiments.

4.5 Model Evaluation

The main dataset is highly skewed, with the **majority class** being hits from background noise and the **minority class** being hits from neutrino events. Thus, the *pattern matrix*

4. HIT CORRELATION

dataset is also skewed with the minority class being related hits and majority class being unrelated hits. In binary classification, the majority class is also referred to as the **negative class** (since it usually has a label of 0) and the minority class is referred to as the **positive class** (since usually has a label of 1). Henceforth this alternative naming convention is used in this report.

While the training dataset contains equal number of samples for each class, the testing data maintains its skewed distribution since this represents realistic data which the model will require to classify.

For highly skewed data, accuracy is not a good metric for evaluating the model performance (?) thus the following alternatives are considered:

1. **Recall** is the ability of the model to correctly identify the minority class. For this problem, the recall of the model is given higher presidency over its precision. This is because the model should be able to identify all instances of the positive class since this determines if the timeslice will ultimately be saved or not.
2. **Precision** is the ability of the model to not misclassify an instance of the negative class (ie. classify it as the positive class). Although this should also be high, it is often inversely proportional to recall.

Bibliography

- [1] SILVIA ADRIAN-MARTINEZ, M AGERON, F AHARONIAN, S AIELLO, A ALBERT, F AMELI, E ANASSONTZIS, M ANDRE, G ANDROULAKIS, M ANGHINOLFI, ET AL. **Letter of intent for KM3NeT 2.0.** *Journal of Physics G: Nuclear and Particle Physics*, **43**(8):084001, 2016. 1, 3
- [2] SEBASTIANO AIELLO, FABRIZIO AMELI, ANNARITA MARGIOTTA, MICHEL ANDRE, GIORGOS ANDROULAKIS, MARCO ANGHINOLFI, ANTONIO MARINELLI, GISELA ANTON, MIQUEL ARDID, CHRISTOS MARKOU, ET AL. **KM3NeT front-end and readout electronics system: hardware, firmware, and software.** *Journal of Astronomical Telescopes, Instruments, and Systems*, **5**(4):046001, 2019. 2, 3
- [3] ANNARITA MARGIOTTA, KM3NET COLLABORATION, ET AL. **The KM3NeT deep-sea neutrino telescope.** *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, **766**:83–87, 2014. 2
- [4] MAARTEN POST. *"KM3NNeT" A neural network for triggering and classifying raw KM3NeT data.* PhD thesis, Universiteit van Amsterdam, 2019. 2, 3
- [5] KONRAD KARAŚ. *Data processing pipeline for the KM3NeT neutrino telescope.* PhD thesis, Universiteit van Amsterdam, 2019. 3, 13
- [6] **NumPy - The fundamental package for scientific computing with Python.** 5
- [7] **Pandas.** 6