UNIVERSITY OF BARI

IceCube - Neutrinos in Deep Ice

Student name: *Roberto Barile*

Course: *Big Data* – Professor: *Michelangelo Ceci*
Date: *October 8, 2023*

Contents

1. Introduction

As we peer into the vast and infinite universe, mysteries abound, and among them lies the elusive neutrino particle. Invisible, nearly massless, and electrically neutral, neutrinos travel virtually undisturbed through matter, making them incredibly challenging to detect and study. Despite these challenges, they are fundamental to understanding a wide range of astrophysical phenomena, including the workings of exploding stars, gamma-ray bursts, and the dynamic interactions involving black holes and neutron stars. This report presents a detailed explanation of a data mining project aimed at predicting the direction of a neutrino particle using data from the IceCube Neutrino Observatory located at the South Pole.

2. Business understanding

Understanding the fundamental properties of neutrinos is a key objective in modern astrophysics. Due to their abundant yet elusive nature, neutrinos are key players in our understanding of the universe and the cataclysmic events that occur within it. The IceCube Neutrino Observatory, an unprecedented project in its scope and capabilities, seeks to capture and study these neutrinos, and in doing so, uncover the secrets they hold. However, a significant challenge exists: algorithms currently used to predict the direction of neutrino events are either quick but inaccurate, or precise but computationally costly.

2.1. Business Objectives

Our primary business objective is to optimize the prediction of neutrino events' direction using data gathered by the IceCube Neutrino Observatory. Improving this prediction will yield two main benefits:

- Enhancing scientific research: By improving the precision and speed of neutrino direction prediction, we can better understand neutrino behavior and their fundamental properties. This, in turn, will contribute to our broader understanding of astrophysical phenomena.
- Enabling real-time analysis: Faster, more accurate algorithms could dramatically increase the number of neutrino events analyzed, potentially enabling real-time analysis. This would revolutionize our approach to studying transient astrophysical events, making it possible for a global network of telescopes to respond swiftly to these phenomena.

2.2. Data Mining Goals

The data mining goal is a regression task, therefore a prediction task. Specifically we aim to predict the direction of a neutrino event. The direction can represented using two angle variables or with a coordinate vector [x, y, z]. Hence, the task is angular regression. The criteria that we adopt to evaluate the prediction is the angular error. It will be accepted an angular error inferior to 1.

3. Data Understanding

3.1. Data Collection

The data for this study was collected from the IceCube Neutrino Observatory. This state-of-the-art detector acquires data on neutrino events. Each neutrino event detected forms a collection of pulses. Each pulse is annotated with several attributes like time, sensor_id, charge, and auxiliary details. These pulses collectively form the events, which are then aggregated into batches.

3.2. Description of Data and Features

The following is a description of the entities and the features present in the dataset:

3.2.1. Sensor Data

- Sensor ID (Categorical - Nominal): The unique identifier assigned to each of the sensors in the IceCube detector.
- x, y, and z (Quantitative - Continuous): These are the coordinates of each sensor within the IceCube detector. The coordinates are given in a right-handed system with the origin at the center of the detector. These are expressed in meters with the z-axis pointing upwards from the South Pole. These coordinates can be used to compute the azimuth and zenith angles of a neutrino event.

3.2.2. Event Data

- Batch ID (Categorical - Nominal): This is the unique identifier of the batch in which the event is placed.
- Event ID (Categorical - Nominal): This is the unique identifier assigned to a neutrino event.
- First/Last Pulse Index (Quantitative - Discrete): These indices represent the first and last rows in the feature dataframe belonging to a given event.
- Azimuth and Zenith (Quantitative - Continuous): These angles are given in radians and signify the direction from which the neutrino came. The azimuth angle is between 0 and 2π, while the zenith angle is between 0 and π.

3.2.3. Pulse Data

- Event ID (Categorical - Nominal): This is the unique identifier for the neutrino event. Each event may comprise multiple pulses.
- Time (Quantitative - Continuous): This feature indicates the time of the pulse within the current event time window. This is given in nanoseconds. The absolute time of a pulse is irrelevant, the relative time with respect to other pulses within the same event is important.
- Sensor ID (Categorical - Nominal): This is the unique identifier for the sensor that recorded this pulse.
- Charge (Quantitative - Continuous): This is an estimate of the amount of light in the pulse, measured in units of photoelectrons (p.e.).
- Auxiliary (Categorical - Binary): This is a binary feature indicating the quality of the pulse. If True, the pulse was not fully digitized and is more likely to originate from noise. If False, the pulse contributed to the trigger decision and was fully digitized.

The dataset is voluminous, consisting of 131 million events divided into 660 batches, averaging about 200,000 events per batch. Each event is made up of a collection of pulses. On average there are 165 pulses per event. This comprehensive set of features makes this dataset an ideal source for developing a predictive model to ascertain the direction of neutrino events.

3.3. Verify data quality

The IceCube Neutrino Observatory data used in this study is complete, with no missing values across all the features associated with sensor data, event data, and pulse data. Every record is fully represented, which reinforces the integrity of the data and the reliability of any data-driven inferences or predictive models generated from this dataset.

The dataset shows uniformity in representation. All data points, regardless of whether they relate to events, sensors, or pulses, are consistently formatted across the dataset. This uniformity in representation improves the efficiency of data processing and analysis, as there are no inconsistencies to address.

In terms of accuracy, the IceCube detector employs advanced techniques to capture precise measurements. Photomultiplier sensors are used to measure light pulses, and the spatial positioning of these sensors within the detector is meticulously managed. Regular calibration and review of this sensor data ensure its accuracy.

However, some degree of noise and error is expected in real-world sensor data. For instance, the 'Charge' value in the pulse data, which estimates the amount of light in the pulse, can show variability around 1 photoelectron (p.e.). Additionally, pulses marked as 'auxiliary' are of lower quality than others.

The dataset was released in 2023, so it is quite recent. However, the IceCube observatory is continously operating and producing new data. Hence, it is always produced newer data to analyze. We run our analysis on this snaphost from 2023 and we plan to align with the drifts periodically as we will explain in the maintenance planning section.

3.4. Data exploration

Exploring the data, visualizing its various aspects, and understanding the underlying distributions and correlations is an important part of the data mining process. In this section, we will discuss some of the key findings from our exploration of the IceCube Neutrino Observatory dataset. For the pulse data we plot the charts for the data in the first batch.

3.4.1. First and Last Pulse Index

The distribution of the first and last pulse index values are mostly uniform, with values evenly spread across the range. However, the frequency diminishes for the highest values. We show the distribution in Figure 1

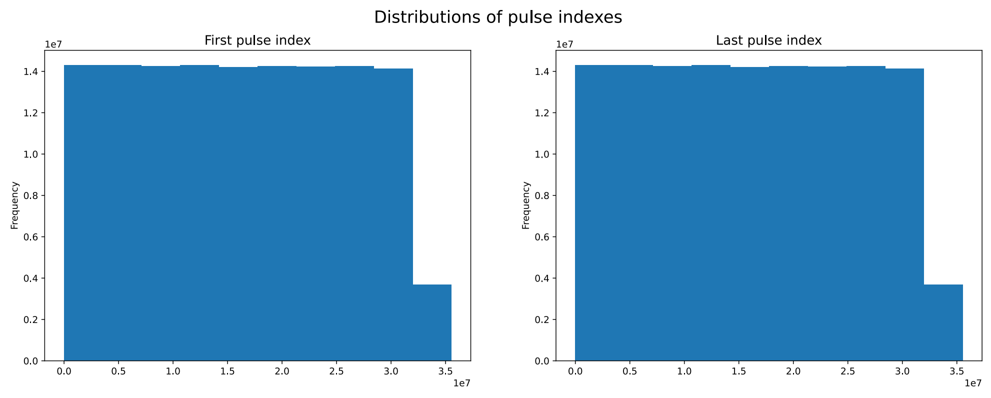

Figure 1: Distributions of First and Last Pulse Index

3.4.2. Azimuth and Zenith

The distribution of azimuth values is uniform across its range from approximately 0 to 6.283, with a mean of 3.144 and a standard deviation of 1.814. On the other hand, zenith values follow a normal distribution, with a mean of 1.534 and a standard deviation of 0.690. The distributions are depicted in Figure 2 It is interesting to note, in Figure 3 that there is no correlation between azimuth and zenith.

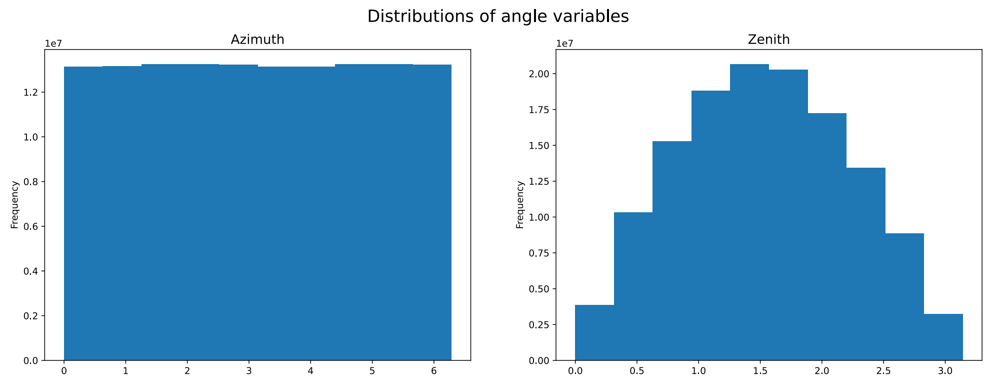

Figure 2: Azimuth and Zenith Distributions

3.4.3. Coordinates

The x, y, and z coordinates of the sensors in the detector all follow a normal distribution. They exhibit different mean values and standard deviations, which can be seen in the distribution plots in Figure 4.

3.4.4. Charge and Time

The distribution of the charge variable is left-skewed, with most of the values concentrated in the first bin. The mean charge is 3.909 p.e. with a standard deviation of 16.289 p.e.. On the other hand, the time variable also shows a left-skewed distribution, albeit less pronounced than the charge variable. The mean time is 13130.5 nanoseconds with a standard deviation of 4876.8 nanoseconds. The distributions are depicted in figure 5. Interestingly, there is a correlation between the charge and the time (or relative time) variables, wherein the charge appears to be the reciprocal of the time. The scatter plots are reported in Figure 6.

4. Data preparation

Given the impressive size of the dataset we adopt sampling tehcnique. To exploit the preliminary division of the data in batches we adopt block sampling. Furthermore, we think a block is big enough to cover the whole data distribution for each variable. Although is not computationally feasible to plot the charts on more than one batch we computed mean and std of the charge and time variables on 50 batches. They are both comparable to the measures on a single batch. Specifically, the mean of the charge is 4.12, the std of the charge is 17, the mean of the time is 12965 and the std of the time is 4486. Hence the main variables in the pulses data have the same parameters of the distribution. This justifies our assumption. We highlight that 50 batches do not fit in memory. Hence, we used the Dask Dataframe tool to compute such statistics. This tool allows to analyze very large dataframe storing them on disk. Therefore, we sample 10 batch out of 660. For the event data we use only the events that figures in the sampled batch. We use all the sensor data.

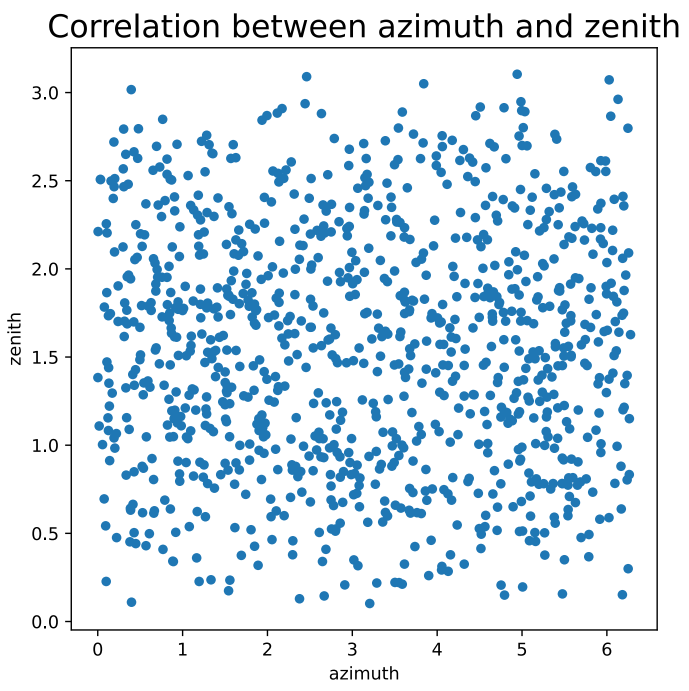

Figure 3: Correlation between Azimuth and Zenith

4.1. Integrate data

To integrate data we join pulse data and sensor data on the sensor ID variable in order to have the coordinates of the sensor activated by each pulse. The *unit of observation* could be the pulse, and, given that prediction will be on the events, they represent the *unit of analysis*. An event could be appear multiple times in the dataset as it is composed of several pulse. Hence, the entity treated is the event, a set of pulses.

4.2. Format data

Data is stored in multiple binary .parquet file. This format represents a compressed version of CSV files. Specifically, we have a .parquet file for the event data, a .csv file for the sensors data and a .parquet file for each event batch.

5. Modeling and Evaluation

In this phase, data mining models are chosen and an evaluation plan is designed. After that, models are built, trained and evaluated. According to the stated data mining goals the task is regression, specifically angular regression. The task is thus in *supervised learning* category. After joining the sensor data and the event data and the pulse data we know for each pulse the coordinates of the activated sensor. Furthermore, we stated that the pulses are the *units of observation* and the events are the *units of analysis*. Hence, the unit of analysis is a set of pulses. For each pulse in the set we now know the coordinates.

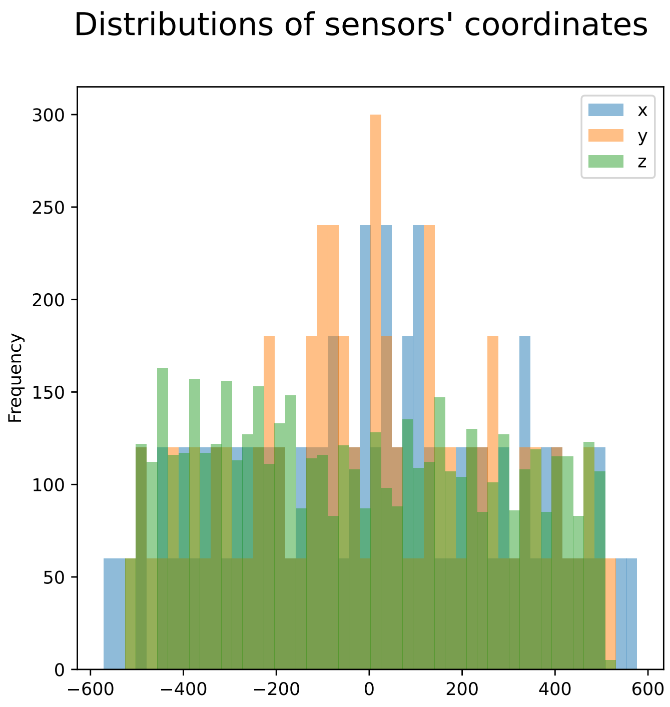

Figure 4: Distributions of sensor coordinates

Ultimately, it means that we can consider an event as a graph of pulses. We provide the visualization of an event as a graph in Figure. Each pulse, in addition to the coordinates, is described by a time variable. Hence, an event can also be interpreted as a sequence of pulse. Given the above considerations, we choose as models DynEdge, DynEdge + LSTM and DynEdge + biLSTM. In the following we delve into the details of each model.

DynEdge is a model based on a Graph Neural Network architecture. It models the spatial aspect of the data. We provide in Figure 8 the architecture of DynEdge. It accepts as input an event, represented as a tensor of dimension $n \times 6$ where n is the number of pulses in the event. DynEdge feeds this tensor to 4 edge conv layers. Next, DynEdge concatenates the outputs of each EdgeConv layer and feed the resulting vector to $MLP^{(1)}$. The output of $MLP^{(1)}$ is a tensor of dimension $n \times MLP^{(1)}_{output_size}$. Such tensor still depends on the

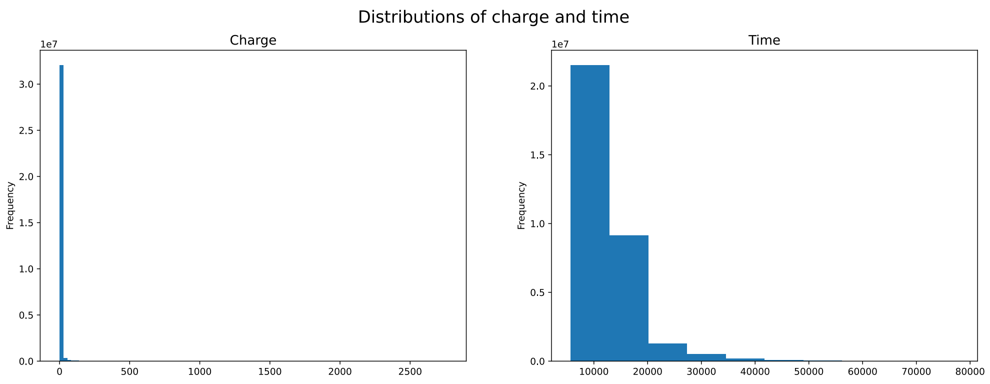

Figure 5: Charge and Time distributions

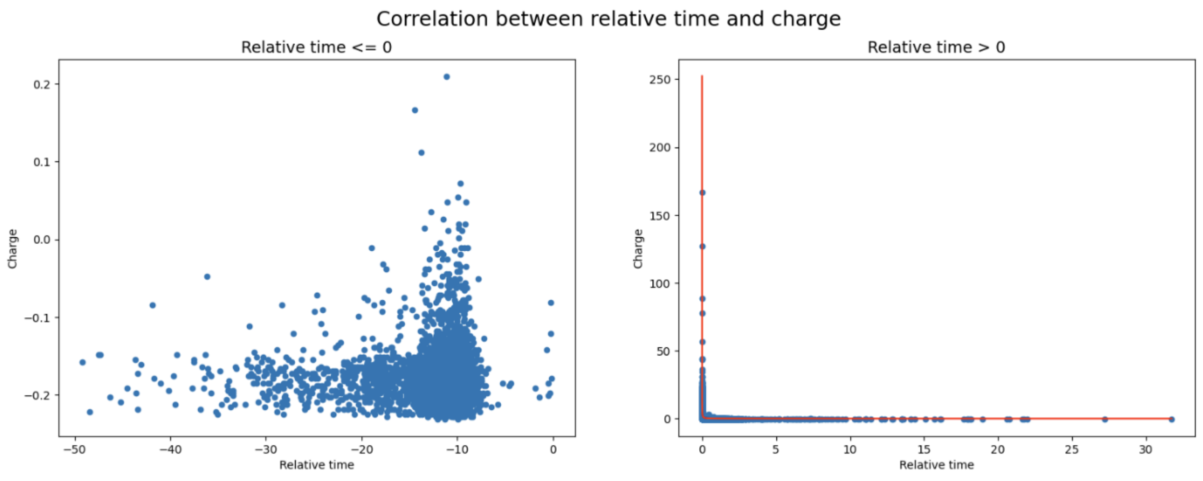

Figure 6: Charge and Time correlation

number of pulses in the event. The number of pulses is dynamic and can be different for each event. Hence, DynEdge adopts a pooling strategy to summarize the information of n vectors in a single vector. This enables the network to be dynamic and support events with a variable number of pulses. The result of the pooling can be extended with additional information containing global statistics on the raw features. This global statistic are the homophily ratio of the three coordinates and the time along with the number of pulses in the event. The result of the pooling, extended with the global statistics, passes through a second MLP, $MLP^{(2)}$. Finally, we have a prediction layers, the output is the prediction of the direction as a [x, y, z] vector.

The loss function adopted is the Von-Mises Fisher 3d Loss (vMF). The vMF distribution is a probability distribution defined on the hypersphere, which is a higher-dimensional generalization of the circle. It is often used to model directional data and is characterized by two parameters: the mean direction, which is a unit vector on the hypersphere, and the concentration parameter, which determines the spread of the distribution. The vMF loss measures the dissimilarity between two unit vectors on the hypersphere, which can be seen as points on the surface of the sphere. It is defined as the negative log-likelihood of the predicted vector given the ground-truth vector under the vMF distribution with a fixed concentration parameter. The vMF loss encourages the predicted vector to be close to the ground-truth vector in terms of their angular distance and penalizes large deviations from the mean direction. The vMF loss can be computed efficiently using standard vector operations and is differentiable, making it suitable for gradient-based optimization algorithms such as stochastic gradient descent.

We report the second model in Figure 9. In this model we enhance the previous one by injecting a temporal component. We model the temporal component using a Long Short Term Memory (LSTM) neural network. Specifically, we integrate the LSTM as a separate branch in the architecture. We feed the same input tensor both to the LSTM and to DynEdge. LSTM is recursively executed on each pulse. This means that we feed the first pulse to the LSTM. We then use the output on the first pulse and feed it as input for the same LSTM. The process goes on until all pulses are analyzed. Hence, also LSTMs are dynamic and supports events with a variable number of pulses. In each step, in addition to the output to feed again to the same network, the LSTM returns also an

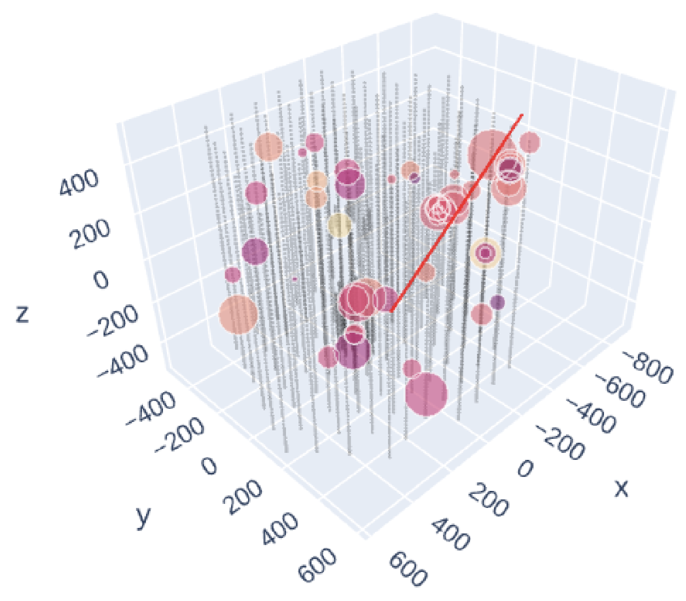

Figure 7: Event graph

intermediate output vector. LSTM outputs a vector for each element of the sequence, in this case for each pulse of the event. The intermediate outputs are useful for sequence to sequence tasks for instance. However, in this case we are interested in finding a features vectors for the whole sequence. Hence, we use the output after the final step. The LSTM is defined using a hidden size parameter and vectors of size equal to such hidden size. We concatenate the output of the LSTM and the output of $MLP^{(2)}$ in DynEdge. We add another MLP to process this concatenated vector and finally we have the prediction layer as before. We adopt the same loss function.

In the last proposal, namely DynEdge + BiLSTM, we inject the temporal component with a bidirectional variant of LSTM. The branch is integrated as in the previous method. The only difference is in the dimension of the LSTM output. This is double in size given that it models both directions of the sequence.

We need to choose the metrics to calculate the performances. In this case, having a regression of angle variables, the adopted metric is the angular error, formalized as follows:

$$\arccos\left(\sum_{i}^{N}(xyz_{\text{pred}} \cdot xyz_{\text{true}})\right)$$

For each model we use the 10 sampled batches as training data, we use batch 11 as validation set to tweak the number of epochs of the training. In particular, we adopt early stopping with patience 5 on the validation loss. The maximum number of epochs is 30.

In the following section we reported the number of epochs chosen for each model along with the charts of validation loss (Figure 10) and validation error (Figure 11) for each epoch. The baseline DynEdge is the first to stop after 23 epochs. Next, we have the

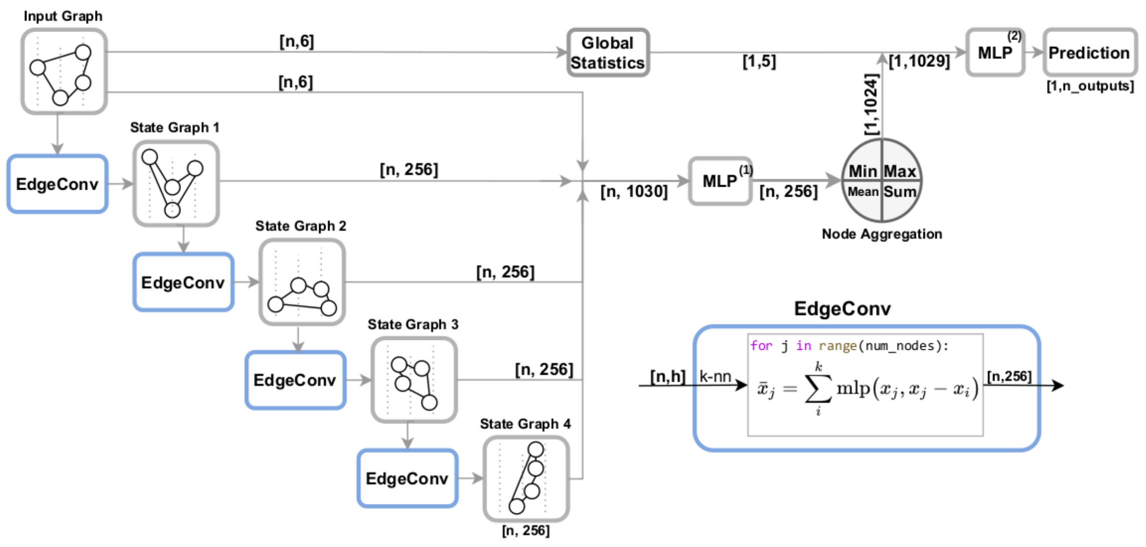

Figure 8: DynEdge architecture

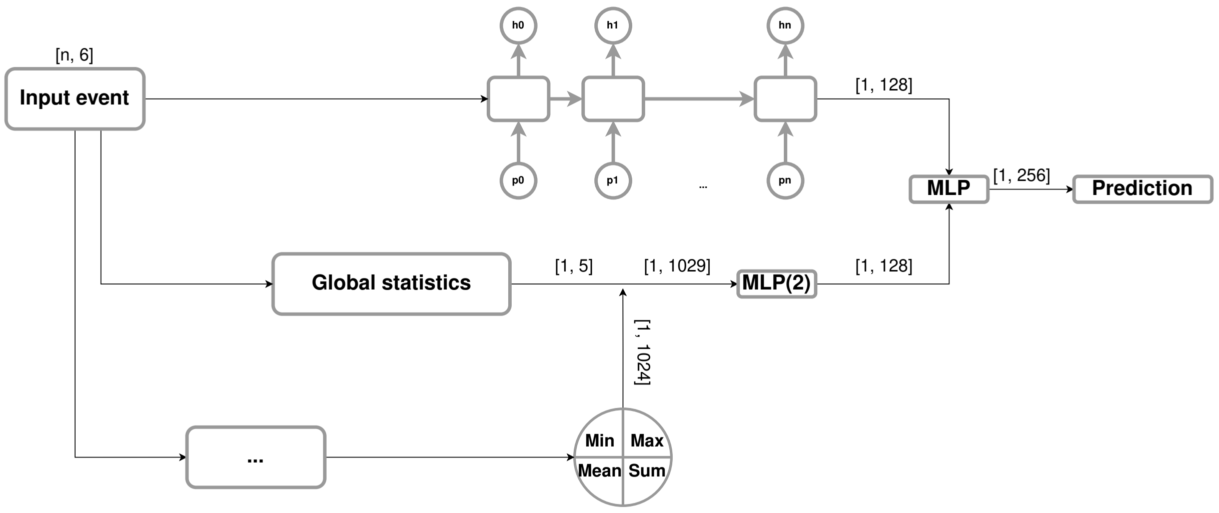

Figure 9: Integration of LSTM in DynEdge

version with the BiLSTM that stops after 24 epochs. Finally, The version with the LSTM is trained for all the 30 epochs. Although the version with BiLSTM is trained for less epochs it offers a greter reduction of the error.

After the epoch parameter has been selected from the validation and the final models have been trained on the whole training set, the models are tested on another unseen batch. Their results have to be compared. We can see them in Table 1.

Recalling the data mining goal, an angular error inferior to 1 is the acceptance threshold. All the models were not able to satisfy the data mining goal, the best performing one, decreased the angular error to 1.047. This is not enough with respect to our goal. However, we identified that integrate spatial and temporal aspects is a promising direction. We plan a further iteration using attention-based architectures to model the sequential aspect of the data.

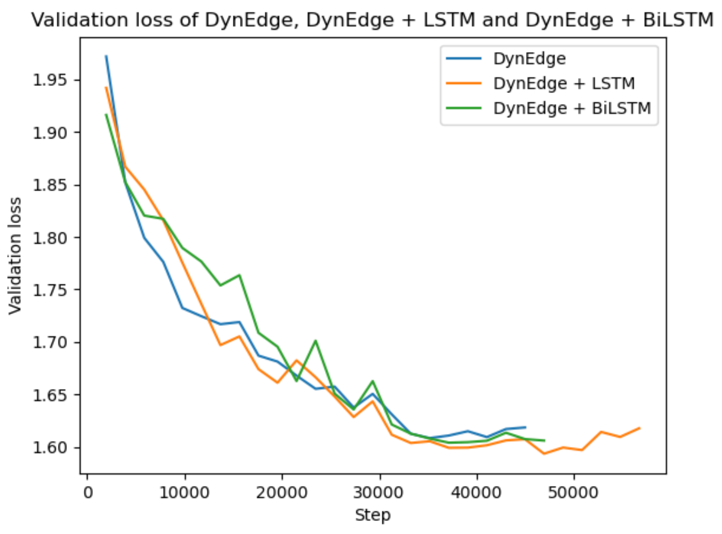

Figure 10: Validation loss

Model	Angular error
DynEdge	1.053
DynEdge + LSTM	1.048
DynEdge + biLSTM	1.047

Table 1: Angular error of the three compared models

6. Review process

This work brought a decrease in the error error to implement a real time prediction of events direction. The integration of the sequential aspect of data, thus obtaining a spatial-temporal architecture was effective. Hence, we may want to further explore this direction, investigating more on how to model the sequential aspect and how to integrate it. Furthermore is more powerful hardware is available we can adopt a larger sample. Thus, an angular error lower than 1 may be achievable exploring the proposed directions.

7. Deployment and Maintenance planning

For this project, the deployment should be done directly in the neutrino observatory, by integrating the inference of the direction after an event. The computation should be done server-side as it involves complex neural architecture. This integration can provide real-time insights to researchers working on neutrino events.

The observatory is always running and continuously producing new data. It's important to plan for regular updates as new research and discoveries emerge. This could involve retraining the model or fine-tuning it with new data. Furthermore, given the impressive volume data drift may verify. The performance of the model should be monitored regularly to ensure it's still providing accurate predictions.

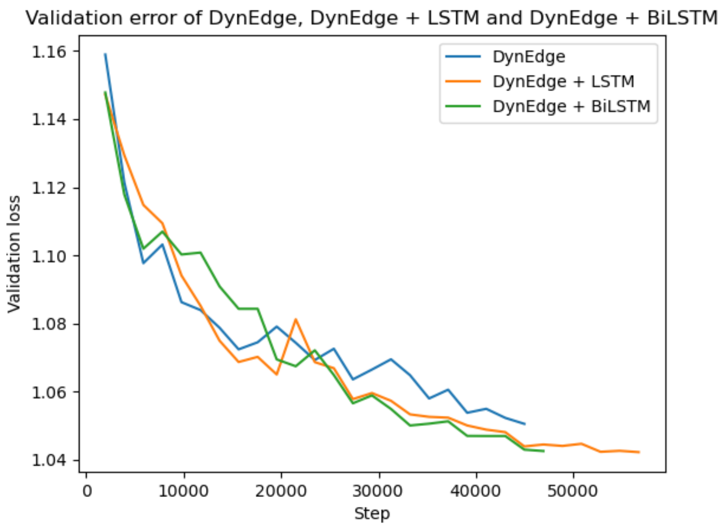

Figure 11: Validation error