**The IIE’s Varsity College**

**Exploring the Performance of Deep Learning Models for Neutrino Direction Prediction in High-Energy Astrophysics**

**Research Report**

**For**

**Post Graduate Diploma in Data Analytics**

**By**

**Christopher Myburgh**

**Supervisor: Dr. Tawanda Chiyangwa**

# **Abstract**

High-energy neutrino detection and event direction prediction are critical tasks in the field of astrophysics. This research presents a comprehensive exploration of the application of Recurrent Neural Networks (RNNs) for azimuth and zenith angle prediction of high-energy neutrino events recorded by the IceCube Observatory. The study begins by elucidating the data preprocessing pipeline, which extracts event-specific features and structures the data into structured arrays. Ethical considerations are addressed, ensuring data privacy and adherence to terms of use.

Three distinct RNN architectures - Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and a hybrid model - are examined for their efficacy in direction prediction. The LSTM model, despite its temporal pattern capturing prowess, exhibits extensive training times due to hardware constraints. In contrast, the GRU model offers competitive accuracy with remarkable computational efficiency, making it a valuable alternative. The hybrid model, while theoretically promising, demonstrates lower performance.

The comparison extends to azimuth and zenith angle predictions, revealing nuanced differences in correlation and prediction accuracy among the models. It is imperative to consider the research's limitations, such as restricted hardware capabilities, which influenced the number of training batches and model complexity.

This study aims to advance the understanding of RNN models for high-energy neutrino event direction prediction, with an emphasis on ethical data handling and model performance. Areas of future research may include exploring alternative model architectures, such as Transformers and ensemble methods, to further enhance predictive accuracy. The findings lay a solid foundation for continued investigation and innovation in the intriguing intersection of astrophysics and machine learning.

# **Definition of Terms**

**Beta decay:** A natural process that occurs within atomic nuclei, where a neutron can transform into a proton, emitting an electron or positron. This transformation helps atoms become more stable and is involved in phenomena like radioactive decay and nuclear reactions.

**Neutrinos**: Elusive particles with no electric charge that can travel long distances without being absorbed by matter. They carry information about their sources and can penetrate through the Earth.

**Neutrino Detectors**: Instruments designed to observe neutrinos and study their properties. They detect and analyse the signals produced when neutrinos interact with matter.

**Neutrino Direction Reconstruction**: The process of determining the direction from which neutrinos originate. It involves analysing the pattern of signals in the detector to identify the astrophysical sources of neutrinos and understand the physics behind their production.

**Deep Learning**: A subfield of machine learning that uses artificial neural networks to learn representations of data. Deep learning models can automatically extract features from complex data and learn patterns that may not be apparent to human experts.

**Convolutional Neural Networks (CNNs)**: Neural networks specifically designed for analysing grid-like data, such as images. They use convolutional layers to extract features and have been successfully applied in various fields like image and speech recognition.

**Graph Neural Networks (GNNs)**: Neural networks that operate on graph-structured data, where nodes represent entities and edges represent connections between them. GNNs can capture complex relationships and have been used for tasks like social network analysis and recommendation systems.

**Long Short-Term Memory (LSTM) Neural Networks**: Recurrent neural networks with specialized memory cells capable of capturing long-range dependencies in sequential data. LSTMs are particularly effective in modelling temporal dynamics and have been applied to tasks like natural language processing and time series analysis.

**Neutrino Event Reconstruction**: The process of reconstructing the properties and characteristics of a neutrino event based on the signals detected by a neutrino detector. It involves analysing the data to infer information about the neutrino, such as its direction, energy, and type.

**Particle Prediction**: The task of predicting the properties and attributes of particles based on observed data. In the context of neutrinos, particle prediction involves estimating the characteristics of the neutrino, such as its direction, energy, and flavour.

**Performance Evaluation**: The assessment of how well a model or system performs on a given task. It involves measuring metrics such as accuracy, precision, recall, and F1 score to evaluate the effectiveness of the model in predicting neutrino directions.

**Generalization**: The ability of a model to accurately predict unseen or new instances beyond the training data. A model with good generalization performs well on unseen examples, indicating its ability to capture underlying patterns and not merely memorize the training data.

**Physics Constraints**: The principles and laws of physics that govern the behaviour of particles and their interactions. Incorporating physics constraints in deep learning models ensures that the predictions align with known physical principles, improving the overall accuracy and reliability of the models.

**Interpretability**: The degree to which a model's predictions can be explained and understood. Models with high interpretability provide insights into the underlying reasoning and factors influencing their predictions, aiding in trust, transparency, and further analysis.

**Hyperparameters**: Parameters that define the configuration and behaviour of a machine learning model. They are set prior to training and include choices such as learning rate, number of layers, activation functions, and regularization techniques.

**Loss Function**: A measure that quantifies the difference between predicted outputs and the true labels. Loss functions guide the model's training by providing a signal for updating the model's parameters through gradient descent optimization.

**Learning Rate**: A hyperparameter that determines the step size in each iteration of the optimization algorithm during model training. It affects the speed and stability of the learning process, and an appropriate learning rate is crucial for achieving optimal performance.

**Feature Extraction**: The process of transforming raw input data into a representation that captures relevant patterns and characteristics. Deep learning models automatically learn feature representations from the data, enabling them to extract useful information for prediction tasks.

**Astrophysical Sources**: Natural sources in the universe that emit neutrinos, such as supernovae, active galactic nuclei, and cosmic rays interacting with interstellar matter.

**Temporal Dependencies**: Relationships and patterns that exist over time in sequential data. Capturing temporal dependencies is important in tasks involving time series or sequential data, such as analysing neutrino events, where the order and timing of signals carry valuable information.

**Sequential Inputs**: Data instances that have a specific order or sequence, where each element depends on the previous elements. Sequential inputs are commonly encountered in time series data, text data, and other domains where the order of information matters.

**Variable-Length Sequences**: Sequences of data that can have different lengths or numbers of elements. Handling variable-length sequences is crucial in tasks like neutrino event analysis, where events may have varying durations or numbers of signals.

# **Table of Figures**

[Figure 1 - Schematic of Sensor Geometry 18](file:///C:\Users\chris\Desktop\Research%20Report.docx#_Toc150100801)

[Figure 2- Preview displaying contents of batch\_1 18](file:///C:\Users\chris\Desktop\Research%20Report.docx#_Toc150100802)

[Figure 3- Example Event from the Dataset (Azimuth = 5.029 rad, Zenith= 2.087 rad) (kaggle.com, n.d.) 19](#_Toc150100803)

[Figure 4 - Schematic depicting Zenith and Azimuth Angles (Zhang et al., 2021) 21](file:///C:\Users\chris\Desktop\Research%20Report.docx#_Toc150100804)

[Figure 5 - Distribution of Azimuth values 22](#_Toc150100805)

[Figure 6 - Distribution of Zenith values 23](#_Toc150100806)

[Figure 7 - LSTM Model Architecture 37](#_Toc150100807)

[Figure 8 - LSTM Training Performance over 20 Epochs 38](#_Toc150100808)

[Figure 9 - LSTM Representation of Predicted Values vs Actual Values for Azimuth and Zenith Angles 38](#_Toc150100809)

[Figure 10 - GRU Model Architecture 39](#_Toc150100810)

[Figure 11 - GRU Training Performance over 20 Epochs 40](#_Toc150100811)

[Figure 12 - GRU Representation of Predicted Values vs Actual Values for Azimuth and Zenith Angles 40](#_Toc150100812)

[Figure 13 - Hybrid Model Architecture 41](#_Toc150100813)

[Figure 14 - Hybrid Model Training Performance 42](#_Toc150100814)

[Figure 15 - Hybrid Representation of Predicted Values vs Actual Values for Azimuth and Zenith Angles 42](#_Toc150100815)

**Contents**

[**Abstract** 2](#_Toc150100739)

[**Definition of Terms** 3](#_Toc150100740)

[**Table of Figures** 5](#_Toc150100741)

[**Chapter 1** 8](#_Toc150100742)

[**Background of the Study** 8](#_Toc150100743)

[**Problem Statement** 9](#_Toc150100744)

[**Purpose of the Study** 10](#_Toc150100745)

[**Justification** 10](#_Toc150100746)

[**Research Questions** 10](#_Toc150100747)

[**Research Objectives** 10](#_Toc150100748)

[**Hypothesis** 11](#_Toc150100749)

[**Scope of the study** 11](#_Toc150100750)

[**Study Design** 11](#_Toc150100751)

[**Brief of research methodology** 11](#_Toc150100752)

[**Chapter outline** 11](#_Toc150100753)

[**Chapter 2** 12](#_Toc150100754)

[**Theoretical Approach** 12](#_Toc150100755)

[**Literature Review** 12](#_Toc150100756)

[**Chapter 3** 17](#_Toc150100757)

[**Research Methodology** 17](#_Toc150100758)

[**Research Approach** 17](#_Toc150100759)

[**Data Sampling** 17](#_Toc150100760)

[**Data Collection** 17](#_Toc150100761)

[**Data Files and Labels** 18](#_Toc150100762)

[**Data Preprocessing** 19](#_Toc150100763)

[**Data Metrics** 22](#_Toc150100764)

[**Model Architecture** 24](#_Toc150100765)

[**Research Ethics** 27](#_Toc150100766)

[**Chapter 4** 28](#_Toc150100767)

[**Data Analysis and Results** 28](#_Toc150100768)

[**Measures of Validity** 30](#_Toc150100769)

[**Content Validity** 31](#_Toc150100770)

[**Construct Validity** 31](#_Toc150100771)

[**Measures of Reliability and Trustworthiness** 31](#_Toc150100772)

[**Test-retest** 31](#_Toc150100773)

[**Transparency** 31](#_Toc150100774)

[**Chapter 5** 32](#_Toc150100775)

[**Findings** 32](#_Toc150100776)

[**Comparison with Baseline Model** 32](#_Toc150100777)

[**Limitations** 32](#_Toc150100778)

[**Conclusion** 33](#_Toc150100779)

[**Future Research** 34](#_Toc150100780)

[**References** 35](#_Toc150100781)

[**Addendum A – LSTM Model** 37](#_Toc150100782)

[**Addendum B – GRU Model** 39](#_Toc150100783)

[**Addendum C – LSTM + GRU Model** 41](#_Toc150100784)

# **Chapter 1**

## **Background of the Study**

Neutrinos are tiny particles that scientists have been fascinated with since 1930 when Wolfgang Pauli first suggested their existence (Brown, 1978). They proposed to explain something strange observed in a process called beta decay, where electrons are involved. Neutrinos belong to a family of fundamental particles called leptons and have some unique characteristics. Characterized by their negligible masses, electric neutrality, and weak interactions with matter, neutrinos pose a unique challenge to experimental detection. Despite their elusive nature, neutrinos play crucial roles in various astrophysical phenomena. (A. de Gouy et al., 2013).

Neutrinos have an interesting property called "flavor," and there are three different types: electron, muon, and tau. Each type of neutrino is associated with a specific lepton—electron, muon, or tau particle. What's interesting is that neutrinos can spontaneously change or "oscillate" between these different flavors as they travel through space. This discovery was made in 1998 by the Super-Kamiokande experiment (Y. Fukuda et al., 1998). The observation of neutrino oscillations shattered the theoretical framework known as the Standard Model of particle physics. Prior to this discovery, the Standard Model presumed neutrinos to be completely massless, in stark contradiction to experimental evidence. Neutrino oscillations confirmed that these enigmatic particles possess non-zero masses, demanding an expansion of our understanding beyond the original model (Giacomelli, 2009).

Various methods have been developed to study them, and one prominent approach is employed by the IceCube Observatory. According to Aartsen et al (2017) a detailed description of the design, production, and calibration is provided for the IceCube digital optical module (DOM), as well as the cable systems, computing hardware, and methodology for drilling and deployment. The DOM serves as a vital component of the observatory, enabling the detection and analysis of neutrino and cosmic ray interactions. One of the key aspects highlighted is the remarkable operational efficiency achieved by the IceCube detector, with an impressive 98.4% of the deep ice DOMs currently operational and actively collecting data. This noteworthy accomplishment is attributed to a rigorous pre-deployment protocol, ensuring the reliability and functionality of the detector's components. Furthermore, by prioritizing software stability and employing vigilant monitoring practices, the IceCube observatory has maintained an exceptional uptime of 99%. These achievements solidify the detector's reliability and pave the way for continued scientific exploration until the culmination of the next decade (Aartsen et al., 2017).

The IceCube detector employs 5160 digital optical modules (DOMs) placed on 86 vertical strings at depths of 1450-2450 m beneath the ice's surface to capture the Cherenkov photons that result from charged particle movement through the ice. Each string contains 60 DOMs situated on a cable made of twisted copper-wire pairs, and the 78 strings of the primary in-ice array are spaced 17 m apart. The strings are organized in a hexagonal shape on a triangular grid with a horizontal spacing of 125 m, and the entire array spans a volume of one cubic kilometer of ice (Aartsen et al., 2017).

Neutrinos, known for their weak interactions with matter, can travel vast distances without being absorbed or deflected. Occasionally, when a high-energy neutrino interacts with the ice, it generates secondary particles, such as charged particles, through interactions with atomic nuclei. These secondary particles, including charged leptons, move faster than light through a medium and emit a faint glow called Cherenkov radiation. Cherenkov radiation is the result of a charged particle surpassing the phase velocity of light in a medium, as first observed by Pavel Alekseyevich Cherenkov. In a vacuum, light travels at a constant speed denoted as "c"; however, in a medium, its velocity decreases. When a high-velocity electron passes through a dielectric medium, it induces polarization in the medium's electromagnetic field, creating a coherent shockwave and emitting Cherenkov radiation, typically in the ultraviolet wavelength range with a continuous spectrum (Watson, 2011).

In the realm of neutrino research, despite notable advancements over time, a fundamental challenge persists: the precise detection of these enigmatic particles. Neutrinos, characterized by their exceptionally weak interactions with matter, present a formidable obstacle to accurate measurement and observation. While detection techniques, exemplified by instruments like the IceCube Observatory, have made significant progress in capturing neutrino interactions, they have yet to achieve the requisite accuracy needed for a comprehensive understanding of neutrino properties and behaviours.

A central issue contributing to the accuracy challenge is the inherently low likelihood of neutrino interactions with matter. The vast majority of neutrinos passing through detection instruments evade capture due to their feeble interactions, severely limiting the collection of comprehensive data. This limitation hampers our ability to precisely determine neutrino characteristics, including their masses, oscillation patterns, and potential deviations from the Standard Model.

Furthermore, background noise in neutrino detectors adds complexity to the problem. This noise can generate false signals and hinder the differentiation of genuine neutrino events from other particle or environmental effects, thus undermining the reliability and precision of collected data.

## **Problem Statement**

The detection of neutrino particles is crucial for advancing our understanding of astrophysical phenomena such as supernovae and gamma-ray bursts. However, accurately predicting the direction of a neutrino particle is a challenging task due to the large volume of data produced by the detection process and the complexity of the underlying physics. The current methods for detecting neutrinos rely on traditional statistical techniques that are limited in their accuracy and speed. Therefore, there is a need to develop a more efficient and accurate machine learning model that can predict the direction of a neutrino particle with high precision and low computational cost. This research problem requires the development and optimization of a machine learning algorithm that can process large volumes of data and accurately predict the direction of a neutrino particle, which can lead to a better understanding of the astrophysical phenomena that produce these particles.

Convolutional neural networks (CNNs) have shown promise in reconstructing neutrino events and predicting particle properties, but they have some limitations that need to be considered. CNNs primarily focus on local patterns within the input data, such as spatial correlations in images. They are data-driven models that learn patterns solely from the provided input data. Additionally, CNNs are typically trained on specific event classes or datasets, making them susceptible to limited generalization to new or diverse event types. Additionally, neutrino detectors can be subject to noise and systematic effects. CNNs are sensitive to noise and variations in the input data, leading to biased or inaccurate predictions. To overcome these limitations, further research is needed to explore more sophisticated models or hybrid approaches that integrate domain knowledge and incorporate global context. Additionally, addressing uncertainties and noise in the data through pre-processing techniques or additional network components could enhance the accuracy and reliability of the predictions (Glaser et al., 2022).

Another Deep Learning method is Graph Neural Networks (GNNs). GNNs have shown promise for reconstructing neutrino events and predicting particle properties, but they also have limitations that must be considered. Firstly, GNNs can face scalability challenges when applied to large-scale graphs representing complex neutrino events. The computational complexity increases with graph size and connectivity, making it difficult to process data from high-density detectors or events with numerous interacting particles. Secondly, constructing an appropriate graph representation of the neutrino event is crucial for GNNs. Determining the optimal node and edge features, as well as the graph structure itself, can be nontrivial. The choice of graph representation can impact the performance and accuracy of the GNN in reconstructing the event or predicting particle properties (Søgaard et al., 2023).

Long Short-Term Memory (LSTM) neural networks offer significant advantages when dealing with sequential data, making them a suitable choice for analysing neutrino event data. One key argument for using LSTMs in this context is their ability to capture long-range dependencies and temporal patterns within the data. Neutrino events often exhibit intricate temporal dynamics, with signals evolving over time. LSTMs can effectively model these temporal dependencies, enabling accurate reconstruction and prediction of neutrino events. Their specialized memory cells allow for the retention of information over extended sequences, enabling accurate reconstruction and prediction of neutrino events. LSTMs' gate mechanisms regulate the flow of information, enabling selective updating or forgetting of relevant information. This adaptability makes LSTMs suitable for handling irregular or variable-length sequences, which is crucial in the context of neutrino events that may have varying durations or numbers of signals (A. Iess et al., 2022). LSTMs have been successfully applied in various domains involving sequential data, such as natural language processing and speech recognition. Their track record of achieving state-of-the-art performance in these areas demonstrates their effectiveness in modelling and extracting meaningful representations from sequential inputs.

This research project aims to explore the effectiveness of Long Short-Term Memory (LSTM) neural networks in predicting the direction of neutrino particles. By leveraging the strengths of LSTMs in handling sequential data and temporal dependencies, we endeavour to develop a machine learning model that can accurately predict the direction of neutrino particles. The outcome of this research can significantly enhance our ability to analyse neutrino events, ultimately contributing to a better understanding of astrophysical phenomena responsible for the creation of these elusive particles.

## **Purpose of the Study**

This study aims to investigate the feasibility of employing Long Short-Term Memory (LSTM) Neural Networks for accurately predicting the direction of neutrino particles. It seeks to assess the comparative performance of LSTM Neural Networks against Convolutional Neural Networks (CNN) and Graph Neural Networks (GNN) in this predictive task.

## **Justification**

Machine learning algorithms, particularly deep learning methods, have emerged as indispensable tools in the field of astronomy. The extensive use of these algorithms is well-documented in the literature, with numerous studies and research papers highlighting their value. As evidenced by Baron (2019), deep learning algorithms excel in handling the massive and intricate datasets generated by astronomy, enabling efficient analysis and interpretation of the data. They possess the ability to extract meaningful patterns and structures from vast datasets, leading to enhanced data-driven insights. Astronomical data encompasses a wide range of sources, including telescopes, satellites, and surveys, which produce copious amounts of information. Deep learning algorithms have proven to be highly effective in automatically learning and adapting to complex features within this rich and diverse data, enabling astronomers to solve intricate tasks such as classification, object recognition, and data analysis.

The robustness and scalability of deep learning algorithms further contribute to their usefulness in astronomy. Once trained on large datasets, these models exhibit a remarkable ability to generalize their knowledge to new and unseen data, enabling the exploration of uncharted territories in the universe. This generalization capability also enhances the reliability of deep learning models, making them robust to noise and uncertainties present (Baron, 2019).

## **Research Questions**

Primary Research Question:

* Can Long Short-Term Memory (LSTM) Neural Networks be used to accurately predict the direction of a neutrino particle?

Secondary Research Questions:

* How does the performance of LSTM Neural Networks compare to that of Graph Neural Networks (GNN) for this task?
* What are the key factors that affect the performance of deep learning models for neutrino direction prediction?

## **Research Objectives**

Primary Research Objective:

* To apply deep learning methods for accurately predicting the direction of neutrino particles using LSTM Neural Networks.

Secondary Research Objectives:

* To evaluate the performance of various deep learning architectures and compare the results with Graph Neural Networks.
* To Investigate key factors influencing the performance of deep learning models for neutrino direction prediction using LSTM Neural Networks.

## **Hypothesis**

H1: It is hypothesized that LSTM neural networks can effectively predict the direction of neutrino particles due to their ability to capture temporal dependencies and handle sequential data.

H2: The hypothesis suggests that LSTM models outperform CNN and GNN models in this specific task, as CNNs are primarily suited for spatial data and GNNs may face limitations in scalability and generalization.

H3: The research further hypothesizes that key factors influencing the performance of deep learning models for neutrino direction prediction include the quality and representation of the input data, the architecture and hyperparameters of the models, and the availability of a sufficient amount of labelled training data.

## **Scope of the study**

The scope of this study encompasses the application of Recurrent Neural Networks (RNNs) for azimuth and zenith angle prediction of high-energy neutrino events detected by the IceCube Observatory. The research delves into data preprocessing, ethical considerations, and the comparative analysis of three RNN architectures (LSTM, GRU, and LSTM + GRU) in terms of model performance, training efficiency, and direction prediction accuracy. Additionally, the research evaluates the correlation and prediction accuracy of these models. This study provides insights into the feasibility and limitations of RNN models for this specific task and highlights the need for ethical data handling. The scope also opens avenues for future research in alternative model architectures, including Transformers and ensemble methods, to further enhance predictive accuracy in the field of astrophysics.

## **Study Design**

This study adopts a retrospective cross-sectional design, employing the method of secondary data analysis. It aims to investigate relationships and patterns within existing datasets, without direct involvement in the data collection process. By utilizing pre-existing information, this approach offers a cost-effective means of addressing research inquiries and testing hypotheses. The study will undertake a systematic examination of historical data, aiming to discern meaningful associations and trends that contribute to the existing knowledge in the relevant field of study. Through rigorous analysis and interpretation of the collected data, the study endeavours to shed light on significant relationships and provide valuable insights to the scholarly community.

## **Brief of research methodology**

The research methodology involves the use of Recurrent Neural Networks (RNNs) to predict azimuth and zenith angles for high-energy neutrino events recorded by the IceCube Observatory. It includes data preprocessing, which entails data cleaning and structuring, and the creation of structured NumPy arrays. Ethical considerations are a critical aspect, ensuring compliance with data usage guidelines. The study focuses on three RNN architectures: LSTM, GRU, and LSTM + GRU, comparing their performance, training efficiency, and direction prediction accuracy. The analysis encompasses measures of validity, reliability, and transparency.

## **Chapter outline**

Chapter 2 focuses on data collection and preprocessing procedures, while Chapter 3 elaborates on the architecture of the implemented models. In Chapter 4, a thorough analysis of the obtained results is presented. Chapter 5 examines the implications of our findings in the context of current models and discusses the potential impact of future research in this field.

# **Chapter 2**

## **Theoretical Approach**

The study was based on the principles and concepts of Statistical Learning Theory, a theoretical framework that combines statistical analysis and machine learning to understand the process of learning from data. This framework provided a solid foundation for analysing and developing learning algorithms that could make accurate predictions and generalizations based on observed data. Statistical Learning Theory focused on the crucial aspect of generalization, which referred to the ability of a learning algorithm to perform well on unseen data beyond the training set. The ultimate goal was to build models that captured the underlying patterns and relationships in the data, enabling accurate predictions on new, unseen instances (Christiansen, 2018).

Statistical Learning Theory encompassed several key principles and concepts that guided the process of learning from data. One fundamental principle was empirical risk minimization, which involved minimizing the average error or loss on the training data. This was achieved by finding the optimal model parameters that reduced the discrepancy between predicted and true outputs in the training set. Additionally, the framework recognized the intricate trade-off between model complexity and the available training data. On one end of the spectrum, overly simplistic models with low complexity might have struggled to capture the underlying structure of the data, leading to a phenomenon known as underfitting. Conversely, excessively intricate models with high complexity could have overfit the training data by capturing idiosyncratic noise and details, thus impeding their ability to generalize well to unseen instances (Frost, Armstrong, and Christiansen, 2019).

## **Literature Review**

The study conducted by Glaser, et al. (2023) offers a novel approach to neutrino direction and energy estimation using deep neural networks trained on simulated raw data. This technique presents significant potential for enhancing the accuracy of radio neutrino event reconstruction.

Glaser et al. (2019) introduced the NuRadioReco framework, which focuses on improving the accuracy of radio neutrino event reconstruction. The framework incorporates a variety of techniques and algorithms, and one standout approach is the forward folding method.

The forward folding technique, as applied by Glaser et al. (2019), has proven to be effective in determining neutrino direction, particularly in scenarios with low signal-to-noise ratios. It operates on the basis of reconstructing the neutrino interaction vertex as input, which has primarily been employed for hadronic particle cascades. However, preliminary results indicate its potential applicability to electron neutrino-charged-current interactions, thereby broadening its utility.

In 2022, Glaser et al. extended their work by introducing a deep-learning-based method for neutrino direction and energy estimation in in-ice radio detectors. Their approach incorporates a Convolutional Neural Network (CNN) architecture inspired by the VGG model.

The CNN architecture utilized in the study consists of multiple convolutional and pooling layers, followed by fully connected layers for classification. Training the model involves using Tensorflow and Keras and optimizing it with the mean absolute error (MAE) loss function with an Adam optimizer, employing a low learning rate (e.g., 5 · 10^-5) to ensure precise fine-tuning.

Assessing the performance of the CNN reconstruction involves two critical metrics: the core resolution, which evaluates the accuracy of determining the position of the shower core along the signal trajectory, and the angular resolution, which measures the precision in reconstructing the shower direction. In the study by Glaser et al. (2022), it is reported that νμ-CC interactions achieve an 8.0 m core resolution, while νe-CC interactions achieve a 1.6 m core resolution. Furthermore, the angular resolution is stated to be 12.9° for νμ-CC interactions and 5.2° for νe-CC interactions. These metrics demonstrate the impressive performance of the CNN, with low core and angular resolutions, indicating a high level of accuracy in both determining the core position and reconstructing the shower direction.

However, it's noteworthy that while the CNN reconstruction shows promise, Glaser et al. (2022) acknowledge that improvements are still possible, and it may not yet surpass alternative techniques in certain aspects of the reconstruction task. Notably, the CNN approach may have limitations when dealing with noise and variations in the input data, which can lead to biased or inaccurate predictions.

In recent years, there has been a growing interest in the development and application of Graph Neural Networks (GNNs) for various scientific and computational tasks. One notable area where GNNs have shown significant promise is in the domain of neutrino and muon event classification, particularly in the context of neutrino telescopes like IceCube.

The research presented by Minh (2021) revolves around the application of Graph Neural Networks (GNN) for the reconstruction and classification of neutrino events in the context of the IceCube experiment. The study aims to improve the resolution of essential parameters such as the zenith angle, energy, and event topology, while also addressing the computational efficiency of the reconstruction process.

The study employs a combination of data from the IceCube experiment and the application of Graph Neural Networks (GNNs) to enhance the reconstruction of neutrino events (Minh, 2021). The utilization of GNNs is a novel and promising approach that allows for the encoding of detector pulses without the need for extensive preprocessing. By leveraging this deep learning technique, the study successfully demonstrates improvements in event reconstruction resolution and computational efficiency.

The use of a Receiver Operating Characteristic (ROC) curve to compare the performance of the GNN with the baseline algorithm (Boosted Decision Tree) is an appropriate method for assessing the effectiveness of the GNN (Minh, 2021).

The research conducted by Minh (2021) demonstrates notable advancements in both resolution and computational efficiency in the context of neutrino event reconstruction using Graph Neural Networks (GNNs). However, it is essential to consider several limitations and gaps in this study. Firstly, it is worth noting that the results pertaining to the IceCube Upgrade are based on an early stage of event selection, suggesting that the reported improvements in resolution may not be conclusive or wholly representative of the anticipated performance for the upgrade. Hence, further validation with a more extensive dataset is required to ensure the reliability of these findings.

Abbasi et al. (2022) introduced a novel method for classifying events as either neutrino or muon-induced, leveraging the power of GNNs to capture local spatial relationships within a graph representation. The framework used for this research, known as GraphNeT, is an open-source Python toolkit designed to facilitate high-quality, user-friendly, end-to-end functionality for performing reconstruction tasks at neutrino telescopes through the utilization of GNNs. GraphNeT's flexibility and accessibility make it an invaluable tool for researchers interested in experimenting with GNNs in the context of neutrino event classification.

A key component of this study was the development of a specific GNN algorithm named Dynedge. Dynedge relies on the Edge Convolution (EdgeConv) operator, originally tailored for computer vision segmentation analysis, to extract critical features from event pulses represented as a graph. The EdgeConv operator operates on individual nodes within a graph, considering their local neighbourhoods. This design makes EdgeConv particularly well-suited for capturing and encoding local spatial relationships crucial for differentiating between neutrino and muon events.

One of the notable contributions of this research is the provision of an example notebook by the authors, which serves as a practical resource for fellow researchers and enthusiasts in the field. This notebook encapsulates the entire process, from installing the GraphNeT framework to training and applying a GNN model to the competition data. Notably, this pre-trained model was trained on event data from batches 1 to 50 and had achieved a mean angular error score of 1.018215.

The provided notebook offers an implementation for training and utilizing a Graph Neural Network (GNN) known as DynEdge. This GNN is applied to the directional reconstruction of neutrino events within the context of the IceCube Neutrino experiment. PyTorch and the DynEdge library are employed to facilitate the GNN training process.

The methodology can be broken down into several key stages, including data preparation, feature engineering, directional target definition and training. These stages collectively form the foundation for the code's application in the directional reconstruction of neutrino events within the IceCube Neutrino experiment. The essential dataset for this process comprises three main components: 'Sensor\_geometry' for sensor coordinates, 'train' for event-specific sensor readings, and 'train\_meta' for event metadata, including azimuth and zenith angles.

The training stage of the methodology involves the construction of the GNN model using the DynEdge framework. This framework includes configuration settings for the IceCube detector and GNN. The training process itself is executed using the PyTorch library, with the Adam optimizer and a piecewise linear learning rate scheduler, allowing the GNN to learn and refine its predictive capabilities.

While dynedge displayed superior performance in low-energy ranges, its effectiveness diminished outside the relevant energy range for neutrino oscillation studies. This suggests that further optimization is required for high-energy events. Additionally, the robustness analysis against systematic uncertainties indicated that dynedge exhibits higher variations compared to the existing reconstruction method, retro, under various sources of uncertainty, particularly for track and cascade reconstruction tasks.

Time series forecasting is indispensable in a multitude of real-world applications, where the ability to predict future values based on historical observations holds immense practical significance. In the realm of finance, time series forecasting assumes a pivotal role by aiding in the prediction of stock prices, currency exchange rates, and commodity prices. The accurate predictions in financial markets empower investors and institutions to refine their strategies, mitigate risks, and make informed decisions. Akin to the financial domain, climate science significantly benefits from forecasting models that predict weather patterns, extreme events, and long-term climate changes. These models are instrumental in disaster preparedness, resource allocation, and climate change mitigation efforts. Moreover, in the field of astrophysics, time series forecasting proves to be of paramount importance. The forecasting of astronomical events, including gamma-ray bursts and transient phenomena, is indispensable for observatories seeking to capture and analyse rare celestial occurrences, shedding light on the mysteries of the universe (Chakraborty, 2023).

Chakraborty, (2023) discusses the rise of Long Short-Term Memory (LSTM) networks marks a significant turning point in the domain of time series prediction. LSTM networks, as a subclass of Recurrent Neural Networks (RNNs), have garnered considerable attention due to their unique ability to capture long-range dependencies and their inherent capacity to process sequential data efficiently. These networks are remarkably adept at modelling complex and non-linear relationships embedded within time series data, rendering them superior to traditional statistical methods in numerous aspects. Notable strengths of LSTM networks include their exceptional prowess in handling long-term dependency modelling. LSTMs can efficiently capture intricate patterns and trends that may elude conventional models, owing to their internal gating mechanisms. These mechanisms, such as the forget gate and input gate, allow LSTMs to regulate the flow of information over extensive sequences, an invaluable feature for learning from historical observations. Moreover, LSTM networks are inherently non-linear, capable of approximating complex, non-linear relationships inherent in time series data. This quality is particularly advantageous in the domain of finance, where price dynamics are governed by a myriad of factors, each exhibiting varying degrees of non-linearity.

One of the remarkable features of LSTM networks is their ability to handle time series data containing missing values, a common scenario in real-world applications. Researchers have developed techniques to seamlessly integrate missing data imputation within the LSTM framework (Chakraborty, 2023). This approach ensures robustness to incomplete information, enabling LSTM networks to continue making accurate predictions even when confronted with missing or incomplete data. Additionally, LSTM networks possess the unique capability of autonomous feature learning. These networks can automatically discern and extract relevant features from raw time series data, eliminating the need for labour-intensive manual feature engineering. This feature extraction capability is of particular significance when dealing with large, diverse datasets.

The applications of LSTM networks in various domains underscore their versatility and wide-ranging utility in time series forecasting. In the domain of finance, researchers have harnessed the power of LSTM networks to predict stock prices, volatility, and currency exchange rates with remarkable precision. Climate science has witnessed the deployment of LSTM networks for short-term and long-term weather forecasting, including predictions related to temperature and precipitation. Meanwhile, in the field of astrophysics, LSTM networks have assumed a critical role in forecasting astronomical transients and predicting the behaviour of celestial objects, such as blazars, which emit high-energy gamma rays. This diverse array of applications underscores the robustness of LSTM networks in tackling complex forecasting challenges across different domains, reflecting their potential for wide-reaching impact (Chakraborty, 2023).

A. Iess et al., (2022) explored the utilization of recurrent neural networks (RNNs), specifically long short-term memory (LSTM) cells, for analyzing sequential data in the context of gravitational wave data analysis. The authors highlight the suitability of RNNs for this task due to their capacity to retain memory of previous inputs and consider them alongside current inputs. LSTMs are chosen as they address the challenge of vanishing gradients and can effectively handle long-term dependencies, making them well-suited for gravitational wave data analysis. The application of the proposed approach to gravitational wave data has been done from both a single detector and a three-interferometer network, aiming to detect core-collapse supernovae signals. The authors demonstrate the effectiveness of their LSTM-based method in identifying supernova signals and showcase its superior performance compared to a convolutional neural network (CNN)-based method.

The utilization of RNNs, particularly LSTMs, in the analysis of gravitational wave data is an emerging area of research. The results presented in the article suggest the potential of this approach in detecting various types of gravitational wave signals. The authors also suggest future investigations into real-time event detection using RNNs and the analysis of data from upcoming gravitational wave observatories. In summary, this study highlights the promising application of RNNs, specifically LSTMs, as a powerful tool for processing sequential data in the field of gravitational wave astronomy (A. less et al., 2022).

While the study does compare LSTM performance to a convolutional neural network (CNN) for the task of supernova signal detection, it does not benchmark against other established methods in gravitational wave data analysis. A more comprehensive evaluation that includes various state-of-the-art algorithms would provide a clearer perspective on LSTM's relative advantages and limitations in this domain.

In the realm of astrophysics and particle physics, the study of neutrinos and their astrophysical sources has gained considerable attention in recent years (ANTARES Collaboration et al., 2014). A key player in this endeavour is the Baikal-GVD neutrino telescope, located in Lake Baikal, Russia. This gigaton-scale underwater telescope comprises optical modules (OMs) distributed throughout the working volume, enabling the detection of photons generated by neutrino interactions. However, the data collected by the Baikal-GVD telescope are not without challenges, as the luminescence of the Baikal water can lead to background noise, necessitating efficient noise rejection algorithms.

As the search for astrophysical neutrinos intensifies, it is imperative to develop robust techniques for background noise rejection within neutrino telescopes. The filtering of noise plays a critical role in distinguishing neutrino-induced events from cosmic ray-induced ones, especially when events are non-vertical. The capability to accurately estimate the energy and arrival direction of incoming neutrinos relies on an effective noise rejection method. Therefore, the development of noise filtering algorithms is integral to enhancing the telescope's overall exposure (Kharuk et al., 2023).

The study presented by Kharuk et al. (2023) addresses the challenge of background noise rejection by introducing a novel machine learning-based approach, utilizing a neural network specifically tailored for the Baikal-GVD neutrino telescope. Monte Carlo simulations serve as the foundation for modelling the telescope's data, with an emphasis on identifying muon neutrinos—a subsample of the neutrino simulations. The decision to focus on muon neutrinos stems from the recognition that the telescope registers both muon and neutrino-induced events, making efficient noise rejection a prerequisite for the subsequent separation of these event types.

To evaluate the performance of the introduced neural network, standard metrics such as signal purity (precision) and survival efficiency (recall) are employed. In the case of a threshold value of 0.5, the neural network achieves approximately 97% signal purity and 99% survival efficiency, outperforming traditional algorithmic noise rejection methods, which attain only 95% for both metrics. The high precision and recall achieved by the neural network underscore its effectiveness in discerning signal hits from background noise, significantly contributing to the overall accuracy of data analysis for the Baikal-GVD telescope (Kharuk et al., 2023).

The neural network's architecture comprises a U-net followed by a bidirectional long short-term memory (LSTM) cell and a final convolution layer with softmax activation. This architecture is specifically designed to leverage both local and global information within an event, thereby facilitating the accurate identification of signal hits. The U-net effectively utilizes convolution operations, making it adept at capturing local information from neighbouring cells, while the LSTM cell enhances performance by considering the temporal order of hits within an event (Kharuk et al., 2023).

In the research presented by Kharuk et al. (2023), several limitations and research gaps warrant consideration:

The study relies on Monte Carlo simulations to model the telescope's data, which raises questions about the realism of the simulated data compared to actual observations. The realism of the dataset is a potential limitation, as simulations may not fully capture the complexities and uncertainties present in real-world data.

While the study mentions that the neural network is robust to minor perturbations in the activation times of hits, it does not delve into how the model handles more significant calibration errors or how these errors may impact its performance.

The paper primarily focuses on a single-cluster regime for background rejection, but it does not discuss the generalizability of the proposed method to scenarios with multiple clusters. Whether the same neural network architecture would be effective in such scenarios remains an open question, highlighting the need for further exploration of its generalizability.

# **Chapter 3**

## **Research Methodology**

## **Research Approach**

This study adopts a retrospective cross-sectional design, employing the method of secondary data analysis. It aims to investigate relationships and patterns within existing datasets, without direct involvement in the data collection process. By utilizing pre-existing information, this approach offers a cost-effective means of addressing research inquiries and testing hypotheses. The study will undertake a systematic examination of historical data, aiming to discern meaningful associations and trends that contribute to the existing knowledge in the relevant field of study. Through rigorous analysis and interpretation of the collected data, the study endeavours to shed light on significant relationships and provide valuable insights to the scholarly community.

The study is based on labelled quantitative data, which includes data points that are assigned specific labels or categories. This type of data allows for supervised learning, making it suitable for training and validating LSTM networks.

## **Data Sampling**

The approach to data collection in this study involves the utilization of a public dataset as a second-party sampling method. This choice is substantiated by several crucial considerations and justifications.

The IceCube Observatory represents a formidable scientific undertaking that accumulates a wealth of data pertaining to neutrino interactions. This publicly accessible dataset serves as an extensive and information-rich source for investigating the prediction of neutrino directions. By harnessing this pre-existing dataset, we gain access to a substantial volume of data that would otherwise be costly to gather independently.

The utilization of a large-scale public dataset like the IceCube dataset offers significant statistical advantages. This dataset contains a substantial number of neutrino events, thereby enabling robust statistical analyses and the detection of subtle patterns or deviations in neutrino direction prediction. The increased sample size enhances the reliability of the research findings, providing a solid foundation for drawing meaningful conclusions (Mooney and Garber, 2019).

## **Data Collection**

For this research study, the primary source of data will be obtained from the publicly available dataset provided by the IceCube Neutrinos in Deep Ice competition hosted on Kaggle (kaggle.com, n.d.). Kaggle is a widely recognized and reputable platform for data science and machine learning competitions, known for hosting high-quality and reliable datasets. The dataset is specifically designed to address the challenge of identifying the direction from which neutrinos detected by the IceCube neutrino observatory originated. The data collection process will be executed within a Jupyter Notebook environment, utilizing Python 3.11 as the primary programming language.

The computational infrastructure for this study is equipped with high-performance hardware, which includes a NVIDIA GeForce RTX 4070 graphics card boasting 12 gigabytes of dedicated memory. TensorFlow is well-suited for use with the CUDA architecture developed by Nvidia. This compatibility allows for faster training times on the GPU, enhancing the efficiency of machine learning tasks in this research. Additionally, the system is equipped with 16 gigabytes of system RAM, further enhancing the processing capabilities. Complementing the graphics card is the AMD Ryzen 5 7600X 6-core processor, ensuring efficient data processing and analysis throughout the research.

## **Data Files and Labels**

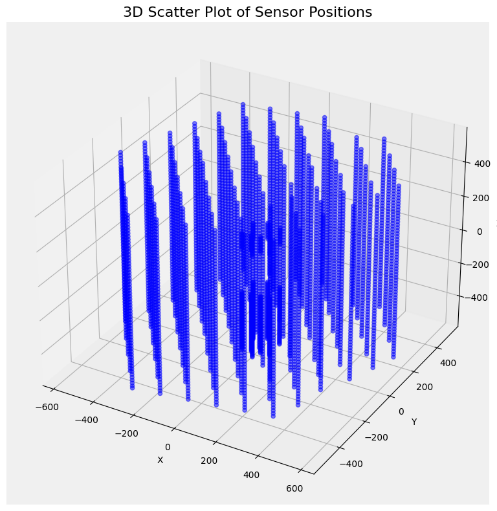


Figure 1 - Schematic of Sensor Geometry

#### *Sensor Geometry Data (sensor\_geometry.csv)*

The sensor geometry data is stored in CSV format as shown in Figure 1 and provides crucial information about the spatial layout of sensors within the IceCube neutrino observatory. This information is presented in the form of coordinates (x, y, z).

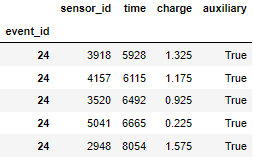


Figure 2- Preview displaying contents of batch\_1

#### Training Batch Data (batch\_1.parquet - batch\_660.parquet)

These Parquet format files represent different training batches. Each batch corresponds to a distinct training set, denoted by the numerical labels in the filenames (e.g., batch\_1, batch\_2, ..., batch\_660).

Each training batch contains the following features as shown in Figure 2:

* **sensor\_id**: An identifier for the sensor involved in the event.
* **time**: The timestamp of the recorded event.
* **charge**: Indicating the energy or charge associated with the event.
* **auxiliary**: If False, the event Contains only Neutrino induced signals. If True, the event contains background noise.

Each pulse has six properties: sensor ID (x, y, z), time, charge, and auxiliary. The auxiliary value determines the readout mode, either "False" for local coincidence or "True" for another mode. Choosing "False" improves event shape and angle reconstruction, so when embedding data, "False" pulses are prioritized when the model's dimension is exceeded.

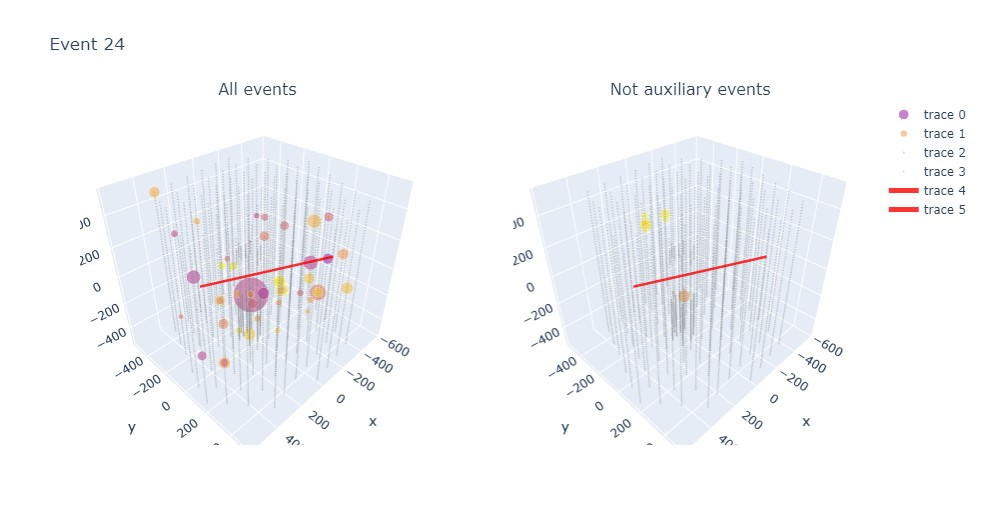


Figure 3- Example Event from the Dataset (Azimuth = 5.029 rad, Zenith= 2.087 rad) (kaggle.com, n.d.)

#### Event Identifier (event\_id)

Notably, each row within these training batch files is associated with an event ID, signifying an event pulse. This event identifier plays a pivotal role in linking and tracking the data within the training batches to specific events or occurrences within the IceCube neutrino observatory. As shown above in Figure 3.

The biggest challenge of the dataset is its size (more than 138 million events distributed in 660 batches with the size of over 100 GBs).

## **Data Preprocessing**

#### Sensor Geometry Data: Establishing Spatial Context

In order to develop an LSTM model architecture for Neutrino direction prediction, it is imperative to establish a clear spatial context within the neutrino detector. This context is enabled through the utilization of sensor geometry data, which encompasses the positions of sensors within the detector. In this section, we elucidate how this sensor geometry data is employed to gain a deeper understanding of the detector's dimensions and the conversion of time to distance.

##### Loading Sensor Geometry Data

Sensor geometry data serves as the foundation of our research, providing the spatial coordinates necessary for our analyses. The data is sourced from a CSV file named "sensor\_geometry.csv" and is loaded into a DataFrame. This data contains critical information about the location of sensors within the detector.

**( 1 )**

##### Min / Max Information

To comprehend the spatial extent of the neutrino detector, we calculate the minimum and maximum coordinates in the X, Y, and Z dimensions.

**( 2 )**

##### Detector Constants

Distance (metres) for Light to travel in one nanosecond.

**( 3 )**

##### Detector Valid Length

With the min and max coordinates established, we proceed to calculate what we term the "valid time length." This metric signifies the duration required for signals, such as neutrino interactions, to traverse the entire length of the detector. It is a critical parameter that bridges the temporal and spatial aspects of our research, allowing us to make accurate predictions.

The calculation of the valid time length involves determining the Euclidean distance between the minimum and maximum coordinates and then dividing this distance by the speed of light to obtain the time-based dimension. This duration is essential for understanding the temporal characteristics of events within the detector.

**( 4 )**

The resulting calculation produces the valid length of time for a pulse to fall within 6,199.700247 nanoseconds.

##### Significance of Sensor Geometry Data

* **Time-to-Distance Conversion:** By utilizing the speed of light constant and sensor coordinates, we establish a robust framework for converting time-based measurements to distance.
* **Event Localization:** The association of sensor coordinates with specific detectors allows us to precisely determine the positions of detected events. This spatial awareness forms the basis for predicting the directions of neutrino interactions.
* **Detector Dimensions:** Our understanding of the physical dimensions of the detector is a cornerstone of our research. It underpins accurate modelling and prediction of neutrino interactions and their directional properties.

##### Event Data Extraction

We begin by taking as input the index of an event (**event\_idx**) from a batch, along with several dataframes: **batch\_meta\_df**, **batch\_df**, and **sensor\_geometry\_df**. The purpose is to retrieve and format event-specific information for further analysis.

* **Metadata Retrieval**: We first extract key metadata related to the event, such as **batch\_id**, **first\_pulse\_index**, and **last\_pulse\_index**, which provide information about the batch to which the event belongs, and the range of pulse data associated with it.
* **Event Data Retrieval**: Subsequently, we extract event-specific features from the **batch\_df**. This includes the sensor ID and various event-specific properties like time, charge, and auxiliary information.

##### Structured Array Creation

To structure the extracted event data, we create a structured NumPy array (**event\_x**). This array is organized according to several predefined data types, including time, charge, auxiliary, and spatial coordinates (x, y, z).

* **Time Alignment**: The **time** values within **event\_x** is adjusted relative to the minimum time value in the event, ensuring that time is relative to the event start time.
* **Spatial Information**: The spatial information, i.e., **x**, **y**, and **z**, is obtained from the **sensor\_geometry\_df** and mapped to the corresponding sensor IDs.

##### Handling Long Events

In cases where the event contains more pulses than a predefined maximum count (**max\_pulse\_count**), a special procedure is applied. In our case, 96 pulses were determined to be the maximum. This is primarily to handle long events and ensure that the data is manageable.

* **Valid Time Window Selection**: A valid time window centred around the peak charge time is defined, and only pulses falling within this time window are considered.
* **Pulse Reduction**: For long events, only the most relevant pulses are retained, up to the specified **max\_pulse\_count**.
* **Time-Based Sorting**: The retained pulses are sorted based on their time, ensuring that they are ordered sequentially.

##### Target Variables

For the training data, the function incorporates the azimuth and zenith information associated with the event as target variables. These values are stored as a NumPy array.

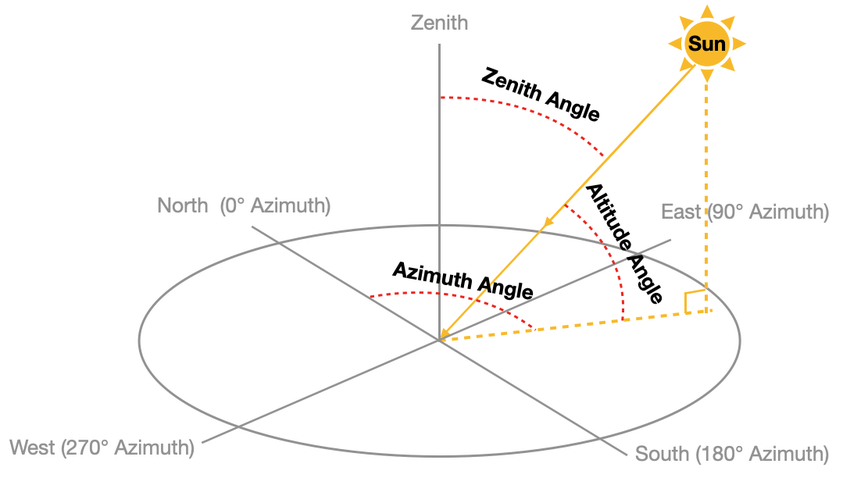


Figure 4 - Schematic depicting Zenith and Azimuth Angles (Zhang et al., 2021)

## **Data Metrics**

In this section, we introduce two essential scoring functions, angular distance score and one-hot encoding for angles, which play a pivotal role in assessing the quality of predictions in the context of azimuth and zenith angles. These functions are integral components of our research where we focus on accurate angular measurements. The conversion to Mean Angle Error (MAE) is provided by the hosts of the IceCube dataset, adding a significant layer of credibility to our research.

**Angular Distance Score**

The function is meticulously designed to calculate the Mean Absolute Error (MAE) of the angular distance between two sets of directions represented by azimuth and zenith values. The underlying principles of this function are as follows:

1. **Conversion to Cartesian Coordinates:** First, the azimuth and zenith angles are converted into Cartesian unit vectors. This transformation allows us to perform vector operations to quantify the angular difference accurately.
2. **Cosine Similarity Calculation:** We compute the cosine of the angle between these vectors by taking their dot product. This step is critical for determining the angular separation between the predicted and true directions.
3. **Angular Distance Calculation:** To arrive at the final angular distance, we take the inverse cosine of the resulting scalar product. This calculation yields the angular distance between the two input vectors in radians.

The function takes as input the true azimuth and zenith values as well as the predicted azimuth and zenith values. It returns the mean angular distance in radians. This score is a robust measure of prediction accuracy, making it an essential component of our research.

**Azimuth and Zenith Angles**

Azimuth represents the angle indicating the sun's direction, measured in radians clockwise from the north along the horizon. Zenith angle, on the other hand, is the angle between the local zenith and the line connecting an observer to the sun as shown in Figure 4. In our dataset, these values are provided in radians, with azimuth ranging from 0 to 2π (equivalent to 0° to 360°) and zenith values ranging from 0 to π (which corresponds to 0° to 180°).

**Angle One-Hot Encoding**

**Azimuth Distribution**

The azimuth distribution is assumed to be flat, meaning that azimuth values are uniformly distributed as evidenced in Figure 5, on the spherical surface.

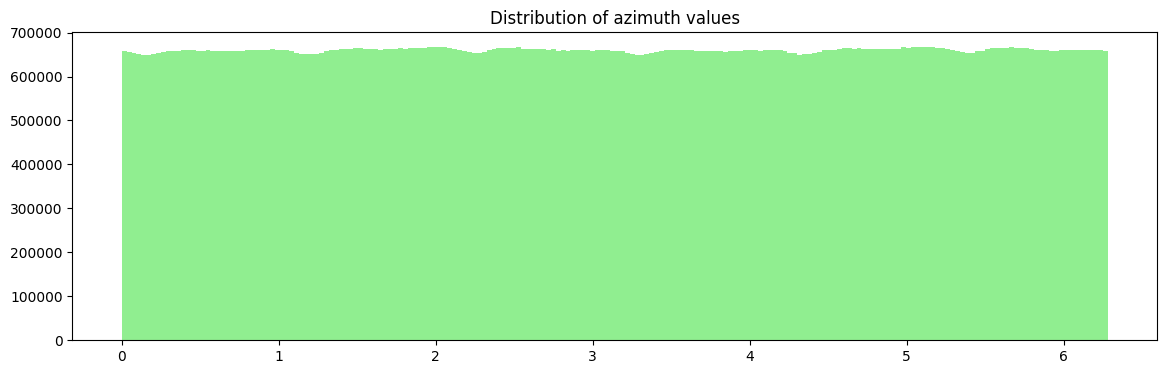


Figure 5 - Distribution of Azimuth values

**Zenith Distribution**

In contrast to azimuth, the zenith distribution follows a sine pattern (normal distribution) as evidenced in Figure 6. This reflects the nature of zenith angles and their distribution over the unit sphere.

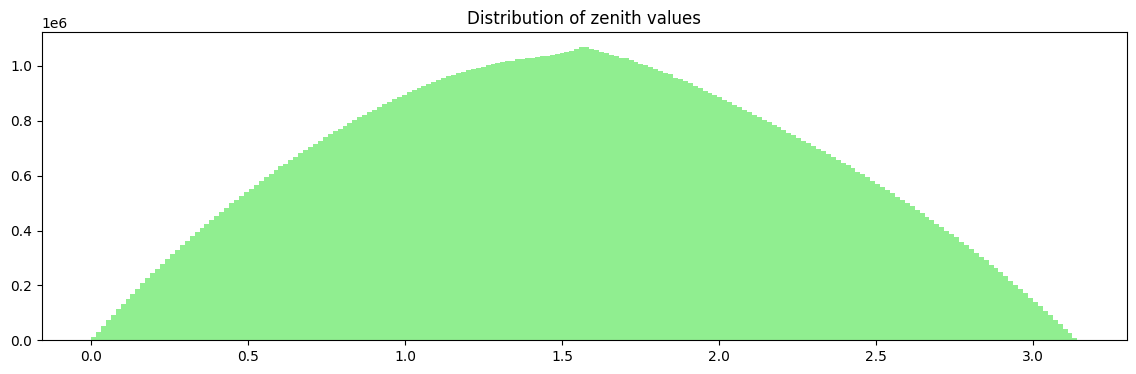


Figure 6 - Distribution of Zenith values

**Binning Strategy**

The encoding strategy employs distinct binning approaches for azimuth and zenith angles:

* **Azimuth Angles:** We utilize a uniform binning approach. The azimuth edges are defined to represent a uniform distribution of azimuth values ranging from 0 to 2π, with the number of bins specified by the bin number.
* **Zenith Angles:** For zenith angles, the zenith edges are defined to represent a uniform distribution of zenith values ranging from 0 to π, with the number of bins determined by bin number. These edges are calculated iteratively to ensure that the cosine values between two consecutive bins differ by a constant amount. This ensures an accurate representation of zenith angles in our encoding.

**Function for One-Hot Encoding of Target Variables**

To prepare target variables (azimuth and zenith) for machine learning models, we have implemented the **y\_to\_onehot** function. This function is designed to convert azimuth and zenith values into one-hot encoded representations. The process involves several steps:

1. **Bin Code Evaluation:** The function evaluates bin codes for azimuth and zenith by comparing the values against the defined bin edges. This step categorizes the angles into specific bins.
2. **Angle Code Calculation:** To create a unified representation for azimuth and zenith, an angle code is calculated by combining the azimuth and zenith bin codes.
3. **One-Hot Encoding:** Finally, the function creates a one-hot encoded representation for the angle code. This results in an array of shape **(batch\_size, bin\_num \* bin\_num)**, where **batch\_size** represents the number of events, and **bin\_num** signifies the number of bins.

**Conversion of Predictions to Angles**

The hosts of the dataset also provide formulas for converting predictions into meaningful angle representations (Eller, 2023). These formulas calculate the mean vector within a specific bin defined by zenith and azimuth ranges. The calculations are based on integrating over the specified ranges, considering spherical coordinates (θ, φ).

**( 5 )**

## **Model Architecture**

**Shared Components**

**Model Input**

The initial component of the model is represented by an input layer, which is designed to accommodate data sequences characterized by two specific dimensions: **max\_pulse\_count** and **n\_features**. These dimensions delineate the maximum extent of temporal steps within the input data sequence and the count of distinctive features attributed to each temporal step, respectively.

**Dense Layers**

Following the sequence analysis layers, the model architecture in all three cases includes two shared Dense layers. The initial Dense layer is comprised of 256 units and applies the Rectified Linear Unit (ReLU) activation function, a fundamental component that introduces non-linear characteristics into the model (builtin.com, n.d.). The subsequent Dense layer, designated as the output layer, similarly features 256 units, and adopts the softmax activation function. This choice is of paramount significance, especially in the context of multi-class classification tasks, as it effectively assigns probabilities to diverse classes (Pinecone, n.d.).

**Model Compilation**

In the model compilation step, a common set of parameters is employed across all three architectures. The model is compiled using the Adam optimizer. Adam is proficient in adapting the learning rates individually for each parameter during the training process. It does so by maintaining two dynamic averages, namely the first moment (mean) and the second moment (uncentered variance) of the gradients. This adaptability ensures that Adam can effectively accommodate varying learning rates for each parameter, consequently leading to expedited convergence and enhanced generalization. The selected loss function is Categorical Crossentropy. The primary objective of a multi-class classification task is to assign a class label to each input. Categorical Crossentropy quantifies the dissimilarity between the predicted class probabilities and the actual class labels. Minimizing this loss function aligns the model's predictions with the ground truth. Finally, accuracy is used as the primary evaluation metric during training and model performance assessment (Terven et al., 2023).

#### **Model 1 – Long Short-Term Memory Model**

**Masking Layer**

Following the input layer, the data is passed through a Masking layer. This layer is employed to handle sequences with padding, ensuring that any padded values are effectively ignored during the model's analysis. The objective here is to maintain the integrity of the temporal sequence data by not considering extraneous padded values.

**Bidirectional LSTM Layers**

The central component of this model architecture consists of multiple Bidirectional Long Short-Term Memory (LSTM) layers. Bidirectional LSTMs are a specialized type of recurrent neural network (RNN) that can capture temporal dependencies in both forward and backward directions within a sequence. The architecture includes the following Bidirectional LSTM layers:

1. **First Bidirectional LSTM Layer**: This layer employs 64 LSTM units, and is configured to return sequences, meaning it produces a sequence output.
2. **Second Bidirectional LSTM Layer**: Following the first LSTM layer, a second Bidirectional LSTM layer with 256 units and sequence output is utilized. This layer serves to further analyse the sequence data and extract higher-level temporal features.
3. **Third Bidirectional LSTM Layer**: In this layer, the sequence output from the second LSTM layer is added to the output of the first LSTM layer, thereby combining the information from both LSTM layers. This additive approach aims to capture more complex dependencies.

**Concatenation Layer**

Subsequent to the Bidirectional LSTM layers, the model incorporates a Concatenation layer. This layer is responsible for merging the results obtained from the first and third Bidirectional LSTM layers. The objective is to consolidate the information extracted from these distinct layers into a unified representation for further processing.

**Additional Bidirectional LSTM Layer**

Following the Concatenation step, an additional Bidirectional LSTM layer is introduced. This layer builds upon the information collected from the previous stages, aiming to capture higher-level temporal patterns and dependencies.

Model architecture is visually described under Addendum A – LSTM Model Architecture, Figure 7.

#### **Model 2 – Gated Recurrent Unit Model**

**Bidirectional GRU Layers**

The core of the model architecture remains focused on leveraging Bidirectional Gated Recurrent Unit (GRU) layers. Bidirectional GRU layers offer similar sequence modelling capabilities to LSTM layers but are computationally more efficient. The architecture incorporates the following Bidirectional GRU layers:

1. **First Bidirectional GRU Layer**: This layer employs 192 GRU units, and is configured to return sequences, which means it produces a sequence output.
2. **Second Bidirectional GRU Layer**: Following the first GRU layer, a second Bidirectional GRU layer is introduced. It retains the same number of units and returns sequence output, further enhancing the model's capacity to analyse sequence data.
3. **Third Bidirectional GRU Layer**: In this layer, the sequence output from the second GRU layer is used as input, and the Bidirectional GRU is employed to capture temporal dependencies. This approach aims to extract more complex temporal patterns and information from the data.

The model architecture is visually described under Addendum B – GRU Model Architecture, Figure 8.

#### **Model 3 – LSTM + GRU Model**

**Bidirectional LSTM Layers**

The architecture commences with Bidirectional LSTM layers designed to capture temporal dependencies:

1. **First Bidirectional LSTM Layer**: This layer utilizes 64 units and is configured to return sequences. It plays a pivotal role in capturing temporal dependencies, and the **return\_sequences=True** setting ensures that it produces a sequence output.
2. **Second Bidirectional LSTM Layer**: Following the first LSTM layer, a second Bidirectional LSTM layer with 64 units and sequence output configuration is employed. This layer further enhances the model's ability to analyse sequence data and extract relevant information.

**Bidirectional GRU Layers**

In addition to LSTM layers, the model also integrates Bidirectional Gated Recurrent Unit (GRU) layers, which are known for their computational efficiency while effectively modelling sequences:

1. **First Bidirectional GRU Layer**: This layer employs **LSTM\_width** units and is configured to return sequences, similar to the first Bidirectional LSTM layer. It aims to capture temporal dependencies and offers a different approach to sequence modelling.
2. **Second Bidirectional GRU Layer**: Building on the information from the first GRU layer, a second Bidirectional GRU layer with the same settings for units and sequence output is introduced. This further enriches the model's understanding of sequence data.

**Concatenation of LSTM and GRU Layers**

After the Bidirectional LSTM and GRU layers, the architecture incorporates a Concatenation layer. This layer is essential for merging the results obtained from the Bidirectional LSTM and GRU layers. It combines the information extracted from these different types of recurrent layers into a unified representation for further processing.

**Bidirectional LSTM Layer**

Following the Concatenation step, the model features another Bidirectional LSTM layer. This layer leverages the combined information to capture higher-level temporal patterns and dependencies within the data.

The model architecture is visually described under Addendum C – LSTM + GRU Model Architecture, Figure 9.

## **Research Ethics**

In the course of conducting our research study on the IceCube Observatory dataset, we must be diligent in addressing various ethical considerations. To begin, safeguarding the privacy and confidentiality of the individuals or entities represented within the dataset is of paramount importance. Given that the dataset encompasses details related to 5160 sensors and their associated events, it is incumbent upon us to take measures that protect any personally identifiable information. This entails handling the data in full compliance with privacy regulations and established guidelines.

When working with labelled data, it is imperative to ensure that appropriate informed consent has been duly obtained. The hosts of the dataset have explicitly granted permission for participants to utilize the data for academic research and educational purposes. It is imperative that we adhere rigorously to the boundaries established by the hosts' rules and guidelines. Respect for the terms of use specified by the hosts is non-negotiable, and any use of the data beyond the realm of academic research or education is strictly proscribed.

Biases and issues of fairness also warrant careful consideration. It is crucial to remain cognizant of potential biases that might exist within the dataset, including imbalances in the representation of different sensor locations or event characteristics. To ensure equity throughout the analysis and modelling process, proactive steps should be taken to recognize and mitigate these biases.

# **Chapter 4**

## **Data Analysis and Results**

**Table 1 - Comparison of Model Architecture**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Batch size | Input Features | Trainable Parameters | Input Width | Output Dimension | Layers |
| LSTM | 512 | (2,090,000, 96, 6) | 1,383,424 | 64 | 256 | 6 |
| GRU | 512 | (2,090,000, 96, 6) | 1,726,464 | 192 | 256 | 5 |
| LSTM + GRU | 512 | (2,090,000, 96, 6) | 500,480 | 64 | 256 | 7 |

In **Table 1**, we present a comparison of model architectures for different recurrent neural network (RNN) variants: LSTM, GRU, and a combination of LSTM and GRU (LSTM + GRU). The focus of this comparison is on key aspects such as the number of trainable parameters, input width, output dimension, and the number of layers in each model. This analysis is essential for understanding the trade-offs and performance characteristics of these RNN architectures in the context of the research objective.

In assessing model complexity, it is observed that all models share a similar output dimension but exhibit differences in the number of trainable parameters. Specifically, the GRU model is marginally more complex than the LSTM. However, the LSTM + GRU hybrid model demonstrates an intriguing aspect, as it manages to reduce the overall parameter count while maintaining the same output dimension with an increase in layers.

**Table 2 - Comparison of Model Performance**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Epochs | Time per Epoch | Total time to Completion | Best Epoch | Accuracy | Validation Accuracy | Loss | Validation Loss | Mean Angular Error |
| LSTM | 20 | 394s | 7,880s | 12 | 0.1687 | 0.1525 | 4.3736 | 4.4675 | 1.0585 |
| GRU | 20 | 230s | 4,600s | 11 | 0.1663 | 0.1508 | 4.3882 | 4.4805 | 1.0606 |
| LSTM + GRU | 20 | 244s | 4,880s | 19 | 0.1546 | 0.1456 | 4.4461 | 4.5010 | 1.0681 |

The LSTM, GRU, and LSTM GRU models are subjected to scrutiny across several dimensions as shown above in **Table 2**, including training epochs, computational efficiency, accuracy, and the pivotal mean angular error.

After completing 12 epochs, the LSTM model, trained over a duration of 20 epochs, achieved an accuracy of 16.87%, referred to Figure 8. Notably, the model's performance did not exhibit further improvement during the subsequent 8 epochs. However, one drawback of the LSTM model is its extensive training time, which amounted to 7,880 seconds (approximately 2 hours and 11 minutes).

The GRU model underwent a training duration of 20 epochs similar to that of the LSTM. It showcased a competitive accuracy of 0.1663 after completing 11 epochs, referred to Figure 11. The model’s performance did not improve during the subsequent 9 epochs. What distinguishes the GRU model is its validation accuracy, which stood at 15.08%, underscoring its effectiveness in minimizing the mean angular error to 1.0606. An appreciable advantage lies in its computational efficiency, with each epoch completing in approximately 230 seconds. The total training time, amounting to 4,600 seconds (equivalent to about 1 hour and 16 minutes), significantly outpaces the LSTM model in terms of efficiency.

The combined LSTM and GRU model show the poorest performance. After having undergone training over 20 epochs, the model’s optimal performance was reached after 19 epochs with a validation accuracy of 14.56% and a mean angular error of 1.0681, as depicted in Figure 14. the total training time for the model was closely similar to that of the GRU model at 4,880 seconds (equivalent to 1 hour and 21 minutes).

Comparing the performance of different recurrent neural network (RNN) architectures, it is evident that the LSTM model stands out for its ability to capture temporal patterns. However, it comes with a notable drawback in terms of training time, which can make it less practical for real-time or time-sensitive applications. On the other hand, the GRU model offers a compelling alternative, displaying competitive accuracy in pattern recognition while significantly reducing training time compared to LSTM. In contrast, the combined LSTM and GRU model, which aims to leverage the strengths of both architectures, surprisingly exhibited the poorest performance among the three. It boasted the advantage of having a lower number of parameters, potentially reducing computational overhead, but it failed to deliver on its promise in terms of overall predictive accuracy.

**Table 3 - Measures of Statistical Tests**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Azimuth  Correlation | Azimuth  Mean Absolute Error | Zenith  Correlation | Zenith  Mean Absolute Error |
| LSTM | 0.2923 | 1.4620 | 0.3712 | 0.6746 |
| GRU | 0.2852 | 1.5005 | 0.3693 | 0.6752 |
| LSTM + GRU | 0.2776 | 1.5078 | 0.3637 | 0.6695 |

**Azimuth Predictions:**

The LSTM model demonstrated an azimuth correlation of 0.2923, indicating a moderate positive linear relationship between its predictions and the actual azimuth values. This suggests that the LSTM model has some capability to capture the directional patterns in the data. Additionally, the azimuth Mean Absolute Error (MAE) of 1.4620 signifies that, on average, the LSTM model's predictions deviate from the actual azimuth values by approximately 1.4620 units.

Similarly, the GRU model exhibited a slightly lower azimuth correlation of 0.2852, which implies a similar moderate positive linear relationship with the actual azimuth values compared to the LSTM. The azimuth MAE for the GRU model is 1.5005, indicating comparable performance to the LSTM in azimuth prediction.

In contrast, the LSTM + GRU model showed the lowest azimuth correlation among the three models, with a value of 0.2776. This indicates a somewhat weaker linear relationship with the actual azimuth values. The model achieved the lowest azimuth MAE of 1.5078, suggesting a trade-off between correlation and prediction accuracy.

**Zenith Predictions:**

The LSTM model attained a zenith correlation of 0.3712, signifying a moderate positive linear relationship with the actual zenith values. Notably, the zenith Mean Absolute Error (MAE) for the LSTM model was 0.6746, a relatively low value that indicates accurate zenith predictions.

Similarly, the GRU model achieved a zenith correlation of 0.3693, putting it on par with the LSTM in terms of linear relationship strength with the actual zenith values. The zenith MAE for the GRU model was 0.6752, which is comparable to the LSTM, underscoring the accuracy of zenith predictions by the GRU.

Conversely, the LSTM + GRU model achieved a zenith correlation of 0.3637, slightly lower than both the LSTM and GRU models. However, this still indicates a moderate positive linear relationship with the actual zenith values. What distinguishes the LSTM + GRU model is its zenith MAE, which was the lowest among the three models at 0.6695.

## **Measures of Validity**

**Accuracy and Validation Accuracy**:

Accuracy is a commonly used metric for classification tasks and is defined as the ratio of correctly classified samples to the total number of samples.

**( 6 )**

Validation accuracy assesses the model's performance on unseen data and is crucial for understanding how well a model generalizes. The limitation of accuracy is that it treats all prediction errors equally.

**Mean Angular Error**:

The Mean Angular Error (MAE) measures the average angular difference between predicted and actual values. In the context of direction prediction, it's relevant to assess the error in angular predictions. The formula, as shown by **( 5 )** depicts the conversion of predictions to their angular representation.

**Azimuth and Zenith Correlation**:

Correlation measures the linear relationship between two variables. In this context, it measures the linear relationship between predicted and actual azimuth or zenith values.

The Pearson correlation coefficient, which ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation), quantifies this relationship.

**( 7 )**

**Mean Absolute Error (MAE)**:

MAE is a robust metric for measuring prediction accuracy, as it calculates the absolute difference between predicted and actual values, which helps in understanding the magnitude of errors. MAE is particularly useful for understanding the average prediction error across all data points. A lower MAE indicates that the model's predictions are, on average, closer to the actual values.

**( 8 )**

### **Content Validity**

According to Almanasreh, Moles and Chen (2019), content validity is met when the accuracy, validation accuracy, and MAE metrics align with the models’ objectives. Content validity is maintained as long as these metrics are applied in assessing the research objectives and evaluating model performance in classification tasks and assessing prediction errors.

### **Construct Validity**

Roche et al. (2021) discuss metrics appropriate for measuring the intended constructs such as model performance. Accuracy, MAE, and correlation align with the theoretical constructs of machine learning model evaluation. Ensuring that they accurately reflect the underlying model performance is vital for maintaining construct validity.

## **Measures of Reliability and Trustworthiness**

### **Test-retest**

Test-retest reliability is a crucial concept in research. It assesses the consistency and stability of measurements or test results over time. The fundamental idea behind test-retest reliability is to determine whether a measurement or assessment produces similar results when performed on the same data set or on two or more separate occasions (Schaaf et al., 2023).

In test-retest reliability studies, it is vital not only to evaluate the consistency of measurements but also to control for potential sources of variation that could affect the results between the initial test and the re-test. One often overlooked factor is the computational infrastructure used for data analysis, particularly when machine learning or data-intensive tasks are involved.

Using similar high-performance hardware ensures that data processing and analysis are carried out consistently during both the initial test and the re-test. If participants use significantly different hardware, variations in computation power may lead to discrepancies in results. Variability in hardware can introduce unwanted variations in results. By using hardware with similar specifications, participants can help minimize potential sources of variability and improve the likelihood of obtaining consistent and reliable results in both test phases. Adequate memory and processing capabilities, such as those provided by high-performance hardware, facilitate efficient data handling. This is crucial for maintaining consistency in data preprocessing and analysis procedures, reducing the risk of measurement errors.

### **Transparency**

We have ensured methodological transparency in the research by providing comprehensive documentation of the data collection procedures. This allows for replication, peer review, and accountability, minimizing bias and error. It also supports building on the work performed. Transparency enhances the trustworthiness of my research and its long-term impact.

# **Chapter 5**

## **Findings**

The research successfully implemented a comprehensive data preprocessing pipeline, which involved event data retrieval, structured array creation, and handling of long events. These preprocessing steps aimed to prepare the data for machine learning models effectively. This aligns with the research's objective to ensure data quality and consistency.

The study focused on comparing three different recurrent neural network (RNN) architectures, namely LSTM, GRU, and a hybrid LSTM + GRU model. The aim was to determine which model architecture performs best for predicting azimuth and zenith angles. The study compared LSTM, GRU, and LSTM + GRU models. Notably, the LSTM model stood out, achieving the highest accuracy and correlation with actual azimuth and zenith values. However, this model had a significant drawback in terms of training time, making it less practical for real-time applications. In contrast, the GRU model offered a competitive alternative with reduced training time and similar performance. Surprisingly, the LSTM + GRU model, which aimed to leverage the strengths of both architectures, showed the poorest performance.

The research used various metrics to assess the performance of the models, including accuracy, validation accuracy, mean angular error, correlation, and mean absolute error (MAE). These measures align with the objective of evaluating the models' performance for the task of predicting azimuth and zenith angles.

**Applying Deep Learning for Neutrino Direction Prediction:**

In our study, we focused on the application of Long Short-Term Memory (LSTM) Neural Networks to predict the direction of neutrino particles. The primary metric we used to evaluate the accuracy of our predictions was the Mean Angular Error, which is a crucial measure for assessing the reliability of the model's predictions. Our LSTM model achieved a Mean Angular Error of 1.0585, signifying the accuracy of our predicted neutrino directions.

A Mean Angular Error of 1.0585 indicates that our LSTM model provides precise predictions of neutrino particle directions. This is in line with our first research objective, which was to explore the feasibility of utilizing deep learning techniques, specifically LSTM, for this specific task. The result confirms that LSTM networks are indeed capable of accurately predicting the direction of neutrino particles.

## **Comparison with Baseline Model**

The baseline GNN model achieved a Mean Angular Error of 1.018, which served as the benchmark for evaluating model performance in predicting azimuth and zenith angles. In contrast, the implemented models, including the LSTM, GRU, and LSTM + GRU models, exhibited higher Mean Angular Errors (1.0585, 1.0606, and 1.0681, respectively).

This comparison suggests that the baseline GNN model outperformed the LSTM, GRU, and LSTM + GRU models in terms of accuracy in predicting azimuth and zenith angles. While the Mean Angular Errors of the implemented models are relatively close to that of the baseline GNN model, the GNN model achieved a lower error, indicating more accurate predictions.

## **Limitations**

The research presented several limitations that influenced the scope and performance of the study. These limitations primarily revolved around hardware constraints and their impact on data handling and model complexity.

One of the prominent limitations was the restriction of RAM to 16GB. This limitation had significant implications on data handling, as it constrained the ability to read and process large datasets efficiently. In the context of this study, the research was only able to train on 11 batches of data due to RAM limitations. This is in contrast to the baseline GNN model, which had the capacity to train on 50 batches. The inability to load and process a larger volume of data can hinder the potential for model generalization and predictive accuracy. As a result, the research was limited to a smaller subset of the available data, which may not fully represent the diversity and complexity of high-energy neutrino events.

Training the models on a local GPU with a dedicated memory of 9.8GB imposed further constraints on the complexity of the models. The available memory on the GPU dictates the size and complexity of neural network architectures that can be accommodated. In practice, this limitation may lead to a reduced number of trainable parameters, potentially preventing the exploration of more intricate model designs that could have improved predictive accuracy. Additionally, it limited the batch size during training, which influences training speed and the quality of learned representations.

The combination of RAM limitations and the constraints of the local GPU had a cascading effect on the complexity of the implemented models. In particular, the LSTM, GRU, and LSTM + GRU models were simplified to accommodate the memory limitations. Reduced model complexity could potentially compromise the models' ability to capture intricate temporal patterns and dependencies in the data. This limitation is especially relevant when comparing the research findings to the baseline GNN model, which was able to train on a more extensive dataset with presumably more complex architecture.

These limitations collectively restricted the research's capacity to perform extensive data processing and model training. While the research produced valuable insights and comparative analyses of different RNN architectures, the hardware limitations posed significant barriers to achieving the same level of predictive accuracy as the baseline GNN model, which had access to a larger training dataset and potentially more complex model architecture.

Researchers seeking to build upon this study should consider addressing these limitations by employing more robust computational resources and exploring techniques for handling larger datasets. These improvements may lead to more accurate and sophisticated models for predicting azimuth and zenith angles in high-energy neutrino events.

## **Conclusion**

The research conducted on the IceCube Observatory dataset has provided valuable insights into the application of recurrent neural network (RNN) models for the prediction of azimuth and zenith angles. The study aimed to address the complex challenge of accurately forecasting these angles in the context of high-energy neutrino events. To this end, various RNN architectures, including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and a combination of LSTM and GRU, were implemented and rigorously evaluated.

One of the significant findings of this research is that the Mean Angular Error (MAE) of the implemented RNN models, specifically the LSTM (1.0585), GRU (1.0606), and LSTM + GRU (1.0681), while competitive, was found to be higher than the baseline Graph Neural Network (GNN) model, which achieved an MAE of 1.018. This result highlights the ability of the baseline GNN model in predicting azimuth and zenith angles. The comparisons drawn between the RNN models and the baseline GNN model emphasize the need for further exploration of different model architectures and methodologies to improve predictive accuracy.

The study revealed that the LSTM model exhibited a stronger ability to capture temporal patterns, albeit with the trade-off of an extensive training time. In contrast, the GRU model provided a compelling alternative, delivering competitive accuracy while significantly reducing the training duration. The combination of LSTM and GRU, despite its potential for reduced computational overhead, demonstrated the poorest performance among the three models.

Throughout the research, ethical considerations, such as privacy regulations and adherence to the terms of use specified by the dataset hosts, were diligently addressed.

In terms of the research objectives, the findings shed light on the challenges and opportunities in the accurate prediction of azimuth and zenith angles for high-energy neutrino events. The results underscore the need for a holistic exploration of model architectures beyond RNNs and the potential of incorporating GNN models to achieve superior predictive accuracy. Future research endeavours can build upon this foundation, delving deeper into novel techniques and approaches to advance the state of the art in this field.

The limitations of this research influenced the scope and complexity of our models. These limitations resulted in a reduced dataset size and simplified model architectures, affecting the research's ability to match the performance of the baseline GNN model.

In light of these limitations, future research in this area should focus on addressing hardware constraints and exploring techniques for handling larger datasets. This would enable the development of more accurate and sophisticated models for predicting azimuth and zenith angles in high-energy neutrino events.

The research has contributed to our understanding of the strengths and limitations of RNN models for azimuth and zenith angle prediction, laying the groundwork for continued investigation and innovation. The interplay between ethical considerations, model performance, and the exploration of different architectures has led to a comprehensive exploration of this complex problem.

## **Future Research**

Future research in this domain could expand the horizons by delving into alternative model architectures, such as Transformers and ensemble methods. Transformers, known for their effectiveness in handling sequential data, offer promising potential for improving predictive accuracy in high-energy neutrino event direction prediction tasks. Additionally, the exploration of ensemble methods, which combine the strengths of multiple models, could enhance prediction robustness and reliability. These avenues open exciting possibilities for advancements in the field of high-energy physics and machine learning.

# **References**

A. Iess, Cuoco, E., Morawski, F., Nicolaou, C. and O. Lahav (2022). LSTM and CNN application for core-collapse supernova search in gravitational wave real data. 669, pp.A42–A42. doi:https://doi.org/10.1051/0004-6361/202142525

Abbasi, R., Ackermann, M., Adams, J., Aggarwal, N., Aguilar, J.A., Ahlers, M., Ahrens, M., Alameddine, J.M., Alves, A.A., Amin, N.M., Andeen, K., Anderson, T., Anton, G., Argüelles, C., Ashida, Y., Athanasiadou, S., Axani, S., Bai, X., Balagopal V., A. and Baricevic, M. (2022). Graph Neural Networks for low-energy event classification & reconstruction in IceCube. *Journal of Instrumentation*, 17(11), p.P11003. doi:https://doi.org/10.1088/1748-0221/17/11/p11003

Almanasreh, E., Moles, R. and Chen, T.F. (2019). Evaluation of methods used for estimating content validity. *Research in Social and Administrative Pharmacy*, 15(2), pp.214–221. doi:https://doi.org/10.1016/j.sapharm.2018.03.066.

ANTARES Collaboration, Adrián-Martínez, S., Albert, A., André, M., Anghinolfi, M., Anton, G., Ardid, M., Aubert, J.-J. ., Baret, B., Barrios-Martí, J., Basa, S., Bertin, V., Biagi, S., Bogazzi, C., Bormuth, R., Bou-Cabo, M., Bouwhuis, M.C., Bruijn, R., Brunner, J. and Busto, J. (2014). Searches for Point-like and extended neutrino sources close to the Galactic Centre using the ANTARES neutrino Telescope. *The Astrophysical Journal*, [online] 786(1), p.L5. doi:https://doi.org/10.1088/2041-8205/786/1/L5.

Baron, D. (2019). MACHINE LEARNING IN ASTRONOMY: A PRACTICAL OVERVIEW. [online] Available at: <https://arxiv.org/pdf/1904.07248.pdf>

Brown, L.M. (1978). The idea of the neutrino. Physics Today, 31(9), pp.23–28. doi:https://doi.org/10.1063/1.2995181

builtin.com. (n.d.). *ReLU Activation Function Explained | Built In*. [online] Available at: https://builtin.com/machine-learning/relu-activation-function.

Chakraborty, N. (2023). *Testing RNN-LSTM Forecasting with Simulated Time-Series : Exploring Univariate Sensitivit*. [online] Europe PMC. Available at: https://europepmc.org/article/ppr/ppr718080 [Accessed 25 Oct. 2023].

Chen, H., Engkvist, O., Wang, Y., Olivecrona, M. and Blaschke, T. (2018). The rise of deep learning in drug discovery. Drug Discovery Today, [online] 23(6), pp.1241–1250. doi:https://doi.org/10.1016/j.drudis.2018.01.039

Christiansen, M.H. (2018). Implicit Statistical Learning: A Tale of Two Literatures. *Topics in Cognitive Science*, 11(3), pp.468–481. doi:https://doi.org/10.1111/tops.12332.

Eller, P. (2023). *Public Kaggle Competition ‘IceCube -- Neutrinos in Deep Ice’*. [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.2307.15289.

Frost, R., Armstrong, B.C. and Christiansen, M.H. (2019). Statistical learning research: A critical review and possible new directions. *Psychological Bulletin*, [online] 145(12), pp.1128–1153. doi:https://doi.org/10.1037/bul0000210.

Giacomelli, G. (2009). The standard model of particle physics. Neutrino oscillations. *Radiation Measurements*, 44(9-10), pp.826–833. doi:https://doi.org/10.1016/j.radmeas.2009.10.083

Glaser, C., McAleer, S., Stjärnholm, S., Baldi, P. and Barwick, S.W. (2022). Deep-learning-based reconstruction of the neutrino direction and energy for in-ice radio detectors. *Astroparticle Physics*, 145, p.102781. doi:https://doi.org/10.1016/j.astropartphys.2022.102781

Glaser, C., Nelles, A., Plaisier, I., Welling, C., Barwick, S.W., García-Fernández, D., Gaswint, G., Lahmann, R. and Persichilli, C. (2019). NuRadioReco: a reconstruction framework for radio neutrino detectors. *The European Physical Journal C*, 79(6). doi:https://doi.org/10.1140/epjc/s10052-019-6971-5

kaggle.com. (n.d.). *❄️IceCube Neutrinos - Domain & EDA for DS folks*. [online] Available at: https://www.kaggle.com/code/mvvppp/icecube-neutrinos-domain-eda-for-ds-folks [Accessed 28 Oct. 2023].

kaggle.com. (n.d.). IceCube - Neutrinos in Deep Ice. [online] Available at: https://www.kaggle.com/competitions/icecube-neutrinos-in-deep-ice/data [Accessed 4 Jun. 2023]

Kharuk, I., Safronov, G., Matseiko, A. and Leonov, A. (2023). *Machine learning in Baikal-GVD experiment PoS(ICRC2023)1077*. [online] Available at: https://pos.sissa.it/444/1077/pdf [Accessed 25 Oct. 2023]

Menghani, G. (2023). Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better. ACM Computing Surveys. doi:https://doi.org/10.1145/3578938

Mooney, S.J. and Garber, M.D. (2019). Sampling and Sampling Frames in Big Data Epidemiology. Current Epidemiology Reports, 6(1), pp.14–22. doi: <https://doi.org/10.1007/s40471-019-0179-y>

Pinecone. (n.d.). *Softmax Activation Function: Everything You Need to Know*. [online] Available at: <https://www.pinecone.io/learn/softmax-activation/>

Plaisier, I., Bouma, S. and Nelles, A. (2023). Reconstructing the arrival direction of neutrinos in deep in-ice radio detectors. 83(5). doi: https://doi.org/10.1140/epjc/s10052-023-11604-w

Roche, C., Kumar, V., Overman, S., Simovitch, R., Flurin, P.-H., Wright, T., Routman, H., Teredesai, A. and Zuckerman, J. (2021). Validation of a machine learning–derived clinical metric to quantify outcomes after total shoulder arthroplasty. *Journal of Shoulder and Elbow Surgery*. doi:https://doi.org/10.1016/j.jse.2021.01.021

Schaaf, J.V., Weidinger, L., Molleman, L. and van den Bos, W. (2023). Test–retest reliability of reinforcement learning parameters. *Behavior Research Methods*. [online] doi:https://doi.org/10.3758/s13428-023-02203-4

Shields, P., Peirce, S. and Dewey, J. (n.d.). PRAGMATISM AS PHILOSOPHY OF SCIENCE: A TOOL FOR PUBLIC ADMINISTRATION All rights of reproduction in nny fornl reserved. Rerearch in Public Admlnlslmtian, [online] 4, p.225. Available at: <https://digital.library.txstate.edu/bitstream/handle/10877/3954/fulltext.pdf?sequence=1&isAllowed=y>

Søgaard, A., Ørsøe, R.F., Holm, M., Bozianu, L., Rosted, A., Petersen, T.C., Iversen, K.E., Hermansen, A., Guggenmos, T., Andresen, P., Minh, M.H., Neste, L., Holmes, M., Pontén, A., DeHolton, K.L. and Eller, P. (2023). GraphNeT: Graph neural networks for neutrino telescope event reconstruction. *Journal of Open Source Software*, 8(85), p.4971. doi: https://doi.org/10.21105/joss.04971

Terven, J., Cordova-Esparza, D., Ramirez-Pedraza, A. and Chavez-Urbiola, E. (2023). *LOSS FUNCTIONS AND METRICS IN DEEP LEARNING. A REVIEW UNDER REVIEW IN COMPUTER SCIENCE REVIEW*. [online] Available at: https://arxiv.org/pdf/2307.02694.pdf.

Vladimir Naoumovitch Vapnik (2010). The nature of statistical learning theory. New York: Springer, Cop.

Watson, A.A. (2011). The discovery of Cherenkov radiation and its use in the detection of extensive air showers. *Nuclear Physics B - Proceedings Supplements*, [online] 212-213, pp.13–19. doi:https://doi.org/10.1016/j.nuclphysbps.2011.03.003

Zhang, Y., Lakitha O. H. Wijeratne, Talebi, S. and Lary, D.J. (2021). Machine Learning for Light Sensor Calibration. 21(18), pp.6259–6259. doi:https://doi.org/10.3390/s21186259

# **Addendum A – LSTM Model**

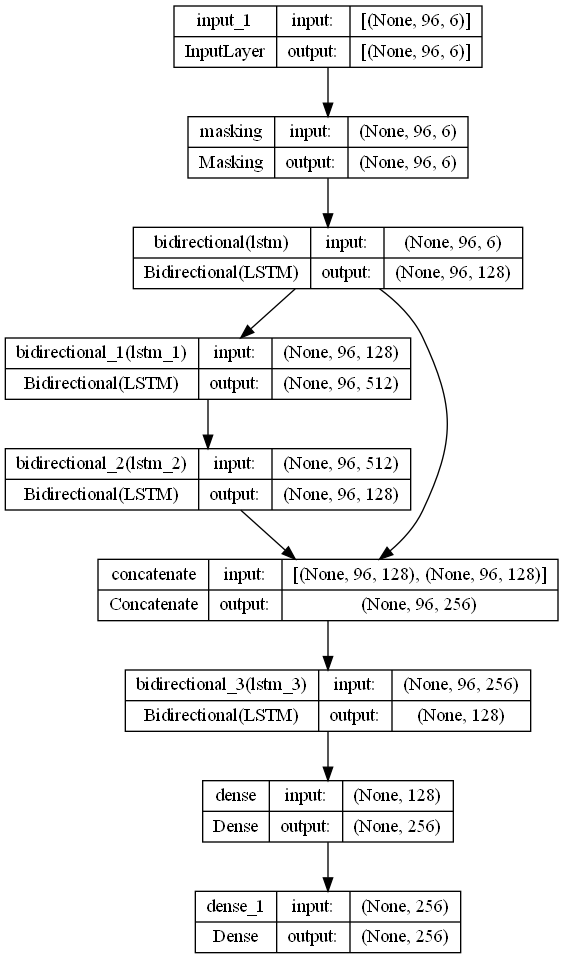


Figure 7 - LSTM Model Architecture

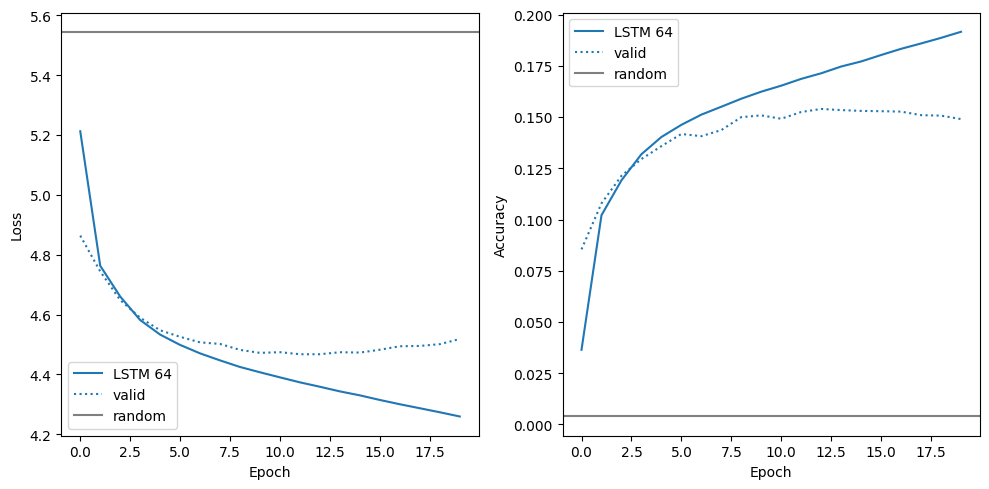


Figure 8 - LSTM Training Performance over 20 Epochs

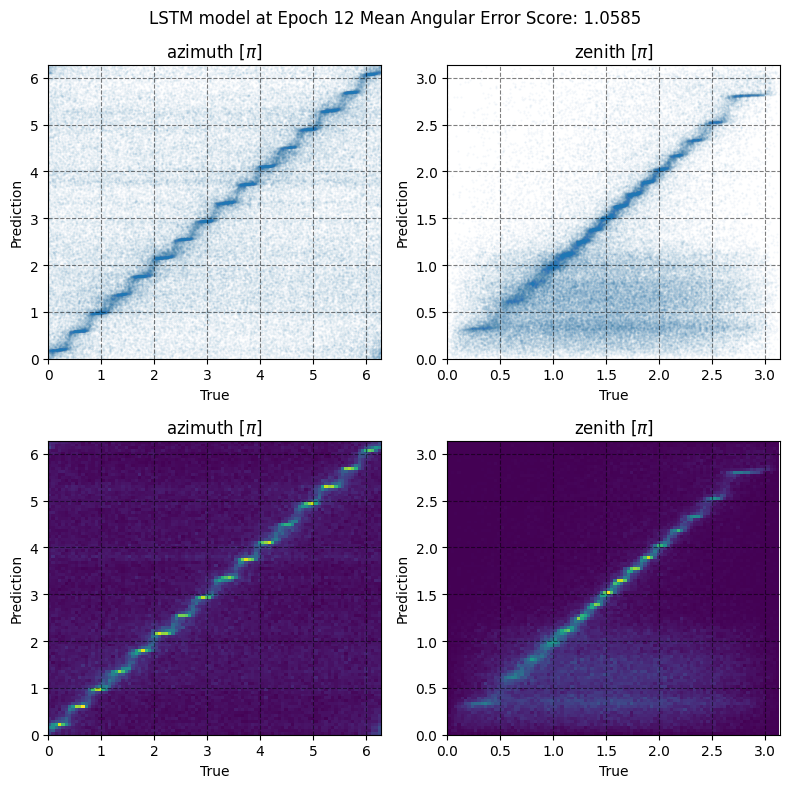


Figure 9 - LSTM Representation of Predicted Values vs Actual Values for Azimuth and Zenith Angles

# **Addendum B – GRU Model**

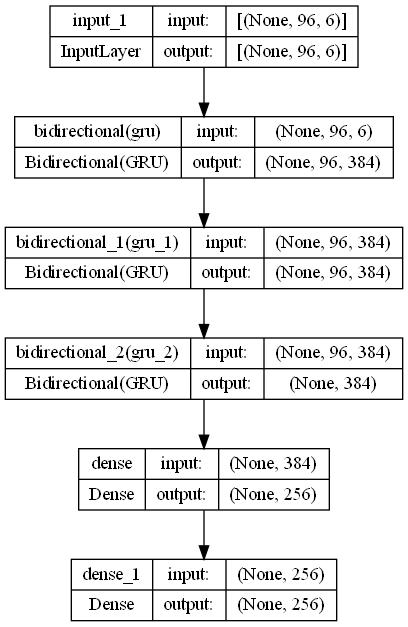


Figure 10 - GRU Model Architecture

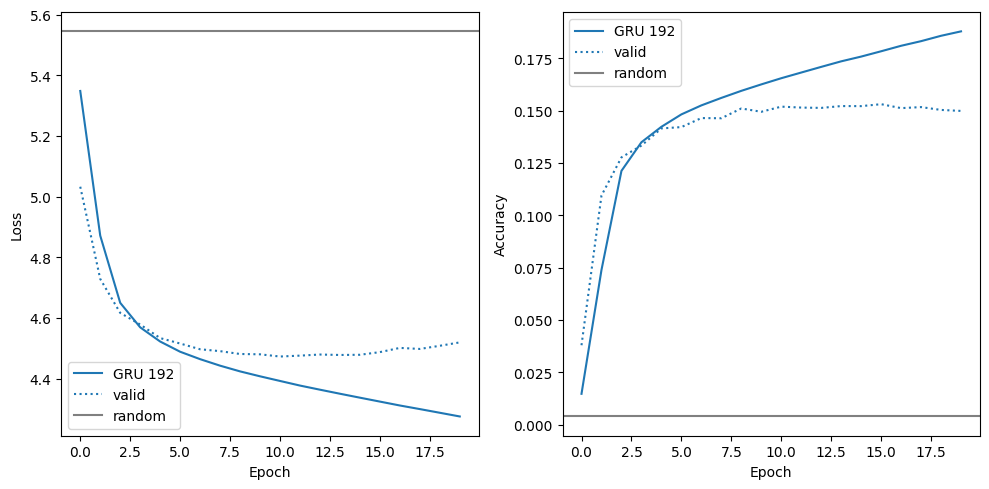


Figure 11 - GRU Training Performance over 20 Epochs

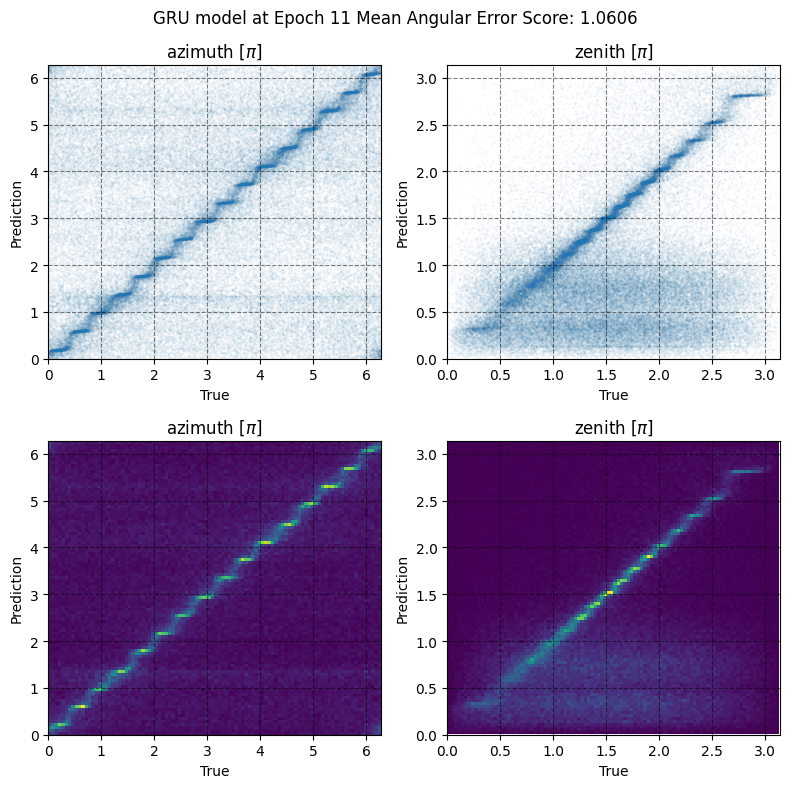


Figure 12 - GRU Representation of Predicted Values vs Actual Values for Azimuth and Zenith Angles

# **Addendum C – LSTM + GRU Model**

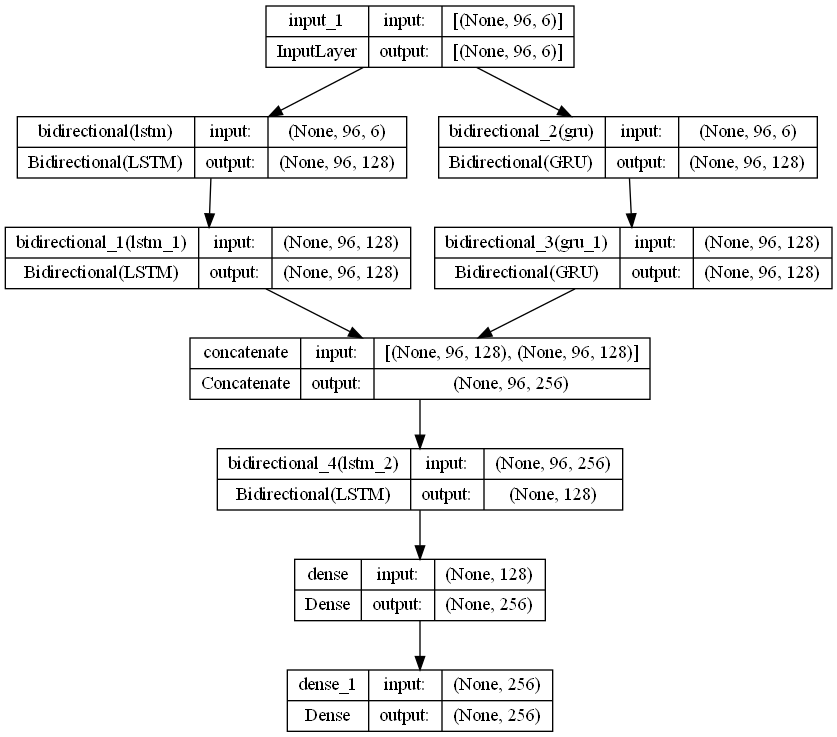


Figure 13 - Hybrid Model Architecture

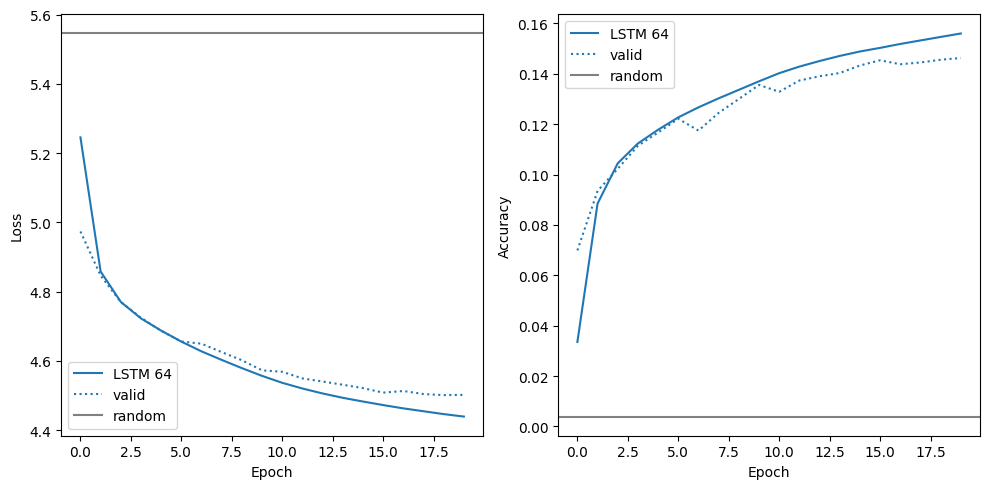


Figure 14 - Hybrid Model Training Performance

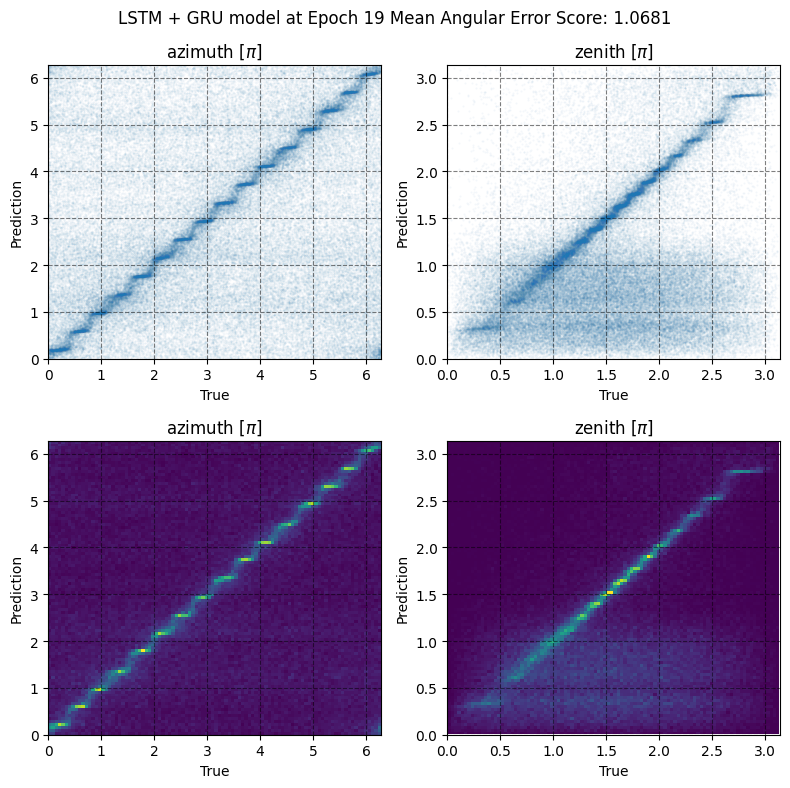


Figure 15 - Hybrid Representation of Predicted Values vs Actual Values for Azimuth and Zenith Angles