UNIVERSITY OF CALIFORNIA,
IRVINE

Improving the Sensitivity and Data Analysis Techniques of the ARIANNA Detector with Deep Learning

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Physics

by

Astrid Lund Anker

Dissertation Committee:
Professor Steven Barwick, Chair
Professor Pierre Baldi
Professor Mu-Chun Chen

2023

DEDICATION

To my parents, who passed down their love of learning to their inquisitive daughter.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Page

ACKNOWLEDGMENTS

I would like to begin by recognizing that this thesis would not be possible without the many scientists whose contributions have been the foundation on which this work was built.

One such individual that is a driving force of ARIANNA as well as my advisor and mentor is Steve Barwick. Thank you for your open mindedness and enthusiasm for bringing deep learning techniques into ARIANNA. Your patience in answering my numerous questions and ability to talk through research roadblocks were invaluable to my growth as a researcher.

I would also like to overall give thanks to Pierre Baldi, Chris Persichilli, Geoffrey Gaswint, and the other members of the ARIANNA Collaboration. In particular, Christian Glaser for his guidance, especially in the early stages of this work. Also Manuel Paul for his experimental and hardware expertise as well as his comradery.

Further, I would like to give a special thanks to my parents for their endless encouragement, support, and guidance, and for raising me to be curious about the world around me. Also thank you to my siblings for always being there for me during every phase of my life.

I would also like to thank the network of friends I have met within and outside of UCI; the day-to-day struggles of getting a Ph.D. are lessened by the support of the communities we have built.

Lastly, I want to give an extra special thanks to Jose for his unending support both during the challenging and rewarding moments in my life. This work would not have been as enjoyable without you by my side through it all.

Thank you to the Machine Learning and Physical Sciences (MAPS) program at UC Irvine for research, collaboration, and funding opportunities through your fellowship. This work received support by the National Science Foundation grant NRT 1633631.

Chapter 4 of this dissertation is a reprint of the material as it appears in [33] with the exception of sections 4.4, 4.5, 4.7, and 4.8. This work is reported here with the permission of IOP Publishing Ltd, and refer to the published work for a complete author list.

Vita

Astrid Lund Anker

EDUCATION

Doctor of Philosophy in Physics **2023**
University of California, Irvine *Irvine, CA*

Bachelor of Science in Physics **2017**
University of California, Santa Cruz *Santa Cruz, CA*

RESEARCH EXPERIENCE

Graduate Research Assistant **2018–2023**
University of California, Irvine *Irvine, California*

TEACHING EXPERIENCE

Teaching Assistant **2017–2022**
University of California, Irvine *Irvine, California*

SELECTED AWARDS

MAPS Fellow **2019**
UC Irvine Machine learning And the Physical Sciences

Honorable Mention Fellowship **2022**
President's Dissertation

SOFTWARE

ARIANNA Analysis `https://github.com/a-anker/ARIANNA_Experiment`

ABSTRACT OF THE DISSERTATION

Improving the Sensitivity and Data Analysis Techniques of the ARIANNA Detector with Deep Learning

By

Astrid Lund Anker

Doctor of Philosophy in Physics

University of California, Irvine, 2023

Professor Steven Barwick, Chair

The ARIANNA experiment is an Askaryan detector designed to record radio signals induced by neutrino interactions in the Antarctic ice. Because of the low neutrino flux at high energies ($E_\nu > 10^{17}\,\mathrm{eV}$), the ability to increase detector sensitivity and data analysis techniques is crucial to maximizing the number of neutrinos measured. In this work, deep learning techniques are explored to improve real-time data collection capabilities and offline neutrino searches. As an introduction, the broader field of multi-messenger astronomy is outlined, an overview of the ARIANNA experiment is provided, and deep learning techniques are detailed. Next, two projects utilizing deep learning to analyze ARIANNA data are presented. In the first project, the amplitudes of the trigger threshold are limited by the rate of triggering on unavoidable thermal noise fluctuations. Here, a real-time thermal noise rejection algorithm is created that enables the trigger thresholds to be lowered, increasing the sensitivity to neutrinos by up to a factor of two (depending on energy) compared to the current ARIANNA capabilities. A deep learning discriminator, based on a Convolutional Neural Network (CNN), is implemented to identify and remove thermal events in real time. This project demonstrated a CNN trained on Monte Carlo data can run on the current ARIANNA microcomputer; the CNN retained 95% of the neutrino signal at a thermal noise rejection factor of 10^5, compared to a template matching procedure which reached only 10^2

for the same signal efficiency. The results are verified by feeding in generated neutrino-like signal pulses and thermal noise directly into the ARIANNA data acquisition system. There are further studies of the CNN including deep learning network interpretability and hyper-parameter optimization. Lastly, the CNN is used to classify cosmic rays events to confirm they are not rejected; the network properly classified 102 out of 104 cosmic ray events as signal.

In the second project, deep learning is used in an offline analysis to classify experimental ARIANNA data collected between 2018-2021. This work compares a more traditional neutrino search technique using cuts on different variables to a new method using deep learning to classify experimental data in an offline analysis. In the second-to-last stage of data cuts, the traditional analysis is found to keep 99% neutrino signal efficiency while rejecting all except 53 experimental background events; the deep learning approach provided significantly better results with 99% signal efficiency while rejecting all except two experimental background events. Both groups of remaining background events were rejected in the final correlation cut stage of the analysis. Due to a limitation in simulating all background event types, the deep learning network was trained on a mixture of simulated and experimental data. A further study was done to check for potential artifacts between the two different types of data that could lead to inaccurate classification results. The study was conducted using the data from a cosmic ray configured ARIANNA station which contained experimentally detected cosmic ray. It is shown through a similar deep learning analysis on the cosmic ray station that there were no artifacts seen in the final model. This provides confirming evidence that artifacts are not affecting the efficiency and background rejection results of the neutrino analysis. This work concludes with a summary of the work done and final recommendations moving forward with deep learning techniques for the ARIANNA experiment.

Chapter 1

Introduction

1.1 Multi-messenger Astronomy

Until recently, the main tool scientists used to learn about the Universe was visible light, or photons. The method of observing light as it moves across the sky led to the discovery of the planets and moons in the solar system and Kepler's Laws of Motion to name a few. With advancements in technology, more sophisticated tools were developed to measure other messengers such as gravitational waves, cosmic rays, and neutrinos [102]. Multi-messenger astronomy, as the name suggests, relies on the observation of many messengers and ideally different types in coincidence to be able to learn more about distant astronomical objects. These additional messengers are needed to extend the observable region of space and see further back in time to earlier points in the Universe.

There are two primary constraints on using photons as messengers to observe very distant regions of the Universe. On the lower energy end, electromagnetic radiation (light) has a cutoff point called the epoch of recombination, and cannot be observed earlier than approximately 380,000 years after the Big Bang [103]. Prior to this point, photons were continuously scat-

tered within an extremely hot plasma and could not escape; after this point, as the Universe expanded and density and temperature decreased, photons became decoupled from matter and were able to travel outward. Photons from this time are the oldest measurable and are known as the Cosmic Microwave Background (CMB) [60].

On the higher energy side, photons are absorbed over cosmic distances due mainly to photon-photon collisions. Typically high energy photons ($\mathcal{O}(10^{12})$) scatter with infrared photons ($\mathcal{O}(10^{-3})$). These collisions produce an electron-positron pair above a center of mass (COM) energy of $E_{COM} > 2m_e c^2 \approx 1.02\,\text{MeV}$ [106], the rest mass of the electron and positron combined. As the COM photon energy becomes much larger than the threshold, pair production becomes the dominant process. The mean free path of the photon drives the pair production, and it is described as the average distance a particle travels before colliding with another particle. For high energy photons, their mean free path is on the order of galactic distances, so extra-galactic photons would not reach earth. Therefore, multi-messenger astronomy allows for the exploration of more distant regions of space that would otherwise be inaccessible if relying on photons alone.

1.1.1 Gravitational Waves

Gravitational waves were predicted by Albert Einstein along with the theory of General Relativity. He theorized that these waves would be generated by accelerating masses that perturb spacetime. Gravitational waves were first observed in a binary pulsar system when observations of the orbital period showed a loss in energy through gravitational radiation [77, 107]. Then in 2015, the Laser Interferometer Gravitational-Wave Observatory (LIGO) experimentally measured the first gravitational wave [18], broadening the opportunity for new messenger particles to be studied.

One benefit of gravitational waves is they travel in a straight line from their source; as a

result, when such waves are measured on Earth, the direction of the signal can be used to pinpoint their source's location. However, one problem with measuring gravitational waves is their signal scale range is broad compared to the size of Earth. The wavelength of gravitational waves is macroscopic, so only the most compact source objects will be detectable with ground-based detectors. Larger scale sources such as supermassive black holes would require a detector length similar to the distance between the Earth and the Sun, which is only possible with a space-based detector. Thus, currently most detectors are Earth-based and only study the most compact objects; stellar mass black holes and neutron stars, at signal wavelengths of $\lambda \sim 1\,\mathrm{km}$. This messenger opens a new and exciting branch of study, though cost and logistics of a space-based detector limit its range of discovery.

1.1.2 Cosmic Rays

Cosmic rays include a diverse group of charged particles traveling through space. They can be electrons, positrons, protons, or atomic nuclei, with the latter two being the most abundant. Cosmic rays were first discovered by Victor Hess in 1912 with a pioneering balloon experiment [52]; their place in multi-messenger astronomy is well established. Their energy spans a large range from 10^6 eV - 10^{20} eV, but their composition, flux, and origin are only well defined below 10^{14} eV. At higher energies, the cosmic ray flux is too low to measure with direct detection so instead it is measured through air showers: cosmic rays collide with the atmosphere and produce cascades of secondary particles. As established through many air shower measurements, limits on the flux for a given cosmic ray energy range is shown in Fig. 1.1.

There are four regions of interest in this distribution and while the sources of these features are not yet confirmed, a discussion of their potential causes follows. The first is the knee around 10^6 GeV, which is theorized to be the cutoff of the acceleration energy of dominant

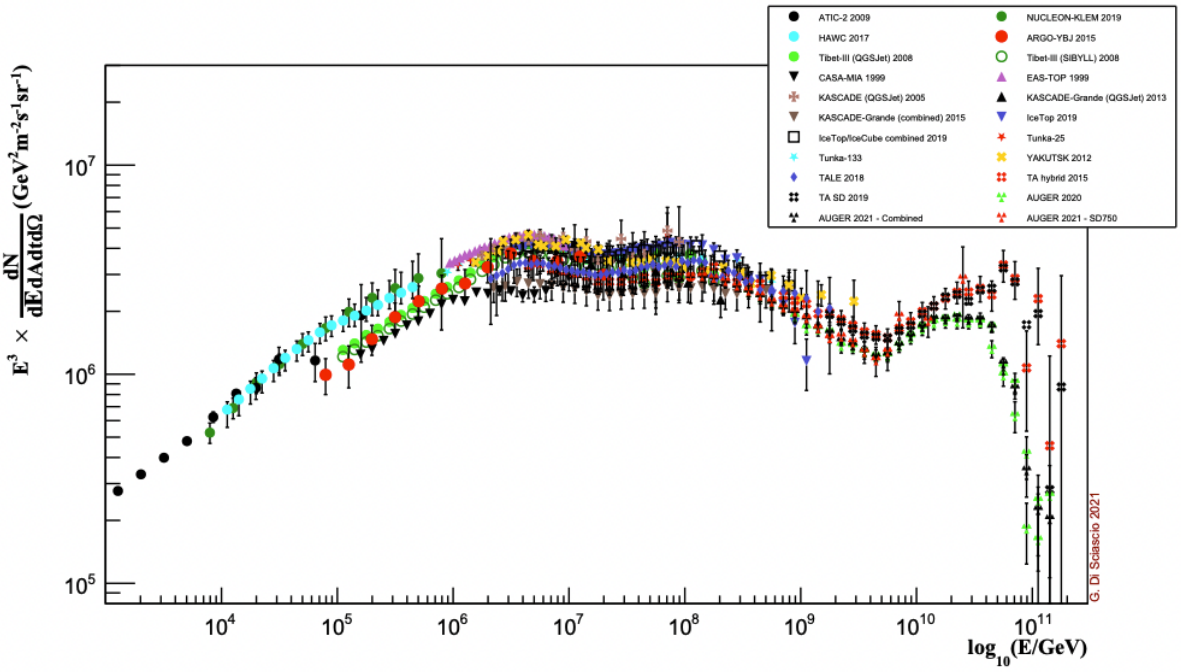

Figure 1.1: The cosmic ray energy spectrum (flux multiplied by E^3) compiled from various direct and indirect experiments. From [58].

galactic sources specifically for light nuclei such as protons. The second knee around 10^8 GeV is again thought to be the acceleration energy cutoff but this time for heavy nuclei such as iron. The ankle starting around 10^9 GeV is argued to be the final cutoff of galactic cosmic rays and beyond this point, they are extra-galactic. Lastly, a cutoff below 10^{11} eV is observed with the cause still unknown [102]; a possible explanation for this cutoff is the Greisen-Zatsepin-Kuzmin (GZK) effect, which describes a particle collision process for reducing the energy of cosmic ray protons. Otherwise, it could mark the maximum energy of cosmic ray accelerators. More accurate measurements of the composition are required to distinguish which of the two different scenarios is the dominant reason for the cutoff.

If the composition is mainly protons, the GZK effect postulates, above a threshold of around $10^{19.5}$ eV, extremely-high energy (EHE, defined here as $E_\nu > 10^{17}\,\text{eV}$) cosmic rays interact with the CMB through a Δ resonance [71, 113]. Similar to photons, this means above a certain energy the Universe will be opaque to cosmic rays. Moreover, a CMB photon in the

rest frame of the proton (cosmic ray) is a high energy photon of order hundreds of MeV, which is equivalent to the energy of a pion. The proton absorbs the photon and has the right energy to excite the proton into a new particle called a Δ resonance (resonance because the cross-section goes way up). This particle has a slightly higher mass than the proton but a much shorter lifetime. The Δ acts differently when absorbing photons. Unlike an atom that absorbs and re-emits the photon, it has another possible decay option. It's a strongly decaying particle that reverts back into a proton but with less energy than before the collision and a pion, see Equation 1.1. Eventually this proton loses enough energy that it is no longer able to collide with CMB photons; this may be the cause of the cosmic ray flux cutoff at extremely-high energies.

$$p + \gamma_{CMB} \rightarrow \Delta^{+} \rightarrow \pi^{+} + n \rightarrow p + \nu_{\mu} + \bar{\nu_{\mu}} + \nu_{e} \tag{1.1}$$

Additionally, the Δ has another decay mode as shown in Equation 1.2. This process not only provides a mechanism for producing EHE neutrinos, but also gamma rays.

$$p + \gamma_{CMB} \rightarrow \Delta^{+} \rightarrow \pi^{0} + p \rightarrow p + 2\gamma \tag{1.2}$$

The types of sources theorized to create these EHE cosmic rays (in excess of 10^{20} eV) are shown in Fig. 1.2. These sources each have their own maximum particle acceleration capabilities, and this is determined by their size (R), their magnetic field (B), and the particle's charge. In particular, the magnetic field and the length over which the particle is accelerated contribute to the final energy of the charged particles as it leaves the source [85].

The exact source (or sources) of EHE cosmic rays is still unknown mainly because cosmic rays cannot be traced back to their creator. They are charged particles that get bent off course by galactic and extra-galactic magnetic fields. While cosmic rays expand the energy range of observable particles in multi-messenger astronomy, past that of the photon, they

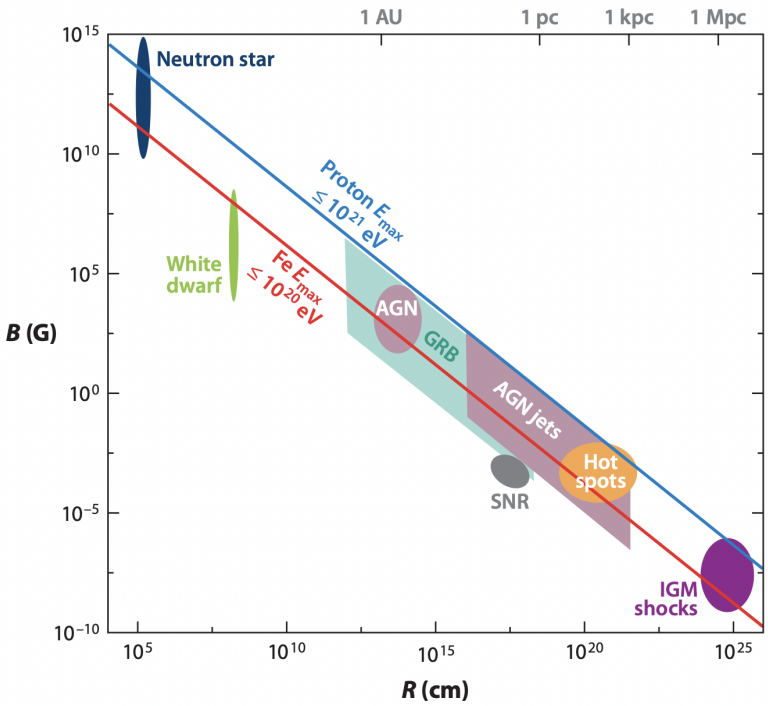

Figure 1.2: The Hillas diagram for the potential sources of EHE cosmic rays. Sources above the slanted blue line can accelerate protons up to 10^{21} eV, and above the slanted red line, can accelerate iron up to 10^{20} eV. From [85].

alone cannot be used to determine the sources of EHE cosmic rays. It will be necessary to measure a EHE cosmic ray in coincidence with another messenger particle such as a gamma ray or neutrino produced near the source (via the GZK effect) to pinpoint the origin of EHE cosmic rays.

1.1.3 Neutrinos

Neutrinos are Standard Model particles that were theoretically proposed in 1930 by Wolfgang Pauli and experimentally measured in 1956 by a team lead by Reines and Cowan [38]. They are neutral in charge, come in three flavors (electron, muon, and tau), and oscillate between these three generations. However, there is still a lot unknown about neutrinos, such as their exact masses [78]. They are very light particles that travel close to the speed of

light and their interaction cross-section is extremely low. Compared to the other messenger particles, neutrinos are neutral and rarely interact. Neutrinos are ideal messengers because they can travel extra-galactic distances without being bent off course by galactic or extra-galactic magnetic fields. In addition they rarely interact with matter in space and can travel unimpeded. With these two properties, neutrinos are promising messengers for probing the EHE regimes of space as they will propagated directly from the source that created them to Earth; plus their trajectory can be reconstructed to determine where in the sky they came from and this process is outlined in Sec. 2.3.3. Furthermore, neutrinos are likely created by the same sources that produce EHE cosmic rays or near the sources via the GZK mechanism as shown in Equation 1.1.

A disadvantage of neutrinos is the very thing that makes them ideal messengers: they do not interact very often. Most of the neutrinos that humans contend with daily are lower energy neutrino that come from the Sun and pass straight through an individual at a rate of trillions per second. Another common source of neutrinos, especially for those living in close proximity, are neutrinos from nuclear reactors. This work is focused on the neutrino energy ranges above those produced by the sun and nuclear reactors. The expected flux of such higher energy neutrinos is shown in Fig. 1.3 along with the flux of gamma rays and EHE cosmic rays. Neutrino flux drops sharply with increased energy. There are benefits to utilizing natural astrophysical sources of neutrinos instead of particle colliders; the highest energy achieved at the Large Hadron Collider is 13 TeV, which produces neutrinos at a maximum energy of approximately 1 TeV [9] whereas astrophysical sources can produce neutrinos up to around 10^{19} eV.

The highest energy cosmogenic neutrino measured so far is on the order of 10^{15} eV, measured from the IceCube experiment [4]. IceCube is one of the most successful neutrino telescopes to this point and has observed a flux of $10^{12} - 10^{15}$ eV astrophysical neutrinos. This experiment has also measured a particle shower at the Glashow resonance, which will provide a method

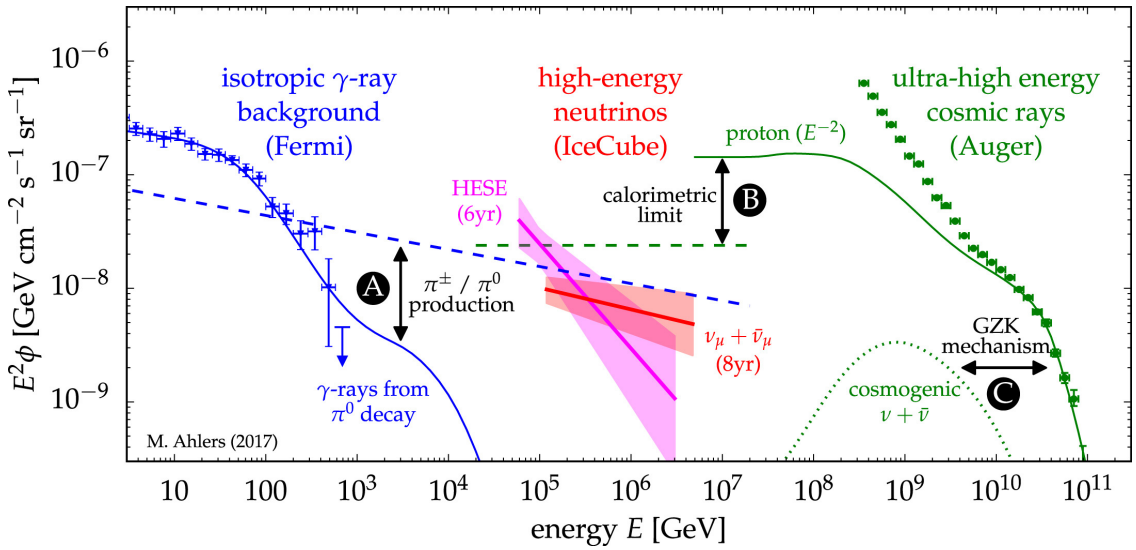

Figure 1.3: The spectral flux of gamma rays (blue), neutrinos (red), and EHE cosmic rays (green) from various experiments and theoretical distributions. The lettered regions mark the interface between messengers. In region A there is joint production of charged and neutral pions from cosmic ray interactions, which leads to the emission of neutrinos and gamma rays. Region B marks the upper limit of high energy neutrino flux (dashed green) due to the Waxman-Bahcall limit [112]; these neutrinos are from the same sources as the high energy cosmic rays (solid green). Region C gives the boundary between EHE cosmic rays and cosmogenic neutrinos; EHE cosmic rays are predicted to emit cosmogenic neutrinos via the GZK mechanism. From [24].

of distinguishing neutrinos from antineutrinos and validation to the Standard Model [12]. Before IceCube were such pioneering neutrino telescopes as AMANDA and others outlined in Sec. 1.3. These neutrino experiments along with the experimental confirmation of the shower mechanism for neutrinos in dense media provided further motivation for neutrino telescope experiments. In particular, radio-based neutrino detectors complement IceCube by covering the highest expected energy scale of cosmogenic neutrino (10^{16} - 10^{19} eV). Thus, there are many experiments using radio techniques to cover the highest energy neutrino regimes and some of these experiments are discussed in Sec. 1.3. To measure these elusive neutrinos, highly specialized detectors that enclose large volumes of matter are needed. If the composition of EHE cosmic rays is proton dominated, the flux of cosmogenic neutrinos produced by the EHE cosmic rays is expected to be one event per year with a detector target volume of 100 km^3 in dense media such as ice [23].

1.2 Radio Emission from Particle Induced Showers

There are two particle shower profiles studied in this section: particle showers in air due to EHE cosmic rays and particle cascades in ice due to neutrinos. The specific dominant physics processes change between different media, so their differences are outlined below.

1.2.1 Particle Showers in Ice

Neutrino induced particle showers are studied in dense media such as ice, salt, and lunar regolith. For the purpose of this section, ice will be the focus. A diagram of the simplified process of particle showers is shown in Fig. 1.4. The neutrino is the primary particle that collides with ice nuclei via either a mediating W^+/W^- (charged current) or a Z boson (neutral current). In either case, these interactions produce a neural pion which initiates an electromagnetic cascade of particles through electron-positron pair production, the inverse Compton effect, and bremsstrahlung. Specifically, many of the positrons annihilate with electrons in the surrounding ice nuclei and many electrons get kicked out of their atoms by ionized photons. This creates a net negative charge excess at the shower front of about 30% of the electrons and positrons [101]. The total charge in the shower is conserved as the charge excess comes from the electrons in the surrounding nuclei and the shower leaves behind a positively charged plasma. The charge build up is fueled by a chain reaction of the secondary particles colliding with other particles and so on. At a certain point, the shower runs out of energy as particles are now absorbed by the ice nuclei and no new particles are being kicked out.

This entire process of creating radio signals from a time dependent charge excess is known as the Askaryan effect [37] and is illustrated in Fig. 1.5 on the right side. Because of the negative charge excess in the middle of the shower, the Askaryan effect produces a radial polarization

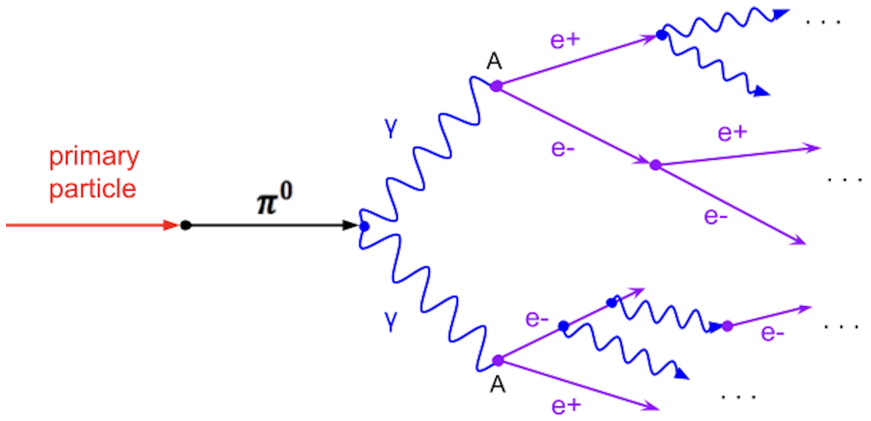

Figure 1.4: A simplified diagram of a particle shower in air or a dense medium.

pattern. Askaryan emission has been experimentally verified in laboratory measurements in ice [70] and other media [98, 69]. The radio emission produced by the Askaryan effect is only observed close to the Cherenkov radiation cone, with an angle of about 56° in deep ice [46]. Cherenkov radiation occurs when ultra-relativistic charged particles travel faster than the speed of light in the given medium. With the enormous amount of energy in these particle showers, Cherenkov radio pulses are created. These extended meter-long pulses add up coherently along the Cherenkov angle and produce a strong bi-polar signal [26]. The frequency of this signal will vary depending on how far from the Cherenkov cone it is measured; in ice, the frequency ranges from about tens of MHz to a couple GHz. The Askaryan emission mechanism is dominant in dense media due to the short longitudinal shower length, on the order of meters.

1.2.2 Particle Showers in Air

Air showers are created when a cosmic ray collides with an atmospheric particle, initiating a cascade of secondary particles much like the process described above for neutrinos in ice. The difference from the process described in Fig. 1.4 is that in addition to a neutral π^0, there are charged pions created that decay to produce leptons such as muons and neutrinos.

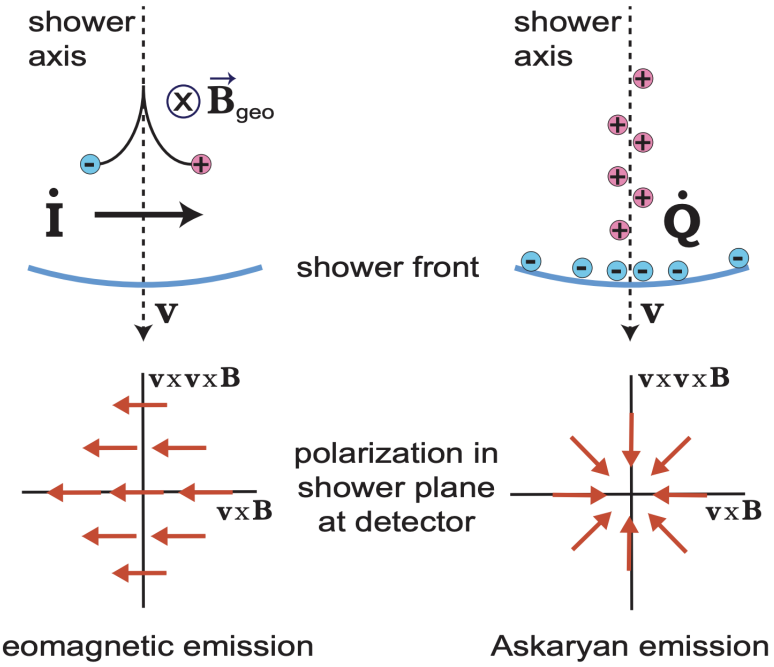

Figure 1.5: The two main emission mechanisms for coherent radio emission of particle showers (in the frequency range of MHz to GHz). Geomagnetic emission occurs when charged particles move through Earth's magnetic field, producing time dependent transverse currents. Askaryan emission results from a time-varying net charge build up in the shower front due to pair production and Compton scattering of the shower particles. From [102].

The electromagnetic showers created in air differ from those in ice as they are longer in longitudinal length. The Askaryan effect is also present in air showers dependent on the shower's properties [84, 99], but it is far less than the geomagnetic effect. Geomagnetic emission occurs when charged particles are deflected by Earth's magnetic field [79]. The Lorentz force causes electrons and positrons to split into opposite directions, which induces a transverse drift current that varies in time as the total number of particles in the shower changes. This process is shown in Fig. 1.5 on the left side, and the polarization pattern points to the left due to the splitting of the charges. Because air showers occur over kilometer extents, they have a longer opportunity to be influenced by the Earth's magnetic field; this is why geomagnetic emission is the dominant emission mechanism in air. Lastly, compared to showers in dense media, air showers produce radio pulses in a lower frequency range, typically below 100 MHz. Geomagnetic emission has also been verified [35] by strong positive correlation between simulation and experimental data.

1.3 Experiments in Neutrino Astronomy

There are many different experiments trying to determine the source or sources of EHE cosmic rays, generally grouped into two categories: Cherenkov light detectors and radio-based detectors. Among those studying the Cherenkov signal, first is the Pierre Auger Observatory which consists of an array of water tank Cherenkov detectors aiming to measure EHE cosmic rays [5]. Second is the Baikal-GVD experiment, which is a net of acoustic antennas located in Lake Baikal looking for high energy neutrinos [73]. Third is the IceCube experiment which consists of an array of photomultiplier tubes buried in the ice at the South Pole, searching for high energy neutrinos [10] via their Cherenkov light emission. IceCube is the largest neutrino detector built to date.

On the radio-based detector side, pilot arrays that exploited the radio detection of high

energy neutrinos were constructed during the last decade. These arrays include RICE located at the South Pole, which consisted of a 20-channel array of dipole radio receivers dispersed within a 200 m^2 cube of ice at a depth of 100-300 m [86]; the ARA test bed located near the South Pole, which consisted of three stations separated by 2 km each containing a string of vertically and horizontally polarized dipoles [25]; ANITA, a balloon-based radio Cherenkov detector with two initial deployments [90]; and the ARIANNA HRA-3 (Hexagonal Radio Array) initially consisting of three prototype stations located on the Ross Ice Shelf, each with four downward facing LPDAs (log-periodic dipole antennas) buried just below the ice surface [40]. The latter experiment is the focus of this thesis.

These efforts helped focus on the radio techniques required to operate in extremely cold and harsh conditions. While these experiments showed the technical feasibility of this approach, they were too small to measure the low neutrino flux. Undeterred, several radio-based experiments in development are further illustrating the capabilities of this detection method, such as ARIANNA-200 [31], the radio component of IceCube-Gen2 [13], the Radio Neutrino Observatory in Greenland (RNO-G) [20], Giant Radio Array for Neutrino Detection (GRAND) [115], Taiwan Astroparticle Radiowave Observatory for Geo-synchrotron Emissions (TAROGE) [88], and Payload for Ultrahigh Energy Observations (PUEO) [16], a successor to ANITA. There is also a newly proposed experiment called GRAMMAR (GReenland Array for Multi-Messenger Astrophysical Research), which consists of an array of surface radio-based neutrino stations in Greenland. Each of these experiments exploit the behavior of neutrinos in one of various target materials including ice, water, earth (mountains), or air.

Even at the extremely-high energies ($E_\nu > 10^{17}$ eV) relevant to radio neutrino detectors, neutrinos rarely interact with matter. A rough estimate suggests that instrumented volumes must reach of order one teraton (10^{12} m^3) to observe a few neutrinos per year for commonly discussed theoretical models of neutrino production [109]. The current experimental limits on

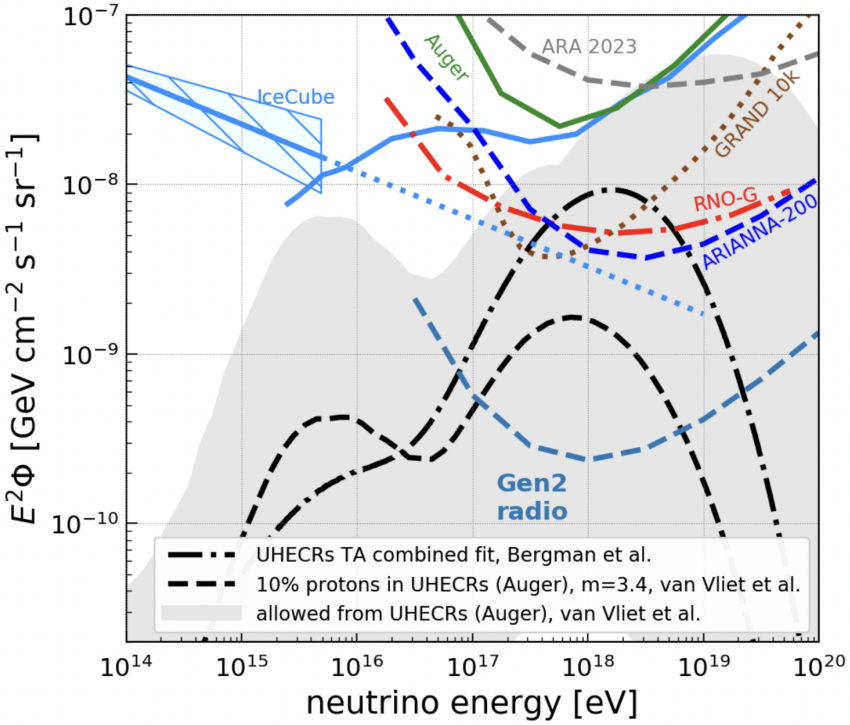

Figure 1.6: The diffuse neutrino flux sensitivity of various experiments at different neutrino energies. These limits are given for current and future extremely-high energy cosmic ray and neutrino experiment. Also included are the GZK neutrino flux predictions from the Pierre Auger Observatory [109] and the Telescope Array [48]. From [43].

neutrino flux are shown in Fig. 1.6 for some of the current and future experiments mentioned above. Also included are three different predicted neutrino flux distributions from the Pierre Auger Observatory [109] and the Telescope Array [48]. Current experiments have only probed a small region of the predicted neutrino flux. Larger detector sensitivities will be needed to probe into the lower flux values.

The challenge for experimenters is to reach the teraton detection volumes required to acquire sufficient signal at a reasonable cost. One of the most promising methods for observing EHE neutrinos is via radio waves in large target volumes of ice [102, 95]. For this reason, locations such as Greenland and Antarctica are popular sites for radio detection experiments. Ice is transparent to radio signals, with field attenuation lengths ranging from 0.5 km at Moore's Bay (Antarctica) [47] to more than a kilometer in the colder ice found at the South Pole [45]

or the Greenland ice sheet [39]. ARIANNA is one of the experiments utilizing ice to look for neutrinos via their radio pulses. The primary goal of ARIANNA has been to demonstrate the robustness of the shallow detector design, prove that it can identify and remove backgrounds while maintaining high efficiency for neutrino selection, and inform the development of the next generation of radio-based neutrino detectors.

In this dissertation, two promising methods are described to improve the sensitivity of the surface design ARIANNA detector. The first study implements deep learning techniques to lower the trigger threshold, while the second study includes a (slightly) more complex network to separate neutrinos from background events more effectively in an offline analysis. Deep learning techniques are used to provide an efficient filter to reduce the rate of uninteresting thermal events by five orders of magnitude, which in turn allows the detector to operate at reduced thresholds. The offline analysis incorporates the information from all antenna channels to improve the efficiency of neutrino identification. The combination of these two studies suggest that large arrays of surface stations can reach the sensitivities require by future detectors such as IceCube-Gen2 [14].

Deep learning techniques for radio neutrino (and cosmic ray) experiments are becoming more widespread [49, 91, 96, 61]. Recently, the work in [68] demonstrated that deep learning networks can reconstruct the energy, direction, and for the first time, neutrino flavor. The work in this thesis adds to the growing body of evidence that deep learning techniques will play a decisive role in future detector designs and analysis techniques within neutrino astronomy.

Chapter 2

The ARIANNA Experiment

There are many neutrino experiments operating in Antarctica, but ARIANNA is one of only three currently using radio-based detection methods. Other experiments such as IceCube use optical Cherenkov light to look for neutrinos, and their ability to measure neutrinos ends above energies of around 10^{16} eV. Radio-based detectors look for the highest energy neutrinos in the range of $10^{16} - 10^{20}$ eV. The other two operational radio-based detectors are ARA, which deploys antenna detectors far below the ice surface up to 200 m and ANITA, which is a balloon-based detector that reaches elevations of 35 km to 40 km. ARIANNA sits between the two with detectors buried right below the surface of the Antarctic ice. This provides an advantage in that the detector can be installed fairly quickly; it is cost effective and can remain in the same location without maintenance. Plus, the detectors can be (and some have been) removed in a short amount of time, so the environmental impact of this experiment is lessened. Antarctica also offers a radio quiet environment with little human activity compared to most other locations on Earth.

The ARIANNA experiment (Antarctic Ross Ice-shelf ANtenna Neutrino Array) [46] is an array of autonomous radio stations located in Antarctica. For radio-based detectors to meet

the large target volume required for neutrino detection, it is more economically feasible to build an array of stations rather than a single extremely large one. ARIANNA utilizes the long attenuation lengths for radio propagation to space out the stations 1 km apart. Stations operated until 2019 at sea-level on the Ross Ice Shelf in Moore's Bay, about 110 km from McMurdo Station, the largest research base on the continent. In addition, up until 2021, two stations operated at the South Pole where it is colder and at a higher elevation compared to Moore's Bay. Only a single station is needed to identify and reconstruct neutrino signals, so stations can be deployed more flexibly in various configurations and locations. These different locations produce similar data as the main hardware components do not vary among stations.

In terms of signal propagation, the structure of the ice between the two locations is slightly different. The temperature of the ice is typically colder and therefore more dense at the South Pole [57]. In addition, Moore's Bay is located on the Ross Ice Shelf, which allows for more sky coverage and more potential neutrino signal to be measured [44]. The boundary between water and ice creates a mirror-like reflector for radio signals; thus, downward traveling neutrinos (relative to the ice surface) can be reflected back into the surface detector whereas in locations such as the South Pole, these signal pulses would not be measured. At the South Pole, the detectors are limited to viewing nearly horizontal traveling neutrinos. The instantaneous sky coverage between the two locations and a proposed future location (Greenland) is shown in Fig. 2.1.

As of the year 2022, ARIANNA has nine inactive detector stations in Moore's Bay and zero detector stations at the South Pole. The stations residing at Moore's Bay became dormant in 2019 because the solar panels were buried over time. Since then, the ARIANNA group has not been allowed to travel to Moore's Bay to uncover the panels. The remaining two South Pole stations were recently recovered in 2021 and the data from one of these stations will be analyzed in Chapter 5. The pilot stage of ARIANNA has ended and the collaboration is working on follow-up proposals for a large scale detector. In the next section, all of the

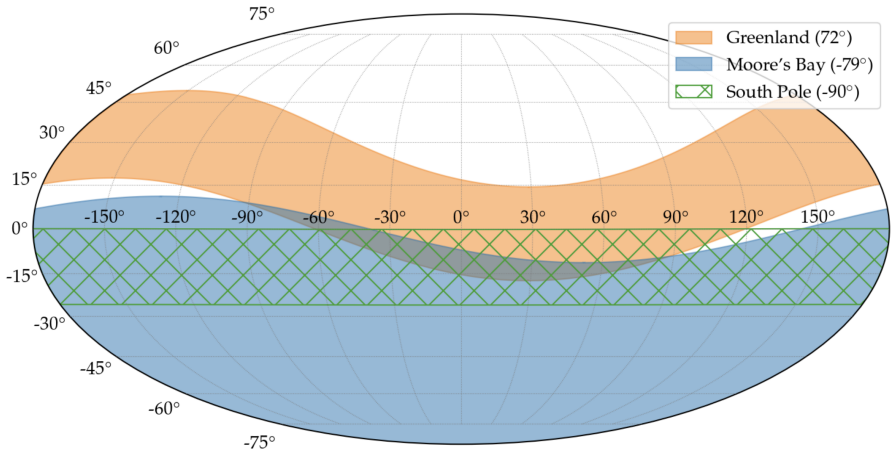

Figure 2.1: The instantaneous sky coverage of ARIANNA at Greenland (orange), Moore's Bay (blue), and the South Pole (green hatch). The units are in Right Ascension and Declination. From [31].

components of the ARIANNA detector will be described as well as past and current projects, and future research proposed for this experiment.

2.1 The ARIANNA Detector Overview

The ARIANNA experiment comprises an array of autonomously functioning stations which each contain data acquisition electronics (DAQ), antennas, a power system, and communications infrastructure. Each component has the requirement of minimizing the power consumption as ARIANNA runs on about 5 watts of solar power. The autonomy of the stations allows the array to be scaled up since no external power infrastructure is needed. Each station also has its own communications system, which can transmit some of the data back to servers at UC Irvine. The flexibility of station design has allowed for many different station configurations to be explored; however, the typical configuration is a 4 or 8 channel station with each channel corresponding to an antenna detector. A standard neutrino station contains four downward facing log periodic dipole array (LPDAs) to measure neutrino signal, as shown on the left side of Fig. 2.2. The two co-polarized pairs are 6 m apart and

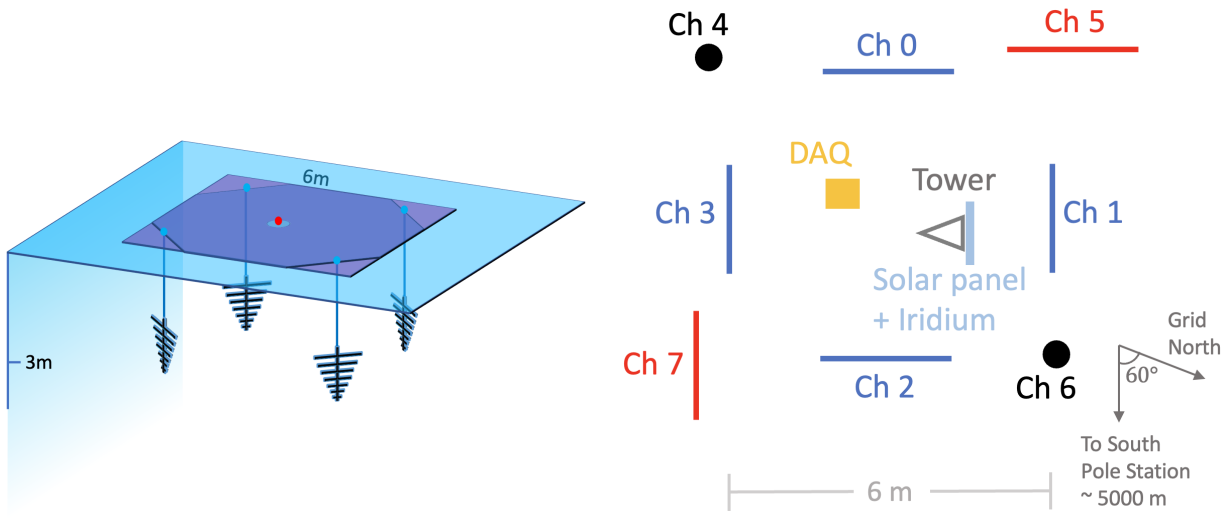

Figure 2.2: A perspective diagram of an ARIANNA neutrino station at Moore's Bay consisting of four downward facing LPDAs (left). This four-channel design is typical of the early Hexagonal Radio Array (HRA) stations. A schematic (aerial view) of an augmented ARIANNA station at the South Pole, station 61 (right). The architecture includes eight antenna channels, with the usual four downward facing LPDAs complemented by adding upward facing LPDAs and vertical dipole antennas. For station 61, channels 0-3 are downward facing LPDAs, 5 and 7 are upward facing LPDAs, and 4 and 6 are vertical dipoles; in the middle of the station is the solar panel, the tower, and the ARIANNA electronics. The long lines indicate the orientation of the plane of the LPDA.

are sensitive to neutrino signal traveling upward. Because the antennas are buried close to the snow surface, installation of different antenna orientations is possible. For example, a station becomes sensitive to cosmic ray signal by orienting the four LPDAs upward instead. In addition, a mixture of downward and upward facing LPDAs can be used to measure neutrinos and veto cosmic ray events that saturate all antennas in the station. There are also dipole antennas that, due to their cylindrical shape, can be buried further down in the ice to help with energy reconstruction.

One example of a station layout (for station 61) is shown on the right side of Fig. 2.2. Station 61 is configured to measure neutrinos and is located at the South Pole. It has four downward facing LPDAs, two upward facing LPDAs, and two dipole antennas. In the middle of the station is the electronics box, solar panel, tower, and communications subsystem. The electronics are housed in two separate aluminum brazed chambers within

one box that provides about 60 dB of radio attenuation. The main chamber is 9 x 12 x 9 inches and contains the motherboard, a Lithium Iron Phosphate ($LiFePO_4$) battery, a battery management unit (BMU), and the two available communication radios (Iridium and Afar). Bolted on top of the main chamber is a smaller 9 x 12 x 2 inch secondary chamber. This contains the low power amplifiers for each channel. The DAQ box is wrapped in an insulated blanket and a plastic bag before being buried just below the snow surface. The 100 watt solar panel and communication antennas are mounted on the central aluminum tower.

2.1.1 Antennas

The two types of antenna used for the ARIANNA experiment are Create CLP-5130-2N log periodic dipole arrays (LPDAs) and vertically oriented Kansas University bicone antennas. For the LPDAs, they have an effective bandwidth of 80 MHz to 1000 MHz in the ice on the Ross Ice Shelf. They are more sensitive to signals arriving near the shortest dipole or 'the nose' of the antenna and signals with a polarization in the direction of the symmetry axis of the dipole. Since the general configuration of the LPDAs in an ARIANNA station are pointing with their nose downward, they are most sensitive to upward traveling, horizontally polarized signals within the ice. To capture the two possible orthogonal polarizations, there are two LPDA pairs for each orientation (refer again to the left side of Fig. 2.2). The bicone dipole is most sensitive to signals arriving perpendicular to, and with polarization parallel to, its symmetry axis. This can aid in measuring the third polarization axis perpendicular to the ice surface, which the LPDAs do not cover. Polarization is important for direction and energy reconstruction of neutrino signal [30]. The bicone dipole antennas are typically placed in the corners of the square shape created by the downward facing LPDAs. Depending on the configuration, current stations have anywhere from none to as many as four dipole antennas. Unlike the LPDAs, the dipoles are cylindrical are therefore easier to install deeper

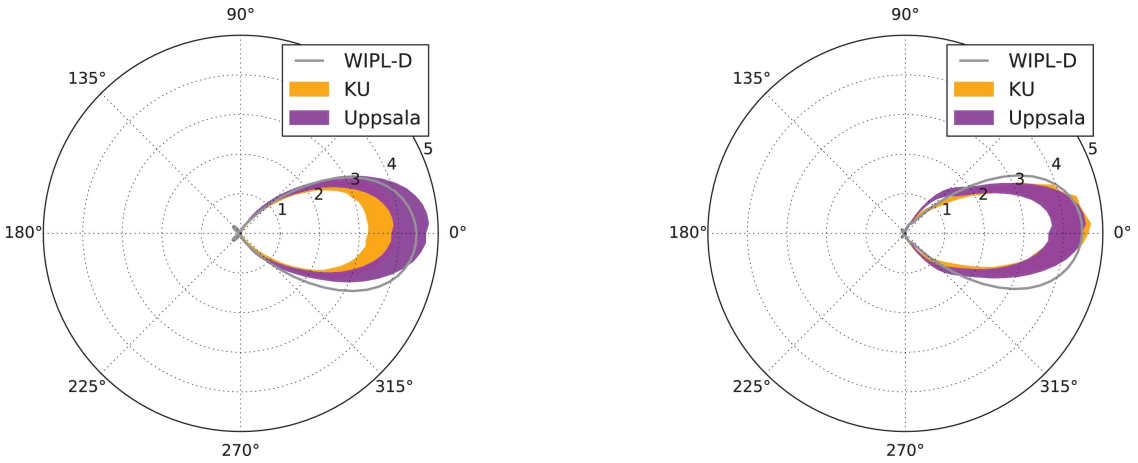

Figure 2.3: LPDA antenna response as a function of direction in the plane of the antenna tines at 330 MHz (left) and 400 MHz (right). The largest gain is at 0° which corresponds to an incoming signal in the direction of the smallest tines. From [42].

in the ice, which aids in energy reconstruction. For both types of buried antenna, their elevation relative to the surface changes; as time goes on, they are buried further down into the Antarctic ice.

The LPDAs have been extensively studied through anechoic chamber measurements [41] and WIPL-D antenna simulations [83, 42]. WIPL-D is powerful simulation software used to model antennas embedded in various media and at the boundary between different media. This software is particularly useful for ARIANNA as the antennas are buried just below the surface and thus show effects from the snow and snow-air boundary. The WIPL-D simulated LPDA antenna response has been compared to two separate anechoic chamber measurements performed at Kansas University and Uppsala University, as shown in Fig. 2.3. All three response curves follow a similar shape and have an average gain agreement within 10%.

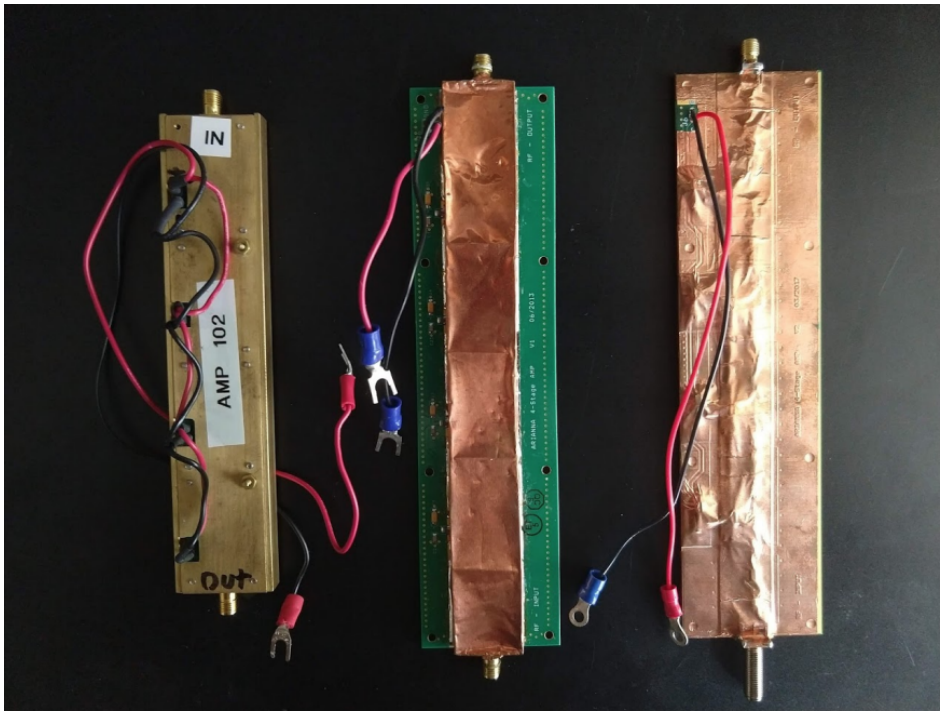

Figure 2.4: Image of the various amplifier generations for the ARIANNA experiment. From left to right is the 100 series, the 200 series, and the 300 series amplifier. From [93].

2.1.2 Amplifiers

Every input channel on the ARIANNA DAQ has its own amplifier board, which was custom designed for ARIANNA by Thorsten Stezelberger at Lawrence Berkeley National Laboratory and fabricated in the engineering lab at UC Irvine. These amplifiers are individually shielded to prevent cross-talk and provide approximately 60 dB of amplification for roughly 0.25 watts of power consumption each. Amplifiers are particularly power consumptive so these were specifically designed to run with minimal power. There are currently three generations of amplifier within the ARIANNA experiment: 100 series, 200 series, and 300 series. All three iterations are shown in Fig. 2.4.

There are a few differences between the amplifiers such as their gain distributions. The 100 and 200 series amplifiers have an effective bandwidth range of about 100 MHz to 1 GHz whereas the 300 series have a more narrow range, as illustrated in Fig. 2.5. All of the

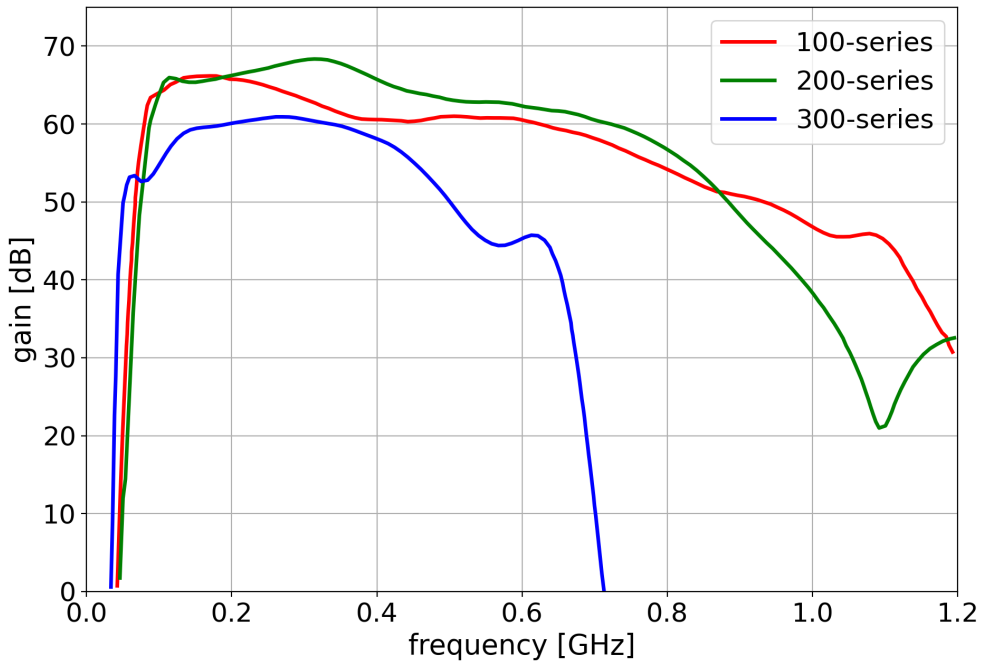

Figure 2.5: Plot of gain versus frequency for the three generations of amplifier. Data taken from [93].

amplifier generations have a maximum output of about 800 mV to protect the electronics. For the 100 series, the filtering and limiting are done outside of the amplifier board, which is composed of a 3 dB attenuator, a 1 V limiter, and another 3 dB attenuator; it also includes a band pass filter of 100 MHz to 1 GHz on the input to reduce low frequency background noise. For the 200 and 300 series, the filtering and limiting are performed by the on-board circuitry. The 300 series amplifiers are designed for an eight channel board where the DAQ system operates at a 1 GHz sampling rate (as compared to the 2 GHz rate for a four channel station).

2.1.3 Communication System

ARIANNA detector's have two possible communication modes to provide redundancy: Afar [19] and Iridium. Both are used in Moore's Bay, but only Iridium is used at the South

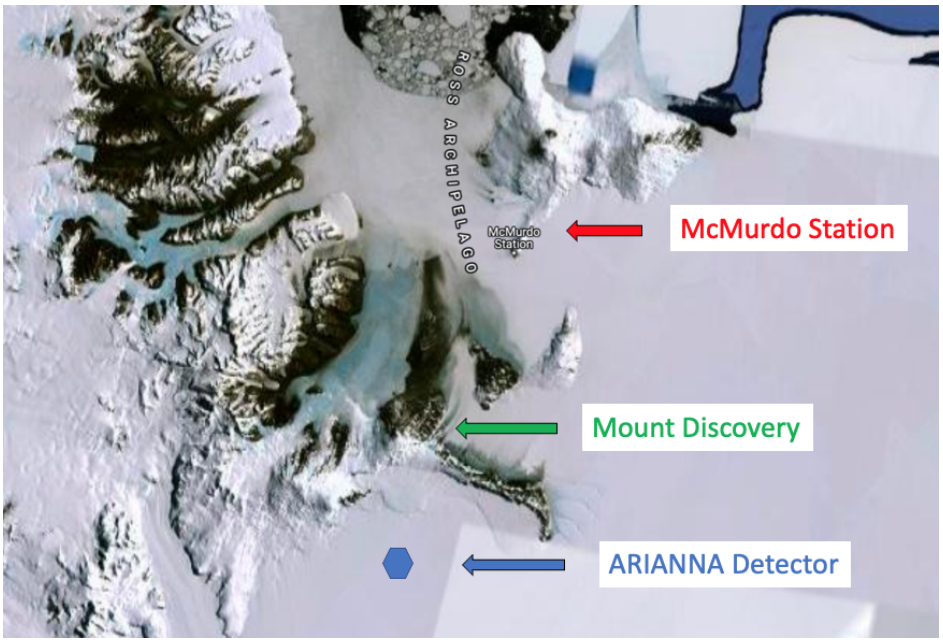

Figure 2.6: Overhead image of the Moore's Bay ARIANNA detector location as well as the surrounding area. Image from [2].

Pole due to radio noise restrictions at the ARIANNA site. In Moore's Bay the primary mode has been Afar, a high speed 2.4 GHz wireless Ethernet connection through an Afar PulsAR bridge that has a transfer speed of 200 kB/s. Each ARIANNA station is equipped with an Afar router that communicates with either an Afar relay near the summit of Mount Discovery or another station's Afar router if its connection is stronger. Next, the data are relayed to an Afar router at McMurdo Station which is connected to the internet (see Fig. 2.6 for an overhead view of the locations described). The Afar communications are managed by a constantly running listener script on the ARIANNA server at UC Irvine called Snowflake. Through Afar, data files from the last data taking window are transmitted and if the connection is too weak, data will be transferred at a later time.

In past ARIANNA deployments, Iridium has been a backup method of data transmission because while it is more reliable in harsh weather, it has a slower data transfer rate. Iridium satellite communication is handled through a Short Burst Data (SBD) protocol. The stations have an on-board SBD modem that uploads individual 300 byte messages to the Iridium

network, which are in turn sent to UC Irvine via email. As with Afar, Iridium also has a listener script hosted on the UC Irvine Snowflake server. The SBD protocol is bi-directional so commands can be sent to the ARIANNA stations via this method; such commands include changing the station's triggering threshold or configuration and requesting missing data files, to name a few.

Iridium satellite communication is relatively slow, with a typical transfer rate of one event every $2-3$ minutes. For both communication methods available to the ARIANNA project, the hardware system is currently limited to either communication or data collection; therefore, neutrino search operations are disabled during data communication. As radio neutrino technologies move beyond the prototype stage, the relatively expensive and power consumptive Afar system will be eliminated. Perhaps it will be replaced by a better wireless system, such as LTE, for sites relatively close to scientific research bases. For more remote locations, only satellite communications such as Iridium are feasible. Given the current limitation of 0.3 events/min imposed by Iridium communications and the fact that neutrino operations cease during data transfer (which generates unwanted deadtime, stations that rely solely on Iridium communication are expected to operate at trigger rates from $\sim$0.3 mHz to keep losses due to data transfer, f_{trans}, below 3%.

2.1.4 Power System

The autonomous ARIANNA stations are each self contained and primarily powered by their own singular 100 watt solar panel. Various solar panel quantities and configurations have been studied and results have shown marginal effects to adding additional panels; one is sufficient for powering a station during the summer months and additional panels only help with livetime during the transitional months. Besides solar power, there was a previous test of an ARIANNA station at the South Pole that was connected directly to the power grid.

There is also a station constructed by a group in Uppsala University, Sweden that is located in Moore's Bay and powered by solar and a wind turbine. This group is currently performing research and development for new wind turbines that could be deployed with ARIANNA stations in the future, see Sec. 2.3.1 for more details.

In order to maintain consistent power during bad weather periods and transitional seasons, every ARIANNA station has a rechargeable battery. Each battery contains a 20 amp-hour battery pack composed of 36 $LiFePO_4$ cells (with 4 cells in series) that were produced by A123 Systems. The nominal voltage of the entire battery pack is 13.2 V or 3.3 V per cell. The entire power system (the solar panel, battery, and system board) is managed by a Texas Instruments BQ40Z60EVM charge controller and is referred to as the battery management unit (BMU). The BMU is responsible for managing the power supplied to the DAQ electronics and maintaining a safe and balanced operational voltage for the battery cells. For example, a station generally runs on solar or wind if there is enough power generated and excess power charges the battery. If there is insufficient solar or wind power, the station is run by the battery until one of the battery cells falls below a low voltage cutoff value. After this point, the battery is disconnected from the station to prevent hysteresis loops in the BMU.

2.1.5 Data Acquisition

The ARIANNA DAQ can be grouped into several key subsystems; a simplified version is shown in Fig. 2.7. First the event signal is captured by the antennas and boosted by the amplifiers. The event is then run through the SST digitizer for reformatting and the FPGA to determine if the event meets the trigger requirements. Throughout these processes, the MBED as the high level control of the station is in constant communication with the other components in the DAQ. Once the event has triggered the station, the MBED saves it to

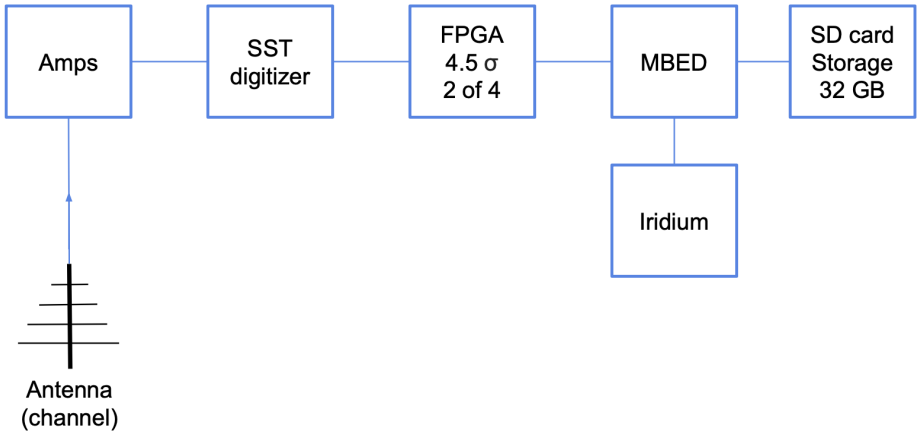

Figure 2.7: Simplified flowchart for the ARIANNA DAQ. Starting at the antenna, the event is amplified, reformatted, checked for a trigger, and then sent through the MBED to save and transmit the event at the next available communications window.

the SD card storage for transmission at the next available communications window. In the previous sections the antennas, amplifiers, and Iridium were discussed; the rest of the elements in the flow chart will be discussed next.

As of 2022, the current ARIANNA board is based around the Synchronous Sampling plus Triggering (SST) chip designed by Stuart Kleinfelder's engineering lab at UC Irvine [82]. The SST offers a simplified calibration procedure, more robust power protections, and a low power consumption of 1.7 watts [46, 94, 53]. There are two board types: the four channel DAQ board with a sampling rate of 2 Giga-Samples per second and 256 samples per channel, and the eight channel DAQ board with a sampling rate of 1 Giga-Sample per second and 256 samples per channel. A photo of the eight channel board can be seen in Fig. 2.8. Incoming voltages are held on a circular buffer until the trigger conditions are met. Once triggered the voltage record for each channel is read out and digitized by a 2.5 V 12 bit analog-to-digital-converter (ADC). The SST chip, event readout, and triggering logic are controlled by a Xilinx Spartan 4 FPGA (field-programmable gate array).

The ARIANNA triggering logic is controlled in analog with a few trigger options from which to select. The first element in the basic trigger is a threshold requirement per antenna/chan-

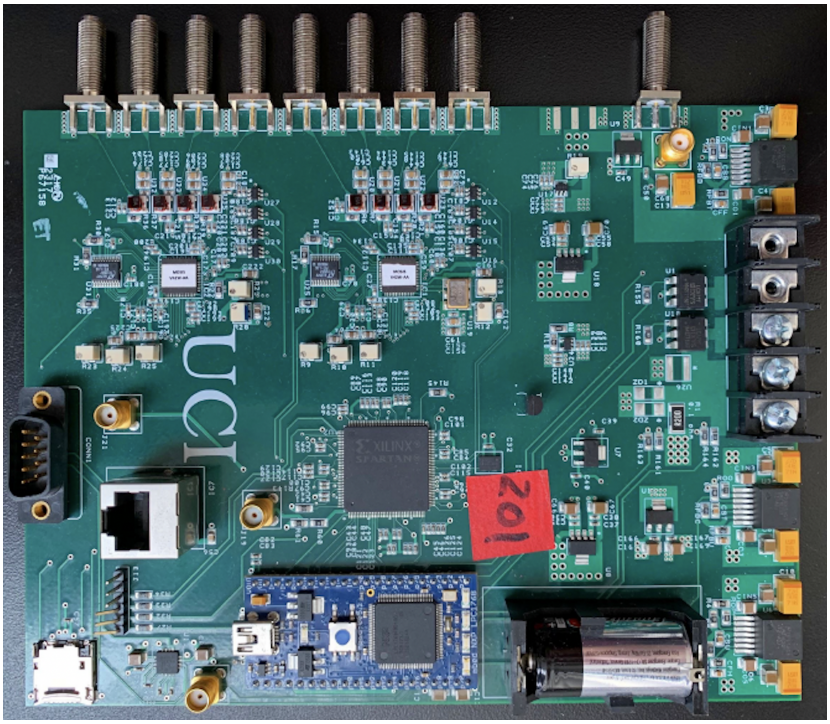

Figure 2.8: The 8 channel motherboard for the ARIANNA station.

nel. The dual threshold is the most commonly used, where a high and low threshold must be met within 5 ns. A high-low threshold requirement significantly lowers the thermal noise trigger rate and narrows in on the bipolar feature of a neutrino pulse. This effectively restricts the signal power to frequencies greater than 100 MHz. The ARIANNA trigger logic was designed by Tarun Prakash and is described in [94]. The high-low logic is for individual channels to trigger, but there are additional requirements to trigger the entire station and save the event. The most common setting is a two of four majority logic wherein any two out of the four similar facing LPDAs channels must trigger within 30 ns of one another.

These criteria are based on the expectation that thermal noise fluctuations are approximately independent. In contrast, neutrino signals produce correlated high-low fluctuations in a given antenna and create comparable signals (from an approximate plane wave source) in multiple antenna channels. These requirements reduce the rate of thermal noise triggers for a given trigger threshold while maintaining the sensitivity to Askaryan pulses from high-energy neutrinos. In addition, the parallel LPDAs are 6 m apart, so the maximum time needed for

light to travel between antennas in a refractive index of $n = 1.3$ is about 26 ns. Thus the 30 ns window allows the thermal noise trigger rates to be further reduced while ensuring that neutrinos are still captured. For both the four and eight channel boards, the standard triggering logic is the same and only uses the four similarly facing LPDAs. Any additional channels are recorded for event analysis but are not currently used to create a more complex triggering system. For stations in neutrino mode, the two of four logic applies to the four downward facing LPDAs. For stations in cosmic ray mode, the two of four majority applies to the four upward facing LPDAs.

The high level control of the station is managed by an MBED LPC1768 microcontroller. The MBED switches power to the various peripherals, sets the trigger threshold, and monitors quantities such as the voltage, current, and temperature. Another capability of the MBED is to add additional trigger requirements, such as the level one (L1) filter. L1 works in real time to veto events that are strongly peaked at a single frequency in the frequency domain. A typical neutrino event's fast Fourier transform (FFT) is broad in frequency content whereas human-made noise sources with a constant wavelength will have most of their signal at one frequency. Besides L1, a new filter is being developed for the MBED, described in Chapter 4, which utilizes deep learning for thermal noise rejection. In that same section, other microcontrollers are explored to potentially replace the MBED in future ARIANNA upgrades.

Once an event has passed all of the trigger requirements, it is read out, saved to an internal 32 GB memory card, and transferred to UC Irvine during the station's next communications window. Generally the stations are configured to send up to 300 events per communication. However, occasionally the communications drop out or event rates spike (such as during periods of high wind), which prevent complete data transfer. In this case, data files, having already been saved to the memory card, can be transmitted to UC Irvine at a later time.

2.2 Noise and Background Sources in Antarctica

Antarctica offers a mostly radio quiet environment, which is desired by in-ice radio neutrino experiments. In particular, Moore's Bay has an extremely low anthropogenic noise level because the surrounding mountains provide shielding from human activity produced by the nearby McMurdo Station. However, sources of noise that must be categorized still exist. The main source of unwanted triggers are thermal noise events, which are random noise fluctuations generated in the amplifiers. There is also galactic radio emission noise, which affects the upward facing LPDAs. These noise sources are less challenging to discriminate from neutrino signal and are lessened by the imposed trigger logic discussed previously. In the past, they would be rejected by performing a template matching procedure, and in Chapter 4 and Chapter 5, deep learning noise rejection methods are explored.

There are also intermittent periods of radio frequency (RF) noise induced by high winds on the ice surface. Stormy periods can produce event rate spikes that overwhelm the station. It is believed that wind speeds above 10 m/s charge up the surface snow and the local metal towers until a static discharge creates radio noise pulses. At Moore's Bay, wind velocities exceed 10 m/s approximately 10% of the time [22]. Wind events can vary in their waveform properties, but in general they have a somewhat pulsed shape that is more similar to neutrinos than other noise sources.

Another background source that could interfere with neutrino searches is atmospheric cosmic rays. The ARIANNA experiment utilizes cosmic rays for testing and calibration purposes, but their signal is similar to neutrinos as they both are produced from electromagnetic showers. In Fig. 2.9, three different cosmic ray air shower events measured by downward facing LPDAs show similar features to neutrino events. One technique to combat this issue is including upward facing LPDAs in the general neutrino detector design, and a few such stations have already been deployed; this aids in vetoing cosmic ray events because the

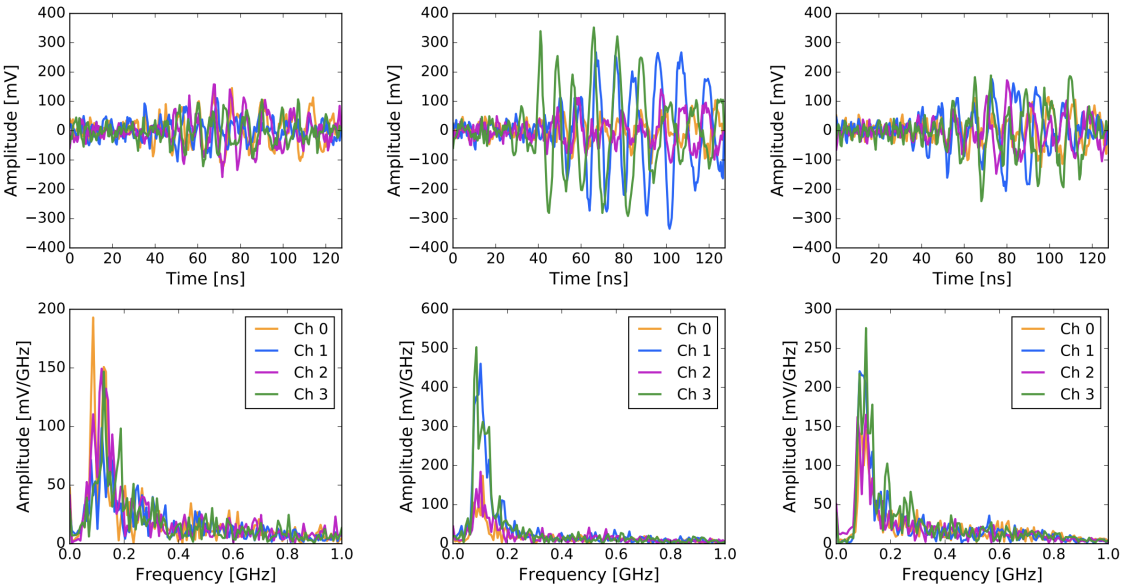

Figure 2.9: Three different events demonstrating the signal from the four downward facing antennas. These events were detected in coincidence with air shower signals from another station. The top row gives the waveforms and the bottom row gives the FFTs for each channel. From [42].

cosmic ray signal is often stronger in the upward facing antennas, and pointing can be used to determine if the signal is coming from above or below the ice surface.

The last type of background discussed here is anthropogenic noise which can be produced by many different human-made sources. In Fig. 2.10, the noise environment at Moore's Bay is shown for various phenomena such as communications and airplane flights. In general, the most significant noise groups have a narrow frequency band and are removed by the current L1 filter.

2.3 Ongoing Research and Development

There are many ongoing and planned future projects within the ARIANNA experiment. Some recent but early stage enhancements to the electronics have been made such as improved amplifiers with less noise, a new board design, and updated microcontroller and

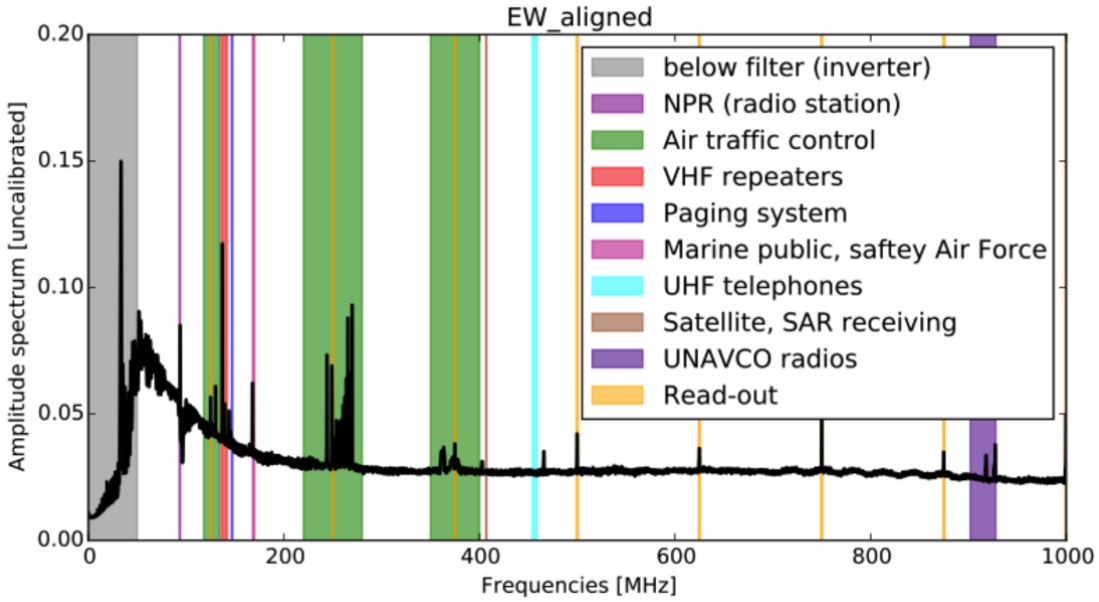

Figure 2.10: Amplitude versus frequency measurement of the noise environment at Moore's Bay. Data was recorded with a 50 MHz LPDA and then fed into an oscilloscope. The LPDA was oriented with horizontal polarity and pointed towards McMurdo Station, so these results are most likely skewed higher than the actual average noise level. From [93].

FPGA components. However in this section, a few of the more established developments will be discussed.

2.3.1 Wind Power

One major goal of ARIANNA is to extend the operation of its stations into winter to increase the livetime. With solar power alone, the stations hibernate between approximately April and September because during this time there is little to no sunlight in Antarctica. To mitigate this, effort has been made towards the addition of wind turbines to the ARIANNA stations. There are a series of engineering challenges for wind turbines in this extreme climate such as the operational temperature (bearings and grease can freeze) and the requirement that the turbine is sufficiently radio quiet. To develop these specialized wind turbines, ARIANNA collaborates with a group in Uppsala, Sweden.

Figure 2.11: Three generations of wind turbine installed at the ARIANNA site. From left to right are the first, second, and third generations installed in December 2016, 2017, and 2018 respectively. From [89].

There have been three generations of wind turbine as illustrated in Fig. 2.11. The first was deployed in 2016 to test its survival and power delivery in the harsh environment, but was not directly connected to a station. The second, installed a year later, included upgraded electronics and was directly connected to a station. The third was installed in 2018 with a larger size and was connected to a station running solely on wind power. Preliminary results with the second generation turbine have shown the livetime increases from 0% in the winter months to 24% [28]. There are, however, still challenges to work on such as making the turbines more radio quiet; in the field their operation still produces elevated trigger rates in nearby stations. After 2018 and up until now, no new deployments have been scheduled for ARIANNA. However, the same team is still developing new systems for other radio neutrino detectors such as RNO-G. A wind turbine was recently deployed for RNO-G in Greenland, with the goal to improve deployment techniques with a larger turbine and achieve 30 watts of power output. Each iteration of wind turbine at different locations and with different designs will ultimately contribute to the finalized design for the ARIANNA experiment.

2.3.2 Limited Bandwidth

Currently, the measurable frequency range used for ARIANNA stations is between 80 MHz and 1 GHz. However, simulation studies were done to measure the detector sensitivity change with restricted bandwidth. It was found that the sensitivity can be improved by more than 50% between neutrino energies of 10^{17} eV and 10^{18} eV by restricting the trigger bandwidth between 80 MHz and 200 MHz [64]. In the ARIANNA station, the bandwidth can be restricted at the trigger level and the actual saved data are read out with the full bandwidth for analysis purposes.

It is a non-intuitive result that restricting the bandwidth increases the effective volume of the ARIANNA detector. Requiring a narrower bandwidth decreases the $V_{\mathrm{RMS}}^{\mathrm{noise}}$, which allows smaller neutrino signals to be measured and increases the sensitivity. Working against this bandwidth narrowing is that noise events are biased towards having a high and low fluctuation that are no longer truly independent from one another. By optimizing for power in a narrow band that corresponds to off-cone events, it is possible to trigger on more potential neutrinos than are lost by the higher rates of the high-low trigger. The advantage of the narrower bandwidth is that the trigger thresholds can remain at 3.6 sigma, while the effective volume is equivalent to a full band trigger at 2 sigma. The implementation of this narrow band trigger would greatly benefit future ARIANNA detectors.

2.3.3 Polarization Reconstruction

One of the main goals of ARIANNA besides coincidence measurements is to measure neutrino signals and reconstruct their energy and direction to learn about their origin. Work has been done to improve the signal reconstruction procedure by comparing simulations with cosmic ray air shower events and a known calibrated transmitter antenna buried below the snow surface [30, 36], as they are well understood processes [114]. The quantities needed for

direction reconstruction are the incoming signal direction, viewing angle, and polarization. For the signal propagation, ice effects change the reconstruction ability but high precision corrections can be made to improve the measurement to a resolution of 0.37 degrees. The viewing angle has been studied through Monte Carlo simulations and has a resolution of approximately 1 degree for 10^{18} eV neutrinos. The polarization can be determined within a few degrees but there are still remaining issues such as oscillations in the measurement with depth, which need further study; thus, the limiting quantity to improve neutrino signal reconstruction is the polarization resolution. The complete resolution of the ARIANNA detector is currently around 3 degrees, which is sufficient for coincidence measurements with IceCube. Nevertheless, improvements can be made to increase the resolution further.

2.3.4 Deep Learning Techniques

The last ongoing project that will be outlined here is deep learning implementation within ARIANNA, which is the focus of this thesis. In the past, most of the analysis techniques used for ARIANNA data involved template matching and correlation for signal discrimination, while performing various cuts on data to narrow in on the desired signal. While these techniques are extremely valuable and still utilized, the exploration of deep learning as another analysis tool has been a beneficial addition. In the past few years there have been a couple of ARIANNA projects utilizing deep learning with two topics published so far: one concerned with neutrino reconstruction [67, 68] and one aiming to reject thermal noise in real time [32, 33].

The neutrino direction and energy reconstruction project uses deep neural networks to determine the reconstruction efficiency as opposed to the more classical analysis in the previous Sec. 2.3.3. Specifically, millions of simulated input voltage waveforms were generated and trained on a combination of convolutional blocks and fully connected layers. The energy is

determined with a standard deviation factor of two around the true energy, and the direction reconstruction shows a narrow peak on the order of 1 degree with tails that extend the 68% quantile to around $4-5$ degrees [68]. Further studies can be done investigating new deep learning neural network architectures and using normalizing flows to obtain an event-by-event probability distribution. Lastly, as with all of these simulated data sets generated by NuRadioMC [66], more work can be done to ensure their validity and understand systematic uncertainty compared to real ARIANNA data.

The published work concerned with rejecting thermal noise in real time is outlined in more detail than [32, 33] in Chapter 4. Another deep learning project that is yet to be published is described in Chapter 5; this study is focused on improving neutrino search techniques on experimental ARIANNA data with deep learning. There are also other areas that have yet to be extensively explored such as using deep learning directly as a trigger with recurrent neural networks. Many projects within ARIANNA could benefit from trying out deep learning techniques. However, this work has shown that deep learning is not a quick fix to analyze all problems; it is notoriously difficult to understand the features a network learns to distinguish between two sets of data.

In the next chapter, deep learning will be defined and its intersection both with general physics research and specifically with the ARIANNA experiment, will be described in more detail.

Chapter 3

Deep Learning

Deep learning (DL) was created to solve complex problems within the field of artificial intelligence (AI). While learning-based AI falls under the umbrella of machine learning, DL is a subbranch that is inspired by the structure of the brain [27]. DL neural networks consist of data inputs and interconnected nodes with associated weight and biases. The quantity and shape of the nodes dictate the network structure, and the deeper the network, the more hidden node layers. Each layer builds off of the previous layer to refine data features and optimize predictions in forward propagation; in backpropagation, errors are typically calculated using gradient descent and weights and biases are adjusted by moving backwards through the layers, with the goal of improving network predictions. While the "deep" part of the network generally describes the many hidden layers, a network only needs one hidden layer to be called deep learning. Some representative use cases of DL include distinguishing cats from dogs in images [87] and classifying handwritten numbers [108].

Unlike generalized machine learning, DL can be more flexible as the input data does not need to be preprocessed or transformed into a different format for the network. In addition, DL utilizes the ordering and relative position of data for feature classification. A few of the

most common types of problems DL excels at include processing audio, video, and natural language data streams [27]. These types of data are usually time-based and their ordering or position relative to each other matters. Thus, the structure of connected neurons uses this information for classification.

There are a few different techniques for building models within DL: unsupervised, semi-supervised, and supervised learning. Each method differs in the amount of labeled data provided to the network, where labels help the DL network categorize a particular event. Labeled data are typically straightforward to obtain with simulated data because the truth level of the events are known. Unlabeled data are more commonly generated by experimental research such as at the Large Hadron Collider (LHC), which obtains particle collision data without knowing the specific processes that occurred. Unsupervised learning is useful for data without labels that contains two or more groups; a popular approach relies on clustering data together that have similar features [97]. For semi-supervised learning, some of the input data will have labels and some will not; this approach is useful in cases where it is difficult to label the entire data set but having partially-labeled data gives somewhat better results compared to completely unsupervised learning. In supervised learning – which is the technique used in this work – data sets have corresponding labels for every event.

As discussed in the previous paragraph, simulated data is quick to label; however in most cases, current experimental ARIANNA data is also straightforward to label as noise since no neutrinos have been measured during the station run time. Recall from Sec. 2.1 that a few ARIANNA stations can search for both cosmic rays (which produce downward travel radio pulses) and neutrinos (which produce upward traveling radio pulses). Cosmic rays are more frequently measured than neutrinos, at a rate of a few per day for each station, whereas neutrinos are expected to be recorded at a rate of a few per century for each station. Nevertheless, the measurement rate of cosmic rays is still low enough such that it should not confuse a DL network if some of the experimentally acquired data are labeled incorrectly.

Perhaps a future project would use unsupervised learning to identify cosmic rays in data collected by the upward facing LPDAs of the next generation of surface radio stations.

3.1 Deep Learning in Physics

In recent years, there has been an increasing number of physics projects that utilize DL; for instance, high energy physics generates a huge amount of data to sort through to find new physics phenomena, an ideal use case for DL. Whether DL is added to a real-time detector or is used for offline analysis, it offers more autonomy for detectors and has the potential to aid in the discovery of new physics phenomena that used to be unobtainable. There are many cases where trained DL networks perform better at classification tasks than humans [62], and new DL analysis methods are improving constantly. Therefore, many physics experiments are increasingly incorporating DL into their data analysis techniques. For example, the LHC has a significant amount of data that needs to be classified in real time with low latency, so these kinds of experiments makes DL an ideal tool to utilize [59]. Plus, the LHC has a large focus on using DL offline to reconstruct collisions and their components [72]. An example closer to radio astronomy is the IceCube project, which utilizes DL for particle cascade reconstruction [17] and signal classification [55] among others. For even more examples of the broader category of machine learning in physics, refer here [80, 51].

3.2 Deep Learning Within the ARIANNA Experiment

While ARIANNA is quite new to DL techniques, the list of potential new projects is extensive as the data from ARIANNA are a good candidate for DL. An event collected by an ARIANNA station involves four (or eight) channels of 256 digital voltages (measured to a precision of 16 bits) corresponding to 256 consecutive time intervals of 1 ns. Thus, when

using the full event record, the input to the DL network consists of 256 x 4 = 1024 16 bit numbers. The time series data have relations between data points but generally do not have time dependence between different events. Lining up all of the channels together by connecting the end of one waveform to the beginning of the next one, the data can be reshaped into one dimension. Another option is to feed in a two dimensional array of stacked channels so the final input data size is, for example, 256x8 16 bit numbers. In both cases, the data sets do not need to be preprocessed extensively to perform DL. The simulated ARIANNA data sets are neutrino signals, cosmic rays (which can be either signal or noise depending on the application), and thermal noise. These data sets are accurately generated by computing tools created for this purpose; also, the simulated data are typically quick to generate, with adequate statistics created on the time scale of a few hours to a few days per computer node. This is important because often the limiting factor of a DL network is not having enough data or having only unlabeled data.

The Askaryan pulses that neutrinos produce are quite distinct from thermal noise; these pulses are typically larger, elongated, and have a wider frequency band compared to thermal noise pulses which usually contain a shorter frequency band and just one high and low pulse followed by smaller random fluctuations. Additionally, Askaryan pulses show up in multiple channels whereas thermal noise fluctuates randomly and thus is less likely to show up in multiple channels of the detector. Depending on the threshold of the trigger, it is not too difficult to distinguish between signal and noise data; in Chapter 4, it will be shown that for a threshold above 3.6 times the $V_{\mathrm{RMS}}^{\mathrm{noise}}$, a DL network identifies 95% of neutrino signals and only one background event in 100,000 is identified as signal. Thus, it is no surprise that DL works well on ARIANNA data.

3.3 Designing a Deep Learning Network

Standard terminology used for DL includes such terms as neural networks, models, parameters, and hyperparameters; these will be defined here in the context of this work. Neural networks, as mentioned above, are the collection of artificial neurons or nodes, and they are the most general element in DL processes. A DL model is the architecture of a neural network; model parameters are changed independently by the DL algorithm during training and the parameters are also described as the weights and biases of the network. In contrast, hyperparameters are the individual values set by the user within the model such as the number of neurons/nodes. Thus, if new hyperparameters are assigned, the network is now considered a different model with different parameters. A trained model not only consists of the chosen hyperparameters and architecture but also the trained parameters (weights and biases).

There are many different software frameworks for building a DL model; the largest two open source platform are PyTorch [92] by Meta and TensorFlow [15] by Google. Within the TensorFlow library is an interface called Keras [54], which is also a popular program for creating neural networks. While Tensorflow has useful tools for model building and analysis, Keras is more straightforward to use; thus, Keras is used for all DL models described in this thesis. Keras offers the ability to easily build models with various training variables for fully connected neural networks (FCNN's), convolutional neural networks (CNN's), and others. For more information on the scripts used in this work, refer to Appendix A.1. Some of Keras' capabilities will be outlined in the next section with the focus on tools that are useful for the projects in this thesis.

3.3.1 Keras Training Variables

A collection of basic Keras training variables are listed and briefly described below.

- **Dense:** the Dense function adds a single hidden layer to the model with a size specified by the user. This is the first step in a fully connected neural network (FCNN) where all of the input data connect to the first hidden layer. These connections have random initial weights and biases that will adjust throughout the training process to obtain the optimal final result. The connections are represented in Fig. 3.1 (left) by the grey lines connecting the detected radio signal and the hidden layer. One can choose how many connectors (nodes) to use and how many hidden layers to add to the model architecture. The input data is typically of the form (events, channels*samples) which lines up channels back-to-back. The one dimensional output of this layer is [input data size] x [number of nodes].

- **Conv2D:** the Conv2D function adds a convolutional hidden layer to the model; the user chooses the convolution dimensions (called the kernel size) and how many different kernels to use. This is the first step in building a convolutional neural network (CNN). Each kernel starts out with random weights and biases that will be adjusted during training and is convolved over the input data to produce an output. In a two dimensional convolution, the input data is of the form (events, channels, samples, 1, 1). The two dimensional output of this step is [input data size - kernel size] by [number of kernels] since no padding is used.

- **Dropout:** this variable describes the percentage of training nodes from the previous layer that are randomly dropped (their values set to zero). Dropout is a method of reducing over-fitting and improving generalization error. It is most useful for more complex networks with multiple hidden layers.

- **Pooling:** this variable specifies the amount of data that is down-sampled after a fully connected or convolutional layer. The pooling box size can be specified and either the maximum value or average value in that box is saved. This effectively concentrates elements into smaller sizes to summarize larger features.

- **Reshape/Flatten:** these variables detail methods for reshaping the data into a one dimensional array without changing the overall data size. For example, if the input data is an array of 10x5, the flatten step will reformat it into an array of 50x1. This step is needed before certain output layers (see below) which require one dimensional inputs. These variables can also be used between hidden layers to reformat data in general.

- **Activation:** activation functions apply various transforms to the network output of the previous layer. Such activation types include ReLU and tanh between layers, and softmax and sigmoid in the final network output layer, to name a few. ReLU is used between layers setting all negative values to zero. In the output layer, softmax and sigmoid both give output values between 0 and 1. Softmax is useful when classifying more than two groups whereas sigmoid is useful for only two groups. The output activation layer is always a sigmoid functions in this work because only two groups are being classified. For the sigmoid function, positive x values occur when y is greater than 0.5 and x is negative for values less than 0.5. This is the threshold between classifying something as group 1 versus group 2.

3.3.2 Fully Connected and Convolutional Neural Networks

After looking at the training variables, the next step is to choose which type of DL neural network to build. Both FCNNs and CNNs are similar in their structure with the main difference being how the hidden layers are constructed; one uses fully connected nodes in

the hidden layers and the other uses a convolutional kernel. Visually the differences are illustrated in Fig. 3.1, and will be outlined next.

FCNNs are the most basic networks to build of the two mentioned. One can choose as many hidden layers as desired and any of the extra variables such as pooling and drop out. Due to the web-like connected structure of input data and nodes, FCNNs use the entire input to determine features and to discriminate between the groups. As a result, FCNNs should work well with ARIANNA data. However, because the signal does not span the entire waveform for a general 256 sample input, it will be shown in later sections to not be as useful in most cases. The Keras code to train a general FCNN is found in Appendix A.3.

CNNs are slightly more complex networks that focus on smaller features in the input data. As with the FCNN, it is possible to add as many hidden convolutional layers as desired; for CNNs, this can include adding such layers as pooling, drop out, and even FCNNs in between the convolutions. Due to the convolutional nature, a neuron in the layer is not connected to the entire input; thus, each feature is identified in smaller sections and given more local information. Unlike FCNNs, CNNs can narrow in on the 50 samples of a typical ARIANNA signal waveform and recognize the other 200 samples as dominantly noise data. The Keras code for training a general CNN in found in Appendix A.2.

3.4 Training Methods

There are typically at least three kinds of data sets within a training and testing cycle: the training data, the validation data for use during training, and the testing data (once the network is already trained). The first two are needed for the training process and Keras can be set to automatically split the input data set into training and training validation. The testing set is used as an independent check to determine how well the trained model

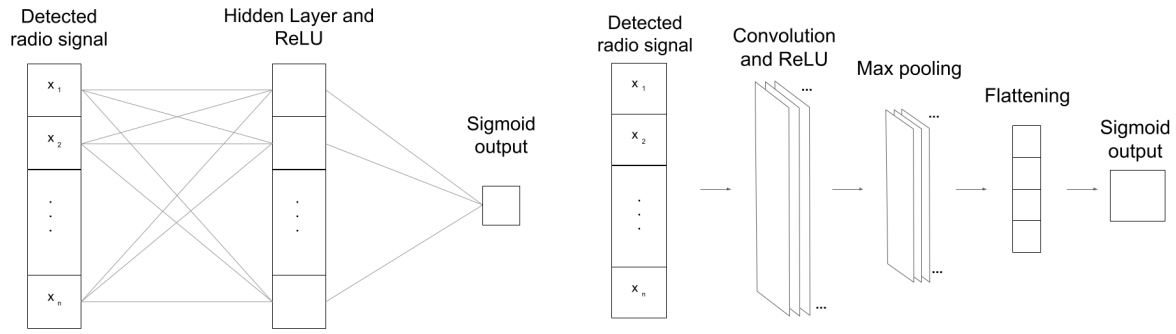

Figure 3.1: Baseline architecture of a fully connected neural network (FCNN) on the left and a convolutional neural network (CNN) on the right. The FCNN contains one hidden layer with ReLU activation and a sigmoid activation in the output layer. The CNN in composed of a convolution with ReLU activation, max pooling, a flattening layer where the data are reshaped, and a sigmoid activation in the output layer.

performs on novel data.

It is important to discuss a few best practices in neural network building. It is better to randomize the order of input data so the network does not accidentally train on the location of data in the input array. Experimental ARIANNA data are time ordered, but diffusely distributed neutrinos and cosmic rays will produce events randomly in time. However, non-thermal noise such as anthropogenic backgrounds and environmental factors vary with weather conditions and seasons; this can introduce a potential time correlation in the training data that is unique to the individual station. In general, simulated signal and noise events are already randomly distributed in time, but it is always better to shuffle these data sets into a random order before training.

In addition, a more complex network with a large amount of parameters may learn to recognize every event in the training set instead of learning how to distinguish between signal and noise events. This is called overfitting and it leads to a network that performs well on the training data but poorly when introduced to new data in the testing phase. Thus, it is recommended that input data ([number of events]x[channels]x[samples]) is of comparable size to the amount of parameters in the training model. Overfitting was not a common

occurrence in this work because of the overall small network size (one hidden layer) of the models. However, some strategies to mitigate over training are utilizing such functions as Dropout in Keras as described in Sec. 3.3.1 and testing a model's performance on new data that was not used for training. A related issue is underfitting a network which occurs when either a network is not complex enough to perform optimally or a network does not train for long enough. The network complexity issue to resolved by scanning over the model hyperparameter space as discussed in Sec. 3.4.1, and the training duration is discussed in Sec. 3.4.2.

There are three steps in the training/testing process that will be outlined in this section. Note that these steps are not sequentially ordered as they all work together to build the optimal model. The first step is scanning the hyperparameter space to determine the basic network structure for a particular analysis. The second is determining the training duration. The third is quantifying variation in the model to better understand the error in the prediction. A more optimal network structure minimizes the predictive errors.

3.4.1 Scanning the Model Hyperparameter Space

The first step in designing a DL model is choosing the basic structure. This can be done using the information above to determine if CNNs, FCNNs, a combination of those two, or perhaps a completely different type of neural network not mentioned here is the optimal option. Once a very general structure is initially determined, the model's hyperparameters need to be chosen; these are the values assigned to a given variable such as the amount of layers or the convolutional kernel size. Once the training process begins, these hyperparameters will be tuned by the user to the optimal parameter values, which can be determined by scanning through many different values. This tuning can be done manually by individually training each network; for example, training a FCNN with the same base layer size but with one,

two, and three layers to determine the optimal depth. This method is tedious and should only be used for small sweeps through model hyperparameters.

A more efficient way to start designing a model is to use a program that automatically does the scanning for the user. One such program that integrates with Keras is called Sherpa [75], a hyperparameter optimization library for machine learning models. After the user sets a range of values for one or more hyperparameters, Sherpa trains the models and then builds a report on the optimal combination of model hyperparameters. This report can be viewed in a dashboard that provides real-time visual feedback on each iteration of the model training. Scanning over a large hyperparameter space can be computationally expensive, so having a smaller range of values for each hyperparameter helps to efficiently narrow in on the optimal model, see Appendix A.1 for an example implementation of Sherpa on ARIANNA data.

3.4.2 Training Duration

This section describes a few options for how much time to spend training a model. The first option is to use Sherpa to scan over different epoch values (training cycles) to get a general sense of the ideal value. There are however more dependable ways to determine the optimal training duration.

Referring to the scripts in Appendices A.2 and A.3, Keras has an option for verbose printouts in the model fit function. These printouts give real time model training accuracy, validation accuracy, loss, and validation loss. These values help the user understand how to train the optimal network. Accuracy outlines how well the network classifies the training data in each epoch, and validation accuracy is similar in describing the training validation data set. Loss in this study is binary cross-entropy loss, which becomes increasingly larger when the predicted value get further away from the expected value, and the validation loss uses the same metric but for the training validation data set.

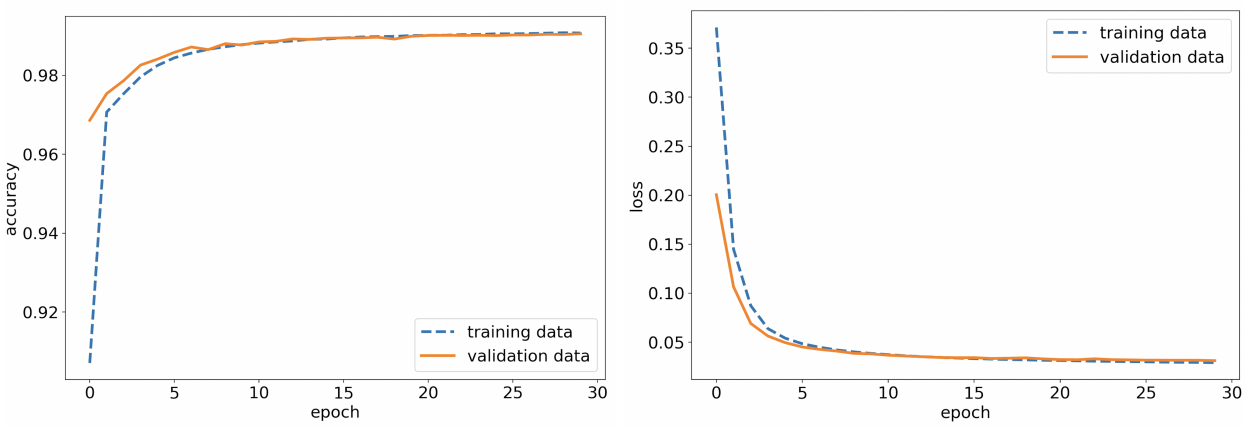

Figure 3.2: Accuracy (left) and loss (right) vs epoch number. The validation data are 20% of the input data and the other 80% are training data. This model is trained with the single channel data and baseline CNN model structure described in chapter 4.

A general method for training a network is to run the training beyond the optimal value to determine the right amount of epochs to use. This can be done visually by plotting these values as in Fig. 3.2. The goal is to use validation loss as the metric because using the training validation data does not introduce as many possibilities to overfit during the training; the validation data is separated from the iterative training and weight updates. The validation loss will keep decreasing as the network is trained, up until a point where it either plateaus (as in Fig. 3.2) or increases. Once this lowest epoch value is identified, future trainings of this same model structure and type of input data will use this value. This process has to be repeated if any of the network hyperparameters are changed. There will be some variations in the optimal amount of epochs between trainings because with randomly generated initial weights, the network does not start from the same place every time. It is possible to set a standard seed for weight starting points, but this is not necessary to obtain good results.

When a certain metric is not improving, there is an option to set automatic stopping. Monitoring software such as EarlyStopping for Keras allows the user to set stopping points when a quantity is not improving for a given amount of epochs. This is particularly useful for networks with a plateauing validation loss; instead of determining where the validation loss is the lowest, the user can end the training when the variation in the loss is low. Last in this

non-exhaustive list is saving the weights at each epoch. Libraries such as ModelCheckpoint allow the weights and biases from every epoch to be saved in a tensor, among other capabilities. However, this makes training slower and creates a lot of useless weights and biases. It is up to the preference of the user as to which training method is used.

3.4.3 Variation in the Model

Another way to avoid overfitting or misrepresenting a model between trainings is to use k-fold cross-validation (CV). There are many different variations of CV, but at its most basic level k-fold CV provides variation for the training of a DL model. There are two main scenarios in which one would use CV: to compare performance between models and to quantify the variation in the finalized model. The k-fold CV process starts by first choosing a k value, with some common choices being 3, 5, and 10. For the purpose of this example, k equals five which would produce a five-fold CV. Next, the data is randomly shuffled before being broken up into five groups. Without this step, certain training rounds could be biased by either easier or more difficult data to classify. As illustrated in Fig. 3.3, the entire randomly shuffled data set is split into five equally sized groups (D1-D5), with four of the data sets being used for training and one for testing. This is repeated for a total of five folds/iterations (K1-K5) on different combinations of the data until each set has been held once for testing. Each of the five trainings has the same model architecture and the only difference between iterations is the input data. The ability to expose a model to multiple groups of novel data in the testing phase is crucial for developing flexibility.

Once the five training rounds have completed, it is possible to measure the accuracy of each model to obtain bounds on efficiency; the mean and standard deviation of the curves describe the average value and error of the training. If the trained networks vary significantly, more than around one or two standard deviations, the model is likely not the optimal one to

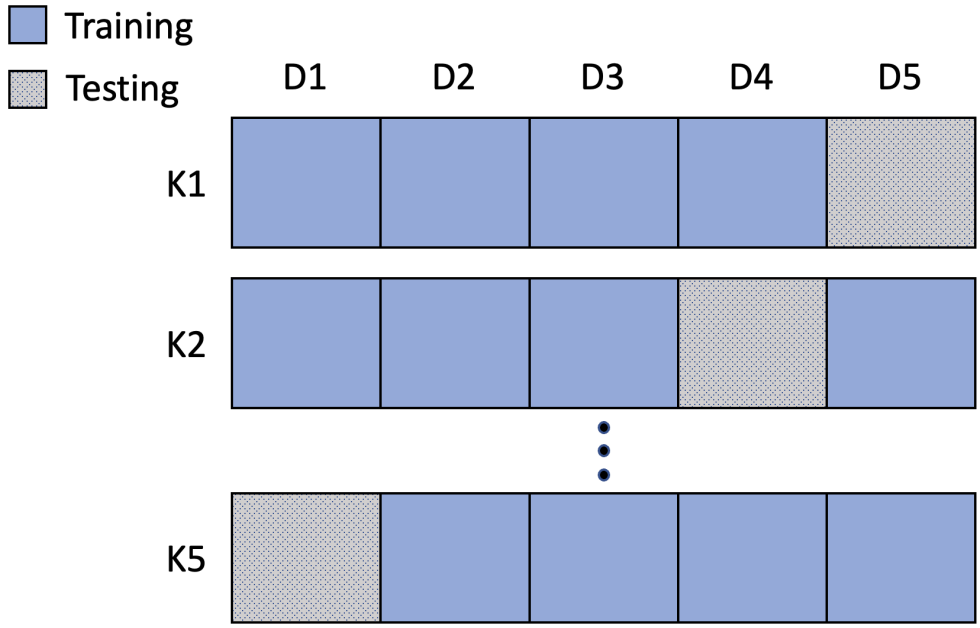

Figure 3.3: Diagram illustrating how a data set is divided for a five-fold cross validation. The blue blocks represent the training data and the gray hashed blocks represent the testing data.

use, and different architectures need to be explored. The network output of a five-fold CV study is shown in Fig. 3.4, using the same data and structure as the baseline 100 sample CNN in Sec. 4.3.2. The various colors (excluding red) represent the five separately trained models, each with its own combination of input data. Then, a separate network, shown in red, is trained on the full data set. The legend of this plot details the efficiency value for each network output of above 0.5 for signal data and below 0.5 for noise data. It is not common practice to use one of the CV networks as the final model, so the final step after cross-validation is to train a final model on the full data set. The CV distributions have the same 99% signal efficiency while noise efficiency varies between 95-97%. The red distribution, trained on the full data set, falls within the variation of the CV curves.

This section offered general guidance for training and testing a DL model within the constraints of the ARIANNA experiment. There are many methods and programs that can perform similar tasks and outcomes can vary widely based on initial requirements. Most im-

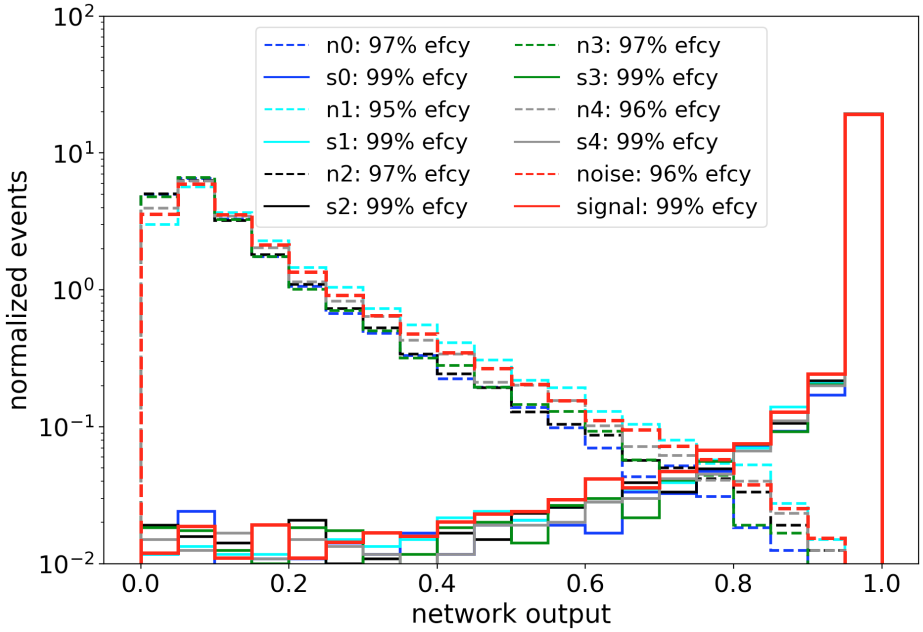

Figure 3.4: Histogram of the network outputs for a five-fold cross validation (n0-n4 and s0-s4). The signal and noise distributions for a network trained on the full data set are in red. The percentages give the efficiency with a network output cut of 0.5 for each distribution. For noise curves, the cut contains all events below 0.5 and for signal curves, the cut contains all events above 0.5.

portant is to start engaging with different model structures and hyperparameters to develop an understanding of how each element in a DL network affects the outcome.

Chapter 4

Real-time Deep Learning Trigger Implementation

Even at the extreme energies relevant to radio neutrino detectors, neutrinos rarely interact with matter. Combined with the low expected fluxes and the experimental upper limits that have been published by the IceCube Collaboration [11], the detector architecture must incorporate teratons of target material. The challenge for experimenters is to reach teraton detection volumes at a reasonable cost. The radio technique enables cost-efficient instrumentation for monitoring large detection volumes [95, 43]. However, because of the low flux of EHE (extremely-high energy) neutrinos, event rates will still be small even for the large array of hundreds of radio detector stations that is foreseen for the next-generation neutrino observatory at the South Pole, IceCube-Gen2 [74]. Thus, improving the sensitivity of the detector is one of the most effective ways to increase ARIANNA's ability to measure neutrinos. The most straightforward way to increase the sensitivity – but also the most expensive way – is to build more radio detector stations using identical technologies that were proven by the ARIANNA pilot array. Scaling up to more stations will be required to reach the long term goals of the field.

In addition to increasing the number of stations, we explore a second strategy to increase the sensitivity of the detector: lowering the trigger threshold of the detector, which would result in recording additional neutrino interactions that produce smaller amplitudes in the time dependent waveforms. In the HRA ARIANNA detectors at Moore's Bay, the trigger threshold was set to approximately 4.4 times the RMS noise, $V_{\mathrm{RMS}}^{\mathrm{noise}}$. As the expected rate of neutrinos is negligible for a single station, the observed rate of events is dominated by random thermal noise fluctuations. The trigger rate on thermal noise fluctuations changes drastically with threshold. For example, an amplitude threshold trigger with a two out of four antenna coincidence logic has a trigger rate increases by about six orders-of-magnitude if the trigger threshold is lowered from four times $V_{\mathrm{RMS}}^{\mathrm{noise}}$, to just three times $V_{\mathrm{RMS}}^{\mathrm{noise}}$ [64]. Therefore, the trigger threshold is limited by the maximum data rate a radio detector can handle which is typically on the order of 1 Hz if a high-speed communication link exists. If the communication relies on Iridium satellites, the maximum data rate is limited to 0.3 mHz. However, if thermal noise fluctuations are identified and rejected in real time, the trigger thresholds can be lowered while maintaining the same data rate, thus increasing the effective sensitivity of the detector. The sensitivity can be improved by up to a factor of two with the intelligent trigger system presented here (see Sec. 4.2).

This chapter demonstrates that DL (deep learning) can be used to reject thermal noise in real time by implementing these techniques in the current ARIANNA data acquisition system. By rejecting thermal events, the trigger rate can be increased dramatically while maintaining the required low rate of event transmission over the communication links from the remotely located ARIANNA stations. Overall, lower thresholds increase the effective volume of ice observed by each station, which is proportional to the sensitivity of the detector:

$$sensitivity \propto \frac{t_{op}\ \Phi\ E\ V_{eff}}{L_{int}} \tag{4.1}$$

where t_{op} is the operational time of the detector, Φ is the chosen neutrino flux model, E is the neutrino energy, V_{eff} is the effective volume (in units of m^3 x $steradian$), and L_{int} is the neutrino interaction length.

In this chapter, additional details will be given on the ARIANNA data acquisition and transmission, along with the expected gain in sensitivity for this study. The trade off between a specific DL network and processing time is assessed to find the optimal DL models for a representative sample of microprocessor platforms. The final network chosen is then studied to understand the network's learned features such as the elongated pulse shape of the neutrino. The DL trigger is compared to a template matching procedure commonly employed in previous ARIANNA analyses [29]. It is shown that the efficiency of neutrino selection and the rejection of thermal events by the DL network is superior to the template-based approach.

The promising results of the deep-learning-based filter were obtained by offline computer studies, but this technique can be implemented on the restricted hardware of the current ARIANNA data acquisition system; Section 4.6.2 describes the successful porting of the DL filter to the ARIANNA MBED microcomputer and Sec. 4.6.3 finds the maximum rate of processing events to be around 10 Hz. Though this rate provides a strong boost in performance, it does not take full advantage of the observed rejection of all but 1 in 10^5 thermal events. To fully exploit the DL trigger, the onboard computer is required to apply the filter at a rate of 1 kHz which is above the incoming event average rate of 100 Hz. This is due to the randomness of incoming events and at an average rate of 100 Hz, some events will be triggered closer together; to limit the deadtime, a factor of about 10 greater than the triggered event rate is needed, see Sec. 4.6.3. We study several potential upgrades to the MBED microprocessor. In Sec. 4.6.4, the DL filter is tested on experimental cosmic rays to verify that they are classified similar to neutrino signal and not rejected as thermal noise. The chosen network is further optimized for efficiency and lastly, the flexibility of the network is

studied by adding continuous waves to the input data. This chapter concludes with a short summary and plans for the future.

The majority of the work in this chapter was published and can be found in [33]. The additional sections added to this chapter include the interpretability of networks in Sec. 4.4, neural network hyperparameter optimization in Sec. 4.7, and network performance with continuous waves in Sec. 4.8. Furthermore, Sec. 4.5 has been expanded to include more information about the Raspberry Pi Zero microprocessor such as its power consumption and processing capability.

4.1 Data Acquisition

Once a triggered event is saved to local storage (the memory card), it is transferred to UC Irvine through a long-range WiFi link (AFAR) [19] during a specified communication window. The ARIANNA stations also use Iridium satellite network as a backup system. Satellite communication is relatively slow, with a typical transfer rate of one event every $2-3$ minutes [63]. For both communication methods, the current hardware system is limited to either communication or data collection; therefore, neutrino search operations are disabled during data communication. As radio neutrino technologies move beyond the prototype stage, the relatively expensive and power consumptive AFAR system will be eliminated. Perhaps it will be replaced by a better wireless system, such as LTE, for sites relatively close to scientific research bases, but for more remote locations, only satellite communications such as Iridium are feasible. Given the current limitation of 0.3 events/min imposed by Iridium communication and the fact that neutrino operations cease during data transfer which generates unwanted deadtime, stations that rely solely on Iridium communication are expected to operate at trigger rates from $\sim$0.3 mHz to keep losses due to data transfer, f_{trans}, below 3%.

The trigger thresholds of ARIANNA are adjusted to a certain multiple of the signal-to-noise ratio (SNR), defined here as the ratio of the maximum absolute value of the amplitude of the waveform to the $V_{\mathrm{RMS}}^{\mathrm{noise}}$. Currently, the pilot stations are set to trigger above 4.4 SNR to reach the constrained trigger rate of order 1 mHz. In the next section, the expected gain in sensitivity is studied for a lower threshold of 3.6 SNR, which corresponds to 100Hz, the maximum operation rate of the stations.

4.2 Expected Gain in Sensitivity

The real-time rejection of thermal noise that is presented in this article would enable the trigger threshold to be lowered significantly – thus increasing the detection rate of EHE neutrinos – while keeping a low event rate of a few mHz. To estimate the increase in sensitivity, the effective volume of an ARIANNA station is simulated for the two trigger thresholds corresponding to a thermal noise trigger rate of 10 mHz (the current ARIANNA capability), and a four orders-of-magnitude higher trigger rate (enabled through the deep-learning filter that rejects 99.99% of all thermal noise triggers). We use the relationship between trigger threshold and trigger rate from [64] to calculate the thresholds.

NuRadioMC [66] is used to simulate the sensitivity of the ARIANNA detector at Moore's Bay. The expected radio signals are simulated in the ARIANNA detector on the Ross ice shelf, i.e., an ice shelf with a thickness of around 576 m and an average attenuation length of approx. 500 m, and where the ice-water interface at the bottom of the ice shelf reflects radio signals back up with high efficiency. The generated neutrino interactions are distributed uniformly in the ice around the detector with random incoming directions. The simulation is performed for discrete neutrino energies and includes a simulation of the full detector response and the trigger algorithm as described above. The resulting gain in sensitivity is shown in Fig. 4.1 and increases by almost a factor of two at energies of 10^{17} eV. The

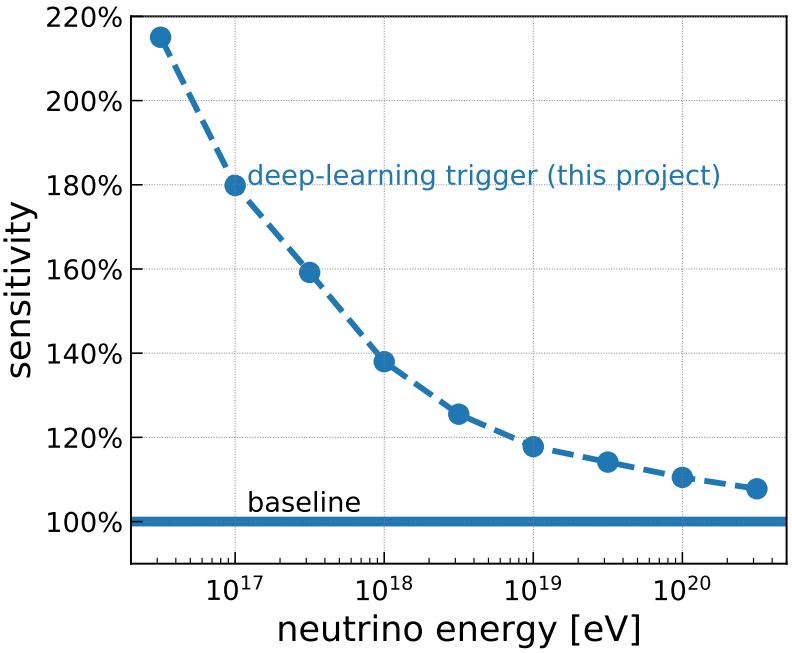

Figure 4.1: Expected improvement in sensitivity to high-energy neutrinos with the deep-learning trigger developed in this work. The improvement in sensitivity directly translates into the number of observed neutrinos. The baseline is the standard ARIANNA high/low trigger with a two out of four antennas coincidence requirement for the nominal bandwidth of 80 MHz to 800 MHz at a thermal noise trigger rate of 10 mHz. The blue dashed curve shows the sensitivity for a trigger threshold corresponding to a trigger rate of 100 Hz and otherwise the same simulation settings.

improvement decreases towards higher energies because fewer of the recorded events are close to the trigger threshold but at 10^{18} eV there is still an increase in sensitivity of 40%.

4.3 Thermal Noise Rejection Using Deep Neural Networks

To implement a DL filter, the general network structure needs to be optimized for fast and accurate classification. For accuracy, the two metrics tracked are neutrino signal efficiency (defined here as the ratio of correctly identified signal events to the total number of signal events) and noise rejection factor (defined here as $\frac{1}{(1-N_{\text{ratio}})}$, where N_{ratio} is the ratio of

correctly identified noise events to the total number of noise events). The goal is to reject several orders-of-magnitude of thermal noise fluctuations while retaining most of the neutrino signals. In this chapter, the target is five orders-of-magnitude thermal noise rejection while providing a high signal efficiency at or above 95%.

Typically using a more complex network structure yields more accurate results, but this also creates a slower network. These trade offs need to be optimized as the DL architecture is developed. In the following two sections, DL techniques are used to train models then study their efficiency and processing time. In Sec. 4.6.1, a commonly used method of template matching will be investigated to compare with the DL approach.

4.3.1 Generation of Training Data Sets

NuRadioMC [66] is used to simulate a representative set of the expected neutrino events for the ARIANNA detector, utilizing the same setup as described in Sec. 4.2 but for randomly distributed neutrino energies that follow an energy spectrum expected for an astrophysical and cosmogenic neutrino flux. To generate the appropriate data, the astrophysical flux measurement by IceCube with a spectral index of $\gamma = 2.19$ [50] is combined with a model for a GZK neutrino flux [24, 110] based on Auger data for a 10% proton fraction [109].

For this project, a simplified version of the ARIANNA detector is used with only four downward facing LPDAs as shown in Fig. 4.2. The resulting radio signals are simulated in the four LPDA antennas of the ARIANNA station by convolving the electric-field pulses with the antenna response, and the rest of the signal chain is approximated with an 80 MHz to 800 MHz band-pass filter. An event is recorded if the signal pulse crosses a high and a low threshold of 3.6 times $V_{\mathrm{RMS}}^{\mathrm{noise}}$ within 5 ns in at least two LPDAs within 30 ns. At such a low trigger threshold, noise fluctuations can fulfil the trigger condition at a non-negligible rate. Therefore, the signal amplitude is required to be at least 2.8 times the $V_{\mathrm{RMS}}^{\mathrm{noise}}$ before adding

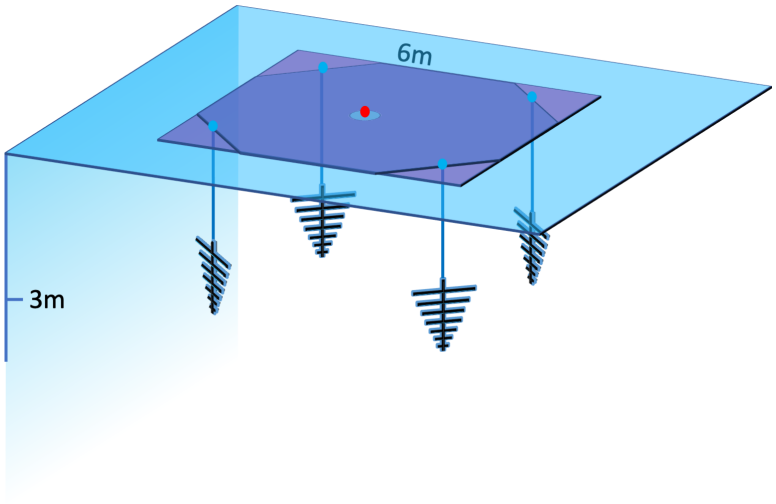

Figure 4.2: Diagram of a typical ARIANNA station consisting of four downward facing log periodic dipole antennas (LPDAs) located three meters below the Antarctic ice.

noise to avoid spurious triggers on thermal-noise fluctuations. In total, 121,597 events that trigger the detector are generated and this is called the *signal data set* in this chapter.

The training data set for thermal noise fluctuations is obtained by simulating thermal noise in the four LPDA antennas and saving only those events where a thermal noise fluctuation fulfills the trigger condition described above. In total, 1.1 million events are generated and this is called the *noise data set* in this chapter.

The limitations of the simulations and their impact on the obtained results are discussed at the end of this article.

4.3.2 Network Structures and Training

Our primary motivation is to develop a thermal noise rejection method that operates on the existing ARIANNA hardware with an evaluation rate of at least 50 Hz, which is a factor of 10^4 higher than our current trigger rate. To increase the execution rate of the neural network, one option is to optimize hardware; however, any alteration to the hardware is constrained

by two main factors: the power consumption of the component and its reliability in the cold climate. Thus, this study will focus primarily on optimizing the execution rate by identifying the smallest neural network that reaches our objective. While the number of trainable parameters can give an indication of network size, the number of Floating Point Operations (FLOPs) is the chosen metric for network size in this work. The number of FLOPs can be approximated by multiplying the amount of (+,-,*,/) operations performed on floating point numbers with the amount of nested loop iterations required to classify incoming data. To obtain the amount of FLOPs for a given model, refer to Appendix A.4 and A.5.

Besides making the network size smaller, another way to improve network speed is to reduce the input data size. Instead of feeding the signal traces from all four antennas into the network, it is possible to cut down on the size of input data by only using the two antennas that caused the trigger. As each signal trace consists of 256 samples, the total input size to the network is 512 samples. In addition, a further reduced input data set is studied for various sizes by selecting the antenna with the highest signal amplitude and only using a window of values around the maximum absolute value. The window size was not fully optimized, but a good balance between input data size and efficiency is 100 samples around the maximum value. The reasoning for this is that the dominant neutrino signal does not span over the whole record length and typically only spans over less than 50 samples.

The two network architectures studied in the following Sec. are a fully connected neural network (FCNN) [100] and a convolutional neural network (CNN) [56, 81], depicted in Fig. 3.1. The FCNN used in this baseline test is a fully connected single hidden layer network with a node size of 64 for the 100 input samples and 128 for the 512 input samples, a ReLU activation, and a sigmoid activation in the output layer. The CNN structure consists of 5 filters with 10x1 kernels each, a ReLU activation, a dropout of 0.5, a max pooling with size 10x1, a flattening step to reshape the data, and a sigmoid activation in the output layer.

Both the CNN and FCNN are trained using the Adam optimizer with varying learning rates from 0.0005-0.001 depending on which value performs most efficiently for each individual model. The training data set contains a total of 100,000 signal events and 600,000 noise events, where 80% is for training and 20% is to validate the model during training. Once the network is trained, the test data are used which contain 21,597 signal events and 500,000 noise events. For more information on the scripts used in this study, see Appendix A.1.

4.3.3 Deep Learning Performance

The signal and noise event classification score distributions from the networks are distinct. With the sigmoid activation in the output layer, the classification distribution falls between zero and one, where close to zero is noise-like data and close to one is signal-like data. Once trained with the 100 input sample CNN mentioned above, the distribution shown in Fig. 4.3 is obtained. From this distribution, the amount of signal efficiency versus noise rejection can be varied by choosing different network output cut values. Training and testing these networks with each input data size yields the signal efficiency versus noise rejection plot in Fig. 4.4. Each data point corresponds to a different network output value, and the final cut value is chosen by optimizing the noise rejection for the desired signal acceptance. All of these input data sizes produce efficiencies above the required threshold of 95% for signal, and all were able to reach at least five orders-of-magnitude noise rejection.

Since all of the networks have efficiencies above our target of 95% for signal at 10^5 noise rejection, the main consideration is the amount of FLOPs required for each network because this directly impacts the processing time. Typically, CNN's have fewer parameters overall due to their convolutional nature, which focuses on smaller features within a waveform; comparatively, the FCNN considers the whole waveform to make its prediction requiring more node connections. The next step is to investigate the FLOPs for each network, and

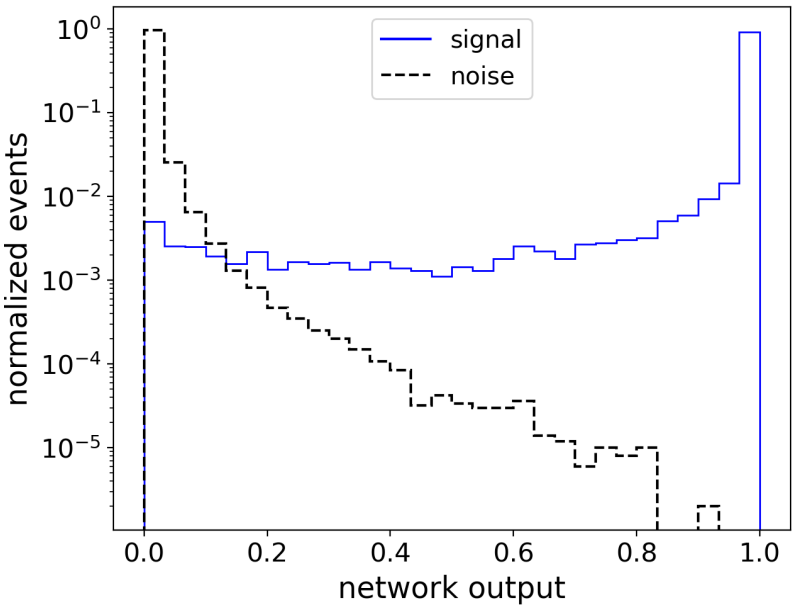

Figure 4.3: Histogram of the network output for signal and noise classification. The network used for training, validation, and testing was a CNN with one convolutional layer comprised of five 10x1 filters and input data of 100 samples around the maximum value of the waveform.

determine the processing time on a given device in Sec. 4.5. Before this, a brief exploration of interpretability will be discussed in the next section.

4.4 Interpretability of Networks

Trained neural networks are notoriously difficult to interpret. It is straightforward to train a network and obtain good prediction results, but the difficult part comes from understanding how the network arrives at its final classification result. One advantage of the real-time DL analysis is that the trained networks are extremely small and consist of only one hidden layer. It can be more straightforward to understand how a network is learning features when it has fewer layers and less complexity. In this section, the 100 input sample CNN from Sec. 4.3.2 will be analyzed to gain insight on learned features. First a new network will be trained on data with an artifact to better visualize the process. Then, the same 100 input

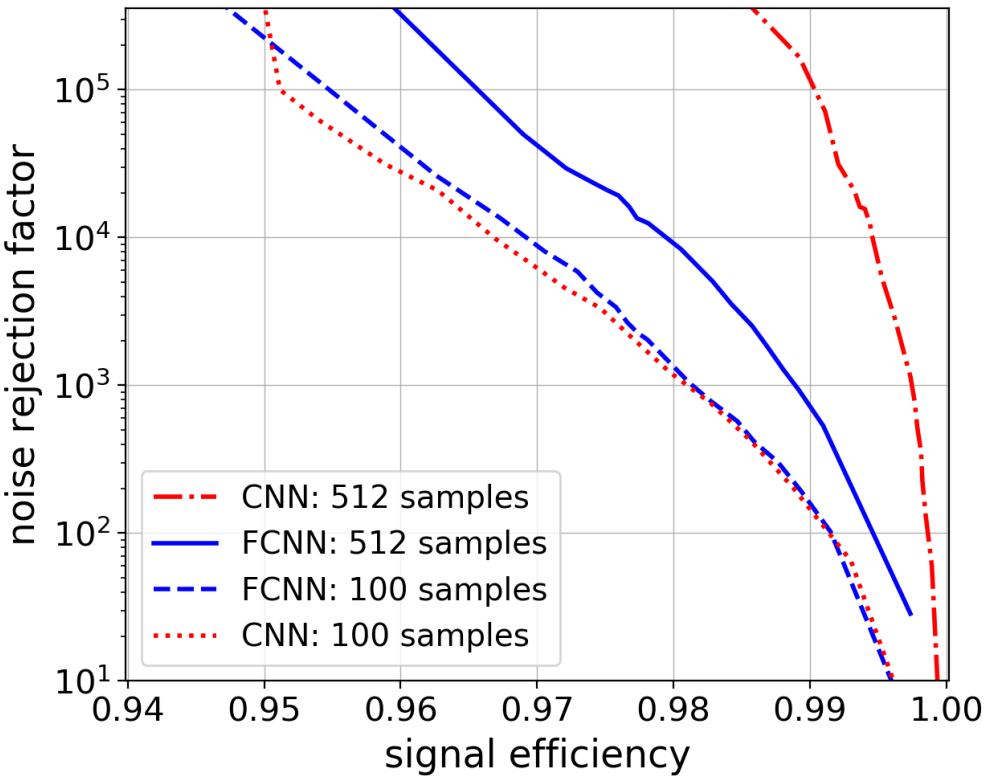

Figure 4.4: Signal efficiency versus noise rejection factor for FCNN's and CNN's with two different input data sizes (100 samples and 512 samples). Both CNN's have the structure of one convolutional layer containing five 10x1 filters. The FCNN's have one fully connect layer with node size 64 for the 100 samples input data and node size 128 for the 512 samples input data.

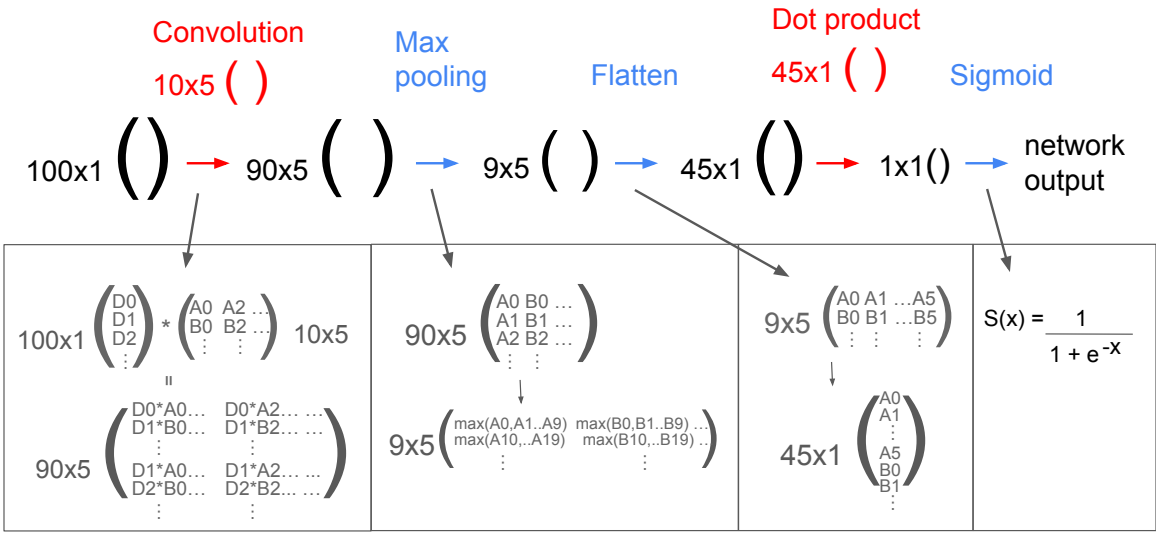

Figure 4.5: Flowchart of the network classification process for the 100 input sample CNN. The steps marked in red are the trained parameters in the network and the steps in blue do not use any trained network parameters. The first number describing the size is the vertical dimension and the second is the horizontal dimension.

sample CNN from before will be investigated.

In both cases, the input data and CNN structure used are the same: a one layer convolution with five 10x1 filters, a max pooling of 10x1, flattening, a fully connected layer, and a sigmoid activation in the output layer. A flowchart of the steps in the classification process is shown in Fig. 4.5. The top half of the figure shows the data dimension after each labeled step in the event classification process (after training has finished). The steps in red use the trained network parameters whereas the blue steps are required for the calculation but do not use the trained parameters. The bottom half of the figure (in boxes) shows a simplified process for how each step is calculated and how each array is manipulated. While it in not necessary for most network building purposes to learn the inner workings of the network, for interpretability it does help to understand each step.

Once Fig. 4.5 is understood, the steps in the classification of an event can be plotted to

determine the important features; also refer to Appendix A.1 for resources on interpretability studies. To begin, the same simulated data from Sec. 4.3.1 are used except the thermal noise data are altered so that the first 10 samples are equal to 50 mV in each event. Then, the network is trained with unaltered simulated neutrino signal and altered simulated thermal noise, called the ARTIFACT network. Because there is this altered feature in the first 10 samples of the noise data, a network could use this as the main learned feature in the training process.

In Fig. 4.6, the classification steps are shown for the ARTIFACT network with two different input events. On the left side is a thermal noise event with the bias added to the first 10 samples and on the right side is an unaltered neutrino signal event. The top plots include the input waveform in black. Below that are the five 10x1 filters/kernels given in different colors, which are plotted spaced apart to visually see their scale compared to the input waveform. The filters are learned parameters of the ARTIFACT network, so they are the same between different classifications. All of the filters are convolved with the input waveform, a bias value is added, and then a ReLU activation is applied that sets all negative values to zero. The middle two plots show the results after these steps, corresponding to the 90x5 array output after the convolution in the flowchart. These filters are then condensed via max pooling so that only the maximum value in each 10 sample interval is kept. The resulting data are plotted in the bottom of Fig. 4.6 in blue. The red data are the trained parameters of the network and are referred to as the dot product step in the flowchart. The red and blue data are then multiplied together to obtain a single value that is fed into a sigmoid function to get the final classification value. To obtain a sigmoid value above 0.5 (classified as signal), the input needs to be positive; values below 0.5 (classified as noise) are obtained with a negative input. Thus, the red curves on the bottom plots indicate which features contribute to the final classification.

Three out of five filters on the left side of Fig. 4.6 (1, 3, and 4) pick up the noise bias as

they are more or less horizontal lines. The noise event will have a high value in the first 10 samples, so the last four filters have fully connected weights in the last plot that compensate for this increase. The last plot shows the red data dips down into negative values for this region when the blue data is peaked, which will contribute a negative dot product. Filter 0 is the main signal-like feature with more positive fully connected weight values. Therefore, a trained network can focus on both signal and noise features, depending on if the final fully connected layer contributes positive or negative values.

When training a neural network, steps need to be taken to reduce the chances of over training and confirm the presence of an unbiased data set. Taking the ARTIFACT network and feeding in the same noise event without the 10 sample bias causes the network to classify the event incorrectly as signal (a classification value of one). The ARTIFACT network learned surface level features to distinguish between the two groups and was not flexible enough to correctly classify a non-altered noise event.

The next example uses the same process of plotting the network output after each classification step but without altering the input data. The trained network is the 100 input sample CNN from Sec. 4.3.2. The same events are used for signal and noise but without the bias added to the noise. The plots are shown in Fig. 4.7. The filters in this network seem to be more similar to one another with slight variation. In general, most of the filters have high and low spikes spanning around five samples. One characteristic of neutrino events is a slightly more spread out high and low pulse compared to the more sharp and quick pulse of thermal noise that might only span a few samples. Note the middle plots in Fig. 4.7 have different scales between the signal and noise event, and the signal events in most cases have much larger peaks; this means the filters correlate more with signal events to produce a significant positive value.

In addition to the filters' shapes, the fully connected weights contribute the most positive values either around 4 samples or 6-7 samples in the bottom plots. Filters 1 and 4 pick up

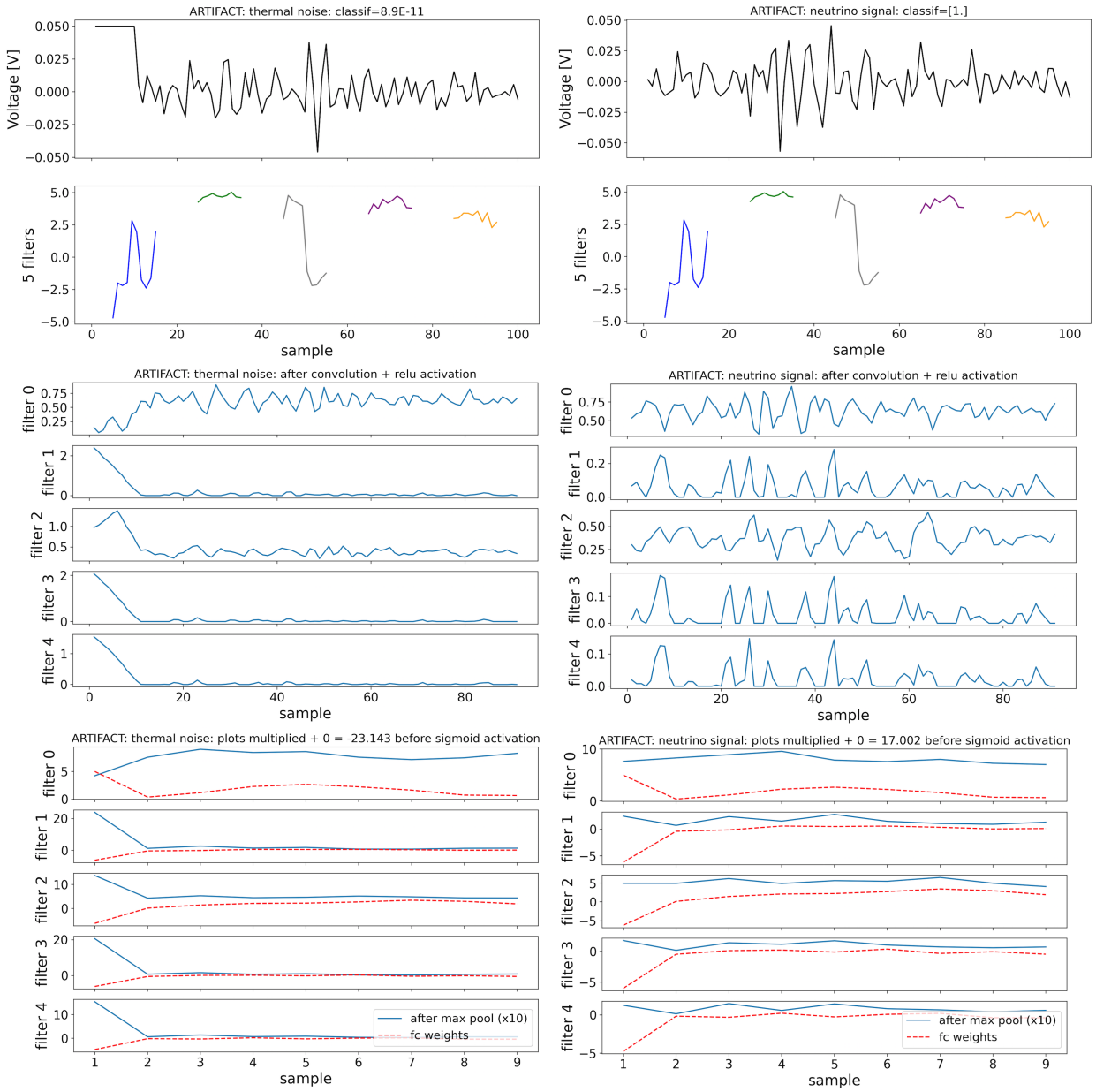

Figure 4.6: Plots of the ARTIFACT network output at various steps in the classification for a biased thermal noise event (left) and a neutrino signal event (right). The top plots are the input waveform (top) and the five 10x1 trained filters (bottom). The middle plots give the output of each filter after the convolution and ReLU step. The bottom plots give the output of the network after max pooling x10 to better see the distribution (blue data) and the trained fully connected weights of the network (red data).

on signal events that peak closer to the beginning, and the others pick up on signal events that peak towards the end of the event. In these data sets, the data is cut to 100 samples around the maximum value; however, if the maximum value occurs sooner than sample 50, the first 100 samples are taken. Thus, these two regions show the network recognizes the various possible pulse locations. Lastly, there are generally more correlation pulses in the right middle plot of the signal event. Because neutrino signal typically has a longer shape and more high and low pulses, the filters positively correlate to many of these spikes. There are potentially more features within these plots that give insight into the training process, but in general this network does prove to be flexible.

When the biased thermal noise event from the previous plots is fed into this 100 sample CNN network trained on regular data, it is classified correctly as noise. Further, many biased noise events were tested with the same correct prediction results. This network is flexible enough that even with a 50 mV bias in the first 10 samples of the input data, the network focuses on the important distinguishing features between the two groups. Studies investigating network performance on slightly altered data can offer some confidence that a particular neural network is functioning correctly.

4.5 Processing Time and Reliability on Devices

In this section, the processing time of the DL filter is studied. As the filter is intended as a real-time trigger, a fast execution time is crucial. The current ARIANNA hardware is used to test and measure the execution time under realistic conditions.

There are several time components that impact the physical capabilities of the ARIANNA detector: (1) the time to transfer the data from the waveform digitizers to the microcomputer, T_{read}; (2) the time to reformat and calibrate the raw data for the DL evaluation, T_f; (3) the

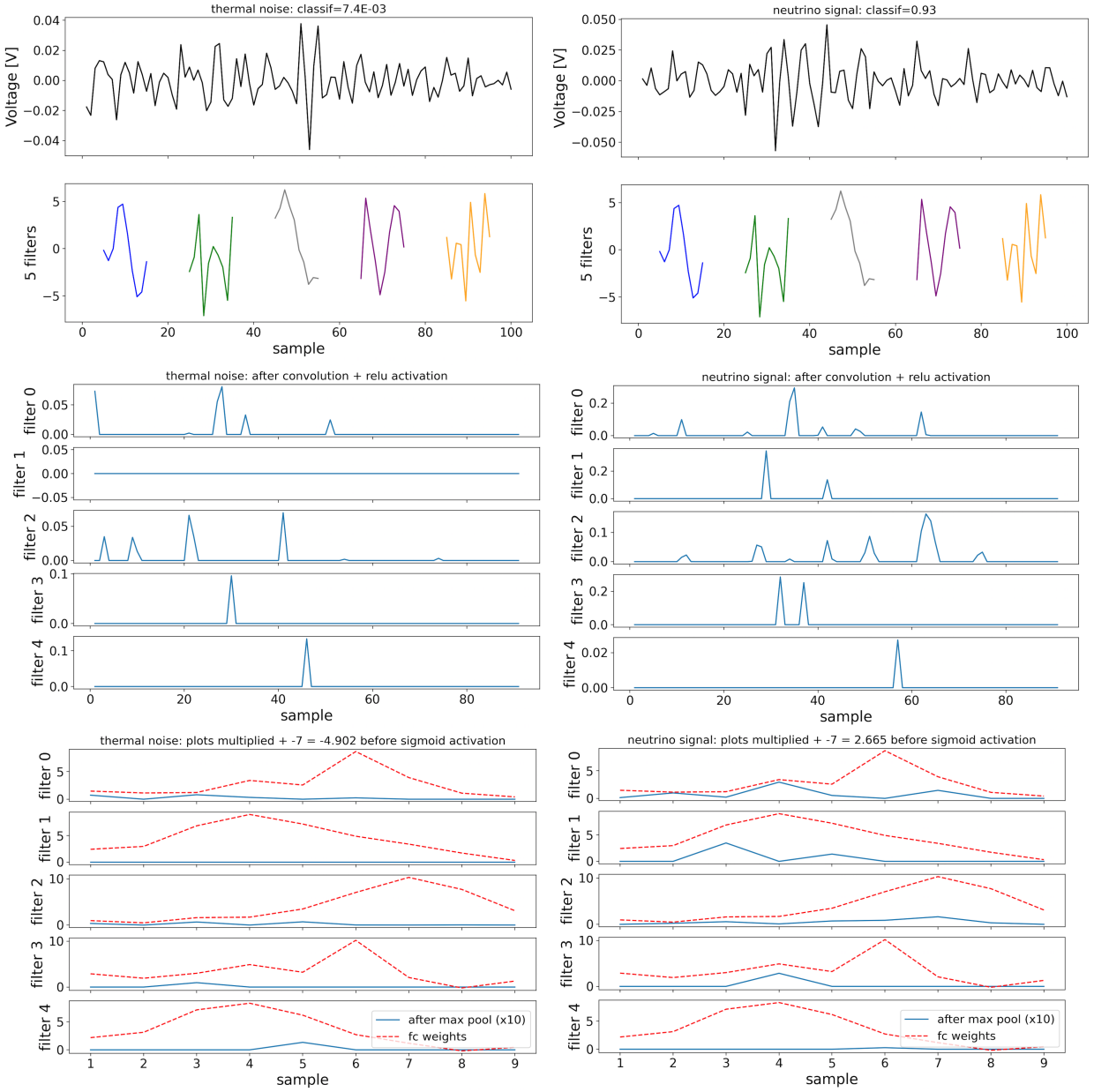

Figure 4.7: Plots of the 100 input sample CNN network output at various steps in the classification for a thermal noise event (left) and a neutrino signal event (right). The top plots are the input waveform (top) and the five 10x1 trained filters (bottom). The middle plots give the output of each filter after the convolution and ReLU steps. The bottom plots give the output of the network after max pooling x10 to better see the distribution (blue data) and the trained fully connected weights of the network (red data).

time to evaluate the event with DL and make a decision, T_{dl}; (4) the time to store an event to the local SD card; and (5) the time to transmit the event via the Iridium satellite network. The architecture of the ARIANNA pilot station prohibits it from acquiring neutrino events while the data acquisition system is processing events or during the transmission of data via Iridium satellite. It is useful to express the processing time in terms of fractional loss of operational time, or deadtime. Operational livetime, L, is the calendar time of nominal operation, T, corrected for the time losses due to event processing and transmission. Often, the operational livetime is reported as a fractional quantity, $f_L = L/T$ and the fractional operational deadtime, f_D=1-f_L.

If the rate of saving events to the SD card is sufficiently low, then the time to process an event using DL is given by $T_{\rm min}$=$T_{\rm read}$+T_f+T_{dl}. As shown in this section, the latter two time scales depend on the microprocessor. The time to transfer data to the microprocessor depends on the details of the data acquisition system. This is known for the pilot ARIANNA stations, $T_{\rm read} = 7.3\,{\rm ms}$, but the design of this station did not focus on minimizing $T_{\rm read}$, and this value can be reduced by redesigning the hardware. It is assumed to be negligible when evaluating new platforms for future designs of the ARIANNA data acquisition system.

4.5.1 Processing Time

Three microprocessors are explored for their processing time and power consumption: a Raspberry Pi compute module 3+ microcomputer, a Raspberry Pi Zero 2 W, and an MBED LPC1768 ARM microcontroller. The MBED is the current device installed in ARIANNA and it runs on a custom C code. The Raspberry Pis are microcomputers with a Raspbian operating system, which is based on Debian. As with the MBED, the neural network is implemented with similar custom code on the Raspberry Pis. Since the optimal networks found in the previous section are small and shallow, custom code is written that implements

the trained neural networks in C for maximum performance. To test the prediction capabilities and the classification time in both devices, a simulated event is read in and either matrix multiplied by the array of weights and biases in the FCNN case or convolved with the weights and bias filters in the CNN case.

Two methods are used to measure the MBED and Raspberry Pi processing times, T_{dl}. For the Raspberry Pis, since they are not attached to the ARIANNA data acquisition system (DAQ), the processing time is measured by looping over the processing code 100 times, while measuring the total time for 100 loops with the clock function in C. The total time divided by 100 is the average processing time per event. Since the MBED is attached to the ARIANNA board, it has the ability to be probed for reset pulses, which are used to reset the logic of the FPGA and trigger circuitry to prepare for a new event. The time between reset pulses will provide the total deadtime. In the case that the station is triggered continuously, which would result in 0% livetime, the time between reset pulses corresponds to the processing time. Livetime is defined as the time between the reset pulse of the previous event and the trigger of a new event. To accomplish this setup, a pulse is injected into the hardware with an amplitude large enough to trigger the system. By increasing the injection rate until the system experiences 0% livetime, the minimum time difference between reset pulses, T_{min}, can be found.

T_{min} can be broken down into three time components that when summed together make up the processing time, $T_{\text{min}}=T_{\text{read}}+T_f+T_{dl}$. To measure each component, different processing functions are disabled. For example, when the DL code and reformatting code are disabled, T_{min} equals T_{read}. The processing time T_{dl} and reformatting time T_f for both devices are shown in Table 4.1 for a given model along with the FLOPs of each network. While Table 4.1 shows that the relationship is not completely linear, FLOPs provide a reasonable proxy to estimate the relative speeds of specific DL models. All models listed reach the required efficiency of 95% neutrino signal at 10^5 noise rejection. Therefore, the fastest network is

chosen for the lab tests, which is the CNN with 100 input samples.

Variable	Model	FLOPs	MBED	Raspberry Pi
T_{dl}	FCNN 512 samples	131,457	45 ms	2.5 ms
	CNN 512 samples	55,816	*	1.5 ms
	FCNN 256 samples	32,961	13 ms	1.0 ms
	CNN 256 samples	27,376	9.4 ms	0.95 ms
	FCNN 100 samples	12,993	4.7 ms	0.46 ms
	CNN 100 samples	10,096	3.7 ms	0.39 ms
T_f	All Networks		1.3 −1.9 ms	0.095 −0.12 ms

Table 4.1: Processing times per event, T_{dl} and the number of Floating Point Operations (FLOPs) of various models that demonstrate the required efficiency, and the reformatting time per event T_f for 100 and 512 input data sizes respectively, for an MBED and a Raspberry Pi. * Memory limitations prevented this measurement.

4.5.2 Reliability and Power Consumption

Optimizing the network architecture and processing time are not the only factors to consider when implementing a DL network onto ARIANNA. Reliability in the harsh Antarctic climate must be considered as well as the limited power available in the remote location of Antarctica. The MBED was tested in the field for reliability in cold temperatures (averaging −60 °C) and meets the specification on power consumption, operating under one watt at about 0.3 watts. In contrast, the Raspberry Pi 3 is rated to −25 °C [7], and the Raspberry Pi Zero is rated to −20 °C [8]. Both will be studied for their operational temperatures and power consumption in this section.

The Raspberry Pis were stress tested under cold conditions in a thermal chamber, running from 20 °C to −60 °C. Both ran continuously with the DL filter for the hour it took to cool down to −60 °C, and then for an additional hour at this temperature. Next they were shut off at this temperature for 1-2 hours and turned back on. There were no issues turning the Raspberry Pi 3 on. During the Raspberry Pi Zero tests, it did not turn on again until −40 °C when using the wall power connector circled in orange in Fig. 4.8; when the board

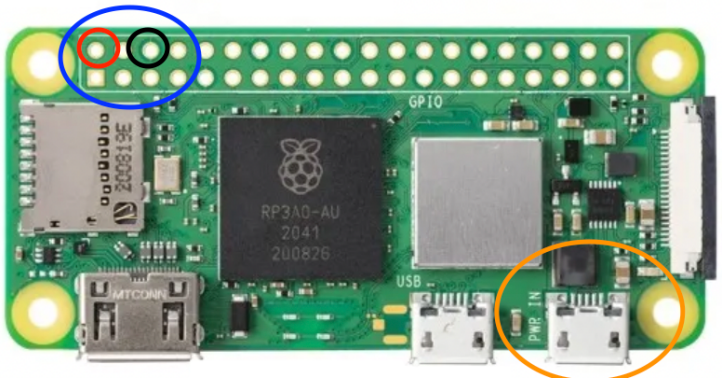

Figure 4.8: Image of a Raspberry Pi Zero with circled regions showing the two powering methods. The orange circle shows the connector using an adapter and wall power outlet. The blue circle gives the power pins for direct soldering onto a board. Both methods operate at 5 V. Modified from [8].

was powered using the pins circled in blue, the board operated without issues at −60 °C and during a cold reboot. This is likely due to either different placements on the board or the board circuitry. Once both Raspberry Pis were brought back to room temperature, they were still operational. If chosen as the optimal device in the future, the Raspberry Pi used would need to be tested further for long term operational reliability such as during temperature cycling.

Next the power consumption is compared between the two Raspberry Pis. A digital multimeter was connected in series to the power connector and the current was measured. The input voltage was at 5 V in both cases. Using $P = IV$, the power is given in Fig. 4.2. The Raspberry Pis have a significantly larger power consumption when running a DL filter compared to the MBED which operates around 0.3 watts when running and in idle; however, the MBED operates at a much slower processing rate comparatively. The Raspberry Pi Zero has a slight smaller running power consumption compared to the Raspberry Pi 3 since it has less components and features. The more probable choice for future upgrades is the Zero since it likely has all of the components needed to run the ARIANNA station, it can withstand cold temperatures, it has a sufficient processing rate, and it has a fairly low power consumption. The next generation of ARIANNA detectors are expected to generate more

power with better solar panels and wind turbine additions, so there will be flexibility in the power consumption limit of new devices.

Model	RP0 [W]	RP3 [W]
Running DL	1.0 - 1.1	1.4
Idling	0.56	0.66

Table 4.2: Power consumption values for the Raspberry Pi Zero (RP0) and compute module 3 (RP3). The values are given for both idling and while running the deep learning network. In both Raspberry Pis, the operating system used is the OS Desktop version.

Due to the extensive work required to implement the DL network on a new device, the first analysis for this work is done with the current MBED microcontroller. As discussed in the previous section, the 100 input sample CNN is chosen for the experimental verification study. Next, the template matching method is applied to the same simulated data set to compare with the 100 input sample CNN. Then, the 100 input sample CNN is implemented into the current MBED software for experimental verification.

4.6 Performance Verification

In this section, the performance of a DL filter is compared to commonly used template matching and verified by lab measurements. In addition, the ARIANNA hardware's computing performance is studied and in situ experimental cosmic ray data are classified by the filter to study signal performance.

4.6.1 Comparison to Template Matching

We compared the performance of a DL filter to a realistic template matching procedure using a single template, similar to what has been used in a previous analysis [29]. It is found that the DL method is typically faster and performs better.

A neutrino template is constructed by convolving a predicted Askaryan pulse with the antenna, amplifier and filter responses of the ARIANNA signal chain as already done in previous analyses [29]. A single template is used to minimize the computational costs, and also because of the observation that the template's overall shape is dominated by the detector response; variations in the predicted Askaryan pulse have only a small influence on the resulting templates (see e.g. [29, 21]). The plot of the general simulated template waveform is found in Fig. 4.9, and for this study, the amplifier response is added to this waveform without noise. Following the same data format as the 100 input sample CNN, the template was trimmed to 100 samples around the maximum absolute value of the waveform. This template was cross-correlated with the simulated signal and noise data sets, and the maximum absolute value of the cross-correlation is used as a measure for signalness, i.e., the output is a number between 0 (noise-like) and 1 (signal-like) as in the DL case. To compare the performance of the template and neural network method, the signal efficiency versus noise rejection factor is computed and compared to the CNN result which is presented in Fig. 4.10.

The CNN method is found to perform significantly better. At the benchmark value of 95% signal efficiency, the template method achieves little more than two orders-of-magnitude noise rejection. A possible explanation is the CNN identifies smaller 10x1 features within the training sets, which gives it more flexibility. Additionally, the CNN has five times the amount of features to extract compared to the template's single waveform feature. It is also important to compare the processing speed of the template matching technique to the CNN method. In estimating the processing speed for the template method, it was found that the FLOPs are roughly 29,900, or close to three times the amount of the 100 input sample CNN. Narrowing in further on the template signal pulse to 50 samples around the maximum of the waveform, the FLOPs are roughly 7,450; this is now a small amount than the 10,096 FLOPs of the 100 input sample CNN, but the efficiency of the template matching is still significantly lower. Therefore, the cross-correlation neutrino template matching method is

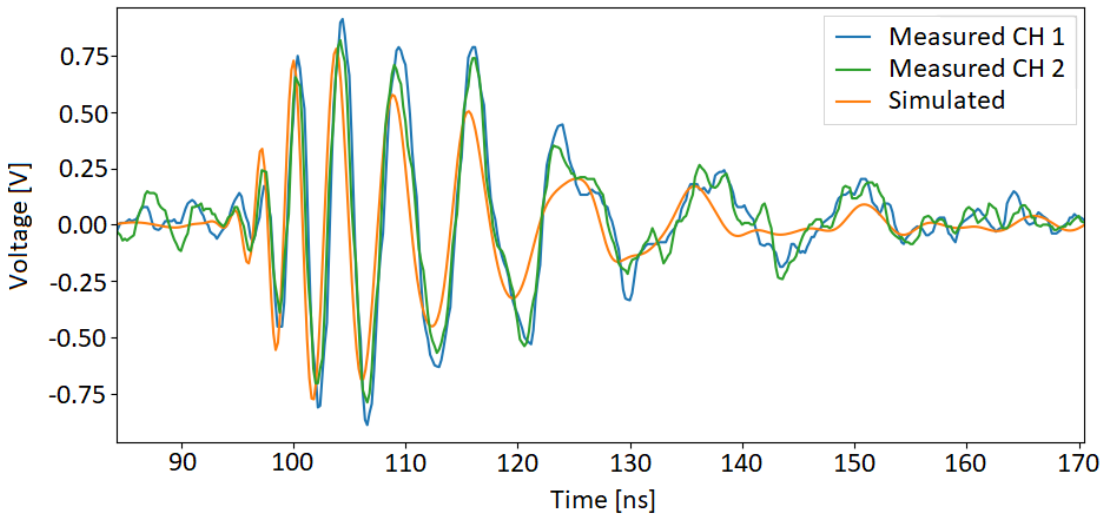

Figure 4.9: Comparison of the analog measured neutrino template signal being produced by the pulse generator and the simulated neutrino template. For details on the pulses see Sec. 4.6.2. Figure from Manuel Paul.

less efficient and (depending on the input data size) slower than the CNN technique.

4.6.2 Laboratory Verification

Next a laboratory study is performed to test the DL implementation and compare the network outputs of simulated and measured data. The DL filter uses the 100 input sample CNN described in Sec. 4.3. Data taking with LPDA antennas proves a challenge in the lab due to the lingering radio frequency noise present in the environment. Without a radio quiet space, one cannot replicate the environment of the Antarctic ice since the antennas would measure local radio sources which would bias the data. Thus, for in-lab tests of the DL implementation, an experimental "post LPDA antenna" radio neutrino pulse is created and injected into the ARIANNA hardware, bypassing the antenna, to verify the simulated results.

The expected neutrino template is generated with the standard simulation tool, NuRadioMC

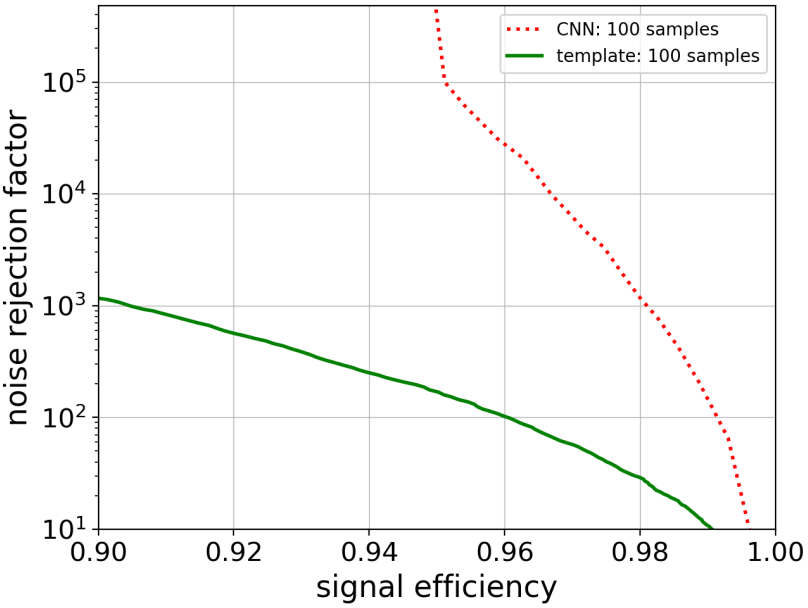

Figure 4.10: Signal efficiency versus noise rejection factor for the 100 input sample CNN and the 100 samples template matching method. The template matching technique uses a simulated neutrino template (with antenna and amplifier response and no noise) to perform a cross-correlation on the same simulated data used to train the 100 input sample CNN.

[66], which convolves the expected electric field at the detector with the LPDA antenna response. This neutrino template is then programmed into an Agilent Technologies 81160A arbitrary pulse generator to produce an analog version of the neutrino template as observed by the LPDA antenna. This waveform is then injected into the Series 300 amplifier of the ARIANNA pilot station, which adds realistic amplifier noise. By adjusting the input amplitude of the template, the SNR (signal-to-noise ratio) can be tuned to arbitrary values. Figure 4.11 shows a diagram of the experimental set up. The noisy signal is then routed to the input of the ARIANNA DAQ for data taking. Figure 4.9 gives a comparison of the simulated neutrino template to those produced by the analog pulse generator, known as measured neutrino template.

The DAQ consists of an SST chip, an FPGA, and an MBED microcontroller. Once digitized by the SST chip, the FPGA passes the digitized data (event) to the MBED, where the channel

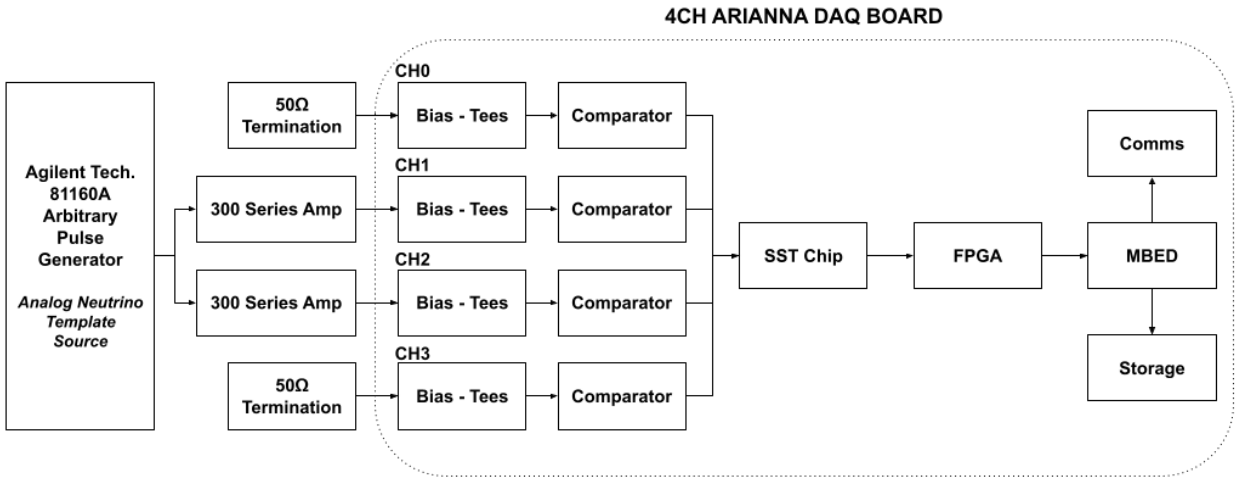

Figure 4.11: Diagram showing the set up for the collection of measured signal. The neutrino template was loaded onto the Agilent Technologies Arbitrary Pulse Generator and converted to an analog signal. The analog neutrino template was injected into a series 300 amplifier then into the ARIANNA DAQ board which contains triggering circuitry, an SST Chip, FPGA, and MBED. Figure from Manuel Paul.

with the largest signal is chosen and the data is run through the DL filter. Once finished, the filter gives the network output value of the event to determine its classification. Any event whose probability is below the cutoff would normally not be saved to memory. For testing purposes, all events are saved into memory along with the DL calculated probability. The DL filter performance on the ARIANNA hardware is also checked on a local computer which takes the digitized data and recalculates the probability. Both methods show equivalent results.

This setup allows for the recording of neutrino signal data sets and noise data sets. To record the noise data set, the neutrino signal generator is deactivated so that the ARIANNA DAQ only sees thermal noise amplified by the amplifiers. This test focuses on low SNR events where differentiating between signal and noise is most difficult. Therefore, the high/low trigger threshold is set to an SNR of 3.6 and a two of four coincidence logic between channels is required. The corresponding SNR distribution is shown in Fig. 4.12, where the SNR is calculated as the maximum amplitude of any of the channels divided by the RMS noise.

The measured noise distribution matches well to the SNR distribution of the simulated noise data set used in the previous sections.

Recording the neutrino signal data set is more challenging because at these low thresholds, a noise fluctuation might trigger the readout. So instead of a self-trigger, the ARIANNA DAQ is externally triggered via the trigger output of the pulse generator. The time delay is adjusted so that the signal pulse is at the correct location as expected from the ARIANNA high/low trigger. The amplitude of the signal pulse is adjusted to produce a low SNR distribution just above the trigger threshold. The neutrino template in Fig. 4.9 remains constant for all of the tests, but since the noise added by the amplifier follows a Gaussian distribution, the resulting SNR follows the same distribution. As seen in Fig. 4.12, the distribution is broader and shifted to a slightly higher mean than the noise data sets. This is expected because the interference of the extended signal pulse with noise gives several chances for an upward fluctuation, and the maximum value of any channel is chosen for the SNR estimate. The resulting SNR distribution of this low-amplitude simulated neutrino signal data set matches the experimental distribution.

To test the DL filter and verify that the simulated hardware components are comparable between measured and simulated data, a histogram of the network output values is plotted. This distribution is obtained by using the trained 100 input sample CNN to classify simulated and measured events. As seen in Fig. 4.13, the distribution of the simulated MC data set agrees well with the experimentally measured distributions. The small deviations are likely due to differences in SNR distributions and environmental effects such as strong radio pulses leaking into the cables. This gives us confidence that the MC simulation describes measured data and that the inferred noise rejection factor and signal efficiency are credible.

In this verification study, a CNN trained solely on simulated data is used to classify both simulated and measured data in Fig. 4.13. There is no potential artifact within the training process in this work because both the signal and noise sets are simulated data. In addition,

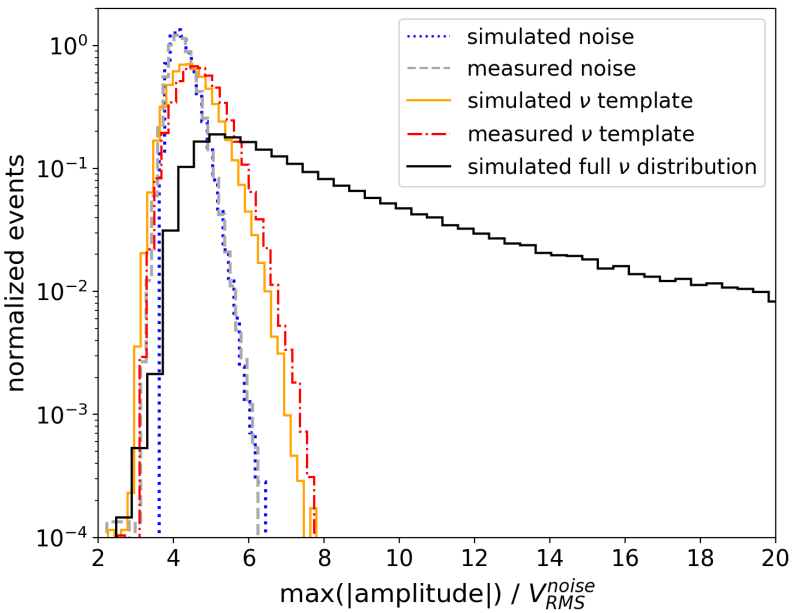

Figure 4.12: Histogram of the Signal to Noise Ratio (defined here as the ratio of the maximum absolute value of the maximum waveform to the noise RMS, $V_{\mathrm{RMS}}^{\mathrm{noise}}$) distributions for simulated thermal noise, measured thermal noise, simulated singular neutrino template, measured singular neutrino template, and simulated full neutrino spectrum. The template data have Gaussian noise added to the template to get the above distribution. The full neutrino distribution data set was used to train all of the networks in Sec. 4.3.

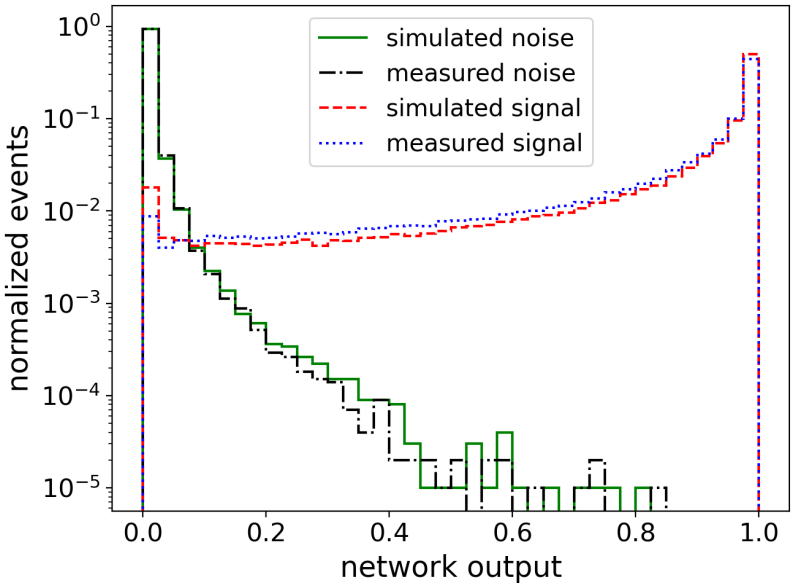

Figure 4.13: Histogram of the network output values of simulated and measured signal template and noise. The values are obtained from the network's output layer (a sigmoid activation function).

the measured signal and noise data sets contain the same thermal noise source (the amplifiers), which are well simulated and understood. Therefore there is not a potential artificial difference between the groups of data that the CNN would identify and use for data classification. In Chapter 5, the investigation of a potential artifact is conducted when training a network on a mixture of simulated and measured data.

4.6.3 ARIANNA Hardware Computing Speed for the CNN Filter

Use of the DL filter will increase the low-level trigger rate to 100 Hz. This significant change in trigger rate raises the question of whether or not the current hardware will be able to handle triggering effectively at that level. This section explores the hardware computing performance of the ARIANNA pilot station, which includes the waveform digitizers, the readout into the FPGA, the transmission from FPGA to the MBED for further processing, and the application of the DL filter. Taking the values from Table 4.1, and now including

$T_{\text{read}} = 7.3\,\text{ms}$ (which was experimentally obtained with the same procedure as T_{dl}) results in the MBED having an event readout time T_{min} of 12.3 ms. During the event readout time when the triggered event is being processed, the station is not taking data and therefore incurs deadtime.

This deadtime is the total processing time spent transferring and packaging the data and running on-board analysis programs. Deadtime is acquired by measuring the time delay between reset pulses, which is sent by the MBED when it is ready to receive a new event. To study the effect of T_{min} on operational livetime, the ARIANNA triggering system is simulated. First, the effect of the deadtime on the triggering system is simulated by writing code that takes an exponential distribution of times, δt, representing the time between incoming triggers with a mean of R_T^{-1}, where R_T is the noise trigger rate. When an event is being processed, no other triggers can be accepted. The simulation uses the distribution of times to calculate a fractional livetime as a function of R_T by calculating the average portion of time spent in deadtime due to R_T over a time T.

To confirm the simulation results for the fractional livetime as a function of noise trigger rate, the ARIANNA DAQ board is injected with signal-like voltage pulses periodically at a rate of 100 mHz with a high enough amplitude to always fulfill the trigger condition. At the same time, the trigger threshold is lowered after every trial to increase the noise trigger rate. For example, at a noise triggering rate of 0 Hz, 100% fractional livetime is expected since the injection rate of the periodic pulses is greater than the deadtime, allowing all injected pulses to be processed and saved to the SD card. In Fig. 4.14, the results of the experiment can be seen as the threshold is lowered, which in turn increases the noise rate and decreases the livetime. The measurement is described well by the corresponding simulation for $T_{\text{min}} = 12\,\text{ms}$.

At 100 Hz noise triggering rate, the livetime decreases below 50% which is unacceptably low. The CNN reaches a rejection factor of 10^5, but trigger rates are limited to 10 Hz or less to

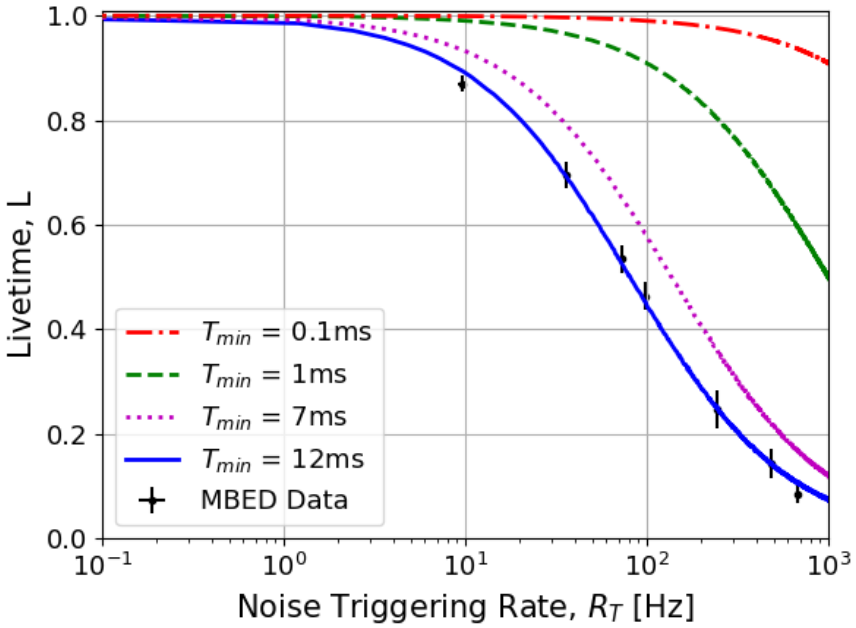

Figure 4.14: Livetime, L, as a function of Noise Trigger Rate, R_T, for three assumptions on the instrumental deadtime. Also plotted are data (black dots) from experimental verification study. Figure from Manuel Paul.

keep instrumental deadtime under 10%. This result suggests considering improvements to hardware, such as replacing the MBED with a faster processing device. A new device would need to run above 1 kHz to operate at 100 Hz with limited deadtime; events will trigger on average at 100 Hz, but some events will come in at a higher rate and the DL filter needs to be able to process each event in time.

4.6.4 Classification of Cosmic Ray Data

The ARIANNA detector stations are simultaneously sensitive to both cosmic rays that interact in the atmosphere and neutrinos: in addition to the downward facing antennas that are designed to detect neutrinos, the ARIANNA detector contains upward facing antennas to detect cosmic rays. Cosmic rays provide a calibration beam with similar radio frequency content and time variation as expected from neutrinos [42]. Moreover, both cosmic rays and

neutrinos have bipolar signals that are short compared to the response time of the antennas and their signals are significantly distinct from thermal fluctuations. Therefore, cosmic rays provide an opportunity to verify the flexibility of the CNN on another kind of signal event that should not be rejected as noise. Cosmic rays trigger a given station at a rate of only a few events per day, so it is important to keep as many cosmic ray events as possible.

A set of cosmic ray events from ARIANNA station 52, collected between November 2018 and March 2019, are used to test the 100 input sample CNN [114]. For this experimental data, the two of four coincidence logic trigger is set to a threshold of 4.4 SNR, and the event set has an expected purity of greater than 95%. When utilizing these cosmic ray data for testing purposes, the 100 input sample CNN trained on neutrino data and described in Sec. 4.3 is used. The network classifies 102 out of 104 cosmic ray events as signal, which is a greater value than neutrino efficiency in this study because the thresholds are slightly larger (4.4 SNR compared to 3.6 SNR in the simulation study). A network trained only with neutrinos still identifies cosmic rays adequately. It is likely that an even better cosmic ray efficiency could be obtained by including that data in the training set for a more targeted CNN filter.

4.7 Neural Network Hyperparameter Optimization

In Sec. 4.3.2, a brief study of the optimal network structure is described with the goal of balancing the amount of FLOPs with the efficiency of the neural network. The network detailed in that section contains input data of size 100 samples, one convolutional layer with five filters with 10x1 kernels each, stride of one, dropout of 0.5, max pooling with size 10x1, flattening, and a sigmoid activation. This is considered the baseline CNN. In this section, a more robust optimization is performed to quantify the efficiency of the model while individually varying the architecture parameters. The final optimized model is then compared to the baseline model in the previous sections. This work is in collaboration with

former undergraduate student Jiayi Liu and is also detailed in her senior thesis. We found that the baseline architecture components that contribute the most to FLOP increases are the input data size, stride, the amount of filters, and the kernel size. Dropout does not contribute and max pooling contributes very little. Therefore, only the four components listed above are investigated.

The first step in optimizing a neural network is to fix all variables except one. The kernel size is investigated first and all other variables are fixed, as seen on the top left of Fig. 4.15. There is a clear efficiency gain as the kernel size increases, and the increase is around 1-2% per doubling of the size between 5x1 and 20x1. Next, the amount of these kernels/filters is varied while keeping each kernel at the baseline of 10x1. The top right plot of Fig. 4.15 shows that the filter number improves more at lower values such as between 3 and 5 or 5 and 10. An unexpected discovery of this optimization training was that the 10 filters model performed better than the 15 filters model; potential explanations include these models not being trained to their full efficiency or perhaps in the larger case the increase of information made it more difficult for the model to generalize each filter feature. Nevertheless, the efficiency does not improve significantly beyond 10 filters. The input size was investigated next, specifically to determine the optimal small input data size. On the bottom left of Fig. 4.15, the input data size does not vary considerably when training on 50 input sample vs 200. However, there is more variation in efficiency when the input data sizes are increased more than these data sets with less than one channels worth of data. Lastly, the bottom right plot in Fig. 4.15 gives the variation in stride values. Stride can be set to modify the movement of the convolution; in most cases it is set to one which means the convolutions move over the data in one sample increments. If this value is increased, the network can perform convolutions with fewer iterations and therefore fewer FLOPs. Unfortunately, the efficiency is significantly lower for an increased stride above one, so this parameter should not be adjusted further.

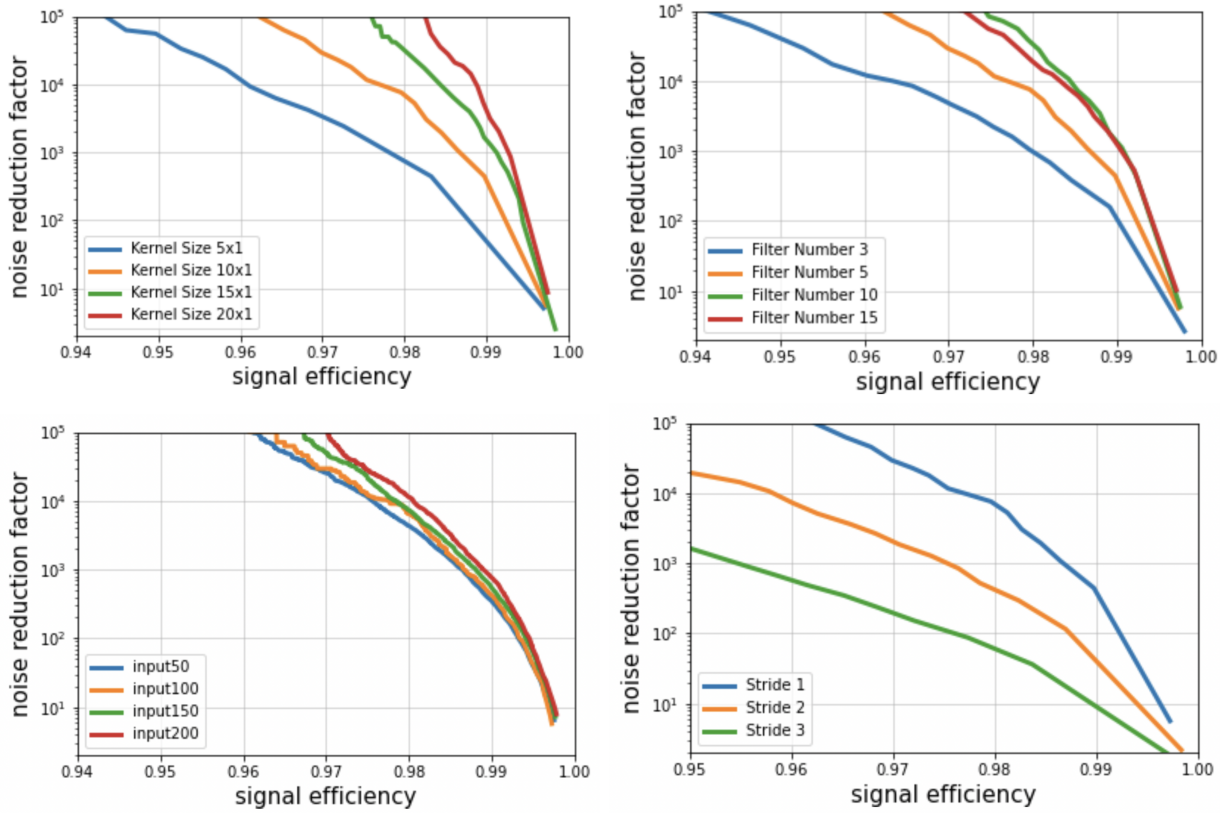

Figure 4.15: The noise reduction factor versus signal efficiency for four separate scans over various hyperparameters. The plots show the variation between different kernel sizes (top left), kernel amounts (top right), input data sizes (bottom left), and stride values (bottom right).

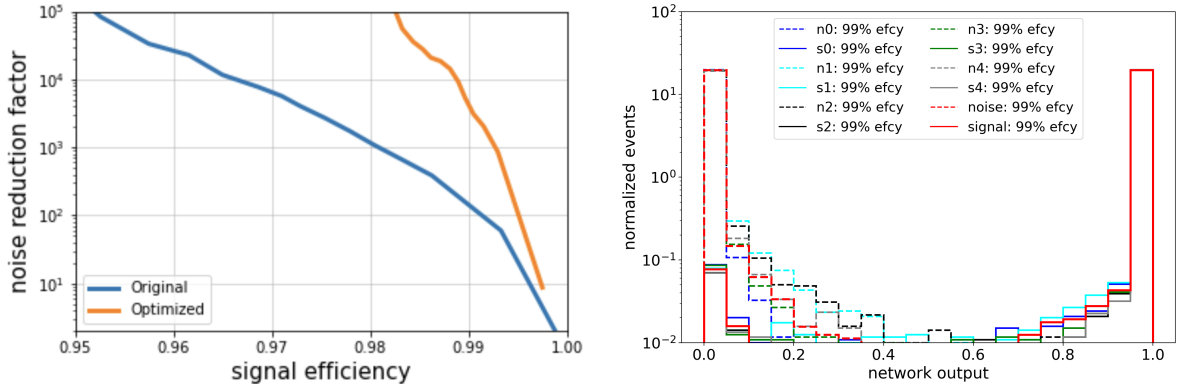

Figure 4.16: On the left, the noise reduction factor versus signal efficiency comparison between the original model from Sec. 4.3.2 and the optimized model found in this section. On the right, a histogram of the network outputs for a five-fold cross validation (n0-n4 and s0-s4). The signal and noise distributions for a network trained on the full data set are in red. The percentages describe the efficiency with a network output cut of 0.5 for each distribution.

With the results above and the goal of keeping a relatively small network size, one model is suggested for a future implementation of the DL trigger. This model has an input data size of 100 samples, a one layer convolution with five 20x1 kernels, a stride of one, a max pooling of 20x1, and trained with a learning rate of 0.05. The network is shown on the left side of Fig. 4.16 and is compared to the efficiency curve of the baseline CNN model. The network output of a five-fold cross validation of this optimized model is shown on the right side of Fig. 4.16. This new model requires 17,046 FLOPs compared to the baseline CNN's 10,096 FLOPs. For a 1.68x increase in FLOPs, the signal efficiency increases by about 3%. at 10^5 noise reduction factor.

4.8 Network Performance with Continuous Waves

One potential issue with neural networks is that they might not be flexible when receiving data that vary from training data sets. Mitigating this concern would require the data a network is trained on (simulated data in this case) to be a complete representation of

experimental data. For instance, one such variation that has been seen in experimental ARIANNA data is anthropogenic noise and in particular noise due to electronics signal leakage. Refer to Sec. 2.2 for more details. Anthropogenic noise events such as those generated from electronics are typically continuous waves. Continuous waves (CW's) have a constant amplitude and frequency, and in this study will be modeled as a sine wave.

In this work, the previously trained 100 sample baseline CNN from Sec. 4.3.2 will be used to classify altered simulated thermal noise data to test the network's flexibility. The signal testing set, which consists of simulated neutrinos, will remain untouched. The altered data consist of the same thermal noise testing data set from Sec. 4.3.1 with the addition of separate CW's with different amplitude and frequency values. The goal is to determine the effect of additional noise sources on the previously trained neural network. At a trigger threshold of 3.6 x $V_{\mathrm{RMS}}^{\mathrm{noise}}$ and a $V_{\mathrm{RMS}}^{\mathrm{noise}}$=10 mV, the waveforms have a high/low peak above 36 mV/−36 mV. Thus, near these values the addition of a CW would severely effect the input waveform. Therefore, the CW amplitudes used are 2 mV, 5 mV, 10 mV, and 20 mV. The frequency baseline value is set to 200 MHz. The bottom four plots in Fig. 4.17 show how a typical thermal noise event (in red) is transformed into a new waveform (in blue) with the addition of a CW (in black) for different amplitudes. As demonstrated, the effects on the thermal noise event are not significant until the addition of the 10 mV CW, as the CW amplitude is now similar to the RMS noise level. The top plot of Fig. 4.17 gives the noise rejection versus signal efficiency for these various thermal noise plus CW testing data sets.

In addition, the frequency of the continuous wave can be studied. Referring to Fig. 2.10 which outlines some of the anthropogenic noise sources and their frequencies, spikes around 133 MHz and 225 MHz are seen. The spikes are mainly due to air traffic control radio signals, VHD repeaters, and electronics read-out. These two frequencies represent realistic values for CW noise and so their efficiency curves are shown in Fig. 4.18. Compared to the baseline CW frequency value at 200 MHz, a 133 MHz lower frequency CW addition is more difficult

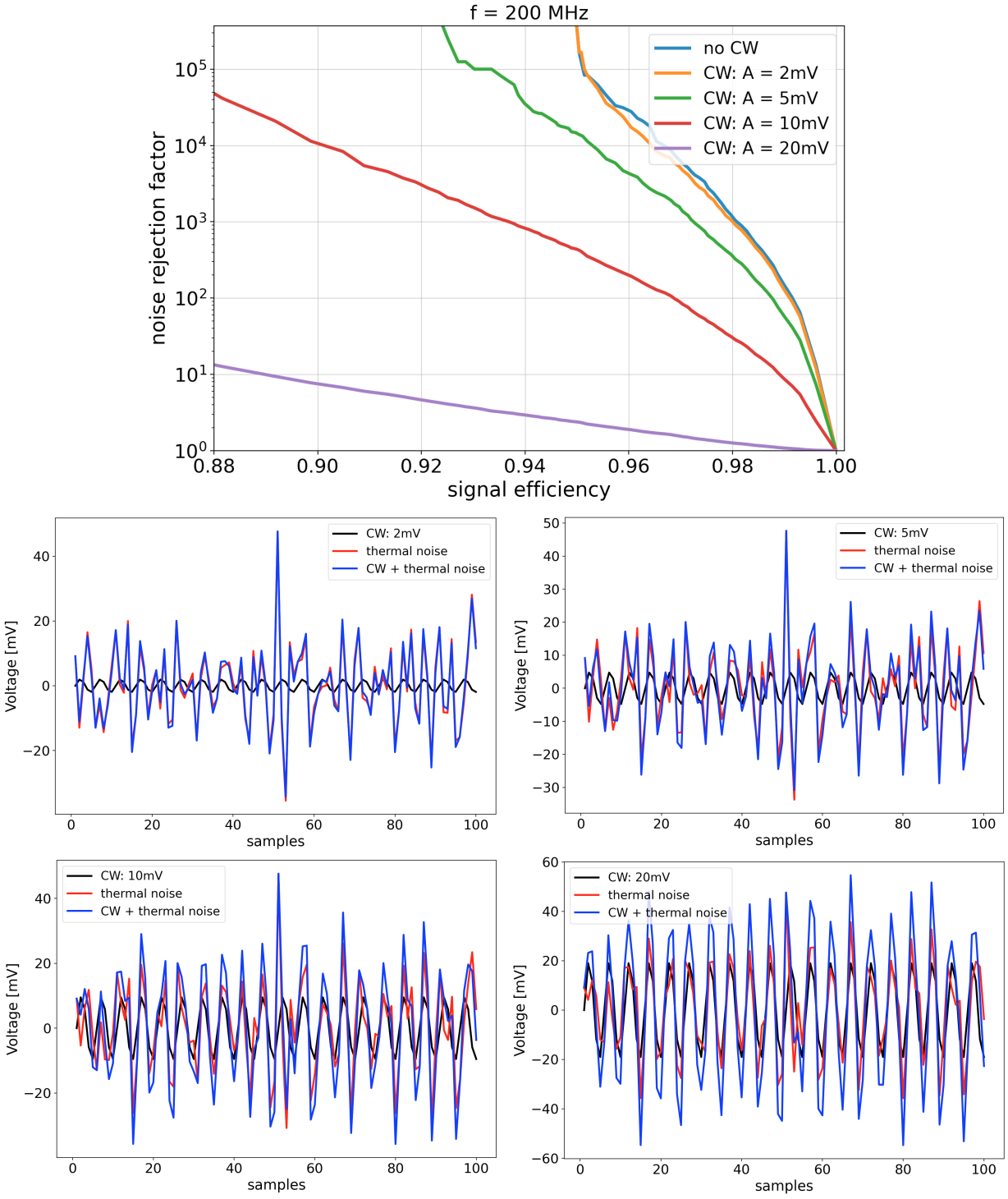

Figure 4.17: The noise rejection factor versus signal efficiency (top) for a regular signal testing set, and a noise testing set with the addition of a 200 MHz CW of various amplitudes. The bottom four plots visually show the influence of CW additions to the final waveform fed into the trained 100 sample CNN (shown in blue). The CW amplitudes vary from 2 mV (mid left), 5 mV (mid right), 10 mV (bottom left), and 20 mV (bottom right).

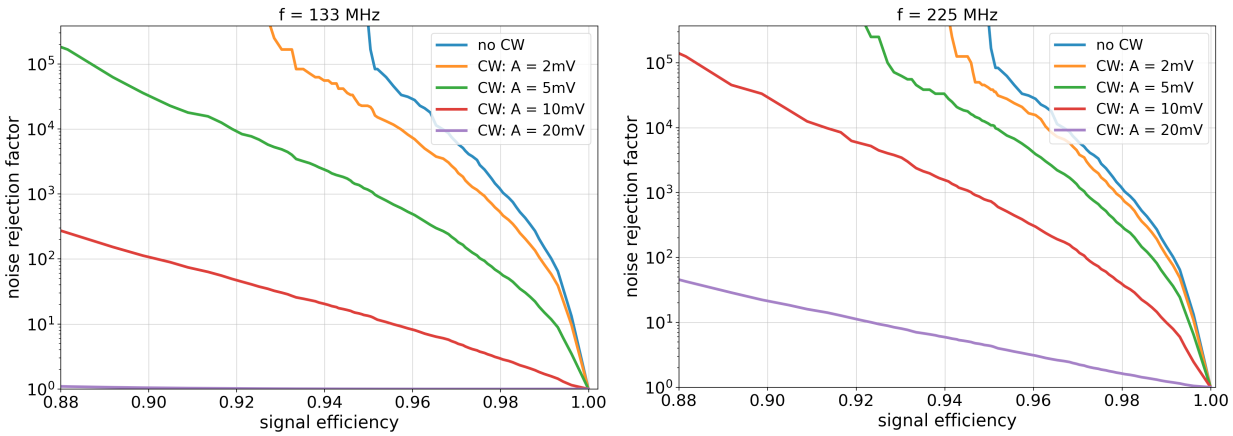

Figure 4.18: The noise rejection factor versus signal efficiency for a regular signal testing set, and a noise testing set with the addition of a 133 MHz CW (left) and a 225 MHz CW (right).

for the 100 sample CNN to classify. This is particularly true with the 20 mV amplitude CW curve. Comparing the baseline CW frequency of 200 MHz to a higher frequency CW of 225 MHz, the efficiencies increase slightly with frequency but remain similar. Therefore, there are limits to the efficiency improvements as the CW frequency increases.

There are already tools in place to handle noise events that are sharply peaked at one frequency (see Sec. 2.1.5 for the L1 trigger). Nevertheless, this work is promising and shows that the 100 sample CNN could be used in the place of the L1 trigger for small CW amplitudes. It also shows that this CNN is flexible enough to classify noise data that have been altered by CW's; this is important because DL networks are often hard to interpret. This flexibility shows that the network is not overfitting on the training data. The goal of ARIANNA is to make electronic components as radio quiet as possible, so perhaps in the future some of these noise sources will be lessened. However, sources such as air traffic control will be unavoidable. Further studies will need to be done on the robustness of the 100 input sample CNN but this is a starting point. Next a summary of this chapter is given and future studies/plans for the DL filter will be outlined.

4.9 Summary, Discussion, and Future Plans

Due to low neutrino flux at extremely-high energies, the sensitivity of the detector to a flux of high-energy neutrinos is limited by statistics. Probing new parameter spaces is made possible by implementing DL techniques to increase the sensitivity of the ARIANNA detector. A small convolutional neural network (CNN) was implemented on the ARIANNA MBED microcontroller to discriminate between thermal noise fluctuations and neutrino signal. It was shown that CNN filters are much more accurate and computationally faster than cross-correlation methods (template matching) in distinguishing between thermal events and neutrino signals. CNN filters were also shown to be flexible when investigating network interpretability. Only one thermal event in every 10^5 thermal triggers was misidentified by the CNN, while 95% of the neutrino signal was correctly identified. Consequently, the trigger rate can be increased by five orders-of-magnitude while transmitting at a rate of 0.3 mHz over the Iridium communication network. This results in an increase in neutrino sensitivity of 40% at 10^{18} eV and up to a factor of two at lower energies. The simulation study was verified by lab measurements which found a strong correlation between the measured and simulated distributions of both neutrino and thermal noise events. The DL processing rate of the MBED was found to be 270 Hz whereas a Raspberry Pi achieves 2.5 kHz. In addition, a group of experimental cosmic rays were run through the 100 input sample CNN with 102 out of 104 events classified as signal. This work concluded with an investigation on hyperparameter tuning and studies of the robustness of the CNN with continuous wave additions to the input data.

At the moment, the processing speed of the ARIANNA hardware is the limiting factor in obtaining an average event classification rate of 100 Hz. The low-level trigger rate is restricted to less than 10 Hz to avoid greater than 10% deadtime. In the future, several improvements to the ARIANNA hardware are being considered. First, the ARIANNA hardware that sends the digitized data from the FPGA to the microprocessor can be parallelized to decrease the

time to readout an event, which would provide the opportunity to trigger the detector at even higher rates. Second, more capable computing on the ARIANNA station through improved electronics will be studied. Third, if the space exists on a device, a buffer could be added to hold 10 events at a time before processing them to limit the deadtime of events coming in close together.

The current generation of ARIANNA hardware is now more than 10 years old, and many recent microcomputer systems offer more performance at comparable power consumption. An increase in computer speed allows more complex CNN architectures with an equivalent improvement in the trade off between neutrino signal efficiency and background noise rejection. The combination of these changes would increase the sensitivity of the ARIANNA detector even if communication transfer rates remained the same. However, the next generation of Iridium satellites, Iridium Next [6], has been recently deployed and has the potential to increase transfer rates by many orders-of-magnitude relative to the SBD message transfer system currently used by ARIANNA.

Reliance on DL filters may lead to misleading results when incoming data deviates from training data, so these filters must be carefully evaluated by laboratory and field studies. The lab measurements described in this work are encouraging and suggest that the simulations describe reality. The next stage of confirmation studies will follow a similar approach as was used to validate the simulation studies of the sensitivity of the ARIANNA detector. After modifying the software in the data acquisition system to include the CNN filter, we plan to (1) use a variety of radio transmitters within a preexisting borehole drilled to a depth of 1700 m at the South Pole (the SPice core [105]), to confirm signal efficiency at various thermal noise trigger rates; and (2) compare the rate and physical properties of cosmic ray events to data samples collected without the filter.

In addition to the current hardware constraints, there are limitations on the simulation tools as well. This analysis was able to use simulated data to achieve five orders-of-magnitude

thermal noise rejection, but the simulations do not yet include a number of real world effects that can create small deviations in the data. One example is in the ice model, which does not include the layering structure of the ice density (stratification) when signals propagate through it. The incompleteness of the simulations can affect the in-field performance. However, these are known limitations that will be explored in the next round of simulations.

Chapter 5

Offline Deep Learning Analysis on ARIANNA Data

The primary focus of the ARIANNA experiment is to search for EHE (extremely-high energy) neutrinos using detectors buried in the ice of Antarctica that are outfitted with downward facing log-periodic dipole antennas (LPDAs) and dipole antennas; however, a few detector stations also have upward facing LPDAs to optimize for cosmic ray detection and aid in the study of detector capabilities, background noise rejection, and calibration techniques. ARIANNA has conducted many projects that explored methods for improving detector and data analysis techniques. A few studies include improving the detector sensitivity of ARIANNA at the trigger level by using a restricted bandwidth trigger [64] and removing thermal noise events in real time by the use of a deep learning filter (refer to [33] and Chapter 4). Consequently, the trigger rate on the ARIANNA station could likely be increased by a factor of 10^5 but the transmission rate over the Iridium satellite link does not change. In effect, the trigger threshold was lowered because the on-station trigger rate increased.

It is also possible to improve the sensitivity at the analysis level (or "offline analysis") by

increasing the fraction of neutrinos at the trigger level that remain after all analysis criteria are applied to eliminate the background events. The ARIANNA collaboration described the analysis tools they developed for a search of high energy neutrinos in 4.5 years of operation [29]. That study used a template matching procedure to search for neutrino candidates in the data and showed that all background events collected during an integrated livetime of approximately 10 station-years were eliminated while keeping about 80% of the neutrinos at trigger level. Though this work was encouraging, we caution that the integrated livetime was only about 1% of the requirement for future generation arrays such as IceCube-Gen2. New analysis tools are required to keep 80% or more of the high energy neutrinos when the total background noise increases by a factor of 100. The analysis used in [29] relied on only four downward LPDAs of information, but several of the ARIANNA stations were augmented with four additional antenna channels: (1) upward facing LPDAs to identify cosmic rays, wind, and anthropogenic backgrounds that propagate through the air. (2) dipoles buried at 5 m and 10 m depths to observe the distinct signature of a double pulse from a neutrino event (the second pulse is due to reflections off the air-firn surface and is only produced by radio emission from within the ice). By incorporating the additional information from these antennas in this analysis, the background rejection and signal efficiency of the more traditional analysis tools can be greatly improved.

In Chapter 4, it was concluded that relatively simple networks could differentiate thermal events from neutrino signal with high reliability. More than 95% of the neutrinos remained at analysis level, while only 1 in 10^5 thermal events passed the filter. In addition, it was shown that DL (deep learning) networks significantly outperform a waveform correlation analysis based on template matching. Performing a cross-correlation between data and expected neutrino waveforms (or templates) is one of the key steps in the traditional analysis method. Also note that the networks of Chapter 4 were restricted to the information provided by the four downward facing LPDAs.

Since DL tools in Chapter 4 outperformed the correlation analysis, we now investigate if DL-based tools could outperform the more conventional analysis on data collected from an eight channel station. This chapter demonstrates that a DL-based analysis can improve the background rejection by a factor of 25 when normalized to comparable signal efficiency. Although processing rates are less of a concern in offline analyses than considered in the prior chapter, there are several new challenges that were not encountered in the real-time filter study of Chapter 4. First, an offline analysis must reject all backgrounds, not just the relatively common thermal fluctuations. For example, pulsed radio emission is generated during periods of high wind on the surface of polar ice sheets, see Sec. 2.2. There is also self-induced noise from ARIANNA electronics that is responsible for charging the Lithium Iron Phosphate batteries (known as the BMU, or battery management unit). Large currents are switched on and off, which creates pulsed noise that infrequently triggers the detector. Second, most non-thermal backgrounds are not well-described theoretically, so no computer models exist to generate a training set of events. Therefore, we chose to use the acquired data by the ARIANNA stations to provide both training and testing sets of background events. Third, the amplifier characteristics associated with the gain as a function of frequency are not identical for all ARIANNA stations, so thermal event samples were developed for each amplifier class. There have been three amplifier designs used by the ARIANNA collaboration, and future arrays will likely employ a fourth variation. Finally, the experimental thresholds for the HRA stations were set to 4.5 - 5 times $V_{\mathrm{RMS}}^{\mathrm{noise}}$, higher than those considered in Chapter 4. Consequently, the thermal events that trigger the ARIANNA station show larger amplitude variations, making it more difficult for the DL analysis to distinguish them from neutrino signal. Fortunately, the rate of thermal events is much smaller so the required rejection rate is less, which partially compensates for the additional challenges imposed by the larger amplitudes.

After the DL-based study on neutrino search efficiency in this section, the offline analysis results are cross checked by replacing the neutrino signal with a known sample of cosmic rays

[34] acquired by ARIANNA station 52. In particular, an investigation is done to determine whether an artifact (that is, a non-physical difference between the background training set and signal training set) could potentially affect the DL results. The experimental signature for cosmic rays that interact in the atmosphere are well modeled in the upward facing LPDAs by simulation tools in NuRadioMC [66], so both the simulated and experimental cosmic ray signals can be used to test the DL analysis.

This work is organized as follows: 1) the relevant ARIANNA stations will be described in more detail; 2) DL models and data sets used in this study will be defined; 3) the network output efficiency for experimental ARIANNA data will be determined using a network trained on a mixture of simulated and experimental data; and 4) a similar study is done with cosmic rays to confirm the neutrino efficiency results from the third step above. The chapter concludes with a summary on the work done and a look forward to future projects.

5.1 Neutrino and Cosmic Ray Stations

The ARIANNA experiment is primarily located on the Ross Ice Shelf in Moore's Bay, although until recently it also included two pilot stations located at the South Pole. The different locations produce similar data if the main hardware components do not vary between stations. Detector configurations include eight antennas with six LPDAs, which are directional and primarily sensitive to horizontal polarization, and two fat-dipole antennas, oriented vertically and therefore sensitive to vertical polarization. The antennas are set in two different configurations; stations searching for neutrinos have four downward facing LPDAs, two upward facing LPDAs (to veto high energy cosmic rays events that may look like neutrinos), and two dipoles. Stations configured to observe cosmic rays have four upward facing LPDAs, two downward facing LPDAs, and two dipoles. The LPDAs are directional, with maximum sensitivity for signals propagating toward the nose of the triangular shaped

antenna. Conversely, LPDAs are not as sensitive to signals with arrival directions in the rear of the antenna, the back lobe. As a result, antenna orientation is important to trigger on the desired particle. For more details on the ARIANNA detector stations, refer to Chapter 2.

The study discussed in this chapter relies on two specific ARIANNA stations: station 61 and station 52. Station 61, with four downward facing LPDAs to operate in a mode that searches for neutrino events. Station 52, with four upward facing LPDAs mostly operated in a mode to observe cosmic rays. All LPDAs of both stations are buried in the firn to a depth that covers all metal tines. Both stations have the same ARIANNA amplifiers, called 300 series, and due to the fact that the amplifier noise dominates the antenna noise, the observed noise does not depend significantly on the different local temperature of the two sites ($-50\,\mathrm{C}$ at South Pole, $-30\,\mathrm{C}$ at Moore's Bay). We measured the $V_{\mathrm{RMS}}^{\mathrm{noise}}$ around $10\,\mathrm{mV}$ for all channels of both ARIANNA stations. These two stations have identical DAQ (data acquisition) boards which include the same software, amplifier type, and amount of antenna channels. The work in this section utilizes these similarities to show that if there is no artifact in cosmic rays station 52 data, then there is likely no artifact in neutrino station 61 data. This idea originates from the fact that these two stations have similar components in their DAQ systems and so there are less chances for differences between both experimental and simulated data sets.

5.2 Deep Learning Architecture and Data Sets

There are a total of six data sets utilized for this DL analysis. For station 61, the neutrino station, there are experimental data, simulated thermal noise (used for both studies), and simulated neutrino signal. For station 52, the cosmic ray station, there are experimental data, simulated cosmic ray signal, and experimental tagged cosmic rays. These data sets will be discussed in more detail in this section and are also summarized in Table 5.1. Before

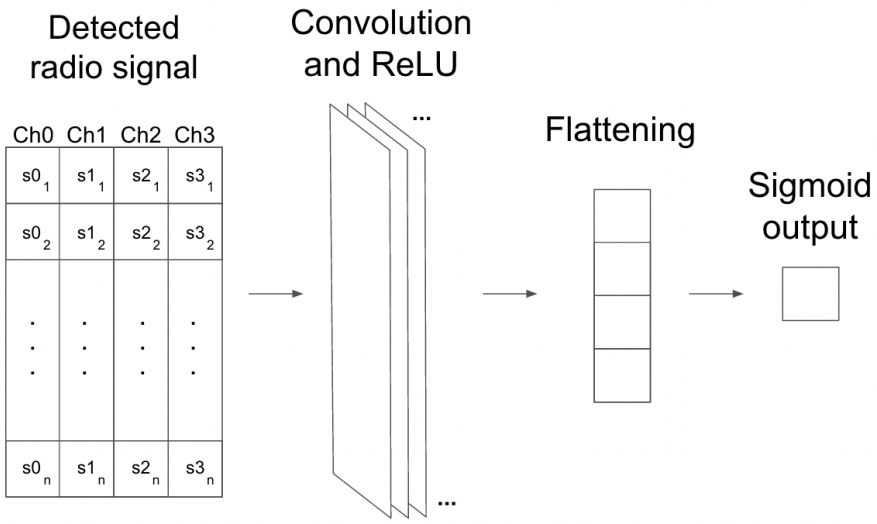

Figure 5.1: Baseline architecture of a convolutional neural network (CNN). First is a two-dimensional convolution with ReLU activation, then a flattening layer where the data are reshaped, and lastly a sigmoid activation in the output layer.

defining these data sets, the DL baseline architecture used for this study will be discussed.

5.2.1 Deep Learning Model Architecture

The DL platform used in this study for training, validation, and testing is called Keras. Refer to Appendix A.1 for resources on recreating this work. The baseline architecture for this study is shown in Fig. 5.1. Each channel (antenna) contains 256 samples per event, and four channels are stacked together to produce an input array of 256x4. The input data go through a convolution with 10 10x4 kernels and a ReLU activation. Next, a flattening step reshapes the data to a one-dimensional array and lastly, the output layer is processed through a sigmoid activation to normalize classification values between zero and one. This small network produced good results and adding more complexity did not improve the network efficiency.

5.2.2 Neutrino Station 61

The experimental data for station 61 were obtained between December 2018 and January 2021. Since this ARIANNA station was powered solely by solar panels, it did not operate during the Antarctic winter months when the sun is located below the horizon (March through September). Events that occurred during periods of human activity at the detector site such as radio emission during weather balloon launches or radio communication to incoming or outbound aircraft, were removed; these dates are known and events collected during these time frames were removed from the data set. Due to the diffuse neutrino flux limits set by IceCube and the duration of data taking from this one station, no neutrinos are expected in the data set. Thus, the data comprise primarily of thermal noise events, wind generated noise events [22], and noise generated from the ARIANNA electronics. It is possible that some events are anthropogenically produced by human activity at or near the South Pole Station (from generators, radio communication, spark plugs, etc.). In total, there are 74,530 triggered events in this data set, which is called station 61 experimental data, or E-BG61.

The simulated neutrino signal data set is generated with NuRadioMC, which simulates a representative set of expected neutrino events for the ARIANNA detector. The randomly distributed events follow a widely-discussed GZK [24, 110] neutrino flux model, with neutrino energies from $10^{17}eV - 10^{19.5}eV$. There is also a weight cut of 10^{-5} performed so events with weights below this value are removed; since simulated neutrinos are generated uniformly in all directions, the weight cut removes lower probability neutrino events (corresponding to larger incoming angles below the horizon) that would not likely be measurable by ARIANNA. Next, these generated events are propagated to the detector through a simulated model of the ice. Then, the ARIANNA station 61 detector is simulated with NuRadioReco [65]; the resulting neutrino radio signals are simulated in all eight channels by convolving the electric field pulses with the antenna response. Noise is added to the neutrino pulses and

the resultant signals are approximated with an 80 MHz to 500 MHz band-pass filter. The station is triggered if the signal pulse crosses a high and low threshold of 4.4 times the RMS noise, $V_{\mathrm{RMS}}^{\mathrm{noise}}$ within the two of four trigger logic criteria outlined in Sec. 2.1.5. In total, there are 10,606 triggered events in this set, called station 61 simulated neutrinos, or S-NU61.

The simulated thermal noise data set is similarly generated using NuRadioMC with the same trigger requirements and a threshold of 4.4 times the $V_{\mathrm{RMS}}^{\mathrm{noise}}$. In total, there are 50,000 triggered events in this set, called station 61 simulated thermal noise, or S-BG61.

5.2.3 Cosmic Ray Station 52

The experimental data from station 52, located at sea level at Moore's Bay, Antarctica, were collected from December 2018 to March 2019. As mentioned earlier, the primary purpose of this ARIANNA station was to measure cosmic rays at a rate of 1-2 events per day. Fortunately, a previous analysis on this exact data sample [34, 114] identified Cosmic Rays with high purity. In contrast, the experimental data set from neutrino station 61 was highly unlikely to contain true neutrino events.

The "tagged" set of cosmic ray signal events are removed from the background training set. Therefore, this data set is expected to contain mainly background noise data, although it is possible that this data set does contain some cosmic rays that were not tagged and removed. As with station 61, the station 52 experimental data contain thermal noise, wind generated noise, and noise generated from ARIANNA electronics. In total there are 97,955 events in this data set, which is called station 52 experimental data, or E-BG52.

Another data set used in this analysis includes only the extracted experimental cosmic ray signal mentioned above. Obtained from [34, 114], this data set was gathered by making various cuts on the trigger rate, correlation, and zenith. In total, there are 85 events in this

set, called tagged cosmic rays, or E-CR52.

The simulated cosmic ray signal data set is created using the CoREAS software. CoREAS is a Monte Carlo codebase for simulating the radio emission from extensive air showers [76]. The CoREAS code simulates cosmic rays over many different arrival directions and energies, but for this study, the events are re-weighted to match the expected energy and arrival direction distribution of the cosmic ray flux. Then, a detector simulation is performed where thermal noise is added to the cosmic ray signal and a 80 MHz to 500 MHz band-pass filter is applied. Only cosmic ray signal events that cross a high and low threshold of 4.4 times V_{RMS}^{noise} within the two of four trigger logic criteria (see Sec. 2.1.5) are used. The simulation is most accurate for the four upward LPDAs as shown in Fig. 5.2. The bottom four plots have large variations in the amplitude between the simulated and experimental cosmic rays; these amplitudes differ by almost a factor of two at certain frequencies, so the two downward facing LPDAs and dipole antennas are not used in this data set. In total, 9,627 triggered events are contained in this set, called station 52 simulated cosmic rays, or S-CR52.

For all of the simulated data sets, enough events were generated to give sufficient results. In general, the amount of events needed to train a model varies depending on the data sets and goals. There is typically a trade off between the amount of (simulated) events needed for training/testing and the computational resources required to produce them. In most analyses within this work, at least 10,000 signal and noise events are needed so this is a common starting point.

5.2.4 Comparison Between Different Event Types

The various data sets outlined above contain many different types of events. To get a better idea of each one's distinguishing features, their waveforms are shown in Fig. 5.3 and discussed in more detail here. Events 1 and 2 are taken from the station 61 configuration, so the two

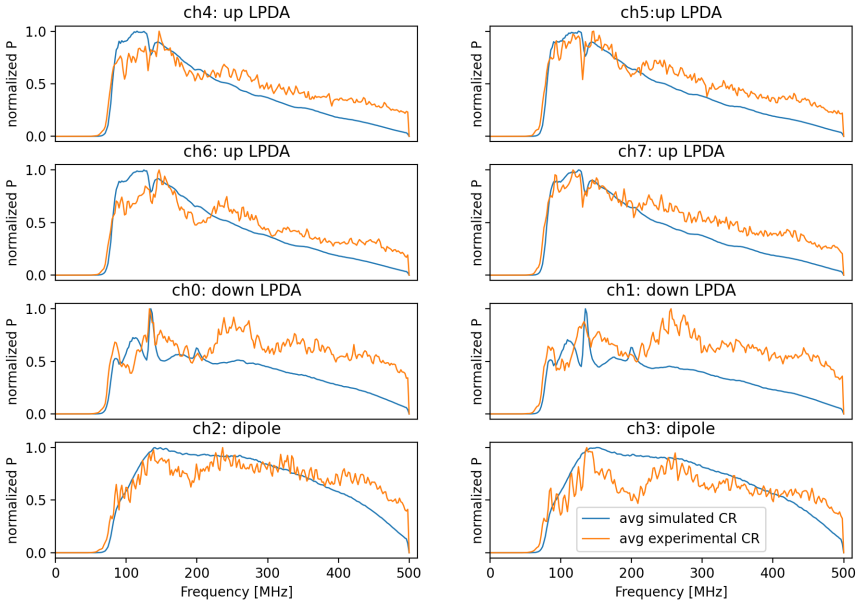

Figure 5.2: The average FFT distributions for the simulated cosmic rays and the 85 experimental cosmic rays.

	Station 61	Station 52
Simulated Signal	S-NU61: 10,606	S-CR52: 9,627
Simulated Thermal Noise	S-BG61: 50,000	-
Experimental Data	E-BG61: 74,530	E-BG52: 97,955
Tagged Cosmic Rays	-	E-CR52: 85

Table 5.1: Each data set is abbreviated when described in this analysis. The first letter denotes simulated or experimental data. The two letters after the dash are NU for neutrino, CR for cosmic ray, and BG for background noise data. The last two numbers are the station ID. Also given is the amount of events in each data set.

of four triggering logic is used on the downward facing antennas. The first four channels 0-3 are the downward facing LPDAs, channels 5 and 7 are upward facing LPDAs, and channels 4 and 6 are dipole antennas. Event 1 is a simulated neutrino with the largest signal in parallel channels 1 and 3. Event 2 is simulated thermal noise that was triggered from the same parallel channels. At thresholds of 4.4 SNR (signal-to-noise ratio), the difference in shape between neutrino signal and thermal noise is significant. Comparing events 1 and 2, neutrino signal has a higher frequency component in the beginning of the pulse and lower frequency components towards the end; thermal noise has a narrow pulse that fluctuates between high and low quickly, as these are random fluctuation. These two types of events are typically straightforward to distinguish from one another.

The events 3, 4, 5, and 6 are taken from the station 52 configuration, which relies on the two of four triggering logic in the upward facing antennas. The channels 0 and 1 are downward facing LPDAs, channels 2 and 3 are dipole antennas, and the last four channels are upward facing LPDAs. Event 3 is a lower amplitude simulated cosmic ray signal whereas event 4 is a larger amplitude experimental cosmic ray event from one of the 85 tagged cosmic rays. Both signal pulses are similar, and more elongated compared to the neutrino signal in event 1. The signal pulse shape differences between neutrinos and cosmic rays are expected due to the two distinct shower profiles; cosmic ray pulses are typically weighted towards lower frequencies whereas neutrinos have more high frequencies contributions in the beginning of the pulse.

Events 5 and 6 are the two other dominant background noise sources besides thermal noise. These types of events occur in both station 61 and 52 with the latter being shown in Fig. 5.3. Event 5 is from the battery management unit electronics with waveforms that are periodic and sinusoidal in nature. These event types will not be present in the improved ARIANNA design. Event 6 is a wind generated event (see [22] for more details), which will likely remain a source of noise in future ARIANNA detectors. These wind events are not a significant

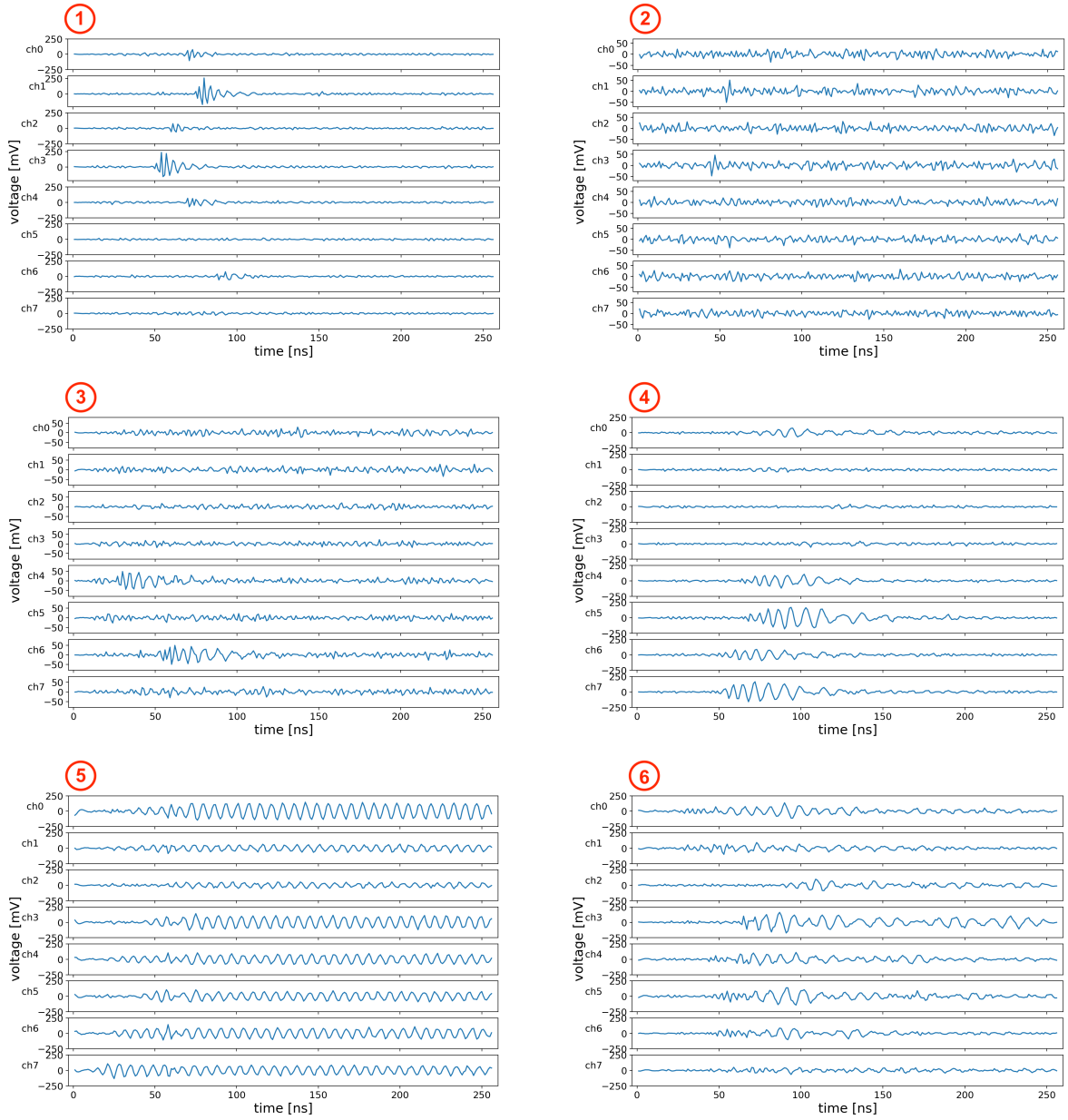

Figure 5.3: A collection of example event types for station 61 and 52 configured data. Event 1 is a simulated neutrino signal for station 61. Event 2 is simulated thermal noise for station 61. Event 3 is a simulated cosmic ray signal for station 52. Event 4 is an experimental tagged cosmic ray from station 52. Event 5 is electronics related noise from the battery management unit of station 52. Event 6 is wind related noise from station 52.

problem for neutrino detectors as their waveforms' shapes are quite different. However, wind event waveforms are similar to those of cosmic ray signal, both with more elongated pulse shapes; a comparison between the two can be seen in Fig. 5.3, events 4 and 6. Current template matching procedures work well to distinguish these two event types for many cosmic rays, but it is still difficult to extract all of the existing experimental cosmic rays in part due to this likeness.

5.2.5 Comparison Between Stations

Stations 61 and 52 have different configurations since they are searching for different particles. The difference between the types of events which trigger upward facing LPDAs as opposed to downward facing LPDAs is significant. For both configurations, the most prominent noise source is thermal noise fluctuations in the ice, but upward facing LPDAs also have galactic radio noise contributions [104]. In addition, the four upward facing LPDAs in station 52 are more susceptible to wind generated noise events since they are likely generated at the ice surface; thus, station 52 is expected to contain more noise which is harder to distinguish from cosmic ray signal in comparison to a neutrino station.

To illustrate these differences, a neural network was trained on experimental station 61 and 52 data as two classification categories. The baseline one layer convolutional neural network (CNN) model was trained on data sets E-BG52 (true network output is 0) and E-BG61 (true network output is 1). The network output of the validation step for these two data sets is given in Fig. 5.4. The two data set are distinguishable from one another and achieve almost perfect separation with 99% efficiency for both at a cut of 0.5.

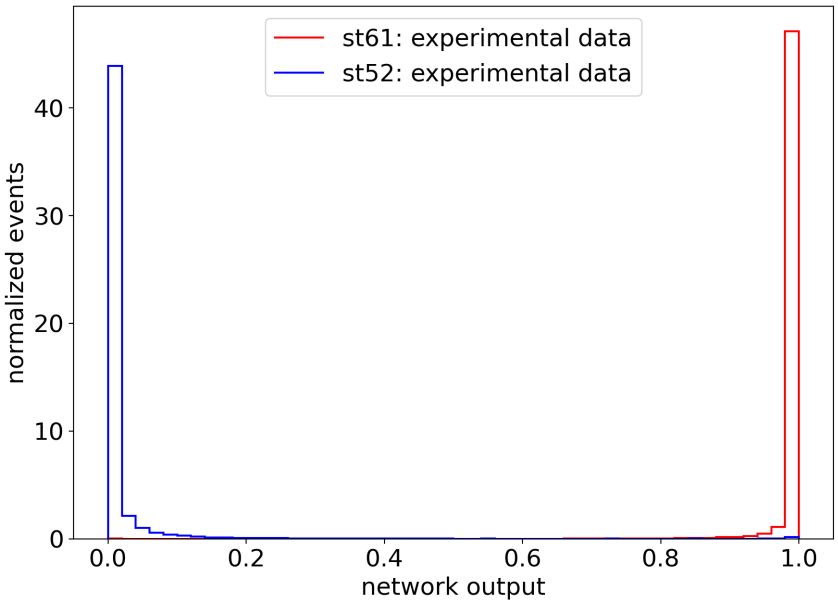

Figure 5.4: Histogram of the network output for experimental background data of station 61 and 52. A network output value close to 0 is classified as station 52 data and close to 1 is classified as station 61 data.

5.3 Neutrino Searches with Deep Learning

In this section, two different neutrino search techniques are discussed: a more conventional method that involves applying various cuts to the data sets, and a new method that utilizes DL to classify the data sets. The goal in both analyses is to reject all of the experimental background noise from station 61 and keep all of the simulated neutrino signal.

5.3.1 Traditional Neutrino Search Techniques

An initial study of data set E-BG61 by former undergraduate student, Leshan Zhao, used traditional methods to obtain an efficiency curve. For this work, cuts were applied on different variables to achieve a clear separation between signal and noise data, including an up-down cut, dipole cut, and template matching correlation cut. The up-down cut required

a ratio below approximately 1.2 for the maximum upward facing versus downward facing LPDA channel value; after this cut, 41,821/74,530 events remained. The dipole cut used the time difference between pulses in the dipole and the LPDAs and after it was applied, only 53 events remained. The last cut matches each event against a neutrino template to obtain a correlation value; after this cut, no events remained and therefore all experimental data were rejected. The results of this study are shown later in Fig. 5.11 when comparing to the DL neutrino search method described next.

5.3.2 Neutrino Search with a CNN Trained on Simulated Noise

Next, a DL approach to neutrino searches is investigated. As a baseline study similar to what was done in Chapter 4, a network is trained on simulated thermal noise and simulated neutrino signal. The trained network is then tested with a different subset of the data as well as the experimental data from station 61. As a preliminary study of the capabilities of DL on station 61 data, the same architecture as the 100 input sample CNN from Chapter 4 is used for training. The results are shown on the left side of Fig. 5.5. The experimental data classified with this model is primarily categorized as signal whereas in this study, close to all of these events need to be rejected as noise events. Therefore, next a slightly larger network architecture is explored using all channels of the input data. The new architecture is a one layer CNN with 10 10x8 kernels, a flatten layer, and a sigmoid output. This network classifies simulated data well with perfect noise rejection and 99% signal efficiency at a network output cut of 0.5 as shown on the right side of Fig. 5.5; however, it is not able to correctly classify the experimental data (black distribution), with a success rate of only approximately 47%.

The reason both of these networks perform with low prediction efficiency on experimental data is likely due to the composition of the experimental station 61 data set; it not only contains thermal noise events but also wind and electronics generated events. Thermal noise

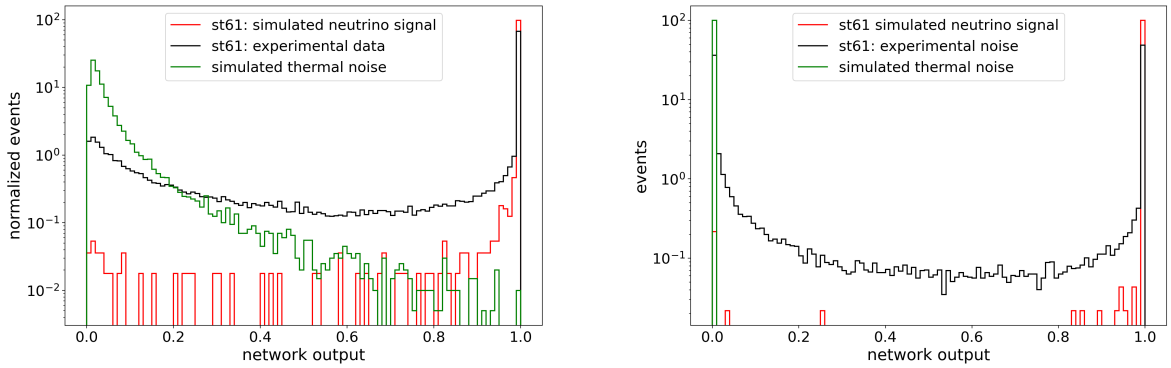

Figure 5.5: Histograms of the network output for experimental background data, simulated neutrino signal, and simulated thermal noise. Both networks were trained solely on simulated thermal noise and simulated neutrino signal. The left plot is trained with the same 100 input sample CNN architecture from Chapter 4 and the right plot is trained with eight input channels of data and 10 10x8 kernels.

events are the only noise type that can be simulated at this time; wind and electronics related events are not currently well enough understood to be simulated by our simulation software and as shown in Fig. 5.3, their waveforms are distinct from thermal noise events. Examining many experimental noise waveforms classified close to one for the black distribution on the right of Fig. 5.5, they mainly consist of wind and electronics events. Inspecting many waveforms classified closer to zero, the majority are thermal events and only a few wind and electronics events. Thermal noise is the only noise type given to the network during training, so the network is not as capable of identifying the other noise sources.

While a neural network can often be flexible in recognizing data that is different from what it was trained on, differences in wind and electronic background noise were significant enough to not be recognized by the models developed for this work. Since these backgrounds are not yet able to be simulated, another approach to this analysis would be training a network on the experimental background data. By training the model on these events, the network will be more flexible in recognizing all of the various background noise event types. The process and challenges that come with training a network on a mixture of simulated and experimental data will be discussed next.

5.3.3 Neutrino Search with a CNN Trained on Experimental Noise

In this study, a network is trained with simulated neutrino signal (S-NU61) and experimental data (E-BG61). Since ARIANNA has yet to measure experimental neutrinos, simulated ones will be used. The experimental data are needed because wind and electronics related noise events cannot be simulated and the network needs to be trained with examples of each background event type. First we again train a network with the 100 input sample CNN architecture from Chapter 4 to get a baseline classification efficiency for these data. The network is trained on 10k experimental events and 1.5k simulated signal events and tested on the remaining data as illustrated in Fig. 5.6. This network performs well in general with 99% signal efficiency and 96% experimental data efficiency at a network output cut value of 0.5; however the goal of this study is to reject all experimental events and this network misclassifies 1,995 events as signal. Thus a more complex network is explored next as well as the minimum amount of training data required for classification.

Next a CNN is trained with all of the channels of input data and an architecture consisting of one hidden layer with size 10 10x8 kernels, a flatten layer, and a sigmoid output. To maximize the amount of classifiable data, the training data set is made as small as possible to keep the testing set large. The network is trained with varying input data sizes and tested against the rest of the data to determine the number of events needed for the training process, see Fig. 5.7. Even with a relatively small amount of data, the network performs well at classifying the events. For the final neutrino search, the black data in Fig. 5.7 is chosen for its size and performance (trained on 10k experimental noise and 1.5k simulated neutrino signal events).

The 10k noise and 1.5k signal distributions shown in black in Fig. 5.7 are then isolated and shown in Fig. 5.9. In addition, the accuracy and loss of this network for 150 epochs are given in Fig. 5.8. The CNN is tested here with the remaining data of 63,530 experimental noise

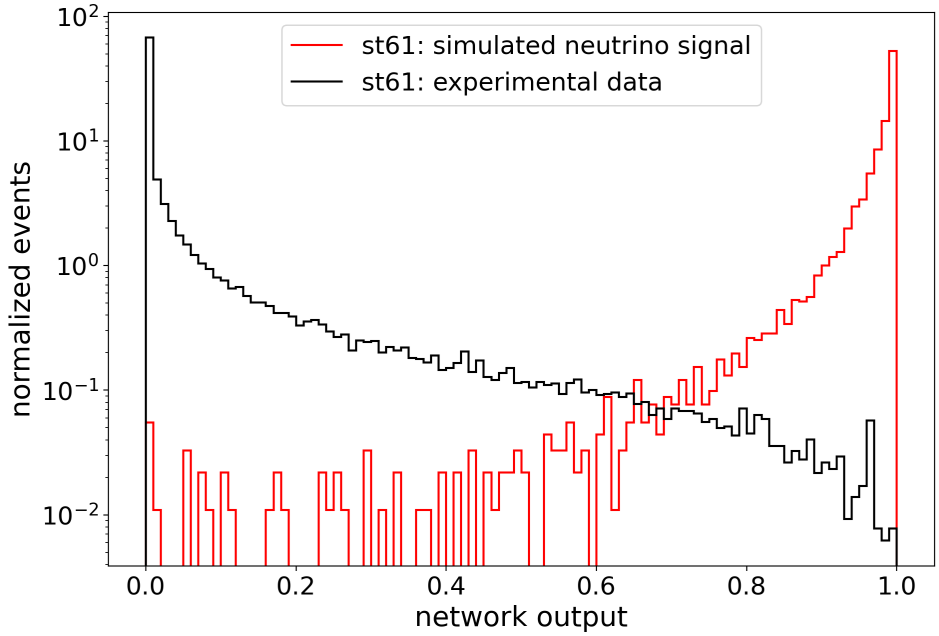

Figure 5.6: Histogram of the network output for experimental background data and simulated neutrino data for station 61 trained on a 100 input sample CNN with 5 10x1 kernels.

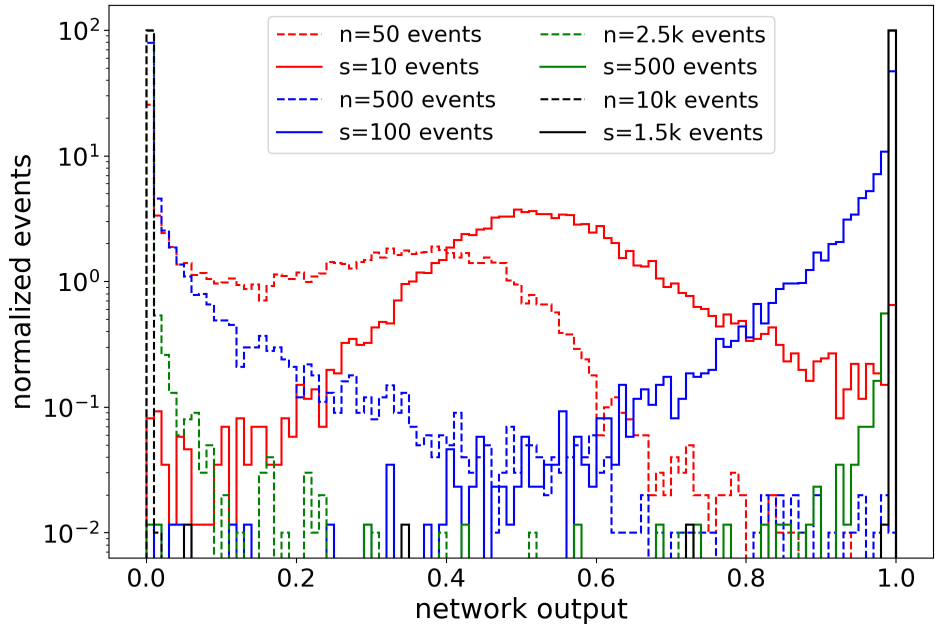

Figure 5.7: The network output for neural networks with varying amounts of input training data (from station 61). The like colors (dashed and solid lines) represent the same network with signal events scaled to match that of the noise events in training.

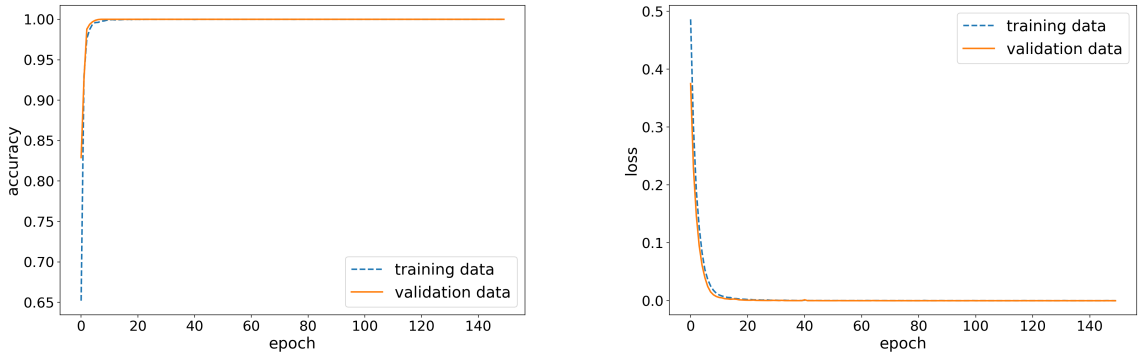

Figure 5.8: Accuracy (left) and loss (right) vs epoch number. The validation data are 20% of the input data and the other 80% are training data. This model is trained with all eight input channels and a one layer CNN with size 10 10x8.

events and 9,106 simulated signal events. At a network output cut of 0.5, the CNN achieves almost perfect separation of data with only two events in E-BG61 incorrectly identified as S-NU61 events. These two events are shown in Fig. 5.10. By analyzing these two waveforms, they are seen to have strong oscillations in channels 4-6, which are the upward facing LPDAs and dipoles; also, the channels involved in triggering the detector (0 and 1 in this case) do not have comparable amplitudes. In neutrino waveforms, the amplitudes of the two triggered channels are generally comparable and the amplitudes of the upward facing LPDAs are smaller than that of the downward facing LPDAs. Thus, these two waveforms would not be considered neutrino events, but since understanding the decision making of DL networks is often challenging, it is difficult to pinpoint the reason for the incorrect classification of these events. Nevertheless, some potential reasons include the uncharacteristically high amplitudes of channel 0 and the more elongated pulse shapes in channels 0 and 1, which look more neutrino-like than most thermal noise events. There are also more high and low pulses in these two events, so it is possible that when the CNN uses kernel correlation to analyze them, it produces more positive contributions to the final network classification.

In the study above, the 10k experimental events data are used for training and thus are not classified by the network. Another analysis is performed here that allows all of the exper-

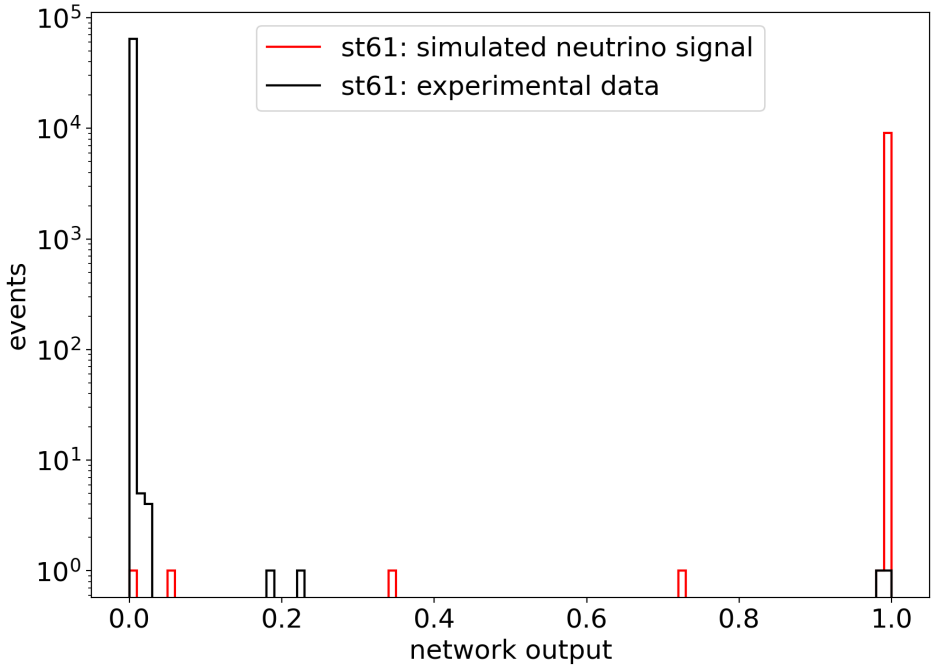

Figure 5.9: Histogram of the network output for experimental background data and simulated neutrino data for station 61. A network output value close to 0 is experimental noise data and close to 1 is simulated neutrino signal data.

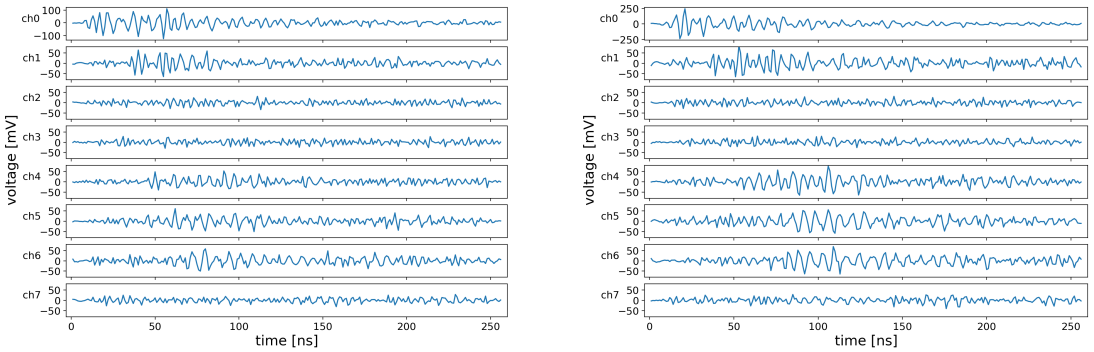

Figure 5.10: The two experimental background events classified incorrectly as simulated neutrino signal. Each waveform is one channel of data, from top to bottom the channel labels start at channel 0 and end at channel 7.

imental data to be classified, which is helpful when comparing this work to the traditional neutrino search analysis method. The experimental and simulated data are each divided into two groups, set A and set B. A network is trained with simulated and experimental events from set A and tested with events from set B. Then, a different network is trained with set B and tested with set A. By training these two networks, all of the experimental data will be classified. The two neural networks are trained with the same model architecture five separate times to search for variations within training. Each network is then tested with either set A and set B, depending on which set it was trained. Examining the classification results for all ten networks, only two events are classified incorrectly, the same two events from Fig. 5.10. Therefore, although the network described in the previous paragraph did not use the 10k experimental noise events in testing, it follows that these events would also be classified correctly and only two noise events remain.

As with the more traditional neutrino search analysis, a neutrino template is used to obtain correlation values for these remaining events in the DL study; after this cut, no events remain and therefore all experimental data were rejected. The plot of correlation of the LPDAs versus SNR is given in Fig. 5.11; the remaining events before the correlation cut are shown for the traditional analysis in red circles and for the DL analysis in white triangles.

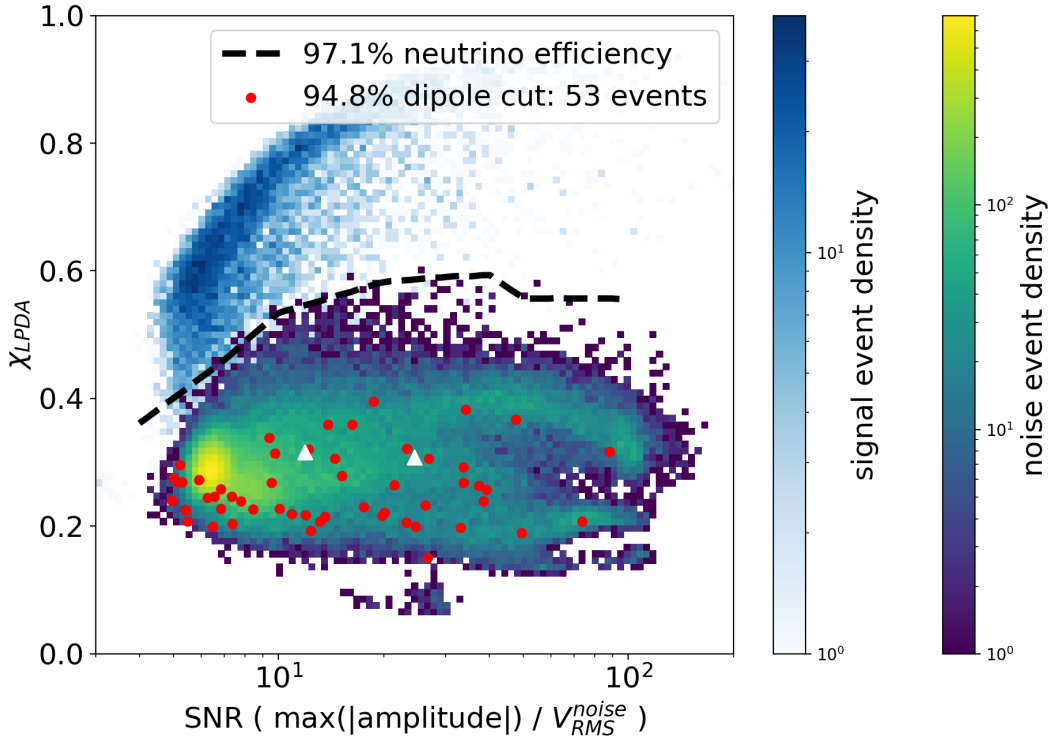

Figure 5.11: The signal-to-noise ratio (SNR) of the LPDA versus the correlation of the LPDA experimental data with a simulated LPDA neutrino template. The simulated neutrino signal event density is given by the blue/white color bar and experimental noise event density is given by the yellow/navy color bar. The 53 remaining experimental background events after the dipole cut in the traditional analysis are shown in red and the two remaining events after the deep learning classification study are white. The black dashed line is the 97.1% neutrino efficiency projection for 1,000 station years of data taking.

Also shown are the simulated neutrino signal event density and the experimental noise event density. In both analyses, all station 61 experimental backgrounds are rejected. In particular, the correlation values of the white triangle data are near the average value for thermal noise, which is approximately $\chi_{\mathrm{LPDA}} = 0.3$. These two methods give similar results after the correlation cut stage of the analysis; however, before the correlation cut, the DL method rejects over 25 times more events for the same signal efficiency.

The network output distribution shown in Fig. 5.9 performs extremely well at classifying both signal and noise events. One may wonder why this network performs significantly

better than the 100 input sample CNN from Chapter 4, Fig. 4.3. The network trained in Chapter 4 used data with a lower trigger threshold of 3.6 SNR compared to the 4.4 SNR threshold of the work here in Fig. 5.9. Events with a lower SNR are closer to the noise level and neutrino signal can be more difficult to differentiate from thermal noise. In addition, the previous network was created with the constraint of being as small as possible, so some classification efficiency was lost. Here, the network uses a larger input data size and more complex architecture because there is no limitation on these parameters in this offline analysis. Therefore, the neural network in this section is expected to perform better than the 100 input sample CNN.

All in all, this is a promising result for the efficiency analysis of station 61. However, since this DL study includes a mixture of experimental and simulated data, there is a concern that the network may be using an artifact between the simulated or experimental data during training to differentiate the two groups. For example, perhaps a small DC offset in the experimental data is not replicated in the simulated signal events, and this difference provides an artificial signature for the network to use to distinguish background from signal. One argument against this idea is illustrated in Fig. 5.7: the less data used for training, the lower the amount of correctly identified events. If a neural network is picking up on an artifact, (i.e. if an artifact is noticeable by the network such that the feature is solely used to distinguish between the two groups) then the network will classify both groups of testing data correctly, even on small amounts of training data. For example, a network is trained on a biased noise set containing a 4 mV DC offset, using 50 input biased noise events and 10 input signal events. With this small amount of training data, the network achieves perfect classification of signal and noise when testing on a few thousand events. Therefore, Fig. 5.7 gives some indication that the network is not picking up on an artifact; if the DL network used an artifact to distinguish signal from background, the signal efficiency would not strongly depend on the number of events in the training set.

Since experimental neutrinos have yet to be measured by the ARIANNA detector, building an experimental data set is not possible. Therefore, experimental cosmic rays are used as a proxy for neutrino signal. The voltage waveforms from cosmic rays are similar to neutrino waveforms, though as Fig. 5.3 shows, there are differences. Cosmic rays will be studied next to determine if the current ARIANNA experimental or simulated data has a distinguishable artifact that would bias the results in Fig. 5.9.

5.4 Cosmic Ray Test Case

Experimentally detected cosmic rays allow us to test for artifacts that may be skewing the results of the neutrino efficiency study. Cosmic rays were detected by ARIANNA station 52, which was configured with four upward facing LPDAs and operated with a trigger sensitive to cosmic rays. This station has the same data acquisition electronics as station 61, including the same amplifier design (Series 300) and power source (solar panels). The idea is that if the promising results of the previous section were influenced by non-physical differences between the simulated signal and experimental background, then the same artifact would cause the network to mistakenly identify experimental cosmic rays as background; that is, if the network were trained on simulated cosmic ray signals and background events obtained from experimental data. In this section an initial study is done to determine the efficiency of a DL network trained on experimental station 52 data, and then later a subset of the data is used during training to further improve the efficiency.

In an analysis similar to that of the last section, a network was trained on a mixture of simulated and experimental data with four channels of input data from S-CR52 (7,000 events) and E-BG52 (42,000 events). The CNN was trained again with a one hidden layer network that consisted of 10 10x4 kernels, a flatten layer, and a sigmoid output. The testing set consisted of the remaining data in S-CR52 and E-BG52 plus the experimental cosmic ray

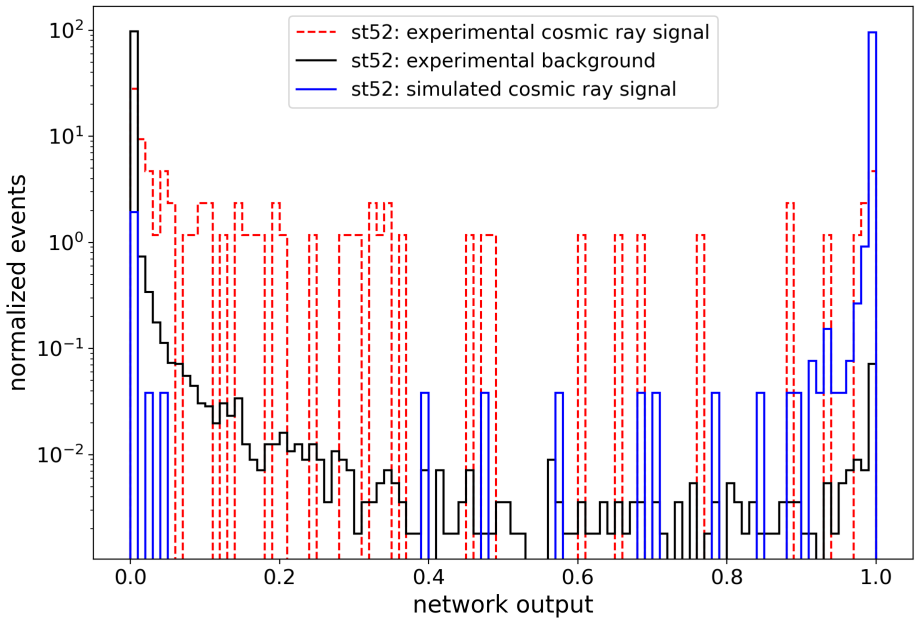

Figure 5.12: Histogram of the network output for experimental station 52 data, simulated cosmic rays, and experimental cosmic rays. The network was trained on the first two data sets mentioned above. A network output value close to 0 is experimental background and close to 1 is cosmic ray signal.

data set, E-CR52. The goal was to determine if a trained neural network uses artifacts in the data to distinguish between the two groups. The expectation is that experimental cosmic ray data will be classified similar to simulated cosmic ray signal rather than being classified as experimental noise due to an artifact. The network output is shown in Fig. 5.12 with the experimental cosmic rays being classified into both signal and noise groups. These results are inconclusive so further analysis needs to be done.

The cosmic ray events classified above 0.5 and below 0.5 are separated and their average FFT values are plotted on top of the average FFT of simulated cosmic rays on the left side of Fig. 5.13. There is a slight difference between the two experimental cosmic ray groups, particularly at high frequencies. The orange data from the below 0.5 classification set has more power in these frequencies compared to the simulated FFT in blue. In general, both groups have similar FFTs which indicates that the network is likely not using the frequency

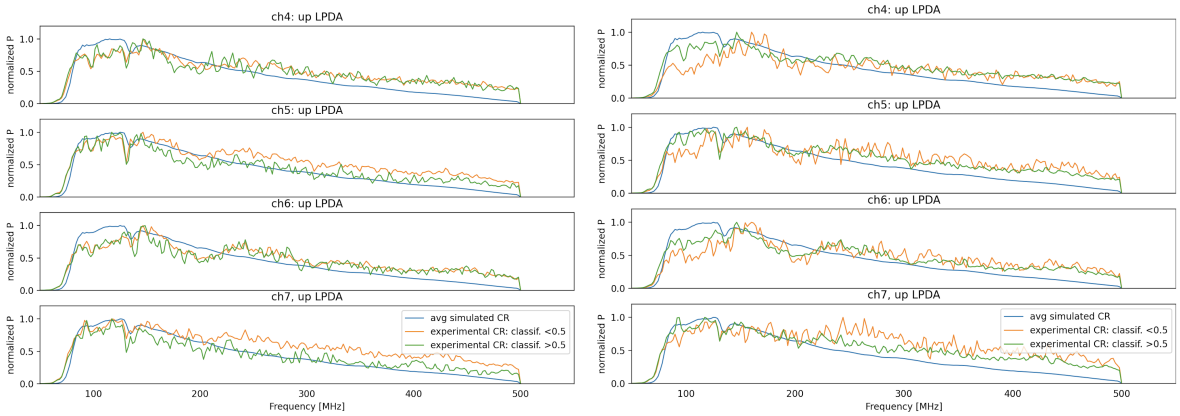

Figure 5.13: The average FFT distributions for simulated cosmic rays and the 85 experimental cosmic rays. The left plot uses a network trained on the entire station 52 data set whereas the right plot uses a network trained on a lower amplitude subset of station 52 data. The 85 cosmic ray events are split into two groups: those classified above 0.5 (14 events on the left and 70 events on the right) and below 0.5 (71 events on the left and 15 events on the right) by the neural network.

content of the waveforms to aid in classification.

There are a few notable differences between the cosmic ray observations by station 52 and the neutrino observations by station 61. First, cosmic rays are observed by upward facing LPDAs, which face the same direction as antropogenic backgrounds that propagate through the atmosphere. In contrast, the backgrounds in the data collected by station 61 arrive in the direction of the backlobe, a significantly less sensitive direction of the LPDA antennas, with different frequency characteristics compared to the front lobe. Consequently, background events from high winds and anthropogenic sources will more closely match the typical waveforms produced by cosmic ray events, see Fig. 5.3. From experience with the ARIANNA HRA, background events tend to have large amplitudes, and a fraction of these will saturate the amplifier, artificially producing more high frequency power than expected for cosmic ray signals.

Furthermore, station 61 is located at the South Pole at an elevation of 2800 m, about 5 km from South Pole Station. Station 52 is located at sea level at Moore's Bay on the Ross Ice Shelf, Antarctica. Due to its proximity to the South Pole, station 61 observes more

anthropogenic noise from weather balloon launches, spark plug noise from snowmobiles and other vehicles with motors, and communications to aircraft and other bases in Antarctica. Station 52 is more remote, so these sources of noise are comparatively small. However, the average winds are stronger at Moore's Bay, so wind generated noise is more frequent. In addition, noise from the electronics on the ARIANNA station is greatest during the time of year when the position of the sun transitions from above the horizon to below the horizon. At Moore's Bay, this happens daily for a period of about a month whereas at the South Pole, this happens only once.

Therefore, the experimental data from station 52 will contain a relatively high fraction of large amplitude background events from electronics and wind. For these reasons, the high amplitude events in the data from station 52 are removed. The purpose is to produce a background data set that is biased to contain mostly thermal events. After the experimental data set from station 52 is cleaned, it should be straightforward for the DL network to differentiate cosmic rays events from thermal backgrounds. However, if most experimental cosmic ray events are identified as background, then this would suggest that artifacts are present in the training set.

Wind and electronic noise events are typically higher in amplitude compared with thermal noise, as shown in Fig. 5.14. There are two populations in the experimental data with the first being thermal noise dominated and the second being wind and electronic noise dominated. Simulated thermal noise is compared to this distribution and although both curves have slightly different thresholds, the general shapes are similar. To isolate the experimental data that matches most closely to the simulated thermal noise distribution, a cut is made on the maximum amplitude of the maximum channel. When deciding which cut value to choose, considerations are taken for the remaining event quantity. A cut on data above 60 mV isolates a majority of the experimental thermal noise and contains 23,478 lower amplitude events. This data set is called E2-BG52 and is used in this cosmic ray study.

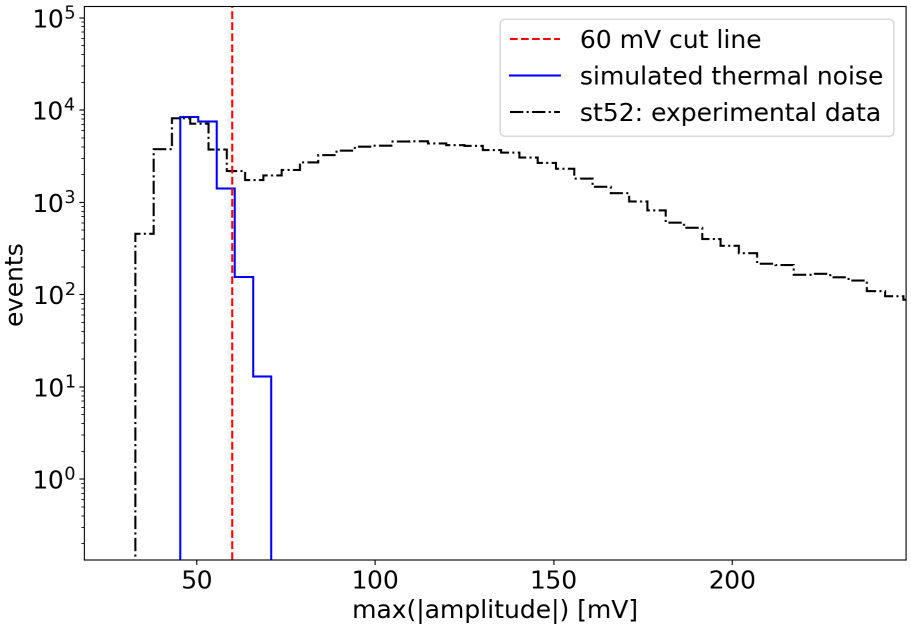

Figure 5.14: Histogram of the maximum absolute value of the maximum waveform. The black curve contains experimental data (E2-BG52), the blue curve is simulated thermal noise (S-BG61), and the red dashed line denotes the 60 mV chosen cut line for thermal noise events.

Next, a mixture of simulated and experimental cosmic ray data will be used to train and validate a DL model, to show there is no training artifact in the model from Fig. 5.9. Specifically, the neural network will be trained on simulated cosmic ray signal and experimental background noise, then validated on both along with experimental cosmic ray data. The data sets for training are E2-BG52 (20,000 events) and S-CR52 (5,000 events), trained with the baseline CNN model structure and four channels of input data. The data sets for validation were the remaining events from E2-BG52 and S-CR52, and all 85 events from E-CR52. Again the FFTs are plotted on the right side of Fig. 5.13 and there is a larger separation in the greater than 0.5 and less than 0.5 groups. In particular, there are some experimental cosmic rays that have less power around the 100 MHz frequency range compared to the events that are correctly classified. This indicates that the network is now picking up on properties of the waveforms such as the FFTs to distinguish the cosmic ray and background noise events.

The histogram of the network outputs are shown in Fig. 5.15 as well as their accuracy and loss plots in Fig. 5.16. Although there are not many events in the experimental cosmic ray data set, its network output follows a similar distribution to the simulated cosmic ray data set. If there were irrelevant artifacts in the simulated or experimental data, the network would pick up these differences between simulated and experimental cosmic rays. Since their network output distributions are well correlated, there is no such artifact or feature seen. To further illustrate the similarities between the experimental cosmic ray and simulated cosmic ray distributions, a statistical analysis is performed on the data.

This statistical analysis uses the Wasserstein distance [111] to compare the tagged cosmic rays distribution to both the simulated cosmic ray and experimental noise distributions. The Wasserstein distance is a mathematical metric for describing the similarity (or distance) between two probability distributions. First the tagged cosmic ray histogram is fluctuated with a Gaussian function for bins containing greater than four events. Specifically each bin

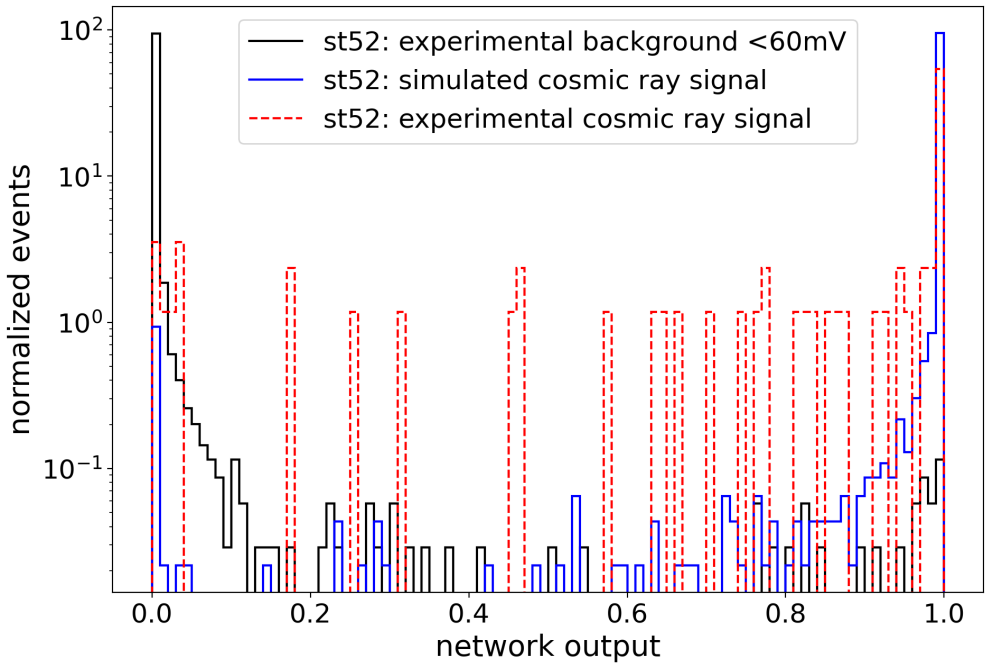

Figure 5.15: Histogram of the network output for a subset of experimental station 52 data with the maximum amplitudes below 60 mV, simulated cosmic rays, and experimental cosmic rays. The network was trained on the first two data sets mentioned above. A network output value close to 0 is experimental background noise and close to 1 is cosmic ray signal.

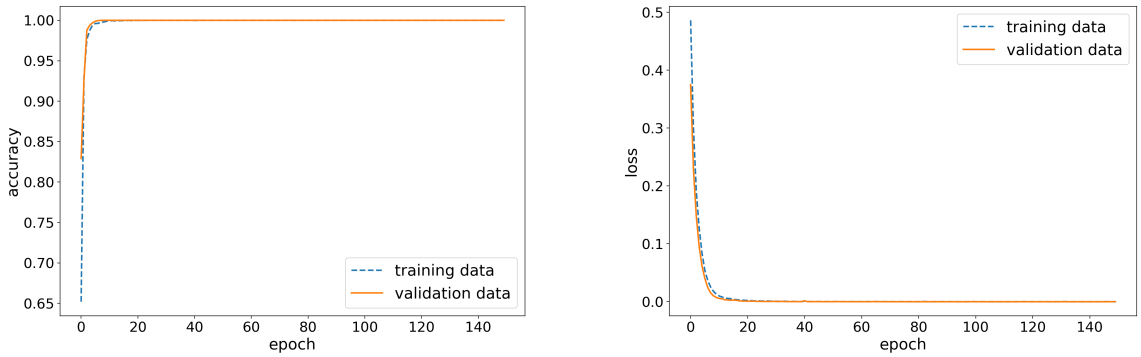

Figure 5.16: Accuracy (left) and loss (right) vs epoch number. The validation data are 20% of the input data and the other 80% are training data. This model is trained with four input channels and a one layer CNN with size 10 10x4.

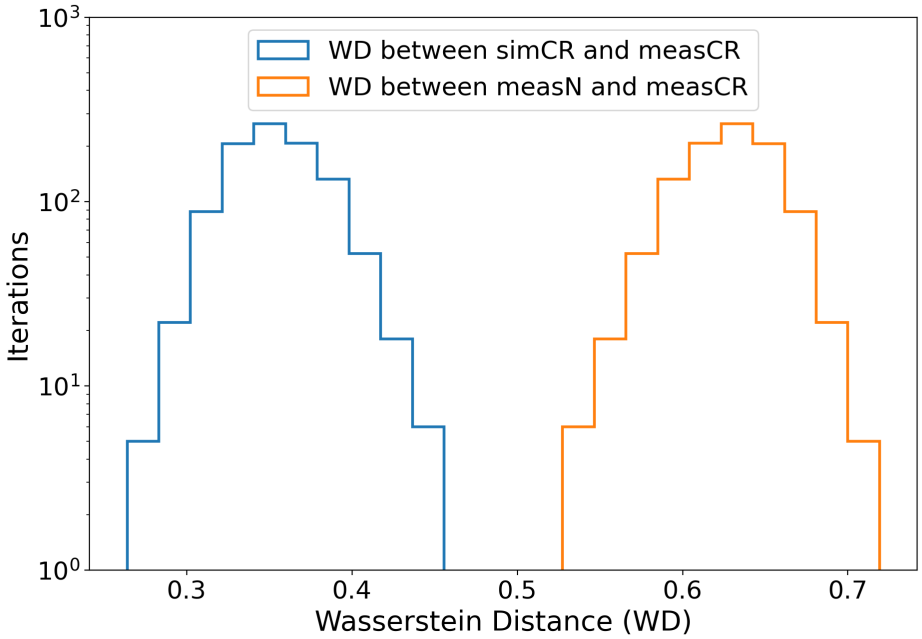

Figure 5.17: The Wasserstein distance (WD) versus the iterations of data fluctuation (1,000 in total). The blue curve gives the WD between the simulated and experimental cosmic ray distributions. The orange curve gives the WD between the experimental noise and cosmic ray distributions.

is fluctuated above and below its mean value with a sigma equal to the square root of the number of events in the bin. For bins containing four or less events, a Poisson distribution is used to fluctuate the bin values. In total, 1,000 different histograms are generated that contain the fluctuated tagged cosmic ray network output values. These histograms are each separately compared via the Wasserstein distance to the simulated cosmic ray distribution and experimental background noise distribution as shown in Fig. 5.17; the curves do not overlap and the Wasserstein distance is smaller for the two cosmic ray distributions (shown in blue). Thus if the tagged cosmic ray distribution is fluctuated over many iterations, it is unlikely to fluctuate into a distribution that matches the experimental background data.

This study demonstrates that the results in the neutrino study can be trusted because the simulation framework for cosmic rays and neutrinos is similar. It is important to note that the distributions in Fig. 5.15 do not show that cosmic rays can be identified efficiently, only

that mixing simulated and experimental data when training a model is possible. Further studies and more complex DL models would need to be built to identify cosmic rays in either real-time or an offline analysis. However, this is not the main mission in this work; here, the main goal is to measure and identify neutrino signal.

5.5 Conclusion

This work compared a more traditional data analysis technique for neutrino searches to a new method using deep learning. In the final analysis step before the correlation cut, the traditional approach rejected all but 53 experimental events whereas the DL approach rejected all but two events. Both approaches retained 99% neutrino signal efficiency. After the correlation cut, both analysis techniques produced 100% experimental data rejection. In scaling these rejection values to larger data sets with many more station years, the DL approach will be more impactful as it was found to be over 25 times more efficient at rejecting experimental noise data.

The DL rejection results were validated by demonstrating that training on a mixture of simulated and experimental data can be performed without the neural net becoming confused by potential artifact features in the data sets. Cosmic ray experimental data were used to validate the neutrino analysis as there were no artifacts seen in the final cosmic ray trained model. This suggests that artifacts are not affecting the efficiency or background rejection results of the neutrino analysis.

The source of electronics related background events from the BMU has been identified and a solution developed [3]. Due to this, BMU noise in future ARIANNA station is expected to be eliminated or significantly reduced, so fewer background event types will need to be identified by the neural network. Using these analysis techniques on experimental data will

be important for most experiments because simulations may not always give a complete picture of real data (i.e. wind events cannot yet be simulated). Particular attention will be focused on filtering out events from high winds, which will likely be the dominant class of non-thermal background events. A future project could involve performing unsupervised learning on experimental ARIANNA data to better categorize the different background groups. After separating wind events from other backgrounds, their features could be studied and possibly quantified in more detail.

Another potential path forward is to take the neural network from this offline analysis and combine it with the real-time DL filter discussed in Chapter 4. This idea was first proposed by ARIANNA collaborator C. Glaser, now at Uppsala University in Sweden. While the networks used in the offline analysis were larger than the real-time filter (at least in the input data size), they are still relatively small networks with only one hidden layer. With improved electronics and data processing, a future DL filter could contain multiple piped networks that each reject a different type of background noise. For example, in addition to the thermal noise filter discussed in Chapter 4, a filter focused on wind event rejection could be added to the data processing pipeline; however, this requires isolating a representative set of wind events and training another neural network on these backgrounds. Lastly, besides a real-time DL filter, another possible approach would be to implement a digital trigger using recurrent neural networks. The offline data analysis study outlined in this chapter has demonstrated the potential of such a digital neural network trigger since DL was used on experimental data and performed well at rejecting these background noise sources.

Chapter 6

Conclusion and Outlook

The ARIANNA experiment is designed to measure EHE (extremely-high energy) neutrinos. Current experimental limits on the EHE flux imply a small detection rate per station, so it is important to improve the sensitivity of each station by lowering the trigger threshold and optimizing the analysis tools. Deep learning (DL) has been shown to improve on more traditional filtering and analysis techniques within the ARIANNA experiment. A final summary of the work in this thesis as well as the future of these projects is discussed next. For project specific conclusions and final thoughts, also refer to sections 4.9 and 5.5.

6.1 Summary

A few other studies were performed with the same simulated data used in the above work to better understand the flexibility of CNNs (convolutional neural networks). The first study focused on optimizing the hyperparameters within a one layer CNN. It found that an optimized network gained around 3% signal efficiency with a trade off of being 1.68 times larger (and therefore slower); perhaps this more optimized network could be used in

the future as a real-time DL filter once a new microprocessor is installed. Another study investigated the impact of continuous wave additions to ARIANNA data. The effect on the efficiency depended on the frequency of the continuous wave, but in general, there was little difference in the classification of thermal noise with the addition of a 2 mV continuous wave. Larger effects were seen at around 10 mV continuous wave additions, which is the $V_{\mathrm{RMS}}^{\mathrm{noise}}$ noise level. These studies further illustrate the capabilities and limitations of DL networks; it is imperative to balance these DL results with tests for robustness and flexibility since the network training process is often difficult to interpret.

Next, an offline analysis of neutrino searches was conducted on experimental ARIANNA data. A traditional analysis technique that applies cuts to variables was compared to DL analysis methods. The traditional approach rejected all except 53 experimental events whereas the DL approach rejected all except two events with both approaches retaining 99% neutrino signal efficiency. After a final cut on correlation values, both analysis techniques produced 100% experimental data rejection. Scaling these rejection values to many more station years, the DL method will be more impactful as it was found to be over 25 times more efficient at rejecting experimental noise data. To validate the DL results which were trained on a mixture of simulated and experimental data, a study on cosmic ray data was performed investigating potential artifacts within the data sets. There were no artifact features discovered, which suggests that artifacts are not affecting the efficiency and background rejection results of the neutrino analysis.

6.2 Outlook

Within the ARIANNA experiment, DL techniques have been shown to improve real-time data collection and offline analysis studies. Currently, template matching methods are common analysis techniques in physics, and the work in Sec. 4.6.1 showed DL gives about three

orders-of-magnitude more noise rejection at the same signal efficiency. These techniques are not limited to the ARIANNA detector as many Askaryan neutrino detectors can benefit from the utilization of deep learning, including the experiments mentioned in Sec. 1.3 and, in particular, radio-based neutrino detectors such as ARA and RNO-G. These experiments have similar data structures to ARIANNA and would make good candidates to pursue DL techniques further.

Looking to the future of DL techniques within ARIANNA, there are many opportunities to improve the capabilities of the experiment. As shown through offline studies, future recommended projects include using supervised learning on experimental data to identify cosmic rays and compare the performance to traditional template matching methods. This technique can also be used to classify experimental background event types, which could aid in building more specialized DL filters that can be implemented in real-time. Various DL filters can be added in succession to reject different background event types. If, for example, the first applied filter rejected thermal noise at 10^5, the remaining data – say wind events – could be piped through another filter with a comparatively slower processing speed due to their less frequent occurrence. Another avenue to explore further is network size optimization; there are ways to further reduce the size of a neural network such as weights pruning and bit size reduction. There is also work being done to improve the hardware components, which will allow for larger networks to be explored. With upgraded hardware, new ideas can be explored such as implementing a digital DL trigger. This trigger could work on its own to reject noise events or perhaps be the first step in filtering data with additional real-time DL filters to follow. There are many options that need to be explored and their combinations need to be tested. While there are many paths to move forward with deep learning, interpretability studies and performance tests should always be included when designing new neural networks. Deep learning analysis techniques will continue to improve in physics as more experiments utilize its capabilities, and the ARIANNA experiment is working towards this goal.

Bibliography

[1] Deep Learning ARIANNA. https://github.com/a-anker/ARIANNA_Experiment.

[2] Google Maps, NASA Map data 2022. https://www.google.com/maps/@-78.2299945,165.9175154,282470m/data=!3m1!1e3.

[3] Manuel Paul, private communication, 2022.

[4] Evidence for high-energy extraterrestrial neutrinos at the IceCube detector. *Science*, 342(6161), nov 2013.

[5] The Pierre Auger Cosmic Ray Observatory. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 798:172–213, oct 2015.

[6] Iridium Next Satellite system. https://https://directory.eoportal.org/web/eoportal/satellite-missions/i/iridium-next, 2019.

[7] Raspberry Pi Compute Module 3+. https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM3plus_1p0.pdf, 2019.

[8] Raspberry Pi Zero 2 W. https://datasheets.raspberrypi.com/rpizero2/raspberry-pi-zero-2-w-product-brief.pdf, 2021.

[9] A portrait of the higgs boson by the CMS experiment ten years after the discovery. *Nature*, 607(7917):60–68, jul 2022.

[10] M. Aartsen et al. The IceCube neutrino observatory: instrumentation and online systems. *Journal of Instrumentation*, 12(03):P03012–P03012, mar 2017.

[11] M. Aartsen et al. Differential limit on the extremely-high-energy cosmic neutrino flux in the presence of astrophysical background from nine years of IceCube data. *Physical Review D*, 98(6), Sep 2018.

[12] M. G. Aartsen et al. Detection of a particle shower at the glashow resonance with IceCube. *Nature*, 591(7849):220–224, mar 2021.

[13] M. G. Aartsen et al. IceCube-Gen2: the window to the extreme Universe. *Journal of Physics G: Nuclear and Particle Physics*, 48(6):060501, Apr 2021.

[14] M. G. Aartsen et al. IceCube-gen2: the window to the extreme universe. *Journal of Physics G: Nuclear and Particle Physics*, 48(6):060501, apr 2021.

[15] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[16] Q. Abarr et al. The Payload for Ultrahigh Energy Observations (PUEO): a white paper. *Journal of Instrumentation*, 16(08):P08035, Aug 2021.

[17] R. Abbasi et al. A convolutional neural network based cascade reconstruction for the IceCube neutrino observatory. *Journal of Instrumentation*, 16(07):P07041, jul 2021.

[18] B. P. Abbott et al. Observation of gravitational waves from a binary black hole merger. *Phys. Rev. Lett.*, 116:061102, Feb 2016.

[19] Afar communications inc. `http://afar.net/`, accessed May 2021.

[20] J. Aguilar et al. Design and sensitivity of the Radio Neutrino Observatory in Greenland (RNO-G). *Journal of Instrumentation*, 16(03):P03025, Mar 2021.

[21] J. A. Aguilar et al. Reconstructing the neutrino energy for in-ice radio detectors. 2021.

[22] J. A. Aguilar et al. Triboelectric backgrounds to radio-based uhe neutrino exeperiments. 2021.

[23] M. Ahlers, L. Anchordoqui, M. Gonzalez–Garcia, F. Halzen, and S. Sarkar. GZK neutrinos after the fermi-LAT diffuse photon flux measurement. *Astroparticle Physics*, 34(2):106–115, sep 2010.

[24] M. Ahlers and F. Halzen. Opening a new window onto the universe with icecube. *Progress in Particle and Nuclear Physics*, 102:73–88, 2018.

[25] P. Allison et al. Performance of two Askaryan Radio Array stations and first results in the search for ultrahigh energy neutrinos. *Physical Review D*, 93(8), Apr 2016.

[26] J. Alvarez-Muñ iz, A. Romero-Wolf, and E. Zas. Čerenkov radio pulses from electromagnetic showers in the time domain. *Physical Review D*, 81(12), jun 2010.

[27] L. Alzubaidi et al. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8, 2021.

[28] A. Anker et al. Targeting ultra-high energy neutrinos with the ARIANNA experiment. *Advances in Space Research*, 64(12):2595–2609, Dec 2019.

[29] A. Anker et al. A search for cosmogenic neutrinos with the ARIANNA test bed using 4.5 years of data. *JCAP*, 03:053, 2020.

[30] A. Anker et al. Probing the angular and polarization reconstruction of the ARIANNA detector at the south pole. *Journal of Instrumentation*, 15(09):P09039–P09039, sep 2020.

[31] A. Anker. et al. White Paper: ARIANNA-200 high energy neutrino telescope. 2020.

[32] A. Anker et al. A novel trigger based on neural networks for radio neutrino detectors. *PoS*, ICRC2021:1074, 2021.

[33] A. Anker et al. Improving sensitivity of the ARIANNA detector by rejecting thermal noise with deep learning. *Journal of Instrumentation*, 17(03):P03007, mar 2022.

[34] A. Anker et al. Measuring the polarization reconstruction resolution of the ARIANNA neutrino detector with cosmic rays. *Journal of Cosmology and Astroparticle Physics*, 2022(04):022, apr 2022.

[35] D. Ardouin et al. Geomagnetic origin of the radio emission from cosmic ray induced air showers observed by CODALEMA. *Astropart. Phys.*, 31:192–200, 2009.

[36] ARIANNA Collaboration et al. Performance of the arianna hexagonal radio array. 2015.

[37] G. A. Askar'yan. Excess negative charge of an electron-photon shower and its coherent radio emission. *Zh. Eksp. Teor. Fiz.*, 41:616–618, 1961.

[38] M. S. Athar et al. Status and perspectives of neutrino physics. *Progress in Particle and Nuclear Physics*, 124:103947, may 2022.

[39] J. Avva, J. M. Kovac, C. Miki, D. Saltzberg, and A. G. Vieregg. An in situ measurement of the radio-frequency attenuation in ice at Summit Station, Greenland. *J. Glaciol.*, 61:1005–1011, 2015.

[40] S. Barwick et al. A first search for cosmogenic neutrinos with the ARIANNA Hexagonal Radio Array. *Astroparticle Physics*, 70:12–26, Oct 2015.

[41] S. Barwick et al. Time-domain response of the ARIANNA detector. *Astroparticle Physics*, 62:139–151, mar 2015.

[42] S. Barwick et al. Radio detection of air showers with the arianna experiment on the ross ice shelf. *Astroparticle Physics*, 90:50–68, 2017.

[43] S. Barwick and C. Glaser. Radio detection of high energy neutrinos in ice. 2022.

[44] S. W. Barwick. Arianna: A new concept for uhe neutrino detection. *Journal of Physics: Conference Series*, 60(1):276, mar 2007.

[45] S. W. Barwick et al. South Polar in situ radio-frequency ice attenuation. *Journal of Glaciology*, 51(173):231–238, 2005.

[46] S. W. Barwick. et al. Design and Performance of the ARIANNA Hexagonal Radio Array Systems. 2014.

[47] S. W. Barwick et al. Radar Absorption, Basal Reflection, Thickness, and Polarization Measurements from the Ross Ice Shelf. *J. Glaciol.*, 61(227):438–446, 2015.

[48] D. Bergman. Telescope Array Combined Fit to Cosmic Ray Spectrum and Composition. *PoS*, ICRC2021:338, 2021.

[49] P. Bezyazeekov et al. Reconstruction of sub-threshold events of cosmic-ray radio detectors using an autoencoder. In *Proceedings of 37th International Cosmic Ray Conference — PoS(ICRC2021)*. Sissa Medialab, aug 2021.

[50] C. Haack for the IceCube collaboration. A measurement of the diffuse astrophysical muon neutrino flux using eight years of IceCube data. *PoS(ICRC2017)1005*, 2017.

[51] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová . Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4), dec 2019.

[52] P. Carlson. Astrophysics and cosmology with balloons. 590:47–52, 2005. 17th ESA Symposium on European Rocket and Balloon Programmes and Related Research, Sandefjord, NORWAY, MAY 30-JUN 02, 2005.

[53] E. Chiem. *Multi-Gigahertz Synchronous Sampling and Triggering (SST) Circuit with Picosecond Timing Resolution*. PhD thesis, University of California-Irvine, 2017. `https://escholarship.org/uc/item/1m29740h`.

[54] F. Chollet et al. Keras. `https://keras.io/`, 2015.

[55] N. Choma et al. Graph neural networks for icecube signal classification. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 386–391, 2018.

[56] D. Ciresan et al. Flexible, High Performance Convolutional Neural Networks for Image Classification. pages 1237–1242, 07 2011.

[57] L. Cremonesi et al. The simulation of the sensitivity of the antarctic impulsive transient antenna (anita) to askaryan radiation from cosmogenic neutrinos interacting in the antarctic ice. *Journal of Instrumentation*, 14(08):P08011, aug 2019.

[58] G. Di Sciascio. Measurement of energy spectrum and elemental composition of pev cosmic rays: Open problems and prospects, 2022.

[59] J. Duarte et al. Fast inference of deep neural networks in FPGAs for particle physics. *Journal of Instrumentation*, 13(07):P07027–P07027, Jul 2018.

[60] R. Durrer. The cosmic microwave background: the history of its experimental investigation and its significance for cosmology. *Classical and Quantum Gravity*, 32(12):124007, jun 2015.

[61] M. Erdmann, F. Schlüter, and R. Š mída. Classification and recovery of radio signals from cosmic ray induced air showers with deep learning. *Journal of Instrumentation*, 14(04):P04005–P04005, apr 2019.

[62] C. Firestone. Performance vs. competence in human–machine comparisons. *Proceedings of the National Academy of Sciences*, 117(43):26562–26571, 2020.

[63] G. G. Gaswint. *Quantifying the Neutrino Energy and Pointing Resolution of the ARIANNA Detector*. PhD thesis, University of California-Irvine, 2021. `https://escholarship.org/uc/item/1bj9r6rb`.

[64] C. Glaser and S. W. Barwick. An improved trigger for Askaryan radio detectors. *Journal of Instrumentation*, 16(05):T05001, may 2021.

[65] C. Glaser et al. NuRadioReco: a reconstruction framework for radio neutrino detectors. *The European Physical Journal C*, 79(6), Jun 2019.

[66] C. Glaser et al. NuRadioMC: simulating the radio emission of neutrinos from interaction to detector. *The European Physical Journal C*, 80(2), Jan 2020.

[67] C. Glaser, S. McAleer, P. Baldi, and S. W. Barwick. Deep learning reconstruction of the neutrino energy with a shallow Askaryan detector. *PoS*, ICRC2021:1051, 2021.

[68] C. Glaser, S. McAleer, S. Stjärnholm, P. Baldi, and S. Barwick. Deep-learning-based reconstruction of the neutrino direction and energy for in-ice radio detectors. *Astroparticle Physics*, 145:102781, mar 2023.

[69] P. W. Gorham et al. Accelerator measurements of the askaryan effect in rock salt: A roadmap toward teraton underground neutrino detectors. *Phys. Rev. D*, 72:023002, Jul 2005.

[70] P. W. Gorham et al. Observations of the askaryan effect in ice. *Physical Review Letters*, 99(17), oct 2007.

[71] K. Greisen. End to the cosmic-ray spectrum? *Phys. Rev. Lett.*, 16:748–750, Apr 1966.

[72] D. Guest, K. Cranmer, and D. Whiteson. Deep learning and its application to LHC physics. *Annual Review of Nuclear and Particle Science*, 68(1):161–181, oct 2018.

[73] GVD Collaboration et al. The baikal-gvd neutrino telescope: First results of multi-messenger studies. 2019.

[74] S. Hallmann et al. Sensitivity studies for the IceCube-Gen2 radio array. *PoS(ICRC2021)1183*, 2021.

[75] L. Hertel, J. Collado, P. Sadowski, J. Ott, and P. Baldi. Sherpa: Robust hyperparameter optimization for machine learning, 2020.

[76] T. Huege, M. Ludwig, and C. W. James. Simulating radio emission from air showers with CoREAS. In *AIP Conference Proceedings*. AIP, 2013.

[77] R. A. Hulse and J. H. Taylor. Discovery of a pulsar in a binary system. , 195:L51–L53, Jan. 1975.

[78] R. Jimenez, C. Pena-Garay, K. Short, F. Simpson, and L. Verde. Neutrino masses and mass hierarchy: evidence for the normal hierarchy. *Journal of Cosmology and Astroparticle Physics*, 2022(09):006, sep 2022.

[79] F. D. Kahn and I. Lerche. Radiation from Cosmic Ray Air Showers. *Proceedings of the Royal Society of London Series A*, 289(1417):206–213, Jan. 1966.

[80] G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman, and D. Shih. Machine learning in the search for new fundamental physics, 2021.

[81] S. Kiranyaz. et al. 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151:107398, 2021.

[82] S. A. Kleinfelder et al. The SST fully-synchronous multi-GHz analog waveform recorder with Nyquist-rate bandwidth and flexible trigger capabilities. *2014 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, Nov 2014.

[83] B. Kolundzija. Wipl-d: From university software to company product. 01 2011.

[84] D. Kostunin, P. Bezyazeekov, R. Hiller, F. Schröder, V. Lenok, and E. Levinson. Reconstruction of air-shower parameters for large-scale radio detectors using the lateral distribution. *Astroparticle Physics*, 74:79–86, 2016.

[85] K. Kotera and A. V. Olinto. The astrophysics of ultrahigh-energy cosmic rays. *Annual Review of Astronomy and Astrophysics*, 49(1):119–153, 2011.

[86] I. Kravchenko. Recent results from the RICE experiment at the south pole. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 662:S42–S45, jan 2012.

[87] Y. Lee. Image classification with artificial intelligence: Cats vs dogs. In *2021 2nd International Conference on Computing and Data Science (CDS)*, pages 437–441, 2021.

[88] J. W. Nam et al. Design and implementation of the TAROGE experiment. *International Journal of Modern Physics D*, 25:1645013, 2016.

[89] A. Nelles. A wind-turbine for autonomous stations for radio detection of neutrinos. *PoS*, ICRC2019:968, 2020.

[90] R. Nichol et al. Radio detection of high-energy particles with the ANITA experiment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 626, 01 2011.

[91] Y. Pan et al. A neural network based UHE neutrino reconstruction method for the Askaryan Radio Array (ARA). *PoS*, ICRC2021:1157, 2021.

[92] A. Paszke et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[93] C. R. Persichilli. Performance and Simulation of the ARIANNA Pilot Array, with Implications for Future Ultra-high Energy Neutrino Astronomy. *UC Irvine Thesis*, 2018. Retrieved from https://escholarship.org/uc/item/1340083p.

[94] T. Prakash. *A Fully-Synchronous Multi-GHz Analog Waveform Recording And Triggering Circuit*. PhD thesis, University of California-Irvine, 2017. `https://escholarship.org/uc/item/5bt060dn`.

[95] P. B. Price. Comparison of optical, radio, and acoustical detectors for ultrahigh-energy neutrinos. *Astropart. Phys.*, 5:43–52, 1996.

[96] A. Rehman, A. Coleman, F. G. Schröder, and D. Kostunin. Classification and Denoising of Cosmic-Ray Radio Signals using Deep Learning. *PoS*, ICRC2021:417, 2021.

[97] M. M. Saeed, Z. Al Aghbari, and M. Alsharidah. Big data clustering techniques based on spark: a literature review. *PeerJ Computer Science*, 6:e321, Nov. 2020.

[98] D. Saltzberg et al. Observation of the askaryan effect: Coherent microwave cherenkov emission from charge asymmetry in high-energy particle cascades. *Phys. Rev. Lett.*, 86:2802–2805, Mar 2001.

[99] P. Schellart et al. Polarized radio emission from extensive air showers measured with LOFAR. *Journal of Cosmology and Astroparticle Physics*, 2014(10):014–014, oct 2014.

[100] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, Jan 2015.

[101] O. Scholten, K. D. de Vries, and K. Werner. Coherent radiation from extensive air showers. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 662:S80–S84, 2012. 4th International workshop on Acoustic and Radio EeV Neutrino detection Activities.

[102] F. G. Schröder. Radio detection of Cosmic-Ray Air Showers and High-Energy Neutrinos. *Prog. Part. Nucl. Phys.*, 93:1–68, 2017.

[103] S. Seager, D. D. Sasselov, and D. Scott. A new calculation of the recombination epoch. *The Astrophysical Journal*, 523(1):L1–L5, sep 1999.

[104] G. Sironi. Cosmic ray electrons and galactic radio noise. *Planetary and Space Science*, 17(1):1–11, 1969.

[105] J. M. Souney et al. The South Pole ice core (SPICEcore) project. *American Geophysical Union, Fall Meeting 2018, abstract C41C-1771*, Dec 2018.

[106] M. Tanabashi et al. Review of particle physics. *Phys. Rev. D*, 98:030001, Aug 2018.

[107] J. H. Taylor and J. M. Weisberg. A new test of general relativity - Gravitational radiation and the binary pulsar PSR 1913+16. , 253:908–920, Feb. 1982.

[108] K. Upender and V. S. K. Pasupuleti. Real time handwritten digits recognition using convolutional neural network. In *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pages 69–73, 2021.

[109] A. van Vliet, R. A. Batista, and J. R. Hörandel. Determining the fraction of cosmic-ray protons at ultrahigh energies with cosmogenic neutrinos. *Physical Review D*, 100(2), jul 2019.

[110] A. van Vliet, R. A. Batista, and J. R. Hörandel. Determining the fraction of cosmic-ray protons at ultrahigh energies with cosmogenic neutrinos. *Physical Review D*, 100(2), Jul 2019.

[111] C. Villani. *The Wasserstein distances*, pages 93–111. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[112] E. Waxman and J. Bahcall. High energy neutrinos from astrophysical sources: An upper bound. *Phys. Rev. D*, 59:023002, Dec 1998.

[113] G. T. Zatsepin and V. A. Kuz'min. Upper Limit of the Spectrum of Cosmic Rays. *Soviet Journal of Experimental and Theoretical Physics Letters*, 4:78, Aug. 1966.

[114] L. Zhao et al. Polarization Reconstruction of Cosmic Rays with the ARIANNA Neutrino Radio Detector. *PoS*, ICRC2021:1156, 2021.

[115] J. Álvarez Muñiz et al. The Giant Radio Array for Neutrino Detection (GRAND): Science and design. *Science China Physics, Mechanics & Astronomy*, 63(1), Aug 2019.

Appendix A

Resources for Deep Learning Analyses

A.1 Software

The scripts to reproduce the analyses in this work can be found on this GitHub page [1]. Refer to the README file in this GitHub to install all of the required software and to see an overview of the scripts in this work. A few of the important scripts are also given in the following sections of the appendix. If there are any questions relating to this work, please email me at aa@astridanker.com.

A.2 Basic convolutional neural network training code

```
import os
import glob
from matplotlib import pyplot as plt
import numpy as np
import keras
from keras import optimizers
from keras.models import Sequential
from keras.layers import Dense,Dropout,Reshape,Activation,Conv2D,MaxPooling2D
from numpy import save, load
import argparse

ch = 1
#run code with this command, specifying your own data path
#python train_cnn.py ~/data/data_signal.npy ~/data/data_noise.npy

# Parse eventfile as argument
parser = argparse.ArgumentParser(description='dl training input parameters')
parser.add_argument('input_signal', type=str,
                    help='Path to signal file plus name of signal file')
parser.add_argument('input_noise', type=str,
                    help='Path to noise file plus name of noise file')
args = parser.parse_args()

input_s = np.load(args.input_signal)
input_n = np.load(args.input_noise)


for i in range(input_s.shape[1]):
  mask = np.ones(input_s.shape[0], dtype=np.bool)
  for i in np.argwhere(np.sum(np.isnan(input_s), axis=2)[:, i] > 0)[:, 0]:
    mask[i] = False
  input_s = input_s[mask]

noise = np.reshape(input_n, (input_n.shape[0], input_n.shape[2], 1, 1))
signal = np.reshape(input_s, (input_s.shape[0], input_s.shape[2], 1, 1))
print(input_n.shape,input_s.shape)

x = np.vstack((signal, noise))
n_samples = x.shape[1]
n_channels = x.shape[2]
y = np.vstack((np.ones((signal.shape[0], 1)), np.zeros((noise.shape[0], 1))))

# randomize order of signal and noise events
s = np.arange(x.shape[0])
np.random.shuffle(s)
x = x[s]
y = y[s]

# will need to tune epoch (and maybe batch size) to train to optimally
BATCH_SIZE = 32
EPOCHS = 19
```

```
def training():

  model = Sequential()
  # n_filters, n_width. Default stride 1 and no padding
  model.add(Conv2D(5,(10, ch), activation='relu',
  input_shape=(n_samples, n_channels, 1)))
  model.add(Dropout(0.5))
  model.add(MaxPooling2D(pool_size=(10, 1)))
  model.add(Reshape((np.prod(model.layers[-1].output_shape[1:]),)))# Flatten
  model.add(Dense(1, activation='sigmoid'))
  opt = keras.optimizers.Adam(learning_rate=0.001) # default 0.001.
  model.compile(optimizer=opt,
                loss='binary_crossentropy',
                metrics=['accuracy'])
  model.fit(x,
            y,
            batch_size=BATCH_SIZE,
            epochs=EPOCHS,
            validation_split=0.2,
            verbose=1)

  model.summary()
  model.save('trained_CNN.h5')


training()
```

A.3 Basic fully connected neural network training code

```
import numpy as np
import keras
import os
from keras.models import Sequential, load_model
from keras.layers import Dense, Activation, Lambda
from keras.backend import argmax
from keras import backend as K
from matplotlib import pyplot as plt

path = "~/data"  # path to input file
n = np.load(os.path.join(path, "data_noise_1ch_3.6SNR_0000.npy"))
signal = np.load(os.path.join(path, "data_signal_1ch_0000.npy"))
# remove nans channels
for i in range(signal.shape[1]):
    mask = np.ones(signal.shape[0], dtype=np.bool)
    for i in np.argwhere(np.sum(np.isnan(signal), axis=2)[:, i] > 0)[:, 0]:
        mask[i] = False
    s = signal[mask]

signal = np.reshape(s, (s.shape[0], s.shape[1] * s.shape[2]))
noise = np.reshape(n, (n.shape[0], n.shape[1] * n.shape[2]))

print(signal.shape)
print(noise.shape)

x = np.vstack((noise, signal))
in_dim = x.shape[1]
y = np.vstack((np.zeros((noise.shape[0],1)),np.ones((signal.shape[0],1))))

# randomly shuffles data
s = np.arange(x.shape[0])
np.random.shuffle(s)
x = x[s]
y = y[s]

BATCH_SIZE = 32
EPOCHS = 10

def training():
    model = Sequential()
    #64 nodes in this layer, can be adjusted
    model.add(Dense(64, input_dim=in_dim))
    model.add(Activation('relu'))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(optimizer='Adam',
                  loss='binary_crossentropy',
                  metrics=['accuracy'])
    model.fit(x,y,validation_split=.1,epochs=EPOCHS,batch_size=BATCH_SIZE)
    model.summary()
    model.save('trained_network_nn_1layer64nodes.h5')

training()
```

A.4 Floating Point Operations (FLOPs) calculations

```
from matplotlib import pyplot as plt
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.python.framework.convert_to_constants
import convert_variables_to_constants_v2_as_graph

def get_flops(concrete_func):
    frozen_func, graph_def =
    convert_variables_to_constants_v2_as_graph(concrete_func)
    with tf.Graph().as_default() as graph:
        tf.graph_util.import_graph_def(graph_def, name='')

        run_meta = tf.compat.v1.RunMetadata()
        opts = tf.compat.v1.profiler.ProfileOptionBuilder.float_operation()
        flops = tf.compat.v1.profiler.profile(graph=graph,run_meta=run_meta,
        cmd="op", options=opts)
        return flops.total_float_ops

initial_model = keras.Sequential(
    [  # CNN 1layer
        layers.Conv2D(5, (20, 1), activation='relu',
        strides=1, input_shape=(100, 1, 1)),
        layers.MaxPooling2D((20, 1)),
        layers.Flatten(),
        layers.Dense(1, activation='sigmoid')

        # NN
        # layers.Dense(128, input_dim=512),
        # layers.Activation('relu'),
        # layers.Dense(1, activation='sigmoid')

        # # CNN 2 layer
        # layers.Conv2D(10,(10,1),activation='relu',input_shape=(100,1,1)),
        # layers.Dropout(0.4),
        # layers.Conv2D(10,(10,1),activation='relu',input_shape=(100,1,1)),
        # layers.Dropout(0.4),
        # layers.MaxPooling2D((10, 1)),
        # layers.Flatten(),
        # # layers.Dense(50, activation='relu'),
        # layers.Dense(1, activation='sigmoid')  ])

model = keras.Model(
    inputs=initial_model.inputs,
    outputs=[layer.output for layer in initial_model.layers],)
concrete = tf.function(lambda inputs: model(inputs))
concrete_func = concrete.get_concrete_function(
    [tf.TensorSpec([1, *j.shape[1:]]) for j in initial_model.inputs])

print(get_flops(concrete_func))
```

A.5 Manual calculation of FLOPs (an illustration)