Università degli studi di Padova

Dipartimento di Fisica e Astronomia "Galileo Galilei"

Master Degree in Astrophysics and Cosmology

Final dissertation

FlareWatch: Toward Open-Access IceCube Gamma Follow-Up Alerts and Collaborative Analysis

Supervisor
Elisa Bernardini

Co-supervisor
Sarah Louise Mancina

Candidate
Sergio Cuenca Lobo

Academic Year
2024-2025

"Per aspera ad astra"

— Someone

Abstract

Astrophysical neutrinos offer a unique window into the most energetic and distant phenomena in the universe. Detecting them in coincidence with gamma rays is a central goal of modern multimessenger astronomy, enabling the discovery of signals that may not be statistically significant on their own. IceCube, a neutrino observatory located at the South Pole, plays a key role in this effort by issuing real-time alerts when high-energy neutrino events are detected.

A key component of this system is the Gamma Follow-Up (GFU) program, which identifies time-clustered neutrino events potentially associated with gamma-ray sources. Traditionally, GFU alerts were distributed privately via a legacy system based on email notifications, accessible only to a restricted network of partner institutions. Dedicated follow-up reports were manually prepared and sent after the fact, limiting the speed and reach of the information. In recent years, there has been a growing effort to modernize this process and make GFU alerts publicly available and easier to access.

This thesis directly contributes to that transition by introducing a modern web-based platform that brings IceCube's GFU cluster alerts into the public domain: the *FlareWatch*. This new platform receives alerts in real time and presents them through an interactive and user-friendly interface, highlighting their significance, timing, and potential association with astrophysical sources. In addition to displaying alerts, the system performs on-the-fly statistical analyses, such as dynamic significance curves, providing deeper insights into the temporal evolution of each source. It also maintains an archival record of alerts, enabling temporal comparisons and source-specific monitoring. All of this is achieved while remaining fully integrated with IceCube's real-time infrastructure, yet lightweight and fully autonomous.

At the time of writing, the IceCube FlareWatch is feature-complete and well-tested, requiring only a brief internal review prior to public deployment. This work marks a concrete step toward a more open, transparent, and connected multimessenger approach.

Contents

Listing of figures

Listing of tables

Listing of acronyms

AGN ..... Active Galactic Nuclei

AJAX ..... Asynchronous Javascript and XML

AMANDA ..... Antarctic Muon And Neutrino Detector Array

API ..... Application Programming Interface

ANTARES ..... Astronomy with a Neutrino Telescope and Abyss Environmental Research

ATWD ..... Analog Transient Waveform Digitizer

BDT ..... Boosted Decision Tree

CC ..... Charged Current

CMB ..... Cosmic Microwave Background

CSP ..... Content Security Policy

CSS ..... Cascading Style Sheets

CTA ..... Cherenkov Telescope Array

DAQ ..... Data Acquisition System

DIS ..... Deep Inelastic Scattering

DOM ..... Digital Optical Module

DUMAND ..... Deep Underwater Muon And Neutrino Detector Project

EAS ..... Extensive Air Shower

EBL ..... Extragalactic Background Light

FAR ..... False Alarm Rate

fADC ..... fast Analog to Digital Converter

GCN ..... General Coordinates Network

GRB . . . . . Gamma-Ray Burst

GFU . . . . . Gamma Follow-Up

GZK . . . . . Greisen-Zatespin-Kuzmin

HESE . . . . . High-Energy Starting Event

H.E.S.S. . . . . . High Energy Stereoscopic System

HLC . . . . . Hard Local Coincidence

HTML . . . . . HyperText Markup Language

HTTP . . . . . HyperText Transfer Protocol

IACT . . . . . Imaging Atmospheric Cherenkov Telescope

JS . . . . . JavaScript

JSON . . . . . JavaScript Object Notation Format

KM3NeT . . . . . Cubic Kilometre Neutrino Telescope

LHAASO . . . . . Large High Altitude Air Shower Observatory

MAGIC . . . . . Major Atmospheric Gamma-Ray Imaging Cherenkov

MuEX . . . . . Muon Energy eXtractor

MPE . . . . . Multi Photo-Electron

MVC . . . . . Model-View-Controller

NC . . . . . Neutral Current

PEP . . . . . Python Enhancement Proposal

PDF . . . . . Probability Distribution Function

PMT . . . . . Photomultiplier Tube

PnF . . . . . Processing and Filtering

PTF . . . . . Palomar Transient Factory

RAM . . . . . Random-access memory

REST . . . . . Representational State Transfer

ROTSE . . . . . Robotic Optical Transient Search Experiment

SMT . . . . . Simple Multiplicity Trigger

SNR . . . . . Supernova Remnants

SPE . . . . . Single Photo-Electron

UHECR . . . . . Ultra High Energy Cosmic Rays

UTC . . . . . Universal Time Coordinated

VERITAS . . . . . Very Energetic Radiation Imaging Telescope Array System

VHE . . . . . Very High Energy

WSGI . . . . . Web Server Gateway Interface

XML . . . . . eXtensible Markup Language

1 Introduction

The history of high-energy astrophysics began over a century ago, when Austrian physicist Victor Hess embarked on a series of balloon flights to measure atmospheric ionization [1]. His 1912 discovery of an increase in radiation with altitude marked the first detection of what we now call *cosmic rays*—charged particles reaching Earth from outer space. This opened the door to a new way of studying the universe, beyond visible light.

Since then, astrophysics has evolved into a discipline that observes the cosmos through many different "messengers": photons across the electromagnetic spectrum, cosmic rays, neutrinos, and, more recently, gravitational waves. Each messenger offers a different window into the physical conditions and processes taking place in extreme astrophysical environments.

One of the central challenges of modern astrophysics is to identify the origin and mechanisms of these high-energy phenomena. While electromagnetic observations provide precise spatial and spectral data, they are often limited by absorption or background. Cosmic rays, on the other hand, are charged and therefore deflected by magnetic fields, losing directional information. This is where the concept of *multimessenger astrophysics* becomes essential.

1.1 Multimessenger Astrophysics

Multimessenger astrophysics aims to combine information from different types of cosmic messengers such as photons, cosmic rays, neutrinos, and gravitational waves, in order to obtain a

more complete understanding of the most energetic processes in the universe [2]. Each messenger is sensitive to different physical environments and mechanisms, and their joint detection can reveal aspects that would remain hidden in single-channel observations.

In the context of this thesis, the focus lies on understanding the origin of high-energy neutrinos and their potential connection to gamma rays and cosmic rays. These neutrinos are expected to originate from hadronic processes occurring in astrophysical accelerators, through two main channels:

- **Photohadronic interactions** ($p + \gamma$), where a high-energy proton collides with a background photon, producing a *delta resonance* (Δ^+) [3]:

$$p + \gamma \to \Delta^+ \to \begin{cases} p + \pi^0 \\ n + \pi^+ \end{cases} \tag{1.1}$$

- **Hadronuclear interactions** ($p + p$), where two energetic protons collide, producing similar secondary particles.

These processes lead to the production of pions. Neutral pions decay into gamma rays, while charged pions decay into muons and neutrinos:

$$\pi^0 \to \gamma + \gamma, \quad \pi^\pm \to \mu^\pm + \nu_\mu(\bar{\nu}_\mu) \quad \text{and} \quad \mu^\pm \to e^\pm + \nu_e(\bar{\nu}_e) + \bar{\nu}_\mu(\nu_\mu). \tag{1.2}$$

Therefore, any source capable of accelerating protons to sufficiently high energies and embedded in radiation or matter fields should produce both gamma rays and neutrinos. This connection forms the foundation of this multimessenger search strategy.

In addition, some of these energetic protons and heavier nuclei reach the Earth directly, forming what we observe as *cosmic rays*.

1.2 Cosmic Rays

Cosmic rays are relativistic ions—mostly protons and heavier nuclei—that travel through space and reach the Earth [5]. Since they are charged, their paths are strongly affected by galactic and extragalactic magnetic fields, making it nearly impossible to trace their arrival direction back to the source. Their main property is their energy spectrum, which spans many orders of magnitude and approximately follows a power law in the form of [6]:

$$\frac{d\varphi}{dE} \propto E^{-\gamma} \tag{1.3}$$

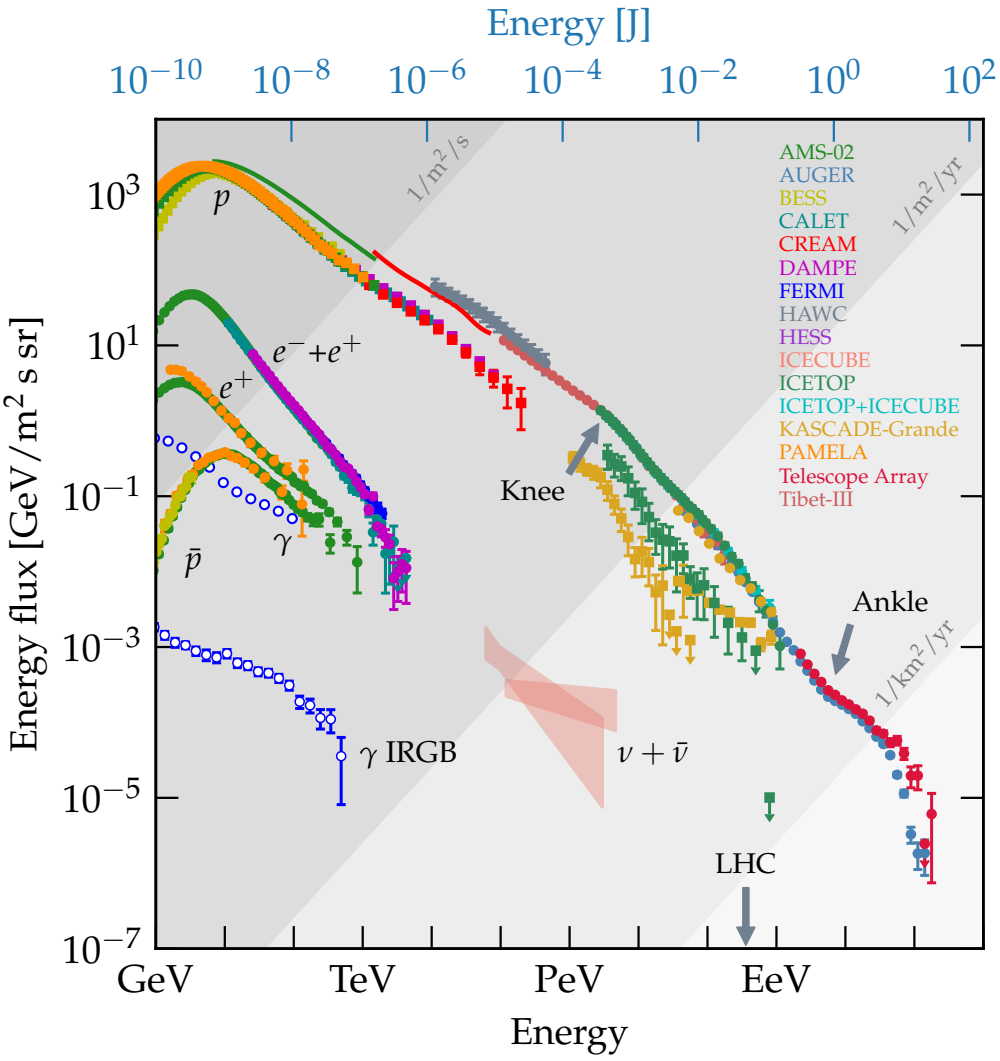


Figure 1.1: Energy spectra of high-energy particles compiled from multiple experiments. The cosmic-ray flux extends across more than ten orders of magnitude in energy. The different curves correspond to protons (p), antiprotons ($\bar{p}$), electrons (e^-), positrons (e^+), gamma rays (γ), and neutrinos ($\nu+\bar{\nu}$). Grey bands indicate the approximate detection capabilities of current instruments, and colored data points correspond to specific observatories listed in the legend. Taken from [4].

where φ refers to the flux and γ to the *spectral index* which is in between 2.5 and 3.3, depending on energy E. As shown in Figure 1.1, this spectrum shows two main structures:

- **The knee** ($\sim 10^{15}$–10^{16} eV), where the spectrum steepens from $\gamma \approx 2.7$ to $\gamma \approx 3.1$. This softening is commonly interpreted as the point where galactic magnetic fields can no longer confine the most energetic cosmic rays, which begin to escape the Galaxy. It

may also reflect the maximum acceleration energy of typical Galactic sources such as supernova remnants [3].

- **The ankle** ($\sim 10^{18}$ eV), a spectral hardening that is often interpreted as a transition to an extragalactic cosmic-ray population [3]. However, its origin is still uncertain. Above this energy, cosmic rays are called Ultra High Energy Cosmic Rays (UHECR).

At energies above $\sim 5 \times 10^{19}$ eV, the flux drops sharply. The origin of this behavior is still unclear, but is typically justified through the Greisen-Zatespin-Kuzmin (GZK) cut-off, which explains this as a consequence of high energy protons interacting with the cosmic microwave background (CMB) photons via the same photohadronic channel mentioned earlier [7, 8]. On the other hand, there is also the possibility of this being the maximum energy cosmic accelerators could produce.

1.2.1 Extensive Air Showers and Secondary Particle Cascades

When a cosmic ray reaches the Earth's atmosphere, it typically interacts with a nucleus at high altitude, initiating a chain of secondary interactions known as an *extensive air shower* (EAS). The primary particle produces pions, kaons and other hadrons, which in turn decay or interact, generating cascades of particles. The resulting shower has three main components:

- **Hadronic component:** composed of baryons and mesons, including π^0, $\pi^\pm$, and nucleons.
- **Electromagnetic component:** from the decay of $\pi^0 \to \gamma\gamma$, leading to electromagnetic cascades.
- **Muon–neutrino component:** from the decay of $\pi^\pm$ and subsequent muon decays.

This is illustrated in Figure 1.2. The neutrinos and some muons can travel long distances and even reach detectors underground or under ice, contributing to the *atmospheric background* seen in neutrino observatories like IceCube, which will be explained in detail in Chapter 2 and 3.

1.3 Gamma Rays

In addition to neutrinos, high-energy gamma rays are one of the most relevant messengers for studying hadronic processes in astrophysical sources [10]. Gamma rays are photons, meaning

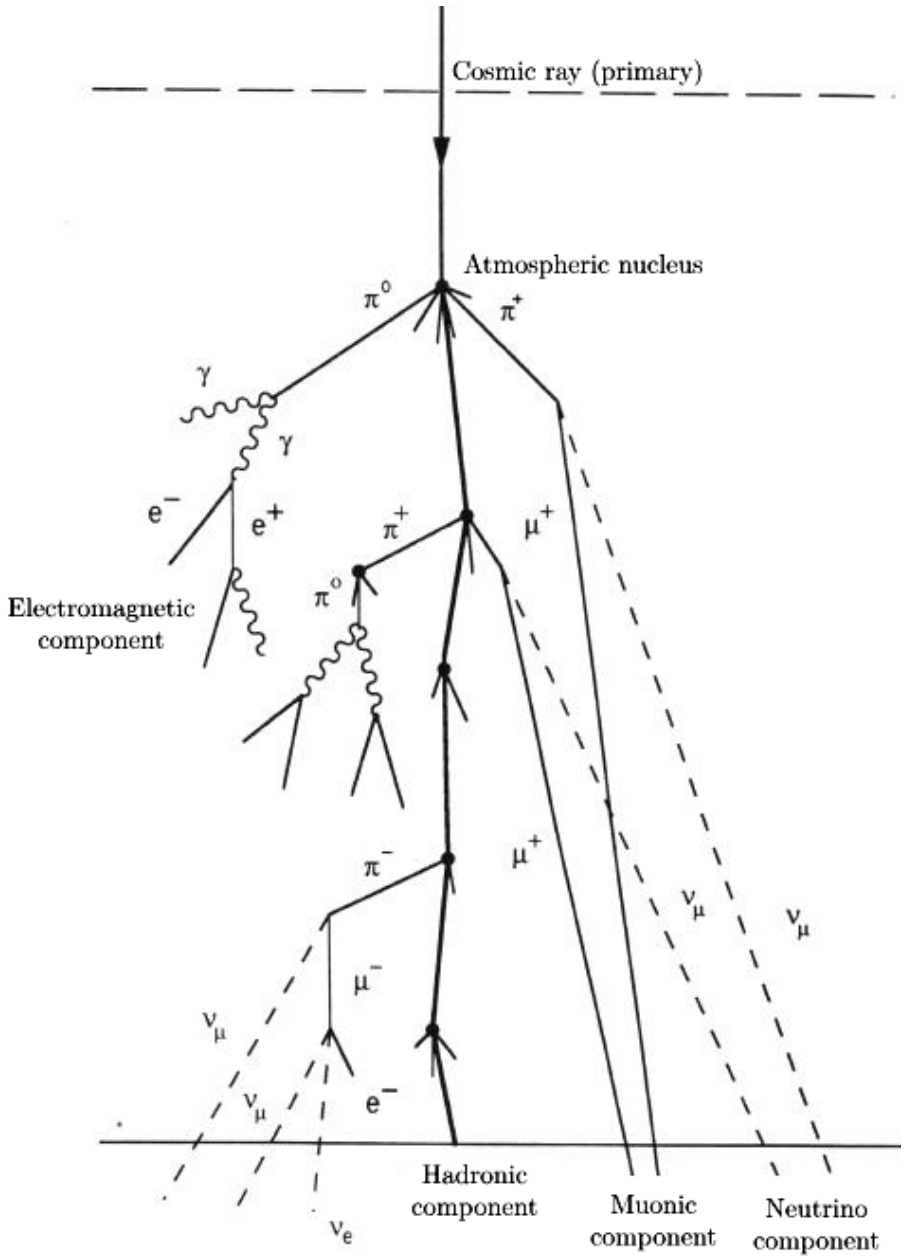


Figure 1.2: Schematic representation of an extensive air shower initiated by a cosmic-ray primary. The cascade develops through hadronic and electromagnetic interactions, generating muons, neutrinos, photons, and electrons. Modified from [9].

they are not deflected by magnetic fields and can be produced through two classes of mechanisms [11]:

- **Leptonic mechanisms:** Involving high-energy electrons accelerated in astrophysical en-

vironments, which can produce gamma rays via inverse Compton scattering off ambient photon fields or via synchrotron radiation under the exposure to magnetic fields.

- **Hadronic mechanisms:** Involving the interactions of high-energy protons or nuclei with ambient gas or radiation. These interactions can generate pions, and the decay of neutral pions ($\pi^0 \rightarrow \gamma + \gamma$) produces gamma rays [3]. This channel is particularly important for multimessenger searches, as it occurs in parallel with the production of neutrinos via charged pion decay.

Despite their ability to travel in straight lines, gamma rays are subject to energy-dependent attenuation during propagation through space. At energies above a few tens of GeV, gamma rays begin to interact with the diffuse photon backgrounds such as the CMB and the Extragalactic Background Light (EBL). These interactions, mainly pair production ($\gamma + \gamma_{\mathrm{EBL}} \rightarrow e^+ + e^-$), limit the distance from which gamma rays can reach Earth. The higher the photon energy, the shorter the horizon of detectability. This propagation constraint defines four observational regimes [3, 10]:

- **Low-energy (LE) gamma rays** (0.5 MeV to ∼100 MeV) interact predominantly via the photoelectric effect and Compton scattering. In this regime, gamma-ray telescopes often employ high-Z scintillator crystals, that efficiently absorb photons and re-emit the energy as visible light. This light is then detected by photomultiplier tubes or silicon photodiodes, allowing for energy reconstruction. Space-based instruments also include anti-coincidence shielding to suppress the charged-particle background.

- **High energy (HE) gamma rays** (0.1 GeV to ∼100 GeV), which are primarily detected by pair conversion telescopes, where the incoming photon converts into an electron-positron pair in a tracker, followed by energy measurement in a calorimeter. Instruments such as the Fermi Large Area Telescope (Fermi-LAT) [12] operate in this regime, providing wide field-of-view observations and all-sky monitoring from orbit.

- **Very High Energy (VHE) gamma rays** (0.1 TeV to ∼100 TeV), which are detected from the ground using Imaging Atmospheric Cherenkov Telescopes (IACTs) like the Major Atmospheric Gamma-Ray Imaging Cherenkov (MAGIC) [13] in La Palma, Spain, the High Energy Stereoscopic System (H.E.S.S.) [14] in Namibia, and the Very Energetic Radiation Imaging Telescope Array System (VERITAS) [15] in Arizona. When these gamma rays enter the atmosphere, they produce electromagnetic showers that emit Cherenkov light, which is then focused by mirrors onto photomultiplier arrays.

- **Ultra-High Energy (UHE) gamma rays** (above 100 TeV), which are extremely rare and require observatories with vast effective areas, such as The Large High Altitude Air Shower Observatory (LHAASO) in Sichuan, China [16]. Due to the strong attenuation at these energies, only galactic sources, relatively nearby, can be observed in this range.

Because IACTs have a narrow field of view and operate in pointed mode, real-time triggers from other messengers (such as neutrinos) are essential for maximizing their scientific output. In this sense, multimessenger coordination enables fast repointing and time-critical follow-up observations. This synergy is at the heart of the work presented in this thesis and the core of the real-time system explained in Chapter 3.

1.4 Neutrinos

Among all cosmic messengers, neutrinos (ν) are particularly valuable for probing the innermost regions of astrophysical sources. Neutrinos are electrically neutral, nearly massless leptons that interact only via the weak force. As a result, they can escape dense environments where photons may be absorbed, and travel cosmological distances without being deflected or scattered. This unique property allows them to carry directional information directly from their production site.

Neutrinos come in three flavors—electron (ν_e), muon (ν_μ), and tau (ν_τ)—defined by the charged lepton involved in their interaction. Although the Standard Model originally predicted them to be massless, experimental evidence of *neutrino oscillations*—the ability to change flavor during propagation—implies that they have a small but nonzero mass. There are two classifications based on their origin:

Atmospheric Neutrinos. When cosmic rays interact with nuclei in the Earth's atmosphere, they produce extensive particle showers that include pions and kaons. The decay of these mesons leads to the production of neutrinos, primarily in the sub-TeV to few hundred TeV range:

$$\pi^\pm \to \mu^\pm + \nu_\mu(\bar{\nu}_\mu), \quad \mu^\pm \to e^\pm + \nu_e(\bar{\nu}_e) + \bar{\nu}_\mu(\nu_\mu)$$

These *atmospheric neutrinos* serve both as a useful probe of neutrino properties, such as oscillations and as a dominant background in the search for astrophysical neutrinos.

Astrophysical Neutrinos. At higher energies, neutrinos are produced in astrophysical environments where cosmic rays interact with surrounding matter or radiation fields. These interactions generate light mesons (pions, kaons) whose decays yield high-energy neutrinos:

$$p + \gamma/N \to \pi^\pm + \text{others} \to \nu + \mu + e^\pm + \gamma$$

This production channel is tightly connected to that of gamma rays via the decay of neutral pions ($\pi^0 \rightarrow \gamma + \gamma$). The joint detection of neutrinos and gamma rays therefore provides compelling evidence for hadronic acceleration processes.

Despite their low interaction probability, neutrinos can be detected through their secondary products. When a high-energy neutrino interacts with a nucleus in ice or water, it can produce a charged lepton that emits Cherenkov light as it travels faster than the phase velocity of light in the medium. Detecting this light provides a way to infer the energy and direction of the incoming neutrino.

The small cross section of neutrino interactions implies that very large detection volumes are needed to accumulate significant statistics. This has motivated the construction of large-scale neutrino telescopes such as IceCube [17] at the South Pole, and the Cubic Kilometre Neutrino Telescope [18] (KM3NeT) in the Mediterranean.

Neutrinos are elusive, but their clean directional information and hadronic production mechanisms make them a powerful trigger for multimessenger follow-up. In particular, at very high energies, Cherenkov telescopes need to be pointed toward the right region of the sky. Neutrino alerts, like those generated by IceCube, enable such repointing and provide early warning of possible cosmic-ray acceleration sites. This is what motivates the technological and scientific efforts described in this thesis.

1.5 Cosmic Accelerators

Identifying astrophysical sources capable of producing high-energy neutrinos, gamma rays, and cosmic rays is one of the central goals of multimessenger astrophysics. The presence of high-energy particles implies the existence of powerful acceleration mechanisms operating in extreme environments throughout the universe. These environments must satisfy the conditions for efficient particle acceleration and provide the necessary targets, such as gas or radiation fields, for hadronic interactions to occur.

Several classes of objects have been proposed as potential sources. While each exhibits different physical conditions, they all share some key characteristics: compactness, high magnetic fields, and relativistic motion. Below we summarize the most important ones.

1.5.1 Active Galactic Nuclei and Blazars

Active Galactic Nuclei (AGN) are among the most luminous persistent sources in the universe. Powered by accretion onto supermassive black holes (mass in the range of $10^6 - 10^{10} M_\odot$), AGN produce intense electromagnetic emission and, in some cases, launch relativistic jets. When these jets are aligned with our line of sight, the source is classified as a *blazar*, which exhibits variability across the entire electromagnetic spectrum [10, 19].

Blazars are compelling candidates for high-energy neutrino production due to their energetic jets and dense photon fields. Hadronic interactions between accelerated protons and ambient radiation can generate charged and neutral pions, leading to the production of both neutrinos and gamma rays. This mechanism has been investigated in the context of events such as the 2017 IceCube neutrino alert regarding a gamma-ray flare observed from TXS 0506+056 [20].

However, recent observations suggest that gamma-ray brightness is not a prerequisite for neutrino emission. The nearby Seyfert II galaxy NGC 1068 was emerged as a promising neutrino source despite its gamma-ray faintness [21]. In this case, the high-energy photons are believed to be absorbed or reprocessed into X-rays by the dense material near the AGN core, while neutrinos escape unimpeded. This challenges prior assumptions and motivates broader searches across the electromagnetic spectrum, especially in X-rays.

1.5.2 Supernova Remnants

Supernova explosions drive powerful shock waves that can accelerate particles via first-order Fermi acceleration [22]. These shocks expand into the interstellar medium and can produce cosmic rays up to at least PeV energies. In the presence of dense molecular clouds, hadronic interactions can also generate gamma rays and neutrinos.

Recent gamma-ray observations of supernova remnants (SNRs) by Fermi-LAT have confirmed the decay of neutral pions in these environments, supporting their role as galactic cosmic-ray accelerators [23]. While SNRs are not expected to be strong extragalactic neutrino sources, they provide a crucial benchmark for understanding acceleration processes.

1.5.3 Gamma-Ray Bursts

Gamma-Ray Bursts (GRBs) are transient events that release enormous amounts of energy in short timescales [10, 3]. Long-duration GRBs are associated with the collapse of massive stars, while short GRBs are linked to the merger of compact objects such as neutron stars. Both sce-

narios involve relativistic jets and intense radiation fields, making them ideal sites for hadronic processes and efficient particle acceleration.

The potential of GRBs as high-energy neutrino sources has motivated extensive searches with detectors like IceCube. So far, no statistically significant association has been confirmed, but their short durations and high energies make them important targets for real-time neutrino follow-up observations.

A recent benchmark case is GRB 221009A, the brightest gamma-ray burst ever observed, which triggered an extensive multimessenger follow-up campaign [24]. The IceCube Collaboration searched for neutrinos from this event over a wide range of energies, from MeV to PeV, using several dedicated analysis channels [25]. No neutrinos were detected, but the study set strong upper limits on how many neutrinos could have been emitted. These limits are important because they constrain the efficiency of particle acceleration in GRBs and help test theoretical models that predict neutrino production in these extreme environments.

1.6 Context

Each of these source classes contributes differently to the flux of neutrinos, gamma rays, and cosmic rays observed at Earth. However, detecting and associating individual neutrinos with their source remains a significant challenge due to their low interaction cross section and the presence of atmospheric backgrounds.

To address this, large-scale detectors like IceCube can operate in coordination with other observatories across the electromagnetic spectrum. Notably, ground-based gamma-ray telescopes such as MAGIC, H.E.S.S., and VERITAS face limitations due to weather, moonlight, and limited fields of view. These collaborations have explicitly requested extended source context to better decide whether to follow up neutrino alerts after delays.

In parallel, the observation of neutrino emission from NGC 1068, an X-ray bright but gamma-ray faint AGN, highlights the need to broaden the search space and encourage participation from observatories at all wavelengths. This realization has sparked a shift toward public dissemination of neutrino alerts, allowing fast and open multimessenger campaigns across the globe.

The development of a web-based platform for real-time GFU public alerts potentially associated with sources, as explored in this thesis, is a direct response to these scientific and logistical motivations. The next chapter introduces the IceCube Neutrino Observatory, the detector at

the heart of this effort. Chapter 3 presents the architecture of IceCube's real-time alert system and the types of alerts it produces. Chapter 4 details the design and implementation of the web-based platform, followed by Chapter 5, which describes the statistical analysis and visualizations developed to interpret incoming alerts. Finally, Chapter 6 offers a summary of the results and discusses future improvements. Additional implementation details and validation results are provided in the appendices.

2

The IceCube Neutrino Observatory

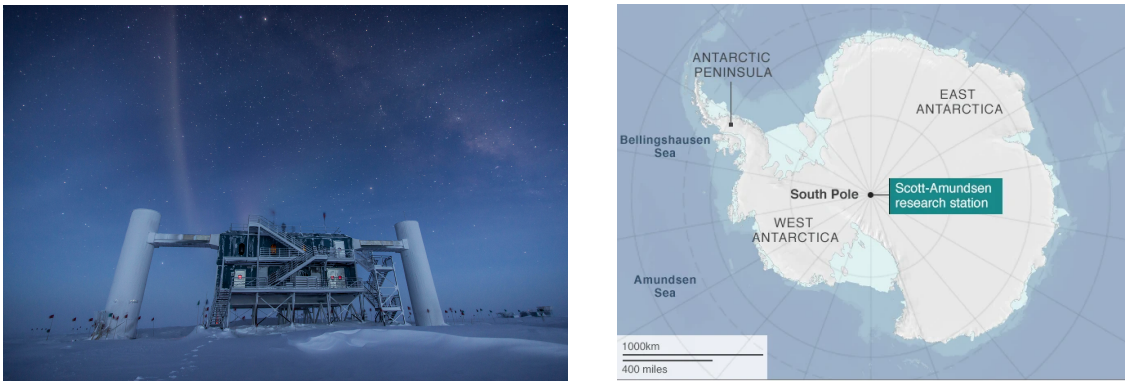
The flux of astrophysical neutrinos follows a power-law spectrum with a negative spectral index, resulting in a steep decline at the higher energies (as discussed in Section 1.2). Detecting a statistically significant number of weak interactions in this regime requires a detector with a large active volume, long observation time, and a transparent medium to reconstruct the charged secondaries via Cherenkov radiation.

Since the 1980s, several detectors have been proposed that exploit naturally transparent environments to observe high-energy neutrinos. Early efforts included the Deep Underwater Muon And Neutrino Detector Project (DUMAND) project, which attempted to deploy photomultiplier arrays across one cubic kilometer of ocean near Hawaii [26]; the Baikal experiment has been operating under the Siberian lake since 2003 [27]; the Astronomy with a Neutrino Telescope and Abyss Environmental Research (ANTARES) [28] and, more recently, KM3NeT [18] use the Mediterranean Sea; finally the Antarctic Muon And Neutrino Detector Array (AMANDA) [29] demonstrated the suitability of Antarctic ice and laid the groundwork for IceCube [17]. These efforts culminated in the realization that achieving a sufficiently large instrumented volume—on the order of one cubic kilometer—was essential to reach the sensitivity needed for detecting astrophysical neutrinos. These efforts shared a common design philosophy: the key to detecting the faint flux of astrophysical neutrinos lies in the total instrumented volume. This concept was further shaped by the theoretical Waxman–Bahcall bound [30], which places an upper limit on the diffuse neutrino flux from sources capable of accelerating ultra-high-energy cosmic rays. Achieving sensitivity near this bound requires

a detector with an active volume on the order of 1 km^3, a scale that became reality with the completion of IceCube in 2010. As the first observatory capable of detecting neutrino fluxes at the level predicted by this bound, IceCube's observation of a diffuse astrophysical neutrino flux provides strong evidence for the viability of neutrino astronomy and validates decades of theoretical and experimental effort [31].

2.1 Detector Design

IceCube is located at the Amundsen–Scott South Pole Station (see Figure 2.1b). Between 2005 and 2010, 86 holes approximately 60 cm in diameter were drilled using hot water, and 5,160 Digital Optical Modules (DOMs) were deployed between depths of 1,450 m and 2,450 m. The site was selected primarily for its enormous ice volume and its exceptional optical clarity, due to the pressure-driven removal of air bubbles during glacial compression [32].



(a) Picture of the IceCube Lab

(b) Position of the observatory in the Antarctica continent.

Figure 2.1: Picture of the IceCube Lab [33] (left) and geographical map of Antarctica [34] (right) showing the location of the Amundsen–Scott South Pole Station, where the observatory is installed.

The operating principle of IceCube relies on detecting Cherenkov light emitted when muons, taus, or electromagnetic/hadronic cascades propagate at superluminal speeds through the ice.

2.1.1 Cherenkov Radiation

When a charged particle travels faster than the speed of light in a medium (not in vacuum), it produces a shock wave of light, analogous to a sonic boom. This phenomenon, discovered by

Pavel Cherenkov in 1934 [35], results in a light cone whose opening angle θ_C satisfies [3]:

$$\cos\theta_C = \frac{1}{\beta n}, \tag{2.1}$$

where $\beta = v/c$ and n is the refractive index of the medium.

In Antarctic ice, where $n \approx 1.32$, this yields an angle of approximately 41° [36]. The emitted photons are mostly in the range 300–600 nm, well within both the transparency window of the ice [32] and the peak sensitivity of the DOMs [37].

2.1.2 Digital Optical Modules

The core sensing elements of the detector are the DOMs, a 33 cm diameter glass sphere that houses all the components required to transform individual photons into digitized signals [37].

Each DOM contains a 25 cm Hamamatsu R7081-02 photomultiplier tube (PMT), which operates via the photoelectric effect: incoming photons strike the photocathode, ejecting electrons that are then amplified through a cascade of dynodes. The result is an electrical pulse on the order of nanoseconds and tens of millivolts [38]. The quantum efficiency—i.e., the probability that an incident photon produces photoelectrons—peaks at 25–34% around 390 nm. A μ-metal shield (an alloy made of nickel and iron) surrounds the PMT to suppress the Earth's magnetic field ($\sim$ 60 μT at the South Pole) and prevent deflection of the electron trajectories [37].

The analog signal is processed by the DOM's Main Board, which provides high-voltage power to the PMT, manages timing, and performs digitization. A very low threshold (0.25 photoelectrons) is used to trigger readout. Upon triggering, the DOM stores 75 ns of pre-trigger and 6.4 μs of post-trigger data. Two parallel systems digitize the *waveform*:

- An Analog Transient Waveform Digitizer (ATWD) operating at 300 MegaSamples per second (MS/s) captures the fast component,
- A fast Analog-to-Digital Converter (fADC) at 40 MS/s records slower tails.

This dual system enables the detection of signals ranging from a single photoelectron to high-charge pulses generated from many photons from a large energy deposition.

Each DOM also includes a secondary circuit, the *flasher* board, containing twelve ultrafast LEDs. These can be triggered on command to emit known light patterns, used to calibrate the optical properties of the ice [39].

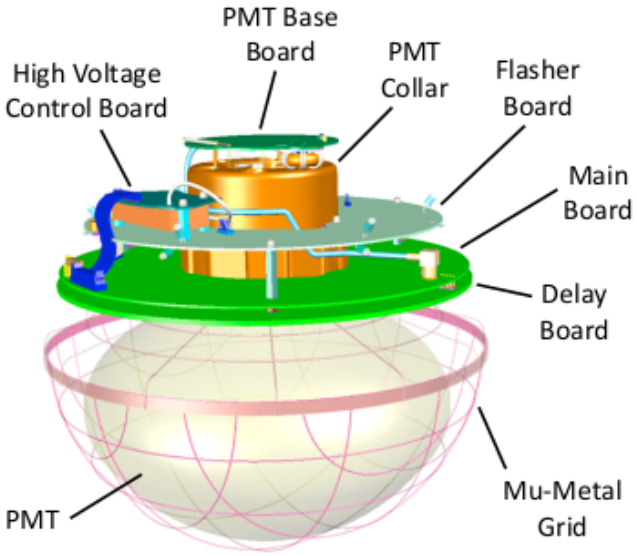


Figure 2.2: Internal components of an IceCube DOM. It houses a PMT and multiple printed circuit boards responsible for high-voltage supply, signal digitization, triggering, calibration, and timing. The μ-metal grid protects the PMT from Earth's magnetic field, and the flasher board contains 12 LEDs used for in situ optical calibration. Taken from [37].

They are also connected through a copper and fiber-optic cable that runs up to the surface. These cables transmit both power and data. At the surface, all strings terminate in the IceCube Laboratory (Figure 2.1a), which houses synchronization, readout, and control electronics.

2.1.3 Instrumented Geometry

There are 86 strings arranged in a nearly hexagonal grid with 125 m horizontal spacing. Each string holds 60 DOMs with vertical spacing of 17 m, defining an instrumented volume close to 1 km^3 (see Figure 2.3). Eight central strings form the *DeepCore* sub-array, with closer spacing (7–10 m vertical, 42–72 m horizontal), allowing sensitivity to neutrinos down to ∼10 GeV [40].

On the surface, the *IceTop* array consists of 162 ice-filled tanks, each with two DOMs. IceTop serves to detect cosmic-ray-induced air showers and to tag atmospheric muons that also pass through the main detector [41].

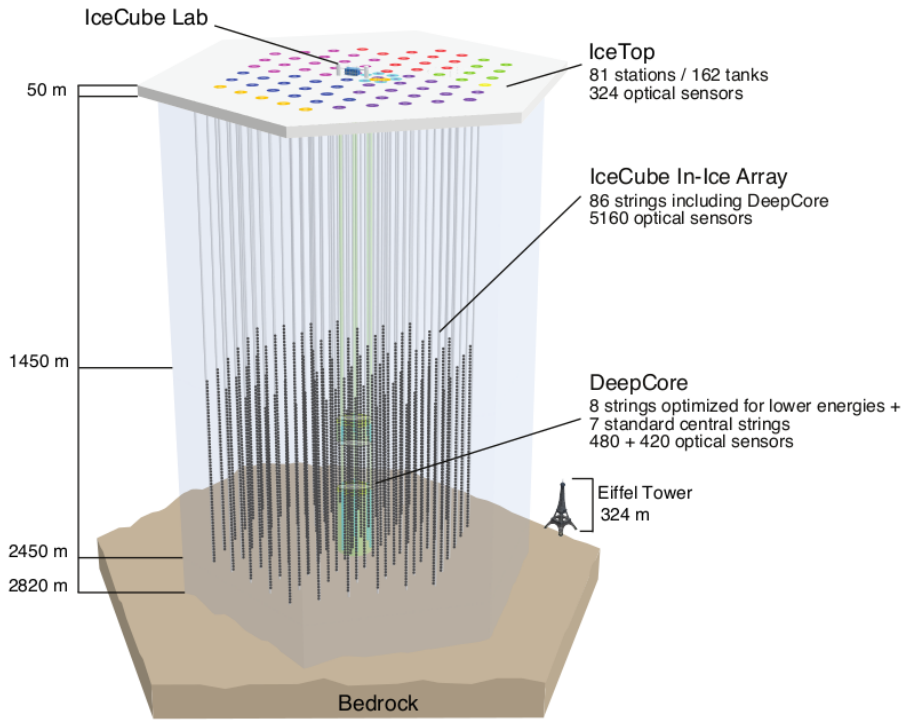


Figure 2.3: Diagram of the IceCube Neutrino Observatory. The detector comprises 86 strings equipped with 5,160 optical sensors deployed between depths of 1,450 m and 2,450 m in the Antarctic ice. Eight of these strings form the DeepCore sub-array, optimized for lower-energy neutrino detection. The IceTop surface array consists of 81 stations with 324 sensors and is in turn optimized for EAS. The Eiffel Tower is shown for scale. Taken from [37].

2.1.4 Optical Properties of the Ice

Antarctic glacial ice offers exceptional optical properties for Cherenkov detection. Due to slow compression over millennia, the ice is almost bubble-free, yielding absorption lengths exceeding 100 m and effective scattering lengths between 20 and 50 m in the blue-green range [32]. While there are localized dust layers, their distribution is well mapped using calibration lasers and incorporated into reconstruction algorithms. Comparative studies have shown that South Pole ice is optically clearer than artificial or lake ice, making it a nearly ideal natural medium [42].

2.2 Event Detection and Signal Types

IceCube continuously records single–photon hits in its DOMs. Whenever a space–time cluster of DOM pulses satisfies an online multiplicity condition (explained in detail in Section 3.1),

the corresponding waveforms are buffered and an event is assembled for subsequent processing [42].

2.2.1 Up-going and Down-going events

The reconstructed arrival direction of each event is expressed in IceCube's internal spherical coordinates:

- **Zenith angle** θ: 0° points from the Earth's centre towards the sky; 180° points into the ground.
- **Azimuth** φ: the projection of the arrival vector on the horizontal detector plane.

These angles divide the sky into two regions of interest (see Table 2.1). In the *up-going* hemisphere every particle must traverse the Earth before entering the detector, so the planet itself absorbs all but a small fraction of the atmospheric muons [43]. On the other hand, the *down-going* zenith range is filled from muons generated in cosmic–ray air showers. A conceptual sketch of this is shown in Figure 2.4.

Region	Typical range in θ	Key feature
Up-going	$\theta \gtrsim 82°$	Traverses the Earth first; the rock shield suppresses most of the atmospheric muons.
Down-going	$\theta \lesssim 82°$	Only the atmosphere and a few kilometres of ice are crossed; hence a copious bundle of atmospheric muons accompanies the signal.

Table 2.1: Zenith ranges and their descriptions.

Statistically, this configuration implies $\mathcal{O}(10^5)$ atmospheric neutrinos versus $\mathcal{O}(10^{11})$ atmospheric muons per year—roughly one million muons for every neutrino [37]. Discriminating the tiny but astrophysically valuable neutrino component requires exploiting their distinct signatures in the detector.

2.2.2 Neutrino interactions

In the 10 GeV–10 PeV range neutrinos interact with the Antarctic ice by deep inelastic scattering (DIS), giving rise to two fundamental channels [44]:

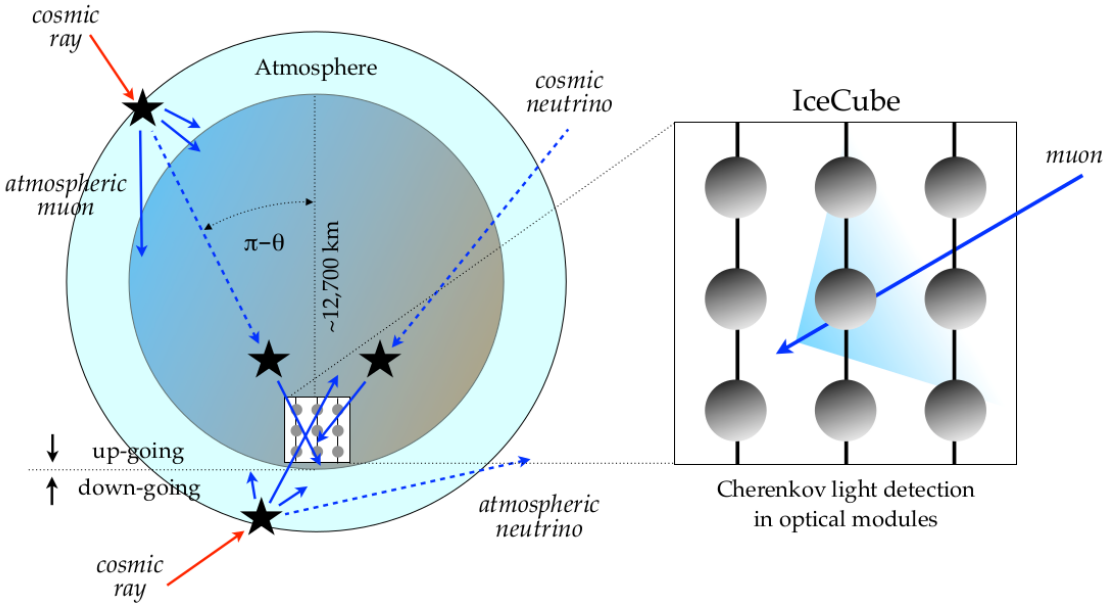


Figure 2.4: Schematic view of IceCube's horizon. Cosmic–ray induced atmospheric muons (solid blue) reach the detector only from above, whereas neutrinos (dashed blue) can traverse the Earth and appear as up-going. The right panel illustrates the detection of the resulting Cherenkov light in the optical modules. Taken from [17].

1. **Neutral current (NC)**

$$\nu_\ell + N \xrightarrow{Z^0} \nu_\ell + X',$$

where the outgoing neutrino escapes and the hadronic remnants X' create a *shower* of secondary particles. They interact through the Z^0 boson.

2. **Charged current (CC)**

$$\nu_\ell + N \xrightarrow{W^\pm} \ell + X',$$

producing a charged lepton $\ell = e, \mu, \tau$ in addition to the hadronic shower. The lepton species determines the observed topology: a track if ℓ is a muon, a (mostly) spherical shower if ℓ is an electron, or a *double bang* at PeV energies when ℓ is a tau, due to its short lifetime, responsible for the decay in the detector volume. They interact through the $W^\pm$ boson.

2.2.3 Signal types

Nearly every IceCube events falls into one of three generic signal classes. First, *muon tracks* arise predominantly from charged–current interactions of ν_μ. A relativistic muon may travel through the ice, sequentially activating strings and painting the characteristic track seen in Fig. 2.5 (a). The resulting long lever arm yields a median pointing precision better than 1° [45] and thus giving rise to a smoking-gun to identify astrophysical sources.

In contrast, *particle showers* (cascades) deposit their energy within only a few metres. They stem from neutral–current interactions as well as from charged–current events of ν_e and ν_τ, producing the spherical morphology shown in Figure 2.5 (b). Their symmetry grants an excellent calorimetric energy resolution of about 10–15 % [45], albeit at the price of a worse angular precision.

Finally, a multitude of background events, principally atmospheric muon bundles (see Figure 2.5 (c)), outnumbers genuine neutrino signatures. Therefore veto strategies are necessary to isolate the astrophysically interesting neutrino subsets.

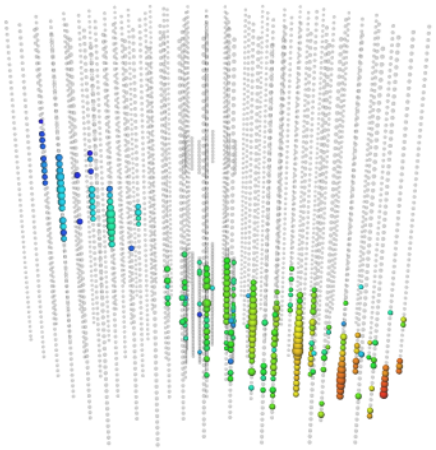
This thesis focuses on muon tracks, since their sub-degree angular resolution provides an unrivalled way for associating neutrinos with astronomical sources. For through-going tracks the energy estimate is only a lower bound (the parent interaction happens outside the instrumented volume), yet the pointing power more than compensates for this limitation. The core of the work therefore concentrates on identifying these high-quality muon tracks and forwarding their real-time alerts to partner observatories in the Northern hemisphere.

2.3 Reconstruction of Muon Tracks

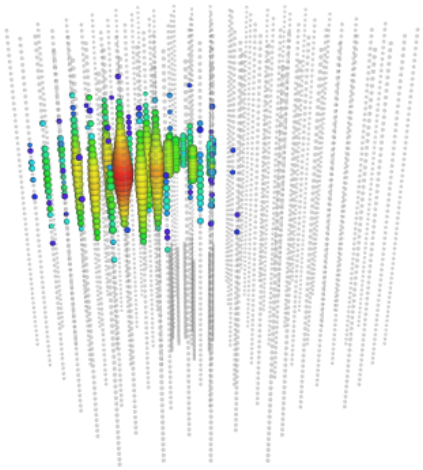
After introducing the detection mechanism and the relevance of track-like events, this section describes how the fundamental parameters of muon trajectories are reconstructed. This step is crucial for all subsequent analyses, as it determines the quality of the selected events and the effectiveness of data filters.

2.3.1 Direction

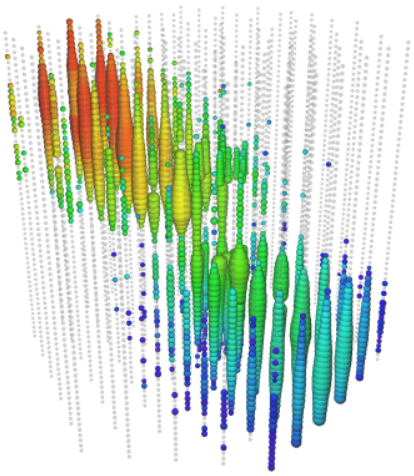
The muon direction is the core parameter of the analysis, as it allows pointing back to potential astrophysical sources of neutrinos. Along the reconstruction pipeline (detailed in Section 3.1), several methods are applied with increasing levels of complexity.



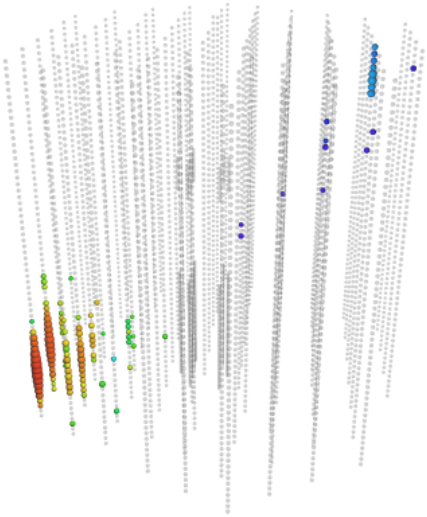
(a) Up-going ∼75 TeV muon track.



(b) Bright electromagnetic / hadronic shower.



(c) Atmospheric muon bundle from a 100 EeV proton.



(d) Two coincident atmospheric muons.

Figure 2.5: Representative IceCube events. Each dot denotes a DOM; its colour encodes the time of the first pulse (red = early, blue = late) and its area is the total collected charge. Panels (a) and (b) illustrate genuine neutrino signatures, whereas (c) and (d) showcase common atmospheric backgrounds. Taken from [46]

The simplest approach assumes a plane wave of light crossing the detector, without considering the characteristic Cherenkov cone angle (Section 2.1.1). Under this assumption, the muon

is considered to travel at constant velocity $\vec{v}$ along a straight line. Its position $\vec{r}$ is then given by:

$$\vec{r}(t) = \vec{r}_0(t_0) + (t - t_0)\vec{v}, \tag{2.2}$$

where $\vec{r}_0$ is the initial position and t_0 the arrival time. Based on the positions $\vec{x}_i$, and detection times t_i of the activated DOMs, a linear fit is performed to minimize the squared distance between the hypothesized straight track and the measured points:

$$\min_{\vec{v},\vec{r}_0} \sum_{i=1}^{N} \|\vec{r}(t_i) - \vec{x}_i\|^2, \tag{2.3}$$

where N is the total number of hit DOMs. This method can be solved analytically [47], providing a fast estimate of the muon direction and velocity. It also serves as the seed for more accurate reconstructions based on maximum likelihood methods [46].

The next level of modeling includes Cherenkov light emission, which forms a cone with a characteristic angle θ_c with respect to the muon direction. In this case, the expected arrival time of the light at each DOM, t_{geo}, assuming no scattering, is given by:

$$t_{\text{geo}} = t_0 + \frac{\vec{v} \cdot (\vec{x}_i - \vec{r}_0) + d \tan \theta_c}{c}, \tag{2.4}$$

where d is the perpendicular distance from the muon track to the DOM (see Figure 2.6 for visual reference), and c is the speed of light in vacuum. The *time residual* is then defined as:

$$t_{\text{res}} = t_{\text{obs}} - t_{\text{geo}}. \tag{2.5}$$

The probability $p_1(t_{\text{res}})$ of observing a given time residual allows constructing the likelihood function:

$$\mathcal{L} = \prod_{i=1}^{N_{\text{ch}}} p_1(t_{\text{res},i} \mid \vec{r}_0, \theta, \phi), \tag{2.6}$$

where N_{ch} is the number of DOMs that registered a signal, and θ, φ are the zenith and azimuth angles of the muon direction. This method, known as the Single Photo-Electron (SPE) fit, uses only the first photon detected by each DOM and is initialized with the result of the previous line fit.

The function p_1 is often approximated by the Pandel function, an analytical model that describes the photon delay distribution due to scattering in the ice [48]. It depends on the op-

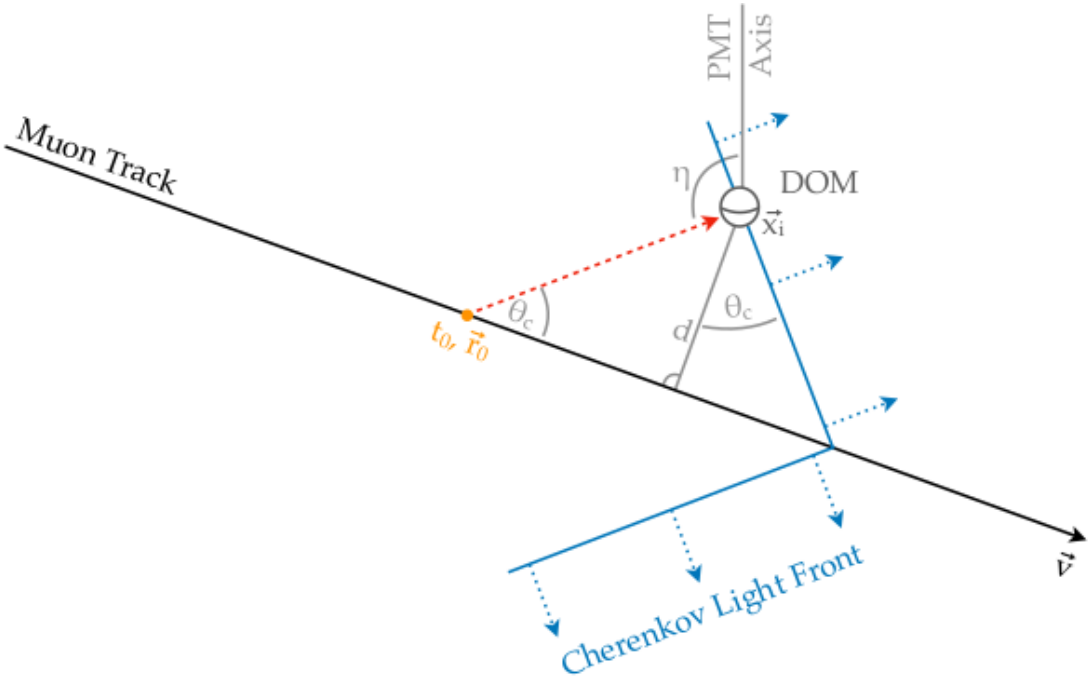


Figure 2.6: Geometry of the SPE reconstruction. The DOM detects Cherenkov light emitted at angle θ_c by a particle traversing the detector along direction $\vec{v}$. The variable d denotes the perpendicular distance from the DOM to the muon track. Taken from [46]

tical properties of the medium, such as the absorption coefficient and the effective scattering length [46].

However, this model becomes less accurate at higher energies, where multiple photons per DOM are typically observed. To address this, there is Multi Photo-Electron (MPE) fit, whose likelihood function accounts for all detected photons:

$$\mathcal{L} = \prod_{i=1}^{N} \left[N_i \cdot p_1(t_{\text{res},i}) \cdot \left(\int_{t_{\text{res},i}}^{\infty} p_1(t_{res,i}) \right)^{N_i - 1} \right], \tag{2.7}$$

where N_i is the number of photons detected by DOM i. This fit is also seeded with the result of the SPE fit.

Finally, instead of using analytical approximations like the Pandel function, the time residual distribution can be determined more precisely using detailed simulations of photon propagation in the ice. These simulations take into account the real optical structure of the Antarctic glacier. The resulting distributions are stored as interpolating splines [49, 50], a computationally efficient method that approximates functions using piecewise polynomials. This approach

significantly improves reconstruction accuracy by adapting the likelihood to the local conditions of the detector [46].

2.3.2 Angular Error

Another fundamental parameter in directional reconstruction is the angular error, which quantifies how precise the estimation of the muon direction is. This uncertainty must be computed by event, and it is essential both for event classification and for the subsequent source analysis.

The most computationally efficient method relies on the Cramér–Rao inequality [51, 52], which provides a lower bound for the variance achievable by any unbiased estimator of a parameter. An estimator is called unbiased if, on average over many repetitions of the experiment, it returns the true value of the parameter being estimated. In contexts where the reconstruction model is well defined and a sufficient number of measurements is available, this condition can reasonably be assumed to hold, making the bound applicable.

Under these assumptions, the covariance matrix of the directional parameters x_i and x_k is bounded from below by the inverse of the Fisher information matrix:

$$\mathrm{Cov}(\vec{x_i}, \vec{x_k}) \geq [\mathcal{I}(\vec{x})]^{-1}_{ik}\,, \tag{2.8}$$

where $\mathcal{I}(\vec{x})$ is defined as the expected value of the negative second derivative of the log-likelihood function:

$$\mathcal{I}_{ik}(\vec{x}) = -\left\langle \frac{\partial^2 \log \mathcal{L}(\vec{x} \mid t_{\mathrm{res}})}{\partial x_i \partial x_k} \right\rangle_{t_{\mathrm{res}}}. \tag{2.9}$$

The square roots of the diagonal elements of the covariance matrix provide estimates of the azimuthal (σ^2_φ) and zenith (σ^2_ϑ) variances, which can be combined into a single directional uncertainty metric, known as the circularized angular error [53]. This approach is extremely fast, as the covariance matrix can often be computed analytically [54]. It is therefore well suited for real-time reconstructions.

For offline analyses, the paraboloid method is used [46] to obtain a more detailed and empirical estimate of the angular error. This method relies on the fact that, when many measurements are available (e.g., many hit DOMs), the likelihood function tends to approximate a Gaussian shape near its maximum. The width σ_x with respect to a parameter x can then be estimated

using:

$$-\log \mathcal{L}(x \pm \sigma_x) = -\log \mathcal{L}(x) + 0.5. \tag{2.10}$$

In practice, the likelihood profile is evaluated in the zenith–azimuth plane around the reconstructed direction. A total of 24 sampling points are defined: three different values of zenith, and for each of them, eight equally spaced values of azimuth. At each point, the directional parameters are fixed, and the likelihood is re-optimized with respect to the remaining parameters. A 2D paraboloid is then fitted to the resulting log-likelihood values, producing an error ellipse in angular space. From the semiaxes σ_x and σ_y of the ellipse, the combined angular error is computed as [53, 55]:

$$\sigma = \sqrt{\frac{\sigma_x^2 + \sigma_y^2}{2}}. \tag{2.11}$$

Although this procedure involves multiple minimizations and is thus more computationally demanding, it provides a highly accurate estimate of the angular uncertainty, tailored to the event's geometry and signal. It is the standard method in high-quality offline reconstructions [53, 56].

A third approach, particularly useful when the theoretical uncertainty distribution is unknown or when capturing real statistical fluctuations is important, is the bootstrapping method. This technique generates artificial replicas of the original event by resampling with replacement from the set of detected pulses, while preserving the total charge.

Each bootstrap event is reconstructed independently, using the same method as for the original event. Once the directions of all bootstrap reconstructions are obtained, their angular differences with respect to the original direction are computed. The final angular error estimate is taken as the mean of these angular differences. This procedure is typically repeated six times in online analyses [46] and eight times in offline analyses [57].

2.3.3 Energy

In addition to direction, the energy of the observed muon is a key parameter, as it provides indirect information about the energy of the originating neutrino. In particular, energy is essential to distinguish between atmospheric and astrophysical neutrinos. This is because they follow different energy spectra: atmospheric neutrinos exhibit a steep power-law spectrum ($\sim E^{-3.7}$), while astrophysical neutrinos are expected to follow a harder spectrum ($\sim E^{-2}$). This spectral

difference allows the use of energy as a discriminator between both sources.

Since most neutrino interactions take place outside the instrumented volume, IceCube can only observe the muon once it enters the detector. As a result, the total neutrino energy cannot be measured directly in most of the cases. Instead, the goal is to estimate the muon energy at the entrance of the detector, which serves as a lower bound on the neutrino energy.

The total amount of detected Cherenkov photons increases with the muon energy [58]. However, due to photon scattering and absorption in the ice, as well as the geometry of the detector, this light must be carefully modeled in order to reconstruct the energy. The reconstruction relies on a template that encodes the expected number of photons Λ observed by each DOM for a muon of known energy and geometry. This template is scaled by a free energy parameter E, and compared to the observed number of photons k. The observed photon count is modeled as a Poisson process with mean:

$$\lambda = \Lambda E + \rho, \tag{2.12}$$

where ρ accounts for additional contributions such as instrumental noise [59]. The likelihood of observing k photons in a given DOM is then:

$$\mathcal{L}(E) = \frac{(\Lambda E + \rho)^k}{k!} \exp[-(\Lambda E + \rho)] \tag{2.13}$$

In practice, two main reconstruction methods are used to parameterize Λ:

MuEX. The Muon Energy eXtractor (MuEX) assumes that the muon emits Cherenkov light approximately uniformly along its trajectory. Based on the distance between the track and each DOM and the layered structure of the ice, it computes the expected number of photons and constructs the likelihood accordingly. This method accounts for both continuous energy losses and stochastic emissions [60, 59].

Truncated Energy. This alternative method focuses on reducing the impact of stochastic fluctuations, which can bias the energy estimate. The muon track is divided into spatial segments (typically 120 m in length to include one DOM), and the energy loss dE/dx is estimated independently in each segment using the same photon-based likelihood. The segments with the highest energy losses are discarded. The remaining segments are averaged to yield a robust energy estimate [61].

This segmented approach requires at least 300 m of visible muon track to be reliable, but

offers improved stability at high energies. For this reason, Truncated Energy is the preferred method for diffuse flux analyses or searches targeting very energetic events. In contrast, MuEX is more flexible and suitable for point-source searches or events with shorter tracks. [59]

The energy resolution of both methods has been thoroughly studied. At low energies, the resolution is limited by the low number of detected photons and by the almost constant energy loss rate of minimum-ionizing muons. At high energies, Truncated Energy benefits from its ability to resolve variations along the track, resulting in slightly better performance between 5 and 500 TeV. For shorter tracks or faster reconstructions, MuEX remains the method of choice.

3

The Realtime Alert System

The ability to rapidly identify and communicate high-energy neutrino events is one of the most powerful features of the IceCube Neutrino Observatory. The detector continuously records a high rate of events, most of which are due to atmospheric muons and neutrinos. The reconstruction techniques applied to muon track events, which are crucial for inferring the direction and energy of the incoming neutrinos, form the core of IceCube's realtime processing system.

Given the great volume of data collected, a highly selective data reduction pipeline is essential. This system filters the stream of reconstructed events in near real time, identifies those that are potentially of astrophysical origin, and disseminates alerts to external observatories to enable rapid multi-messenger follow-up.

This chapter provides an overview of the Realtime Alert System, with a particular focus on its architecture, the filtering pipeline, and the criteria used to classify and trigger different types of alerts. It also introduces the concept of continuous monitoring of predefined astrophysical sources, as well as the complementary unbiased or *all-sky* search and a selection of the most energetic track-like events, paving the way for the main contribution developed in the course of this work: FlareWatch, the publicly accessible web platform that enables realtime cluster alerts potentially associated with sources.

3.1 Alert Generation Pipeline

The primary goal of the alert generation system, in the context of multi-messenger astrophysics, is to transmit relevant information about neutrino events to external observatories in near real-time. This enables complementary observations using different messengers (photons, cosmic rays, gravitational waves), thereby increasing the statistical significance of potential discoveries. IceCube, with its unique capability of operating over 99% of the time, is ideally positioned to serve as an initial trigger in this global network of coordinated observations [62].

This effort began in 2006 with neutrino analysis and follow-up observations with MAGIC and AMANDA II [63], and later evolved into the current Gamma Follow-Up (GFU) program [64] between IceCube, VERITAS [15] and MAGIC. In 2008, the Optical Follow-Up (OFU) [65] program was added, incorporating instruments such as Robotic Optical Transient Search Experiment (ROTSE) [66], the Palomar Transient Factory (PTF) [67], and the Swift satellite [68]. These efforts eventually evolved into the current realtime alert system [62, 46], which is the core focus of this thesis.

The realtime system is built on a data-reduction pipeline that filters the signals recorded by the DOMs, starting from the initial detection stage up to the generation and dissemination of astrophysical alerts. The process begins with waveform digitization at the DOMs themselves, which transmit the signals through the cables to the central IceCube laboratory. There, the Data Acquisition System (DAQ) aggregates the signals into preliminary event structures.

These events are then passed to the Processing and Filtering (PnF) system, which applies trigger conditions, fast reconstruction algorithms, and a sequence of analysis-specific filters. For the purposes of this work, the relevant filters are the Muon Filter, the OnlineL2 Filter, and the GFU Filter, applied in this order. The processing is highly parallelized to minimize latency, which averages 29 s between trigger and sending [46]. Once filtered, selected events are transmitted via satellite to the northern hemisphere, where final processing is performed and alerts are issued if specific criteria are met.

3.1.1 Processing in the Southern Hemisphere

The following describes the data processing performed on-site at the South Pole, listed in chronological order. The main goal of this system is to drastically reduce the data volume to be transmitted through the satellite link, which has extremely limited bandwidth. Data connectivity is provided around the clock via the Iridium satellite network [69], offering approximately

2.4 kbit/s per modem. Even with multiple modems in parallel, the effective throughput remains orders of magnitude below standard Internet speeds. For comparison, streaming a low-quality video requires around 500 kbit/s, and sending a simple SMS message (140 bytes) would take a fraction of a second—yet large scientific datasets would take hours or even days to transmit. As a result, on-site data reduction and filtering are essential.

The full processing chain at the South Pole can be organized into a sequence of increasingly complex steps, beginning with low-level noise suppression and hit selection at the detector module level, and culminating in high-level reconstructions and event classification. The steps described below are presented in chronological order and grouped conceptually into three tiers:

- **Trigger and DAQ**: initial stages that detect coincident hits and group them into raw events.
- **Cleaning and Reconstruction**: intermediate steps that isolate individual physics events and estimate basic properties such as direction and topology.
- **Filtering and Classification**: advanced selection steps that apply quality cuts, compute event features, and classify candidates using machine learning.

Each step in this pipeline is critical to reducing the data volume and refining the selection of astrophysical neutrino candidates prior to transmission. The full procedure is detailed below and summarized in Figure 3.1.

Strict Multiplicity Trigger at the DOM level. This is the first filtering stage after PMT signal detection and occurs directly within each DOM. The DOM verifies the presence of a Hard Local Coincidence (HLC), meaning that neighboring DOMs (or next-to-nearest neighbors) also recorded a signal within 1 microsecond. If this condition is satisfied, the full waveform is sent to the IceCube lab's trigger software; otherwise, only the signal amplitude is transmitted [37].

Simple Multiplicity Trigger (SMT-8). This step checks whether at least 8 HLCs occur within a sliding 5-microsecond window. If this condition is fulfilled, a dynamic data acquisition window is opened, which closes once no more HLCs are recorded in the subsequent 5-microsecond intervals. All hits recorded during this acquisition window are grouped into a DAQ event, stored to disk, and passed to the PnF system. At this stage, the system records about 2.7 kHz of events, corresponding to approximately 1 TB of data per day. These unfiltered data are stored locally and transferred to the northern hemisphere once per year [37].

Event Splitting. A single DAQ event may contain multiple overlapping physical interactions. Therefore, it must be subdivided into individual physics events. This is achieved by identifying time intervals within the DAQ event where the SMT-8 condition is satisfied, expanded by 4 microseconds before and 6 microseconds after the core window. These intervals are interpreted as potential physics events. However, if two events truly occur simultaneously, this step will not separate them; more advanced algorithms are required in such cases [37].

Pulse Cleaning. Each physics event contains many recorded pulses. This stage applies a cleaning algorithm to discard noise and isolate meaningful signals. First, only HLC pulses are retained. Then, any other pulses near the HLCs are kept if they are closer than 150 meters and recorded within 1 microsecond. These are the Single Local Coincidences (SLC). This procedure is repeated three times. Finally, the 6-microsecond time window with the highest number of pulses is selected. The goal is to extract a coherent cluster of pulses that are both temporally and spatially consistent with a true physical event [37].

Muon Filter. After pulse cleaning, events are subjected to various filters depending on the particle type. In this work, the focus is on the Muon Filter, which aims to identify track-like muon events. This step reduces the event rate from 2.7 kHz to about 40 Hz using a combination of the LineFit method and SPE reconstructions (see Section 2.3.1).

For up-going[1] events (originating from the northern sky), the likelihood value is used as a quality metric to reject shower-like backgrounds. In contrast, for down-going events (from above the detector), the background of atmospheric muons is much more intense. In these cases, the selection is based on the total recorded charge ($\mathcal{Q}_{\text{tot}}$) as a function of the event's declination. The resulting data rate after this filtering step is low enough to allow daily transmission to the northern hemisphere [46].

OnlineL2 Filter. The OnlineL2 filter is the next stage in the event selection chain following the muon filter. The filtering begins with an improved directional reconstruction. While the SPE fit used in the previous stage performs well for bright through-going tracks, it struggles with more challenging topologies, such as corner-clipping muons or tracks aligned with detector symmetry axes. To mitigate this, the reconstruction is repeated with two additional seed directions rotated by 120° with respect to the initial track hypothesis.

[1] Zenith angle $\theta \gtrsim 82°$

Subsequently, the MPE fit (see Section 2.3.1) is performed using the direction obtained from the SPE step. Based on the fit quality and the integrated charge, additional cuts are applied, which reduce the event rate from 40 Hz to approximately 6 Hz [70], while retaining >99% of the up-going and >80% of the down-going signal for an E^{-2} spectrum [46].

At this reduced rate, more computationally expensive reconstructions become feasible. The SplineMPE algorithm is used at this stage to improve directional accuracy [45]. It works on a cleaned subset of pulses and accelerates convergence of the likelihood minimization. Using this refined track, the event energy is estimated with both the MuEX and TruncatedEnergy methods described in Section 2.3.3.

The events passing all these criteria are then passed to the final selection stage: the GFU filter.

GFU Filter. The final step in the event selection pipeline is the GFU filter, which aims to reject the remaining atmospheric-muon background and retain a high-purity sample of well-reconstructed neutrino candidates. At this stage the event rate is low enough to run additional classification logic.

To further refine the selection, a multivariate classifier based on Boosted Decision Trees (BDTs) is used. The classifier is trained using the observables reconstructed in the previous steps, including direction, energy, fit quality, and the spatial distribution of pulses. These variables help distinguish well-reconstructed signal-like events from background, particularly misreconstructed atmospheric muons.

Because the dominant background varies between the hemispheres—being primarily mis reconstructed and low quality in the north and down-going[2] higher rate (and better quality) in the south—two independent classifiers are trained, each using a dedicated set of input variables optimized for its respective sky region.

The training is performed offline using the `scikit-learn` library [71], while the operational classification at the South Pole is handled by the Python library `pyBDT` IceCube's custom implementation, that has been validated in previous analyses and is suitable for deployment under the detector's constraints [72].

Only events passing this final filter are considered alert candidates. These events exhibit high reconstruction quality and are accompanied by an estimate of their angular uncertainty. Full details on the classifier design, training, and input variables can be found in [46].

[2] Zenith angle $\theta \lesssim 82°$

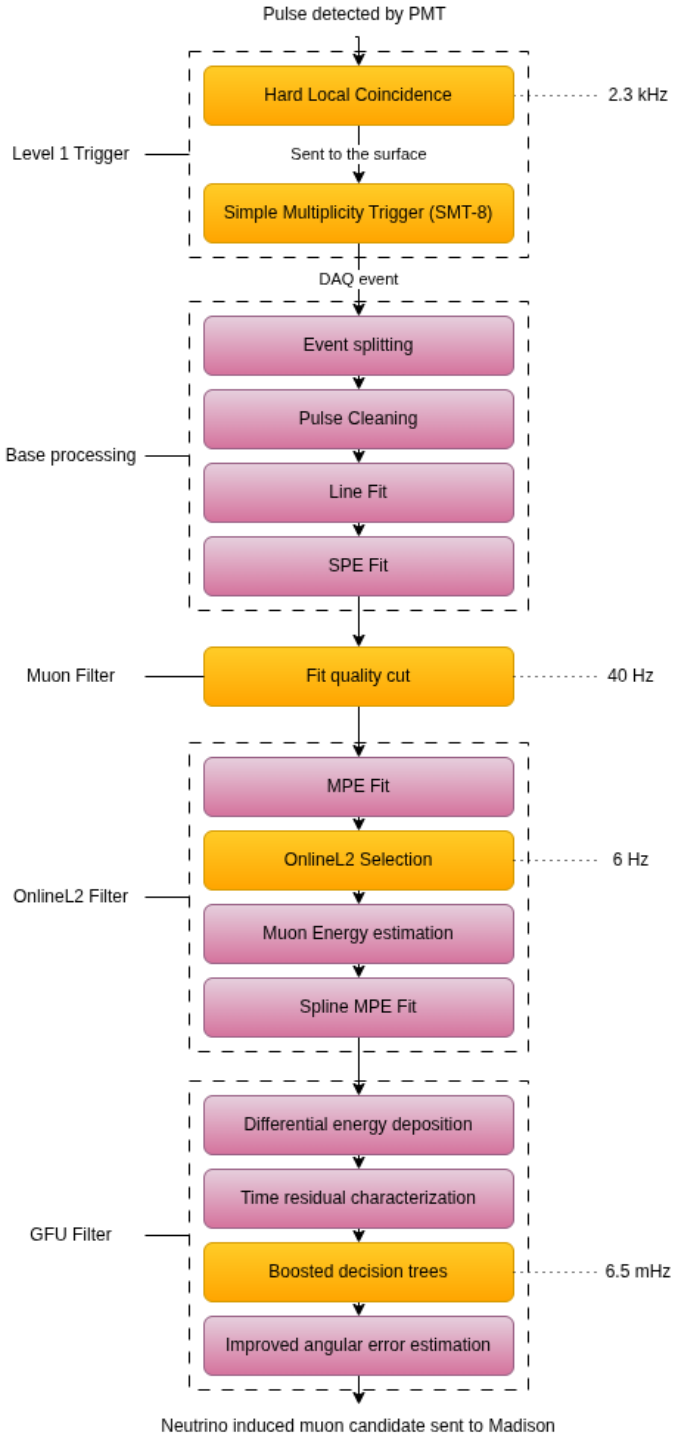


Figure 3.1: Reconstruction and data reduction pipeline currently running at the South Pole. Orange boxes correspond to data cuts, while blue ones correspond to reconstruction steps. Adapted and expanded from [46].

3.1.2 Data transmission

Once an event passes the full selection chain, including the GFU filter, a compact summary is transmitted from the South Pole to the Northern Hemisphere for further processing and dissemination. This transmission relies on the Iridium satellite network, a low-bandwidth but reliable system well suited to the geographical placement [69].

To ensure efficient use of this limited bandwidth, the transmitted summary includes only essential metadata such as the run and event numbers, trigger time, reconstructed direction, angular uncertainty estimate, and estimated muon energy. This information is serialized in JavaScript Object Notation (JSON) [73], chosen for its simplicity and wide compatibility. The typical JSON payload size is about 1.3 kilobytes. In contrast, the full waveform data for the same event—which are transferred separately once per day—range from 150 to 250 kB [46]. This significant reduction in size is necessary to accommodate Iridium's limited bandwidth of approximately 2.4 kilobits per second per modem.

An overview of this transmission process is shown in Figure 3.2. After selection, the JSON payload is transmitted via Iridium to the computing center in Madison, Wisconsin, where it is ingested into a dedicated database. From there, downstream systems access the data through a lightweight messaging infrastructure based on the `ZeroMQ` [74] protocol. This architecture enables multiple analysis systems to subscribe to specific event streams in real time.

Coordinating this entire dataflow is the software framework known as `I3Live` [37]. It plays a central role in managing the lifecycle of alerts, monitoring system health, and ensuring structured and timely communication between the South Pole and the Northern Hemisphere.

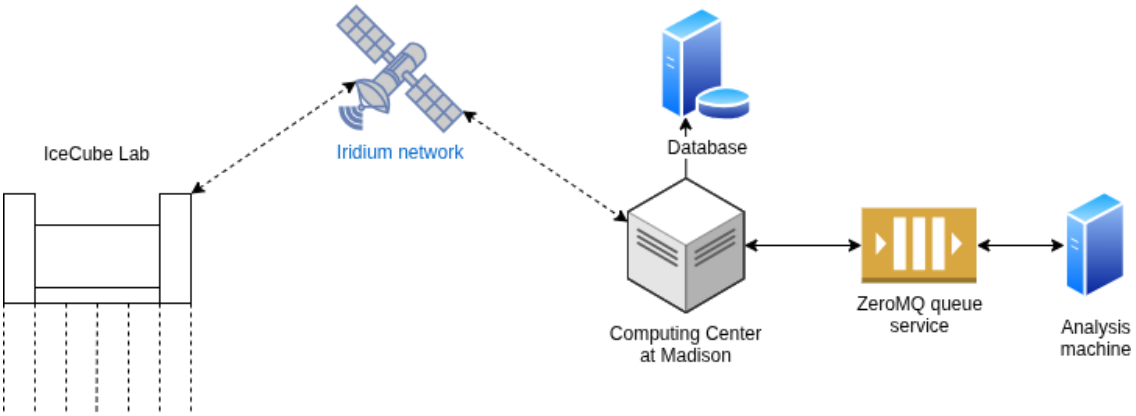


Figure 3.2: Schematic overview of the IceCube data transmission architecture. After event selection, a JSON-formatted summary is transmitted via the Iridium satellite system to the computing center in Madison. From there, the event is stored and made available to analysis clients via a `ZeroMQ` queue.

In terms of performance, the system achieves a median latency of approximately 29 seconds for receiving JSON summaries, while full waveform data arrive with a median latency of 39 seconds [46]. Such low latencies are essential, particularly during periods of high trigger rates. To prevent queue backlogs and maintain throughput, the median per-event processing time at the South Pole remains under 10 seconds.

Finally, beyond event summaries, the system also transmits real-time monitoring data, including detector status, trigger rates, and system diagnostics. These additional data streams are crucial for supporting operational continuity and enabling automated quality control mechanisms.

3.1.3 Processing in the Northern Hemisphere

Once the data from neutrino-induced muons have been transmitted using the infrastructure described above, the next stage of the alert pipeline takes place in the North. The analysis server (illustrated in Figure 3.2) runs a process supervisor called `supervisord` [75], which manages multiple clients responsible for executing various analyses in parallel. These clients are part of IceCube's internal software stack, among which the most relevant for this work is the GFU Client.

The GFU Client runs the `realtime` software suite, which handles real-time time clustering analysis and alert generation. Its primary role is to identify potential astrophysical sources based on the incoming events. It operates in two modes of particular interest for this thesis: the first is the monitoring of known gamma-ray sources, which has been modified in the context of this work and will be explained in detail in the following sections; the second is the unbiased or all-sky scanning mode, which searches for significant temporal clusters of events across the entire sky without prior assumptions about source locations.

Both modes are designed to generate cluster alerts, meaning they rely on the detection of statistically significant time coincidences among multiple events. When the significance exceeds a predefined threshold, an alert is issued through various dissemination channels, marking the final stage of the realtime processing pipeline.

3.2 Time-Variable Point Source Analysis

The main objective of the real-time software in its GFU *source monitoring mode*, is to identify clusters of events that may be associated with an astrophysical point source. In other words, it

aims to detect space-time accumulations of muons induced by neutrinos that deviate significantly from the expected atmospheric background.

3.2.1 Time-integrated Analysis

As a first step, in the case of a time-independent analysis, the probability density function (PDF) associated with each event can be factorized into two components: a spatial term and an energy term. These are defined under the signal hypothesis ($\mathcal{S}_i$) and the background hypothesis ($\mathcal{B}_i$) as:

$$\mathcal{S}_i = P^S_{\text{spatial}}(\vec{x}_i \mid \vec{x}_s, \sigma_i) \cdot P^S_{\text{energy}}(E_i \mid \vec{x}_i, \gamma), \tag{3.1}$$

$$\mathcal{B}_i = P^B_{\text{spatial}}(\vec{x}_i) \cdot P^B_{\text{energy}}(E_i \mid \vec{x}_i). \tag{3.2}$$

Here, $\vec{x}_s$ is the hypothetical location of the source in the sky, and $\vec{x}_i$, σ_i, and E_i are the reconstructed direction, angular uncertainty, and energy proxy of the event, respectively. The parameter γ denotes the spectral index of the astrophysical neutrino flux, typically expected to be around 2 based on Fermi acceleration models [22]. The separation between spatial and energy components is justified by the high angular and energy resolution of the detector, which allows both dimensions to be modeled independently [76, 46].

The function P^S_{spatial} models the probability of an event originating from an astrophysical source located at $\vec{x}_s$. It is represented by a circular two-dimensional Gaussian; the Raleigh distribution, and is given by:

$$P^S_{\text{spatial}}(\vec{x}_i \mid \vec{x}_s, \sigma_i) = \frac{1}{2\pi\sigma_i^2} \exp\left(-\frac{\Delta\psi(\vec{x}_i, \vec{x}_s)^2}{2\sigma_i^2}\right), \tag{3.3}$$

where $\Delta\psi$ is the angular distance between the reconstructed direction and the source position. This expression accounts for the angular uncertainty of each event and evaluates how compatible its direction is with that of a point source.

In contrast, the background spatial PDF $P^B_{\text{spatial}}(\vec{x}_i)$ describes the probability of an event originating from the atmospheric background. This distribution depends mainly on the zenith angle: up-going events are less likely to be atmospheric muons due to the Earth acting as a filter. A mild azimuthal dependence is also included, arising from the detector's hexagonal geometry, although this effect is smoothed over time by Earth's rotation. The background PDF is constructed from archival data under the assumption that the vast majority of detected events are

of atmospheric origin.

Regarding the energy component, the signal energy PDF P^S_{energy} is derived from simulations of astrophysical neutrinos following a power-law spectrum of the form $E^{-\gamma}$, where γ is a free parameter of the fit. In practice, a base sample is generated with a relatively flat spectrum (e.g., E^{-1}), and statistical reweighting is applied to produce distributions for different spectral indices (typically between 1 and 4). This approach offers flexibility in modeling a wide range of possible signal scenarios.

The background energy PDF P^B_{energy}, on the other hand, is constructed from real archival data, under the assumption that nearly all past events are of atmospheric origin. This approach avoids relying on simulations, whose atmospheric muon rates and distribution shapes are known to diverge from actual detector data due to incomplete modeling. Instead, using experimental data captures the real detector response, even if it includes a small contribution from astrophysical neutrinos. To generate this distribution, the events are randomly scrambled in time, under the assumption that the background is approximately time-independent. This temporal scrambling preserves the detector's spatial and energy response while removing any genuine time correlations, resulting in an empirical distribution of the energy proxy as a function of the reconstructed direction. This background PDF is then used to estimate the likelihood that a given event is compatible with the background.

Thanks to the marked difference in energy spectra—a hard power-law with $\gamma \sim 2$ for the signal versus a much softer $\gamma \sim 3.7$ for the atmospheric background—this component provides strong additional discrimination power, particularly in the high-energy regime where astrophysical events are more likely to dominate.

Both the signal and background distributions are combined into the likelihood function $\mathcal{L}$, defined as the product of the probabilities of observing each event under a composite hypothesis [77]:

$$\mathcal{L}(n_s, \gamma) = \prod_{i=1}^{N} \left(\frac{n_s}{N} \cdot \mathcal{S}(\vec{\theta}_i \mid \gamma) + \left(1 - \frac{n_s}{N}\right) \cdot \mathcal{B}(\vec{\theta}_i) \right), \tag{3.4}$$

where $\vec{\theta}_i = (\vec{x}_i, E_i)$ are the direction and energy parameters of event i, N is the total number of events in the sample, and n_s and γ are free parameters representing, respectively, the number of signal events and the spectral index of the neutrino flux.

These parameters are determined by maximizing the likelihood function under physical constraints: n_s must lie between 0 and N, and γ is typically restricted to the interval $[-4, -1]$. Max-

imizing $\mathcal{L}$ yields the best-fit values of the parameters, denoted by $\hat{n}_s$ and $\hat{\gamma}$. These values are then used to define the test statistic Λ, based on the likelihood ratio between the alternative hypothesis and the null hypothesis ($n_s = 0$):

$$\Lambda = 2 \log \frac{\mathcal{L}(\hat{n}_s, \hat{\gamma})}{\mathcal{L}(n_s = 0)}. \tag{3.5}$$

By directly maximizing this expression, the test statistic can be rewritten as:

$$\Lambda = 2 \sum_{i=1}^{N} \log \left[1 + \frac{n_s}{N} (w_i - 1) \right], \tag{3.6}$$

where $w_i = \mathcal{S}_i / \mathcal{B}_i$ is the event weight, i.e., the ratio between the signal and background probability densities for event i. Intuitively, high values of w_i indicate that the event is more compatible with the signal hypothesis than with the background.

The value of Λ obtained from real data is compared to its distribution under the null hypothesis to assess the significance of any observed excess, allowing us to determine whether there is statistical evidence for an astrophysical source [46].

3.2.2 Time Clustering

Once the time-integrated likelihood has been defined, the analysis can be extended to include temporal information, aiming to identify clusters of neutrino events that are spatially and temporally correlated. This strategy is particularly relevant given that many candidate astrophysical neutrino sources are expected to be transient in nature. According to current models, these sources undergo periods of enhanced activity or flares, during which they emit bursts of neutrinos on the scale of days.

The goal of the time clustering algorithm is to identify the time window that maximizes the test statistic Λ, therefore increasing the statistical significance of a potential signal. To this end, a general time likelihood model is adopted [46]: a box-shaped time profile defined by a start time t_i and an end time t_k, within which the point-source likelihood analysis is applied.

The procedure is illustrated in Figure 3.3. Starting from a *triggering event* (the most recent one, at time t_k), all previous events with $t_i < t_k$ that satisfy certain conditions are considered. For each time window $[t_i, t_k]$, the same likelihood analysis described in the previous section is performed, restricted to the events within that window.

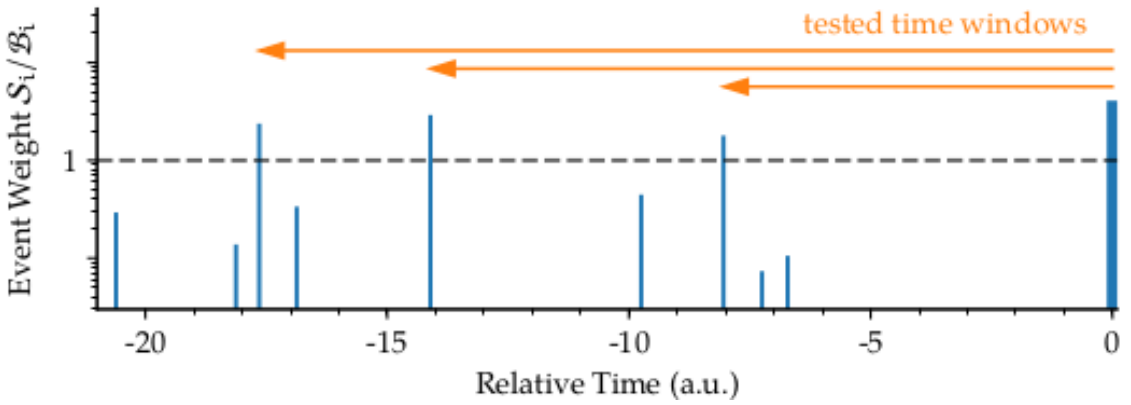


Figure 3.3: Illustration of the time clustering algorithm. Vertical bars represent the event weights S_i/B_i as a function of time in arbitrary units. The rightmost event acts as the trigger for the clustering procedure. Orange arrows denote the time windows considered (extending backwards from the trigger). Only those starting with an event whose weight exceeds a predefined threshold (set to 1) are included in the analysis. Taken from [46].

To reduce the computational cost associated with trying all these tuples of events, two constraints are introduced:

1. A dynamic time window limit T_{max}, defined as the point beyond which the sensitivity becomes comparable to that of the time-integrated analysis.
2. A minimum event weight threshold $w_i > 1$, since events with lower weights can only decrease the value of the test statistic.

Additionally, to avoid a statistical bias toward short flares (which are more numerous), a time-penalty factor is incorporated into the test statistic, leading to the modified expression:

$$\Lambda = 2 \log \left[\frac{\mathcal{L}(\hat{n}_s, \hat{\gamma})}{\mathcal{L}(n_s = 0)} \cdot \frac{\mathcal{U}(t_i, t_k)}{T_{\text{max}}} \right], \tag{3.7}$$

where $\mathcal{U}(t_i, t_k)$ denotes the detector's uptime during the tested time window $[t_i, t_k]$. The final result of the time clustering algorithm is the maximum value of Λ among all the tested combinations.

3.2.3 Statistical Significance and Sensitivity Concepts

In order to quantify the significance of a result and define detection thresholds, several statistical concepts are used in the analysis of IceCube alerts. These include the *p-value*, *sensitivity*,

and *discovery potential*, which are derived from the test statistic used to compare the null and alternative hypotheses. The following introduces these key quantities, which will be employed in subsequent chapters.

p-value The *p-value* quantifies the probability of obtaining a result equal to or more extreme than the observed test statistic $\hat{\Lambda}$, assuming the null hypothesis (i.e. background-only) is true. It is defined as:

$$p(\hat{\Lambda}) = \int_{\hat{\Lambda}}^{\infty} P^{\mathrm{B}}(\Lambda)\, d\Lambda = \int_{\hat{\Lambda}}^{\infty} \chi^2(\Lambda; n_{\mathrm{DoF}})\, d\Lambda, \tag{3.8}$$

where $P^{\mathrm{B}}(\Lambda)$ is the probability distribution of the test statistic under the background-only hypothesis, and χ^2 is the chi-squared distribution with n_{DoF} degrees of freedom [46].

The *power* of the test, denoted β, represents the probability of correctly rejecting the null hypothesis when the alternative is true.

Sensitivity The *sensitivity* of an analysis is defined as the median upper limit that the analysis would produce in the absence of a true signal. It is expressed as the minimum flux that can be excluded at 90% confidence level, i.e., for a signal that yields:

$$p \geq 50\% \quad \text{and} \quad \beta = 90\%. \tag{3.9}$$

This quantifies the experiment's expected ability to constrain the signal strength under the assumption that no true signal is present.

Discovery Potential The *discovery potential* refers to the signal strength required to achieve a predefined significance level of 5σ (corresponding to $p \approx 2.86 \times 10^{-7}$) with a given power, commonly $\beta = 50\%$. This represents the flux level at which the analysis would have a 50% probability of producing a statistically significant discovery. This quantity is essential for evaluating the discovery capability of a given search configuration, though it can be computationally expensive to estimate [46].

3.3 Gamma-Ray Sources Monitoring

Following the description of the alert generation pipeline and the statistical analysis applied to muon neutrino candidates in the realtime system, this section presents the source moni-

toring mode of operation, which implements the time-dependent analysis introduced in Section 3.2.2.

This mode is based on a predefined list of astrophysical gamma-ray sources, regularly observed by ground-based Cherenkov telescopes. Every time a muon-neutrino candidate is received by the GFU Client—approximately every two minutes via a ZeroMQ message queue—the client evaluates the event against all positions in the source catalog. This is done by computing the event weight w_i. If the condition $w_i > 1$ is met, the time-clustering algorithm is triggered.

The event that triggers the analysis defines the end of the temporal window (the so-called *trigger time*). For each tested window, the likelihood is maximized, and the best test statistic is selected. If this value exceeds a statistical significance threshold of 3σ, an alert is issued and disseminated via several communication channels (described in Section 3.4).

To allow for the detection of long-duration flares, the maximum window size was set to $T_{\text{max}} = 180$ days. This value was motivated by previous studies of gamma-ray activity and does not affect computational performance: the analysis is implemented efficiently in C++, with a Python interface that facilitates integration and control [46].

3.3.1 Pre-Defined Source List

The selection of monitored sources was driven both by astrophysical criteria and by practical considerations related to follow-up capabilities. Specifically, the catalogs include only those sources that are scientifically interesting and visible to the partner observatories, which at the time of writing (2025) are MAGIC [13], VERITAS [15], and H.E.S.S [14]. Each telescope has a tailored list of sources that meet the following conditions:

- they are visible for part of the year from the corresponding site, taking into account the elevation of the source above the horizon,
- they are extragalactic, with redshift $z \leq 1$ (the most distant object observed in VHE gamma rays [78]),
- they exhibit variable behavior (i.e., potential flaring sources),
- their gamma-ray flux measured by Fermi-LAT is compatible with a 5σ discovery potential in the IACTs within two hours of observation (extrapolated above 100 GeV).

As a final remark, a new and updated source list is currently under development as part of ongoing efforts to modernize and broaden the scope of the alert system. This work was pre-

sented at the 40th International Cosmic Ray Conference (ICRC 2025) [79] and represents a complementary step toward a more inclusive and publicly accessible framework for generating neutrino cluster alerts.

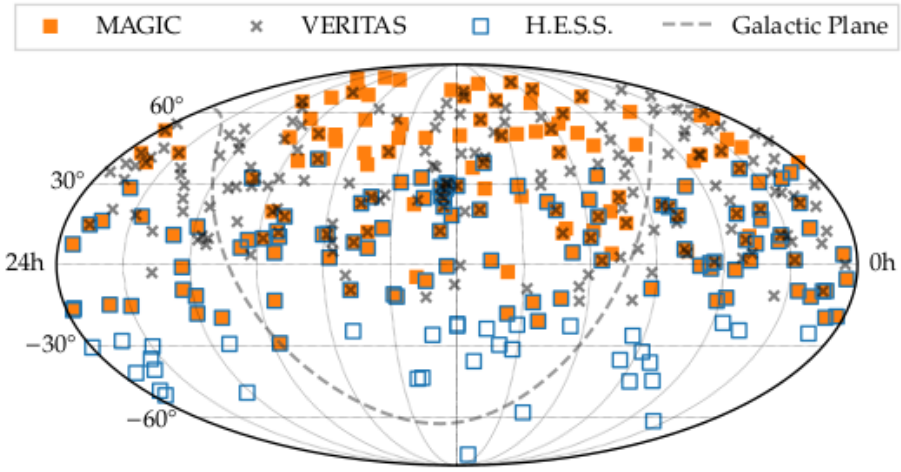


Figure 3.4: Skymap showing the positions of the monitored sources in equatorial coordinates. Different symbols represent sources monitored by MAGIC (orange squares), VERITAS (grey crosses), and H.E.S.S. (blue open squares). The dashed line indicates the Galactic plane. Taken from [46].

A new and updated source list is currently under development as part of ongoing efforts to modernize and broaden the scope of the alert system. This work was presented at the 40th International Cosmic Ray Conference (ICRC 2025) [79] and represents a complementary step toward a more inclusive and publicly accessible cluster alert generation system.

3.3.2 Significance Threshold and P-value Estimation

The current threshold for alert generation is based on the *pre-trial p-value*, which is computed from the test statistic value and the background-only distribution obtained through toy simulations. These simulations are generated by scrambling historical data, allowing for the empirical estimation of the probability to obtain a given test statistic under the null hypothesis.

The pre-trial p-value corresponds to the p-value defined in Section 3.2.3, without accounting for the intrinsic timescale of the event (except for the fixed T_{max}), and is thus unaffected by the system's live-time. Its fast computation makes it ideal as a real-time alert criterion.

In contrast, the *false alarm rate* (FAR) quantifies the probability of observing the same event signature within one year of scrambled background data. It accounts for the number of statisti-

cal trials performed by the system over that time period, and is computed by applying the same analysis to one full year of scrambled events, seeking the best fit for each monitored source.

The current significance threshold of 3σ was defined by studying the expected alert rate: scrambled data was analyzed across various declinations over one year, and the p-value threshold yielding 10 alerts per year for MAGIC and VERITAS, and 6 for H.E.S.S., was selected [46].

It is worth mentioning the existence of two other kinds of alerts that work using the GFU events:

- the **Unbiased** or **All-Sky** alerts, where the GFU client scans over the entire sky, searching for significant temporal clusters of neutrino events. Instead of relying on a predefined source list, the algorithm evaluates every direction in the sky to identify statistically significant excesses above the expected atmospheric background.

- The **Most Energetic Track-Like Event Alerts** are issued for individual neutrino events that exhibit high reconstructed energy and a well-defined track-like topology. These alerts, known as **Gold** and **Bronze**, are distinct from cluster alerts in that they do not rely on temporal coincidence with other events. Although they are based on the same event stream used by the GFU Client, they are processed separately by a dedicated client. Gold alerts correspond to events with a likelihood of 50%, on average, of being of astrophysical origin, while Bronze alerts correspond to events with a 30%. These are currently the only alerts derived from the GFU pipeline that are made publicly available, and they are disseminated via NASA's General Coordinates Network (GCN) [80].

3.4 Alert System

The final component of the system is the generation and dissemination of alerts. As described above, an alert is issued when the best-fit result of the time-clustering analysis exceeds a predefined significance threshold, currently set at 3σ.

The GFU Client is designed to support configurable alert senders or *alerters*, which are modular components responsible for delivering alerts once a detection condition is fulfilled. These senders are attached to different analysis modes, such as the source monitoring or the all-sky search. When a significant event is identified, the senders are triggered sequentially, and the alert is disseminated.

An important characteristic of the system is that once an alert is sent for a given source, subsequent events—regardless of whether they have higher significance—do not generate new alerts during the same flare. This behavior, known as *muting*, prevents repeated alerts from the same region and promotes a form of operational *blindness* intended to limit bias in subsequent offline IceCube analyses of the same source classes and to reduce follow-up fatigue at partner observatories. Detailed follow-up and reprocessing are performed offline during unblinding analyses. This is discussed in detail in Section 5.1.2, since the developed platform removes this behavior from the flow.

Figure 3.5 illustrates the concept: while several trigger events may occur during a flare, only the first one that crosses the threshold initiates the alert. The final alert is associated with the best-fit event and is sent only once per flare.

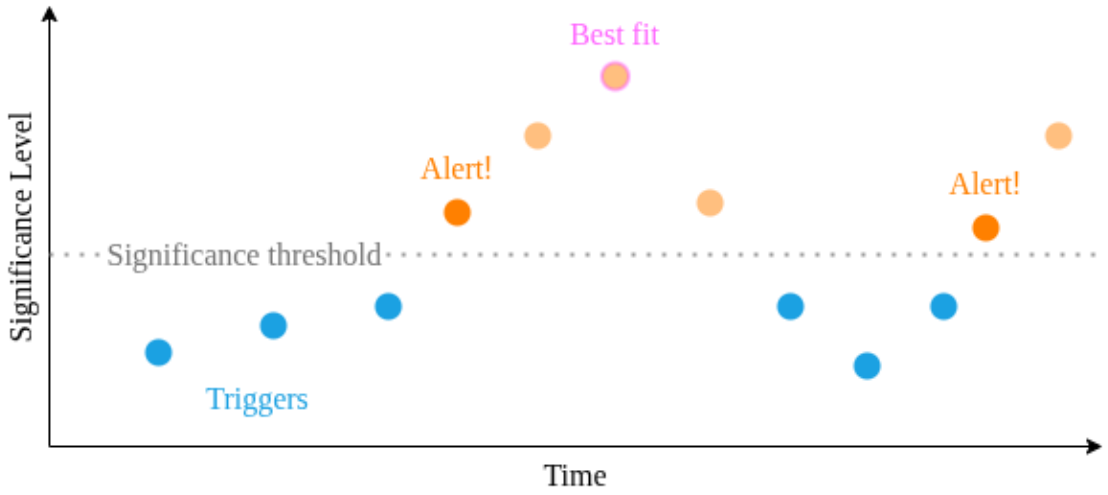


Figure 3.5: Conceptual sketch illustrating the alert generation mechanism. Several time-cluster triggers may occur (blue), but only when the significance exceeds the defined threshold (grey line), an alert is sent (orange). The first such crossing initiates the alert, and subsequent events are muted. The event with the highest significance (pink) is labeled as the best fit, and its identification occurs during offline reprocessing.

Alerts are disseminated through multiple channels. For both GFU source and unbiased all-sky alerts, an email sender component composes a summary message containing key information about the detection: the source name and aliases, right ascension and declination, the time window defined by the first trigger and best-fit events, the flare duration, the pre-trial p-value and the FAR. These emails are sent to a list of partner observatories using IceCube's private mailing infrastructure.

Another channel is Slack, the collaboration's official internal workspace. A dedicated Slack channel receives all alerts in real time from every enabled client. Finally, it is also sent to I3Live through a dedicated alerter.

4

Design and Development of FlareWatch

4.1 Motivation and Design Objectives

A real-time gamma-ray follow-up system is already operational within IceCube for the GFU cluster events. This internal system monitors both all-sky and events and a predefined list of astrophysical sources, issuing alerts when a statistically significant clustering of neutrino events is detected. However, these alerts are currently private and distributed only to a limited group of partner institutions via email, relying on a legacy infrastructure that restricts transparency and broader scientific engagement.

To foster wider participation in the search for astrophysical neutrino sources, IceCube is now moving toward a more open model. The goal is to make these real-time detections accessible to the public, thereby increasing the potential for rapid multi-messenger follow-up and scientific discovery.

This thesis contributes to that transition by developing **FlareWatch**, a new public-facing interactive platform that integrates directly with the GFU alert infrastructure. The name underscores its mission—watching for neutrino flares and clustered transient activity, associated astrophysical sources, in real time. FlareWatch replaces the outdated email-based system with a modern web interface, providing immediate access to alert information, dynamic visualizations, and contextual data analysis of GFU detections as they occur.

4.2 Architecture and Implementation

4.2.1 Underlying Technologies

The interactive platform was implemented as a web application. It was developed using the latest stable release of Python [81], version 3.13, one of the most widely used programming languages in both scientific computing and web development [82, 83]. This choice also ensured full compatibility with the existing IceCube software used for event monitoring and real-time analysis described in Chapter 3.

Python web applications are typically built using frameworks, which are packages that provide essential features and abstractions to ease development. In this case, Flask was chosen as the framework due to its simplicity and general-purpose design [84]. It is a good fit for projects that need to adapt quickly and keep evolving without becoming unorganized or hard to maintain. Compared to other frameworks (like Django or more recent ones like FastAPI), Flask's fast learning curve was a decisive factor.

Flask works using *server-side rendering*, meaning that HTML pages (the content users see on their computers) are generated dynamically on the server and then sent to the client's browser already preprocessed [85]. This approach reduces the computational load on the client side and ensures compatibility with resource-limited devices.

4.2.2 Structure Design

The internal structure of the application follows the *Model–View–Controller* (MVC) design pattern [86, 87], which is widely used in web development to clearly separate features. In this pattern:

- **Model** handles data management, including storage and loading of alert triggers.
- **Controller** manages logic, including filtering, processing, and transformation of data.
- **View** is responsible for preparing and rendering the user-facing content.

This modular separation facilitates code readability, testing, and maintenance. A diagram summarizing this can be seen in Figure 4.1.

The application was structured into sub-modules under a main module named `src`, which implement the layers of the MVC design. Then, there are additional modules that handle errors, constants, and shared utilities.

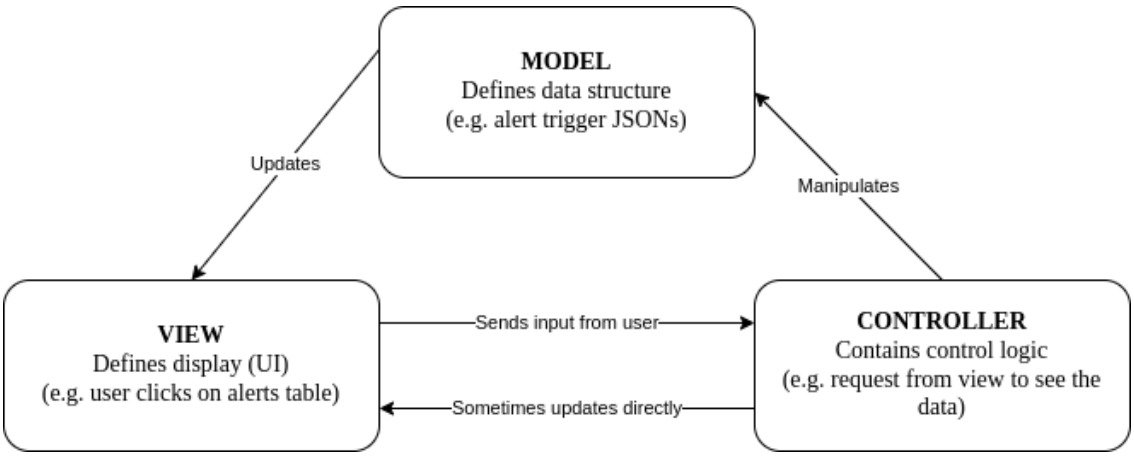


Figure 4.1: Simplified representation of the Model–View–Controller (MVC) architecture used in the application, inspired by [88].

4.2.3 Client Side

The client side, also known as *frontend*, of the application was designed to be simple and efficient. The visual layout is built using HTML5 [89] and CSS3 [90], while interactivity is achieved through JavaScript [91] and the jQuery library [92].

Dynamic data tables and plots are rendered using *AJAX* calls, which are asynchronous requests sent from the frontend (browser) to specific backend *endpoints* (server). These endpoints are like the server's gateways: the frontend sends a request to a specific address (the gateway the server has opened for such task), the backend processes it, and the data is returned in JSON format [73]. This way of handling data between the users and the server follows what are called the *RESTful architectural principles* [93], which help to reduce page load time and improve responsiveness.

A notable example of this flow is in one of the main components of the tool, the active alerts table (described in 4.4.1): the page structure is rendered first, and the table is populated afterwards by a separate request to the server, also known as the *backend*. This decoupling improves performance.

4.2.4 Backend Components

The backend's framework allows to define routes (addresses or URLs the user can visit) as well and HTML rendering via the Jinja2 package [85]. This library allows embedding logic (e.g., loops, conditionals) into templates, making it suitable for dynamic and real-time visualizations.

Data processing relies on scientific computing libraries such as NumPy [94] and SciPy [95], which provide efficient tools for array manipulation, numerical operations, and basic statistics. These are particularly useful when handling the large volumes of archival data in the JSON database (see Section 4.2.5).

4.2.5 Data Management and In-Memory Caching

To keep the architecture as simple as possible, and with the idea of future easy maintenance, the analysis triggers are stored in a single JSON file, organized by a list of potential counterpart astrophysical sources. This file is loaded into the server's random-access memory (RAM) when the application starts and is periodically monitored for changes. If a modification is detected, through the last modification timestamp of the file, the data is automatically reloaded. A sketch of the data flow ins shown in Figure 4.2.

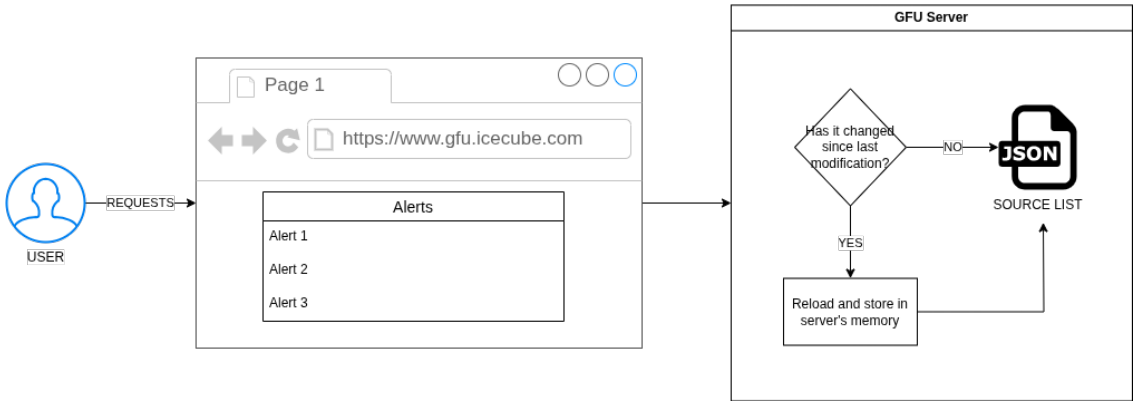


Figure 4.2: Flow of the data caching system: every time a user requests data, either from the events or the alerts tables, the system checks whether the file has been modified since the last request. In case of modification, it reloads the full file into the server's memory.

This approach enables fast data access and avoids the complexity of setting up a full database infrastructure. Moreover, it was tested that this solution would work up to an order of magnitude higher in volumetry of data based on the ingest of the past decade.

In addition to the analysis triggers, the application also loads per-year event files in Numpy format, which contain the full set of neutrino events processed offline. These files are essential for computing the event weights used in the statistical analysis of each flaring episode (see Chapter 5). They are managed using the same in-memory caching system, and follow an identical strategy of load-on-initialization and periodic revalidation.

4.2.6 Deployment and Portability

The application is deployed using a web server called Gunicorn [96]. The web server is in charge of orchestrating the different requests coming from different users, by creating parallel processes, known as *workers*, that handle them independently.

The entire system is packaged into a Docker container [97, 98]. Docker is a software that allows to create isolated environments for applications, with their own independent file system, and makes the reproducibility, deployment and testing across environments significantly easier.

The runtime configuration is managed through environment variables, which can be injected directly into the execution environment or loaded from a specific file format. These are stored parameters in the configuration of the machine where the tool is hosted (either a server, a local computer or the Docker container itself, like in this case). This mechanism permits the separation of sensitive or environment-specific parameters from the application.

Among the configurable parameters are the *secure ingestion token* (see Section 4.3.4), the current application mode ("development", for testing, or "production", for the public version), and various technical-related options. The whole set of available environment variables is shown in the appendix A.3.

This approach ensures that the application can be instantiated in different contexts —with custom behavior—while preserving security [99].

4.2.7 Testing

To ensure the robustness and security of the application during future updates, a dedicated testing module has been developed. This module loads the tool with simulated data and executes two complementary test submodules.

First, a set of unit tests verifies the stability of individual components in isolation, ensuring that their internal logic remains consistent and reliable across changes. Second, an integration test suite validates the overall functionality of the application, including the rendering of pages, interactive tables, and visual plots. These integration tests are designed to cover full user workflows and to confirm the correct interaction between multiple components.

The testing framework is based on `pytest`, a widely used and extensible testing library for Python, well suited for both unit and functional testing [100]. This structure enables fast and automated validation of new features, reduces the risk of bugs, and supports the continuous development of the tool.

4.3 Integration with IceCube's Alert Streams

4.3.1 Receiving Analyses Triggers from IceCube

The key data elements manipulated, analyzed, and displayed by the application are the so-called analyses triggers. These entities represent the step immediately preceding the generation of an official public alert: they have already passed the preliminary filtering and event reconstruction performed at the South Pole (see Section 3.1) and have been transferred to IceCube's northern data center in Madison, Wisconsin (USA). Therefore, they are well-reconstructed muon track events.

Once in Madison, they are processed using IceCube's internal real-time framework, known as the GFU Client, where a variety of analyses are run (Section 3.3. Since these data streams trigger a significance analysis, which may lead to the issuance of a public alert, their output is referred to as an *analysis trigger*.

This tool focuses exclusively on those related to known astrophysical sources, as mentioned previously.

4.3.2 Coupling with the Source Monitoring Infrastructure

The integration between the tool and IceCube's alert generation infrastructure was achieved by extending the Source Monitoring client, located in the realtime software as part of the general GFU client mentioned before (see Section 4.3.1).

A new submodule (named `trigger_loggers`) was introduced in the course of this work, allowing the definition of logging instances responsible for tracking and transmitting alert trigger data. This addition generalizes the behavior of the original source monitoring, enabling the registration of multiple loggers. One of such loggers was implemented to handle the integration with the external web service.

This logger acts as the bridge between the internal alert system and the external application. Upon the generation of a trigger associated with an astrophysical source, the logger sends an `AnalysisTrigger` object to the web application in the form of a JSON payload.

4.3.3 Structure of Transmitted Data

Each trigger, and therefore also alerts, is represented by an `AnalysisTrigger` object, which encapsulates the essential metadata describing a potential astrophysical flare.

Field	Description
`source_name`	Primary name of the potential astrophysical source
`associations`	List of additional source identifiers (e.g., from other catalogs)
`ra_rad`	Right ascension in radians (equatorial coordinates)
`dec_rad`	Declination in radians (equatorial coordinates)
`t_0_mjd_best`	Estimated flare start time
`t_trigger_mjd`	Time of the analysis trigger event
`fit_ns`	Best-fit number of signal-like events
`fit_gamma`	Best-fit spectral index of the neutrino flux
`test_statistic`	Test statistic of the time-clustering analysis
`pre_p`	Pre-trial p-value associated with the analysis trigger

Table 4.1: Fields included in each analysis trigger payload sent to the web application.

This compact data structure is designed to provide all relevant information for the tool to analyze and display. Table 4.1 summarizes the contents of each transmitted object.

This transmission is performed via an HTTP request using the `requests` Python library, a widely adopted tool for server communication [101]. The data is serialized in JSON format for portability and ease of integration with modern web frameworks. This minimal yet informative structure ensures that real-time alerts can be processed efficiently while still providing the necessary context for immediate interpretation and display on the client side.

4.3.4 Receiving Endpoint in the Web Application

On the application side, a dedicated endpoint was created to receive incoming analysis triggers. As depicted in Figure 4.3, the web application is exposed through this to receive them, defining the data ingestion workflow.

To restrict access to this "updating" feature and ensure integrity, the endpoint is protected by an *apikey* or token. Every request must include this key in the HTTP header `x-apikey`. Upon receiving a request, the application validates the token before processing the content. The token is fixed and was generated using Python's `secrets` [102]. This mechanism mitigates unauthorized access and provides a lightweight but effective security layer.

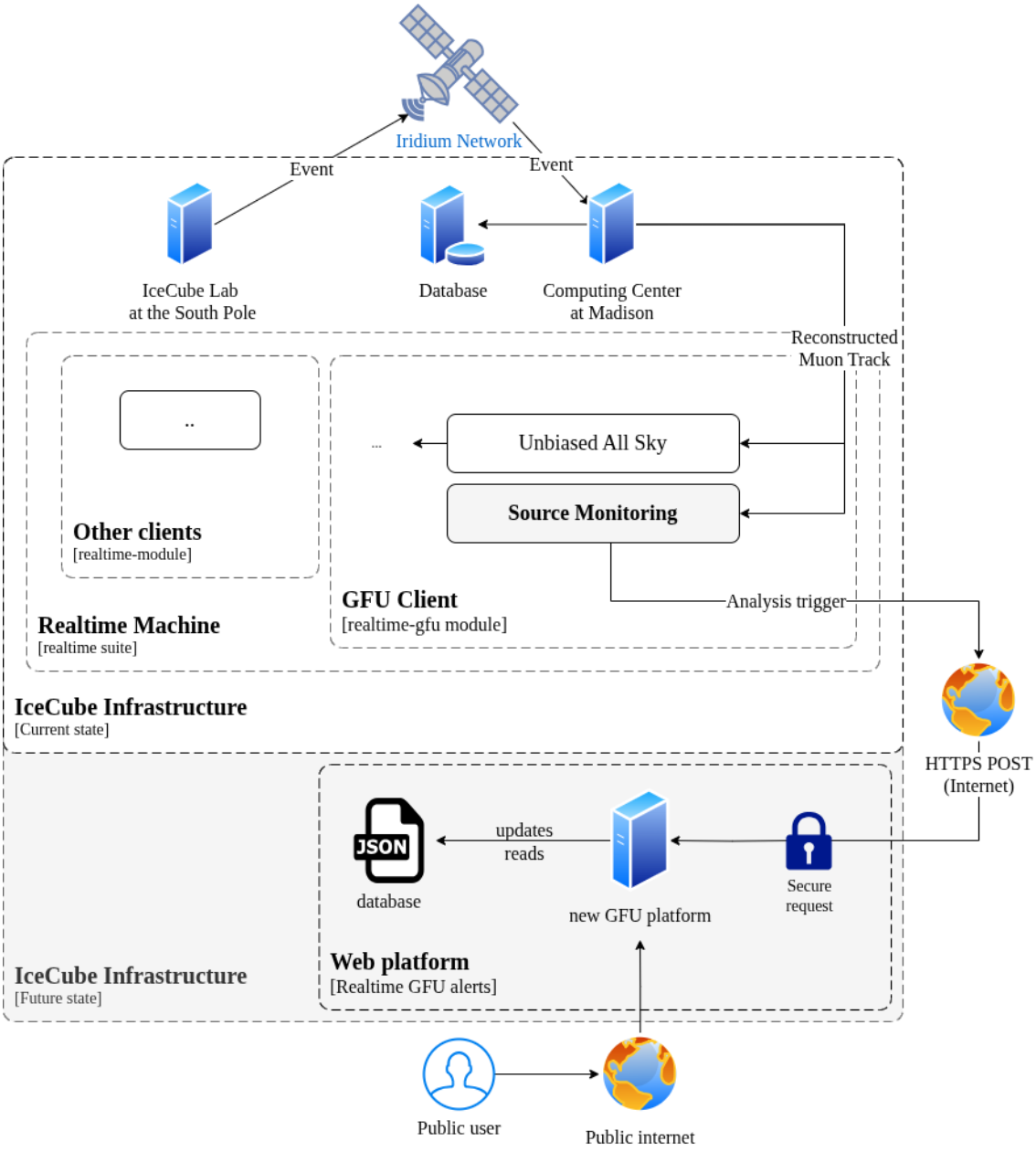


Figure 4.3: Flow of an event travelling from IceCube up to the public facing platform. The event travels through the Iridium network, and arrives to Masison as a reconstructed muon track. Then, this is collected by the GFU client, where the Source Monitoring module logs the trigger (through the `WebsiteSourceTriggerLogger`) in the form of a `SourceMonitor` object through a secure HTTPS POST request to a dedicated API endpoint in the platform's server. Then, after token validation, the latter stores the new trigger into the cached database. Finally, it becomes accesible to the public user. Shaded areas represent modified or contributed assets in this thesis.

This integration strategy also allows authorized clients, such as members of IceCube, to update the web application's trigger dataset in real time, without requiring direct access to the hosting server. This design ensures that, in the event of a failure in the realtime software or a disruption in the communication between systems, researchers retain full control and can manually update the alert information in a straightforward and reliable manner.

The result is a secure, and maintainable interface between the internal IceCube alert infrastructure and the publicly accessible web platform.

4.4 Interactive Visualization Capabilities

The web application offers four main views:

- Active alerts (Figure 4.4)
- Archival alerts (Figure 4.5)
- Support page (Figure 4.6)
- Source detail (Figure 4.7)

A navigation bar at the top allows switching between these functionalities. The Alerts section provides access to both *active* and *archival* alert views, enabling users to explore past detections and monitor ongoing astrophysical activity in real time. The Support section includes a glossary and general documentation covering key concepts, usage instructions, and terminology definitions—essential for ensuring accessibility and interpretability, particularly for users who are not part of the original IceCube collaboration but wish to engage with the data in a meaningful way.

4.4.1 Active and Archival Alerts

The landing page of the application displays the active alerts table, which lists astrophysical sources currently in a *flaring* state—i.e., those with recent triggers exceeding a predefined significance threshold (see Chapter 5).

This table is built dynamically on the client side using the DataTables.js library [103] and is populated via an HTTP request to the platform's server. Each row in the table represents a source and includes:

- Source name

- Right ascension (in degrees)
- Declination (in degrees)
- Duration of the flare (in days)
- Time that triggered the analysis of the event time (UTC)
- Spectral index γ
- Number of signal events n_s
- Pre-trial p-value
- Alert level

Trigger times and flare start times are originally stored in Modified Julian Date (MJD) format and converted to Coordinated Universal Time (UTC). A source is considered active if its most recent cluster exceeds at least 3σ in significance, calculated from the pre-trial p-value (see Sections 3.2.3 and 5.1) and stays in that state as long as there is not another cluster event whose significance is under the base threshold. This also connects with the alert levels (low, medium and high) which are displayed visually using stars (one, two and three), and are defined according to the multithreshold scheme detailed in Section 5.3. A screenshot of the active alerts, as it was seen in , January 27th, 2019, is shown in Figure 4.4.

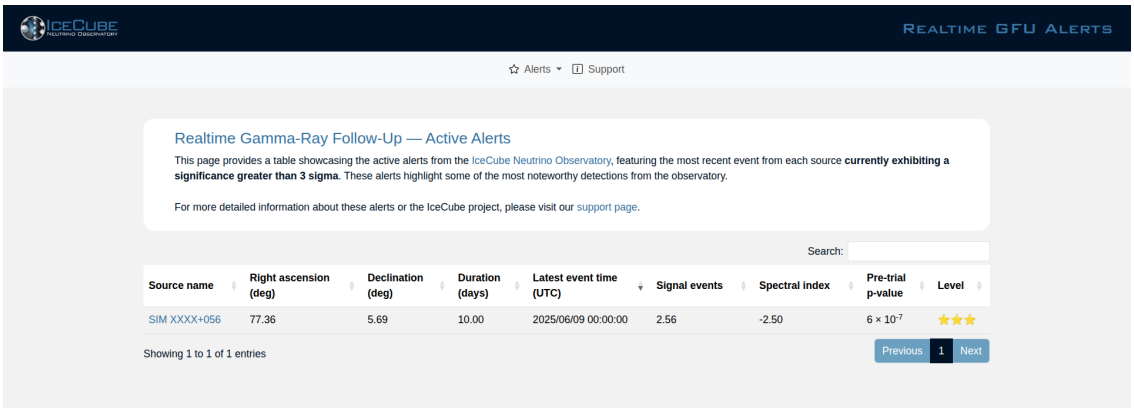


Figure 4.4: Screenshot of the active alerts table in the testing environment, loaded with a simulated source to enforce the active flare shown in the table. This view allows users to see the last alert sent grouped by source.

It is worth mentioning that there is work in progress towards the General Coordinates Network (GCN) [80] alerts implementation and that would imply the representation of the GCN notice address in the same table, which is already prepared to display.

The archival alerts table (Figure 4.5) lists all alerts that have entered a flaring state at any point since 2011. Unlike the active table, which reflects only the current status per source, the archival table includes past alert triggers that crossed the significance threshold. The system identifies and stores these episodes by scanning all alert triggers and selecting those whose significance exceeded the 3σ threshold at the time of their occurrence. The relevant data—source name, coordinates, timestamps, pre-trial p-value, and alert level—is preserved in the JSON file database and returned in the same format as the active alerts. This view helps in long-term monitoring and retrospective analysis of flaring episodes of the past.

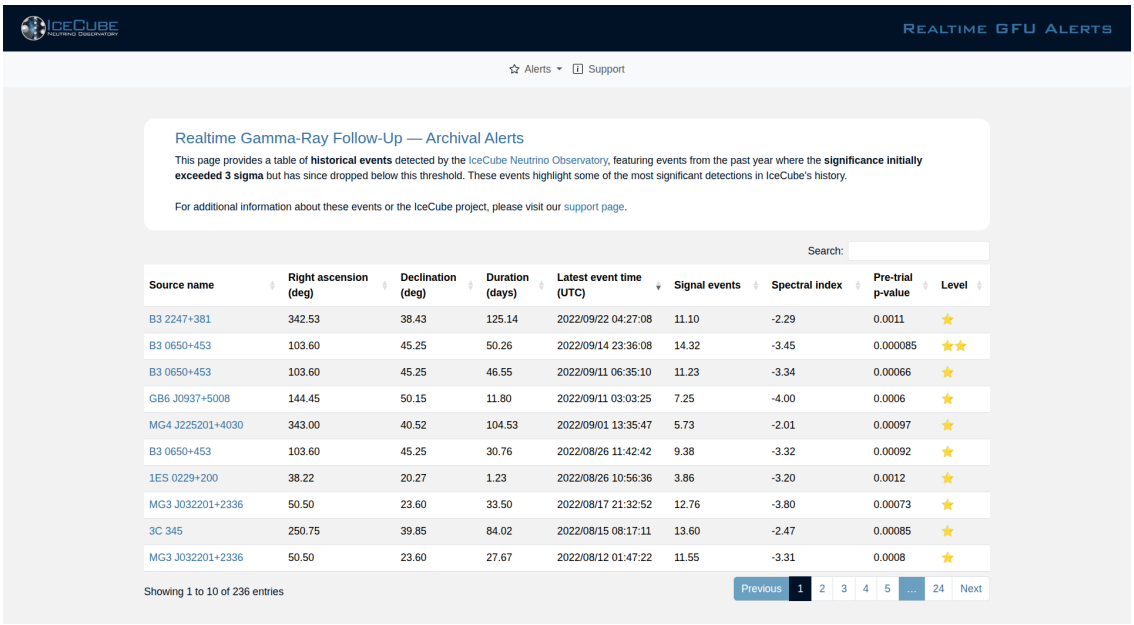


Figure 4.5: Screenshot of the archival alerts table in the testing environment loaded with data from the past decade. This view allows users to scan through all sent alerts and sort by all columns (pre-trial p-value, level, etc) and search by source name.

4.4.2 Source Detail View

Clicking a source name in either the active or archival alerts table opens a dedicated detail page for that source. This view consolidates all available data and analytical visualizations relevant to the astrophysical object.

The layout is structured into four main sections:

- **Left panel:** displays key information such as the source's name, equatorial coordinates (in degrees), and associated identifiers from external catalogs.
- **Right panel:** features an interactive sky map rendered using the Plotly.js library in a Mollweide projection. The map is centered in equatorial coordinates and includes a visual overlay of the galactic plane, computed using standard coordinate transformations [104].
- **Header:** shows the name of the source. It also includes a clickable link, next to it, to the SIMBAD astronomical database [105], generated dynamically using the source name as a search parameter. This functionality is inspired by the browsing behavior provided by the TeVCat catalog [106].
- **Bottom section:** provides an interface with two subtabs for temporal analysis: a significance curve and a flare history table.

The first sub-tab shows the significance curve: a line plot tracking the evolution of the statistical significance of alert triggers over time. Each point represents a trigger and is associated with a computed pre-trial p-value and test statistic. Additional information, such as event weights, could be displayed interactively when hovering or selecting specific points if activated, although it was hidden in the last phase of this work. These visualizations allow users to identify flaring trends or isolate outlier triggers (see Chapter 5).

The second sub-tabb contains the alert history table, a table populated via an endpoint that lists all flaring episodes associated with the source. It includes the timestamp of the alert, the corresponding pre-trial p-value and a symbolic representation of the alert level transition (e.g., from no alert to low, or from medium to high). These transitions are computed by segmenting the timeline into flaring windows and evaluating changes in significance level based on the multithreshold scheme defined in Section 5.3. For each episode, the first trigger that crossed the threshold is used as the representative data point.

Together, these four sections provide a comprehensive and user-friendly interface for monitoring and analyzing the temporal behavior of astrophysical sources in near real time.

4.5 Public and Internal Access Modes

During the development of the web platform, the need arose to maintain two distinct versions of the interface: a public version, openly accessible on the Internet, and a private version, restricted to authorized members of the IceCube collaboration.

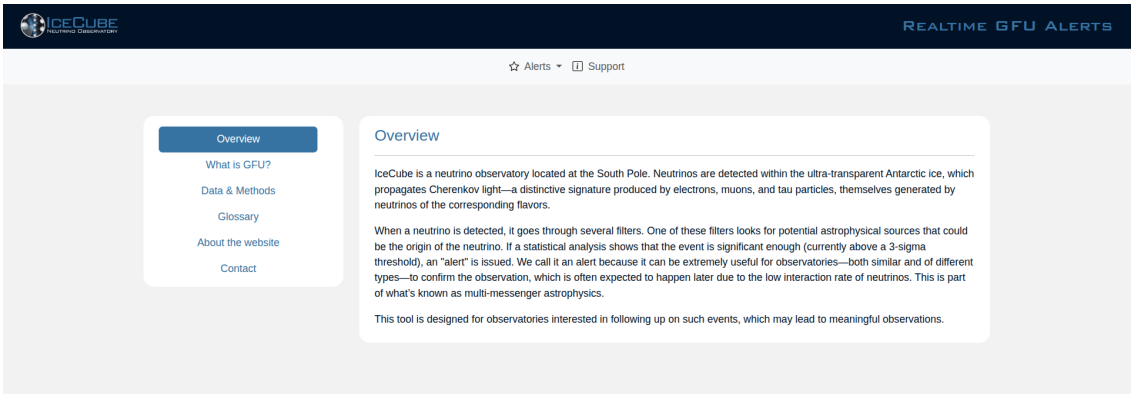


Figure 4.6: Screenshot of the support page, where the public users could check the physical meaning of the displayed quantities, among other information that is still up to discussion within the research group.

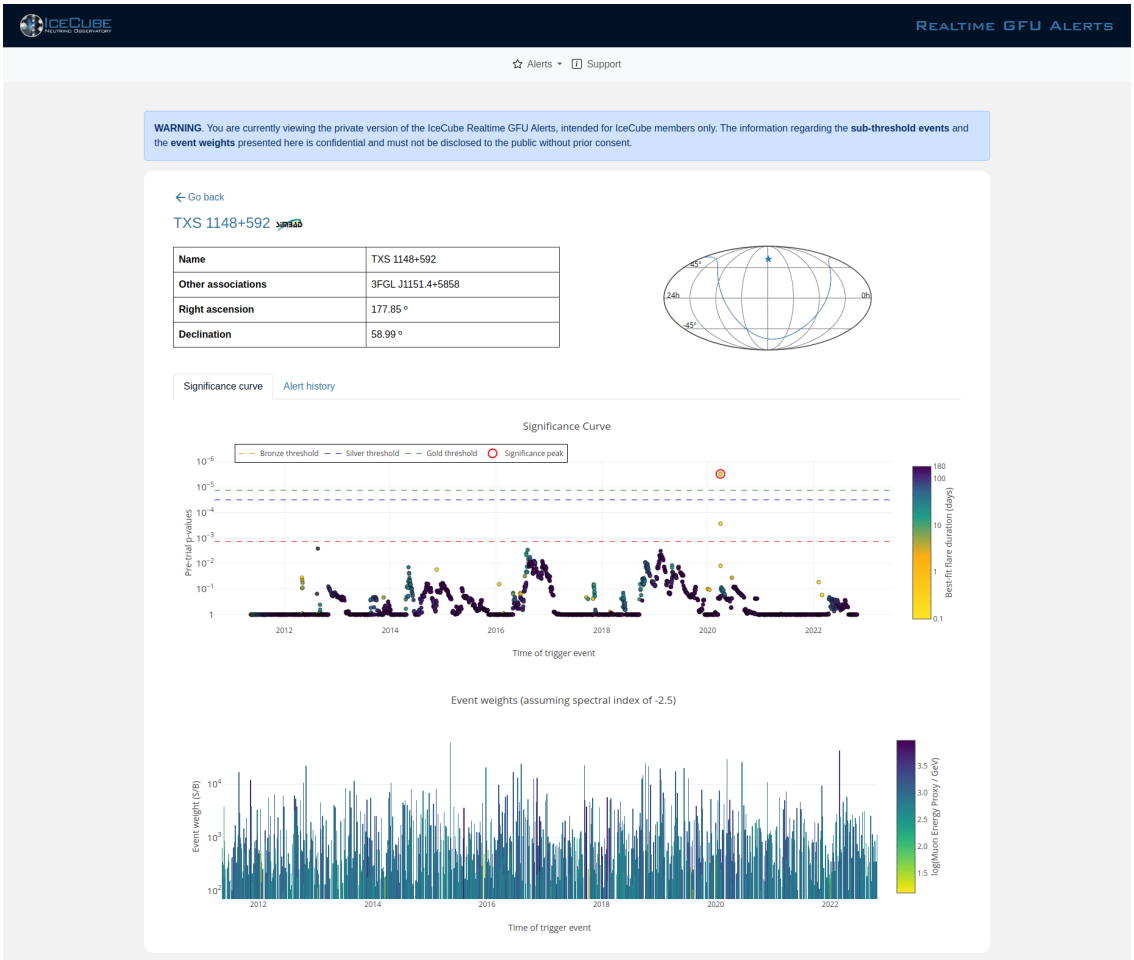


Figure 4.7: Screenshot of the source detail, with the event weights enabled. It also shows the warning message above, which ended up being hidden since both versions, public and private, were combined into one.

This decision was motivated, on the one hand, by the intention to keep the public interface as clean and intuitive as possible, even for non-specialist users unfamiliar with high-energy astrophysics. On the other hand, it responds to scientific privacy considerations, as certain data products may be technically complex or subject to internal collaboration policies.

In the end, in the final phase of this work was decided to turn the private version into the public, unifying both versions and add proper documentation, in turn, for the public. The next subsection summarizes the train of thought followed as well as the features of those two version.

Differences Between Versions

The public-facing version, deliberately hid selected technical details across different views:

- In both the active and archival alerts tables, the columns displaying the best-fit number of signal events (n_s) and the spectral index(γ) were not shown, as these quantities required appropriate context and expertise for interpretation.
- In the source detail view, only alert triggers that exceed the hard threshold of 3σ were visible.
- Sub-threshold triggers were filtered out from the public view to avoid confusion regarding the actual activity state of a source.
- The event weight plot was also disabled in the public version. This plot relies on quantities derived from likelihood models and it was thought not to be informative to general users.

Conversely, the internal or extended version enabled all features and provided full access to sub-threshold triggers and analysis parameters.

The application distinguished between public and private access modes via a single environment variable (see Section 4.2.6) named `FULL_VISIBILITY`, which could be set either through a `.env` file at the root of the project or directly in the system's environment configuration. When this flag was enabled, the application activated the extended internal view.

To ensure awareness, a banner message was displayed at the top of the interface when full visibility mode was active (as seen in Figure 4.7). This explicitly informs users that they were viewing the internal version of the platform.

5

Statistical Processes and Data Visualization

5.1 Significance Curves and Temporal Evolution

The interactive tool developed in this thesis, the IceCube FlareWatch, provides a compact yet powerful visualization of the time evolution of significance levels for potential astrophysical sources monitored by the IceCube GFU realtime pipeline. Rather than showing flux or counts, it focuses on the statistical significance of observed time clusters of neutrino events. The plot, shown in Figure 5.1, is not a light curve in the traditional astronomical sense, but a significance curve: it shows how significant each detected excess is in terms of the null hypothesis of background-only fluctuations. Moreover, adjacent points are correlated since they use almost the same data. It is the real representation of the sketch seen in Figure 3.5 computed dynamically.

Each dot in the plot corresponds to a trigger computed by the realtime clustering algorithm described in Section 3.3. The x-axis shows the trigger time in UTC, while the y-axis displays the corresponding pre-trial p-value in logarithmic scale. Lower p-values, which indicate stronger statistical deviations from background, are plotted higher on the graph. Colored markers encode the best-fit temporal duration of each flare, giving the user insight into the timescale of the excess.

The data shown in this figure are not static but loaded dynamically from the cache stored in the server's memory. This allows the frontend to reflect changes in real time without requiring

manual updates. The interactive environment (built with Plotly.js [107]) enables users to:

- Zoom into specific regions of interest for detailed inspection.
- Pan through the timeline to study long-term behavior.
- Hover over any trigger to reveal its exact UTC time, p-value, and best-fit duration.
- Export the figure in high-quality vector format (SVG) for presentations or publications.

A particularly important feature of this visualization is that it facilitates the temporal tracking of flaring behavior for each source, which the current legacy system does not enable through mailing and requires a manual approach.

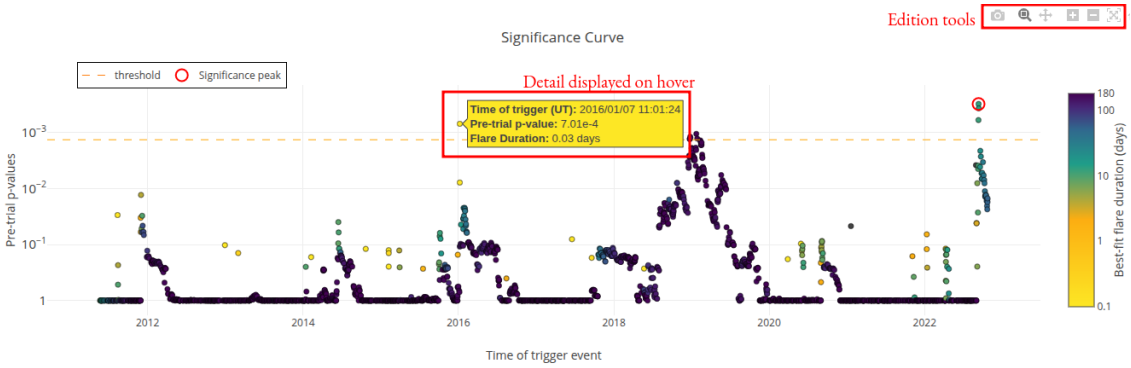


Figure 5.1: Interactive significance curve of a monitored source. The vertical position of each point reflects the logarithmic pre-trial p-value of the trigger, while the color encodes the best-fit flare duration in days. Hovering reveals detailed information about each point. The plot includes tools for zooming, panning, and exporting in vector format, as well as the 3σ significance threshold shown as a dashed line.

5.1.1 Data Handling

The significance curve is constructed dynamically by processing alert data cached on the server. This data, obtained from the realtime pipeline, includes for each trigger its pre-trial p-value, the time of the trigger in UTC, and the best-fit parameters of the flare (start time and duration).

Once loaded, the backend processes this information using vectorized operations through the NumPy package. In particular, the flare duration is computed as the difference between the trigger time and the best-fit start time for each alert. This allows the client side of the tool to encode this duration in the color of each point on the plot, offering a visual cue of the temporal extent of the flare.

A critical part of the visualization involves interpreting the statistical relevance of each trigger. The y-axis of the plot shows the pre-trial p-value of each event, which quantifies how likely it is that a signal of that strength could arise from background fluctuations alone. However, thresholds in statistical analyses are often expressed in terms of σ, or standard deviations of normal distributions, as seen in the case of the significance threshold seen in the tool, in alignment with the threshold of 3σ set in the realtime GFU gamma-ray monitored sources explained in Section 3.3.2.

The conversion from a significance value expressed in standard deviations (σ) to a p-value is based on the cumulative distribution of the standard normal distribution. Specifically, for a one-sided test (as appropriate in our context, where we are only interested in excesses), the corresponding p-value is computed as:

$$p = \frac{1 - \mathrm{erf}\left(\frac{\sigma}{\sqrt{2}}\right)}{2} \tag{5.1}$$

This formula arises from the definition of the error function, which is mathematically related to the cumulative distribution function (CDF) of a Gaussian. The function $\mathrm{erf}(x)$ gives the probability mass within the interval $[-x, x]$, normalized over the total probability. Subtracting it from 1 gives the total probability in the two tails (i.e., values more extreme than $\pm\sigma$). Since we are only interested in one tail (the upper one, corresponding to event excesses), we divide by 2.

For example, a 3σ deviation corresponds to a one-sided p-value of approximately 1.35×10^{-3}, meaning that there is only a 0.13% chance that such a fluctuation would occur under the background-only hypothesis. This transformation allows the backend to consistently convert statistical thresholds expressed in sigmas into p-values that can be directly compared to the pre-trial values returned by the clustering algorithm.

5.1.2 Unmuting and Unblinding

Historically, the IceCube realtime alert system has employed a conservative strategy based on two key mechanisms: *muting* and *blinding*. Under this model, once a statistically significant trigger (i.e., one above a predefined threshold) is detected and an alert is issued, the source enters a *muted* state. During this period, any subsequent triggers from the same source, even if statistically relevant, are temporarily suppressed and not communicated externally. Simultaneously, all sub-threshold triggers (i.e., those that do not reach alert-level significance) remain *blinded*

and are not exposed at all.

This filtering was originally intended reduce redundant notifications and to avoid the unblinding of the full IceCube dataset for monitored sources, as several offline analyses—often targeting the same source catalogs—relied on maintaining data blindness. Premature or uncontrolled visibility of realtime sub-threshold activity could compromise the integrity of these independent studies. Discussions about whether or not to publicly display sub-threshold triggers on the website arose more recently, driven by concerns about their potential misinterpretation.

However, the old muting strategy comes with a significant drawback: it limits temporal resolution and obscures the full dynamical behavior of active sources. Flares are inherently variable in intensity and duration, and suppressing post-alert data may hide the continuation or reactivation of physical processes. Likewise, blinding sub-threshold activity precludes the possibility of recognizing trends or sustained low-level flaring.

In the present work, it has been adopted a new philosophy based on full visibility: the unmuted and unblinded model. All analysis triggers are displayed, regardless of their individual significance or temporal ordering. This allows external users to trace the full temporal evolution of a source's activity, including rebrightenings and prolonged flaring episodes. Figure 5.2 illustrates the contrast between the previous and current models.

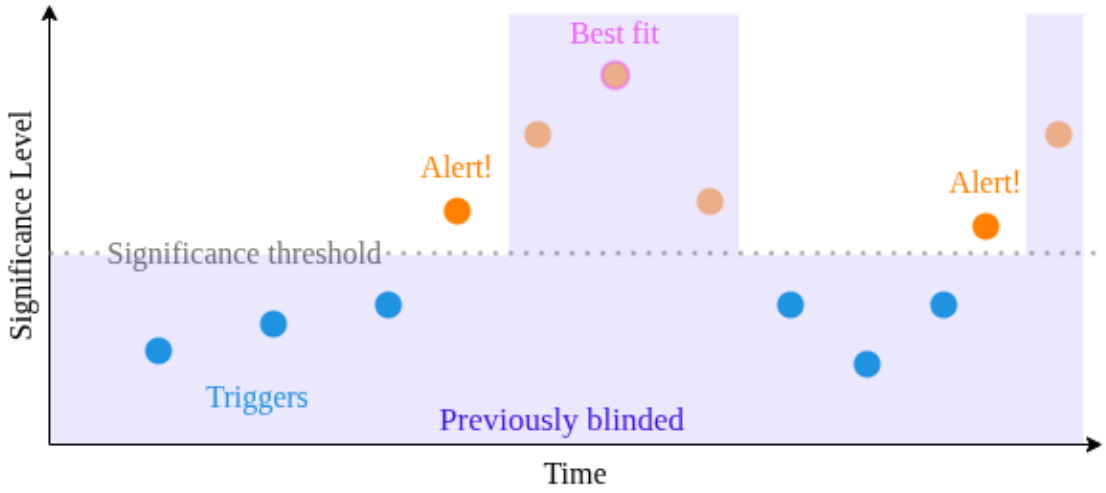


Figure 5.2: Conceptual diagram illustrating the transition from a muted/blinded strategy to an unmuted/unblinded approach. Previously hidden triggers are now visible, providing continuous context around and beyond the alert region.

Importantly, the web tool implements this behavior in a configurable way. The visibility of

post-alert or sub-threshold triggers can be dynamically toggled through a setting in the system configuration. This ensures long-term maintainability and flexibility, as IceCube's alert policies continue to evolve.

This paradigm shift aligns with the broader movement towards open data practices in multimessenger astrophysics. By offering external observatories and researchers access to the full history of realtime significance triggers, more follow-up efforts are promoted.

Nevertheless, the transition to unblinding may raise legitimate concerns regarding the potential misuse of the data and it is still prone to change after the tool is completely reviewed.

5.2 Signal-to-Background (S/B) Weighting

In addition to the significance curve, the web interface initially included an advanced visualization layer designed to expose the internal workings of the clustering algorithm described in Section 3.2.2. This optional panel aimed to display the signal-to-background weights assigned to each event, making explicit the contribution of each data point to the total significance of a given trigger.

The concept is illustrated in Figure 3.3, which reproduces the clustering setup. Events with a weight $w_i = S_i/B_i > 1$ are potential contributors to a flare and are taken as potential start times of possible time windows. These slide over the sequence of event times to find the optimal clustering region that maximizes the likelihood.

The interface implementation allowed users to click on any trigger in the significance curve and dynamically load the corresponding event-weight plot below it. This plot included a highlighted region marking the best-fit flare window, visually connecting the trigger to the underlying distribution of contributing events. It was originally designed as an expert-level diagnostic for internal IceCube use, and fully developed in what was referred to as the *private* version of the tool. This behavior is illustrated in Figure 5.3, where the user selects a specific trigger and the platform highlighting the best-fit flare window in the time series.

Technically, implementing this feature required a significant effort, as the event weights are not included in the standard JSON trigger files. These weights depend on low-level quantities produced during the muon track reconstruction, such as the event direction, angular uncertainty, and energy proxy—none of which are exported by default in the realtime alert output.

To enable their visualization, a parallel caching system was introduced. Alongside the usual JSON files storing trigger metadata, the backend can also maintain a second cache of binary files in NumPy format. These should contain all the necessary per-event quantities and should

be in sync with the alert stream. This mirrors the structure of the primary cache but focuses on the raw inputs used by the clustering algorithm.

Whenever a user requests to inspect the event weights associated with a particular trigger, the backend loads the corresponding event sample from the cache and processes it through a dedicated likelihood module. This module was developed to reproduce the same logic as the clustering code, assigning a weight to each event based on its angular proximity to the source and its reconstructed energy.

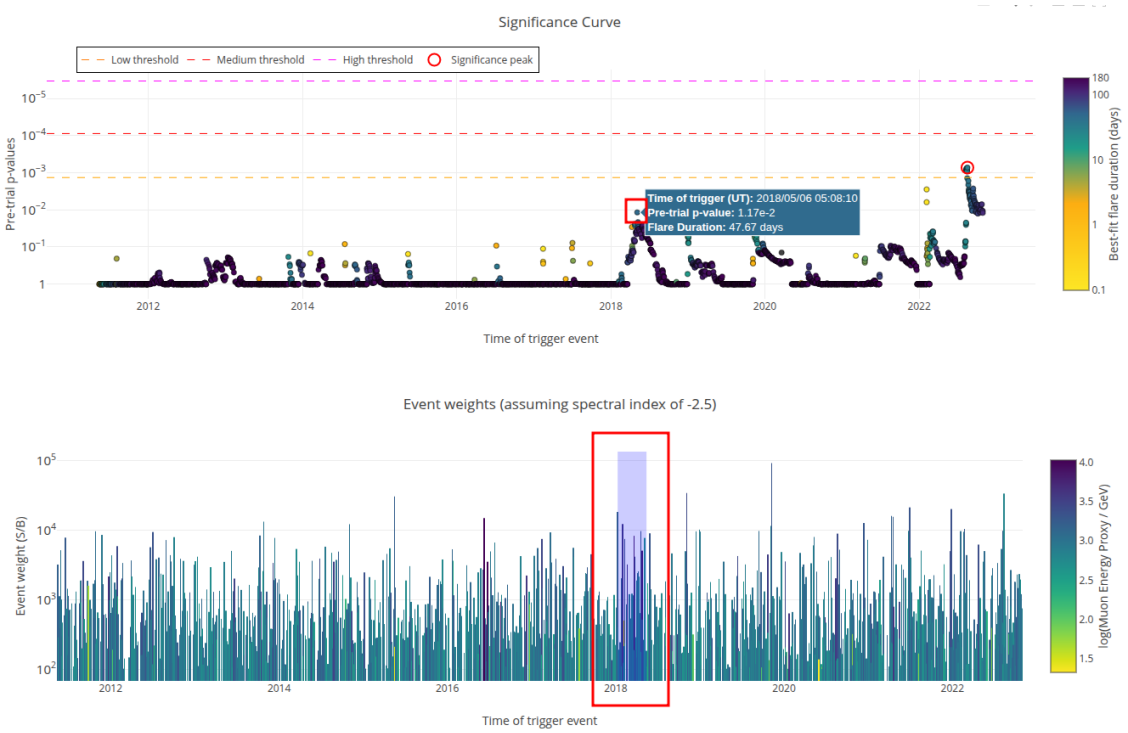


Figure 5.3: Interactive visualization of event weights associated with a selected trigger. The upper panel shows the significance curve over time, where a specific trigger (highlighted with a red marker) has been selected. The lower panel displays the corresponding event weights, computed assuming a spectral index of $\gamma = -2.5$. The shaded region indicates the best-fit flare window, clearly showing the set of events contributing to the cluster. The color scale encodes the reconstructed muon energy proxy for each event. This advanced diagnostic tool is implemented in the backend but currently hidden from the most updated version of the web interface.

The weight calculation is done following the same logic seen in Section 3.3. It relies on two probability density functions: a spatial PDF that evaluates how likely an event to be signal based on its angular separation, and an energy PDF that compares the observed energy proxy to expected distributions under different spectral assumptions.

This module, fully integrated into the web backend, is capable of dynamically selecting a region of interest around a source, computing the weights efficiently, and filtering out events below a configurable threshold. All necessary PDFs are cached to minimize latency during user interaction.

Although fully implemented, this visualization remains hidden in the public version of the web interface. The rationale behind this decision is twofold: first, the plot offers limited practical value to most users outside the collaboration; second, the algorithm behind the weights is not widely known within the community, and exposing it without proper context could lead to confusion or misinterpretation. The infrastructure remains in place, however, and can be reactivated easily should future use cases require it.

Although fully implemented and functional, this visualization was ultimately disabled by default in the unified version of the platform. The decision was based on multiple factors:

- The added complexity was deemed of limited benefit for the broader user community, who may not be familiar with the internal workings of the clustering algorithm.
- Concerns were raised about the potential for misinterpretation of event-level weights, particularly by non-expert users.
- The added computational cost and dependencies (e.g., static PDFs) introduced additional maintenance burdens.

Nevertheless, the system was designed with modularity in mind, and this functionality can be re-enabled in the future. All required data and computation modules remain present in the backend and can be exposed through a simple configuration switch if deemed appropriate.

This case highlights one of the challenges of developing open tools for collaboration: balancing transparency with interpretability. While the real-time system continues to evolve, having these technical features ready ensures that future demands for more granular diagnostics can be met without reengineering the tool.

5.3 Multi-threshold system

One of the major additions implemented in the web interface during this work is the support for a *multi-threshold alert system*. This feature was introduced following discussions within the research group, with the goal of providing external observatories with a more flexible and informative classification of alert significance. Instead of using a single hard threshold, the new

system defines multiple levels of statistical significance, allowing each user or observatory to choose their own follow-up strategy—whether by monitoring all potentially interesting flares, or by focusing only on the most exceptional cases.

This update aligns with a recent contribution presented at the ICRC 2025 [108], which outlines the extension of the GFU cluster alerts dissemination channels to include NASA's GCN (General Coordinates Network). The multi-threshold scheme developed in this work provides a natural foundation for mapping the internal classification system to the upcoming public alerts via GCN. To avoid confusion with the existing single-neutrino alert scheme (which uses the labels "gold" and "bronze"), the new scale adopts a distinct but analogous labeling: low, medium, and high.

The thresholds were selected based on the same statistical framework used to define the original 3σ baseline. Using simulations to estimate the false alert rate (FAR) over one year of data, the group decided to define:

- A **low** threshold targeting $\sim$20 false alerts per year (current base threshold),
- A **medium** threshold targeting $\sim$1 false alert per year,
- A **high** threshold targeting $\sim$0.5 false alerts per decade.

These criteria correspond approximately to significance thresholds of 3.75σ and 4.5σ, respectively. The baseline 3σ threshold remains mandatory, while the additional levels are configured through the system's settings file.

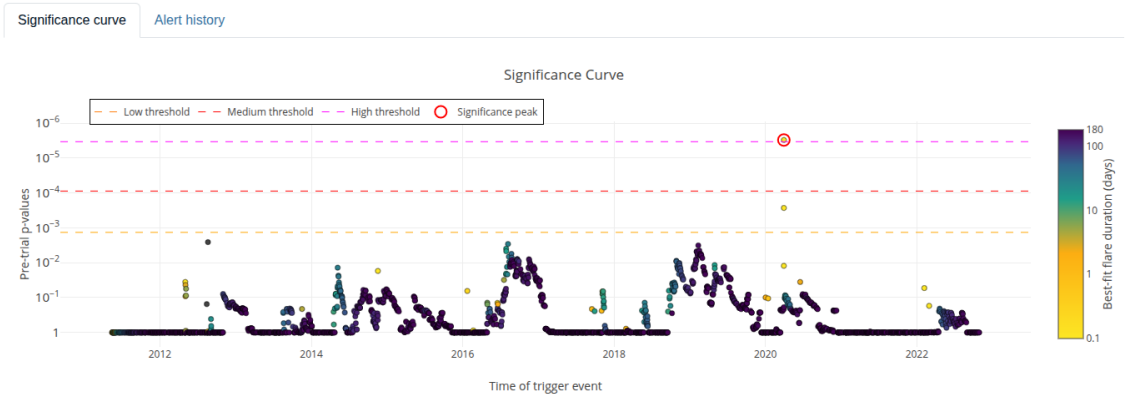
From an implementation perspective, the multi-threshold logic is fully integrated into the platform's configuration system. Each optional level can be defined with an associated color and label, ensuring consistent integration across the entire interface. The assigned levels are reflected in both the alert table and the detailed source view:

- In the **active** and **archived alert tables**, the level is displayed using star icons: one star for low, two for medium, and three for high significance.
- In the **source detail view**, the thresholds are represented as horizontal dashed lines on the significance curve, color-coded according to the configuration.

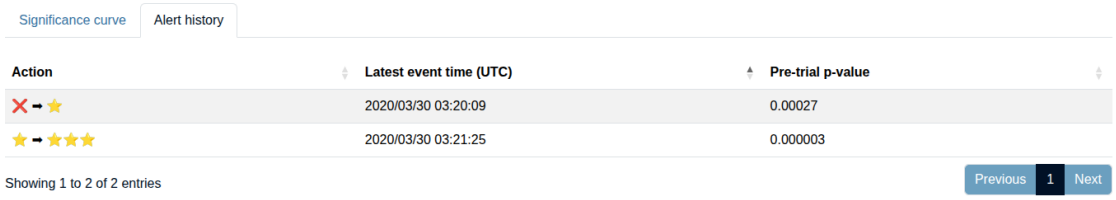
This classification also affects alert propagation (see Figure 5.6). As described earlier in Section 5.1.2, alerts now become eligible for dissemination not only when a flare initially crosses the baseline threshold, but also when a subsequent trigger from the same source exceeds a higher threshold level. Alerts are never issued when a flare decreases in significance.

Source name	Right ascension (deg)	Declination (deg)	Duration (days)	Latest event time (UTC)	Signal events	Spectral index	Pre-trial p-value	Level
TXS 1148+592	177.85	58.99	0.05	2020/03/30 03:21:25	3.99	-2.17	0.000003	★★★
1ES 0347-121	57.35	-11.98	0.29	2015/01/01 03:28:37	3.93	-2.47	6.5×10^{-7}	★★★
B3 0650+453	103.60	45.25	50.26	2022/09/14 23:36:08	14.32	-3.45	0.000085	★★
SBS 1646+499	251.87	49.84	48.45	2021/12/03 06:29:37	7.57	-2.06	0.000053	★★
1ES 1215+303	184.48	30.11	25.09	2016/07/05 22:52:48	12.06	-3.31	0.000086	★★
B2 1215+30	184.47	30.12	25.09	2016/07/05 22:52:48	12.08	-3.31	0.000085	★★
1ES 1215+303	184.48	30.11	16.31	2016/06/27 04:19:01	11.16	-3.49	0.00003	★★
B2 1215+30	184.47	30.12	16.31	2016/06/27 04:19:01	11.14	-3.34	0.000029	★★
1ES 0347-121	57.35	-11.98	0.29	2015/01/01 03:23:33	2.92	-2.30	0.000059	★★
RX J1702.6+3115	255.66	31.26	142.75	2014/07/17 14:49:38	22.08	-4.00	0.00008	★★

Figure 5.4: Archival alert table with threshold levels shown as star icons (surrounded by a red square). This visual scale provides users with a quick assessment of alert significance. Three stars mean high, two mean medium and one means low threshold.



(a) Significance curve with additional thresholds.



(b) Archival alert timeline with icons indicating threshold transitions. The cross illustrated the transition from an inactive state to an active one.

Figure 5.5: Detailed view of a monitored source. The top panel shows the significance evolution with visual thresholds, and the bottom panel displays the corresponding historical log of alert actions.

Additionally, the platform includes a dedicated tab for historical alerts, where each change in significance level is logged. The corresponding action (e.g., promotion to a higher threshold)

is shown with intuitive icons to help users interpret the flare evolution over time. Figures 5.4 and 5.5 llustrate these components respectively.

This multi-threshold strategy represents an important step towards a more modular, and community-driven dissemination system.

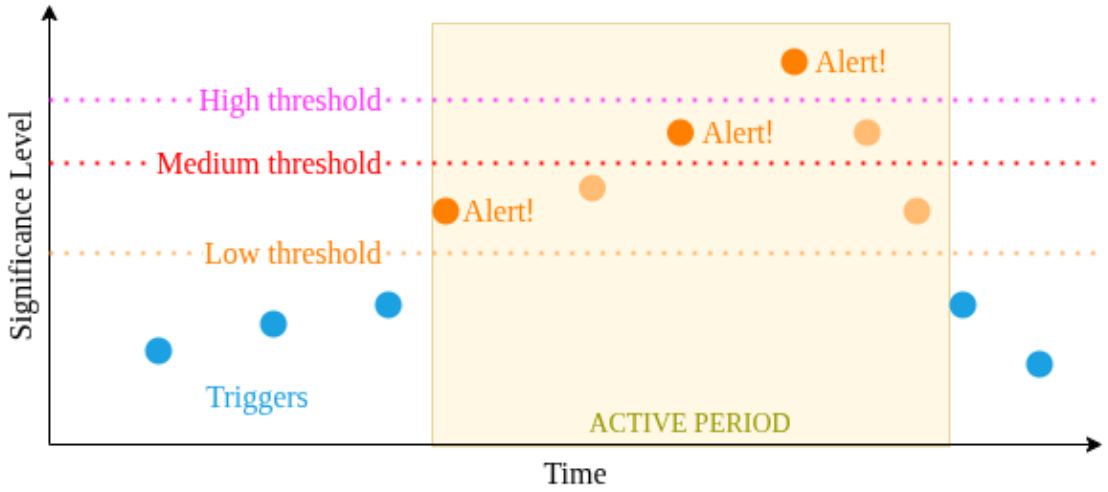


Figure 5.6: Schematic representation of the multi-threshold system. Alerts can be issued both at the initial crossing of the baseline threshold and upon promotion to a higher significance level.

6 Conclusions

6.1 Summary

Throughout this thesis, we have followed a path that begins with the foundations of multimessenger astrophysics and leads to the development of a tool that supports its practical implementation. We began by exploring the theoretical motivations that make neutrinos such powerful cosmic messengers: their ability to escape dense astrophysical environments and point back to their sources makes them essential in understanding the high-energy astrophysics. We then turned our attention to IceCube, the neutrino observatory located at the South Pole, which transforms Antarctic ice into a gigantic Cherenkov detector and has become a cornerstone of this scientific field.

Within IceCube's broader infrastructure, we focused on its realtime alert system, a software framework that transforms raw neutrino detections into timely alerts that can trigger follow-up observations. In particular, we studied the GFU alerts, which are designed to detect clusters of neutrino events around preselected astrophysical candidates. These alerts are a practical expression of the multimessenger vision: a way for IceCube to "call" the attention of other observatories when something statistically interesting is happening in the neutrino sky.

While studying this system, we encountered both its strengths and its limitations. On one hand, it provides a robust framework for near real-time analysis and dissemination. On the other, the tools used to communicate and explore these alerts, especially in a broader and open

context, were limited. It became clear that, if IceCube is to fully embrace public alert sharing, a new kind of interface would be needed: one that is not only technically robust, but also flexible and user-friendly.

This realization led to the core contribution of this thesis: the design and implementation of FlareWatch, a new interactive web platform for real-time IceCube GFU cluster alerts. This tool is fully integrated with the alert pipeline and provides an intuitive visualization of the alert stream as well as dynamic significance curves. Unlike the legacy system, it is capable of displaying all analysis triggers, including sub-threshold and post-alert activity, offering a complete picture of the temporal behavior of monitored sources.

In addition to that, and to provide external observers with clearer criteria for interpreting alerts, we introduced a multi-threshold alert classification, which assigns alerts to different significance tiers based on their statistical strength. The tool is also designed with maintainability and performance in mind, using in-memory caching and modular architecture to ensure responsiveness and scalability.

What has been built is not a prototype, but a production-ready platform. Although it still awaits final review by the collaboration, it already meets the technical and scientific requirements for deployment. Its development has involved both conceptual and practical challenges, from designing visualizations that reflect statistical meaning, to synchronizing with internal infrastructure, to thinking carefully about what it means to "open up" IceCube's realtime data stream. The outcome is a tool that enables both better science and better collaboration.

6.2 Outlook

Looking forward, this platform opens up several promising directions, some of which are already under informal discussion within the collaboration, and others proposed here as natural extensions. One of the most immediate is the integration of all-sky alerts, which would extend the platform beyond source-specific monitoring and enable the visualization of unexpected signals across the full neutrino sky.

Other improvements envisioned include the addition of dynamic likelihood visualizations, allowing users to explore the spatial and statistical structure around each alert in real time. From a technical standpoint, the backend could be enhanced with better data access mechanisms and new components such as a monitoring and administration interface, improving its maintainability over the long term.

In a more speculative but compelling direction, this platform could eventually evolve into

a centralized multimessenger dashboard for IceCube, capable of aggregating different types of alerts and serving as a public interface to IceCube's real-time science. By unifying these efforts in a single and coherent tool, we would not only improve how the observatory communicates science, but also how it practices it: through faster coordination, broader participation, and deeper understanding.

In summary, this thesis has aimed to bridge a gap between high-level scientific goals and the technical tools needed to support them. The platform developed here is a key step in that direction: both a practical solution and a statement of intent for a more open and more connected future in neutrino astronomy.

Appendices

A

Technical implementation details

This appendix provides technical documentation and implementation notes that complement the system described in the main chapters. It is intended to support future maintenance, integration, and potential extensions of FlareWatch. This section serves both as a developer guide and as documentation.

The project was developed following the PEP[1] (Python Enhancement Proposal) guidelines, which define a set of best practices and coding conventions for Python code. These standards are also endorsed by the IceCube collaboration to ensure clarity, maintainability, and consistency across different codebases. In particular, style compliance was enforced automatically throughout development using modern tooling, in line with IceCube's internal software quality practices.

The material included in this appendix provides an overview of the main technologies used in the project, detailed documentation of the REST API endpoints, and the system configuration, including environment variables and the mechanism for adding new significance thresholds. It also addresses key security concerns and the strategies implemented to mitigate them, as well as performance insights based on the average response times of the main API routes.

Overall, this appendix is designed to provide a comprehensive technical overview of the tool, facilitating its deployment, adaptation, or further development by other members of the IceCube collaboration or the broader astrophysical community.

[1] https://peps.python.org/pep-0008/

A.1 List of Used Technologies

Table A.1: Core technologies used in the development of the alert visualization platform.

Technology	Description
Python 3.13	Main programming language used for backend logic and data handling.
UV	Fast Python package manager built in Rust, used to install and manage dependencies with significant speed improvements over pip.
NumPy	Library for efficient numerical operations on arrays, used to compute flare durations and manage cached trigger data.
Flask	Lightweight WSGI web framework used to implement the HTTP server, API endpoints, and templating engine.
Docker	Containerization platform used to isolate the application environment, simplify deployment, and ensure reproducibility.
Pytest	Framework for writing and running tests, used to verify correctness of backend logic and API responses.
Ruff	All-in-one linter and formatter for Python, used to automatically enforce PEP8 and IceCube coding standards at every code change.
Pre-Commit	Framework used to manage and run automated checks on code before each git commit. It ensures that tools like Ruff are executed consistently across all contributors, enforcing code quality and style compliance.
JavaScript	Language for the client-side logic, enabling interactivity and dynamic behavior.
jQuery	Simplifies AJAX requests used for fetching and updating alert data dynamically.
DataTables.js	JavaScript plugin for rendering interactive tables with filtering, pagination, and sorting functionalities.
Plotly.js	Powerful library for rendering interactive plots, including the significance curves and event-weight visualizations.
Bootstrap 5	Frontend UI framework used to create responsive layouts and consistent styling, adapted to match the official IceCube color palette.

A.2 API Documentation

The table below summarizes the main REST API endpoints exposed by the backend. These endpoints are consumed asynchronously by the web interface and are grouped according to the view or component that makes use of them.

Table A.2: REST API endpoints grouped by functionality and HTTP method.

Endpoint	Method	Description
Views		
/	GET	Redirects to `/alert/active` as the homepage.
/alert/active	GET	Loads the view for currently active alerts.
/alert/archival	GET	Loads the view for archived alerts.
/source/{source-name}	GET	Loads the detailed view of a monitored source.
/support	GET	Loads the documentation and support page.
Data Loading (Asynchronous)		
/api/alert/active	GET	Returns active alerts, grouped by source and sorted by time.
/api/alert/archival	GET	Returns historical alerts.
/api/alert/{source-name}	GET	Returns the alert history for a given source, including significance level transitions.
/api/source/{source-name}/significance-curve	GET	Returns the significance curve data (optionally including event weights).
Trigger Ingestion		
/api/alert-trigger	POST	Entry point to ingest new alert triggers into the web tool (authenticated with API token).

A.3 Configuration and Threshold Management

The web tool relies on environment variables and internal configuration modules to control its behavior. This modular design allows the system to switch between development and production modes, and to modify alert processing parameters without altering the core codebase.

Environment Variables

At startup, the application reads from a `.env` file or directly from the system's environment. The following variables control the main runtime settings:

Table A.3: Main environment variables used by the application.

Variable	Description
`FLASK_ENV`	Flask runtime environment. Accepts `development` or `production`.
`SECRET_KEY`	Secret key used for secure session handling and CSRF protection.
`HOST`, `PORT`	Host and port for the application server. Typically `localhost:5000` during development.
`APIKEY`	Secret token used to authenticate incoming POST requests to the alert trigger endpoint.
`SOURCE_TRIGGERS`	(Optional) Path to the JSON file containing the list of monitored sources and their associated metadata.
`SPATIAL_BACKGROUND_PDF`	(Optional) Path to the background spatial probability density function used in event-weight diagnostics.
`ENERGY_PDFS`	(Optional) Path to the JSON file containing energy PDFs for signal and background under different spectral assumptions.

These variables are loaded at runtime and injected into the application context to configure file paths, enable optional features, and secure API access.

Threshold Definitions and Statistical Parameters

In addition to environment-based configuration, the application includes a dedicated Python module (`config.py`) that defines internal parameters for statistical analysis and system behavior. This configuration centralizes the definition of alert thresholds and diagnostic options, ensuring consistency across backend logic and frontend rendering.

The threshold system is built around a base level (currently 3σ) and a list of additional thresholds used to distinguish between low, medium, and high significance alerts. Each threshold is defined by:

- A numerical `value` (e.g., 3.75 or 4.5), representing the significance level in standard deviations.
- A human-readable `label` (e.g., `Medium`, `High`) used for legends and classification.
- A display `color`, used to highlight alerts and annotate the significance curve.

An additional parameter, the spectral index `gamma`, is also defined in this module and used for the optional computation of event-level weights.

At startup, the application reads these values and propagates them to the statistical pipeline, frontend visualizations, and alert classification logic.

Extending Thresholds

To add a new alert level, the user must:

1. Define a new threshold entry in the configuration module, specifying its `value`, `label`, and `color`.
2. Restart the backend service to load the new configuration.
3. Optionally, adjust frontend styling or icons if the new level introduces visual elements.

No additional logic is required—the system dynamically adapts to the thresholds provided. This design allows easy experimentation with new significance levels and supports future tuning of the alert system without code changes.

A.4 Security Considerations

Security was a key concern throughout the development of this tool, especially considering its connection to internal alert infrastructure and its public-facing nature. Several mechanisms have been implemented to mitigate potential risks and ensure robustness against common vulnerabilities [109].

Client-Side Hardening

The frontend is served with a strict Content Security Policy (CSP), which restricts the execution of inline scripts and enforces control over the origin of loaded resources. All third-party JavaScript libraries—including `Plotly.js`, `DataTables.js`, `jQuery`, and `Bootstrap`—are bundled locally and served from the application container itself. This eliminates the need for external Content Delivery Networks (CDNs) and mitigates risks associated with external script injection (dependency hijacking). The server also includes appropriate HTTP security headers, depending on the deployment context.

API Endpoint Protection

The ingestion endpoint responsible for receiving real-time alert triggers is secured via an API token that must be included in the request headers. Requests without a valid token are rejected before any processing occurs. This protects the system from unauthorized data injection and ensures that only trusted sources (such as IceCube's alert generation modules) can submit new triggers.

Vulnerability Scanning and Package Audit

A custom script has been developed to automatically scan the Python dependency tree using `pip-audit`. This tool checks installed packages for known vulnerabilities using the Python Packaging Advisory Database. Although this script is currently executed manually, it is intended to be integrated into the future deployment pipeline to automate periodic checks and prompt dependency updates as necessary.

Deployment Isolation

The system is designed to run as a self-contained service within a dedicated container. In a production scenario, this container is expected to be deployed on a separate virtual machine or service node, decoupled from the core IceCube `realtime` software stack. As such, any performance degradation or security incident affecting the web application would remain isolated and would not impact the realtime processing infrastructure.

A.5 Performance Metrics

To ensure responsive interaction, the web application was optimized for low-latency data access and rendering. Performance benchmarks were obtained using realistic alert datasets under a single-user scenario. The following table summarizes average response times for key endpoints:

Table A.4: Average response times for all endpoints (single user, localhost).

Endpoint	Avg. Response Time
`/api/alert/archival`	450 ms
`/api/alert/active`	200 ms
`/api/alert-trigger`	200 ms
`/api/source/{source-name}/significance-curve`	24 ms
`/api/alert/{source-name}`	20 ms
`/source/{source-name}` (HTML view)	20 ms
`/alert/active` (HTML view)	15 ms
`/alert/archival` (HTML view)	15 ms
`/support` (HTML view)	15 ms

These times are achieved because of efficient in-memory caching and lightweight data models. For production deployment, additional scalability strategies (e.g. external caching, pagination, or async queries) may be explored depending on traffic.

References

[1] Victor F. Hess. Über Beobachtungen der durchdringenden Strahlung bei sieben Freiballonfahrten. *Phys. Z.*, 13:1084–1091, 1912.

[2] Péter Mészáros, Derek B. Fox, Chad Hanna, and Kohta Murase. Multi-messenger astrophysics. *Nature Reviews Physics*, 1(10):585–599, October 2019.

[3] Maurizio Spurio. *Particles and Astrophysics: A Multi-Messenger Approach*. Astronomy and Astrophysics Library. Springer, Cham, illustrated edition, 2014.

[4] Carmelo Evoli. The cosmic-ray energy spectrum, October 2018.

[5] M. Ave, P. J. Boyle, F. Gahbauer, C. Höppner, J. R. Hörandel, M. Ichimura, D. Müller, and A. Romero-Wolf. Composition of primary cosmic-ray nuclei at high energies. *The Astrophysical Journal*, 678(1):262–273, May 2008.

[6] Maurizio Spurio. *Particles and Astrophysics: A Multi-Messenger Approach*. Astronomy and Astrophysics Library. Springer International Publishing, 2014.

[7] K. Greisen. End to the cosmic ray spectrum? *Physical Review Letters*, 16(17):748–750, 1966.

[8] G. T. Zatsepin and V. A. Kuzmin. Upper limit of the spectrum of cosmic rays. *JETP Lett.*, 4:78–80, 1966.

[9] Acelerando la Ciencia. Extensive air shower, s.f. Accessed: 2025-06-04. Image titled "Extensive Air Shower".

[10] Malcolm S. Longair. *High Energy Astrophysics*. Cambridge University Press, 3 edition, 2011.

[11] George B. Rybicki and Alan P. Lightman. *Radiative Processes in Astrophysics*. Wiley-VCH, 1986.

[12] W. B. Atwood et al. The Large Area Telescope on the Fermi Gamma-Ray Space Telescope Mission. , 697(2):1071–1102, June 2009.

[13] Ciro Bigongiari. The magic telescope, 2005.

[14] Gerd Pühlhofer, Fabian Leuschner, and Heiko Salzmann. *H.E.S.S.: The High Energy Stereoscopic System*, page 1–41. Springer Nature Singapore, December 2023.

[15] T.C Weekes et al. Veritas: the very energetic radiation imaging telescope array system. *Astroparticle Physics*, 17(2):221–243, May 2002.

[16] Zhen Cao et al. The large high altitude air shower observatory (lhaaso) science book (2021 edition), 2022.

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

[18] Annarita Margiotta. The km3net deep-sea neutrino telescope. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 766:83–87, December 2014.

[19] P. Padovani, D. M. Alexander, R. J. Assef, B. De Marco, P. Giommi, R. C. Hickox, G. T. Richards, V. Smolčić, E. Hatziminaoglou, V. Mainieri, and M. Salvato. Active galactic nuclei: what's in a name? *The Astronomy and Astrophysics Review*, 25(1), August 2017.

[20] Aartsen et al. Multimessenger observations of a flaring blazar coincident with high-energy neutrino icecube-170922a. *Science*, 361(6398), July 2018.

[21] R. Abbasi et al. Evidence for neutrino emission from the nearby active galaxy ngc 1068. *Science*, 378(6619):538–543, November 2022.

[22] E. Fermi. On the origin of the cosmic radiation. *Physical Review*, 75:1169–1174, 1949.

[23] M. Ackermann et al. Detection of the characteristic pion-decay signature in supernova remnants. *Science*, 339(6121):807–811, 2013.

[24] P. Veres, C. A. Meegan, E. Bissaldi, and et al. Grb 221009a: The brightest gamma-ray burst of all time. *The Astrophysical Journal Letters*, 939(1):L1, 2022.

[25] IceCube Collaboration. Limits on neutrino emission from grb 221009a from mev to pev using the icecube neutrino observatory. *The Astrophysical Journal Letters*, 944(1):L15, 2023.

[26] J. Babson et al. Cosmic Ray Muons in the Deep Ocean. *Phys. Rev. D*, 42:3613–3620, 1990.

[27] I. A. Belolaptikov et al. The Baikal underwater neutrino telescope: Design, performance and first results. *Astropart. Phys.*, 7:263–282, 1997.

[28] M. Ageron et al. ANTARES: the first undersea neutrino telescope. *Nucl. Instrum. Meth. A*, 656:11–38, 2011.

[29] Francis Halzen et al. The AMANDA neutrino telescope. *Nucl. Phys. B Proc. Suppl.*, 77:474–485, 1999.

[30] Eli Waxman and John Bahcall. High energy neutrinos from astrophysical sources: An upper bound. *Physical Review D*, 59(2), December 1998.

[31] M.G. Aartsen et al. Evidence for high-energy extraterrestrial neutrinos at the icecube detector. *Science*, 342(6161), November 2013.

[32] Ackermann et al. Optical properties of deep glacial ice at the South Pole. *Journal of Geophysical Research (Atmospheres)*, 111(D13):D13203, July 2006.

[33] Felipe Pedreros. The icecube lab under the stars, 2013. Image retrieved from IceCube official website.

[34] BBC News. Amundsen-scott research station, antarctica, 2016. Image retrieved from BBC News.

[35] P. A. Čerenkov. Visible radiation produced by electrons moving in a medium with velocities exceeding that of light. *Phys. Rev.*, 52:378–379, Aug 1937.

[36] J. Ahrens et al. Muon track reconstruction and data selection techniques in amanda. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 524(1–3):169–194, May 2004.

[37] Aartsen et al. The icecube neutrino observatory: instrumentation and online systems. *Journal of Instrumentation*, 12(03):P03012–P03012, March 2017.

[38] Glenn F. Knoll. *Radiation Detection and Measurement*. John Wiley & Sons, 4 edition, 2010.

[39] Abbasi et al. Calibration and characterization of the icecube photomultiplier tube. *Nuclear Instruments and Methods in Physics Research Section A Accelerators Spectrometers Detectors and Associated Equipment*, 618:139–152, 07 2010.

[40] Abbasi et al. The design and performance of icecube deepcore. *Astroparticle Physics*, 35(10):615–624, May 2012.

[41] Aartsen et al. Measurement of the cosmic ray energy spectrum with icetop-73. *Physical Review D*, 88(4), August 2013.

[42] Aartsen et al. The icecube neutrino observatory: instrumentation and online systems. *Journal of Instrumentation*, 12(03):P03012–P03012, March 2017.

[43] M.G. Aartsen et al. Characterization of the atmospheric muon flux in icecube. *Astroparticle Physics*, 78:1–27, May 2016.

[44] J. A. Formaggio and G. P. Zeller. From ev to eev: Neutrino cross sections across energy scales. *Reviews of Modern Physics*, 84(3):1307–1341, September 2012.

[45] M. G. Aartsen et al. Search for sources of astrophysical neutrinos using seven years of icecube cascade events. *The Astrophysical Journal*, 886(1):12, November 2019.

[46] Thomas Kintscher. *Rapid Response to Extraordinary Events: Transient Neutrino Sources with the IceCube Experiment*. PhD thesis, Humboldt U., Berlin, Humboldt U., Berlin, 2020.

[47] V. J. Stenger. Track fitting for dumand-ii octagon array. Technical Report HDC-1-90, University of Hawaii, 1990.

[48] D. Pandel. *Bestimmung von Wasser- und Detektorparametern und Rekonstruktion von Myonen bis 100 TeV mit dem Baikal- Neutrinoteleskop NT-72*. 1996.

[49] P. H. C. Eilers and B. D. Marx. Flexible smoothing with b-splines and penalties. *Statistical Science*, 11:89–121, 1996.

[50] N. Whitehorn, J. van Santen, and S. Lafebre. Penalized splines for smooth representation of high-dimensional monte carlo datasets. *Computer Physics Communications*, 184(10):2214–2220, 2013.

[51] Harald Cramér. *Mathematical Methods of Statistics*. Princeton Mathematical Series. Princeton University Press, 1999.

[52] C. Radhakrishna Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37, 1945.

[53] R. Abbasi et al. Time-integrated searches for point-like sources of neutrinos with the 40-string icecube detector. *The Astrophysical Journal*, 732(1), 2011.

[54] Jens Lünemann. *Suche nach Dunkler Materie in Galaxien und Galaxienhaufen mit dem Neutrinoteleskop IceCube*. Phd thesis, Johannes Gutenberg-Universität Mainz, 2013.

[55] T. Neunhöffer. Estimating the angular resolution of tracks in neutrino telescopes based on a likelihood analysis. *Astroparticle Physics*, 25:220–225, 2006.

[56] M. G. Aartsen et al. Searches for extended and point-like neutrino sources with four years of icecube data. *The Astrophysical Journal*, 796(2), 2014.

[57] M. G. Aartsen et al. All-sky search for time-integrated neutrino emission from astrophysical sources with 7 yr of icecube data. *The Astrophysical Journal*, 835(2), 2017.

[58] L. Rädel and C. Wiebusch. Calculation of the cherenkov light yield from electromagnetic cascades in ice with geant4. *Astroparticle Physics*, 44:102–113, 2013.

[59] M. G. Aartsen et al. Energy reconstruction methods in the icecube neutrino telescope. *Journal of Instrumentation*, 9:P03009, 2014.

[60] J.-D. Zornoza and D. Chirkin. Muon energy reconstruction and atmospheric neutrino spectrum unfolding with the icecube detector. In *30th International Cosmic Ray Conference (ICRC)*, volume 5, Mérida, Mexico, 2007. Universidad Nacional Autónoma de México.

[61] R. Abbasi et al. An improved method for measuring muon energy using the truncated mean of de/dx. *Nuclear Instruments and Methods in Physics Research Section A*, 703:190–198, 2013.

[62] M.G. Aartsen et al. The icecube realtime alert system. *Astroparticle Physics*, 92:30–41, June 2017.

[63] M. Ackermann, E. Bernardini, N. Galante, F. Goebel, M. Hayashida, K. Satalecka, M. Tluczykont, and R. M. Wagner. Neutrino triggered target of opportunity (ntoo) test run with amanda-ii and magic, 2007.

[64] M. G. Aartsen et al. Very high-energy gamma-ray follow-up program using neutrino triggers from icecube. *JINST*, 11:P11009, 2016. (IceCube, MAGIC and VERITAS Collaborations).

[65] M. G. Aartsen et al. The detection of a sn iin in optical follow-up observations of ice-cube neutrino events. *Astrophysical Journal*, 811:52, 2015. (IceCube, PTF, Swift, Pan-STARRS1 Science Consortium).

[66] C. W. Akerlof et al. The rotse-iii robotic telescope system. *Publications of the Astronomical Society of the Pacific*, 115:132–140, 2003.

[67] N. M. Law et al. The palomar transient factory: System overview, performance and first results. *Publications of the Astronomical Society of the Pacific*, 121:1395–1408, 2009.

[68] N. Gehrels et al. The swift gamma-ray burst mission. *Astrophysical Journal*, 611:1005–1020, 2004. (Swift Science Team).

[69] K. Maine, C. Devieux, and P. Swan. Overview of iridium satellite network. In *Proceedings of WESCON'95*, 1995. Presented at WESCON'95.

[70] M. Voge. *Searches for Neutrinos from Supernovae Using Cherenkov In-Ice Detectors*. Phd thesis, Humboldt-Universität zu Berlin, 2016.

[71] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[72] R. B. Vimieiro, L. R. Borges, and M. A. C. Vieira. Open-source reconstruction toolbox for digital breast tomosynthesis. In R. Costa-Felix, J. Machado, and A. Alvarenga, editors, *XXVI Brazilian Congress on Biomedical Engineering*, volume 70/2 of *IFMBE Proceedings*, Singapore, 2019. Springer.

[73] Tim Bray. The javascript object notation (json) data interchange format. *RFC*, 7159:1–16, 2014.

[74] Zeromq – distributed messaging. https://zeromq.org/, 2024. Accessed: 2025-06-02.

[75] Supervisor Developers. Supervisor: A process control system, 2024. Accessed: 2025-06-02.

[76] G. Punzi. Sensitivity of searches for new signals and its optimization. In *Statistical Problems in Particle Physics, Astrophysics and Cosmology (PHYSTAT2003)*, SLAC, Stanford, CA, 2003. https://doi.org/10.48550/arXiv.physics/0308063.

[77] J. Braun, J. Dumm, F. De Palma, C. Finley, A. Karle, and T. Montaruli. Methods for point source analysis in high energy neutrino telescopes. *Astroparticle Physics*, 29:299–305, 2008.

[78] M. L. Ahnen et al. Detection of very high energy gamma-ray emission from the gravitationally lensed blazar qso b0218+357 with the magic telescopes. *Astronomy amp; Astrophysics*, 595:A98, November 2016.

[79] Caterina Boscolo Meneguolo, Elisa Bernardini, Jean-Pierre Jonckheere, Sarah Mancina, and Koji Noda. An updated list of target sources for icecube neutrino cluster alerts. Contribution to the ICRC 2025, Geneva, Switzerland. Oral presentation, not yet published, 2025.

[80] Leo Singer and Judith Racusin. General Coordinates Network (GCN): NASA's Next Generation Time-Domain and Multimessenger Astronomy Alert System. In *American Astronomical Society Meeting Abstracts*, volume 241 of *American Astronomical Society Meeting Abstracts*, page 108.02, January 2023.

[81] Python Software Foundation. Python 3.13 release notes. https://www.python.org/downloads/release/python-3130, 2024. Accessed: 21-05-2025.

[82] Travis E Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3):10–20, 2007.

[83] K.J. Millman and M. Aivazis. Python for scientists and engineers. *Computing in Science & Engineering*, 13(2):9–12, 2011.

[84] Miguel Grinberg. *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, 2nd edition, 2018.

[85] Pallets Projects. Jinja2 documentation. https://jinja.palletsprojects.com, 2023.

[86] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. *Pattern-Oriented Software Architecture: A System of Patterns*. John Wiley & Sons, 1996.

[87] Trygve Reenskaug. The model-view-controller (mvc) paradigm, 1979. Xerox PARC internal report.

[88] MDN Web Docs. Mvc. https://developer.mozilla.org/es/docs/Glossary/MVC. Accessed: 21-05-2025.

[89] Matthew Hoy. Html5: A new standard for the web. *Medical reference services quarterly*, 30:50–5, 01 2011.

[90] Swathi Kaluvakuri. Harnessing the potential of css: An exhaustive reference for web styling. *Engineering International*, 4:95–110, 12 2016.

[91] Ecma International. ECMAScript Language Specification, 14th Edition (ECMA-262). https://tc39.es/ecma262/, 2023. Accessed: 2025-05-07.

[92] Q. Wang and Y. Sun. An introduction to ajax and web application development. *Journal of Computing Sciences in Colleges*, 26(5):140–147, 2011.

[93] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[94] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

[95] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, et al. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.

[96] Benoît Chesneau and contributors. Gunicorn: Python wsgi http server for unix. https://gunicorn.org/, 2024. Version 21.2.0, Accessed: 2025-05-07.

[97] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*, (239):2, 2014.

[98] Carl Boettiger. An introduction to docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1):71–79, 2015.

[99] Adam Wiggins. The twelve-factor app, 2011.

[100] Holger Krekel and contributors. pytest: helps you write better programs. https://docs.pytest.org/, 2024. Accessed: 2025-06-08.

[101] Kenneth Reitz et al. Requests: Http for humans. https://docs.python-requests.org/en/latest/, 2023. Accessed: 2024-12-20.

[102] Python Software Foundation. Python secrets module. https://docs.python.org/3/library/secrets.html, 2023. Accessed: 2024-12-20.

[103] DataTables. Datatables - table plug-in for jquery, 2024.

[104] Astropy Collaboration. The astropy project: coordinates transformation tools, 2022.

[105] M. Wenger, F. Ochsenbein, D. Egret, P. Dubois, F. Bonnarel, S. Borde, F. Genova, G. Jasniewicz, S. Laloë, S. Lesteven, and R. Monier. The simbad astronomical database: The cds reference database for astronomical objects. *Astronomy and Astrophysics Supplement Series*, 143(1):9–22, April 2000.

[106] S. P. Wakely and D. Horan. TeVCat: An online catalog for Very High Energy Gamma-Ray Astronomy. *International Cosmic Ray Conference*, 3:1341–1344, 2008.

[107] Alex Johnson, Chris Parmer, and et al. plotly.js – a high-level, declarative charting library. https://github.com/plotly/plotly.js, 2015. Accessed: 2025-06-05.

[108] Sarah Mancina, Sergio Cuenca, and Elisa Bernardini. New public neutrino alerts for clusters of icecube events. Contribution to the ICRC 2025, Geneva, Switzerland. Oral presentation, not yet published, 2025.

[109] OWASP Foundation. *OWASP Application Security Verification Standard 4.0.3*. Open Worldwide Application Security Project, 2023. https://owasp.org/ASVS.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor Elisa Bernardini for giving me the opportunity to develop this project and for introducing and training me within such a major observatory as IceCube. It has been an honor to contribute real value to this collaboration.

My deepest thanks also go to my co-supervisor Sarah Mancina, who has been there from the very beginning. Thank you for patiently explaining concepts again and again, for standing by me through every challenge, and for helping me overcome all the hurdles. This tool would not have been possible without you.

I also want to thank the rest of the IceCube research group in Padova, Caterina and Diego, for their regular feedback during our weekly meetings and making me feel comfortable even when I was on remote.

Many thanks as well to my fellow master colleagues Jean-Pierre and Alex, who have gone through this journey alongside me and have always been willing to offer feedback and support.

On a special note, I would like to thank those who have been key figures throughout my academic path and made it possible for me to reach this point.

To my mother, I owe my curiosity and fascination for the stars. She is the direct reason why I have always pursued knowledge so persistently. Everything I've achieved is thanks to her, and no words will ever be enough to express my gratitude. To Toni, for encouraging me to always keep going despite all odds. To my father, who I would wish was here to see this milestone. To my grandmother for being a cornerstone when looking for excellence, and to my aunt and uncle, thank you for always being there and pushing me never to give up.

To Isabel, the brightest star in my small universe. Without her, I would most likely never have ended up in Padova. She has been my rock, and together we have navigated every difficulty. We are like muon and neutrino.

To my close friends—Rubén, Xavi, Edgar, Joan, Roger, Víctor, and Pol—thank you for standing by me through every tough moment and for making life easier with your presence. You have been a constant reminder that there is more to life than equations and deadlines—sometimes dangerously so.

To my fellow classmates in Padova—Blanca, Jean Pierre, Too, Julia, Ivan, Giovanni, Isabella, Lauren, Frank, and Augusto—the core of my social life here. Thanks to you, this experience has been truly unforgettable.

Thank you all, from the bottom of my heart, for being part of my life and for walking with me as I reached the dream of becoming an astrophysicist.