



Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y
Tecnología

Máster Universitario en Inteligencia Artificial
Reconstrucción de cascadas atmosféricas extendidas
detectadas en observatorio HAWC

Trabajo fin de estudio presentado por:	Daniel Galán Triguero
Tipo de trabajo:	Desarrollo de software
Director/a:	Ibrahim Torres Aguilar, Javier Rodrigo Villazón Terrazas
Fecha:	10/07/2024

Resumen

Los rayos cósmicos, partículas de alta energía procedentes del espacio, impactan en la atmósfera terrestre creando cascadas atmosféricas extendidas. Estas cascadas se originan en la colisión inicial de los rayos cósmicos con núcleos atómicos presentes en la atmosfera expandiéndose de forma lateral y vertical, siendo detectadas por observatorios especializados.

HAWC, ubicado en la ladera del volcán Sierra Negra, México, detecta estas cascadas mediante detectores Cherenkov en agua. La mayoría de los algoritmos del observatorio están optimizados para rayos gamma por lo que no son óptimos para el estudio de rayos cósmicos.

Este trabajo propone utilizar inteligencia artificial para crear un modelo adaptado específicamente a los rayos cósmicos, permitiendo aportar una de las variables imprescindibles a la hora de reconstruir las cascadas atmosféricas extendidas, la predicción de la posición del núcleo de la cascada.

Palabras clave: Rayos cósmicos, cascadas atmosféricas extendidas, aprendizaje automático

Abstract

Cosmic rays, high-energy particles from space, impact the Earth's atmosphere, creating widespread atmospheric cascades. These cascades originate from the initial collision of cosmic rays with atomic nuclei present in the atmosphere, expanding laterally and vertically, being detected by specialized observatories.

HAWC, located on the slope of the Sierra Negra volcano, Mexico, detects these cascades using Cherenkov detectors. The observatory's current algorithms are implemented for gamma rays, so they are not optimal for the study of cosmic rays.

This work proposes using artificial intelligence to create a model specifically adapted to cosmic rays, making it possible to provide the prediction of one of the essential variables when reconstructing extended atmospheric cascades, the position of the core of the cascade.

Keywords: Cosmic rays, widespread atmospheric cascades, machine learning

No dejes que termine el día sin haber crecido un poco,
sin haber sido feliz, sin haber aumentado tus sueños.
No te dejes vencer por el desaliento.
No permitas que nadie te quite el derecho a expresarte, que es casi un deber.
No abandones las ansias de hacer de tu vida algo extraordinario.
No dejes de creer que las palabras y las poesías,
sí pueden cambiar el mundo.
Pase lo que pase nuestra esencia está intacta.
Somos seres llenos de pasión.
La vida es desierto y oasis.
Nos derriba, nos lastima, nos enseña, nos convierte en protagonistas
de nuestra propia historia.
Aunque el viento sople en contra, la poderosa obra continúa:
Tú puedes aportar una estrofa.
No dejes nunca de soñar, porque en sueños es libre el hombre.
No caigas en el peor de los errores: el silencio.
La mayoría vive en un silencio espantoso.
No te resignes. Huye.
“Emito mis alaridos por los techos de este mundo”, dice el poeta.
Valora la belleza de las cosas simples.
Se puede hacer bella poesía sobre pequeñas cosas,
pero no podemos remar en contra de nosotros mismos.
Eso transforma la vida en un infierno.
Disfruta del pánico que te provoca
tener la vida por delante.
Vívela intensamente, sin mediocridad.
Piensa que en ti está el futuro
y encara la tarea con orgullo y sin miedo.
Aprende de quienes puedan enseñarte.
Las experiencias de quienes nos precedieron
de nuestros “poetas muertos”,
te ayudan a caminar por la vida
La sociedad de hoy somos nosotros: Los “poetas vivos”.
No permitas que la vida te pase a ti sin que la vivas.

Walt Whitman

Por la inmensidad del universo viaja la luz de las estrellas, siendo la velocidad de esta la mayor conocida hasta el momento. Es tan grande este, que a pesar de lo rápido que viaja la luz, podemos observar desde nuestro planeta el resplandor de estrellas hace millones de años apagadas.

Este trabajo está dedicado a todas esas estrellas que, a pesar de habernos dejado hace tiempo, siguen iluminando nuestras vidas.

Índice de contenidos

1. Introducción.....	15
1.1. Motivación.....	15
1.2. Planteamiento del trabajo.....	16
1.3. Estructura de la memoria	17
2. Contexto y estado del arte	18
2.1. Astronomía de altas energías	18
2.1.1. Rayos cósmicos	18
2.1.2. Cascadas atmosféricas extendidas	18
2.1.1. Efecto Cherenkov.....	20
2.1.2. Observatorio HAWC.....	21
2.2. Aprendizaje automático	26
2.2.1. Descripción	26
2.2.2. Aprendizaje supervisado.....	26
2.2.3. Aprendizaje no supervisado.....	30
2.2.4. Redes neuronales artificiales	31
3. Objetivos y metodología de trabajo	39
3.1. Objetivo general	39
3.1.1. Reconstrucción de las cascadas	39
3.2. Objetivos específicos.....	40
3.2.1. Predicción de coordenadas del núcleo de la cascada	40
3.3. Metodología del trabajo.....	40
3.3.1. Obtención de datos	40
3.3.2. Selección de variables.....	42
3.3.3. Entrenamiento del modelo	43
3.3.4. Modelo y predicción	45
4. Identificación de requisitos.....	46
4.1. Problema	46
4.2. Contexto habitual de uso	46
4.3. Requisitos funcionales	47

4.4.	Requisitos no funcionales.....	47
5.	Descripción de la herramienta desarrollada	48
5.1.	Proceso de desarrollo.....	48
5.1.1.	Fase 1: Lectura de datos	48
5.1.2.	Fase 2: Procesamiento de datos	49
5.1.3.	Fase 3: Desarrollo y entrenamiento del modelo	50
5.1.4.	Fase 4: Evaluación.....	50
5.1.5.	Fase 5: Implementación y despliegue.....	50
5.2.	Arquitectura red convolucional utilizada	51
5.3.	Requisitos de ejecución.....	52
5.4.	Ejecución.....	52
5.4.1.	Entrenamiento.....	52
5.4.2.	Predicción	54
5.5.	Archivo de salida predicciones	54
6.	Evaluación	55
6.1.	Descripción de los resultados	55
6.1.1.	Grafica de aprendizaje	55
6.1.2.	Histograma 2D de comparación entre predicciones y valores reales.....	56
6.1.3.	Histogramas de las diferencias en los ejes de coordenadas	60
6.2.	Evaluación de los resultados	62
6.2.1.	Entrenamiento.....	62
6.2.2.	Histogramas 2d	63
6.2.3.	Histogramas diferencias en X e Y.....	63
7.	Conclusiones y trabajo futuro.....	69
7.1.	Conclusiones	69
7.2.	Líneas de trabajo futuro	69
8.	Apéndice	71
8.1.	Histogramas complementarios apartado 6.1.3	71
8.2.	Histogramas complementarios apartado 6.2.3	75
9.	Bibliografía	79

Índice de figuras

Figura 1 Cascada atmosférica extendida	19
Figura 2 - Plano y eje de la cascada	20
Figura 3 - Efecto Cherenkov	21
Figura 4 - Vista aérea observatorio HAWC	22
Figura 5 - Water Cherenkov Detector	23
Figura 6 - Tubo fotomultiplicador (PMT)	24
Figura 7 - Regresión lineal	27
Figura 8 - Ejemplo de clasificación con SVM	29
Figura 9 - Ejemplo de clasificación con árboles de decisión	30
Figura 10 - Agrupación de datos con K-Means	31
Figura 11 - Morfología de la neurona biológica	32
Figura 12 - Comparación neurona biológica / artificial	33
Figura 13 - Función sigmoide	34
Figura 14 - Funciones ReLU y Leaky ReLU	35
Figura 15 - Red neuronal 'fully connected'	36
Figura 16 - Funcionamiento interno entre capas	36
Figura 17 - Red neuronal convolucional	37
Figura 18 – Procesado de información en la capa convolucional	38
Figura 19 - Rendimiento estimador de máxima verosimilitud vs redes neuronales	39
Figura 20 - Variables tanque WCD	41
Figura 21 - Cascada atmosférica extendida de un rayo cósmico detectado en HAWC	43
Figura 22 - Representación imagen RGB en diferentes capas	44
Figura 23 - Arquitectura red convolucional desarrollada	51
Figura 24 - Secuencia entrenamiento	53

Figura 25 - Evolución aprendizaje	55
Figura 26 – Histograma 2D todos los eventos.....	57
Figura 27 – Histograma 2D eventos con logaritmo energía mayor que 1 y menor que 2.....	58
Figura 28 - Histograma 2D eventos con logaritmo energía mayor que 2 y menor que 3	59
Figura 29 - Histograma 2D eventos con logaritmo energía mayor que 3 y menor que 4	59
Figura 30 - Histograma 2D eventos con logaritmo energía mayor que 4 y menor que 5	60
Figura 31 - Histograma diferencias coordenadas X (red convolucional)	61
Figura 32 - Histograma diferencias coordenadas Y (red convolucional)	61
Figura 33 - Entrenamiento con eventos similares.....	62
Figura 34 – Histograma diferencias en coordenada X (valores HAWC)	64
Figura 35 - Histograma diferencias en coordenada Y (valores HAWC)	64
Figura 36 – Relación entre energía del evento y el valor sigma.....	66
Figura 37 – Relación entre energía y el valor de resolución en coordenada X	67
Figura 38 - Relación entre energía y el valor de resolución en coordenada Y	68
Figura 39 - Histograma diferencias coordenadas X (red convolucional), $1 \leq \log \text{Energy} < 2$..	71
Figura 40 - Histograma diferencias coordenadas Y (red convolucional), $1 \leq \log \text{Energy} < 2$..	71
Figura 41 - Histograma diferencias coordenadas X (red convolucional), $2 \leq \log \text{Energy} < 3$..	72
Figura 42 - Histograma diferencias coordenadas Y (red convolucional), $2 \leq \log \text{Energy} < 3$..	72
Figura 43 - Histograma diferencias coordenadas X (red convolucional), $3 \leq \log \text{Energy} < 4$..	73
Figura 44 - Histograma diferencias coordenadas Y (red convolucional), $3 \leq \log \text{Energy} < 4$..	73
Figura 45 - Histograma diferencias coordenadas X (red convolucional), $4 \leq \log \text{Energy} < 5$..	74
Figura 46 - Histograma diferencias coordenadas Y (red convolucional), $4 \leq \log \text{Energy} < 5$..	74
Figura 47 - Histograma diferencias coordenadas X (valores HAWC), $1 \leq \log \text{Energy} < 2$	75
Figura 48 - Histograma diferencias coordenadas Y (valores HAWC), $1 \leq \log \text{Energy} < 2$	75
Figura 49 - Histograma diferencias coordenadas X (valores HAWC), $2 \leq \log \text{Energy} < 3$	76

Figura 50 - Histograma diferencias coordenadas Y (valores HAWC), $2 \leq \log \text{Energy} < 3$	76
Figura 51 - Histograma diferencias coordenadas X (valores HAWC), $3 \leq \log \text{Energy} < 4$	77
Figura 52 - Histograma diferencias coordenadas Y (valores HAWC), $3 \leq \log \text{Energy} < 4$	77
Figura 53 - Histograma diferencias coordenadas X (valores HAWC), $4 \leq \log \text{Energy} < 5$	78
Figura 54 - Histograma diferencias coordenadas Y (valores HAWC), $4 \leq \log \text{Energy} < 5$	78

Índice de tablas

Tabla 1 - Matriz de confusión	28
Tabla 2 - Variables seleccionadas	42
Tabla 3 - Valores de entrada para predicciones	45
Tabla 4 - Tecnologías necesarias	52
Tabla 5 - Campos archivo salida	54
Tabla 6 - Comparativa resultados sigma (HAWC – Red convolucional)	65
Tabla 7 - Comparativa media diferencias (HAWC – Red convolucional)	66

1. Introducción

1.1. Motivación

Un rayo cósmico es una partícula de alta energía procedente del espacio exterior a velocidades cercanas a la de la luz e impacta contra la atmósfera de la Tierra.

Los rayos cósmicos se originan en distintas fuentes del universo, como supernovas, agujeros negros, o núcleos activos de galaxias. Al llegar a la tierra, estas partículas chocan con los átomos presentes en el ambiente creando una cascada de partículas secundarias que pueden ser detectadas en observatorios especializados.

Estas cascadas, tienen efectivamente un punto de origen, siendo el lugar donde la partícula del rayo cósmico colisiona inicialmente con un núcleo atómico presente en la atmosfera, dando lugar a la cascada de reacciones secundarias.

A partir de este punto de colisión, la energía del rayo cósmico se dispersa y produce un gran número de partículas secundarias, como mesones, electrones, fotones, y neutrinos. Estas partículas secundarias continúan interactuando con la atmósfera, creando más partículas y extendiendo la cascada.

La forma y extensión de estas cascadas dependen de varios factores, como la energía del rayo cósmico primario y la altitud a la que ocurre la primera colisión. En general, las cascadas se expanden tanto en dirección lateral como hacia abajo, siguiendo aproximadamente un cono. Por eso, aunque hay un punto de origen, la cascada puede cubrir un área extensa al llegar al observatorio HAWC.

Estos rayos son de gran interés para los científicos porque proporcionan información sobre procesos físicos de alta energía que ocurren en el universo. Más en concreto, la medición del espectro de los rayos cósmicos es crucial para entender las fuentes y la propagación de estas partículas en el espacio.

HAWC (observatorio a gran altura Cherenkov en agua por sus siglas en inglés), es un centro de investigación ubicado en una de las laderas del volcán Sierra Negra, en el estado de Puebla, México, a una altura aproximada de 4100 metros. Este observatorio tiene la capacidad de detectar estas cascadas atmosféricas que atraviesan el observatorio gracias a los 300 detectores Cherenkov en agua.

Una parte importante de las cascadas no son detectadas plenamente por el observatorio debido a la limitación del área que cubre. Esto provoca que, en muchos casos, no se disponga de la información del "núcleo" de dichas cascadas ni de los datos asociados que se generan fuera de la región observada. Como consecuencia, se pierde información valiosa de los eventos, lo que puede afectar a la precisión en la reconstrucción de parámetros críticos, como la energía y la dirección de las partículas incidentes. Ampliar el área de cobertura o mejorar los algoritmos de estimación podría ayudar a mitigar este problema y aumentar la eficiencia del observatorio.

Actualmente la reconstrucción de las cascadas atmosféricas extendidas se está realizando con algoritmos creados y optimizados para rayos gamma ya que es la línea principal de investigación para conocer el origen de este tipo de partículas, de manera que su uso no es el adecuado para los rayos cósmicos.

Con este trabajo se pretende crear un modelo entrenado completamente adaptado a las características propias de los rayos cósmicos para así poder estudiar su energía y origen en más profundidad.

1.2.Planteamiento del trabajo

Actualmente HAWC es un observatorio optimizado para los rayos gamma, por lo que los algoritmos que dispone no están optimizados para el estudio con los rayos cósmicos ya que los actuales introducen cierto error en el análisis.

Se pretende utilizar la inteligencia artificial para reconstruir las cascadas atmosféricas extendidas generadas por los rayos cósmicos, tanto en los casos en los que esta sea observable en su totalidad como en los que no, siempre y cuando la información obtenida sea suficiente para realizar una predicción mediante modelos previamente entrenados.

1.3.Estructura de la memoria

En el capítulo de Contexto y Estado del Arte se va a explicar la parte teórica de la astronomía de altas energías y de la inteligencia artificial, primero con una introducción descriptiva del contexto para luego entrar al detalle sobre aquellos conceptos importantes con los que se va a desarrollar el trabajo. Es muy importante conocer al detalle el fenómeno de las cascadas atmosféricas extendidas generadas a partir de los rayos cósmicos, así como entender el proceso de obtención de toda la información del observatorio HAWC.

Tras entender esto y revisar las opciones de la inteligencia artificial, se determinará y justificará el algoritmo más adecuado para solucionar el problema.

En el capítulo de Objetivo y Metodología se detalla el objetivo general del proyecto en el cual está trabajando el observatorio HAWC, así como el objetivo específico con el que este trabajo persigue contribuir. En este apartado también se describe la metodología usada para dar a entender como se ha abordado el problema y las particularidades de como se ha llevado a cabo.

En el apartado Identificación de requisitos se describe cual es el problema para solucionar y su contexto con el fin de extraer cuales son los requisitos necesarios para llevar a cabo el desarrollo de la herramienta.

A continuación, se describe la herramienta, primero con el detalle cronológico del proceso de desarrollo y después con todos los detalles técnicos tales como los requisitos de ejecución, arquitectura de la solución de aprendizaje automático, diagrama de secuencia y el archivo de salida.

En el apartado de evaluación se describen y a continuación se evalúan los resultados obtenidos al realizar las predicciones mediante el uso de la herramienta desarrollada. Concretamente se mostrarán las gráficas de aprendizaje, histogramas con la comparativa entre valores reales y predichos entre otro tipo de graficas donde se podrá validar la precisión de la herramienta a la hora de predecir el núcleo de cada evento.

2. Contexto y estado del arte

2.1. Astronomía de altas energías

2.1.1. Rayos cósmicos

Un rayo cósmico es una partícula subatómica altamente energética procedente del espacio exterior que viaja a velocidades cercanas a la de la luz y se origina en distintas fuentes del universo, como supernovas o núcleos activos de galaxias.

En este trabajo vamos a diferenciar entre los rayos cósmicos primarios (RCP) y las partículas secundarias.

Los rayos cósmicos primarios son las partículas que viajan a través del espacio exterior mientras que las partículas secundarias son las que se generan como resultado de las interacciones de los RCP con las partículas presentes en la atmosfera terrestre.

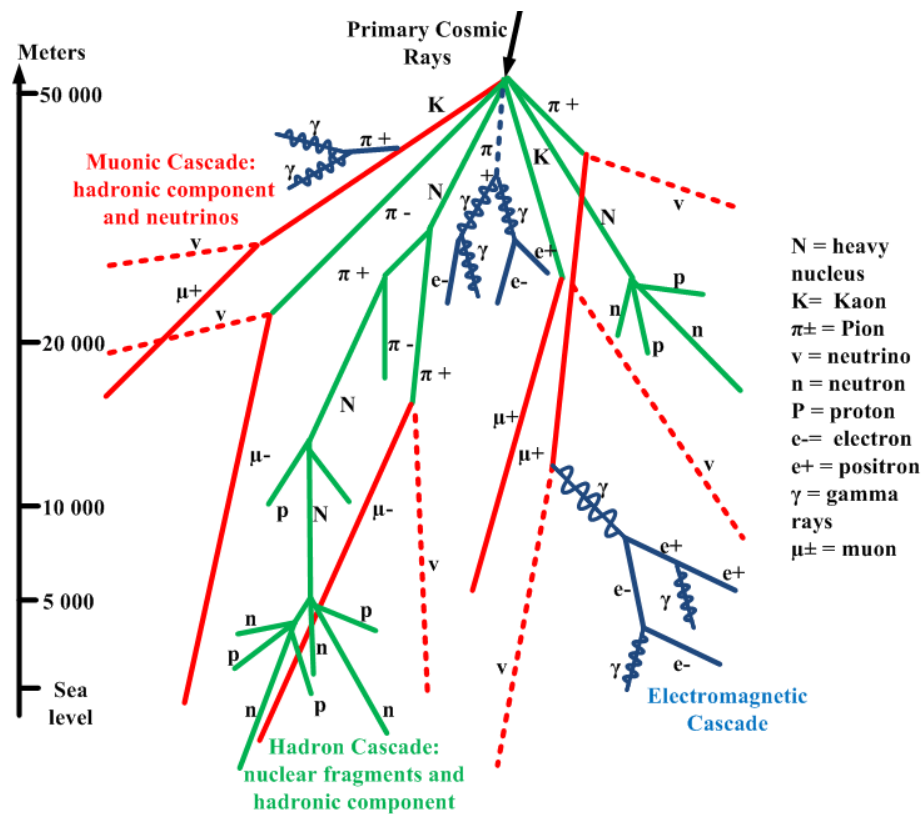
Se ha demostrado en los RCP que pueden estar compuestos por casi cualquier tipo de núcleo, aunque cabe destacar que generalmente (89%) están compuestos por núcleos de hidrogeno, al ser el elemento más común en el universo. Otra partícula común (10%) es el núcleo del helio, la cual puede está compuesta por dos protones y un neutrón (He-3) o dos (He-4). El 1% restante de los RCP lo componen núcleos más pesados como el hierro o el silicio entre otros. (CERN, 2024)

2.1.2. Cascadas atmosféricas extendidas

Como se ha comentado en párrafos anteriores, los rayos cósmicos, al llegar a la tierra, chocan con los átomos presentes en la atmósfera originando una cascada de reacciones que resultan en la producción de numerosas partículas secundarias, siendo los piones, muones, electrones, y fotones las principales, en la figura 1 se presenta una ilustración que recrea este fenómeno a modo de ejemplo.

Esta cascada de partículas secundarias se propaga a través de la atmósfera en forma de un cono expansivo que se mueve casi a la velocidad de la luz.

Figura 1 Cascada atmosférica extendida



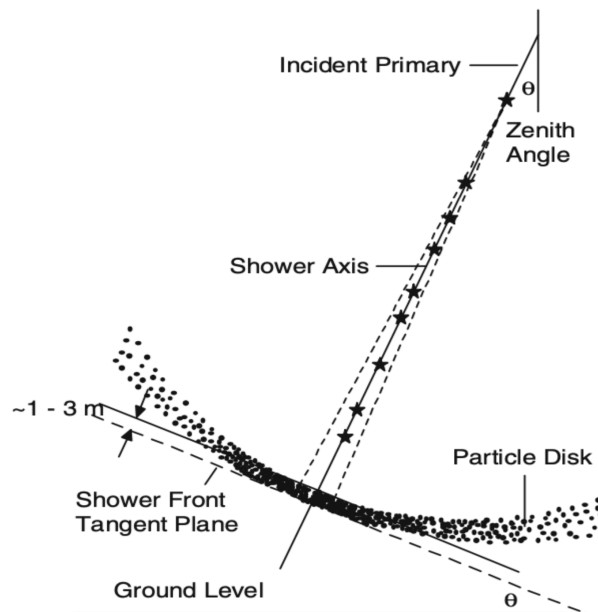
Fuente: Gomez Toro, Daniel. (2014). Temporal Filtering with Soft Error Detection and Correction Technique for Radiation

Hardening Based on a C-element and BICS.

Desde el punto exacto o región de la atmosfera donde se inicia la cascada y en dirección hacia donde se va produciendo esta, se traza un eje de manera que, el núcleo se podría considerar como el punto exacto donde este eje incide en el plano de observación. La forma de detectarlo en el conjunto de datos obtenidos por el observatorio es determinando por el punto donde se junta la mayor cantidad de partículas.

El observatorio detecta estas cascadas atmosféricas extendidas y conforme a los datos obtenidos reconstruye las características más relevantes.

Figura 2 - Plano y eje de la cascada



Fuente: Grieder, P. K. F. (2010). Extensive Air Showers: High Energy Phenomena and Astrophysical Aspects - A Tutorial, Reference Manual and Data Book.

2.1.1. Efecto Cherenkov

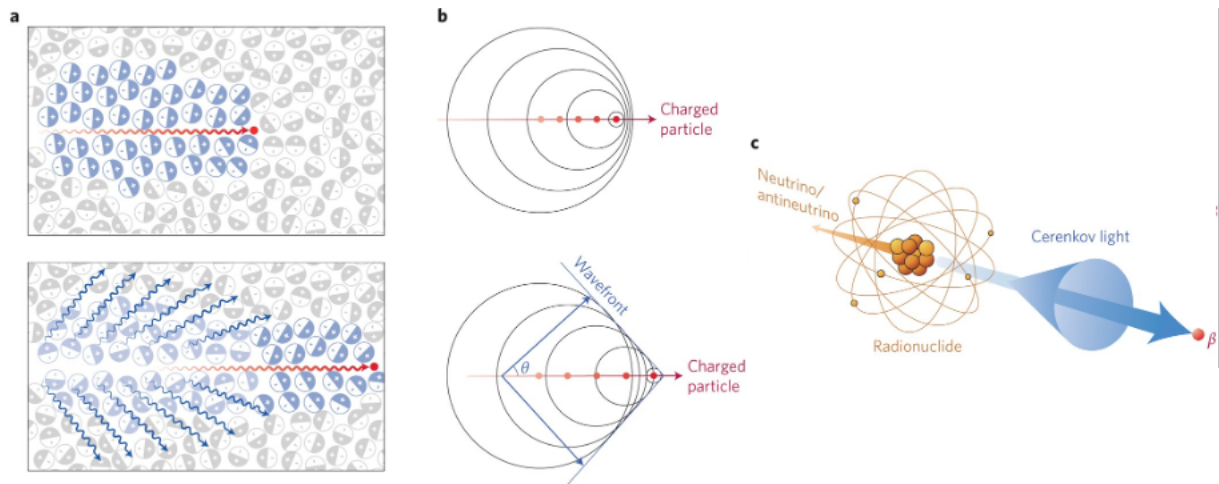
Pavel Cherenkov, descubrió en 1934 que cuando una partícula cargada viaja por un medio dieléctrico a una velocidad superior a la de la luz en ese mismo medio, se produce un efecto que genera una radiación, siendo esta un cono de luz respecto a la dirección de movimiento de la partícula cargada.

El ángulo de apertura (θ) del cono se puede obtener mediante la siguiente formula:

$$\cos(\theta) = \frac{1}{n\beta}$$

siendo n el índice de refracción en el medio, $\beta = v/c$, donde v es la velocidad de la partícula y c la velocidad de la luz en el vacío.

Figura 3 - Efecto Cherenkov



Fuente: Shaffer, T., Pratt, E. & Grimm, J. Utilizing the power of Cherenkov light with nanotechnology. *Nature Nanotech* 12, 106–117 (2017). <https://doi.org/10.1038/nnano.2016.301>

En la figura 3, podemos observar en el apartado a, el efecto que produce la partícula en el medio dieléctrico dando explicación a la importancia de éste para que el efecto tenga lugar. En el apartado b se aprecian las ondas esféricas electromagnéticas con el paso de la partícula siendo estas ondas las que conforman la forma del cono de luz tal. Finalmente, en el apartado c se aprecia como la partícula avanza (dirección flecha naranja) y va generando el cono de luz Cherenkov.

2.1.2. Observatorio HAWC

2.1.2.1. Descripción

El observatorio a gran altura Cherenkov en agua, o HAWC por sus siglas en inglés (High Altitude Water Cherenkov), es un observatorio ubicado en una de las laderas del volcán Sierra Negra, en Puebla, México, a una altura aproximada de 4100 m sobre el nivel del mar.

HAWC está especializado en la detección de rayos gamma, aunque también tiene la capacidad de detectar rayos cósmicos, todo esto gracias a la técnica que utiliza: Cherenkov en agua. Está compuesto por 300 detectores Cherenkov en agua, dispuestos a una distancia muy corta entre sí en forma de parrilla ocupando un área aproximada de 22.000 m².

Gracias a esto, es capaz de detectar las partículas secundarias de la parte de la cascada atmosférica extendida que atraviesa el área que comprende la parrilla de detectores. Una vez recogida toda la información, se procesa con los algoritmos que dispone para así estudiar en profundidad el fenómeno.

Figura 4 - Vista aérea observatorio HAWC



Fuente: High Altitude Water Cherenkov Gamma Ray Observatory. <https://www.hawc-observatory.org>

2.1.2.2. Water Cherenkov Detector

Los detectores Cherenkov en agua, de ahora en adelante WCD (Water Cherenkov Detector), del observatorio tienen el objetivo de recrear las condiciones óptimas para que el efecto Cherenkov se produzca y así detectarlo con los sensores que este contiene en el interior.

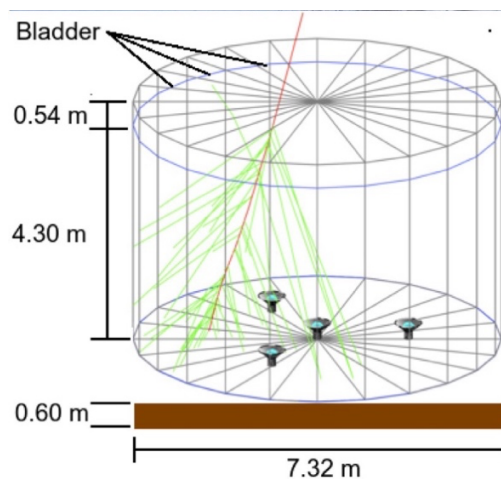
El recipiente externo es una estructura de acero en forma de cilindro con un diámetro de 7,3m y una altura de 4,5m. Esta estructura tiene el objetivo de contener 180.000 litros de agua ultrapura y el de proteger a esta del exterior, particularmente de la luz.

En la parte superior se ubica una lona flexible compuesta de polietileno de baja densidad, con un grosor de unos 0,4mm; esta constituye una barrera ligera que junto a la estructura metálica permite al agua estar en completa oscuridad.

El agua se extrae de un pozo del valle que está a unos 2640 metros sobre el nivel del mar. Antes de descalcificarla, pasa por unos pre-filtros de 30 micras de manera que se consigue reducir la dureza del agua. Una vez el agua es transportada al observatorio, se vuelve a filtrar a $15\mu\text{m}$.

Se utiliza una bolsa plástica para contener 180,000 litros de agua, en cuyo fondo se instalan cuatro tubos fotomultiplicadores (PMTs). Uno de ellos se coloca en el centro, mientras que los otros tres se disponen alrededor formando un triángulo. Cuando la partícula atraviesa el agua produce el efecto Cherenkov y este es detectado por los PMTs.

Figura 5 - Water Cherenkov Detector



Fuente: The High-Altitude Water Cherenkov (HAWC) observatory in México: The primary detector.

arXiv:2304.00730v2 [astro-ph.HE]

2.1.2.3. PMTs (sensor)

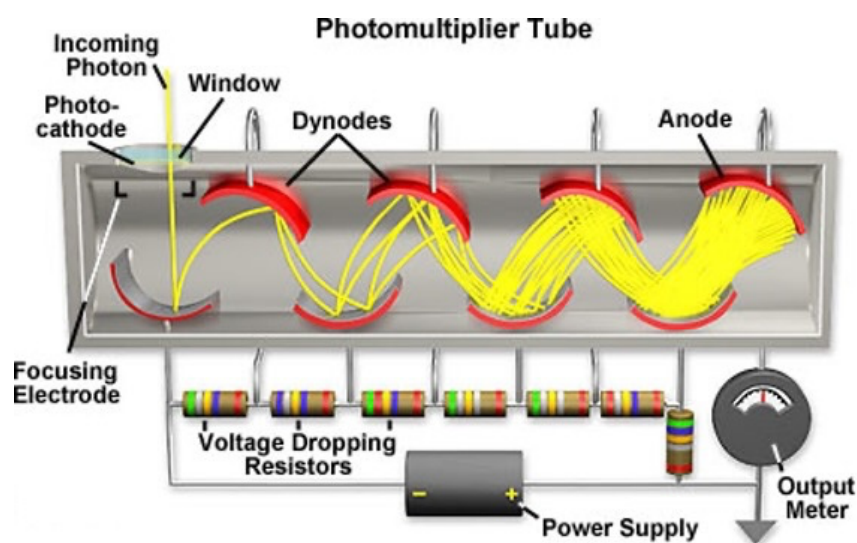
Los tubos fotomultiplicadores o PMTs por sus siglas en inglés, son los sensores de radiación Cherenkov.

Como se puede observar en la figura 5, cada WCD contiene 4 PMTs mirando hacia arriba que se ubican a unos 4 metros de profundidad del agua. Dejar esta distancia de agua por encima del sensor mejora la sensibilidad para detectar la radiación Cherenkov emitida por las partículas secundarias. Sin embargo, si la distancia de agua por encima de los PMT fuera muy superior, la sensibilidad a la cantidad de fotoelectrones por energía bajaría considerablemente.

Esto se debe a que la luz Cherenkov debe atravesar el agua para llegar a los sensores siendo más tenue conforme la partícula cargada recorre más distancia, de manera que la cantidad de agua puede afectar la cantidad de luz que llega al sensor. (Spiering, 2023)

Un tubo fotomultiplicador funciona convirtiendo la luz incidente en una señal eléctrica amplificada a través de una serie de etapas de multiplicación de electrones en una cadena dinámica.

Figura 6 - Tubo fotomultiplicador (PMT)



Fuente: Hamamatsu. Concepts in Digital Imaging Technology.

<https://hamamatsu.magnet.fsu.edu/articles/photomultipliers.html>

Cuando los fotones de la radiación Cherenkov inciden en el fotocátodo del PMT, este libera un electro por el efecto fotoeléctrico (Einstein, 1905) que recorre una serie de dinodos aumentando la cantidad de electrones a su paso debido a que cada dinodo está a un voltaje ligeramente superior al anterior.

Finalmente, llega al ánodo una cantidad suficiente de electrones generando así una señal eléctrica que pueda ser medida, siendo proporcional a la cantidad de luz que llegó al fotocátodo.

2.1.2.4. Problema actual

El estudio del espectro de rayos cósmicos es fundamental para entender las fuentes y la propagación de estas partículas por el espacio.

Tal y como se indica en la motivación de este trabajo, HAWC está diseñado y optimizado para trabajar con rayos gamma pero también ha demostrado ser eficaz en el estudio de rayos cósmicos.

Sin embargo, los algoritmos actuales utilizados no están optimizados para el estudio de los rayos cósmicos. Aunque estos algoritmos han demostrado ser eficaces para el análisis de los rayos gamma, su rendimiento no es óptimo cuando se aplican a los datos de los rayos cósmicos.

Este trabajo tiene como objetivo abordar esta limitación mediante el desarrollo de algoritmos de aprendizaje automático (ML) optimizados específicamente para el estudio de los rayos cósmicos. Estos algoritmos estarán diseñados para extraer características útiles de los datos de los rayos cósmicos y utilizar estas características para la reconstrucción de las cascadas atmosféricas extendidas.

2.2. Aprendizaje automático

2.2.1. Descripción

El aprendizaje automático es el área de la inteligencia artificial que trata de aprender en base a la experiencia. Para ello se emplean algoritmos que procesan grandes cantidades de datos dando lugar a un modelo, siendo este el que representa lo aprendido al ajustar los parámetros internos. Posteriormente, este modelo permite procesar otro conjunto de datos no conocido y generar un resultado aproximado respecto a los patrones encontrados en los datos para así hacer predicciones, clasificar o agrupar datos en segmentos significativos.

Los algoritmos de aprendizaje automático se pueden dividir en los siguientes tipos: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semi-supervisado, aprendizaje por refuerzo y *Deep Learning*. (Fumo, 2017)

Otra forma de clasificar los algoritmos de aprendizaje automático es en base a la forma de realizar el entrenamiento: *online* u *offline*. En el entrenamiento offline, el modelo aprende de un set de datos obtenidos anteriormente y la base de entrenamiento es más lenta y costosa computacionalmente. Una vez que se ha finalizado el entrenamiento, el modelo deja de aprender. En el aprendizaje *online*, el modelo se entrena usando pequeños sets de datos, cada fase de entrenamiento es rápida y con bajo coste computacional. El modelo mejora su rendimiento constantemente. (Tusar, 2022)

2.2.2. Aprendizaje supervisado

El aprendizaje supervisado tiene como objetivo hacer predicciones sobre datos desconocidos o futuros, utilizando para ello ejemplos observados previamente, es decir, toda la información que se usa para entrenar el modelo esta etiquetada.

Uno de los objetivos principales del aprendizaje automático supervisado es la capacidad de generalizar, que significa tener un buen rendimiento con datos nunca vistos y no solo con los datos de entrenamiento. Dentro del aprendizaje supervisado existen dos tipos de algoritmos: regresión y clasificación. (Wang, Cui, & Ke, 2023)

2.2.2.1. Algoritmos de regresión

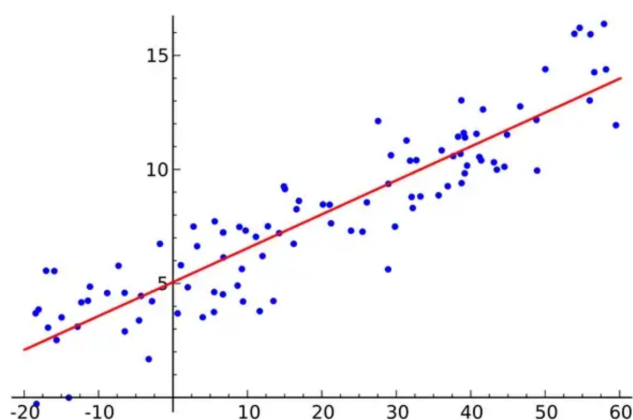
Los algoritmos de regresión que predicen un valor continuo para una variable, por ejemplo: predecir ingresos en base a una edad, predecir el precio de la vivienda o estimar proyecciones de venta.

La regresión lineal es uno de los algoritmos más comunes dentro del aprendizaje automático y se usa para encontrar una relación lineal entre la variable dependiente y las variables independientes.

Existe la regresión lineal simple donde únicamente hay una variable independiente y la regresión lineal múltiple donde puede haber múltiples variables independientes.ⁱ La regresión polinomial establece una relación polinomial de n grado entre las variables dependientes e independientes y se aplica cuando la relación entre la variable dependiente y las independientes no tienen un carácter lineal.

Una forma de visualizar el error de un algoritmo de regresión es a través de un gráfico donde se dibuje la función de predicción y los valores reales, de esta forma podemos ver si los valores reales siguen la tendencia o hay que buscar un algoritmo que se ajuste mejor. La evaluación de los modelos de predicción se hace a través de aplicar métricas de error, las más comunes son: Error cuadrático medio (MSE), error absoluto medio (MAE), raíz cuadrada del error cuadrático medio (RMSE) y logaritmo de la raíz cuadrada del error cuadrático medio (LMLSE). (Yashwanth, 2020)

Figura 7 - Regresión lineal



Fuente: Evaluation of Linear Regression Models. <https://medium.datadriveninvestor.com/evaluation-of-linear-regression-models-257851e77e28>.

2.2.2.1. Algoritmos de clasificación

Los algoritmos de clasificación categorizan los datos de entrada en clases y para un elemento en concreto te dice a qué clase tiene mayor probabilidad de pertenencia. Existen clasificadores binarios, donde solamente hay dos opciones de clase, y los clasificadores multi-clase donde hay más de dos clases donde categorizar.

La evaluación del algoritmo puede hacer basada en la precisión que representa el número de aciertos entre el número de predicciones. La sensibilidad también es una métrica que evalúa la ratio de verdaderos positivos entre el número total de positivos. La especificidad indica la ratio de verdaderos negativos entre el total de negativos.

La precisión es la proporción de verdaderos positivos entre todas las predicciones positivas, es decir, mide la calidad de las predicciones positivas del modelo.

Por otro lado, el recall, también conocido como sensibilidad, evalúa la capacidad del modelo para identificar correctamente todas las instancias positivas reales, es decir, la proporción de verdaderos positivos entre todas las instancias positivas reales.

La matriz de confusión se utiliza para organizar los resultados de las clases reales y las predicciones para obtener las métricas anteriormente descritas y así determinar si son verdaderos positivos, verdaderos negativos, falsos positivos o falsos negativos.

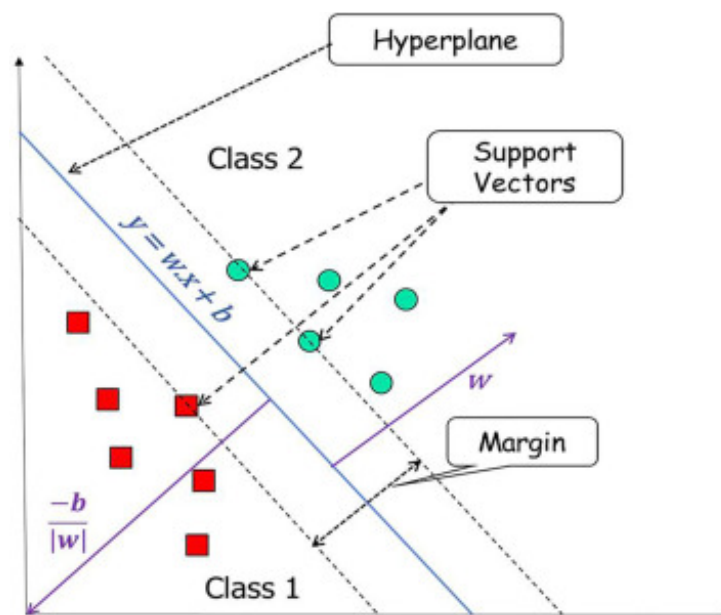
Tabla 1 - Matriz de confusión

Matriz de confusión	Positivo Real	Negativo Real
Predicción Positiva	Verdadero positivo	Falso positivo
Predicción Negativa	Falso negativo	Verdadero negativo

Algunos ejemplos de algoritmos de clasificación:

- **Regresión logística** > Establece una división entre las clases y asigna una probabilidad entre 0 y 1 de pertenecer dicha clase basada en la distancia del punto a la línea de división.
- **Support Vector Machines** > Buscan encontrar un hiperplano en un espacio N-dimensional (N - el número de características) que clasifica claramente los puntos de datos. Las SVM buscan maximizar el margen entre las diferentes clases aprovechando los puntos de datos más cercanos al hiperplano de separación, conocidos como vectores de soporte

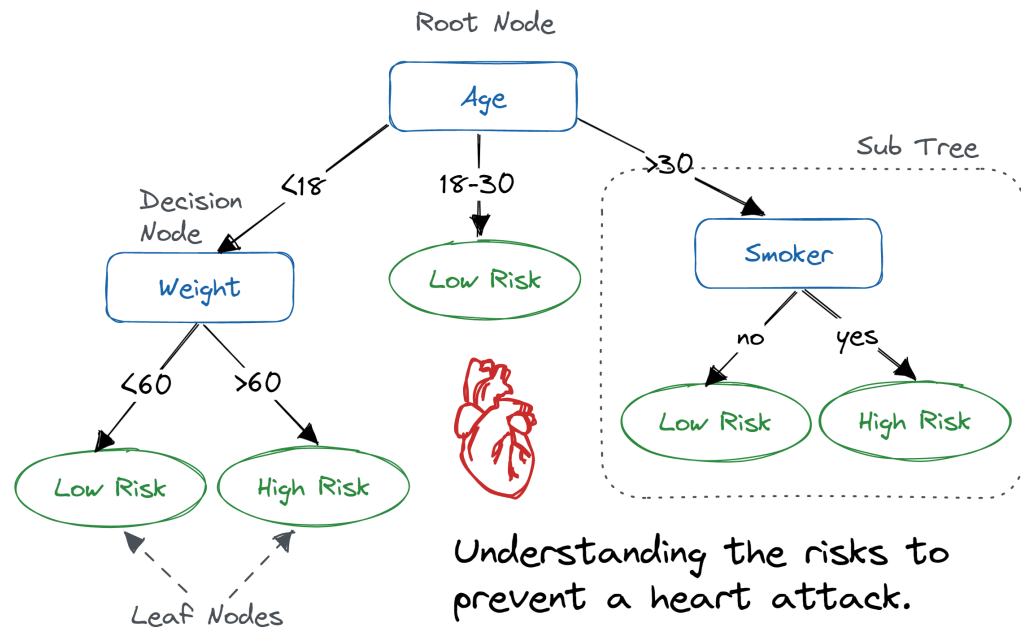
Figura 8 - Ejemplo de clasificación con SVM



Fuente: Buff ML. <https://buffml.com/support-vector-machine/>

- **Árboles de decisión** > Tienen como objetivo simplificar la toma de decisiones. Inicialmente se divide el problema en una serie de preguntas formando árbol donde cada nodo del árbol representa una pregunta o una decisión y cada rama representa una respuesta a esa pregunta. Al final de cada serie (hojas) están las respuestas finales.

Figura 9 - Ejemplo de clasificación con árboles de decisión



Fuente: <https://www.datacamp.com/tutorial/decision-tree-classification-python>

2.2.3. Aprendizaje no supervisado

En el aprendizaje no supervisado, a diferencia del supervisado, el modelo se construye en base al uso de datos que no están etiquetados, es decir, no hay un conocimiento previo de la información sobre la que se va a aprender. El fin es encontrar patrones en los datos a fin de agruparlos o detectar anomalías.

Dentro de cada uno de estos dos problemas se pueden dar múltiples objetivos diferentes, por ejemplo, en los problemas de agrupamiento los objetivos pueden ser desde la segmentación de personas en base a unos criterios a detectar regiones en una imagen mediante la similitud de píxeles. Por otro lado, en el caso de los problemas de detección de anomalías, un ejemplo muy claro sería el de detección fraude en un conjunto de compras.

2.2.3.1. Algoritmo K-Means

El objetivo del algoritmo es el de detectar patrones en los datos y agruparlos basándose en su similitud. El funcionamiento es el siguiente:

1. Determinar la cantidad de clústeres (K)
2. Seleccionar K puntos aleatorios como centroides iniciales
3. Se asigna cada punto al clúster cuyo centroide esté más cerca.
4. Se recalculan los centroides una vez que todos los puntos han sido asignados.
5. Se repiten los pasos 3 y 4 hasta que se cumple el criterio de parada, suele ser cuando el centroide no cambia o lo hacen muy poco.

Figura 10 - Agrupación de datos con K-Means



Fuente: BIO Statistics collaboration of Australia

https://bookdown.org/tpinto_home/Unsupervised-learning/k-means-clustering.html

2.2.4. Redes neuronales artificiales

2.2.4.1. Descripción

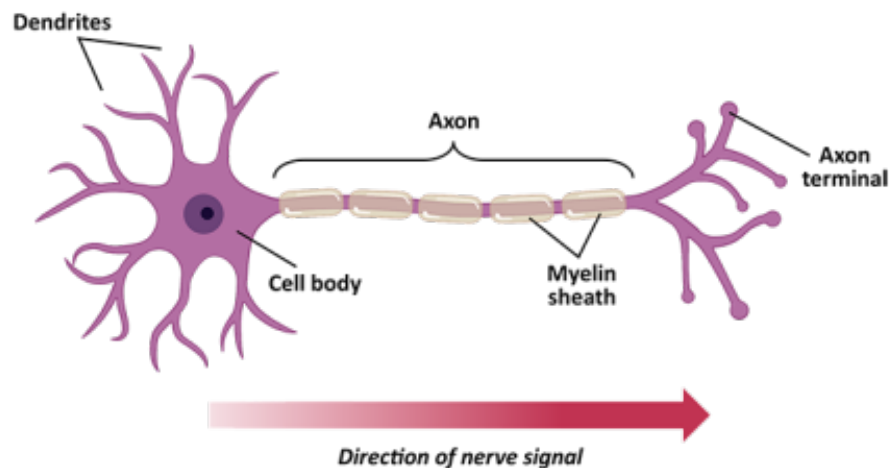
El Deep Learning es el área del aprendizaje automático que basa su aprendizaje en el uso de redes neuronales con al menos tres capas y grandes cantidades de datos para su entrenamiento. Este tipo de aprendizaje trabaja con datos no estructurados como textos e imágenes pudiendo extraer características de estos.

Las redes neuronales artificiales están compuestas por una serie de neuronas artificiales organizadas en capas y conectadas entre sí.

2.2.4.2. Neurona biológica

Las neuronas artificiales están inspiradas en las neuronas biológicas habiendo similitudes entre su estructura y funcionamiento. Como se puede observar en la figura 11, las neuronas biológicas están formadas por unas dendritas, el núcleo de la célula, un axón y las terminales de este.

Figura 11 - Morfología de la neurona biológica



Fuente: National Institute of Neurological Disorders and Stroke. <https://www.ninds.nih.gov/health-information/public-education/brain-basics/brain-basics-know-your-brain>

La función de las dendritas es la de recibir las señales en forma de impulsos eléctricos procedentes de otras neuronas conectadas, al llegar esta señal los canales iónicos se abren. Esto permite que los iones positivos entren en la neurona, lo que disminuye la diferencia de voltaje entre el interior y el exterior de la neurona, un proceso conocido como despolarización. Si esta despolarización alcanza un cierto umbral, se inicia un potencial de acción, que es una señal eléctrica que viaja a lo largo del axón de la neurona.

Esta señal eléctrica generada en determinadas condiciones se envía a través de los terminales del axón con el objetivo de que llegue a las siguientes neuronas. (National Institutes of Neurological Disorders and Stroke, s.f.)

2.2.4.3. Neurona artificial

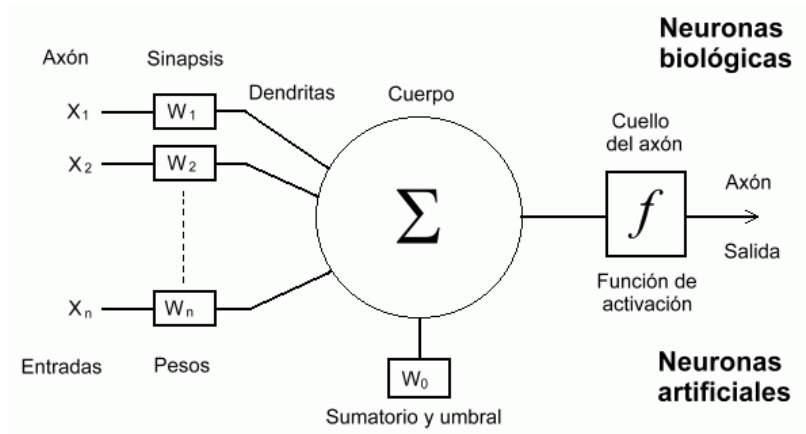
Una neurona artificial está compuesta por múltiples entradas (x_i) ponderadas por pesos (w_i), una función de suma y una función de activación. La función de suma calcula la suma ponderada de las entradas y los pesos, lo que puede representarse como:

$$Z = W_1X_1 + W_2X_2 + \dots + W_nX_n$$

Donde X son las entradas y W los pesos asociados a cada entrada como se puede observar en la figura 12.

Existen diferentes técnicas de inicialización de los pesos al empezar el entrenamiento de una red, pero lo más habitual es que se asignen valores aleatorios a todos los pesos con el fin de romper la simetría y permitir que las neuronas aprendan características diferentes. Durante el entrenamiento los pesos se van ajustando con técnicas de optimización como la del descenso de gradiente al comprobar el error que hay entre las predicciones del modelo y los valores reales. (Durán, 2019)

Figura 12 - Comparación neurona biológica / artificial



Fuente: Departamento de Ciencias de la Computación e Inteligencia Artificial. Universidad de Sevilla

https://www.cs.us.es/~fsancho/Blog/posts/Redes_Neuronales.md

Al aplicar la función de activación $f(z)$ a la suma ponderada z se obtiene la salida de la neurona. La función de activación nos permite evitar la linealidad en la salida de la neurona y así permitir que la red pueda aprender y modelar relaciones y patrones complejos en los datos.

2.2.4.4. Función de activación

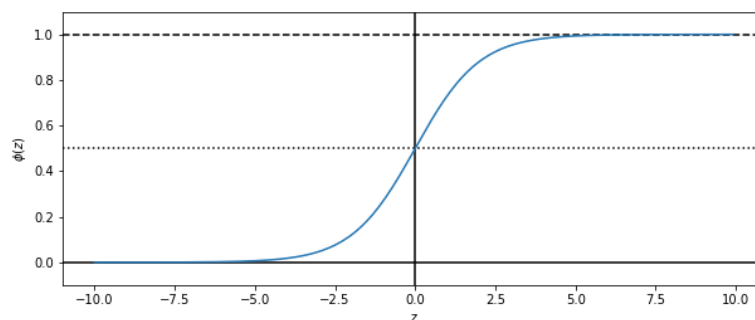
Como se indica en el apartado anterior, la función de activación es fundamental dado que aporta la no linealidad en la salida de cada neurona siendo algo imprescindible para que la red neuronal pueda aprender.

Una vez se obtiene la suma ponderada de las entradas y los pesos, el resultado pasa por la función de activación donde se determina y en qué medida si se activa la neurona. Dependiendo de las características del problema tratado se puede optar por un tipo de función u otra. Algunas de las funciones más comunes son:

Función sigmoide

Como se puede apreciar en la figura 13, la salida será próxima a cero cuando z tiende a menos infinito y próxima a 1 cuando tienda a infinito. Debido a la naturaleza asintótica de la función, nunca tomará los valores 0 o 1.

Figura 13 - Función sigmoide



Fuente: http://personal.cimat.mx:8181/~mriviera/cursos/aprendizaje_maquina/regresion/regresion.html

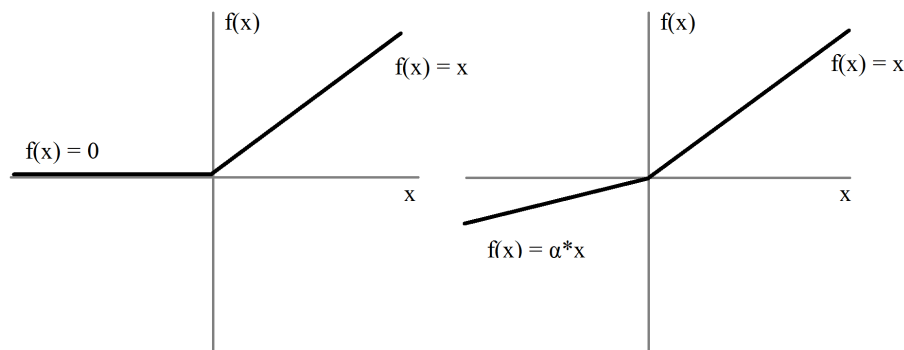
Sin embargo, en los valores de Z cercanos a 0, una pequeña variación implica un cambio importante en el valor de salida, haciendo esta característica muy interesante en un contexto de clasificación ya que un cambio menor en la entrada produce una diferencia discernible en la salida proporcionando la probabilidad de pertenencia a una clase u otra.

Los dos principales problemas de esta función son la desaparición del gradiente en redes neuronales profundas y que no está centrada en el cero, algo que puede provocar la desaparición del gradiente en las capas posteriores. (Wang C.-F. , 2019)

Función RELU

La función ReLU, retorna x si este valor es positivo y cero en el resto de los casos. Dado que no realiza ningún cálculo es muy eficiente computacionalmente siendo algo muy positivo en escenarios de aprendizaje profundo donde se trabaja con grandes cantidades de datos.

Figura 14 - Funciones ReLU y Leaky ReLU



Fuente: <https://blog.paperspace.com/vanishing-gradients-activation-function/>

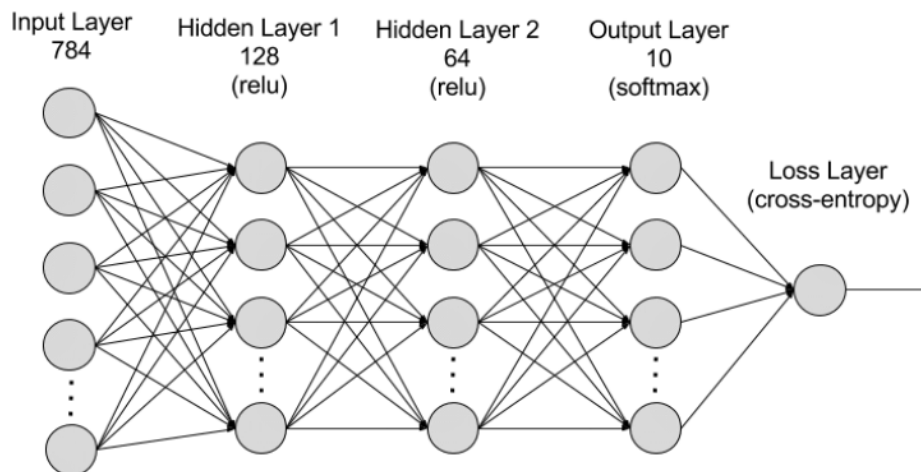
El problema mencionado en la función sigmoide se soluciona con ReLU y Leaky ReLU ya que no satura en los valores positivos. (Kathuria, 2018)

2.2.4.5. Arquitectura red neuronal

Una red neuronal es, como su propio nombre indica, una red de neuronas artificiales interconectadas entre diferentes capas siendo capaz de detectar patrones y características de un conjunto de datos de manera que, una vez entrenada, tiene la capacidad de aplicar lo aprendido sobre unos datos de entrada no conocidos.

La arquitectura es la estructura y organización de cómo está construida, desde el número de neuronas artificiales, tipos o cantidad de capas, hasta la forma de cómo se propagan los datos entre ellas. En la figura 15 podemos observar el ejemplo de una red neuronal con la capa de entrada, dos capas ocultas donde las neuronas tienen configurada la función de activación ReLU, una capa de salida tipo softmax y una última capa con una sola neurona.

Figura 15 - Red neuronal 'fully connected'

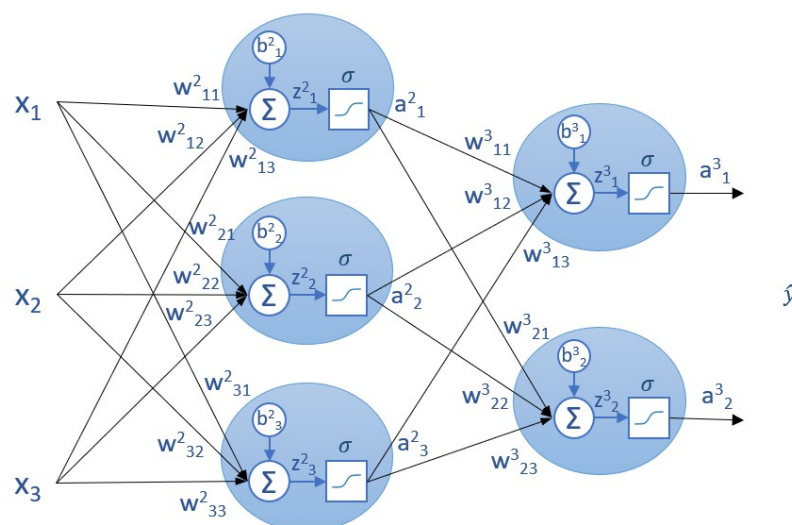


Fuente: <https://aws.amazon.com/es/what-is/neural-network/>

La capa de entrada es la primera capa de neuronas que reciben los datos, los procesa inicialmente y los propaga a las siguientes capas.

Las capas ocultas de una red neuronal son capas intermedias situadas entre la capa de entrada y la capa de salida. Su función principal es detectar y aprender características complejas a partir de los datos de entrada. Cada neurona en una capa oculta recibe señales de las neuronas de la capa anterior, aplica una función de activación (como las indicadas en el apartado 2.2.4.4) para introducir no linealidades, y luego pasa el valor a la siguiente capa.

Figura 16 - Funcionamiento interno entre capas



Fuente: <https://aws.amazon.com/es/what-is/neural-network/>

La capa de salida es la que representa la predicción en base al entrenamiento realizado en las capas ocultas. Esta capa recibe la información desde la última capa oculta y al aplicar una función de activación adecuada se obtienen los datos de salida.

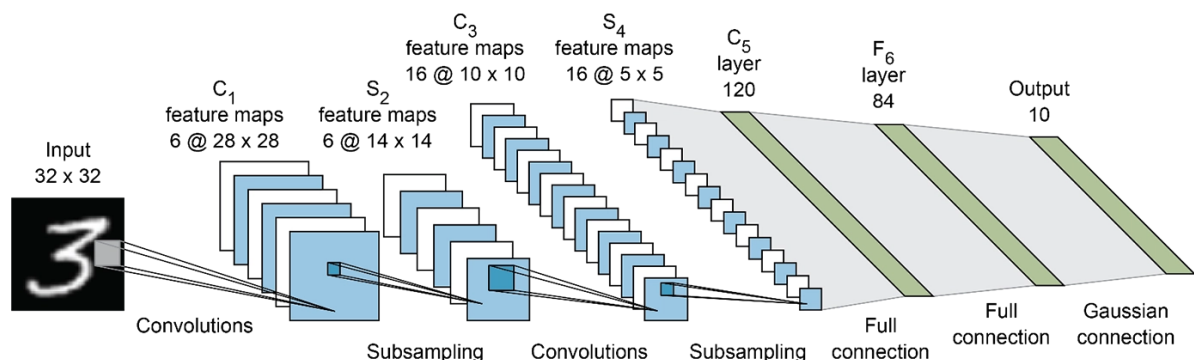
2.2.4.6. Redes neuronales convolucionales

Las redes neuronales convolucionales, en adelante CNN por sus siglas en inglés, son un tipo de red especializada en el aprendizaje donde la estructura de los datos tiene forma de parrilla, siendo muy útiles por ejemplo en el procesamiento de imágenes.

Su arquitectura consiste en una serie de capas convolucionales que aplican filtros (kernels) para detectar características locales, seguidas de capas de pooling que reducen la dimensionalidad. Estas redes también incluyen capas completamente conectadas para la clasificación final.

Como se puede observar en la figura 17, la imagen original donde hay un 3 de forma manuscrita pasa por las diferentes capas convolucionales donde se extraen las diferentes características detectadas en la imagen, como pueden ser las curvas, las texturas, etc.

Figura 17 - Red neuronal convolucional



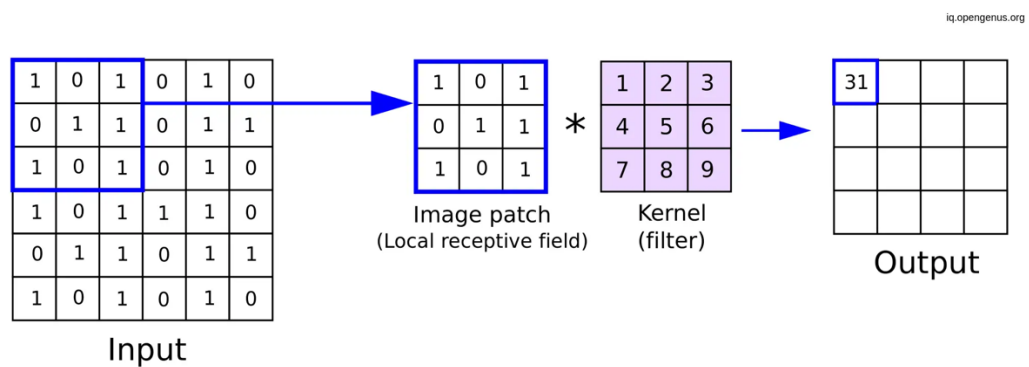
Fuente: <https://www.superannotate.com/blog/guide-to-convolutional-neural-networks>

Finalmente, los datos extraídos en las capas convolucionales pasan a una red neuronal 'fully connected' como la descrita en el apartado 2.2.4.5 donde finalmente obtendremos la salida que represente la predicción.

Capas convolucionales

En cada capa convolucional, la imagen pasa por filtros que detectan características específicas, como bordes o texturas en los casos de imágenes reales. Las imágenes se representan como matrices de valores numéricos, donde cada valor corresponde al color de un píxel. Como se muestra en la figura 18, estos valores son sometidos a cálculos matemáticos mediante los filtros, y los resultados se transfieren a las capas siguientes, permitiendo la extracción progresiva de características más complejas. (Alom, 2019)

Figura 18 – Procesado de información en la capa convolucional



Fuente: <https://www.superannotate.com/blog/guide-to-convolutional-neural-networks>

3. Objetivos y metodología de trabajo

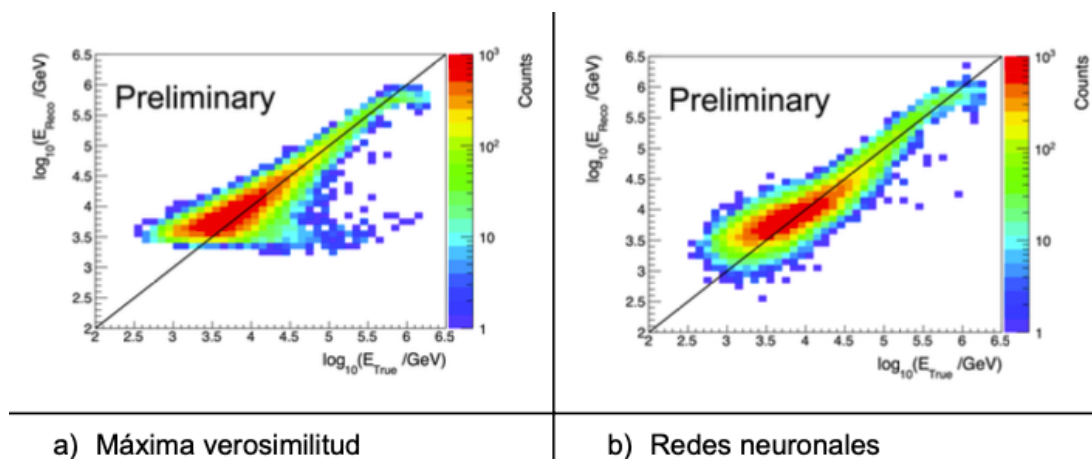
3.1. Objetivo general

3.1.1. Reconstrucción de las cascadas

Para analizar los rayos cósmicos, se utiliza un estimador de energía que aplica la técnica de máxima verosimilitud y el método de unfolding para investigar su composición. No obstante, con la reciente actualización del software en el observatorio HAWC, este estimador ha mostrado varias inconsistencias, lo que sugiere la necesidad de actualizar y mejorar este parámetro (ver Figura 19).

La primera tentativa para mejorar el estimador fue entrenar una arquitectura simple para reconstruir la energía, la cual demostró un rendimiento superior al eliminar las colas presentes en el modelo actual. Esto lleva a la hipótesis central de este proyecto: al emplear un modelo más avanzado donde podamos obtener parámetros de reconstrucción más precisos para los rayos cósmicos.

Figura 19 - Rendimiento estimador de máxima verosimilitud vs redes neuronales



Fuente: Cosmic-ray energy reconstruction using machine learning techniques

(A. Alvarado, 2023)

El modelo óptimo es aquel en el que la mayoría de los eventos se agrupan en torno a la línea de identidad, representada en color negro.

3.2.Objetivos específicos

3.2.1. Predicción de coordenadas del núcleo de la cascada

Como parte crucial en la reconstrucción de las cascadas atmosféricas extendidas, es fundamental predecir con precisión las coordenadas del núcleo de estas cascadas. Mejorar la precisión en la localización del núcleo contribuirá significativamente a la reconstrucción completa y detallada de los eventos de cascadas atmosféricas, aumentando la exactitud y fiabilidad de los análisis realizados.

3.3.Metodología del trabajo

3.3.1. Obtención de datos

Para el propósito de este trabajo, HAWC proporciona una gran cantidad de eventos simulados de rayos cósmicos. Es crucial utilizar datos simulados generados por el software propio del observatorio, ya que los datos reales no nos proporcionarían las posiciones exactas del núcleo de la cascada, una información fundamental para este trabajo.

Estos eventos simulados están almacenados en un clúster de servidores de HAWC en archivos de formato XCDF. Este formato es ideal debido a su alta capacidad de compresión, lo que permite gestionar eficientemente la enorme cantidad de datos almacenados.

3.3.1.1. Fuentes de datos

Disponemos de dos fuentes de datos necesarias para que el algoritmo trabaje, los ya mencionados eventos en formato XCDF y un archivo XML donde se almacena toda la información relativa a los tanques.

Eventos

Hay dos tipos de archivos que contienen la información de los eventos:

- **Archivos REC:** Utilizados para el análisis físico del evento, generados para todos los datos reconstruidos, pero almacenan sólo los datos necesarios para el análisis.
- **Archivos extendidos:** Contienen información detallada sobre la reconstrucción, destinados a comprender el detector y desarrollar herramientas de reconstrucción.

La opción idónea serían los archivos REC, pero para el entrenamiento se ha de usar la versión extendida debido a que las variables a predecir (ver tabla 2) están ahí.

Tanques WCD

HAWC proporciona además un archivo llamado *'survey.xml'* en el cual viene toda la información relacionada con los tanques WCD. A continuación, se detalla la información que contiene propia de un tanque y el primer PMT (channel) de ese tanque:

Figura 20 - Variables tanque WCD

```
1      <tank id="1">
2        <name> E9 </name>
3        <type> 1 </type>
4        <position>
5          <x unit="cm"> 12779.69 </x>
6          <y unit="cm"> 21359.65 </y>
7          <z unit="cm"> 5.63 </z>
8        </position>
9        <channels>
10         <channel id="1">
11           <name> E9A </name>
12           <number> 1 </number>
13           <type> 1 </type>
14           <position>
15             <x unit="cm"> 46.73 </x>
16             <y unit="cm"> 179.07 </y>
17             <z unit="cm"> 46.130000 </z>
18           </position>
19         </channel>
20       </channels>
21     </tank>
```

Fuente: HAWC observatory

Las variables más relevantes son las posiciones de cada tanque y las de cada PMT junto a los identificadores respectivos, esta información es la que nos permite identificar exactamente cuál es el PMT correspondiente a cada HIT del archivo extendido. Un hit es la detección de un PMT cuando la luz Cherenkov pasa por él.

En el archivo del evento viene el identificador del tanque, pero no del PMT, sin embargo, se indica la posición del hit detectado. También es necesario tener en cuenta las coordenadas del tanque, aunque tengamos su identificador, debido a que las coordenadas de cada PMT no son absolutas, es decir, son las coordenadas del PMT respecto del tanque de manera que se

ha de calcular las coordenadas del PMT. Toda esta información necesaria se va a extraer de este archivo XML y se va a almacenar en un CSV para su uso en cada ejecución.

3.3.2. Selección de variables

Para el entrenamiento del modelo, se utilizarán datos de diversas variables seleccionadas, las cuales se ha determinado que son adecuadas para ser usadas en el entrenamiento para posteriormente predecir las coordenadas del núcleo del evento.

Por un lado, están las variables que nos indican la idoneidad del evento para determinar si este ha de ser incluido en el entrenamiento (validación) y por otro lado las propias variables del evento que entrenarán el modelo. Un paso previo al inicio del algoritmo es el de carga de eventos de los archivos XCDF convirtiendo los eventos en objetos, es en este punto donde se van realizando las diferentes validaciones para así crear el objeto o invalidar el evento.

En la tabla 2 se muestran las variables con las que va a trabajar el algoritmo, desde las usadas como validación a las propias del evento que se van a usar en el entrenamiento.

Tabla 2 - Variables seleccionadas

Tipo	Nombre	Descripción	Comentarios
Validación	rec.status	Estado del evento reconstruido	Incluir sólo si es 0
Validación	rec.eventID	Identificador del evento	
Validación	rec.runID	Identificador de la ejecución	
Validación	rec.angleFitStatus	Estado del valor del ángulo	Incluir sólo si es 0
Validación	rec.coreFitStatus	Estado del valor del núcleo	Incluir sólo si es 0
Validación	rec.nHit	Cantidad de hits del evento	Incluir si es mayor a 75
Validación	rec.nChAvail	Canales disponibles por evento	Incluir si $nHit/nChAvail > 0.22$
Calculo posición	event.hit.tankId	Identificador del tanque	
Entrenamiento	rec.zenithAngle	Ángulo cenital del evento	
Entrenamiento	eventHitCharge	Carga del hit	
Entrenamiento	eventHitXPMT	Posición X del PMT	
Entrenamiento	eventHitYPMT	Posición Y del PMT	
Entrenamiento	mc.coreX	Coordenada X del núcleo del evento	Valor que predecir
Entrenamiento	mc.coreY	Coordenada Y del núcleo del evento	Valor que predecir

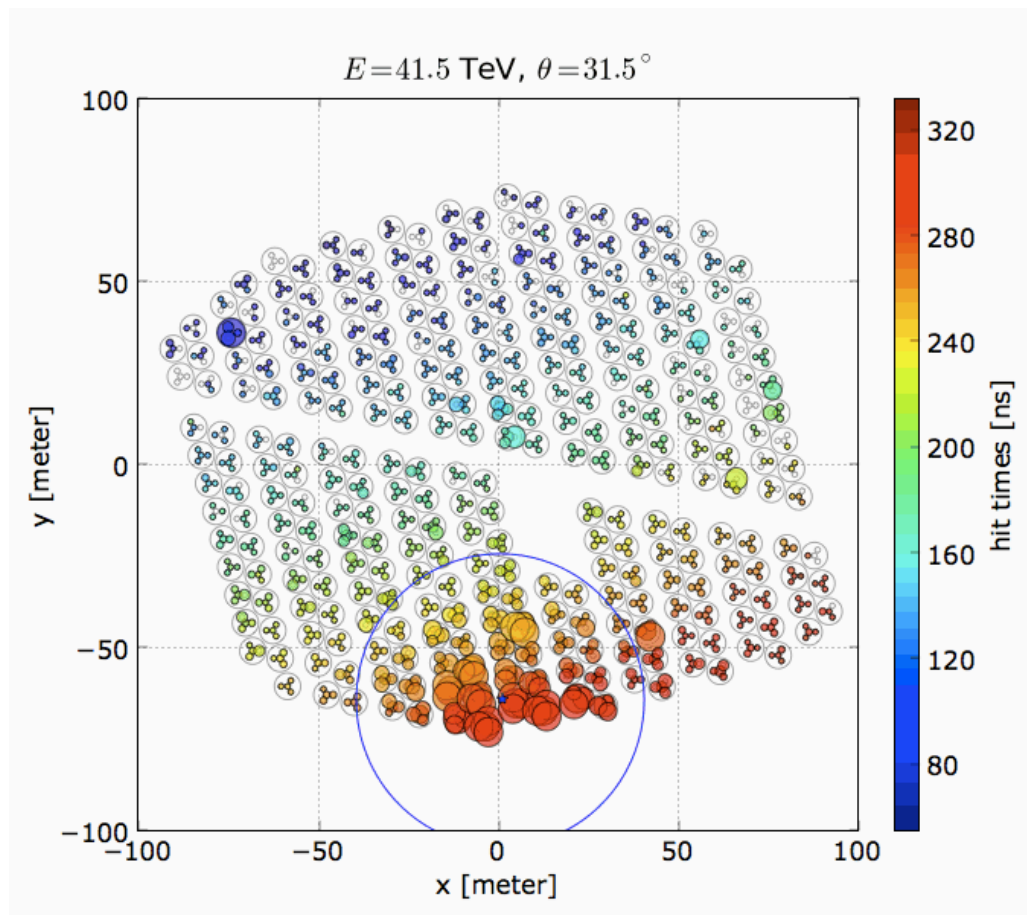
3.3.3. Entrenamiento del modelo

Dado que el objetivo es predecir valores espaciales, específicamente las coordenadas de una posición en un plano bidimensional y la entrada representa una serie de valores en ese mismo plano, se determina que el uso de una red neuronal convolucional es la opción más adecuada.

En la figura 21 podemos observar la simulación de una cascada atmosférica extendida donde los tanques están dispuestos en forma de parrilla y cada uno contiene 4 sensores PMT de manera que la información de los eventos se puede representar de forma similar.

Los tanques estarán representados en sub-matrices de 2x2 dentro de una matriz de 40 x 30 donde cada índice representa un PMT en concreto en el que almacenar la información pertinente.

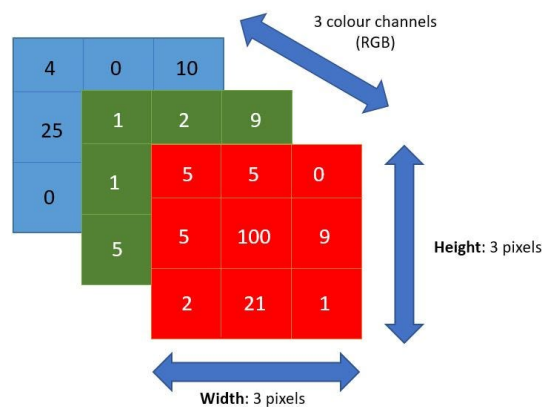
Figura 21 - Cascada atmosférica extendida de un rayo cósmico detectado en HAWC



Fuente: HAWC observatory. <https://www.hawc-observatory.org/observatory/ghsep.php>

De manera similar a las imágenes RGB, donde los valores de cada color se almacenan en capas separadas para su posterior procesamiento (ver figura 22), se representarán los diferentes valores de cada PMT en capas distintas.

Figura 22 - Representación imagen RGB en diferentes capas



Fuente: <https://towardsai.net/p/machine-learning/beginner-guides-to-convolutional-neural-network-from-scratch-kuzushiji-mnist-75f42c175b21>

En cada evento válido, se inicializa una matriz 40 x 30 x 4 con todos los valores en cero y se asignan los valores correspondientes a cada celda, conformando así de forma matemática, una imagen del evento.

Una vez toda la información del evento está representada en la matriz, esta se añade a un listado llamado **events_list** el cual será el que se envía a la CNN para su entrenamiento. Al mismo tiempo se va añadiendo a otra lista llamada **core_coordinates** las coordenadas del núcleo de cada evento de manera que cada evento de **events_list** corresponde a las coordenadas de **core_coordinates** en la misma posición.

En el momento de separar los datos y así obtener los conjuntos de entrenamiento y test, se le pasan estas dos listas siendo **events_list** las características y **core_coordinates** las etiquetas.

La complejidad del algoritmo es $O(n \times m)$, donde n es el número de eventos y m es el tamaño del array de hits detectados en cada evento. La obtención de las posiciones para almacenar los valores no afecta la complejidad previamente mencionada, ya que la posición se calcula mediante los identificadores del tanque junto con las coordenadas del PMT y del hit de manera que la complejidad en este paso es $O(1)$.

3.3.4. Modelo y predicción

Una vez entrenado el modelo, este será almacenado en un archivo para su posterior uso. Para hacer las predicciones simplemente se ha de cargar el modelo con la librería '*load_model*' de *TensorFlow.Keras*. Una vez cargado el modelo, se ha de llamar al método '*predict*' pasándole como parámetro un array de eventos en el mismo formato que el del entrenamiento dando como resultado la predicción de las coordenadas de cada evento.

Los valores de entrada han de ser los indicados en la tabla 3, como se puede apreciar son los mismos que de entrada excepto las coordenadas del núcleo.

Tabla 3 - Valores de entrada para predicciones

Tipo	Nombre
Validación	rec.status
Validación	rec.eventID
Validación	rec.runID
Validación	rec.angleFitStatus
Validación	rec.coreFitStatus
Validación	rec.nHit
Validación	rec.nChAvail
Cálculo posición	event.hit.tankId
Entrenamiento	rec.zenithAngle
Entrenamiento	eventHitCharge
Entrenamiento	eventHitXPMT
Entrenamiento	eventHitYPMT

4. IDENTIFICACIÓN DE REQUISITOS

4.1. PROBLEMA

La necesidad de reconstruir cascadas atmosféricas extendidas producidas por rayos cósmicos mediante el uso de algoritmos de aprendizaje automático.

Los rayos cósmicos son partículas de alta energía procedentes del espacio exterior que, al interactuar con la atmósfera terrestre, generan cascadas de partículas secundarias. La determinación de las coordenadas del núcleo aportará un parámetro imprescindible para mejorar la reconstrucción de otros parámetros importantes de las cascadas.

4.2. CONTEXTO HABITUAL DE USO

A continuación, se describen los principales aspectos del contexto de uso:

1. **Reconstrucción de cascadas atmosféricas extendidas:** El algoritmo de aprendizaje automático será utilizado para predecir las coordenadas del núcleo de las cascadas atmosféricas extendidas detectadas por el observatorio HAWC. Esto permitirá una correcta reconstrucción de la cascada.
2. **Ejecución en un clúster de servidores:** El software se ejecutará en un clúster de servidores dedicado donde se almacenan todos los archivos con la información de los eventos detectados y el software que utiliza el observatorio HAWC. Este entorno computacional proporciona la capacidad necesaria para procesar grandes volúmenes de datos y ejecutar los algoritmos de aprendizaje automático de manera eficiente.
3. **Gestión de datos:** Los datos provenientes de los detectores de HAWC incluyen información detallada e importante sobre las cascadas, como la energía, dirección y tiempo de llegada de las partículas secundarias entre otras variables. La herramienta debe ser capaz de integrar y analizar estos datos para generar reconstrucciones precisas.
4. **Aplicación en investigación y análisis científico:** Los resultados generados por la herramienta serán utilizados por los científicos y técnicos del observatorio para avanzar en el conocimiento sobre los rayos cósmicos y sus propiedades.

4.3. REQUISITOS FUNCIONALES

Los principales requisitos funcionales identificados son:

- **Recopilación de datos históricos:** Cargar y procesar los datos de los eventos desde los archivos XCD proporcionados en una ruta concreta. En el procesado de datos debe filtrar el conjunto de eventos manteniendo solo aquellos que se consideran válidos.
- **Entrenamiento:** Entrenar un modelo de aprendizaje automático al que se le proporciona, por un lado, un conjunto de datos de eventos y, por otro, las coordenadas de cada evento, generando un archivo que contiene el modelo ya entrenado y preparado para realizar predicciones futuras.
- **Predicción de coordenadas:** Capacidad para generar predicciones en tiempo real o mediante procesamiento por lotes.
- **Visualización de resultados:** Los resultados deben ser almacenados en archivos para su posterior uso en la reconstrucción de las cascadas.

4.4. REQUISITOS NO FUNCIONALES

Estos requisitos describen los criterios de operación de la herramienta, como:

- **Confiabilidad:** Las predicciones deben ser precisas.
- **Usabilidad:** La interfaz de usuario debe ser intuitiva y fácil de usar para los diferentes tipos de usuarios previstos.
- **Performance:** La herramienta debe responder rápidamente a las consultas y generar reconstrucciones en un tiempo razonable, todo teniendo en cuenta la magnitud de procesamiento que se requiere en cada ejecución.

5. DESCRIPCIÓN DE LA HERRAMIENTA DESARROLLADA

5.1.PROCESO DE DESARROLLO

5.1.1. Fase 1: Lectura de datos

Hito 1: Crear estructura base del algoritmo

- Descripción: Desarrollar la estructura base del algoritmo que sea capaz de leer archivos XCDF indicados en una ruta específica.
- Tareas:
 - Instalación librerías en el entorno de desarrollo.
 - Implementar funciones para la lectura de archivos XCDF.
 - Verificar la correcta lectura de datos desde archivos XCDF.

Hito 2: Lectura de eventos y transformación a objetos en memoria

- Descripción: Cargar todos los eventos desde los archivos XCDF y transformarlos en objetos de datos estructurados.
- Tareas:
 - Desarrollar clases y estructuras de datos para representar los eventos.
 - Implementar la lógica para relacionar los datos en atributos de objeto.
 - Realizar pruebas de carga y asegurar la correcta relación de los datos.

Hito 3: Cargar y procesar PMT

- Descripción: Cargar información de todos los PMT y guardar los datos necesarios en un archivo CSV para su uso posteriormente.
- Tareas:
 - Leer datos de PMTs desde el archivo *survey.xml*.
 - Filtrar y procesar los datos relevantes de PMTs (ver apartado [3.3.1.1](#)).
 - Guardar los datos procesados en un archivo CSV.

5.1.2. Fase 2: Procesamiento de datos

Hito 4: Diseño de la estructura de datos para entrenamiento

- Descripción: Diseñar una estructura de datos adecuada para el entrenamiento del modelo de aprendizaje automático.
- Tareas:
 - Diseño del formato de la estructura de los datos.
 - Diseño del modo en el cual se van a almacenar los datos guardando una relación entre las características y las capas de la estructura.
 - Implementar la estructura de datos en el código.

Hito 5: Separar los datos para el entrenamiento

- Descripción: Separar los datos en conjuntos de entrenamiento, validación y prueba.
- Tarea: Implementar la lógica para dividir los datos en los conjuntos correspondientes.

Hito 6: Cálculo de la posición de almacenamiento de datos

- Descripción: Calcular la ubicación en la estructura donde se almacenarán los datos preprocesados (ver apartado [3.3.3](#)).
- Tareas:
 - Desarrollar clases y estructuras de datos para representar los PMT.
 - Implementar la lógica para relacionar los datos en atributos de objeto.
 - Cálculo de la posición en el array dependiendo del PMT.

Hito 7: Guardar eventos en la estructura de datos

- Descripción: Guardar los datos almacenados en objetos en la estructura diseñada para el entrenamiento.
- Tareas:
 - Implementar el uso del método que calcula la posición en la estructura
 - Implementar la lógica para almacenar los datos en la estructura de datos.

5.1.3. Fase 3: Desarrollo y entrenamiento del modelo

Hito 8: Crear una arquitectura de red convolucional adecuada

- Descripción: Diseñar y desarrollar una arquitectura de red neuronal convolucional (CNN) adecuada para el problema de predicción de coordenadas.
- Tareas:
 - Investigar y seleccionar la arquitectura de CNN más apropiada.
 - Implementar la arquitectura.
 - Validar la implementación con datos de prueba.

Hito 9: Entrenar el modelo

- Descripción: Entrenar el modelo de CNN con los datos de entrenamiento.
- Tareas:
 - Configurar el entorno de entrenamiento.
 - Ejecutar el proceso de entrenamiento, monitorizando la precisión y el error.
 - Ajustar el modelo en función de los resultados obtenidos durante el entrenamiento.

5.1.4. Fase 4: Evaluación

Hito 10: Predicción y validación

- Descripción: Validar que el modelo entrenado predice correctamente las coordenadas de las cascadas.
- Tarea: Comparar las predicciones del modelo con las coordenadas reales.

5.1.5. Fase 5: Implementación y despliegue

Hito 11: Despliegue del modelo

- **Descripción:** Implementar el modelo entrenado en el clúster de servidores del observatorio HAWC.

- **Tareas:**
 - Preparar el entorno y asegurar su compatibilidad.
 - Realizar pruebas para asegurar el correcto funcionamiento en producción.

5.2.ARQUITECTURA RED CONVOLUCIONAL UTILIZADA

Para el entrenamiento del modelo, la herramienta hace uso de una red convolucional tal y como se explica en el apartado 3.3.3. Esta arquitectura de red convolucional combina capas Conv2D con MaxPooling y seguida de una capa densa donde finalmente se realiza la predicción de coordenadas mediante regresión lineal.

Las capas Conv2D permiten detectar patrones espaciales importantes en los datos, mientras que MaxPooling reduce la dimensionalidad manteniendo las características más relevantes. El uso de Leaky ReLU como función de activación es fundamental para evitar problemas de desvanecimiento de gradientes, lo cual es importante si los datos de entrada contienen valores negativos relevantes. A diferencia de ReLU que convierte los valores negativos en cero, Leaky ReLU los escala por un pequeño factor, preservando esta información durante el entrenamiento.

Tras extraer las características, la capa Flatten convierte los mapas tridimensionales en un vector, permitiendo que las capas densas procesen la información y produzcan una salida numérica, en este caso las coordenadas del núcleo de una cascada atmosférica. Esta arquitectura es eficiente para transformar información espacial compleja en predicciones precisas de variables continuas. En la figura 23 podemos observar el método que genera la arquitectura.

Figura 23 - Arquitectura red convolucional desarrollada

```
def get_cnn():  
    return Sequential([  
        Conv2D(128, (3, 3), activation=ACTIVATION_FUNCTION, input_shape=(ARRAY_WIDTH, ARRAY_HEIGHT, ARRAY_LAYERS)),  
        MaxPooling2D((2, 2)),  
        Conv2D(256, (3, 3), activation=ACTIVATION_FUNCTION),  
        MaxPooling2D((2, 2)),  
        Conv2D(256, (3, 3), activation=ACTIVATION_FUNCTION),  
        Flatten(),  
        Dense(256, activation=ACTIVATION_FUNCTION),  
        Dense(2, activation='linear')  
    ])
```

5.3.REQUISITOS DE EJECUCIÓN

Para una correcta ejecución de la aplicación es necesario disponer de lo siguiente:

- Ruta de los eventos *.xcd con todos los archivos que van a entrenar el modelo
- Ruta del archivo channels.csv creado en este proyecto con los datos de los PMTs
- Entorno virtual con las siguientes tecnologías de la tabla 3 instaladas

Tabla 4 - Tecnologías necesarias

Nombre	Descripción
Python v3	Lenguaje de programación
Keras	Librería específica para redes neuronales integrada en TensorFlow
Scikit-Learn	Separación datos entrenamiento
XCDF	Librería para la lectura de archivos xcd
CSV	Librería para la gestión de archivos CSV
Matplotlib Pyplot	Librería para la visualización gráfica de resultados.

5.4.EJECUCIÓN

5.4.1. Entrenamiento

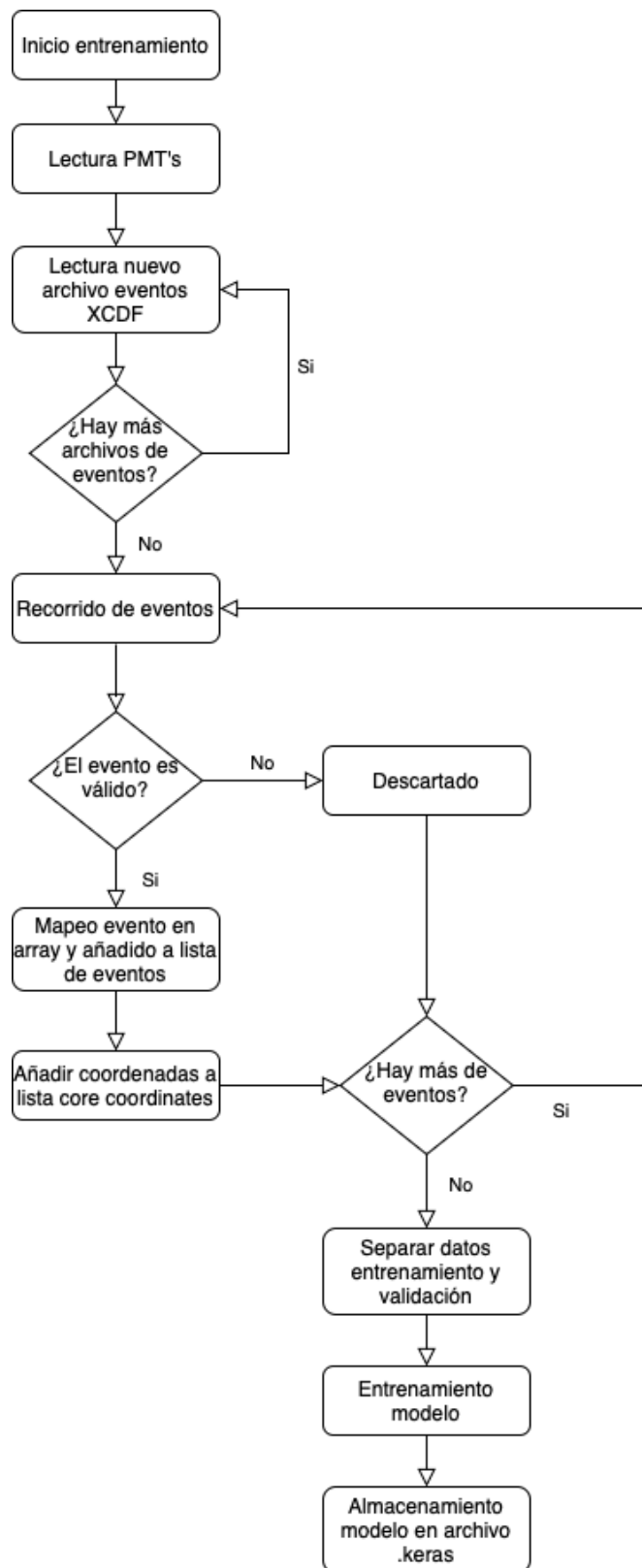
Para proceder al entrenamiento del modelo, primero se ha de indicar la ruta de los archivos con los eventos en la constante *FILES_TO_TRAIN_PATH* al inicio del archivo *core_coordinates_predictor.py*.

Una vez definida la ruta y activado el entorno con todas las librerías instaladas, se ha de ejecutar el siguiente comando:

```
> python core_coordinates_predictor.py train
```

En la figura 24 podemos observar la secuencia que recorre el algoritmo cuando se ejecuta en modo entrenamiento.

Figura 24 - Secuencia entrenamiento



5.4.2. Predicción

Para proceder a la predicción de coordenadas mediante el uso del modelo generado con el entrenamiento, primero se ha de indicar la ruta de los archivos con los eventos que necesitamos predecir sus coordenadas en *FILES_TO_PREDICT_PATH* al inicio del archivo *core_coordinates_predictor.py*.

Una vez entrenado el modelo, definida la ruta y activado el entorno con todas las librerías instaladas, se ha de ejecutar el siguiente comando:

```
> python core_coordinates_predictor.py predict
```

5.5.ARCHIVO DE SALIDA PREDICCIONES

Una vez completado el entrenamiento del modelo, el algoritmo también se encarga de predecir los eventos presentes en los archivos localizados en la ruta especificada para ello, siguiendo las instrucciones descritas en el apartado 5.3.2. Las predicciones generadas por el algoritmo se almacenan en un archivo de salida. Este archivo incluirá, para cada predicción, la información detallada en la tabla 5.

Tabla 5 - Campos archivo salida

Nombre	Descripción
runID	Identificador de la ejecución de captura del evento
eventID	Identificador del evento
rec.zenithAngle	Angulo cenital del evento (reconstrucción del observatorio)
mc.logEnergy	Logaritmo de energía del evento
mc.coreX	Coordenada X del núcleo proporcionada por HAWC
mc.coreY	Coordenada Y del núcleo proporcionada por HAWC
pred.coreX	Predicción de la coordenada X del núcleo del evento
pred.coreY	Predicción de la coordenada Y del núcleo del evento

6. EVALUACIÓN

6.1.DESCRIPCIÓN DE LOS RESULTADOS

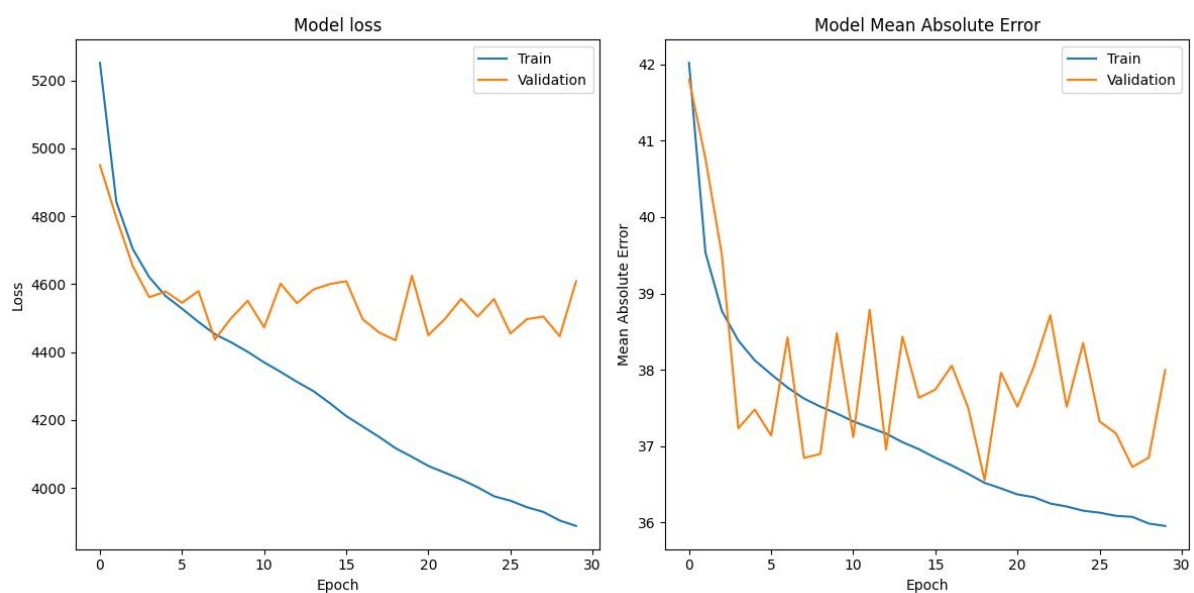
6.1.1. Grafica de aprendizaje

La gráfica del aprendizaje muestra la evolución de dos métricas clave a lo largo del proceso de entrenamiento del modelo: el loss (pérdida) y el error absoluto medio (mean absolute error).

- **Loss:** Esta métrica refleja qué tan bien se ajusta a los datos de entrenamiento. A lo largo del tiempo, se espera que esta métrica disminuya, indicando que el modelo está mejorando su capacidad para predecir correctamente los valores esperados. Un descenso constante en el valor de la pérdida a lo largo del tiempo sugiere que el modelo está aprendiendo de manera efectiva.
- **Mean Absolute Error (MAE):** Esta métrica mide la magnitud promedio de los errores en un conjunto de predicciones. Un valor más bajo del MAE indica predicciones más precisas. Al igual que el loss, una tendencia decreciente en el MAE a lo largo del tiempo indica una mejora en la precisión del modelo.

En la figura 25 podemos observar la evolución de estas dos métricas del modelo finalmente utilizado a lo largo de las diferentes épocas del entrenamiento.

Figura 25 - Evolución aprendizaje



Evaluación de la pérdida (Loss) en el entrenamiento y validación:

- La curva de pérdida para el conjunto de entrenamiento (Train) disminuye de manera constante a lo largo de las épocas, lo que indica que el modelo está aprendiendo y ajustándose mejor a los datos de entrenamiento.
- Sin embargo, la curva de pérdida para el conjunto de validación (Validation) no sigue el mismo patrón. Después de una ligera disminución inicial, realiza varias subidas y bajadas manteniéndose en el mismo nivel sin que consiga mejorar con el tiempo.

Evaluación del error absoluto medio (Mean Absolute Error, MAE):

- Al igual que con la pérdida, el MAE del entrenamiento disminuye de manera constante, lo que nuevamente indica que el modelo está mejorando su precisión en los datos de entrenamiento.
- El MAE en el conjunto de validación muestra fluctuaciones y una tendencia a aumentar en las últimas épocas.

6.1.2. Histograma 2D de comparación entre predicciones y valores reales

Un histograma 2D es una herramienta gráfica que representa cómo se distribuyen dos variables continuas en conjunto. Divide el plano cartesiano en una cuadrícula, donde cada celda (o "bin") cuenta cuántos pares de valores caen dentro de ese rango específico para ambas variables. La intensidad del color refleja la densidad de los datos en esa área, permitiendo visualizar relaciones, tendencias y concentraciones entre las dos variables.

En este trabajo nos presenta una comparación visual entre el valor calculado a partir de las predicciones del modelo y el valor correspondiente a las coordenadas reales siendo 'r' el valor obtenido con la siguiente formula:

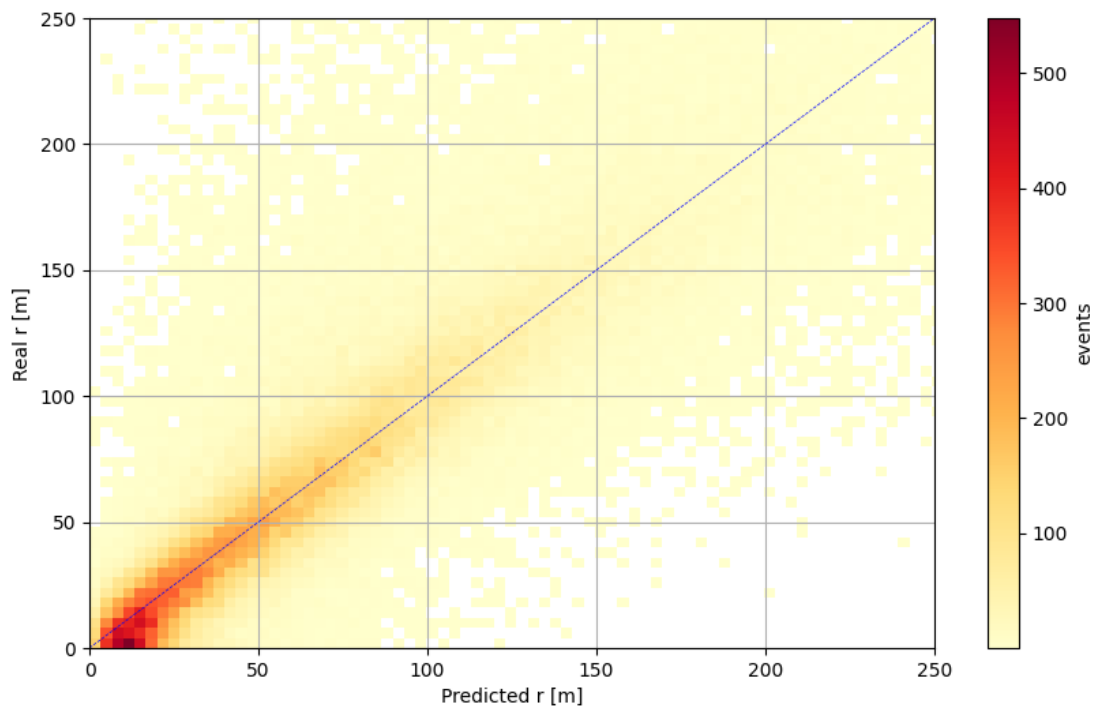
$$1. \quad r = \sqrt{x^2 + y^2}$$

Idealmente, si las predicciones del modelo fueran perfectas, todos los puntos deberían alinearse en la diagonal $y = x$, lo que indicaría que los valores r_{pred} y r_{real} son iguales. Una alta concentración de puntos cerca de esta línea diagonal sugiere que las predicciones son precisas y están en estrecha concordancia con los valores reales. En contraste, si los puntos están

dispersos lejos de la diagonal, indicaría que las predicciones del modelo difieren considerablemente de los valores reales.

En la figura 26 se muestra el histograma bidimensional que compara las predicciones generadas por el modelo entrenado con los valores reales para un conjunto de eventos.

Figura 26 – Histograma 2D todos los eventos



6.1.2.1. Histograma 2D por niveles de energía

Dado que ' r ' se calcula a partir de las coordenadas del núcleo de un evento, la posición de un punto en el histograma puede reflejar su ubicación geográfica en o cerca del observatorio. La proximidad a la línea de identidad indica la precisión de las predicciones del modelo. El punto (0,0) en el histograma corresponde al centro del observatorio, mientras que un punto cercano a (100,100) estaría situado cerca del perímetro (ver figura 21).

La razón por la que tenemos valores de coordenadas más allá de este perímetro es que muy probablemente hayamos detectado las partículas secundarias más alejadas del núcleo del cono (ver las partículas de los extremos de la figura 2) y aun así nos predice la posición del núcleo fuera del observatorio. Este fenómeno ocurre cuando la cascada atmosférica es

altamente energética, lo que sugiere una relación entre la energía detectada y la distancia registrada en el histograma.

A continuación, desde la figura 27 hasta la figura 30 podemos observar los diferentes histogramas bidimensionales con los eventos según su energía, en este caso el logaritmo de la energía detectada del evento. En cada gráfico, se ha trazado la línea de identidad para facilitar la evaluación de la precisión del modelo.

Figura 27 – Histograma 2D eventos con logaritmo energía mayor que 1 y menor que 2

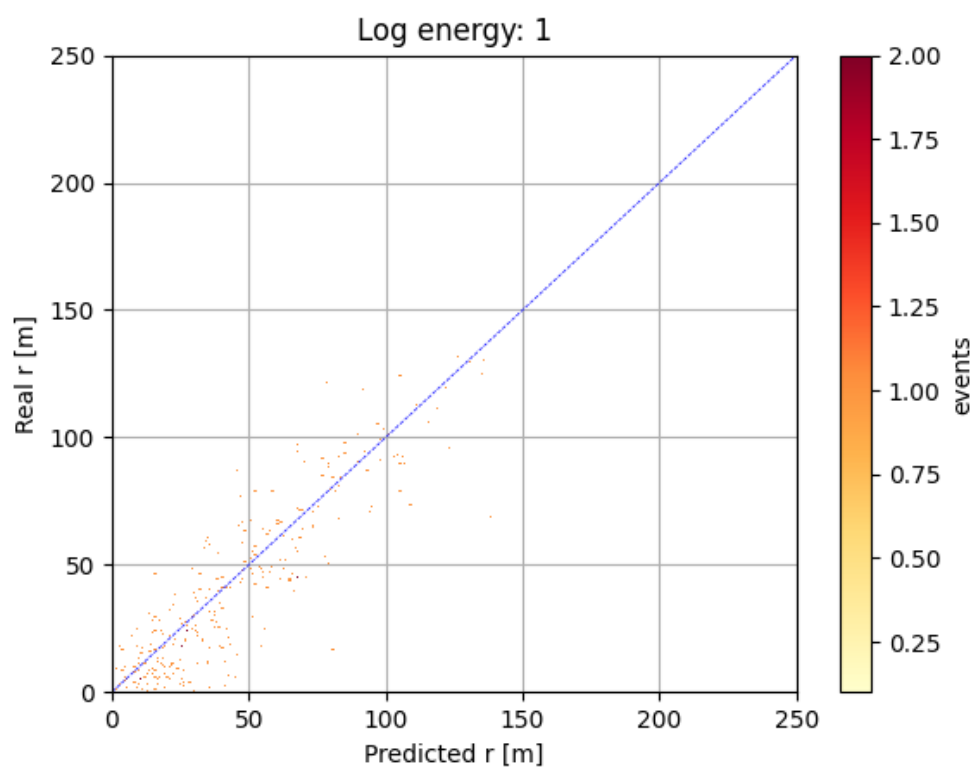


Figura 28 - Histograma 2D eventos con logaritmo energía mayor que 2 y menor que 3

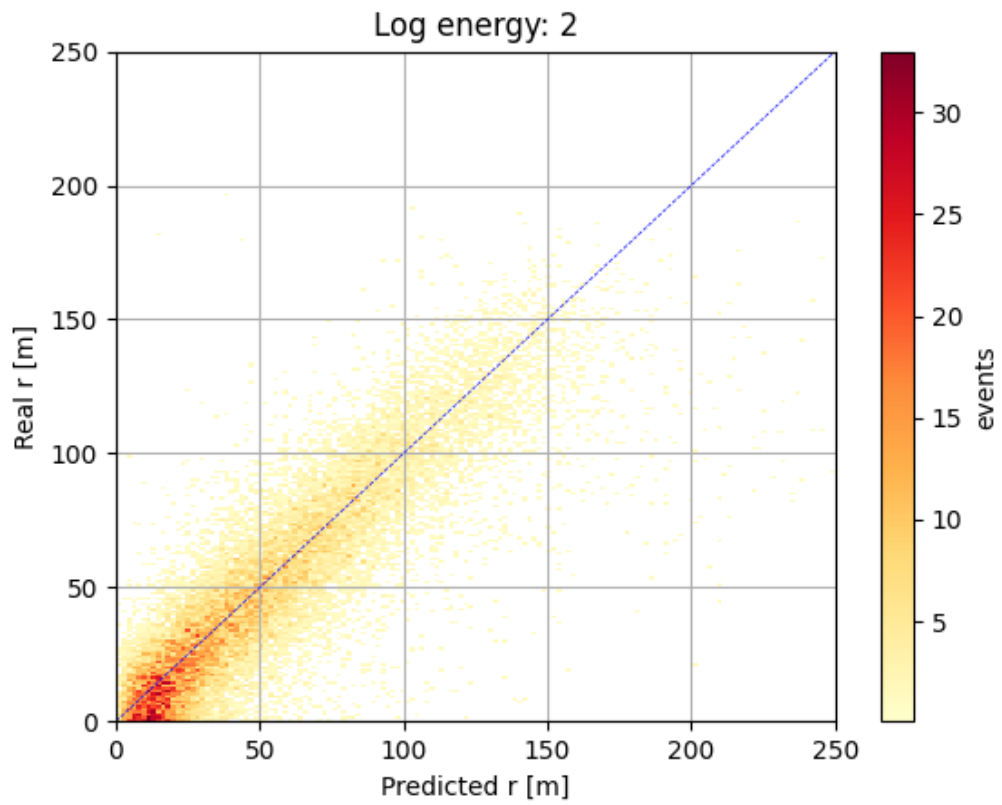


Figura 29 - Histograma 2D eventos con logaritmo energía mayor que 3 y menor que 4

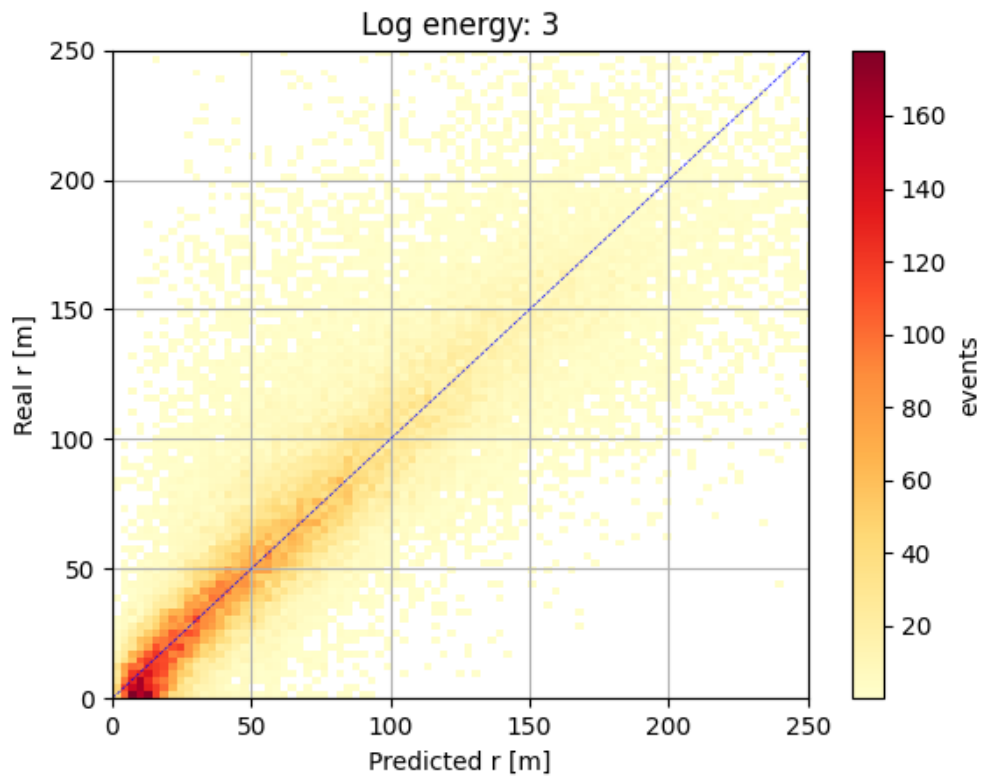
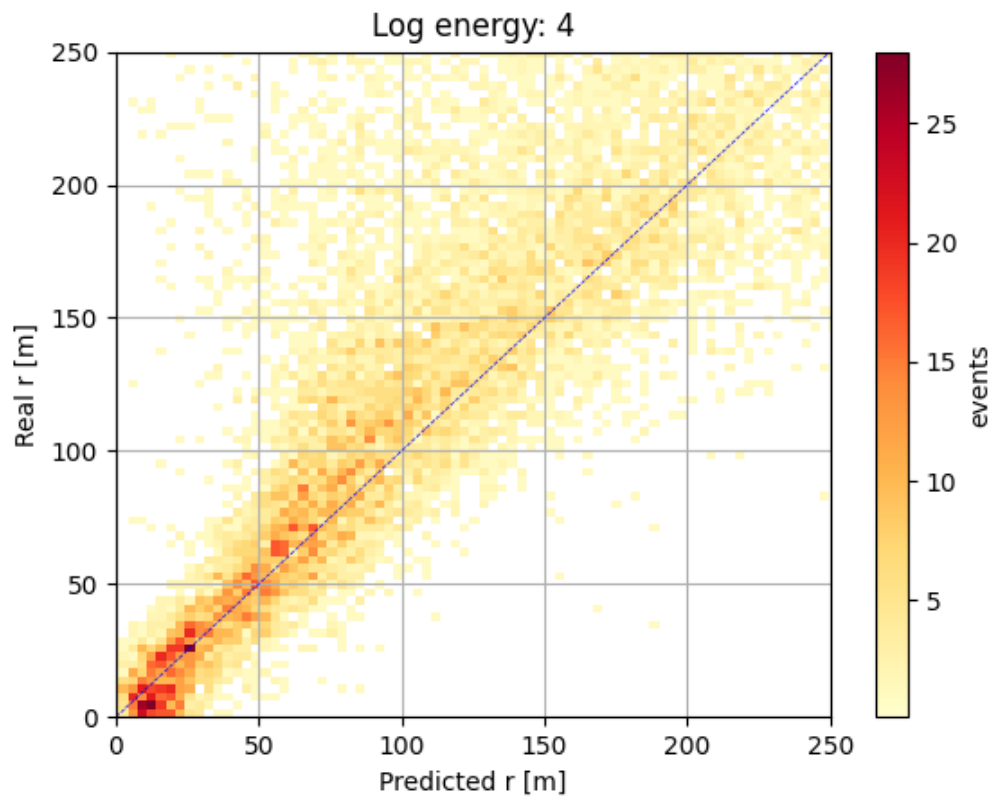


Figura 30 - Histograma 2D eventos con logaritmo energía mayor que 4 y menor que 5



6.1.3. Histogramas de las diferencias en los ejes de coordenadas

Un histograma es un gráfico que representa la distribución de un conjunto de datos, mostrando la frecuencia con la que ocurren ciertos rangos de valores. Es particularmente útil para visualizar cómo se distribuyen los datos.

En este caso, el histograma muestra las diferencias entre las coordenadas predichas y las reales. Al ajustar una función gaussiana alrededor del cero para estas diferencias, podemos visualizar cómo se distribuyen los errores.

Idealmente, la mayor parte de los casos debería concentrarse en la barra correspondiente a cero, disminuyendo rápidamente a medida que nos alejamos de este valor. Para medir esta dispersión, utilizamos el valor de sigma (σ): una sigma más baja indica que el modelo predice con mayor precisión.

A continuación, se presentan los histogramas de las diferencias en las coordenadas X (figura 31) e Y (figura 32) entre las predicciones y los valores reales para todos los eventos, sin diferenciar por el logaritmo de la energía.

Figura 31 - Histograma diferencias coordenadas X (red convolucional)

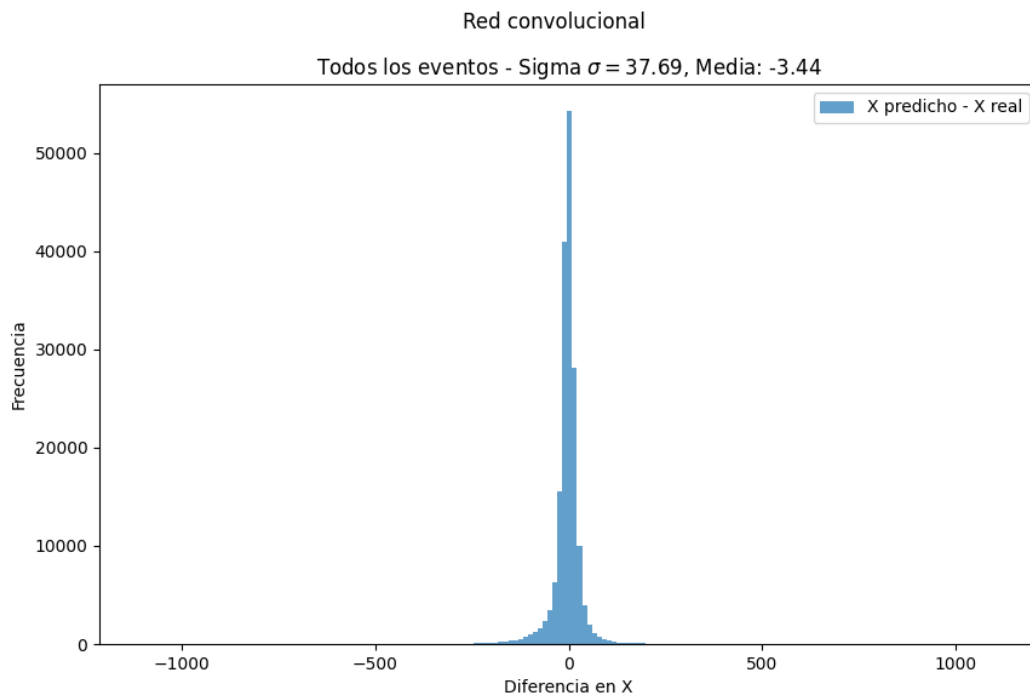
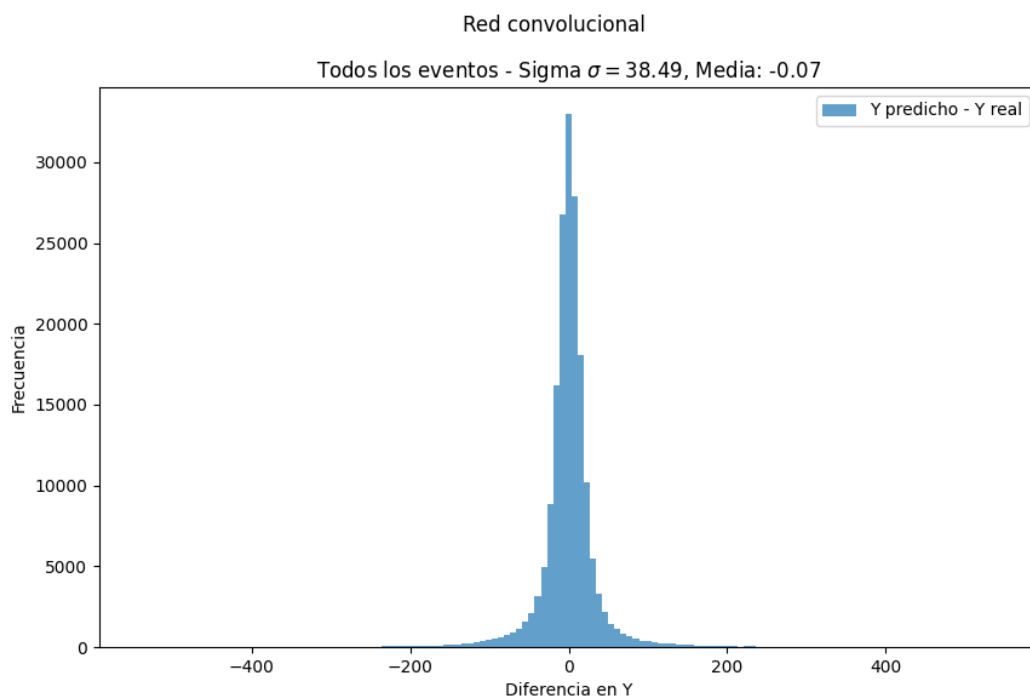


Figura 32 - Histograma diferencias coordenadas Y (red convolucional)



En el apéndice 8.1, se presentan los mismos histogramas, pero separando los eventos por logaritmo de energía, de esta manera se podrá apreciar la precisión del modelo a la hora de predecir las coordenadas del núcleo según el nivel de energía del evento.

6.2.EVALUACIÓN DE LOS RESULTADOS

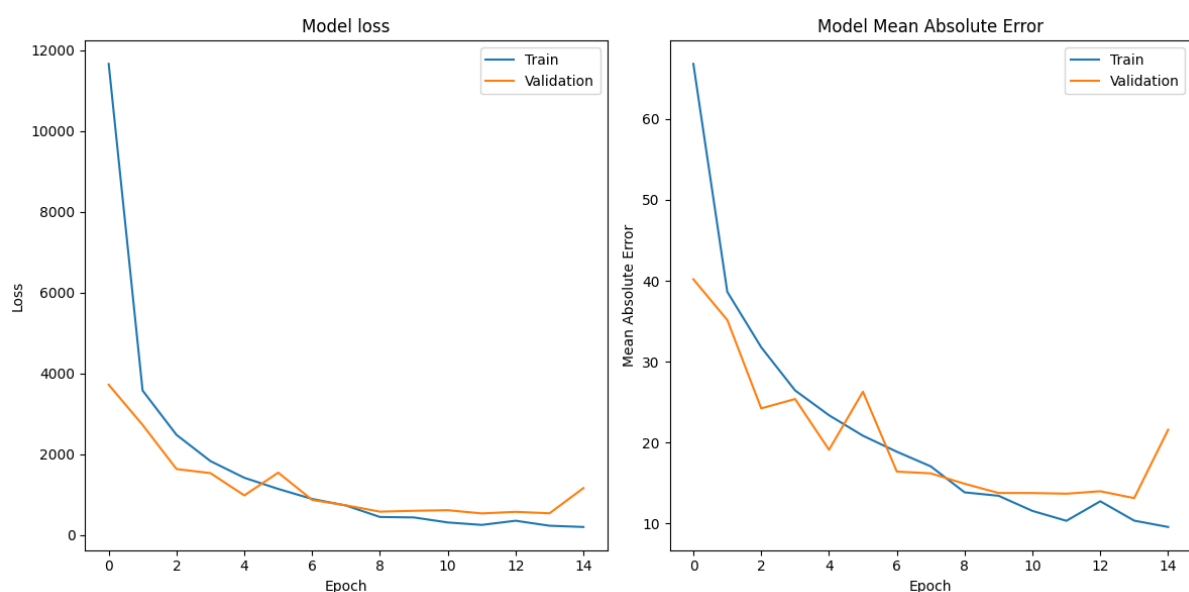
6.2.1. Entrenamiento

Como se detalla en el apartado 6.1.1, las dos métricas de entrenamiento muestran un rendimiento de aprendizaje óptimo, evidenciado por una disminución constante. Sin embargo, las métricas de validación dejan de mejorar y comienzan a fluctuar dentro de un rango de valores, sin mostrar signos de progreso a medida que avanzan las épocas. Esta situación se atribuye a la alta variabilidad de los datos, donde un evento puede presentar características muy diferentes a otros, ya sea por la posición en el observatorio, la energía de cada hit o la inclinación de este.

Para validar esta teoría, se realizó un entrenamiento utilizando mil eventos, ajustando sus características para que fueran más homogéneas. Específicamente, se seleccionaron mil eventos con una inclinación entre 45 y 50 grados, y todos ellos con un número de hits comprendido entre 600 y 800.

En la figura 33 se puede observar cómo las métricas de validación presentan un comportamiento mucho más similar al de las métricas de entrenamiento. Además, podemos observar que los valores de las métricas de entrenamiento son significativamente mejores.

Figura 33 - Entrenamiento con eventos similares



6.2.2. Histogramas 2d

En los histogramas 2D, se observa que los resultados en los distintos niveles de energía concentran la mayoría de los eventos en torno a la diagonal imagen ($y = x$), lo que sugiere que el modelo está realizando la mayoría de las predicciones correctamente.

También se observa que la mayor densidad de eventos se concentra cerca del origen (0,0), disminuyendo gradualmente a medida que los valores se alejan de este punto. Además, conforme nos alejamos del origen, se aprecia una creciente dispersión de los puntos respecto a la diagonal de la imagen, especialmente en valores superiores a (100, 100).

Esto se debe a que, conforme nos alejamos del perímetro del observatorio, el modelo empieza a perder precisión en sus predicciones. Esto es comprensible, ya que cuanto más distante esté el evento del observatorio, más difícil resulta obtener información suficiente, lo que complica tanto el entrenamiento del modelo con eventos de este tipo como la predicción de estos.

Este fenómeno también se observa al revisar los distintos histogramas por logaritmo de energía, donde se aprecia que, a mayores niveles de energía, la dispersión es más pronunciada y las escalas en los ejes de coordenadas son mayores. Se observa que el modelo produce buenos resultados para aquellos eventos en los que el logaritmo de la energía es inferior a 5.

6.2.3. Histogramas diferencias en X e Y

Tal como se indica en el apartado 6.1.3, el objetivo es obtener valores de sigma lo más bajos posible, con las distribuciones centradas en cero. Esto indicaría un número mínimo de errores en las predicciones.

Al comparar los gráficos y valores obtenidos con el algoritmo desarrollado en este trabajo con los resultados actuales del observatorio HAWC, se pueden destacar dos mejoras significativas. En primer lugar, las gaussianas presentadas en las figuras 39 a 46 ubicadas en el apéndice 8.1, están centradas en cero, a diferencia de las obtenidas por HAWC, cuyos centros de las gaussianas se alejan de este escenario ideal, lo que refleja una menor precisión en la determinación de las coordenadas del núcleo de la cascada.

En segundo lugar, se observa una notable mejora en todos los valores de sigma obtenidos con el algoritmo de este trabajo en comparación con los valores obtenidos por HAWC, tanto en

los valores medios como en los diferentes valores de sigma segmentados por logaritmo de energía.

Figura 34 – Histograma diferencias en coordenada X (valores HAWC)

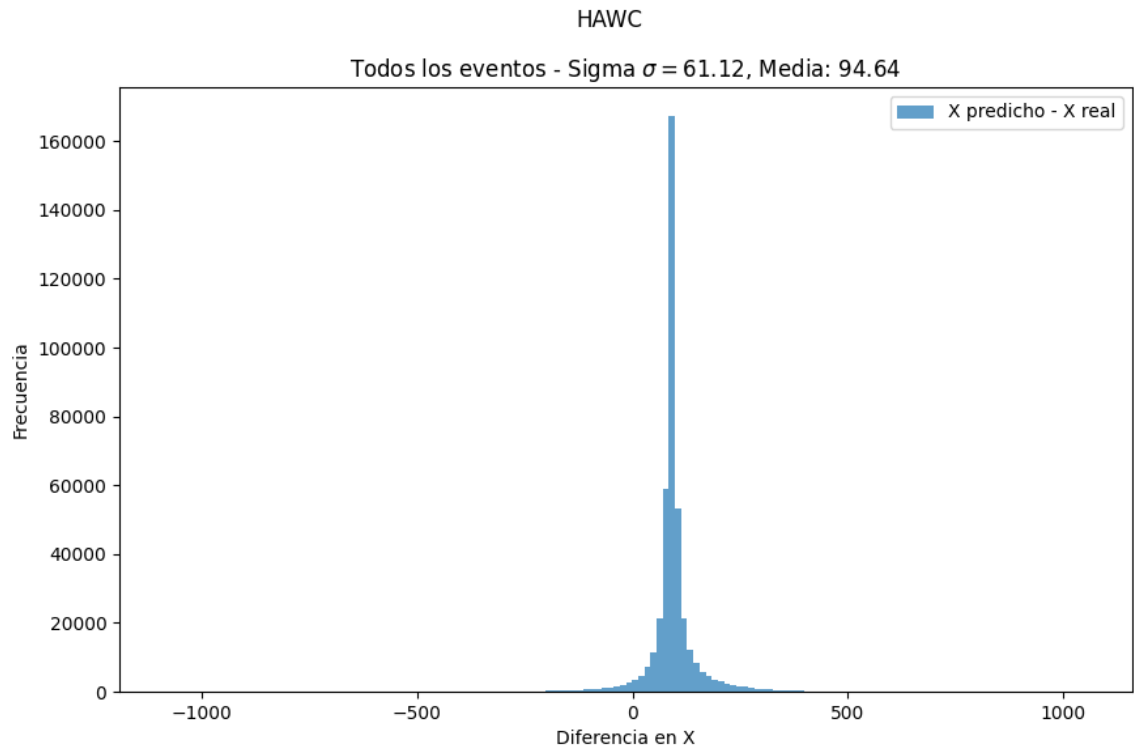
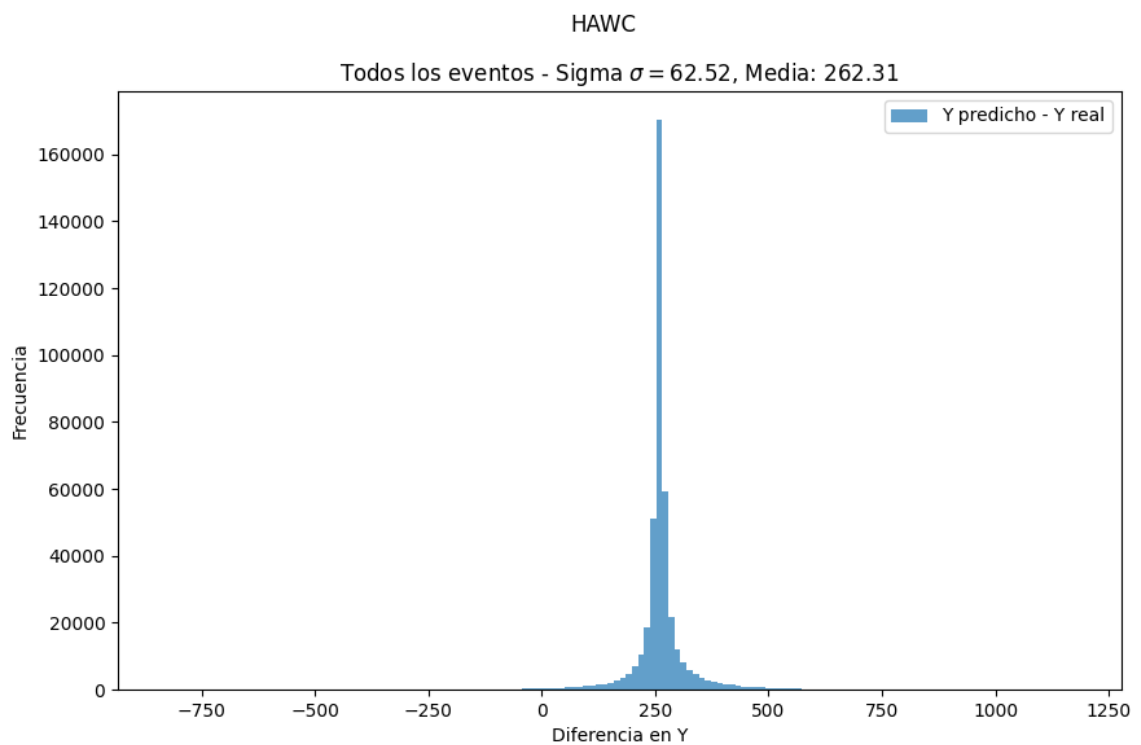


Figura 35 - Histograma diferencias en coordenada Y (valores HAWC)



Los valores de sigma obtenidos por el observatorio HAWC con sus algoritmos actuales son de '61.12' para la coordenada X y '62.52' para la coordenada Y, como se muestra en las figuras 34 y 35. Estos valores representan una sigma media de todos los eventos sin distinción por logaritmo de energía. Desde la figura 47 a la 54 ubicadas en el apéndice 8.2, se pueden ver los resultados actuales de HAWC en los diferentes niveles de energía.

A continuación, en la tabla 6 se resumen todos los valores de sigma correspondientes a los histogramas presentados en los apartados 6.1.3, apéndices y en este apartado.

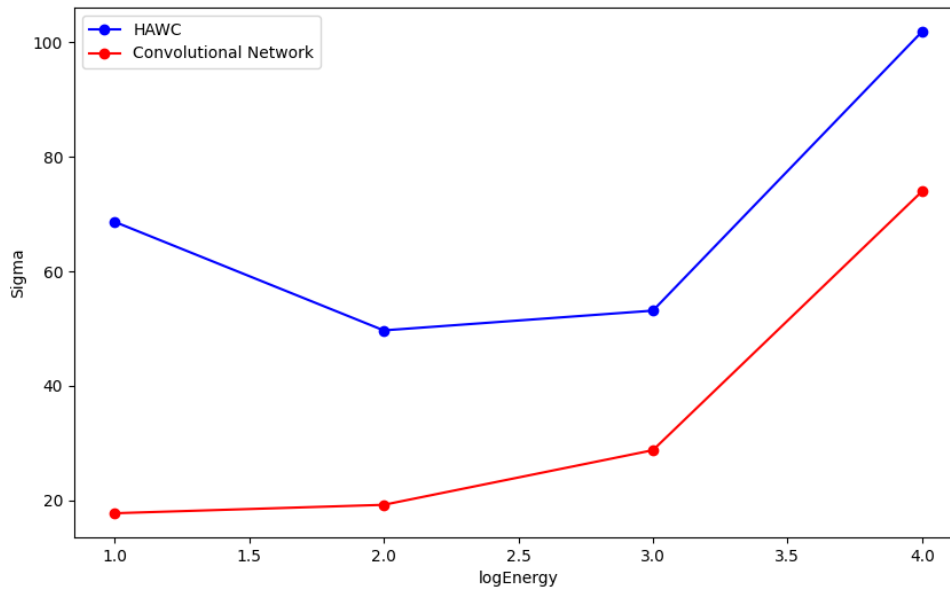
Tabla 6 - Comparativa resultados sigma (HAWC – Red convolucional)

Coordenada	Energía	HAWC	Red Convolucional
X	Todos los eventos	61.12	37.69
Y	Todos los eventos	62.52	38.49
X	$1 \leq \log \text{Energy} < 2$	69.34	17.60
Y	$1 \leq \log \text{Energy} < 2$	67.95	17.92
X	$2 \leq \log \text{Energy} < 3$	48.45	19.75
Y	$2 \leq \log \text{Energy} < 3$	50.93	18.70
X	$3 \leq \log \text{Energy} < 4$	52.56	28.50
Y	$3 \leq \log \text{Energy} < 4$	53.71	29.03
X	$4 \leq \log \text{Energy} < 5$	101.31	72.47
Y	$4 \leq \log \text{Energy} < 5$	102.41	75.46

El gráfico de la figura 36, extraído de los datos de la tabla 6, muestra la relación entre sigma y el logaritmo de la energía de los eventos, comparando los valores actuales del observatorio con los obtenidos en este trabajo.

Como se menciona en el apartado 6.2.2, esta gráfica también muestra un aumento de sigma a medida que incrementa el nivel de energía de los eventos. Esto indica que, debido a la relación entre distancia y energía, cuanto más lejos esté el evento del centro, menor será la precisión de la predicción.

Figura 36 – Relación entre energía del evento y el valor sigma



En la tabla 7 podemos comparar las medias de las diferencias de cada histograma entre HAWC y el algoritmo desarrollado en este trabajo. Esta tabla junto a la 6 nos proporciona la suficiente información para generar el gráfico de la figura 37.

Tabla 7 - Comparativa media diferencias (HAWC – Red convolucional)

Coordenada	Energía	HAWC	Red Convolucional
X	Todos los eventos	94.64	-3.44
Y	Todos los eventos	262.31	-0.07
X	$1 \leq \log \text{Energy} < 2$	93.30	-1.92
Y	$1 \leq \log \text{Energy} < 2$	265.57	-2.17
X	$2 \leq \log \text{Energy} < 3$	96.93	-0.60
Y	$2 \leq \log \text{Energy} < 3$	263.29	-0.48
X	$3 \leq \log \text{Energy} < 4$	93.41	-2.29
Y	$3 \leq \log \text{Energy} < 4$	261.43	0.31
X	$4 \leq \log \text{Energy} < 5$	92.62	-12.39
Y	$4 \leq \log \text{Energy} < 5$	261.82	-0.67

En las figuras 37 y 38, se presenta una comparativa detallada del error entre las dos soluciones analizadas: HAWC y la red convolucional. En el eje X se representa el logaritmo de la energía

de los eventos, mientras que el eje Y muestra los valores de la media obtenidos, conforme se presentan en la tabla 7.

Cada uno de los puntos dibujados en la gráfica está acompañado de una línea vertical que representa el valor de sigma, tal como se indica en la tabla 6, lo que permite visualizar las desviaciones estándar correspondientes para cada media.

- Para los resultados de HAWC, se observa que las medias son significativamente más altas, particularmente en la coordenada Y, con valores que superan los 260, mientras que la coordenada X presenta valores cercanos a los 93. Las líneas de error (sigma) en HAWC son bastante pronunciadas, especialmente para los eventos de mayor energía.
- En contraste, los valores medios obtenidos por la red convolucional son mucho más bajos, con medias cercanas a cero. Las líneas de error para la red convolucional son más moderadas, especialmente en los primeros intervalos de energía, pero crecen de manera significativa en el último.

En resumen, la figura ilustra de manera efectiva las diferencias en las predicciones de ambas soluciones, destacando la mayor incertidumbre en las predicciones de HAWC en los eventos de alta energía y la tendencia a obtener valores más bajos y estables en las predicciones de la red convolucional.

Figura 37 – Relación entre energía y el valor de resolución en coordenada X

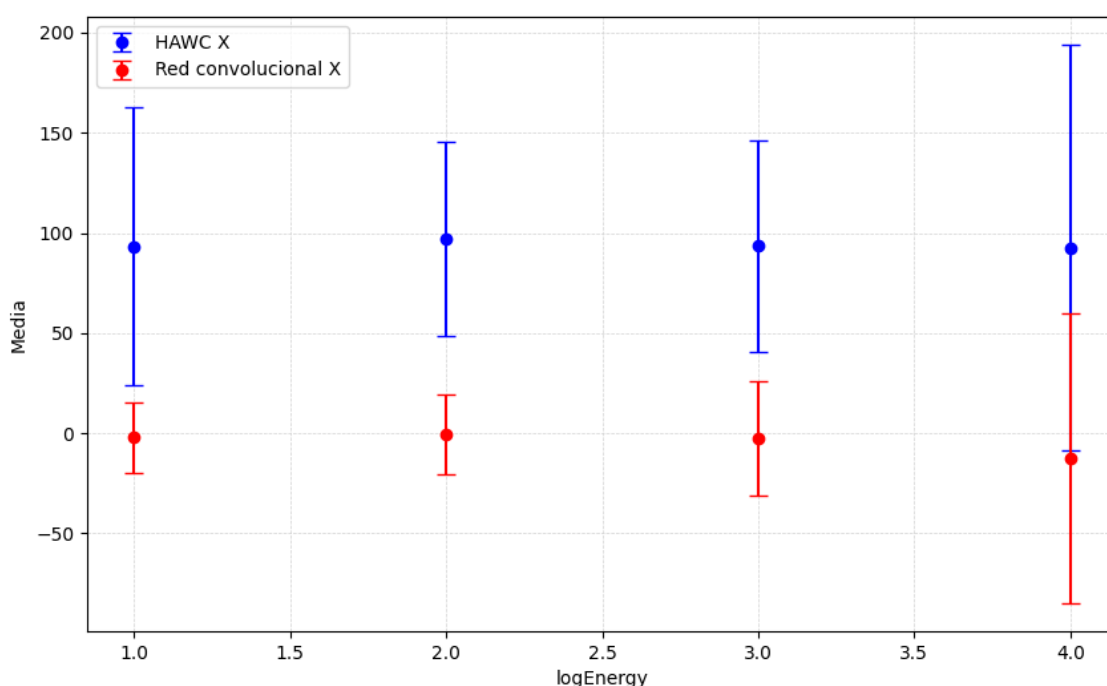
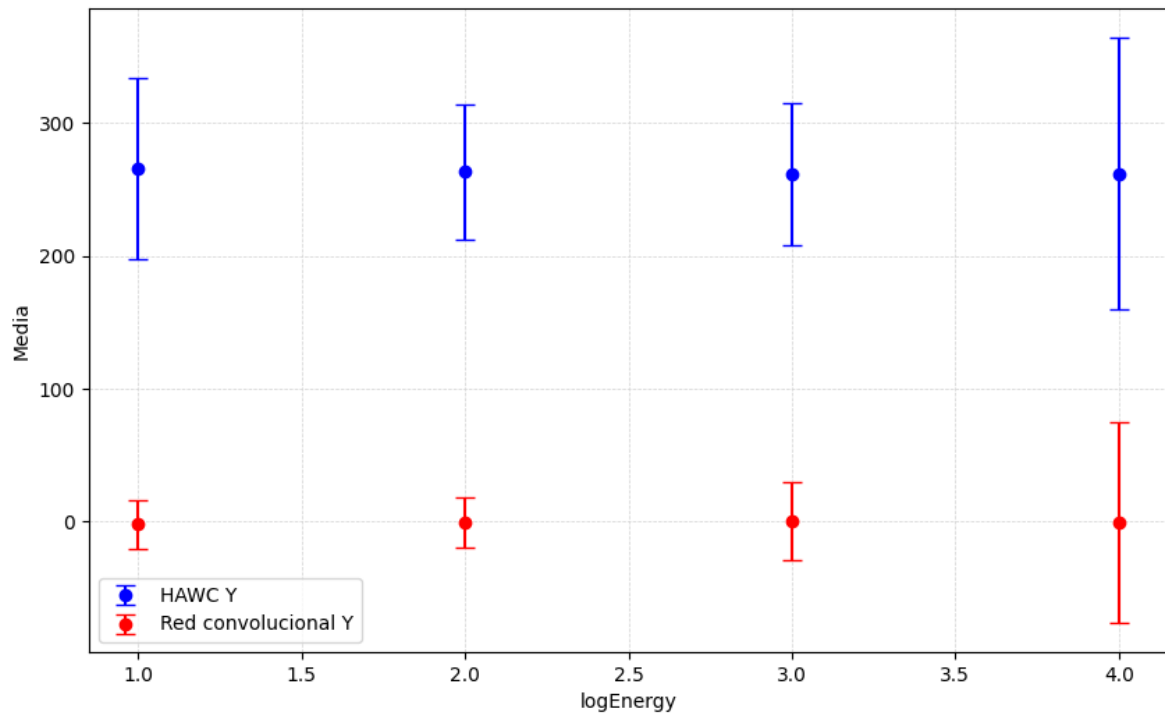


Figura 38 - Relación entre energía y el valor de resolución en coordenada Y



7. Conclusiones y trabajo futuro

7.1. Conclusiones

Debido a los hallazgos de realizar investigación en rayos cósmicos, el observatorio HAWC se ha propuesto como objetivo verificar y cuantificar los parámetros empleados para su análisis. Uno de los parámetros más cruciales es la determinación precisa de las coordenadas del núcleo de la cascada atmosférica, ya que es el primer parámetro que se calcula y de su precisión dependen otros, como la estimación de la energía. El objetivo principal de este trabajo fue desarrollar un modelo capaz de predecir la posición de los eventos generados por rayos cósmicos.

Se ha desarrollado un algoritmo que emplea una red neuronal convolucional que, a pesar de no tener imágenes de los eventos, la información se representó de forma similar en matrices disponiendo los datos de forma análoga a las posiciones geográficas de los sensores. Dado que cada detector genera distintos parámetros, estos se han almacenado en diferentes capas de la matriz, de manera similar a cómo las imágenes a color almacenan los datos de cada canal en capas separadas.

Tras la recopilación de un número significativo de eventos, el algoritmo ha generado un modelo que predice estas coordenadas con mayor precisión que el modelo actualmente utilizado en el observatorio HAWC, como lo demuestran los resultados obtenidos en los histogramas presentados en el apartado 6.

Adicionalmente, los resultados comparativos entre ambos modelos, como se observa en las figuras 37 y 38, muestran que la red convolucional presenta un menor error en la predicción de las coordenadas, especialmente en eventos de menor energía, mientras que el modelo de HAWC exhibe mayores errores en todos los niveles de energía.

7.2. Líneas de trabajo futuro

A pesar de que el modelo desarrollado ya ha demostrado una precisión superior en la predicción de las coordenadas del núcleo de las cascadas atmosféricas extendidas, se considera que un entrenamiento más intensivo podría generar mejoras significativas

adicionales. Una de las principales estrategias para lograr esto es la ampliación de los recursos disponibles y el aumento de la cantidad de datos utilizados en el entrenamiento.

Otra línea de trabajo interesante sería desarrollar diferentes modelos entrenados con conjuntos de eventos similares, agrupados según ángulo de llegada, energía y cantidad de hits del evento. Posteriormente, durante la predicción de resultados, se seleccionarían los modelos más adecuados para cada caso específico.

8. APÉNDICE

8.1. HISTOGRAMAS COMPLEMENTARIOS APARTADO 6.1.3

Figura 39 - Histograma diferencias coordenadas X (red convolucional), $1 \leq \log \text{Energy} < 2$

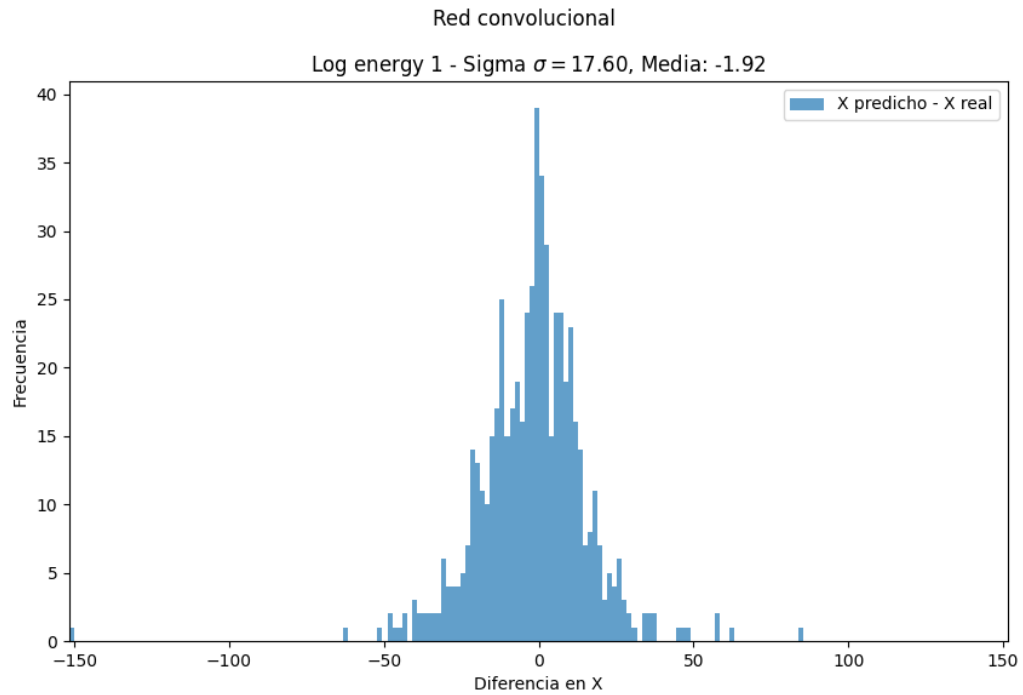


Figura 40 - Histograma diferencias coordenadas Y (red convolucional), $1 \leq \log \text{Energy} < 2$

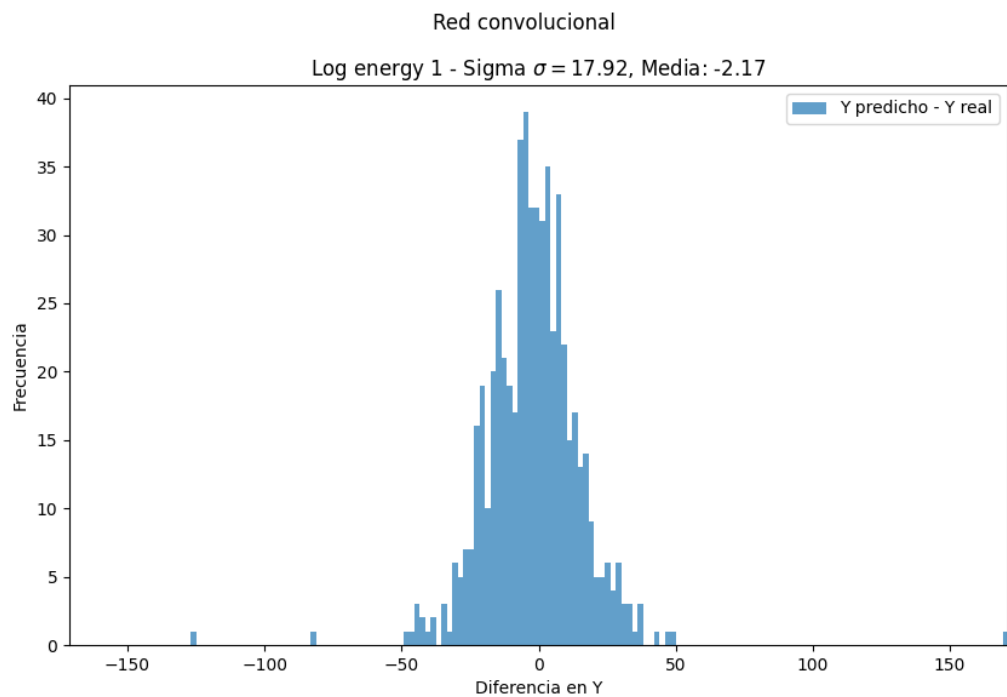


Figura 41 - Histograma diferencias coordenadas X (red convolucional), $2 \leq \log \text{Energy} < 3$

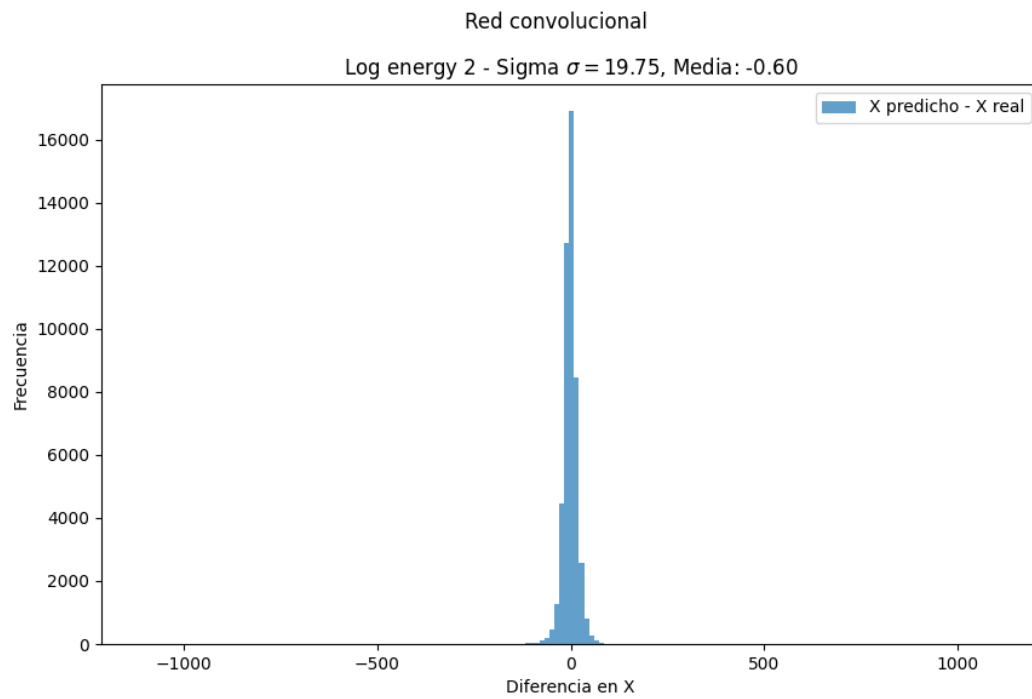


Figura 42 - Histograma diferencias coordenadas Y (red convolucional), $2 \leq \log \text{Energy} < 3$

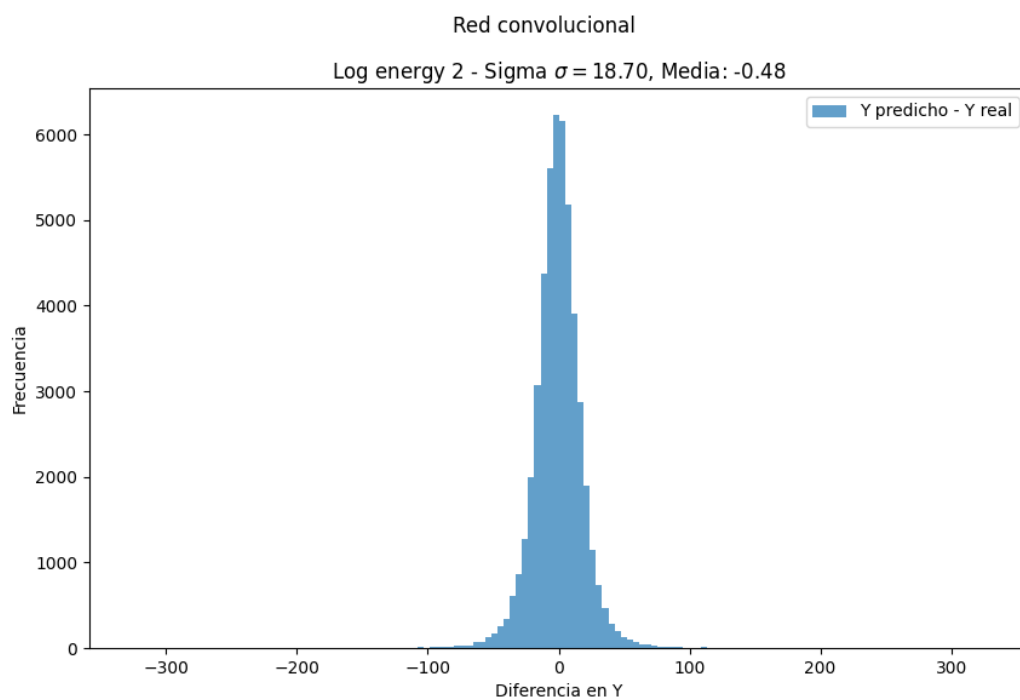


Figura 43 - Histograma diferencias coordenadas X (red convolucional), $3 \leq \log \text{Energy} < 4$

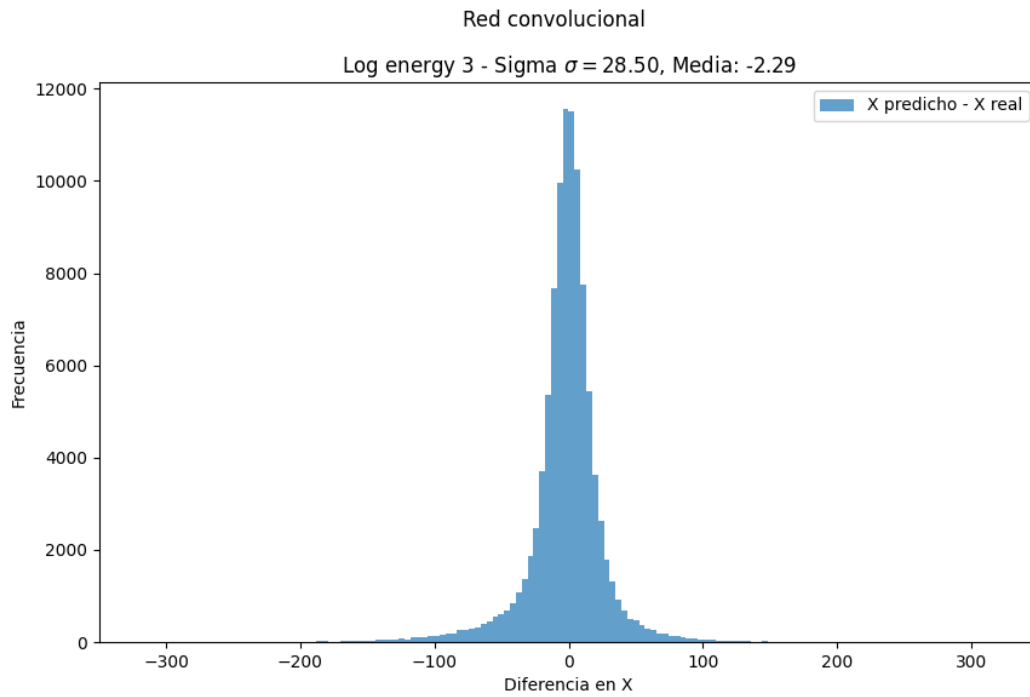


Figura 44 - Histograma diferencias coordenadas Y (red convolucional), $3 \leq \log \text{Energy} < 4$

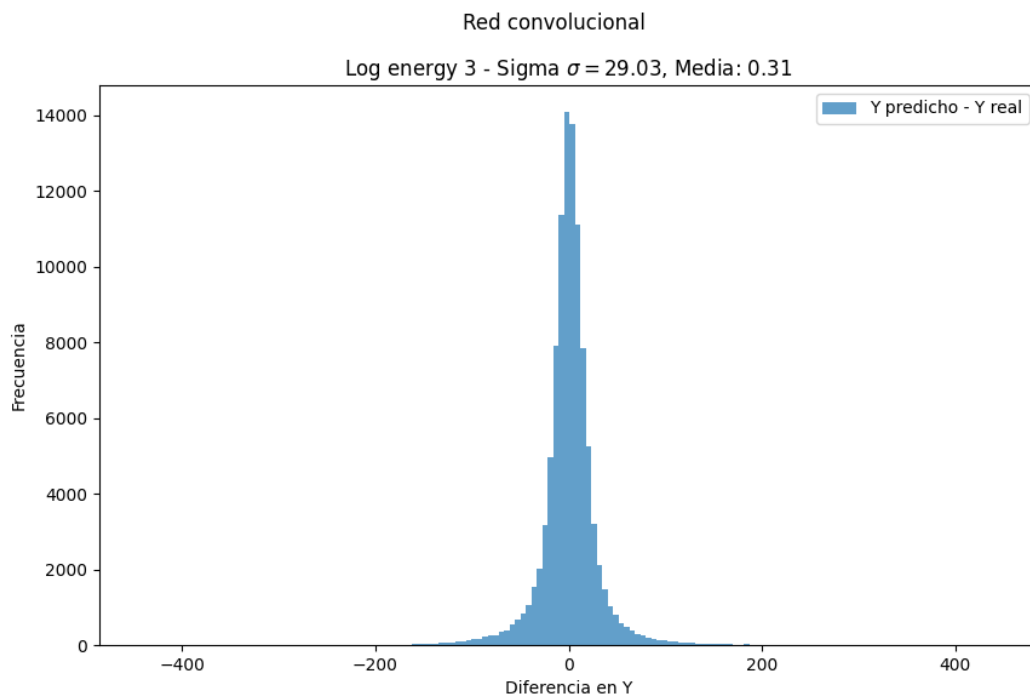


Figura 45 - Histograma diferencias coordenadas X (red convolucional), $4 \leq \log \text{Energy} < 5$

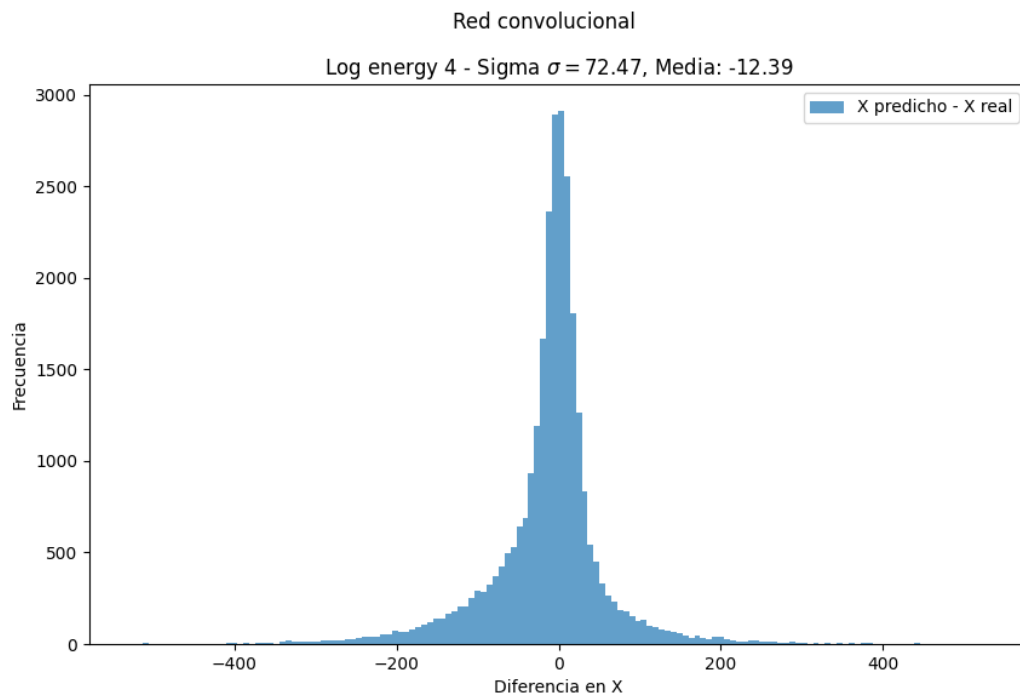
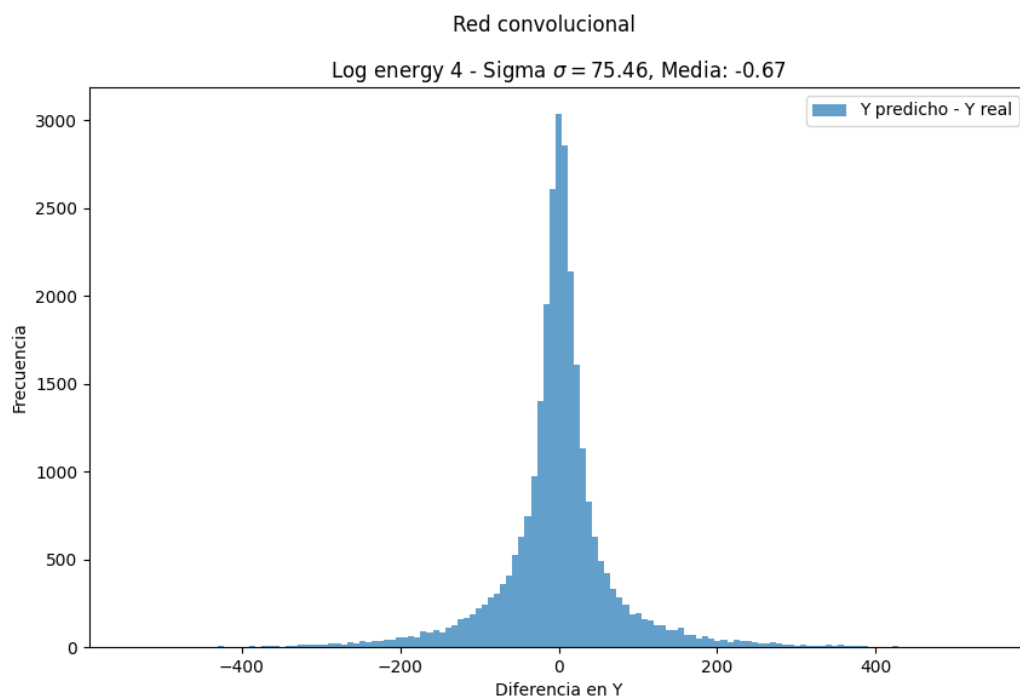


Figura 46 - Histograma diferencias coordenadas Y (red convolucional), $4 \leq \log \text{Energy} < 5$



8.2.HISTOGRAMAS COMPLEMENTARIOS APARTADO 6.2.3

Figura 47 - Histograma diferencias coordenadas X (valores HAWC), $1 \leq \log \text{Energy} < 2$

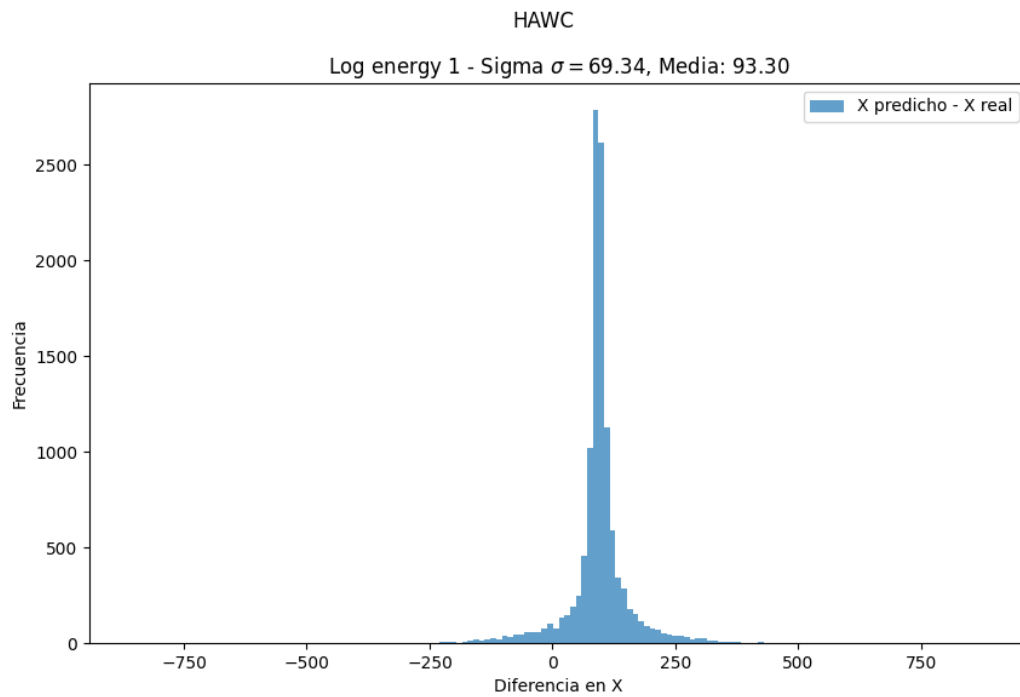


Figura 48 - Histograma diferencias coordenadas Y (valores HAWC), $1 \leq \log \text{Energy} < 2$

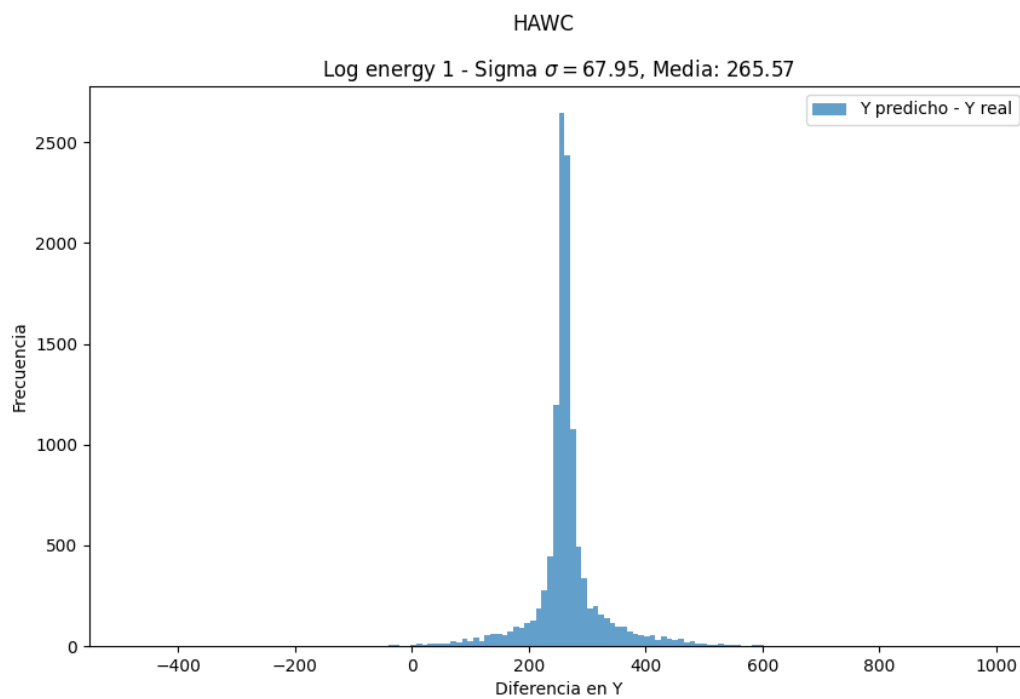


Figura 49 - Histograma diferencias coordenadas X (valores HAWC), $2 \leq \log \text{Energy} < 3$

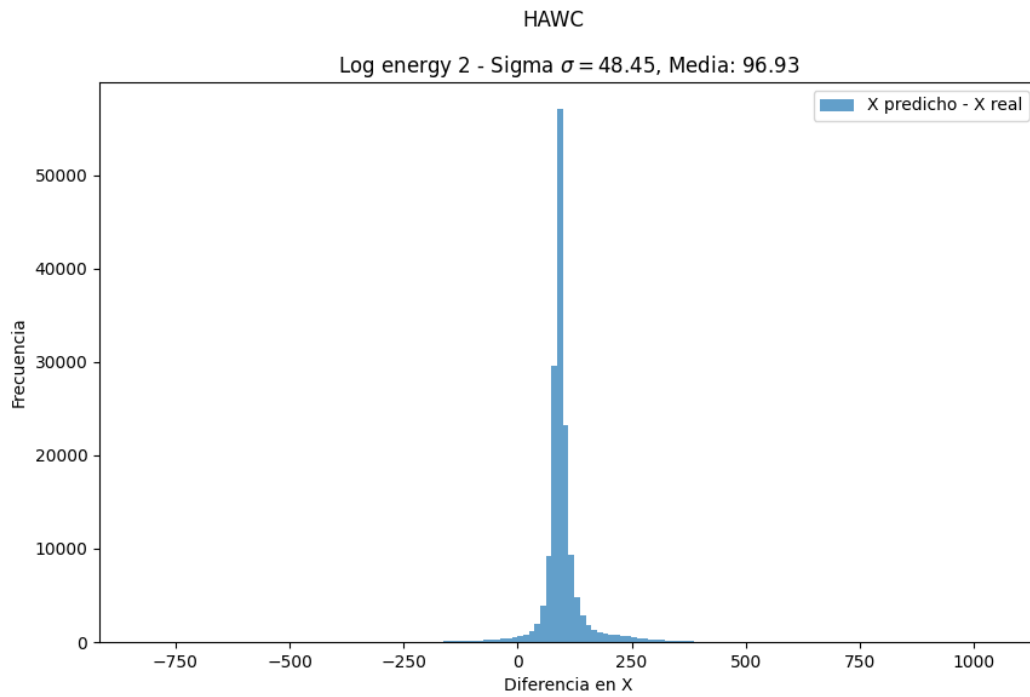


Figura 50 - Histograma diferencias coordenadas Y (valores HAWC), $2 \leq \log \text{Energy} < 3$

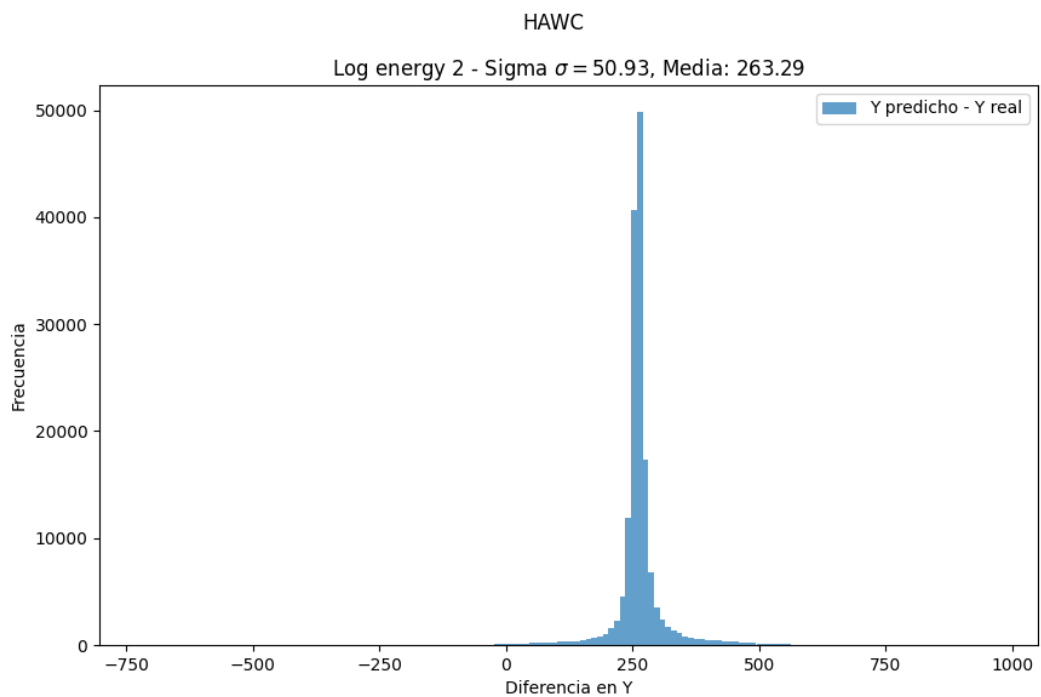


Figura 51 - Histograma diferencias coordenadas X (valores HAWC), $3 \leq \log \text{Energy} < 4$

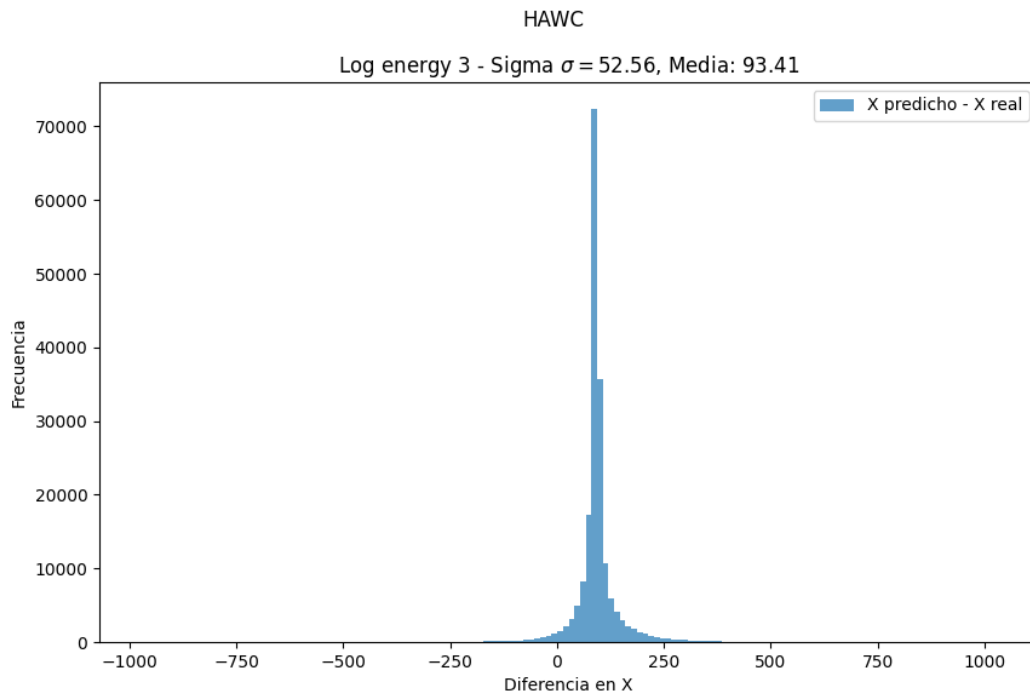


Figura 52 - Histograma diferencias coordenadas Y (valores HAWC), $3 \leq \log \text{Energy} < 4$

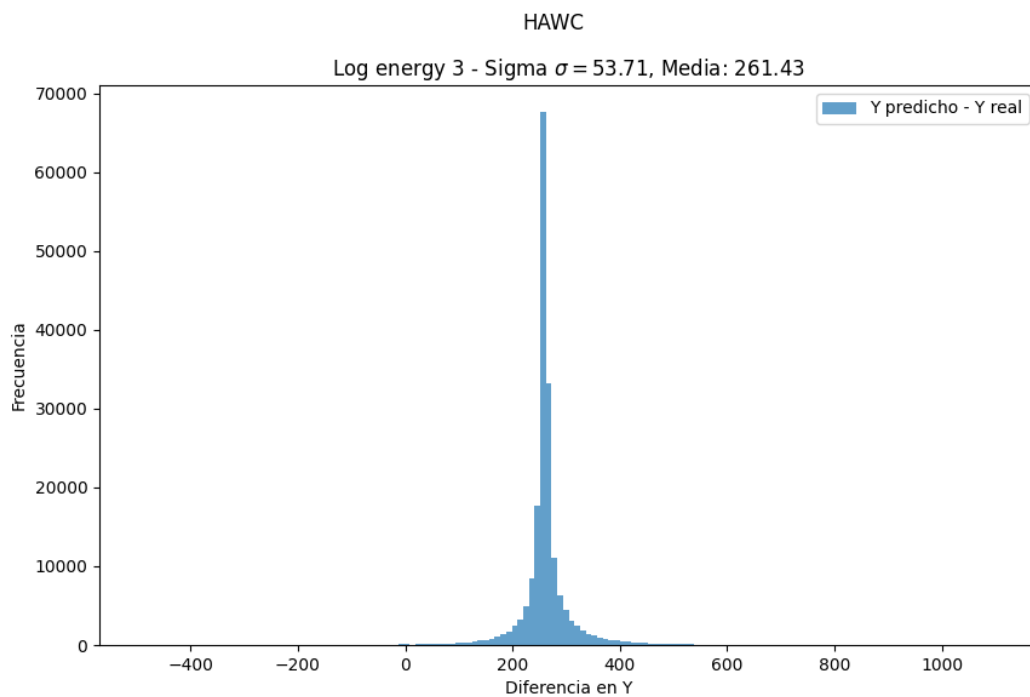


Figura 53 - Histograma diferencias coordenadas X (valores HAWC), $4 \leq \log \text{Energy} < 5$

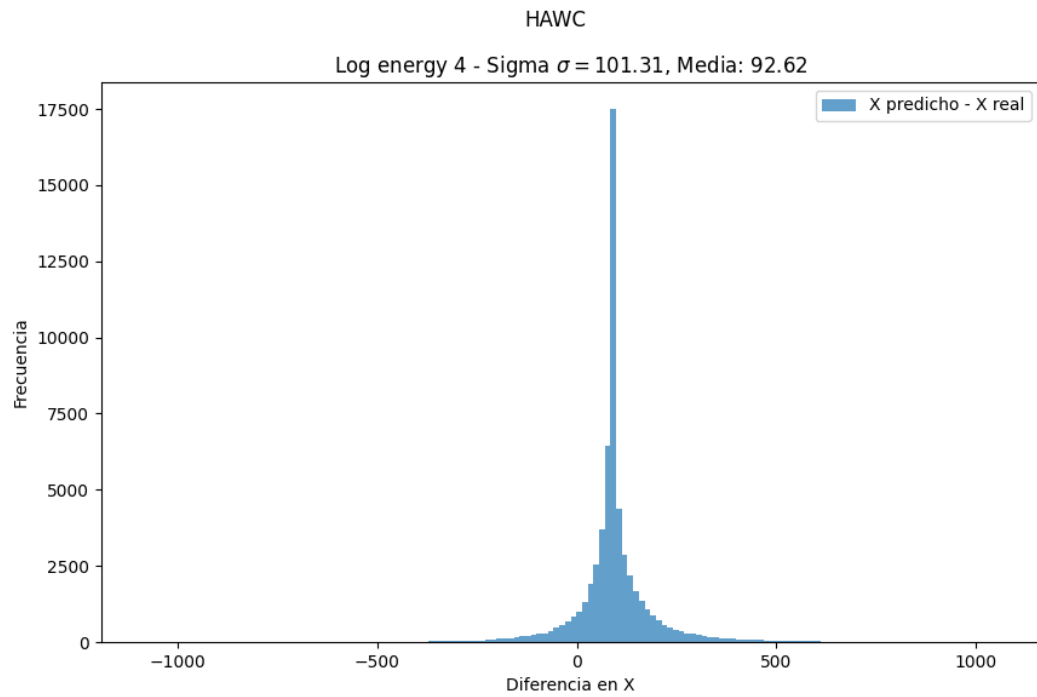
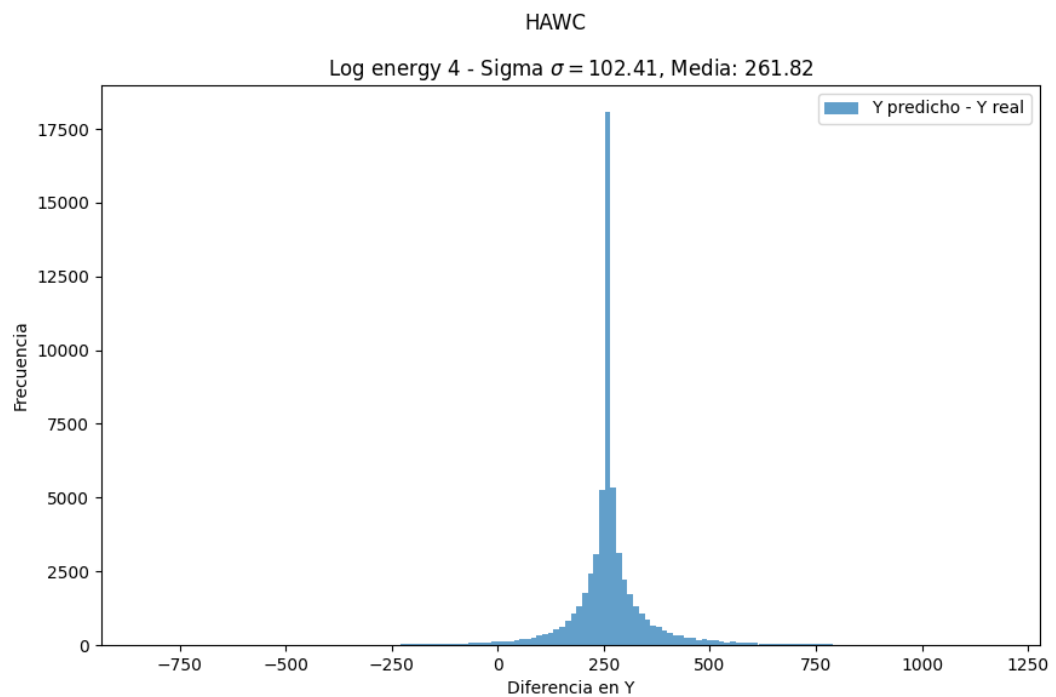


Figura 54 - Histograma diferencias coordenadas Y (valores HAWC), $4 \leq \log \text{Energy} < 5$



9. BIBLIOGRAFÍA

- A. Alvarado, T. C. (2023). Cosmic-ray energy reconstruction using machine learning techniques. *Proceedings of science*, 4.
- Alom, M. Z. (2019). A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics*, 11-12.
- CERN. (2024, Abril). *Cosmic rays: particles from outer space*. Retrieved from CERN | Accelerating science: <https://home.cern/science/physics/cosmic-rays-particles-outer-space#:~:text=He%20had%20discovered%20cosmic%20rays,the%20way%20up%20to%20uranium.>
- Durán, J. (4 de septiembre de 2019). *Todo lo que Necesitas Saber sobre el Descenso del Gradiente Aplicado a Redes Neuronales*. Obtenido de Medium: <https://medium.com/metadatos/todo-lo-que-necesitas-saber-sobre-el-descenso-del-gradiente-aplicado-a-redes-neuronales-19bdbb706a78>
- Einstein, A. (1905). On a heuristic point of view concerning the production and transformation of light.
- Fumo, J. (15 de Junio de 2017). *Types of Machine Learning Algorithms You Should Know*. Obtenido de Towards Data science: <https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861>
- Kathuria, A. (2018). *Intro to Optimization in Deep Learning: Vanishing Gradients and Choosing the Right Activation Function*. Obtenido de Paperspace.com: <https://blog.paperspace.com/vanishing-gradients-activation-function/>
- Loce, R. P. (2017). *Computer vision and imaging in intelligent transportation systems* (Vol. 1).
- National Institutes of Neurological Disorders and Stroke. (s.f.). *Brain Basics: Know Your Brain*. Obtenido de ninds.nih.gov: <https://www.ninds.nih.gov/health-information/public-education/brain-basics/brain-basics-know-your-brain>
- Spiering, C. (2023, Apr 5). *Cherenkov Detectors in Astroparticle Physics*. Retrieved from arxiv.org: <https://ar5iv.labs.arxiv.org/html/2304.02340>

- Tusar, M. T. (14 de diciembre de 2022). *Batch (Offline) Learning and Incremental (Online) Learning in Machine Learning with Implementation*. Obtenido de LinkedIn: <https://www.linkedin.com/pulse/batch-offline-learning-incremental-online-machine-tusar-stmieee/>
- Wang, C.-F. (8 de enero de 2019). *The Vanishing Gradient Problem*. Obtenido de Towards Data Science: <https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484>
- Wang, Y., Cui, Z., & Ke, R. (2023). Chapter 3 - Machine learning basics. En *Machine Learning for Transportation Research and Applications* (págs. 25-40). Elsevier. Obtenido de <https://www.sciencedirect.com/topics/earth-and-planetary-sciences/supervised-learning>
- Yashwanth, N. (7 de 10 de 2020). *Evaluation metrics & Model Selection in Linear Regression*. Obtenido de Towards Data Science: <https://towardsdatascience.com/evaluation-metrics-model-selection-in-linear-regression-73c7573208be>
-