

Deep-learning aplicado a la identificación de muones del experimento CONNIE

Javier Bernal¹, Diego Stalder², Jorge Molina³

¹Autores, ²⁻³Orientadores,

Facultad de Ingeniería

Ingeniería Mecatrónica

Resumen

Durante el experimento CONNIE (Coherent Neutrino-Nucleus Interaction Experiment) fueron colocados 12 sensores CCDs (charge-coupled devices) en las inmediaciones del reactor nuclear Angra dos Reis (Rio de Janeiro, Brasil), con el propósito de detectar los antineutrinos generados en el reactor y testear las interacciones no estándar de los neutrinos a través de la medición de los recoils de baja energía (CE ν NS). Los sensores han registrado imágenes de partículas durante los últimos 2 años, en intervalos de 3 horas, donde la mayoría son muones, producidos en el impacto de los rayos cósmicos con la atmósfera. Este trabajo tiene como objetivo trazar la serie temporal de los muones, para poder estudiar y estimar el background del experimento. Para este fin se elaborará un algoritmo de clasificación de muones basados en conceptos de deep learning, que permitirá diferenciarlo de las otras partículas. Se implementarán redes neuronales convolucionales (CNN) para encontrar un clasificador de mayor eficiencia. El trabajo también implica la creación de un conjunto de datos de entrenamiento.

Palabras Claves: CONNIE, Muones, Redes Neuronales Convoluciones.

1. Introducción

El aumento de la capacidad computacional y el avance en redes durante el siglo XXI han permitido acumular colecciones de datos cada vez mayores, trayendo así avances generalizados en las ciencias naturales y sociales [1] [2]. Esto ha acelerado el desarrollo de nuevas técnicas para analizar los datos y extraer información útil de los mismos. A partir de esto nace el campo de la ciencia de datos.

Utilizando principalmente métodos de análisis tradicionales, la física ha avanzado rápidamente, estableciendo el Modelo Estándar de física de partículas y, más recientemente, su homólogo cosmológico, CDM. Los próximos años traerán cantidades sin precedentes de datos y complejidad en el Gran Colisionador de Hadrones (LHC), acelerando los protones en el CERN, así como en la frontera de intensidad y en otros lugares. Extraer la física subyacente de la misma manera se vuelve cada vez más desafiante, o simplemente imposible de manera oportuna. Eso explica la reciente chispa

de interés en el aprendizaje automático [3].

Las ciencias físicas se encuentran en una posición única. Mientras que en muchos otros campos existen fundamentos o modelos teóricos menos firmes, los físicos tienen métodos bien establecidos para predecir y comparar los resultados de los experimentos con los cálculos teóricos, como atestiguan los numerosos éxitos del Modelo Estándar. Esto significa que se pueden desarrollar y aplicar métodos de aprendizaje automático motivados por la física, acelerando el proceso de aprendizaje y haciéndolo más eficiente y preciso. Al mismo tiempo, los impresionantes avances en la ciencia de datos y la tecnología informática ayudarán a abordar los próximos desafíos en la física de partículas.

En el experimento CONNIE, el cual busca detectar principalmente antineutrinos, se han detectado además, dentro del ruido de fondo, unas partículas de interés: los muones. Debido a la enorme cantidad de estas partículas detectadas, sería imposible hacer un conteo manual de las mismas, por este motivo este trabajo final de grado propone aplicar el aprendizaje profundo a la identificación de muones en datos del experimento CONNIE.

2. Objetivos

Objetivo General

Estudiar el comportamiento de los muones en función del tiempo, utilizando técnicas de aprendizaje automatizado, basándose en los resultados del Coherent Neutrino-Nucleus Interaction Experiment (CONNIE).

Objetivos Específicos

- Analizar y comparar los criterios físicos utilizados para la identificación de muones.
- Generar el conjunto de datos de entrenamiento necesario para la identificación de muones.
- Aplicar algoritmos de aprendizaje automático para mejorar la identificación.
- Evaluar la implementación de un algoritmo basado en el aprendizaje profundo para la identificación automática de muones en las imágenes de los CCDs.
- Con el mejor clasificador, generar la serie temporal del flujo de muones.

- Analizar las series temporales del flujo de muones.

3. CONNIE

El experimento CONNIE (llamado así por sus siglas en inglés, Coherent Neutrino-Nucleus Interaction Experiment) es desarrollado en colaboración con más de 20 científicos de Argentina, Brasil, México, Suiza y USA, del cuál la Facultad de Ingeniería de la Universidad Nacional de Asunción (FIUNA) participa como miembro fundador.

El experimento busca detectar los antineutrinos generados en el reactor nuclear de Angra 2, situado en la planta Almirante Alvaro Alberto en Angra dos Reis, en el estado de Rio de Janeiro, Brasil. El sistema experimental utiliza dispositivos de carga acoplada (en inglés charge-coupled devices, conocido también como CCDs) para medir los recoils (retrocesos) de baja energía de la CEvNS (por sus siglas en inglés, Coherent Elastic Neutrino Nucleus Scattering) y estudiar las interacciones de los neutrinos con los núcleos.

Con el fin de detectar los antineutrinos, el experimento CONNIE se instaló en el año 2016 un sistema de detección a 30 m del núcleo del reactor, lo que permitió tomar datos por mas de 3 años. Debido a que se intentan encontrar partículas de muy baja energía, el sistema de lectura de los detectores demoran un tiempo de 40 minutos aproximadamente para extraer la información, lo que puede considerarse que es un tiempo suficiente para mitigar el ruido intrínseco del sistema de lectura. La extracción de la información al principio se realizaba cada 3 horas al inicio del experimento, tiempo que luego fue cambiado a 2 horas debido a la gran cantidad de ruido radiactivo ambiental detectado.

A este ruido radiactivo se les denomina background, y son partículas del ambiente que impactan en el detector, y que, por lo tanto, confundirán la detección del antineutrino. Las principales fuentes de estas partículas no deseadas son la radiactividad ambiental presente en todos los lugares de la tierra, por lo que pueden llegar a los CCDs desde el exterior del castillo donde se colocan los detectores, así como de los propios materiales que forman al sistema experimental. También las partículas del background provienen de los rayos cósmicos que llegan a todos los lugares del planeta, entre las que se incluyen los muones (partículas μ), que poseen energía mucho mayor que la de los neutrinos [4].

3.1. Detector

El experimento CONNIE consta de 14 dispositivos de carga acoplada (CCDs por sus siglas en inglés) superpuestos dentro de una caja de cobre puro, creado específicamente para blindar a los detectores de los fotones y de los rayos gamma de bajas energías. En CONNIE se utiliza los CCDs para medir partículas y no fotones por lo que se necesita que el espesor sea lo mayor posible, ya que una mayor masa implica una mayor probabilidad de interacción entre los antineutrinos y los núcleos del detector de Silicio.

Cada sensor consiste en un cuadrado de 16 millones de píxeles, de $15\ \mu\text{m} \times 15\ \mu\text{m}$ cada píxel, generando

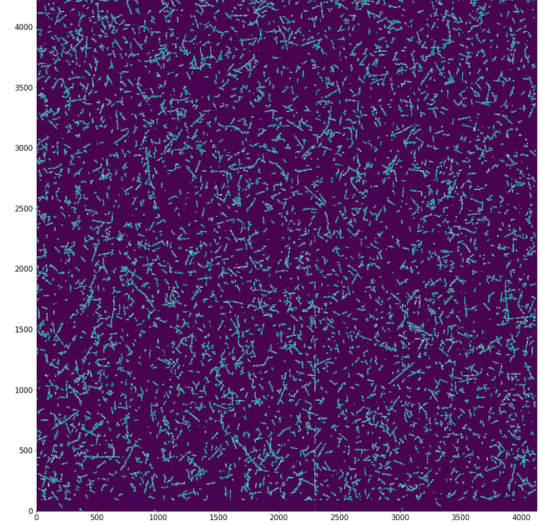


Figura 1: Imagen digital extraída del archivo que corresponde al runID: 35 del Ohdu: 6 . **Fuente:** Elaboración Propia.

así una imagen digital de 4 mil píxeles de largo por 4 mil píxeles de ancho, donde se ven los trazos marcados por las partículas que impactaron al detector durante el tiempo de exposición en que el sensor estuvo registrando los datos. Posteriormente, el detector es leído pixel por pixel a través de la técnica de Correlated Double Sampling, que permite guardar la información en registros seriales acoplados a los bordes del CCD. El formato de almacenamiento son .fits, que son muy utilizados en astronomía.

3.2. Muones

El muon (μ) es una partícula elemental que posee carga negativa y su masa es aproximadamente 200 veces la del electrón, su espín es $\frac{1}{2}$ y tiene una vida media de $2,2\ \mu\text{s}$ [5]. Son el resultado de la colisión de los protones provenientes de los rayos cósmicos con los átomos presentes en la atmósfera terrestre, y de acuerdo a su energía pueden penetrar hasta varias centenas de metros dentro de la tierra. La energía media de los muones sobre la superficie terrestre es de 4 GeV según las mediciones realizadas por distintos experimentos

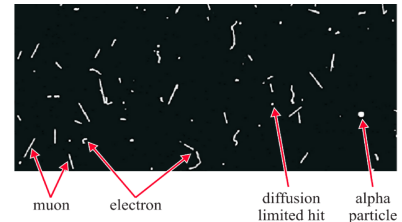


Figura 2: Imagen digital extraída del archivo que corresponde al runID: 35 del Ohdu: 6 . **Fuente:** Extraído de [4]

A esas altas energías los muones no se desvían al chocar con los átomos de silicio, pero si ceden un poco de su momento para liberar a los electrones de la red

cristalina del detector, que son los que dan la información de que la partícula paso por allí. De esta manera las imágenes formadas en los detectores mostraran una línea recta, cuya longitud dependerá del ángulo de penetración. Cada partícula determina una traza distinta, cómo se muestra en la figura 2, determinando un background para el experimento CONNIE.

4. Redes Neuronales Convolucionales

Las redes convolucionales son utilizadas mayormente en imágenes, debido como trabajan las capas convolucionales. Estas capas por lo general tienen entradas de 3 dimensiones: ancho, alto y profundidad. En estas redes las neuronas de una capa no necesitan conectarse con todas las neuronas de la capa siguiente, solamente se conectan por áreas o regiones. El hecho que se conecten por regiones también insta a la utilización en imágenes, porque así la capa no tendrá que calcular todos los valores de los píxeles y estimar una cantidad innecesaria de parámetros [6] [7].

Las capas utilizadas en una red convolucional son particulares y distintas a las que se utilizan en una red convencional. Hay que tener en cuenta que al ser imágenes las entradas de una red convolucional, las transformaciones realizadas entre capas son distintas, y cada una entre como resultado una matriz de 3 dimensiones. Las capas mas utilizadas en las redes convolucionales sencillas son:

- **Capa de convolución:** La capa de convolución es sin duda la capa mas importante de las redes neuronales convolucionales. cada neurona esta conectada solo a los píxeles de su campo receptivo de la entrada. Esto quiere decir que una neurona colocada en la fila i y en la columna j esta conectada a la salida de de la capa anterior que corresponden a la posición de las filas i hasta $i + f_h - 1$ y de las columnas j hasta $j + f_w - 1$, donde f_h y f_w son el alto y el ancho del respectivo campo. Para que la salida de la red tenga la misma cantidad de elementos es normal rellenar los extremos con 0, esto se llama «zero padding».

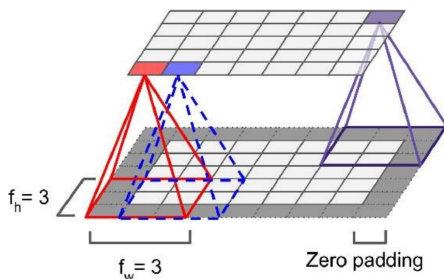


Figura 3: Capa Convolutiva. **Fuente:** [2].

En el caso de las redes convolucionales, los pesos van a estar representados por una pequeña matriz, que será el filtro que se le aplique a la imagen, y la recorrerá obteniendo las características para lo

cual fue entrenado. Este filtro se llama kernel de convolución. Cuando se habla de profundidad en esta capa uno se refiere a la cantidad de kernels de convolución que va a utilizar para sacar los distintos mapas de parámetros y cuando se habla de pasos se refiere a la cantidad de elemento que va a saltar el kernel al hacer la convolución.

El volumen de salida de la capa de convolución se puede calcular con la formula:

$$W_0 = \frac{W - F + 2P}{S + 1} \quad (1)$$

Donde W es el volumen de entrada, F es el tamaño del kernel, P el zero-padding y S el paso.

- **Capa de pooling o de reducción:** Tiene como objetivo reducir la carga de computacional, el uso de memoria, y la cantidad de parámetros de la red. Reduce el tamaño de la imagen sin perder la información, también sirve para generalizar los datos con los que trabaja la red, por lo cual ayuda a reducir el overfitting, debido a que luego de esta capa la información va a estar un poco cambiada.

Como en la capa de convolución en esta las neuronas no van a estar conectadas a todas las neuronas de la capa anterior, solo a una región cuyo tamaño esta previamente definido. La diferencia con la capa anterior es que en esta capa no hay pesos, si no que saca un resultado directo de la región donde se hace el pooling, o sea que se puede usar cualquier función como «max» o «mean» dependiendo de que tipo de pooling se desea hacer. La capa de pooling mas común es la max-pooling layer (capa de máximo pooling), donde la entrada de la neurona que corresponde a una región obtiene el valor máximo encontrado en esa región.

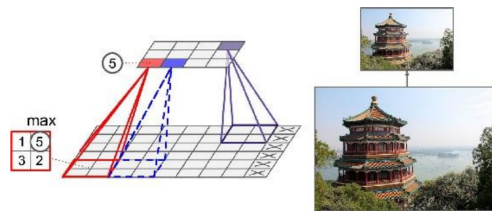


Figura 4: Capa de pooling. **Fuente:** [2].

- **Capa completamente conectada:** Es en la que todas las neuronas de una capa están completamente conectadas a las neuronas de la siguiente capa, en esta capa los pesos son valores numéricos normales y es la capa mas parecida a las capas convencionales. Para que las salidas de la etapa convolución puedan ser procesadas por estas capas, la matriz de entrada de la etapa completamente conectada debe ser sometida a un aplanamiento, donde todas las filas se concatenan generando un solo vector que contiene la información de la matriz. Es en esta etapa donde se realiza la clasificación [8].

5. Arquitectura VGG16

La arquitectura VGG16 es una de las mas populares de los últimos años, debido a sus altos desempeños clasificando imágenes, el nombre proviene de «visual geometry group» y fue desarrollada por Karen Simonyan y Adrew Zisserman en la universidad de Oxford en el año 2014.

La arquitectura fue desarrollada primero para imágenes de 224x224 RGB. La imagen pasa por una grupo de capas convolucionales, donde los filtros están predefinidos para una región pequeña de 3x3 (este es el tamaño mínimo de región para encontrar patrones en la imagen). Luego se aplica una capa de max-pooling y se vuelve a poner una etapa convolucional buscando el doble de parámetros de la etapa anterior. Se continúa así hasta que la salida de la capa es una matriz de 7x7x512 y ahí se aplica el aplanado para entrar a una etapa completamente conectada de 3 capas. La salida tendrá tantas neuronas como clases se quieran identificar y se utilizará la función de activación softmax para esta capa [9].

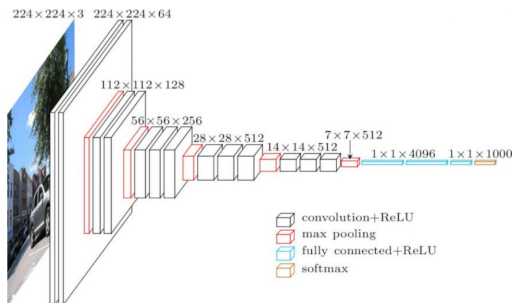


Figura 5: Arquitectura VGG16. **Fuente:** Adaptado de [9].

6. Dataset

Durante la generación de las imágenes para el dataset se exportaron las variables del archivo root a una estructura de datos mas simple que se puedan filtrar y leer desde python 3 con la librería pandas. El cambio de estructura de datos se realizó gracias a la biblioteca root_pandas. Luego se extrajeron las variables conformadas por los vectores coordenados xPix y yPix, para generar el mapa de puntos al cuál a cada coordenada de X e Y se le va a asignar el valor almacenado en ePix.

Se generaron 3171 imágenes de eventos completamente aleatorios tomadas de todas las épocas del experimento. Estas imágenes fueron almacenadas en formato .fits para la lectura rápida y posterior clasificación.

La clasificación del dataset consiste en la observación de las 3171 imágenes. Es necesario analizar los tipos de clases que se encuentran en la observación previa, para luego una clasificación que resultará en el etiquetado de estas imágenes para ser utilizadas como entrada a la red neuronal posteriormente.

Las clases encontradas durante las primeras observaciones fueron:

- Puntos: Partículas de bajo valor energético que generan un punto en el CCD.

- Falla de píxeles: Píxeles saturados por fallas del artefacto.
- Indefinido: Valores indefinidos, en su mayoría ruido.
- Gusanos: Partículas que se desvían al interactuar con el silicio del CCD, en su mayoría electrones.
- Errores de segmentación: Son trazas de partículas cruzadas debido al tiempo de exposición del CCD.
- Muones inseguros: Partículas que se asemejan a muones, pero no hay evidencia suficiente para concluir que lo son, posiblemente son muones que ingresaron en un ángulo muy vertical.
- Muones seguros: Son partículas con sus trazas rectas y alargadas, por lo cual se esta seguro que son muones.

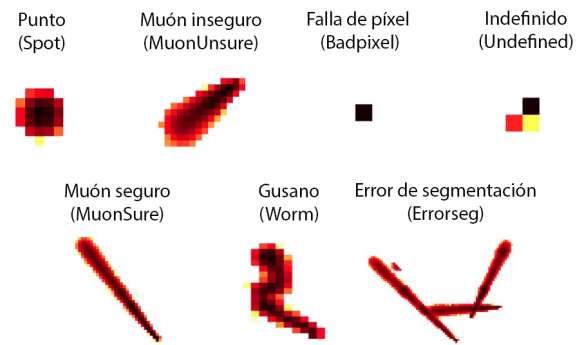


Figura 6: Clases observadas en el dataset. **Fuente:** Elaboración Propia

En la figura 7 se muestra la distribución de los datos que fueron clasificados por dos colaboradores del CONNIE, basándose en los criterios morfológicos para distinguir entre partículas.

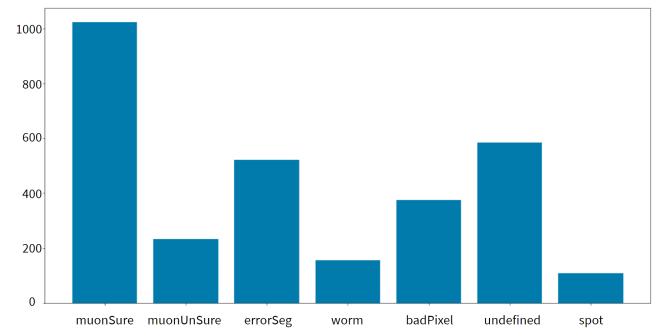


Figura 7: Distribución de los datos en crudo. **Fuente:**Elaboración Propia

- 1023 muones seguros.
- 233 muones no seguros.
- 522 errores de segmentación.
- 156 worms.
- 375 píxeles dañados.
- 585 elementos indefinidos.
- 108 puntos.

7. Preprocesamiento y aumento de datos

Las imágenes obtenidas del experimento vienen en un tamaño determinado por el mínimo rectángulo capaz de contener a un evento, todas las imágenes fueron extraídas en formatos complementemente aleatorios y distintos entre sí. Es por esto que se debe realizar un preprocesamiento a las imágenes clasificadas, de modo a que las imágenes queden adaptadas al formato de entrada de la red.

Las imágenes se ajustaron a los formatos de entrada tomando el lado mas largo de una imagen (puede ser el ancho o el largo) y se ajustó al de la imagen cuadrada, el resto se relleno con ceros para obtener una imagen cuadrada. Se realizaron normalizados de modo que las imágenes no tengan valores negativos y que queden ajustadas entre 0 y 255 de intensidad, con fines experimentales las imágenes estaban en escala normal como logarítmica, dependiendo de la red que se utilizaría.

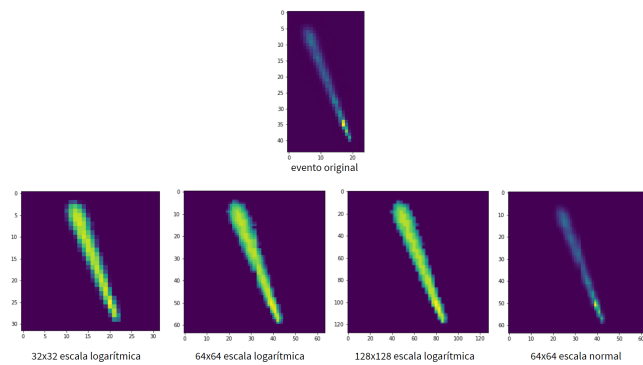


Figura 8: Resultados de los distintos preprocesamientos. **Fuente:** Elaboración Propia

Desabalance en los datos se refiere a que en una clase se observaron considerablemente mas muestras que en otra, lo cual puede producir que la red entre en overfitting, alterando así la capacidad de clasificación de las imágenes. Ya que los eventos pueden ser tomados en cualquier dirección (porque se esta analizando la naturaleza de la partícula solo por su forma e intensidad), los dos métodos utilizados para el aumento de datos fueron las rotaciones y los espejados. Produciendo datasets considerablemente mejor distribuidos como se muestran en las imágenes 9 y 10

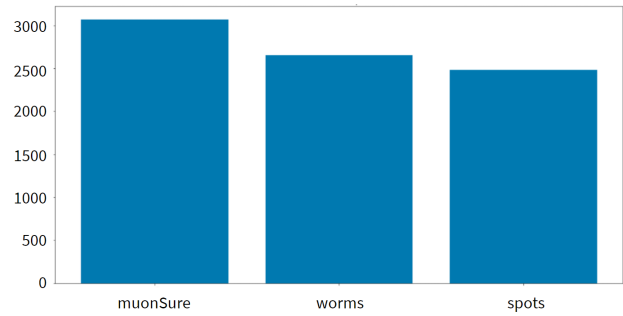


Figura 9: Distribución de los datos luego del aumento de datos para la red de 3 salidas. **Fuente:** Elaboración Propia

Los resultados de la distribución del aumento para el conjunto de datos de la red que tendrá 3 salidas fueron:

- 3069 muones seguros.
- 2652 worms.
- 2484 puntos.

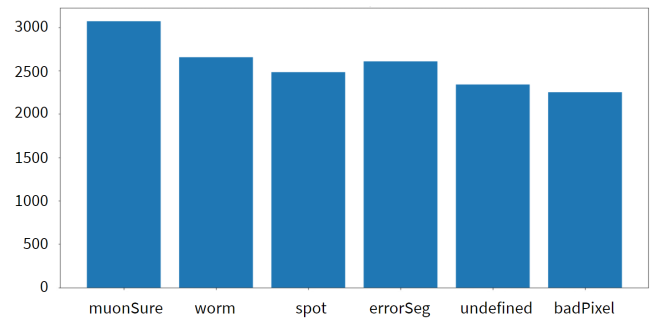


Figura 10: Distribución de los datos luego del aumento de datos para la red de 6 salidas. **Fuente:** Elaboración Propia

Los resultados de la distribución del aumento para el conjunto de datos de la red que tendrá 6 salidas fueron:

- 3069 muones seguros.
- 2652 worms.
- 2484 puntos.
- 2610 errores de segmentación.
- 2340 elementos indefinidos.
- 2250 píxeles dañados.

8. Modelos propuestos

Los modelos propuestos fueron todos basados en VGG16, difiere la entrada y el escalado de las imágenes y por lo tanto la profundidad, también difiere en que las imágenes de entrada son todas de un solo color. Los modelos cuentan con las siguientes capas:

- conv: capa de convolución 2D.
- DO: dropout.

- Act: capa de activación.
- Pool: capa de max-pooling.
- Dense: capa completamente conectada.
- Flat: capa de aplanamiento.

Todos los modelos fueron utilizados con optimizador adam y con la función softmax para la determinación de la clase.

En la tabla 1 se muestran los distintos modelos utilizados

Cuadro 1: Modelos DL utilizados

Entrada	Profundidad	Escalado	Padding	Número de salidas
32x32	18	logarítmico	si	3
64x64	22	logarítmico	no	3
64x64	22	logarítmico	si	3
64x64	22	normal	si	3
64x64	22	normal	si	6
128x128	26	logarítmico	si	3

9. Métricas utilizadas

9.1. Matriz de confusión

Para una mejor comprensión de las pruebas y resultados, se presenta previamente una descripción de las métricas utilizadas para analizar el rendimiento de los algoritmos de clasificación utilizados.

		PREDICHO		
		Clase A	Clase B	Clase C
R E A L	Clase A	5	2	0
	Clase B	3	3	2
	Clase C	0	1	11

Figura 11: Ejemplo de matriz de confusión.

9.1.1. Exactitud

La exactitud (ACC del inglés *Accuracy*) es una métrica que nos indica el porcentaje de predicciones correctas sobre el total de datos.

$$ACC = \frac{\sum \text{Predicciones Verdaderas}}{\sum \text{Predicciones totales}} \quad (2)$$

En el ejemplo de la figura 11 se tiene:

$$AC = \frac{5 + 3 + 11}{5 + 2 + 0 + 3 + 3 + 2 + 0 + 1 + 11} = 70,37 \%$$

9.1.2. Sensibilidad

La sensibilidad o razón de verdadero positivo (TPR del inglés *True positive rate*) es la proporción de casos positivos que se identificaron correctamente en una clase sobre el total de verdaderos de esa clase.

$$TPR = \frac{\text{Predicciones Verdaderas}}{\text{Total verdaderos}} \quad (3)$$

En el ejemplo de la figura 11 se tiene para la clase A:

$$TPR = \frac{5}{5 + 2 + 0} = \frac{5}{7} = 71,43 \%$$

9.1.3. Precisión

La precisión (P) es la proporción de los casos positivos que se identificaron correctamente sobre el total de predicciones de esa clase.

$$P = \frac{\text{Predicciones Verdaderas}}{\text{Predicciones totales}} \quad (4)$$

En el ejemplo de la figura 11 se tiene para la clase A:

$$P = \frac{5}{5 + 3 + 0} = \frac{5}{8} = 62,5 \%$$

9.2. Periodograma de Lomb-Scargle

El periodograma de Lomb-Scargle es un algoritmo que permite calcular el comportamiento de una serie temporal discreta en el espectro de la frecuencia o del periodo. Este método es una extensión del periodograma convencional especializado para tiempos de muestreo no regulares [10].

Para comprender el periodograma de Lomb-Scargle es necesario mostrar como se relaciona con la transformada de Fourier y con el periodograma convencional. La transformada discreta de Fourier esta dada por la siguiente ecuación:

$$\hat{g}_k = \sum_{n=0}^N g_n e^{-\frac{2\pi i}{N} kn} \quad (5)$$

Donde k tiene valores enteros de 0 a N-1.

Una vez definida la transformada discreta de Fourier en la ecuación 5 se le puede aplicar la definición de espectro de intensidad para computar el periodograma convencional, resultando en la siguiente ecuación:

$$P_s(f) = \frac{1}{N} \left| \sum_{n=1}^N g_n e^{-2\pi i f t_n} \right|^2 \quad (6)$$

Para la definición del periodograma de Lomb-Scargle primero debemos expresar en su forma trigonométrica la ecuación 9.2 que corresponde al periodograma convencional

$$P(f) = \frac{1}{N} \left| \sum_{n=1}^N g_n e^{-2\pi i f t_n} \right|^2 = \frac{1}{N} \left[\left(\sum_n g_n \cos 2\pi f t_n \right)^2 + \left(\sum_n g_n \sin 2\pi f t_n \right)^2 \right]$$

Para una forma mas generalizada, Scargle representó el periodograma de la siguiente forma:

$$P(f) = \frac{A^2}{2} \left(\sum_n g_n \cos 2\pi f [t_n - \tau] \right)^2 + \frac{B^2}{2} \left(\sum_n g_n \sin 2\pi f [t_n - \tau] \right)^2$$

Donde A, B y τ son funciones arbitrarias de la frecuencia f y los tiempos de observación t_i . Y demostró que hay una sola forma de encontrar A, B y τ tal que:

- El periodograma se reduzca a la forma clásica en el caso de una frecuencia de muestreo uniformemente separada.
- La distribución del periodograma sea analíticamente computable.
- El periodograma sea sensible a cambios globales de tiempo en los datos.

Teniendo en cuenta estas condiciones el periodograma queda de la siguiente forma:

$$P_{LS}(f) = \frac{1}{2} \left\{ \frac{(\sum_n g_n \cos 2\pi f [t_n - \tau])^2}{\sum_n \cos^2(2\pi f [t_n - \tau])} + \frac{(\sum_n g_n \sin 2\pi f [t_n - \tau])^2}{\sum_n \sin^2(2\pi f [t_n - \tau])} \right\}$$

donde τ es específico para cada frecuencia y así asegurar cada variación de tiempo de muestreo:

$$\tau = \frac{1}{4\pi f} \tan^{-1} \left(\frac{\sum_n \sin 4\pi f t_n}{\sum_n \cos 4\pi f t_n} \right)$$

10. Resultados

10.1. Resultados de los entrenamientos de los algoritmos de deep-learning

10.1.1. resultados del entrenamiento de entrada de 32x32

En la imagen 12 se exponen los resultados de la evaluación del entrenamiento para la red que tenía una entrada de 32x32 píxeles, con un escalado logarítmico y utilizando el padding.

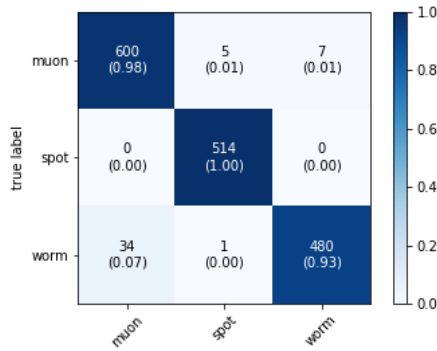


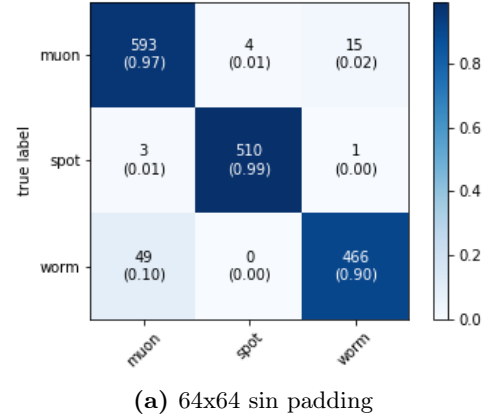
Figura 12: Matriz de confusión de la red con entrada de 32x32. **Fuente:** Elaboración Propia

Basándose en las métricas mostradas en la sección ?? esta red tiene una exactitud del **97.2 %**, una sensibilidad de **98 %** para los muones y una precisión de **94.6 %** para muones.

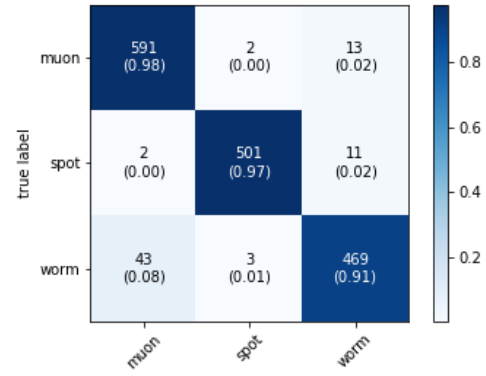
10.1.2. Resultados del entrenamiento de 64x64

A diferencia de la red con entrada de 32x32, para la red de entrada de 64x64 se utilizaron distintos preprocesamientos distintos, esto se muestra en la tabla ?? del capítulo ??.

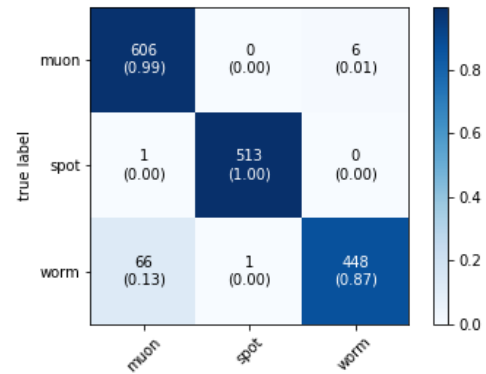
Todas las matrices mostradas en la imagen 13 se obtuvieron del entrenamiento con el conjunto de datos cuya distribución se muestra en la imagen ???. Por otro lado, la matriz de confusión mostrada en la imagen 14 utiliza la distribución mostrada en el gráfico ?? debido a la cantidad de salidas que tiene.



(a) 64x64 sin padding



(b) 64x64 con padding.



(c) Escala normal

Figura 13: Matrices de confusión de la clasificación de 3 clases con entradas de 64x64 píxeles. **Fuente:** Elaboración Propia

Para la red con entrada de 64x64 en escala logarítmica y entrenamiento sin padding, mostrada en la matriz 13a, se tiene una exactitud del **96.0 %**, una sensibilidad de **96.9 %** para los muones y una precisión de **91.9 %** para muones. La red de entrada de 64x64 en escala logarítmica cuyo entrenamiento si tuvo padding

dio resultados de exactitud del **95.4 %**, con una sensibilidad y precisión para muones de **97.5 %** y **92.9 %** respectivamente. Mientras que para la red que fue entrenada con una escala no logarítmica y con padding demostró una exactitud del **94.6 %**, con una sensibilidad y precisión del **99.0 %** y **99.0 %** **90.0 %**.

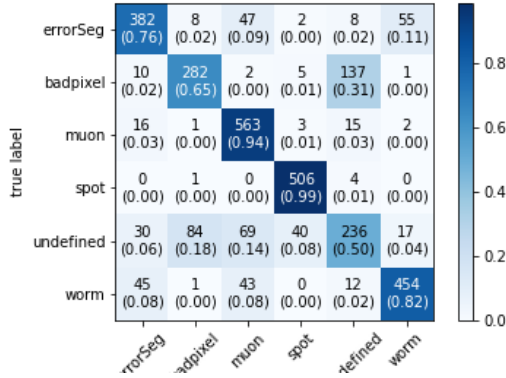


Figura 14: Matriz de confusión de la red con entrada de 64x64 y de 6 salidas.

Para el caso de la figura 14 se utilizó el conjunto de datos cuya distribución se muestra en la imagen ?? debido a que esta red tenía 6 salidas. El entrenamiento de esta red muestra una exactitud de **79.0 %**, una sensibilidad de **93.8 %** para los muones y una precisión del **78.0 %**.

Sí bien los resultados de las métricas de esta red no son tan significativos como los de las redes de 3 salidas, esta red aparente tener un buen desempeño, debido a que esta entrenada para los casos reales del CONNIE, donde los grupos no están divididos en solo 3 clases.

10.1.3. Resultados del entrenamiento de 128x128

La red entrenada con imágenes de 128x128 píxeles utilizó el mismo conjunto de datos que las 3 primeros de 64x64 píxeles y que la de 32x32. En la figura 15 se ve la matriz de confusión de este entrenamiento.

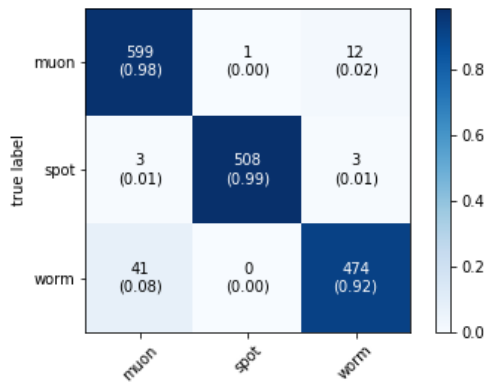


Figura 15: Matriz de confusión de la red con entrada de 128x128. **Fuente:** Elaboración Propia

En este caso la exactitud fue del **96.3 %**, la sensibilidad y precisión para los muones fue del **97.8 %** y

93.1 % respectivamente. Cabe destacar que el tiempo utilizado para el entrenamiento de esta red fue casi el doble de los entrenamientos anteriores.

10.1.4. Resultados de la clasificación por deep-learning

Los muones clasificados se muestran en la imagen 16 en el runID habían 4265 eventos de los cuales 2047 fueron clasificados como muones por el algoritmo

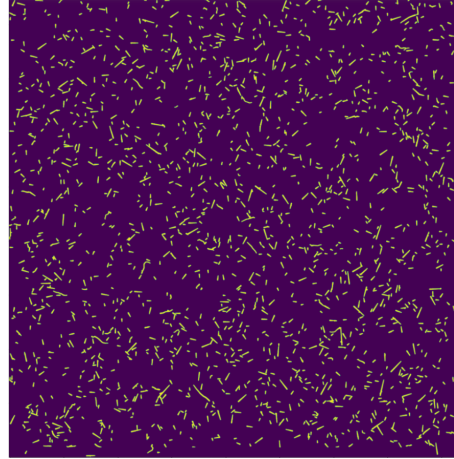


Figura 16: muones del runID 246. **Fuente:** Elaboración Propia

Si bien la intención del algoritmo es detectar los muones, este también fue capaz de detectar partículas que dejan trazos como gusanos o como puntos. Debido a la red de 6 salidas, también se pudo determinar la información que no puede ser utilizada, debido a que son problemas de segmentación o ruido. En la imagen ?? se muestran los eventos clasificados por el algoritmo que no fueron muones, de los cuales 329 tenían trazos de gusanos y 327 con forma de puntos, la imagen 19 muestra la información contenida en errores de segmentación y ruido.

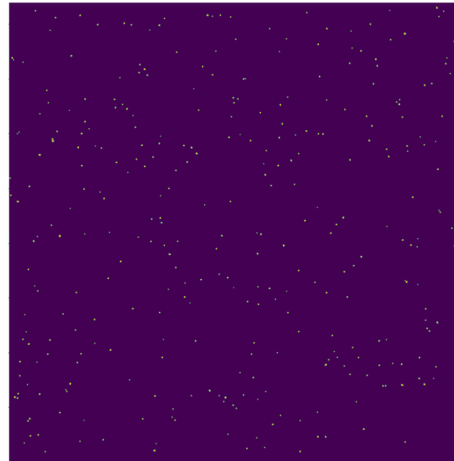


Figura 17: puntos del runID 246. **Fuente:** Elaboración Propia

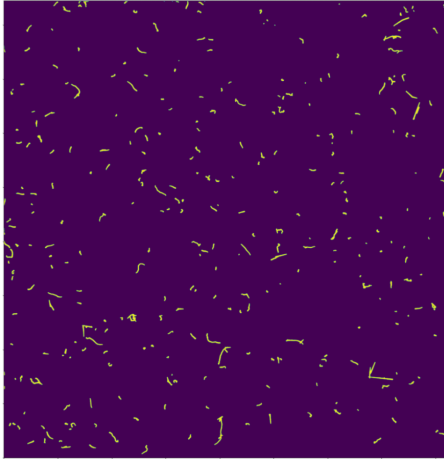


Figura 18: gusanos del runID 246. **Fuente:** Elaboración Propia

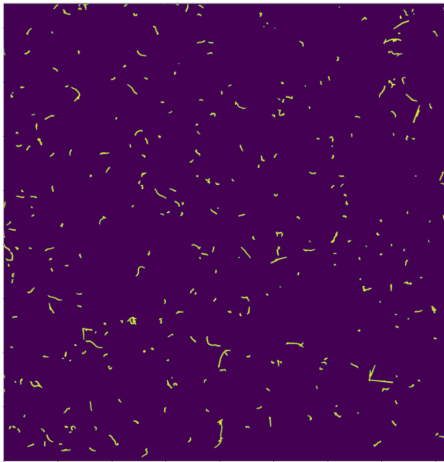


Figura 19: otros eventos del runID 246. **Fuente:** Elaboración Propia

10.2. resultados de la serie temporal y análisis usando el periodograma de Lomb-Scargle

Para la elaboración de la serie temporal de los muones fue utilizada la red neuronal con 6 salidas, mostrada en 14, debido a que esta tenía un mejor desempeño con los datos que no eran worms, muones o puntos. Todos los datos fueron analizados sobre el ohdu2.

La serie temporal muestra el conteo de muones realizado por la red, tomando datos desde el 15 de agosto del año 2016, hasta el 23 de marzo del 2019. Los cortes en los gráficos son debido a problemas durante la realización del experimento en los que el detector quedó sin operar durante esos plazos de tiempo.

Para el análisis de la periodicidad se aplicó el periodograma de Long-Scargle, ya que es mas eficiente que la transformada de Fourier al tener periodos de muestreo irregulares. El periodograma se aplicó a los tramos 2, 3, 4 y 6, debido a que son los que tenían mayor concentración de puntos.

Como las hipótesis indican que el comportamiento periódico es diario, se tomó un rango de análisis del periodograma donde el mayor periodo capaz de ser ana-

lizado es el de una semana. El límite inferior es de 9 horas, teniendo en cuenta que el periodo de muestreo es cada 3 horas y satisfaciendo el teorema de Nyquist.

10.2.1. Tramo 1

En la figura 20 se muestran los resultados del primer tramo de la serie temporal. Este tramo toma datos desde el 15 de agosto del año 2016 a las 22 horas con 11 minutos, hasta el 18 de agosto del mismo año a las 16 horas con 48 minutos.

Durante este tramo se hicieron 21 extracciones de datos donde hubo una media de 1609 muones encontrados, la menor cantidad de muones medida en este periodo fue de 1348 y la mayor de 1706. Para el caso del tramo 1 es irrelevante el análisis frecuencial debido a la poca cantidad de datos.

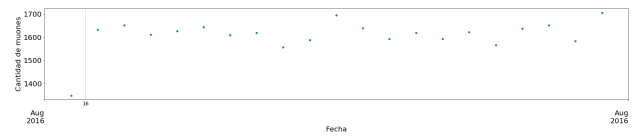


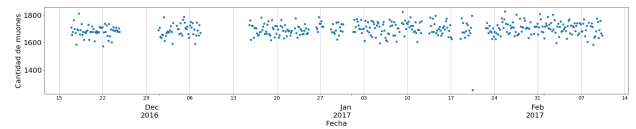
Figura 20: Tramo 1 de la serie temporal. **Fuente:** Elaboración Propia

10.2.2. Tramo 2

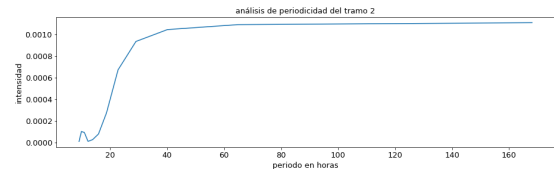
El segundo tramo (mostrado en la figura 21a) contiene los datos extraídos desde el 11 de noviembre del 2016 a las 21:35, hasta el 2 de febrero del año 2017 08:30. Tiene dos discontinuidades de una semana cada una.

El total de datos que se extrajo durante este periodo es de 435. En esta parte la media de los datos fue de 1697,04 muones por extracción y el máximo y menor conteo fueron de 1826 y 1256 respectivamente.

En la figura 21b se muestra el resultado del periodograma de Lomb-Scargle aplicado a este tramo. Si bien se muestra un alza en la intensidad para el periodo de 30 horas, esto es de tan solo 0.0008 niveles de intensidad.



(a) tramo 2 de la serie temporal



(b) análisis de periodicidad del tramo 2

Figura 21: Resultado y análisis del segundo tramo de la serie temporal.. **Fuente:** Elaboración Propia

10.2.3. Tramo 3

Para el tramo 3 de la serie temporal, el rango de extracción de datos va desde el 25 de junio del 2017 a las 6:25 horas, hasta el 15 de septiembre a las 14 horas con 14 minutos. En este tramo hay una discontinuidad de aproximadamente 2 semanas.

Se hicieron 444 extracciones, en las cuales se encontraron una media de 1729.99 muones por extracción, el máximo y mínimo de muones encontrados fue de 1854 y 1592, respectivamente.

En la figura 23b se nota como el comportamiento periódico tiene una intensidad superior para los periodos de 10 horas, pero aún así es muy poca la variación.

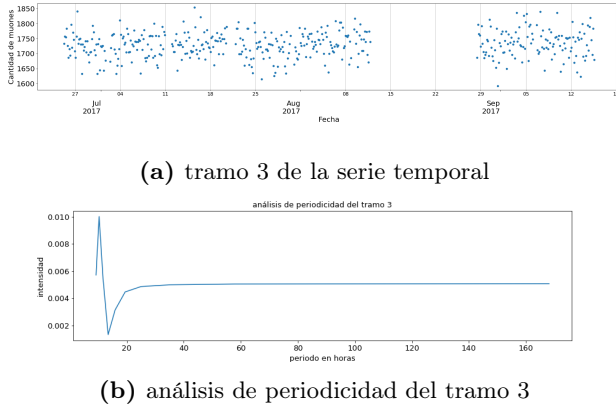


Figura 22: Resultado y análisis del tercer tramo de la serie temporal. **Fuente:** Elaboración Propia

10.2.4. Tramo 4

El tramo 4 de la serie temporal (figura 23a) abarca desde el 21 de diciembre del 2017 a las 4:54, hasta el 5 de abril del 2018 a las 4:15. Tiene dos discontinuidades, una de 3 semanas y otra de 1 semana.

El total de extracciones durante este tramo fue de 525 y la media de los muones fue de 1727, mientras que la máxima y mínima fueron de 1904 y 1565 respectivamente.

El resultado del periodograma para este tramo se ve en la figura 23b. Si bien la intensidad es mayor para periodos superiores al de 40 horas, esta diferencia no es significativa.

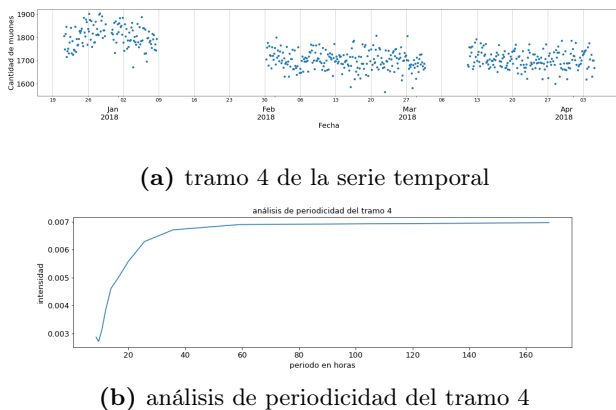


Figura 23: resultado y análisis del cuarto tramo de la serie temporal.

10.2.5. Tramo 5

El tramo 5 de la serie esta representado en la figura 24 y muestra los datos extraídos desde el 28 de julio del 2018 a las 21:03 hasta el 19 de agosto del mismo año a la 01:00. Fueron 125 extracciones de las cuales el promedio fue 1693.80, el máximo conteo fue de 1797 y el mínimo de 1604. No se realizo el periodograma para este tramo debido a la poca cantidad de datos.

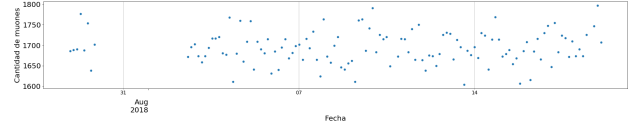


Figura 24: Tramo 5 de la serie temporal. **Fuente:** Elaboración Propia

10.2.6. Tramo 6

El último tramo de la serie temporal comprende un periodo desde el 28 de septiembre del 2018 a las 03:25, hasta el 23 de marzo del 2019 a las 03:00. Tiene 4 discontinuidades de una semana cada una.

Durante este tramo se extrajeron 989 conjuntos. la media, máxima y mínima de los conteos de muones para este tramo son de 1682.74, 1842 y 1547 respectivamente.

Al igual que en los otros casos, el periodograma mostrado en la figura 25b no muestra un periodo dominante. A pesar de mostrar un pico para un periodo de cada 10 horas, este valor no es significativo debido al nivel de intensidad mostrado.

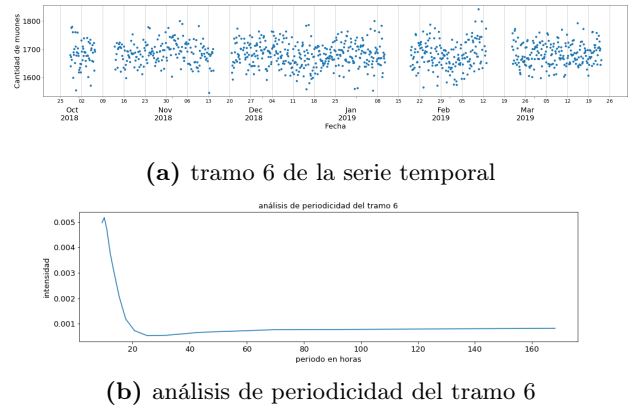


Figura 25: Resultado y análisis del sexto tramo de la serie temporal. **Fuente:** Elaboración Propia

En la tabla 2 se muestran el total de los resultados, mostrando cual fue el promedio, el máximo conteo de muones, el mínimo conteo de muones, la desviación estándar (σ), la fecha e inicio y la fecha de fin de todos los tramos y del total.

La media de muones corresponde a la cantidad de muones encontrados en cada extracción, cómo las extracciones se hicieron cada 3 horas, se puede calcular que el promedio de muones por hora fue de **563 muones por hora**. También se sabe que el área del CCD es de $0,0036 \text{ m}^2$ por lo cuál se calcula un flujo de **$130.32 \frac{\text{muones}}{\text{m}^2 \text{segundo}}$** .

Cuadro 2: Distribuciones de los resultados por tramo

	tramo1	tramo2	tramo3	tramo4	tramo5	tramo6	total
media de muones	1609	1697	1729	1727	1693	1682	1689
máxima de muones	1706	1826	1854	1904	1797	1842	1904
mínima de muones	1348	1256	1592	1565	1604	1547	1256
desviación estándar (σ)	68.674	49.346	42.965	60.45	39.5	42.75	53.091
fecha de inicio	15/8/2016	11/11/2016	25/6/2017	21/12/2017	28/7/2018	28/9/2018	15/8/2016
fecha de fin	15/8/2016	2/2/2017	15/9/2017	5/4/2018	19/9/2018	23/3/2019	23/3/2019

11. Conclusiones

- Se generaron las imágenes de los eventos de CONNIE desde la base de datos de la estructura de datos root, con el fin de analizar los eventos y de crear un conjunto de datos de entrenamiento.
- El conjunto de datos se clasificó en 7 grupos teniendo en cuenta la morfología y la intensidad de los eventos.
- Se realizó una clasificación a ojo de los eventos generados, demostrando que en una muestra aleatoria de 3200 eventos aproximadamente un tercio fueron muones. También se observó que los errores de segmentación y los elementos no definidos se encuentran en una cantidad considerable dentro del conjunto de datos, siendo la segunda y la tercera clase mas encontrada respectivamente.
- Para tener un conjunto de datos de entrenamiento balanceado se le aplicó el aumento de datos mencionado en la sección ???. Con los datos aumentados se crearon dos conjuntos para el entrenamiento de las redes neuronales. El primero conjunto de entrenamiento contenía 3 clases (muones, spots y tracks) y el segundo 6 clases (muones, spots, tracks, indefinidos, píxeles fallados y errores de segmentación).

El aumento de datos del primer conjunto de datos duplicó la cantidad de muones, los spots y worms fueron multiplicados por 16 y por 22, respectivamente. Para el segundo conjunto de datos se mantuvo el mismo aumento que en el primero, pero los errores de segmentación se multiplicaron por 4, los elementos indefinidos se multiplicaron por 3 y los píxeles dañados por 5.

Los únicos métodos de aumentos de datos que pueden ser utilizados en estas imágenes son los de espejados y los de rotación, esto es por la naturaleza de los eventos.

- Fueron cuatro los preprocesamientos utilizados para acondicionar las imágenes a las entradas de las redes neuronales: 32x32 a escala logarítmica, 64x64 a escala logarítmica, 128x128 a escala logarítmica y 64x64 a escala normal.
- Para las redes de 3 salidas la que tuvo mejor resultado fue la de 32x32. Debido a que usar esa cantidad de salidas produciría una sesgo muy grande

ante la presencia de mas de 3 clases para ser clasificadas, para los experimentos se utilizó la red de 64x64 con 6 salidas. Esta red tiene una exactitud de solamente 79 % pero demuestra una sensibilidad de 93.8 % para los muones.

- De los 4 cortes analíticos propuestos, fue el de Juan Estrada el de mejor resultado. El algoritmo de deeplearning demuestra tener un rendimiento mejor al corte de Juan Estrada, y es capaz de identificar también otros tipos de clases como los worms y spots.
- La serie temporal se dividió en 6 tramos. El análisis de periodicidad se le hizo a 4 de estos tramos, elegidos por la densidad de puntos. No se puede afirmar que existe comportamiento periódico a ninguno de los tramos que se le aplicó el algoritmo de Lomb-Scargle.
- La media de los muones observados en la serie temporal fue de 1689 cada 3 horas, la máxima y mínima fue de 1904 y 1256 respectivamente, la desviación estándar (σ) es de 53,091. Estos datos indican que el flujo de muones encontrado es de 130.32 muones por m^2 cada segundo.

12. Trabajos futuros

- Utilizar un algoritmo de detección de muones basándose en una red neuronal de una sola clase, con el fin de utilizar un conjunto de datos de entrenamiento simulado.
- Testar el ángulo de los muones detectados por el algoritmo de deeplearning para poder calcular el área de la distribución de ángulo de los muones y tener mas información de la eficiencia.
- Crear un segmentador de instancia capaz de separar los errores de segmentación.
- Trazar las trayectorias los muones a través de los distintos CCDs del CONNIE para obtener los ángulos de penetración.

13. Datos técnicos

A continuación se describen todas las herramientas con las que se realizó el proyecto.

13.1. Hardware

- Procesador Intel E5-2630L V3 de 8 núcleos y 16 threads.
- 16 GB de RAM, 2666MHz DDR4.
- 1(GPU) NVIDIA GeForce RTX 2080ti (11GB RAM).

13.2. Software

- Sistema operativo Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-46-generic x86_64).
- TensorFlow 2 [11]
- Python 3.6 [12]
- Pandas dataframe [13]
- Keras api [14]

Referencias

- [1] J. Alzubi, A. Nayyar, and A. Kumar, “Machine learning from theory to algorithms: an overview,” in *Journal of physics: conference series*, vol. 1142, p. 012012, IOP Publishing, 2018.
- [2] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O’Reilly Media, 2019.
- [3] A. Kaplan and M. Haenlein, “Siri, siri, in my hand: Who’s the fairest in the land? on the interpretations, illustrations, and implications of artificial intelligence,” *Business Horizons*, vol. 62, no. 1, pp. 15–25, 2019.
- [4] A. Aguilar-Arevalo, X. Bertou, C. Bonifazi, M. Butner, G. Cancelo, A. C. Vazquez, B. C. Vergara, C. Chavez, H. D. Motta, J. D’Olivo, J. D. Anjos, J. Estrada, G. F. Moroni, R. Ford, A. Foguel, K. P. H. Torres, F. Izraelevitch, A. Kanner, B. Kilminster, K. Kuk, H. Lima, M. Makler, J. Molina, G. Moreno-Granados, J. Moro, E. Paulini, M. S. Haro, J. Tiffenberg, F. Trillaud, and S. Wagner, “The CONNIE experiment,” *Journal of Physics: Conference Series*, vol. 761, p. 012057, oct 2016.
- [5] S. H. Neddermeyer and C. D. Anderson, “Note on the nature of cosmic-ray particles,” *Phys. Rev.*, vol. 51, pp. 884–886, May 1937.
- [6] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [7] D. J. Matich, “Redes neuronales: Conceptos básicos y aplicaciones,” *Universidad Tecnológica Nacional, México*, vol. 41, 2001.
- [8] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [9] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv e-prints*, p. arXiv:1409.1556, Sept. 2014.
- [10] J. T. VanderPlas, “Understanding the lomb-scargle periodogram,” *The Astrophysical Journal Supplement Series*, vol. 236, no. 1, p. 16, 2018.
- [11] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattemberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. Software available from tensorflow.org.
- [12] G. van Rossum, “Python tutorial,” Tech. Rep. CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995.
- [13] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1345–1359, 2010.
- [14] F. Chollet, “keras.” <https://github.com/fchollet/keras>, 2015.