



Proyecto de I+D+i financiado por el
Consejo de Seguridad Nuclear

**MEJORAS EN LAS NUEVAS TÉCNICAS DE INTELIGENCIA
ARTIFICIAL PARA LA DETECCIÓN DE ANOMALÍAS EN
REACTORES NUCLEARES**

**IMPROVING NEW AI TECHNIQUES FOCUSED TO
ANOMALIES DETECTION IN NUCLEAR REACTORS (INAIA)**

Entidad investigadora: Universitat Politècnica de València



Investigador principal: Gumersindo Verdú Martín

Convocatoria: 2021

1. INTRODUCCIÓN

Con el envejecimiento del parque de reactores nucleares en Europa, y especialmente en España, el seguimiento de estos reactores a través de modelos complejos se ha vuelto de gran interés para mantener un alto nivel de disponibilidad y seguridad. Es de especial interés localizar el lugar donde se ha producido una posible anomalía, así como el tipo de la misma, para garantizar la seguridad del reactor a través del análisis de fluctuaciones de flujo neutrónica. Por lo tanto, proponemos un marco de aprendizaje profundo para la deconvolución de las funciones de transferencia del reactor a partir de fuentes de ruido neutrónico inducido por perturbaciones. El principal objetivo de este proyecto ha sido desarrollar herramientas basadas en técnicas de *'Deep Learning'* para clasificar el tipo y localizar la perturbación, trabajando con datos simulados con diferentes niveles de ruido y estudiando el número de detectores que necesitan estar activos.

En concreto se ha realizado el estudio del reactor BIBLIS 2D y posteriormente el estudio del reactor 3D IAEA PWR.

2. OBJETIVOS

El objetivo principal de este proyecto ha consistido en desarrollar herramientas basadas en técnicas de *'Deep Learning'* para el estudio y monitorización del estado del núcleo del reactor en plantas nucleares. Para ello, tenemos que ser capaces de detectar y clasificar las anomalías producidas. En concreto, hay que trabajar los siguientes objetivos:

- Detectar el lugar donde se ha producido una perturbación.

- Detectar qué tipo de perturbación es.

- Estudiar el nivel máximo de ruido en la señal que se puede llegar a clasificar correctamente.

- Estudiar cuál es el mínimo grupo de detectores a emplear para conseguir clasificar con una fiabilidad alta.

3. ESTUDIO REACTOR BIBLIS 2D

Los datos utilizados han sido simulados para el reactor BIBLIS utilizando el código FEMFFUSION. Este trabajo se ha realizado utilizando la librería Keras, basada en TensorFlow, logrando desarrollar dos Redes Neuronales Convolucionales (CNN) que se adaptan bien al modelo de datos. Se obtienen resultados de alta precisión tanto en la predicción del tipo de perturbación como en la localización del lugar de la perturbación, con una baja tasa de error incluso cuando se utiliza un número mínimo de detectores

3.1 DESCRIPCIÓN DE LA SIMULACIÓN

3.1.1 Datos simulados

Para iniciar con la aplicación de técnicas de aprendizaje profundo para la detección de perturbaciones, se ha trabajado con datos en 2 dimensiones. Los datos de entrada son los correspondientes al reactor BIBLIS 2D discretizado en un mallado de 17x17, con 257 detectores activos, donde se han introducido perturbaciones a

diferentes frecuencias mediante la librería FEMFFUSION. Los puntos del mallado donde no haya detector activo se pondrán a cero. Por tanto, tendremos señales de 289 elementos combustibles, pero con 257 datos válidos. El motivo de emplear este formato es para poder trabajar con los datos en dos dimensiones. En concreto, se ha trabajado con 2570 ficheros correspondientes a la siguiente combinación de parámetros en la introducción de las perturbaciones:

Frecuencia (Hz): {0.1,0.5,1.0,5.0,10.0}

Tipo: {'sc', 'va'}

Origen de la perturbación: {1-257}

Por cada perturbación, se obtiene un fichero que contiene los siguientes ´arrays´ de datos:

Neutron Power

Group 1 flux

Group 2 flux

Flux Noise Group 1

Flux Noise Group 2

En particular, se trabaja con los datos flux2 y noise2, que juntos conforman la señal de entrada. En concreto, la señal de entrada será la división del valor absoluto de Flux Noise Group 2 por Group 2 flux.

3.1.2. Preparación de los datos.

La preparación adecuada de los datos no es algo trivial, ya que el resultado del proceso de entrenamiento y clasificación puede fallar si no están debidamente seleccionados, estructurados y normalizados. Por tanto, los siguientes pasos deben llevarse a cabo para considerar que los datos pueden ser empleados en el proceso de ´Deep Learning´:

1. Normalizar los valores de las señales de entrada entre 0 y 1.
2. Añadir ruido gaussiano con media 0 y desviación típica: 0.01, 0.02, 0.03, 0.04, 0.05. Por tanto, multiplicamos por 6 el número de señales que tenemos (la señal original más las 5 adicionales con diferentes niveles de ruido).

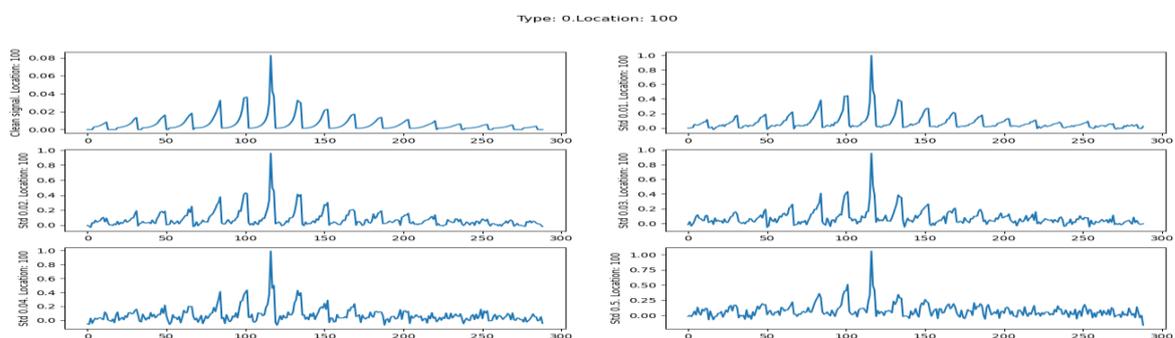


Figura 3.1: Ejemplo señal con ruido añadido.

En esta fase del proyecto se está añadiendo el ruido sobre todas las señales de entrada, tanto para entrenamiento como para test. Será necesario evaluar en el futuro si es la manera más correcta de introducir el ruido en los datos.

En la Figura 3.1 se puede apreciar cómo quedaría una señal de entrada con los diferentes niveles de ruido aplicados.

3. Dividir las señales en los diferentes conjuntos necesarios para la red neuronal:

Datos de entrenamiento: 67.5 % (10408 arrays)

Datos de validación: 22.5 % (3470 arrays)

Datos de test: 10 % (1542 arrays)

Estos datos se dividen de manera aleatoria mediante la función *train test split* del paquete de Python *sklearn* (*sklearn.model selection*). Sin embargo, aunque la división es aleatoria, se utiliza el parámetro *stratify*, para que en los tres grupos haya aproximadamente el mismo número de señales correspondientes a cada tipo que se quiere clasificar y no queden grupos mal balanceados. Es importante tener estos tres grupos de datos. El conjunto de entrenamiento sirve para ir afinando los parámetros en la fase de entrenamiento. El conjunto de validación sirve de apoyo al algoritmo durante su entrenamiento, confirmando si las decisiones que se están tomando se ajustan mejor o peor a la realidad. Con esto se evita el *overfitting*, es decir, que la red se aprenda todos los datos de entrenamiento, pero no sea capaz de generalizar con otros datos. Por último, el conjunto de test es un grupo de datos nunca vistos por la red neuronal en la etapa de entrenamiento. De este modo, se evalúa su rendimiento con datos nuevos.

4. Pasar las señales a formato 2D (17×17). En lugar de trabajar con arrays unidimensionales, vamos a transformar las señales a 2 dimensiones para trabajar con ellas como si de imágenes se tratara.

5. Etiquetar cada señal según el tipo de perturbación si se quiere clasificar por tipo.

6. Etiquetar cada señal según el detector en el que se haya introducido la perturbación si se quiere localizar el origen. 3

3.2. MATERIALES Y MÉTODOS

3.2.1. Simulación de las señales perturbadas

El caso de estudio es el reactor bidimensional BIBLIS 2D [4] que se puede observar en la Figura 3.2. Este reactor está compuesto por 257 conjuntos. Los datos simulados se obtuvieron con el código FEMFFUSION [5–7].

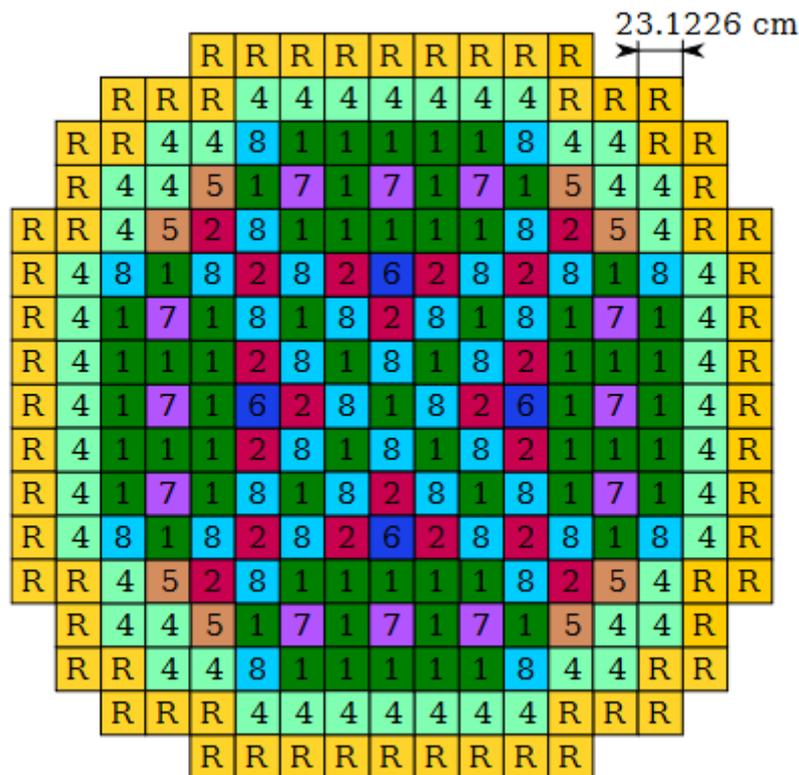


Figura 3.2. Reactor BIBLIS

FEMFFUSION es un Programa de Código Abierto escrito en lenguaje de programación C++, desarrollado en la Universidad Politécnica de Valencia. Este código se puede utilizar para la simulación de la ecuación neutrónica en reactores de cualquier tipo de geometría y dimensión del problema, cualquier tipo de reactor (comercial o de investigación), de cualquier generación (Gen-II, Gen-III, Gen-III+, Gen-IV) y para reactores SMR (Small Modular Reactor). Es un código de elementos finitos para modelar sistemas nucleares. FEMFFUSION se ha utilizado para obtener los datos de dos tipos de perturbaciones a diferentes frecuencias, $f = 0.1, 0.5, 1, 5$ y 10 Hz, y variando la ubicación de la perturbación para cada uno de los 257 elementos combustibles posibles. Se realizaron un total de 2570 simulaciones de ruido de neutrones.

Los dos tipos de perturbaciones son:

- Un cambio de la sección transversal de absorción $\Sigma_a(t) = \Sigma_a(0) + 0.01\Sigma_a(0) \text{sen}(2\pi f t)$ en el elemento de combustible (i, j).
- Un cambio de la sección transversal de dispersión $\Sigma_s(t) = \Sigma_s(0) + 0.01\Sigma_s(0) \text{sen}(2\pi f t)$ en el elemento de combustible (i, j).

En este estudio, FEMFFUSION resuelve la ecuación de difusión de ruido de neutrónico en el dominio de la frecuencia en la aproximación de dos grupos de energía sin upscattering usando el método de elementos finitos con polinomios cúbicos en la discretización espacial. Estas ecuaciones diferenciales de valor complejo se pueden escribir como:

$$\mathcal{A}\delta\Phi = \mathcal{B}\Phi_0, \quad (1)$$

where

$$\mathcal{A} = \begin{bmatrix} \frac{i\omega}{v_1} - \vec{\nabla} D_1 \vec{\nabla} + \Sigma_{a1}^0 + \Sigma_{12}^0 - \gamma \nu \Sigma_{f1}^0 & -\gamma \nu \Sigma_{f2}^0 \\ -\Sigma_{12}^0 & \frac{i\omega}{v_2} - \vec{\nabla} D_2 \vec{\nabla} + \Sigma_{a2} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} -\delta \Sigma_{a1} - \delta \Sigma_{12} + \gamma \delta \nu \Sigma_{f1} & +\gamma \delta \nu \Sigma_{f2} \\ \delta \Sigma_{12} & -\delta \Sigma_{a2} \end{bmatrix}. \quad (2)$$

El sistema de ecuaciones complejas tiene que resolverse después del problema de estado estacionario que calcula el sistema algebraico de vectores de pesos, asociados al flujo de neutrones estáticos, φ_0 . El problema de valor propio estático relacionado es resuelto con la misma discretización espacial que el ruido neutrónico en el dominio de la frecuencia para obtener resultados coherentes.

Cabe señalar que el resultado de cada una de estas simulaciones es la distribución espacial del ruido neutrónico ($\delta\varphi/\varphi_0$) para la perturbación y frecuencia seleccionadas. Un ejemplo del flujo de neutrones estáticos y el ruido de neutrones producido por un cambio de la sección transversal de dispersión en el conjunto (8, 4) se muestra en las Figuras 3.3 y 3.4.

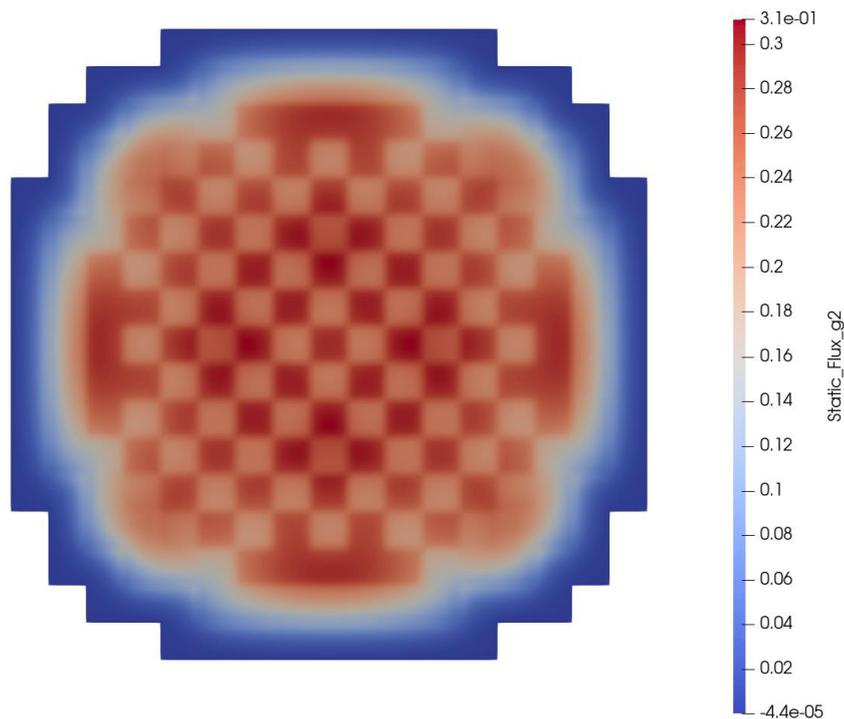


Figura 3.3 Flujo Estático.

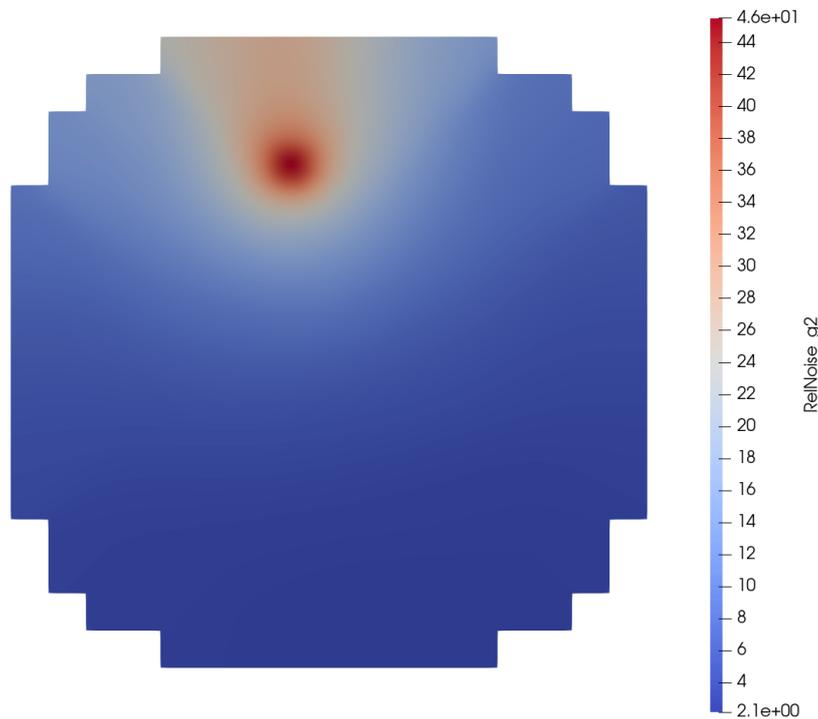


Figura 3.4. Ruido neutrónico

Para cada perturbación, se obtiene un archivo que contiene las siguientes matrices de datos: Grupo 1 de potencia de neutrónica (φ_1), Grupo 2 de potencia neutrónica (φ_2), Grupo 1 de ruido de flujo ($\delta\varphi_1$), Grupo 2 de ruido de flujo ($\delta\varphi_2$). A partir de estas señales, la red neuronal funcionará con Neutron Power Group 2 y Flux Noise Group 2, que juntos formarán la entrada de la señal. De esta forma, la función de transferencia del detector se establece en el flujo de neutrones térmicos, como se especifica en la ecuación. (3).

$$\text{Señal del detector} = |\delta\varphi_2|/\varphi \quad (3)$$

3.2.2. Preparación de los datos

En 'Deep Learning', la cantidad y calidad de los datos que se utilizarán para entrenar una red es fundamental para obtener resultados satisfactorios. Por esta razón, las técnicas de ampliación de datos suelen emplearse cuando los datos originales son limitados [8,9]. En este proyecto se han realizado dos experimentos para cada caso: uno utilizando las señales originales (2570 señales simuladas) y otro que agrega ruido a esas señales y amplía el tamaño de los datos de entrada.

Para los estudios con datos ruidosos, se agregó ruido gaussiano blanco aditivo (AWGN) a las señales. La potencia de ruido se fijó en un porcentaje de la potencia de la señal, que va desde el 5% al 25% con incrementos del 5%.

El proceso de agregar AWGN a una señal implica varios pasos. Primero, la potencia de la señal original P_s se calcula tomando el valor cuadrático medio de la señal (Ec. (4)), siendo n el número de elementos de las señales (17×17). A continuación, se determina el porcentaje de ruido que se desea añadir ($noiseLevel$), que se utiliza para calcular la potencia de ruido P_n (ecuación (5)). Entonces, se genera ruido gaussiano blanco como una distribución normal con media de cero y una desviación estándar de uno (Ec. (6)). Luego, la potencia del ruido generado se ajusta para que coincida con el porcentaje deseado (Eq. (7)), y la amplitud del ruido se escala por un factor correspondiente al nivel de ruido deseado (Ec. (8)). Finalmente, el ruido se agrega a la señal original, dando como resultado una señal con AWGN añadido (Ec. (9)).

$$P_s = \frac{1}{n} \sum_{k=1}^n signal(k)^2 \quad (4)$$

$$P_n = noiseLevel \times P_s \quad (5)$$

$$noise = \sim \mathcal{N}(0, 1) \quad (6)$$

$$noise = \sqrt{\frac{P_n}{\frac{1}{n} \sum_{k=1}^n noise(k)^2}} \times noise \quad (7)$$

$$noise = noiseLevel \times noise \quad (8)$$

$$signal_{noise} = signal + noise \quad (9)$$

En las Figuras 3.5 y 3.6 se puede apreciar la forma de las señales con el máximo nivel de ruido para ambos tipos de perturbación y frecuencia 0,1 Hz. Después de agregar el ruido, todas las señales se estructuran como 2-matrices dimensionales que se proporcionarán como entrada a las CNN bidimensionales [10].

Type: SC. Frequency: 0.1 Hz. Location: 50

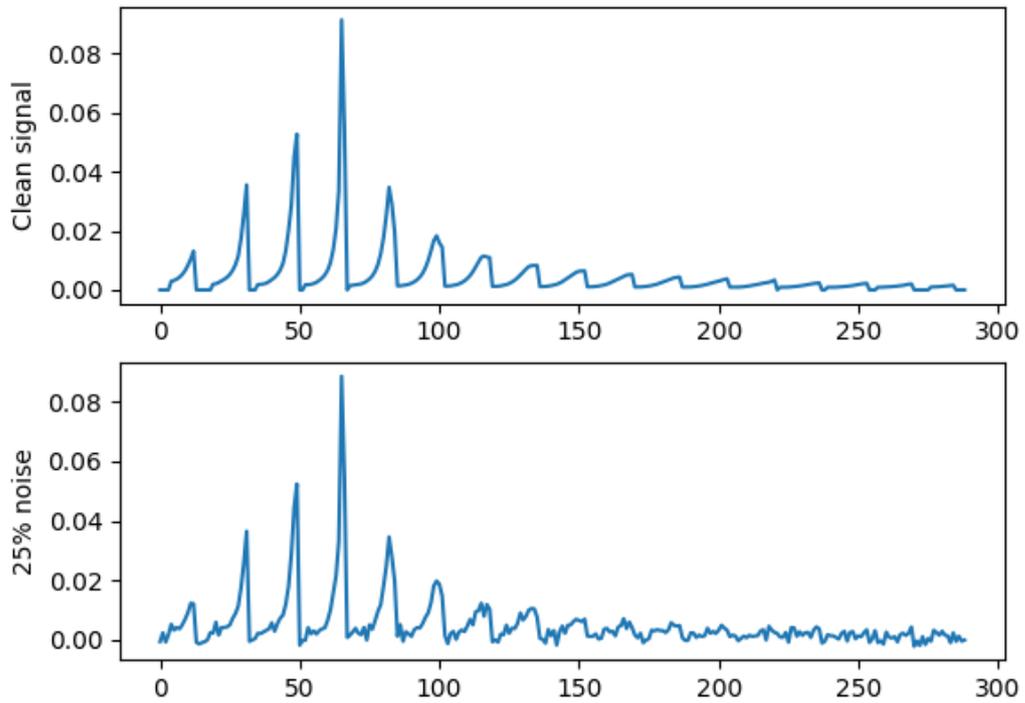


Figura 3.5. Señal limpia y señal ruidosa para la perturbación tipo 'scattering'

Type: VA. Frequency: 0.1 Hz. Location: 50

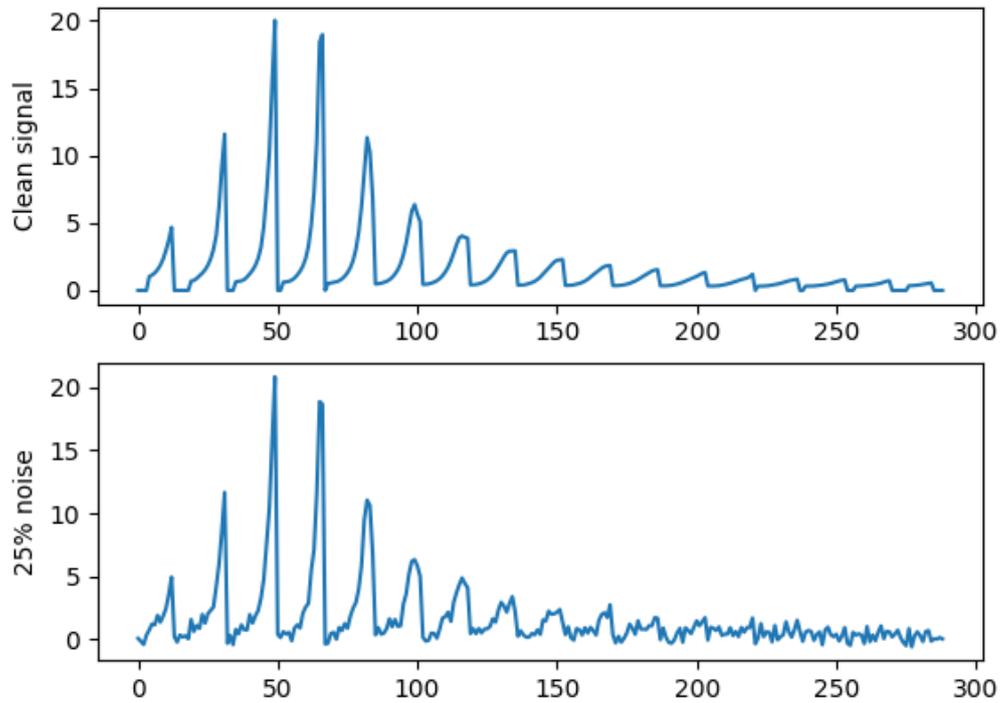


Figura 3.6. Señal limpia y señal ruidosa para la perturbación tipo absorción.

Una vez que se procesaron todos los datos de entrada, el conjunto completo de datos se divide aleatoriamente en tres grupos: un 15% para test, un 10% como conjunto de validación en la fase de entrenamiento, y el 75% restante como conjunto de entrenamiento.

Vale la pena mencionar que el conjunto completo tiene 2570 señales cuando las señales son limpias, o 25700 cuando hay ruido, ya que se agregaron 9 niveles de ruido a los datos originales. La división en los tres sets está equilibrada, por lo que aproximadamente tenemos el mismo número de señales de cada grupo de clasificación asignado a cada set.

2.3. Diseño de Redes Neuronales Convolucionales

Los modelos CNN han sido desarrollados y entrenados usando la librería Keras [11] en una GPU Nvidia A100.

El reactor BIBLIS 2D tiene 257 elementos de combustible, clasificados por su material, y se puede representar como una malla 17×17 . En este proyecto se supone que cada elemento de combustible contiene un detector de flujo de neutrones. Por lo tanto, las señales de entrada tienen 257 elementos, pero se envían a las 2D CNN como matrices de 17×17 para mantener la información espacial de los datos, Además, el número de detectores activos varía para determinar el número óptimo de detector para clasificar y detectar de manera confiable el tipo de perturbación y la ubicación donde se produce la perturbación original.

Cuando se reduce el número de detectores, las posiciones en las que se ha desactivado un detector se pone la señal a 0, pero las señales de entrada todavía están estructuradas como matrices de 17×17 . Para disminuir el número de detectores activos, se han seguido dos estrategias. Con la primera, los detectores se seleccionan en base a su material. Por ejemplo, si el número deseado de detectores es 8, el material seleccionado es el marcado por un 5 (Figura 2). Para 12 detectores, seleccionamos el material 7 ó el 5 y el 6 juntos, y así sucesivamente. Con la segunda estrategia, los detectores se seleccionan manualmente para que se distribuyan uniformemente a lo largo de la malla de 17×17 . Por ejemplo, con 4 detectores, se selecciona uno por trimestre. Con las dos estrategias se puede analizar si la posición de los detectores activos afecta los resultados de la clasificación.

La Figura 3.7 muestra el diseño de la CNN utilizada para clasificar las señales por el tipo de perturbación. Como puede verse, la red recibe las señales, luego aplica varias etapas de convolución, activación ReLU y Normalización por lotes (BN). El paso de Normalización por lotes aplica una transformación que mantiene la media de los datos salida cerca de 0 y la desviación estándar de salida cerca de 1. Después de cada etapa convolucional se aplica un paso Max Pooling para reducir la muestra de los datos. Después de todas las etapas convolucionales hay dos capas totalmente conectadas, cada una seguida de una capa de abandono. La tasa de deserción se fijó en 0.3 y se utiliza para establecer las unidades de entrada en 0 con una frecuencia de tasa en cada paso durante el tiempo de entrenamiento, lo que ayuda a evitar el sobreajuste.

En el paso final, hay una capa unidimensional con dos neuronas que usan una función de activación sigmoidea para clasificar los datos en las dos clases deseadas. Para encontrar la ubicación de la perturbación, la CNN es idéntica excepto la última fase, donde hay una capa final con 257 neuronas (ya que el modelo tiene 257 células) y una función de activación softmax para hacer una clasificación multiclase. Esta clasificación da una predicción de la posición en el reactor donde se origina la perturbación. Este diseño de CNN se puede ver en la Figura 3.8.

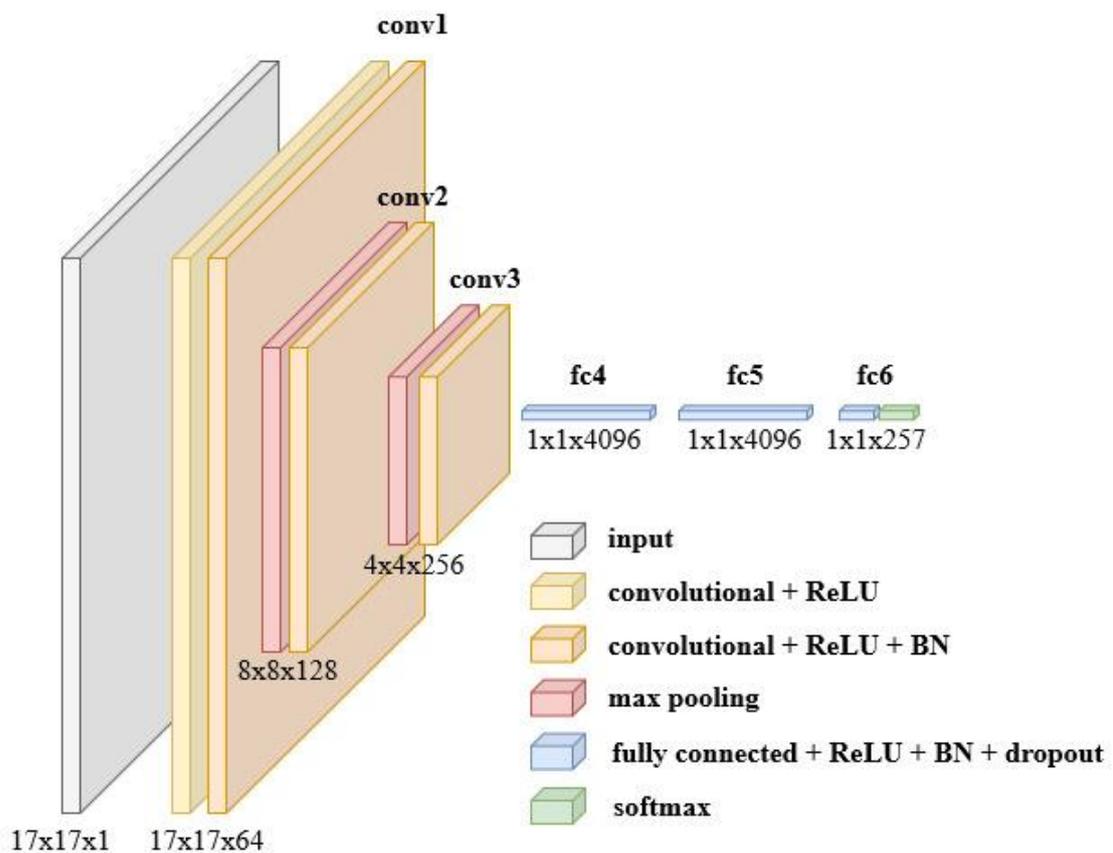


Figura 3.7. CNN utilizada para clasificar el tipo de perturbación en las señales.

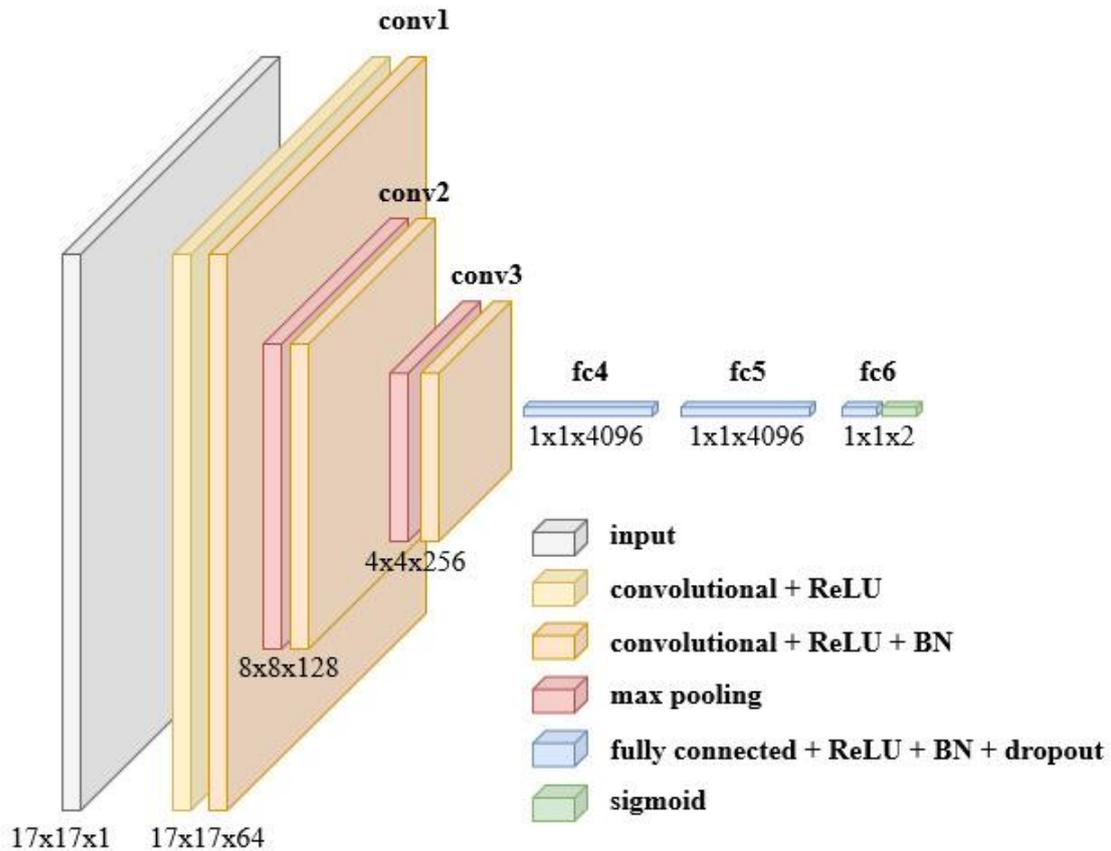


Figura 3.8. CNN utilizada para clasificar la ubicación de la perturbación en las señales

3.3. RESULTADOS NUMÉRICOS

3.3.1. Clasificación por tipo de perturbación

Una vez entrenadas las CNN, se obtiene un modelo del clasificador. Este modelo se puede utilizar entonces para clasificar datos de entrada que no se han incluido en los conjuntos de entrenamiento o validación. En la división inicial de los datos, una prueba de test se incluyó, que ahora se utiliza para evaluar el rendimiento del modelo.

En primer lugar, se predice el tipo de perturbación de las señales de prueba, utilizando el modelo que se presentó en la Figura 3.7. Los resultados obtenidos cuando se seleccionan los detectores por sus materiales se muestran en la Tabla 3.1. Para la selección uniformemente distribuida, los resultados están en la Tabla 3.2. En cada tabla, se proporciona la precisión de la clasificación, teniendo en cuenta si se ha añadido ruido a las señales (en el entrenamiento del modelo y también en las señales de test) o si las señales estaban limpias. En este caso, todos los resultados son similares. La clasificación por tipo de perturbación obtiene una alta precisión (más de 0.94, o 94%) en todas las configuraciones. Añadir ruido a las señales para extender el conjunto de datos original parece ser beneficioso para la precisión, aunque no muy significativo para esta clasificación. En cuanto a la estrategia utilizada para seleccionar los detectores activos, las diferencias se

aprecian cuando el número de detectores es elevado. Si se seleccionan por material, la mayor precisión obtenida es de 0.966. Sin embargo, si se distribuyen uniformemente, se puede alcanzar una precisión de 0.995 cuando se utilizan 65 ó 48 detectores.

Estos resultados muestran que el modelo utilizado para clasificar las señales por el tipo de perturbación está bien diseñado y se adapta bien a los datos. Con cada configuración, más del 94% de las señales incluidas en el equipo de prueba son clasificadas correctamente, incluso cuando el número de detectores activos es sólo 4.

Material Selected	Active Detectors	Accuracy (Test Set, no noise)	Accuracy (Test Set, added noise)
All	257	1	1.0
1	65	0.966	1.0
4	48	0.987	1.0
8	36	0.951	1.0
2	20	0.961	1.0
5, 7	20	0.951	1.0
7	12	0.951	0.966
5,6	12	0.989	0.999
5	8	0.951	0.992
6	4	0.951	1.0

Table 3.1. Precisión en la clasificación de tipos, reduciendo el número de detectores activos (por material del detector).

Active Detectors	Accuracy (Test Set, no noise)	Accuracy (Test Set, added noise)
257	1.0	1.0
65	1.0	1.0
48	0.992	1.0
36	0.982	1.0
20	0.979	1.0
12	0.966	0.999
8	0.956	1.0
4	0.951	0.999

Table 3.2. Precisión en la clasificación de tipos, reduciendo el número de detectores activos (distribuidos uniformemente)..

3.3.2. Ubicación de la perturbación

Una vez predicho el tipo de perturbación de las señales de prueba, también es de interés saber dónde se origina la perturbación. Para este propósito, usamos la CNN que se muestra en la Figura 3.8, donde realizamos una clasificación multiclase de las señales, teniendo 257 posibilidades de localización. En esta etapa de clasificación, aunque la ubicación predicha por el modelo no es correcta, se puede medir el error de la predicción.

La distancia euclídea entre dos píxeles en una imagen 2D se puede calcular usando la ecuación. (10), donde x_1 y y_1 son las coordenadas del primer píxel, y x_2 e y_2 son las coordenadas del segundo píxel. La variable d representa la distancia euclídea entre los dos píxeles. En este caso, el error de la ubicación prevista puede medirse como la distancia entre la ubicación prevista y la ubicación correcta, ya que el reactor se puede ver como una malla de 17×17 .

$$d = \text{SQRT} ((x_1-x_2)^2 + (y_1-y_2)^2) \quad (10)$$

Los resultados de la clasificación se muestran en la Tabla 3.3 con los detectores seleccionados por su material y Tabla 3.4 para la selección uniformemente distribuida. En ambas tablas, el error se da como la media de la distancia euclídea entre todas las predicciones y su ubicación correcta.

Cuando todos los detectores de flujo de neutrones están activos, la ubicación de la perturbación es correcta el 100% de las veces.

La precisión disminuye a medida que disminuye el número de detectores activos. Mirando los resultados obtenidos con los modelos que son entrenados sin ruido en las señales en ambas tablas, se puede ver que la precisión desciende hasta un 76% en algunos casos (con 4 detectores en la Tabla 3.3 y con 12 en la Tabla 3.4). Aunque no todas las señales se predicen correctamente, el error medio da una idea de la distancia entre la predicción y la ubicación correcta. El error es bajo en todos los casos, y esto podría proporcionar una estimación de la ubicación de la perturbación.

Sin embargo, viendo los resultados obtenidos por los modelos que fueron entrenados con señales ruidosas, la precisión es muy superior, siempre por encima del 96% incluso utilizando sólo 4 detectores. Los resultados son muy similares para ambas distribuciones, tanto en términos de precisión como de error. Con esto, se determina que aumentar el número de señales (y por lo tanto el tamaño del conjunto de entrenamiento) es crucial para mejorar la precisión de los modelos cuando los datos simulados son limitados. La precisión de los modelos con ruido ha aumentado entre un 12% y un 20% cuando se utilizan pocos detectores, dependiendo de la distribución. Esta mejora es significativa.

	Active Detectors	Location Accuracy (Test Set, no noise)	Mean Distance (Error)	Location Accuracy (Test Set, added noise)	Mean Distance (Error)
All	257	1	0	1	0
1	65	0.948	0.271	0.999	0.004
4	48	0.943	0.476	0.999	0.008
8	36	0.886	0.578	0.998	0.003
2	20	0.788	0.77	0.986	0.038
5, 7	20	0.917	0.512	0.986	0.085
7	12	0.909	0.621	0.96	0.263
5,6	12	0.868	0.552	0.985	0.055
5	8	0.868	0.545	0.962	0.137
6	4	0.762	0.813	0.961	0.154

Table 3.3. Precisión de ubicación reduciendo el número de detectores activos (por material del detector).

Active Detectors	Location Accuracy (Test Set, no noise)	Mean Distance (Error)	Location Accuracy (Test Set, added noise)	Mean Distance (Error)
257	1	0	1	0
65	0.982	0.181	0.9997	0.004
48	0.938	0.313	0.9997	0.0003
36	0.933	0.307	0.999	0.009
20	0.927	0.54	0.965	0.163
12	0.883	0.504	0.98	0.081
8	0.834	0.832	0.963	0.132
4	0.84	0.483	0.968	0.112

Table 3.4. Precisión de ubicación reduciendo el número de detectores activos (distribuidos uniformemente).

3.4. CONCLUSIONES

Por lo tanto, se ha realizado una primera aproximación a la aplicación de CNNs para el estudio de anomalías en el reactor BIBLIS 2D. En primer lugar, se realizaron 2570 simulaciones numéricas de ruido neutrónico, una para cada tipo de perturbación, frecuencia y ubicación. Luego, ha sido necesario preparar los datos para su correcto uso en redes neuronales convolucionales según el propósito de la clasificación

Además, se han estudiado varios diseños de CNN para aplicarlos a estos datos. Como se muestra, hemos desarrollado dos modelos de CNN en dos dimensiones que se adaptan bien al problema a resolver. Evaluando estos modelos con los datos actuales, la precisión obtenida en las clasificaciones ha sido perfecta cuando todos los detectores están activos. Al disminuir el número de detectores activos, se ha observado que cuando se entrenan los modelos con sólo los datos simulados (señales limpias), la precisión es menor que cuando el conjunto de datos se aumenta agregando ruido gaussiano blanco aditivo a las señales. En ambos casos de clasificación obtiene una precisión muy alta, especialmente cuando los modelos se entrenan con el conjunto de datos extendido. Este podría ser el punto de partida para desarrollar una herramienta que ayude a los técnicos de las centrales nucleares en el seguimiento del estado del reactor y asegurar una detección temprana de posibles anomalías.

REFERENCIAS

- [1] T. Tasakos, G. Ioannou, V. Verma, G. Alexandridis, A. Dokhane y A. Stafylopatis. “Detección de anomalías basada en aprendizaje profundo en núcleos de reactores nucleares”. En Actas de la Conferencia Internacional sobre Matemáticas y Métodos Computacionales Aplicados a la Ciencia e Ingeniería Nucleares (M&C 2021), En línea, págs. 3 a 7 (2021).
- [2] A. Papaoikonomou, J. Wingate, V. Verma, A. Durrant, G. Ioannou, T. Papagiannis, M. Yu, G. Alexandridis, A. Dokhane, G. Leontidis, S. Kollias, and A. Stafylopatis. “Deep learning techniques for in-core perturbation identification and localization of time-series nuclear plant measurements.” *Annals of Nuclear Energy*, volume 178, p. 109373 (2022). URL <https://www.sciencedirect.com/science/article/pii/S0306454922004078>.
- [3] D. A. Ejigu and X. Liu. “Gradient descent-particle swarm optimization based deep neural network predictive control of pressurized water reactor power.” *Progress in Nuclear Energy*, volume 145, p. 104108 (2022). URL <https://www.sciencedirect.com/science/article/pii/S0149197021004601>.
- [4] E. Muller and Z. Weiss. “Benchmarking with the multigroup diffusion high-order response matrix method.” *Annals of Nuclear Energy*, volume 18(9), pp. 535–544 (1991).
- [5] A. Vidal-Ferrandiz, A. Carreño, D. Ginestar, and G. Verdú. “ $\hat{\sim}$ FEMFFUSION: A finite element method code for the neutron diffusion equation.” <https://femffusion.webs.upv.es/> (2022).

- [6] A. Vidal-Ferrandiz, A. Carreño, D. Ginestar, and G. Verdú. “Edge-wise perturbations to model vibrating fuel assemblies in the frequency-domain using FEMFFUSION: Development and verification.” *Annals of Nuclear Energy*, volume 175, p. 109246 (2022).
- [7] A. Carreño, A. Vidal-Ferrándiz, D. Ginestar, and G. Verdú. “Frequency-domain models in the SPN $\hat{\rho}$ approximation for neutron noise calculations.” *Progress in Nuclear Energy*, volume 148, p. 104233 (2022).
- [8] L. Taylor and G. Nitschke. “Improving deep learning with generic data augmentation.” In 2018 IEEE symposium series on computational intelligence (SSCI), pp. 1542–1547. IEEE (2018).
- [9] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu. “Data augmentation for deep learning-based radio modulation classification.” *IEEE access*, volume 8, pp. 1498–1506 (2019).
- [10] Y. LeCun, Y. Bengio, and G. Hinton. “Deep learning.” *nature*, volume 521(7553), pp. 436–444 (2015).
- [11] F. Chollet et al. “Keras.” <https://keras.io> (2015).

4. REACTOR BENCHMARK 3D-IAEA

Partiendo de los modelos desarrollados para la geometría 2D, se han propuesto diversas mejoras además de la aplicación de las técnicas empleadas para una geometría 3D. El reactor empleado para este proyecto es el benchmark 3D-IAEA, sobre el que se han aplicado perturbaciones axiales y radiales empleando el código de elementos finitos FEMFFUSION. Se ha observado que los resultados obtenidos aplicando estas técnicas presentan una elevada precisión, aun habiéndose aplicado un elevado ruido blanco a la señal y para un bajo número de detectores activos.

El objetivo principal de este proyecto ha sido el desarrollo de herramientas, basadas en técnicas de ‘*deep learning*’, para el estudio y monitorización de las perturbaciones que se producen en el núcleo de los reactores nucleares [1][2]. Para ello los objetivos parciales sobre los que se han trabajado han sido:

- Clasificar con elevada precisión el tipo de perturbación.
- Localizar con elevada precisión dónde se ha desarrollado la perturbación.
- Estudiar el mínimo número de detectores activos para obtener una elevada precisión.

4.1 DESCRIPCIÓN DE LA SIMULACIÓN

4.1.1. Datos simulados

Los datos de entrada que han sido empleados para el desarrollo de estos nuevos modelos en 3 dimensiones provienen del reactor 3D-IAEA. Esta geometría está constituida por 19 planos en 2D de 241 elementos introducidos como 19 mallas de 17x17. Mediante el código FEMFFUSION se introducen, sobre cada uno de los

elementos que constituyen la geometría, 3 tipos distintos de perturbaciones. Para este nuevo caso, además de las perturbaciones sobre las secciones eficaces de dispersión y absorción, se introduce un cambio en la velocidad axial del flujo de refrigerante. El cambio de geometría a tridimensional y la perturbación axial que se añade, provoca que el número de ficheros con los que se van a entrenar los modelos aumente considerablemente, pasando a ser de 51815.

Perturbación axial.

- Posible ubicación de la perturbación: 1 – 241.
- Velocidades del flujo (cm/s): 100, 200, 300, 400, 500.
- Frecuencias de oscilación (Hz): 0.1, 0.5, 1.0, 5.0, 10.0.

Oscilación secciones eficaces.

- Tipos de perturbación: sección eficaz de dispersión (sc), sección eficaz de absorción (va).
- Posible ubicación radial de la perturbación: 1 – 241.
- Posible ubicación axial de la perturbación: 1 – 19.
- Frecuencias de oscilación (Hz): 0.1, 0.5, 1.0, 5.0, 10.0.

Cada uno de los archivos generados contiene valiosa información para conocer el estado del reactor. Concretamente se obtiene, para cada posición en el reactor, la potencia neutrónica, el flujo y el ruido neutrónico rápidos y térmicos. No obstante, interesa tomar como datos de entrada el flujo y ruido de neutrones térmicos, puesto que son los encargados de generar las fisiones y son los evaluados por la instrumentación interna del núcleo del reactor [3].

4.1.2. Preparación de los datos.

La preparación de los datos para el caso tridimensional incluye:

1. Obtener el ruido neutrónico relativo para los neutrones térmicos. Los sistemas del reactor actúan para garantizar su seguridad cuando existe una desviación del flujo neutrónico respecto de su valor medio. Por ello evaluamos de la misma forma que los detectores, la señal medida perturbada (ruido neutrónico) frente el flujo neutrónico.
2. Añadir ruido gaussiano con media 0 y desviación típica 1 a distintos niveles (5% hasta 25% en incrementos de 2.5%) para proceder con el “*data augmentation*” y evitar el “*overfitting*” y además introducir los errores de medición que los detectores pueden cometer.
3. Dividir las señales en los diferentes conjuntos necesarios para la red neuronal:
 - Datos de entrenamiento: 73 % (37825 arrays)
 - Datos de validación: 12 % (6218 arrays)
 - Datos de test: 15 % (7772 arrays)

Estos datos se dividen de manera aleatoria mediante la función *train test split* del paquete de Python *sklearn* (*sklearn.model selection*). Sin embargo, aunque la división es aleatoria, se utiliza el parámetro *stratify*, para que el número de datos de cada grupo a clasificar queden balanceados. El conjunto de entrenamiento sirve para ir afinando los parámetros en la fase de entrenamiento. El conjunto de validación sirve de apoyo al algoritmo durante su entrenamiento, confirmando si las decisiones que se están tomando se ajustan mejor o peor a la realidad. Con esto se evita el “*overfitting*”, es decir, que la red se aprenda todos los datos de entrenamiento, pero no sea capaz de generalizar con otros datos. Por último, el conjunto de test es un grupo de datos nunca vistos por la red neuronal en la etapa de entrenamiento. De este modo, se evalúa su rendimiento con datos nuevos.

4. Pasar las señales a formato 3D (17×17x19). En lugar de trabajar con vectores unidimensionales y arrays bidimensionales como el caso previo, asumiendo que los datos de entrada se comportan como imágenes bidimensionales, pero colores RGB para introducir los planos 19 axiales.
5. Etiquetar cada señal según el tipo de perturbación si se quiere clasificar por tipo.
6. Etiquetar cada señal según el detector en el que se haya introducido la perturbación si se quiere localizar el origen.

4.2. MATERIALES Y MÉTODOS

4.2.1. Simulación de las señales perturbadas

El caso de estudio es el reactor tridimensional 3D-IAEA que se puede observar en la Figura 4.1. Este reactor está compuesto por 241 elementos y 19 planos [4]. Los datos simulados se obtuvieron con el código FEMFFUSION.

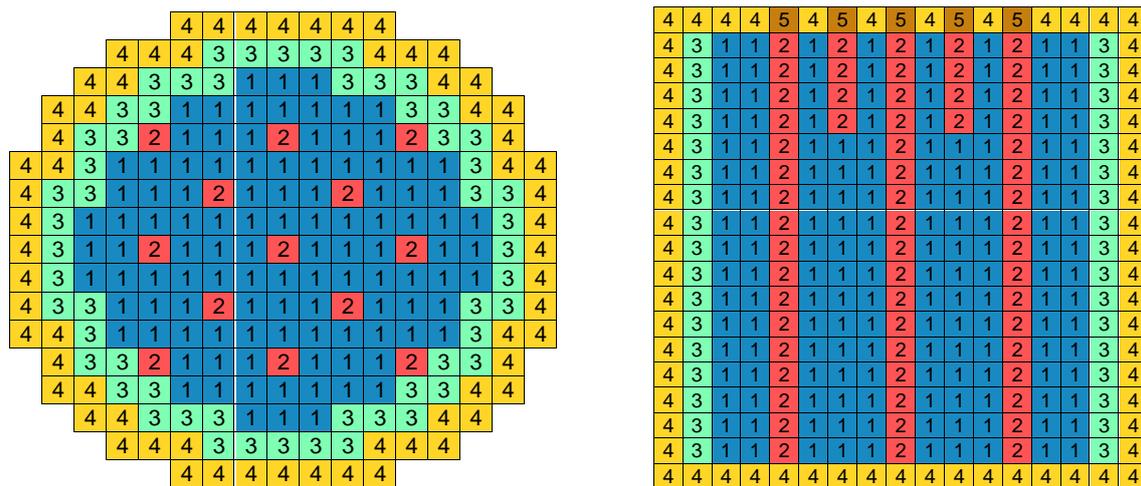


Figura 4.1. Reactor 3D-IAEA.

FEMFFUSION es un código de elementos finitos abierto escrito en lenguaje de programación C++, desarrollado en la Universidad Politécnica de Valencia. Este código se puede utilizar para la simulación reactores nucleares 1D, 2D y 3D

empleando la ecuación de la aproximación a la difusión neutrónica y aproximaciones SPN. Además, puede ser aplicado para cualquier tipo de reactor (comercial o de investigación), de cualquier generación (Gen-II, Gen-III, Gen-III+, Gen-IV) e incluso para reactores SMR (Small Modular Reactor). FEMFFUSION se ha utilizado para obtener los archivos de entrada a las redes neuronales convolucionales. Sobre las secciones eficaces iniciales se introducen las perturbaciones, teniendo en cuenta las distintas frecuencias de oscilación. Además, se realizan dichos cálculos sobre cada uno de los elementos que componen el reactor, siendo un total de 241 posiciones y por cada uno de los 19 planos.

Los dos tipos de perturbaciones sobre las secciones eficaces son las siguientes.

- Sección eficaz de absorción: $\Sigma_a(t) = \Sigma_a(0) + 0.01\Sigma_a(0)\sin(\omega t)$ en el elemento de combustible (i, j, k).
- Sección eficaz de dispersión $\Sigma_s(t) = \Sigma_s(0) + 0.01\Sigma_s(0)\sin(\omega t)$ en el elemento de combustible (i, j, k).

En este trabajo, FEMFFUSION resuelve la ecuación de difusión de ruido de neutrónico en el dominio de la frecuencia, para 2 grupos de energía y sin considerar *upscattering*, haciendo uso del método de elementos finitos con polinomios cúbicos en la discretización espacial [5][6]. Estas ecuaciones diferenciales de valor complejo se pueden escribir como:

$$\mathcal{A}\delta\phi = \mathcal{B}\phi_0, \quad (1)$$

$$\mathcal{A} = \begin{bmatrix} \frac{i\omega}{v_1} - \vec{\nabla}D_1\vec{\nabla} + \Sigma_{a1}^0 + \Sigma_{12}^0 - \gamma v\Sigma_{f1}^0 & -\gamma v\Sigma_{f2}^0 \\ -\Sigma_{12}^0 & \frac{i\omega}{v_2} - \vec{\nabla}D_2\vec{\nabla} + \Sigma_{a2}^0 \end{bmatrix}, \quad (2)$$

$$\mathcal{B} = \begin{bmatrix} -\delta\Sigma_{a1} - \delta\Sigma_{12} + \gamma\delta v\Sigma_{f1} & -\gamma v\Sigma_{f2}^0 \\ \delta\Sigma_{12} & -\delta\Sigma_{a2} \end{bmatrix},$$

$$\gamma = (1 - \beta) + \sum_{k=1}^K \frac{\lambda_k \beta_k}{i\omega + \lambda_k}. \quad (3)$$

El sistema de ecuaciones complejas tiene que resolverse después del problema de estado estacionario que calcula el sistema algebraico de vectores de pesos, asociados al flujo de neutrones estáticos, ϕ_0 . El problema de valor propio estático relacionado es resuelto con la misma discretización espacial que el ruido neutrónico en el dominio de la frecuencia para obtener resultados coherentes.

Cabe señalar que el resultado de cada una de estas simulaciones es la distribución espacial del ruido neutrónico $\left(\frac{\delta\phi}{\phi_0}\right)$ para la perturbación y frecuencia seleccionadas. Un ejemplo el flujo de neutrones estáticos y el ruido de neutrones producido por un cambio de la sección transversal de dispersión en el conjunto (8, 4) se muestra en las Figuras 4.2 y 4.3.

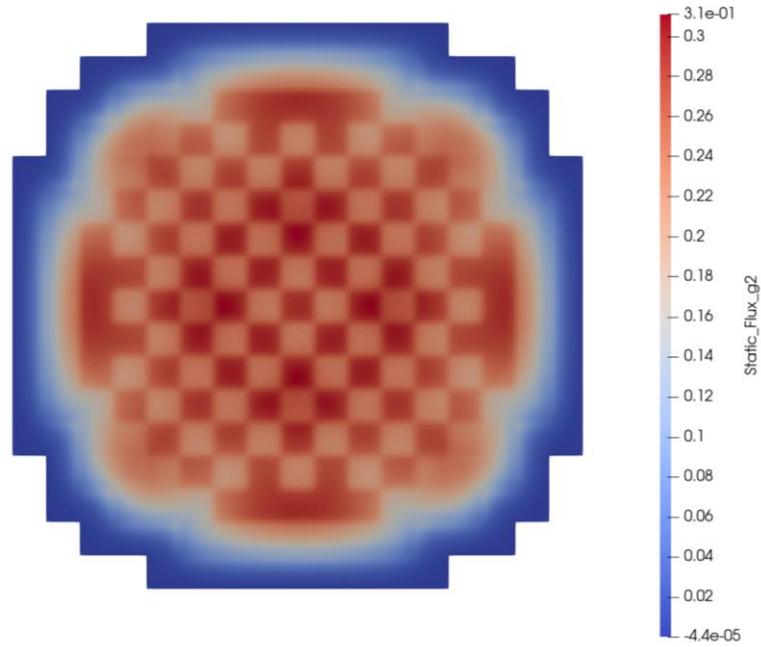


Figura 4.2. Flujo Estático.

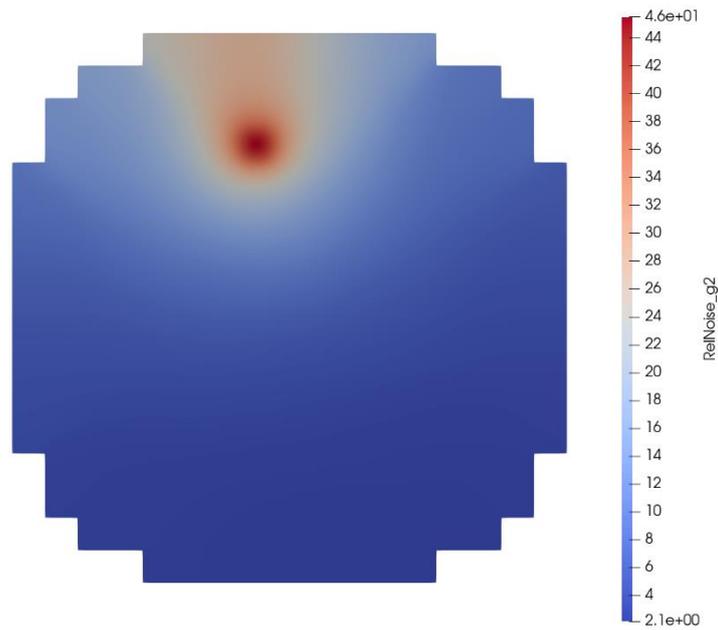


Figura 4.3. Ruido neutrónico.

Para cada perturbación, se obtiene un archivo que contiene las siguientes matrices de datos:

- Potencia neutrónica en cada plano.
- Flujo neutrónico rápido (ϕ_1) en cada plano.
- Flujo neutrónico térmico (ϕ_2) en cada plano.

- Ruido neutrónico rápido ($\delta\phi_1$) en cada plano.
- Ruido neutrónico térmico ($\delta\phi_2$) en cada plano.

A partir de estas señales, la red neuronal funcionará con el ruido neutrónico relativo, para el que emplea el flujo y ruido térmicos. De esta forma, la función de transferencia del detector se establece como se especifica en la ecuación (4).

$$\text{Señal del detector} = \frac{|\delta\phi_2|}{\phi_2}. \quad (4)$$

4.2.2. Preparación de los datos

En este proyecto se estudia también el mínimo número de detectores activos necesario para que las perturbaciones sean clasificadas y localizadas correctamente. Esto también es importante, ya que en la aplicación real no se ubican detectores en todos los elementos combustibles. Por ello, se aplican un total de 13 máscaras distintas de detectores activos (Figura 4.4), que nos permiten observar el efecto que éstas tienen sobre la precisión de la clasificación y localización de la perturbación. Los detectores activos se definen con el valor 1, mientras que si no están activados muestran el valor 0. De este modo, cuando la máscara es aplicada, solo habrá un conjunto de elementos que contendrán valores distintos de cero.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 4.4. Ejemplo de distribución de 8 detectores activos.

Además, en “*deep learning*”, la cantidad y calidad de los datos que se utilizarán para entrenar una red es fundamental para obtener resultados satisfactorios. Por esta razón, las técnicas de ampliación de datos suelen emplearse cuando los datos originales son limitados [7][8]. En este proyecto se ha agregado ruido a las señales de entrada como método de “*data augmentation*”. El tipo de ruido que se agrega a las señales es ruido blanco gaussiano (Additive White Gaussian Noise, AWGN). La potencia de ruido se fija en un porcentaje de la potencia de la señal, que va desde el 5% al 25% con incrementos del 2.5%.

El proceso de agregar AWGN a una señal implica varios pasos. Primero, la potencia de la señal original P_s se calcula tomando el valor cuadrático medio de la señal, siendo n el número de elementos de las señales (5).

$$P_s = \frac{1}{n} \sum_{k=1}^n \left(\frac{\delta\phi_2(k)}{\phi_2(k)} \right)^2 \quad (5)$$

A continuación, se determina el porcentaje de ruido que se desea añadir (l), que se utiliza para calcular la potencia de ruido P_n (6),

$$P_n = lP_s. \quad (6)$$

Entonces, se genera ruido gaussiano blanco como una distribución normal con media de 0 y una desviación estándar de 1, $\varphi = \mathcal{N}(0, 1)$. Luego, la potencia del ruido generado se ajusta para que coincida con el porcentaje deseado, $\bar{\varphi}$, (7):

$$\bar{\varphi} = \sqrt{\frac{P_n}{\frac{1}{n} \sum_{k=1}^n \varphi(k)^2}} \varphi, \quad (7)$$

y la amplitud del ruido se escala por un factor correspondiente al nivel de ruido deseado, φ_l , (8).

$$\varphi_l = l\bar{\varphi}. \quad (8)$$

Finalmente, el ruido se agrega a la señal original, dando como resultado una señal con AWGN añadido, s_φ (9).

$$s_\varphi = \frac{\delta\phi_2(k)}{\phi_2(k)} + \varphi_l \quad (9)$$

En las Figuras 4.5 y 4.6 se puede apreciar la forma de las señales con el máximo nivel de ruido para ambos tipos de perturbación y frecuencia 0,1 Hz. Después de agregar el ruido, todas las señales se estructuran como 2-matrices dimensionales que se proporcionarán como entrada a las CNN bidimensionales [9].

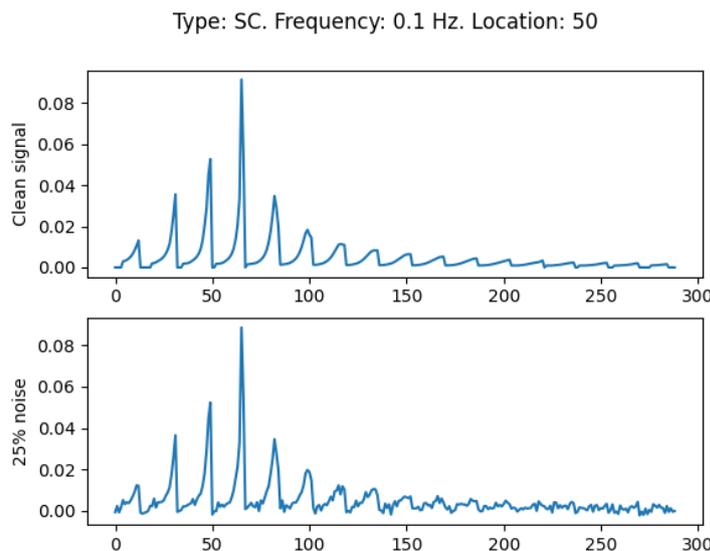


Figura 4.5. Señal limpia y señal ruidosa para la perturbación tipo 'scattering'.

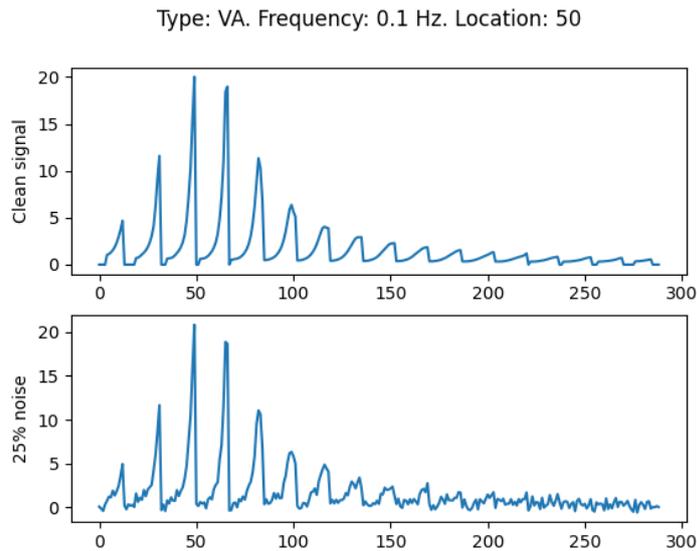


Figura 4.6. Señal limpia y señal ruidosa para la perturbación tipo absorción.

Una vez que se procesan todos los datos de entrada, el conjunto completo de datos se divide aleatoriamente en tres grupos: un 15% para test, un 12% como conjunto de validación en la fase de entrenamiento, y el 73% restante como conjunto de entrenamiento.

4.2.3. Diseño de Redes Neuronales Convolucionales

Los modelos de redes neuronales convolucionales (CNN) han sido desarrollados y entrenados usando la librería Keras [10] en una GPU Nvidia A40.

Debido a que la geometría del reactor es tridimensional, los modelos de CNN a emplear pasan a ser de 3D, ya que estos nos permiten introducir los datos como matrices de 17x17x19 elementos. No obstante, dependiendo la máscara de detectores activos que se ha aplicado, parte de los elementos de la matriz introducida contendrán valor 0.

Los diseños de las CNN empleados para ambos entrenamientos han sido tomados del caso para el reactor 2D. Diversas capas convolucionales con activaciones “ReLU” y “Batch Normalization” dan paso a capas “Max Pooling” para reducir la cantidad de datos. Como paso final se introducen 2 capas “fully connected” activadas con “ReLU” y sobre las que se aplica “Batch Normalization” y la técnica de “dropout”.

Para realizar las clasificaciones de los datos que salen de las capas unidimensionales “fully connected” se emplea la función de activación “softmax”. La activación “sigmoid” únicamente nos permite hacer una clasificación cuando solo hay 2 tipos, por lo que no nos sirve para nuestro caso en 3D donde hemos añadido un tipo más de perturbación. La Figura 4.7 muestra el modelo de entrenamiento para clasificar el tipo de perturbación, mientras que la Figura 4.8 muestra el entrenamiento para la localización de la perturbación

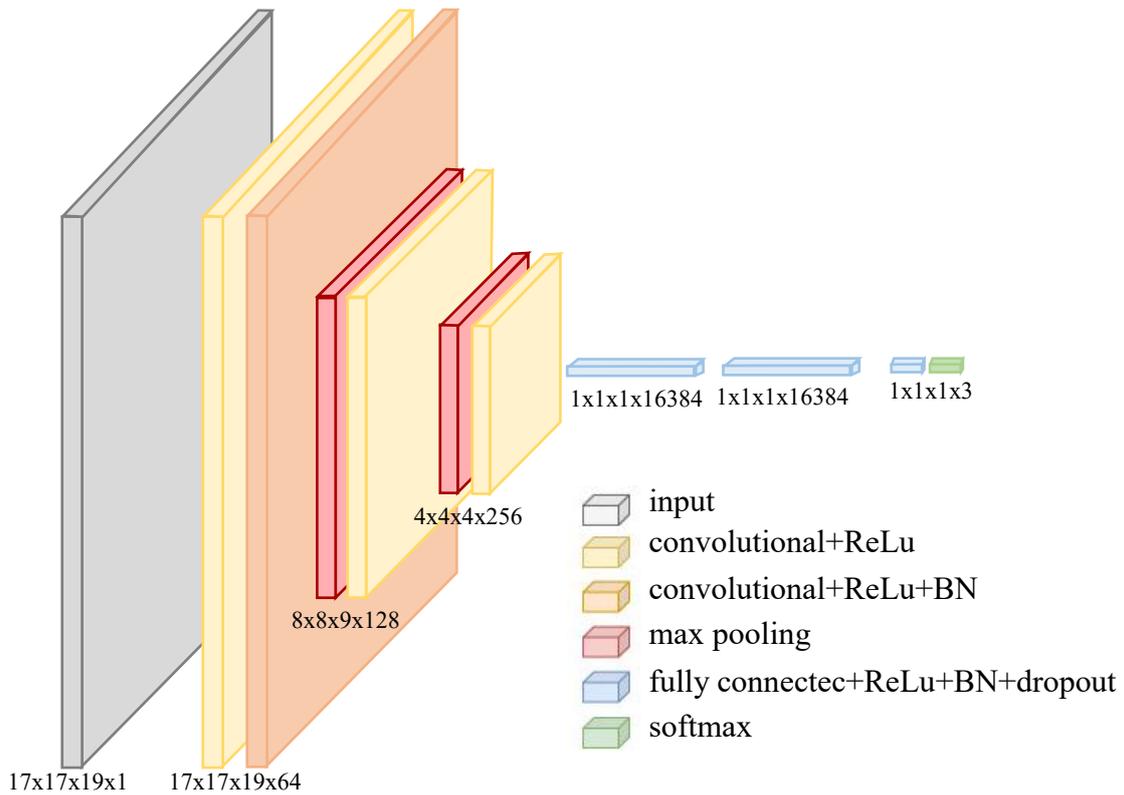


Figura 4.7. CNN para clasificar el tipo de perturbación.

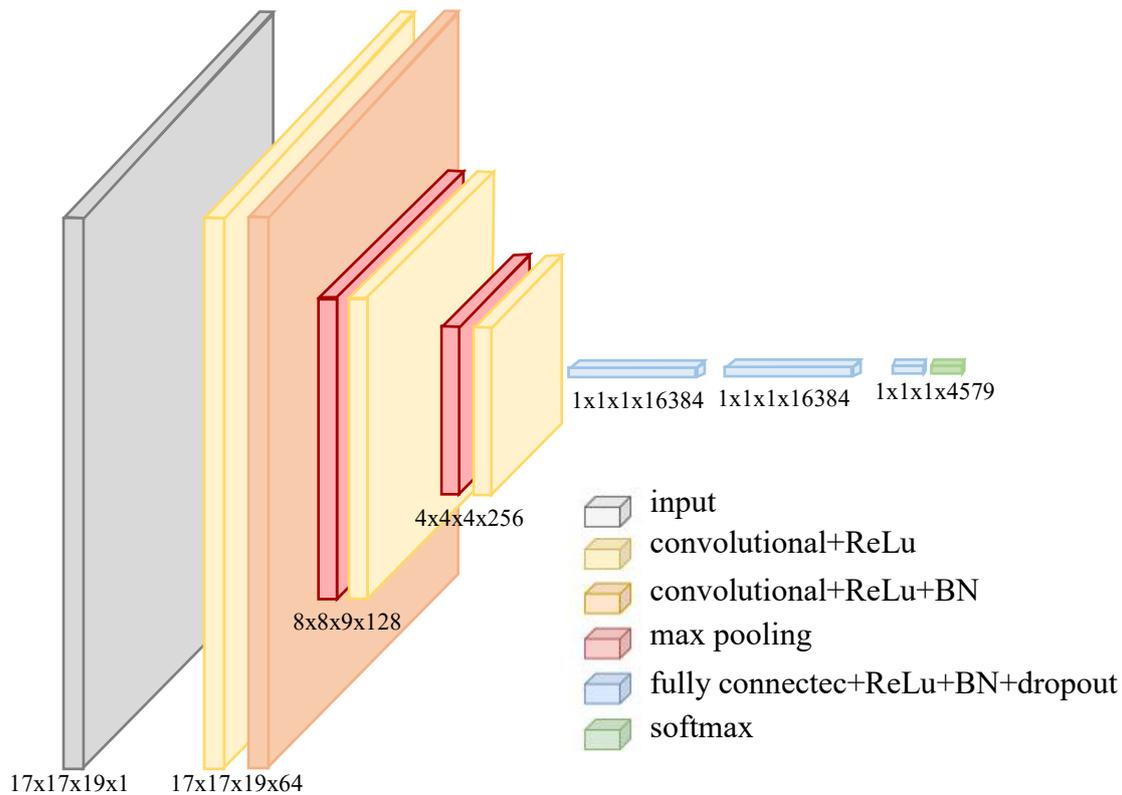


Figura 4.8. CNN para localizar la perturbación.

4.3. RESULTADOS NUMÉRICOS

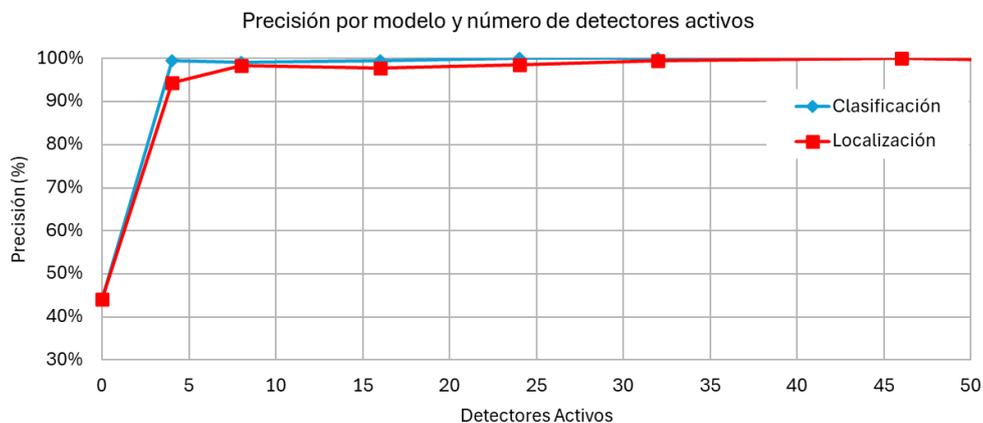
Una vez entrenadas las CNN el modelo pasa a ser validado mediante el 12% de los datos que se habían guardado. Tras la validación, el modelo es puesto a prueba mediante test con el 15% de los datos que se habían guardado para tal efecto. De este último proceso se obtienen los resultados de precisión de nuestros modelos, que además se muestran en función de cada una de las máscaras de detectores activos que se han introducido.

En la Tabla 4.1 se recogen todos los resultados de precisión por modelo de entrenamiento (clasificación y localización) y por número de detectores activos. Todos ellos se muestran para un porcentaje de ruido blanco gaussiano del 25%. Como se observa en ella, la tendencia es aumentar la precisión conforme aumenta el número de detectores activos, comportamiento que es de esperar. Además, puede observarse que la precisión obtenida con 4 detectores activos en todo el volumen del reactor supera el 90% tanto para clasificar el tipo de perturbación como para localizarla. Si bien las precisiones son cercanas al 100%, el modelo de clasificación de la perturbación alcanza más rápidamente este valor que el modelo de localización. Resultado que es obvio puesto que la clasificación es a 3 tipos mientras que la localización se extiende a 4579 posiciones.

Tabla 4.1. Valores de precisión de los modelos por número de detectores activos.

Detectores Activos	Precisión	
	Clasificación	Localización
0	44,18%	44,19%
4	99,39%	94,34%
8	99,11%	98,26%
16	99,57%	97,80%
24	100%	98,47%
32	100%	99,48%
46	100%	100%
64	100%	98,72%
82	100%	99,97%
97	100%	100%
128	100%	100%
184	100%	100%
241	100%	100%

Todo ello puede observarse también en las curvas de la Figura 4.9, en la cual se muestra la tendencia que presentan ambos modelos de alcanzar el 100% de precisión. No obstante, no es factible utilizar 241 detectores activos para asegurar que la precisión para ambos casos sea siempre del 100%, se debe escoger un número óptimo que mantenga una elevada precisión pero que sea factible aplicarse



a la realidad.

Figura 4.9. Gráficas de precisión de los modelos de entrenamiento.

A vista de los resultados, puede decirse que con 8 detectores activos es suficiente para clasificar y localizar las perturbaciones, ya que presenta una precisión del 99.11% para identificar el tipo de anomalía y del 98.26% para localizarla. La evolución a partir de este número de detectores no es significativa, por ello se escoge como distribución óptima.

4.4 CONCLUSIONES

En el proyecto se ha extendido el uso de las CNNs para un reactor bidimensional y para un reactor tridimensional. En concreto, se ha empleado el modelo de reactor 3D-IAEA, sobre el que se han introducido 3 tipos distintos de anomalías sobre cada uno de los 4579 elementos que lo componen. Las perturbaciones introducidas se han dividido en: pequeñas variaciones sobre las secciones eficaces de absorción y dispersión para 5 frecuencias de oscilación distintas y cambios en la velocidad del flujo del refrigerante para 5 velocidades distintas y 5 frecuencias de oscilación. A todos los datos generados se les ha aplicado una preparación previa a ser pasados para el entrenamiento, constando este de la aplicación de diversas máscaras de detectores activos distribuidos, así como de la introducción de ruido blanco gaussiano (AWGN).

A su vez, se han aplicado los diseños de CNN ya modelizados para el reactor 2D con la finalidad de estudiar su aplicabilidad en 3D. A pesar de haber realizado cambios menores para ajustar la convolución a tridimensional, los resultados han mostrado tan buenas precisiones como lo hicieron estos modelos en el caso bidimensional. Si bien estos modelos de CNN tienen un amplio margen de mejora, han demostrado ser una herramienta muy útil para facilitar la detección de las perturbaciones de los reactores. Este podría ser el punto de partida para desarrollar una herramienta que ayude a los técnicos de las centrales nucleares en el seguimiento del estado del reactor y asegurar una detección temprana de posibles anomalías.

Puesto que el tipo de anomalías introducidas hasta el momento son limitadas, para trabajos futuros se propone extender el número de perturbaciones sobre los que entrenar los modelos. Además, resultaría de gran interés aplicar estos entrenamientos sobre reactores más avanzados, como los *Small Modular Reactors*, de cara a su utilidad más futura.

REFERENCIAS

- [1] T. Tasakos, G. Ioannou, V. Verma, G. Alexandridis, A. Dokhane y A. Stafylopatis. "Detección de anomalías basada en aprendizaje profundo en núcleos de reactores nucleares". En Actas de la Conferencia Internacional sobre Matemáticas y Métodos Computacionales Aplicados a la Ciencia e Ingeniería Nucleares (M&C 2021), En línea, págs. 3 a 7 (2021).
- [2] A. Papaoikonomou, J. Wingate, V. Verma, A. Durrant, G. Ioannou, T. Papagiannis, M. Yu, G. Alexandridis, A. Dokhane, G. Leontidis, S. Kollias, and A. Stafylopatis. "Deep learning techniques for in-core perturbation identification and localization of time-series nuclear plant measurements." *Annals of Nuclear Energy*, volume 178, p. 109373 (2022). URL <https://www.sciencedirect.com/science/article/pii/S0306454922004078>.
- [3] D. A. Ejigu and X. Liu. "Gradient descent-particle swarm optimization based deep neural network predictive control of pressurized water reactor power." *Progress in Nuclear Energy*, volume 145, p. 104108 (2022). URL <https://www.sciencedirect.com/science/article/pii/S0149197021004601>.

- [4] A. Vidal-Ferrandiz, A. Carreño, D. Ginestar, and G. Verdú. "FEMFFUSION: A finite element method code for the neutron diffusion equation." <https://femffusion.webs.upv.es/> (2022).
- [5] A. Vidal-Ferrandiz, A. Carreño, D. Ginestar, and G. Verdú. "Edge-wise perturbations to model vibrating fuel assemblies in the frequency-domain using FEMFFUSION: Development and verification." *Annals of Nuclear Energy*, volume 175, p. 109246 (2022).
- [6] A. Carreño, A. Vidal-Ferrándiz, D. Ginestar, and G. Verdú. "Frequency-domain models in the SPN ϵ approximation for neutron noise calculations." *Progress in Nuclear Energy*, volume 148, p. 104233 (2022).
- [7] L. Taylor and G. Nitschke. "Improving deep learning with generic data augmentation." In 2018 IEEE symposium series on computational intelligence (SSCI), pp. 1542–1547. IEEE (2018).
- [8] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu. "Data augmentation for deep learning-based radio modulation classification." *IEEE access*, volume 8, pp. 1498–1506 (2019).
- [9] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning." *nature*, volume 521(7553), pp. 436–444 (2015).
- [10] F. Chollet et al. "Keras." <https://keras.io> (2015).

5. DIFUSIÓN DE RESULTADOS

ARTÍCULOS PUBLICADOS EN REVISTAS DE INVESTIGACIÓN:

- Chillarón, M.; Vidal-Ferràndiz, A.; Vidal, V.; Verdú, G. **Classification and Location of Neutron Noise Perturbations Using Convolutional Neural Networks**. Nuclear Science and Engineering, 2024.
<https://www.tandfonline.com/doi/full/10.1080/00295639.2024.2357395>

CONGRESOS/ JORNADAS:

- Verdú Martín, G. **Mejoras en las nuevas técnicas de inteligencia artificial para la detección de anomalías en reactores nucleares (INAIA)**. Jornada Anual de I+D+i del CSN. 2 de febrero del 2023.
- Chillarón-Pérez, Mónica; Vidal-Ferràndiz, Antoni; Vidal-Gimeno, Vicente Emilio; Verdú-Martín, Gumersindo Jesús. **Classification and location of neutron noise perturbations using CNNs**. International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2023). Niagara Falls, Canadá, 17/08/2023.
- Verdú, G.; Vida, A.; Chillarón, M.; Vidal, V. **Mejoras en nuevas técnicas de inteligencia artificial para la detección de anomalías en reactores nucleares**. 48ª Reunión Anual de la Sociedad Nuclear Española, 4 – 6 de octubre de 2023, Toledo, Castilla-La Mancha.
- Navarro, Empar; Vidal, Antoni; Verdú, Gumersindo. **Desarrollo de redes neuronales convolucionales para la identificación y localización de perturbaciones en reactores nucleares**. 50ª Reunión Anual de la Sociedad Nuclear Española, 9 – 11 de octubre de 2024, Córdoba, Andalucía.

TESIS EN DESARROLLO:

Nombre: Empar Navarro Gamón (en proceso)

Director/a: Gumersindo Jesús Verdú Martín

Título: Desarrollo de nuevas técnicas de IA enfocadas a la detección de anomalías en reactores nucleares.

Organismo: Universitat Politècnica de València