Study of Convolutional Neural Networks (CNNs) for Classifying and Locating Neutron Noise Perturbations

E. Navarro-Gamón[1,*], A. Vidal-Ferràndiz[2], M. Chillarón[3] and G. Verdú[1]

(1) Instituto de Seguridad Industrial, Radiofísica y Medioambiental,
Universitat Politècnica de València,
València, 46022, Spain.
(2) I. U. de Matemática Multidisciplinar,
Universitat Politècnica de València,
València, 46022, Spain.
(3) Dep. de Sistemes Informàtics i Computació,
Universitat Politècnica de València,
València, 46022, Spain.

1 Introduction

Throughout the years, studying the effects of perturbations in nuclear reactors performance has become a crucial issue to assure the safety of these reactors and consequently, guarantee their long-term operations. With the help of in-core instrumentation, neutron flux fluctuations, called neutron noise, can be analyzed, and provide information of the performance of the reactor core. By studying the neutron noise, we can localize and detect anomalies or perturbations in the nuclear reactor using deep learning techniques.

A clear example of the successful use of deep learning is presented by Tasakos et al. [1], whose work studies the use of Convolutional Neural Networks for classifying and locating perturbations in the core of a 3-loop Swiss pre-KONVOI pressurized water reactor (PWR), using SIMULATE-3K for creating several types of perturbations.

He-Lin Gong et al. [2] identify the location of detectors as an important source of improvement in core monitoring and CNN developing. They propose a Voronoi tessellation to obtain a discrete grid-structure representation of the distribution of in-core instrumentation and feed with it a CNN model. What is suggested in this work allows to study the optimal location of the sensors in order to increase the detection accuracy.

This work contributes developing three different CNN models to study the perturbations and their locations, using synthetic data generated from the BIBLIS 2D benchmark but adding neutron noise expressed as variations in cross-sections values. To present our work the document is distributed as follows. First, the methodology used for the three training models is explained, that includes the analysis of neutron noise in the frequency domain and the description of the CNNs structures to identify and locate the perturbations. Then, to verify the methodology, the results of accuracy for every model and the distribution of the detectors are presented. This section leads finally to several conclusions that are summarized in the last section.

* emnaga@upvnet.upv.es

2 Methodologies

2.1 Frequency domain first order neutron noise equation

Among the different techniques used to solve the diffusion equation, the frequency-domain solution has proven to be successful [3]. The methodology used to obtain the neutron noise distribution is based on the method implemented in the finite element method code FEMFFUSION [4]. Developed by UPV, this code has introduced a tool, FEMFFUSION-FD, to solve the 2-energy groups diffusion approximation in the frequency-domain, using the first-order approximation of the neutron noise.

Taking the first-order neutron noise as a split of the time dependent term $U(\vec{r},t)$ into its mean steady state value $U_0(\vec{r})$ and its fluctuation around the mean value $\delta U(\vec{r},t)$, it can be expressed as:

$$U(\vec{r},t) = U_0(\vec{r}) + \delta U(\vec{r},t). \tag{1}$$

Assuming the fluctuations of the time dependent term are small compared to the mean values, the second-order terms can be neglected as well as any fluctuation of the diffusion coefficients $(\delta D_g = 0)$. The validity of such approximations is already demonstrated for light water reactor in several studies [5]. Then, the Fourier transform can be applied to the neutron diffusion equation resulting in:

$$-\vec{\nabla}\left(D\vec{\nabla}\delta\phi(\vec{r},\omega)\right) + \Sigma_{dyn}\delta\phi(\vec{r},\omega) = \delta S(\vec{r},\omega). \tag{2}$$

That can be written as [6]:

$$\mathcal{A}\delta\phi = \mathcal{B}\phi_0, \tag{3}$$

where

$$\mathcal{A} = \begin{pmatrix} \frac{i\omega}{v_1} - \vec{\nabla}D_1\vec{\nabla} + \Sigma_{a1}^0 + \Sigma_{12}^0 - \gamma\nu\Sigma_{f1}^0 & -\gamma\nu\Sigma_{f2}^0 \\ -\Sigma_{12}^0 & \frac{i\omega}{v_2} - \vec{\nabla}D_2\vec{\nabla} + \Sigma_{a2} \end{pmatrix}, \delta\phi = \begin{pmatrix}\delta\phi_1 \\ \delta\phi_2\end{pmatrix},$$
$$\mathcal{B} = \begin{pmatrix} -\delta\Sigma_{a1} - \delta\Sigma_{12} + \gamma\delta\nu\Sigma_{f1} & \gamma\nu\Sigma_{f2} \\ \delta\Sigma_{12} & -\delta\Sigma_{a2} \end{pmatrix}, \phi_0 = \begin{pmatrix}\phi_1 \\ \phi_2\end{pmatrix} \tag{4}$$

and

$$\gamma = (1-\beta) + \sum_{k=1}^{K}\frac{\lambda_k\beta_k}{i\omega+\lambda_k}. \tag{5}$$

Which depicts an in-homogeneous equation with complex coefficients that needs to be solved first in steady state using the steady state fast flux ϕ_1 and thermal flux ϕ_2. Then, the

spatial discretization used in the frequency domain neutron noise equation has to be applied to the static eigenvalue problem in order to obtain logical results.

Applying the continuous Galerkin finite element discretization [6, 7] to Eq. (2), an algebraic linear system of equations is reached with the structure shown in Eq. (6). In it as in Eq. (4), $\delta\phi_1$ and $\delta\phi_2$ are, respectively, the algebraic fast and thermal neutron noise weight vectors, and δS_1 and δS_2 the noise source for both energy groups.

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} \delta\phi_1 \\ \delta\phi_2 \end{pmatrix} = \begin{pmatrix} \delta S_1 \\ \delta S_2 \end{pmatrix}, \tag{6}$$

By only introducing the specific frequencies of the noise source to the frequency-domain neutron equation, a solution is delivered, since only a linear system of equations in the complex domain need to be solved. Unlike in the frequency-domain approximation, the time-dependent equation approach calculates for each time step, which leads to a great amount of linear systems to solve and, consequently, increasing considerably the computational time. To solve the complex linear system, FEMFFUSION uses the PETSc library [8] with the GMRES solver, giving the complex $\delta\phi_1$ and $\delta\phi_2$.

These results are normally used to express the relative neutron noise $|\delta\phi_1|_{REL}$ for the fast neutrons and $|\delta\phi_2|_{REL}$ for the thermal neutrons, which is an evaluation of the fast and thermal neutron noise amplitudes $|\delta\phi_1|$ and $|\delta\phi_2|$, respectively, through the steady state fast and thermal neutron fluxes ϕ_1 and ϕ_2:

$$|\delta\phi_1|_{REL} = \frac{|\delta\phi_1|}{\phi_1}, \; |\delta\phi_2|_{REL} = \frac{|\delta\phi_2|}{\phi_2}. \tag{7}$$

The frequencies introduced to create the noise are chosen according to the typical vibration and thermal-hydraulic anomalies frequencies, and are in the range from 0.1 Hz to 10 Hz, in particular the chosen frequencies were 0.1, 0.5, 1.0, 5.0 and 10 Hz. Each of these frequencies are applied to every assembly and, since the anomalies studied are variations in absorption (Σ_a) and scattering (Σ_s) cross-sections, a total of 2570 files are generated to train the CNN models. However, in-core monitors only detect the thermal relative noise thus, only the values of the thermal flux and thermal noise are useful.

2.2 Convolutional Neural Networks Design

The preprocessed data with the added neutron noise feeds 3 different CNN models, all of them inspired in the VGG16 network [9], a multiclass image classification. In the case of our work, the parameters and hyperparameters are updated according to the necessities of our data, and the number of hidden layers is selected experimentally to reduce computational time but maintaining the capability to extract discriminative features.

Taking the data and geometry from the BIBLIS 2D benchmark as in Figure 1, a total of 257 assemblies distributed in a checkerboard shape are introduced to the models as 2-dimensional 17x17 meshes so the spatial information is maintained. The effect that the number of the active detectors has in the accuracy results is evaluated using 13 different distributions

of active detectors, introduced to the models as 17x17 matrices. This study also allows to determine the minimum number of detectors needed in order to get the optimal model with high reliability.

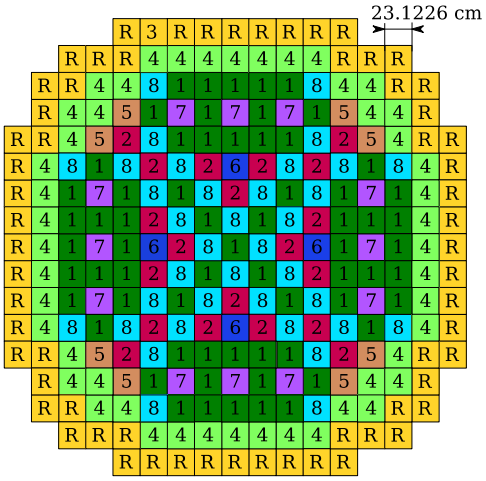

Figure 1. BIBLIS 2D checkerboard distribution for different 2-group homogenized materials.

To increase the reliability of the models, *Additive White Gaussian Noise* (AWGN) is added to the data as a tool to introduce the possible inaccuracies in detection that the sensors could commit. This technique is also valid to generate more data to the training so the overfitting is avoided and is called data augmentation. The process of adding the AWGN starts with calculating the power of the original signal P_s, using the mean squared value of the signal:

$$P_S = \frac{1}{n}\sum_{k=1}^{n}\left(\frac{\delta\phi_2(k)}{\phi_2(k)}\right)^2, \tag{8}$$

being n the number of elements in the matrices. Next, the percentage of the noise level, l, is added so the noise power P_n is calculated:

$$P_n = lP_s. \tag{9}$$

The white Gaussian noise is generated as a normal distribution $\varphi = \mathcal{N}(0,1)$, and applied to adjust the noise power to the wanted percentage $\bar{\varphi}$:

$$\bar{\varphi} = \sqrt{\frac{P_n}{\frac{1}{n}\sum_{k=1}^{n}\varphi(k)^2}}\,\varphi. \tag{10}$$

Finally, the noise amplitude is scaled to match the desired noise level, φ_l and added to the original signal:

$$\varphi_l = l\bar{\varphi},\ s_\varphi = \frac{\delta\phi_2(k)}{\phi_2(k)} + \varphi_l. \tag{11}$$

For these trainings the percentages of AWGN introduced are in the range between 5% to 25% in increments of 2.5%, meaning that the dataset will be increased 10 times, the original values and the 9 from the white noise.

The CNNs Sequential structures to identify and locate the perturbation are shown in Figure 2 and Figure 3, respectively. They show that, after the data is read, several 2D convolutional layers with *ReLu* activation and *Batch Normalization* (BN) are applied, followed by a *Max Pooling* layer. Finally, two stages of *fully connected* layers with a *drop out* are added. The last step of a CNN is to transform the shape of the output into a one-dimensional vector to classify into the desired number of classes. In Figure 2 the activation used is a *sigmoid*, which is useful when there are only 2 classes, as is our case. Meanwhile, the multiclass classification is done with a *softmax* activation, as can be seen in Figure 3, resulting in a vector of 257 classes of neurons.

Similarly, the third CNN model uses the same layers structure but are applied to a Keras *Model class* instead of a *Sequential class*, which allows to identify and locate the perturbations at the same time instead of individually.

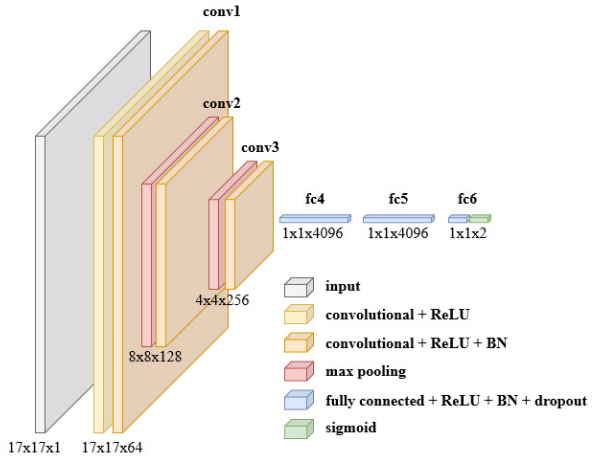

Figure 2. CNN structure to identify the type of perturbation.

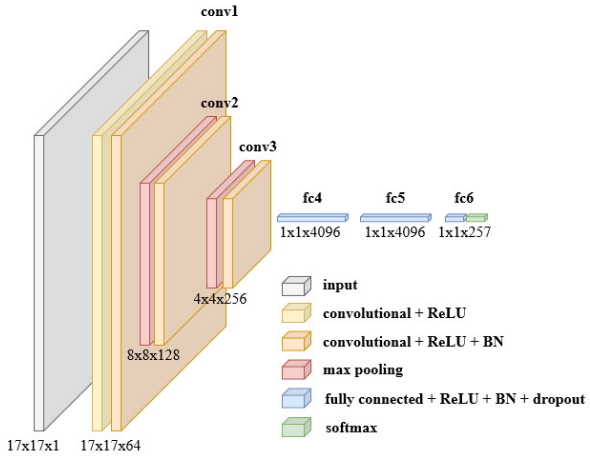

Figure 3. CNN structure to locate the perturbation.

With respect to the hyperparameters, the values for the *learning rate* and *drop out* are selected experimentally, and the *batch size* is maintained in 32. The optimizer used is *Adam* and the loss functions used are *binary cross-entropy* for type classification and *categorical cross-entropy* to locate the perturbations.

The training is done using the Keras library [10] on a GPU Nvidia A40 and the *early stopping* tool is used to save computational time.

3 Numerical Results

Once the trainings are completed, the CNNs perform the validation and test of the trained models. The initial data set is split keeping 15% of the data for testing the model and 12% for the validation. Then, the remaining 73% of the data is provided for the training.

The validation is done immediately after the training to evaluate the model and tune the hyperparameters, with an unbiased perspective. Similarly, the test uses unseen data to evaluate the final model after the validation, returning the accuracy trend values for each *epoch* and the final model accuracy for every configuration of detectors.

In Table I the accuracy results for the 3 trained CNN models are depicted. It can be seen that with 4 active detectors, accuracies of 90% are reached and the 100% accuracy is easily achieved in the classification of the type of perturbation. However, getting the 100% accuracy in the location training is only feasible for higher numbers of active detectors. These tendencies are also seen in the coupled training, where the CNN trains, simultaneously, to identify and locate the perturbations. Figure 4 compares visually the exponential behavior of the accuracy when the number of active detectors increases.

Table I.
CNN accuracy results for each training model and detector distribution.

Active detectors	**Type of perturbation**	**Location of perturbation**	**Coupled training**	
			Type	**Location**
0	50.01%	0.39%	48.72%	0.28%
4	98.31%	90.35%	83.58%	91.10%
8	99.95%	98.52%	100%	97.04%
16	100%	99.04%	100%	99.69%
24	100%	99.56%	100%	99.69%
32	100%	99.69%	100%	99.77%
46	100%	99.71%	100%	99.79%
64	100%	99.97%	100%	99.97%
82	100%	100%	100%	100%
97	100%	100%	100%	100%
128	100%	100%	100%	100%
184	100%	100%	100%	100%
257	100%	100%	100%	100%

Notwithstanding, introducing active detectors in every assembly to assure the highest accuracy is not realistic nor necessary. The number of active detectors and their distribution has to be optimal, using the minimum number of detectors that give a high enough accuracy. In general terms, it can be said that the developed models are well-designed and meet the needs of the problem in hand, partially thanks to the variability of the data due to the AWGN addition. Moreover, the training has been done for a range of specific frequencies, meaning that the validity of these CNN training models cannot be assured for other frequencies of neutron noise.

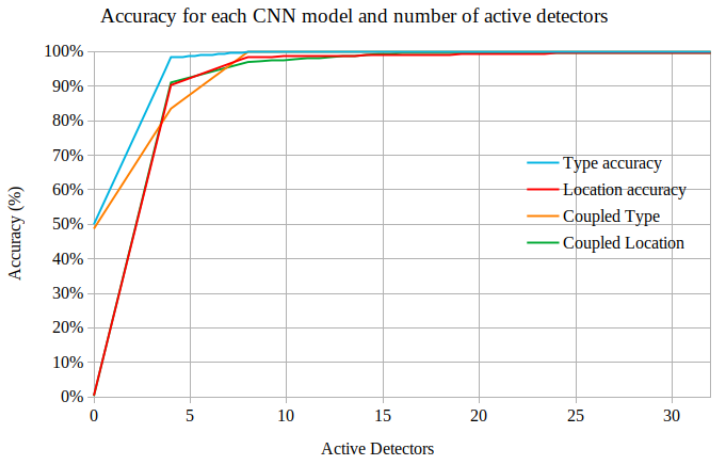

Figure 3. Accuracy for each CNN model and active detectors

4 Conclusions

In this work, a study to use Convolutional Neural Networks to detect anomalies in neutron flux is carried out. To achieve this aim, a specific data preprocessing was done to the initial data from the BIBLIS 2D benchmark, which involved adding neutron noise for different frequencies. As a result, a dataset of 2570 neutron noise output files are obtained, one for each type of perturbation and its location and also for every noise frequency introduced. Then, the dataset is fed to the CNN models and AWGN is added to widen the data and avoid overfitting.

As is previously shown, the accuracy of the chosen architectures is directly dependent on the number of active detectors, getting closer to guessing the 100% of the cases when augmenting the number of active detectors. Nonetheless, the optimal number of working sensors chosen for this problem can be said that is 8 due to the fact that the accuracy for the perturbation classification is already 99.95% and for locating the perturbation is 98.52%. In the case of the coupled training, the accuracy for the 8 active detectors distribution is similar to the obtained with the individual training, 100% for classifying and 97.04% for locating the perturbation. Then, the most suitable CNN model is the coupled training, for the classification and location are simultaneously obtained.

Therefore, due to the good results reached with this research it can be concluded that the developed CNN models are valid techniques to create future tools based in Deep Learning, complementing the current monitoring and the tasks that technicians in nuclear power plants practice. Currently, these CNN models are being applied to 3D models and different types of perturbations are being considered. For future works, it is planned to use this models to study its usefulness in advanced nuclear reactors such as Molten Salt Reactors Small Modular Reactors.

Acknowledgements

This research was supported by “Universitat Politècnica de València” and “ Consejo de Seguridad Nuclear” under the project “Cátedra CSN-UPV en Seguridad Nuclear y Protección Radiológica”, developed in “Instituto de Seguridad Industrial, Radiofísica y Medioambiental”.

References

[1] Tasakos, T., Ioannou, G., Verma, V., Alexandridis, G., Dokhane, A., and Stafylopatis, A., Deep learning-based anomaly detection in nuclear reactor cores. *Proceedings of the international conference on mathematics and computational methods applied to nuclear science and engineering (M&C)*, 54: 2026-2037, 2021.

[2] He-Lin Gong, Han Li, Dunhui Xiao, Sibo Cheng, Reactor field reconstruction from sparse and movable sensors using Voronoi tessellation-assisted convolutional neural networks. *Nuclear Science and Techniques,* 35, 43, 2024.

[3] Demazière, C., and Pázsit, I., Numerical tools applied to power reactor noise analysis. *Progress in Nuclear Energy*, 51: 67-81, 2009.

[4] Vidal-Ferràndiz, A., Carreño, A., Ginestar, D., and Verdú, G., FEMFFUSION: A finite element method code for the neuron diffusion equation. *https://femffusion.webs.upv.es/*, 2022.

[5] Larsson, V., and Demazière, C., Comparative study of 2-group 1P and diffusion theories for the calculation of the neutron noise in 1D 2-region systems. *Annals of Nuclear Energy*, 36: 1574-1587, 2009.

[6] Vidal-Ferràndiz, A., Carreño, A., Ginestar, D., and Verdú, G., Edge-wise perturbations to model vibrating fuel assemblies in the frequency-domain using FEMFFUSION: Development and verification. *Annals of Nuclear Energy*, 175, 109246, 2022.

[7] Vidal-Ferràndiz, A., Ginestar, D., Carreño, A., Verdú, G., Dokhane, A., Verma, V., Perin, Y., Herb, J., Mylonakis, A., Demazière, C., and Vinai, P., Modelling and simulations of reactor neutron noise induced by mechanicalvibrations. *Annals of Nuclear Energy*, 177, 109300, 2022.

[8] Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E.M., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W.D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M.G., Kong, F., Kruger, S., May, D. A., L. McInnes, L.C., Mills, R.T., Mitchell, L., Munson, T., Roman, J. E., Rupp, K., Sanan, P., Sarich, J., Smith, B. F., Zampini, S., Zhang, H., Zhang, H., and Zhang, J., PETSc: Portable, Extensible Toolkit for Scientific Computation. *https://petsc.org/*, 2024.

[9] Simonyan, K., and Zisserman, A., Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations*, 1-14, 2015.

[10] Chollet, F., et al., Keras. *https://keras.io*, 2015.