

A Deep Learning Approach to Anomaly Detection in Nuclear Reactors

Francesco Calivá* and
Fabio De Sousa Ribeiro*
School of Computer Science
MLearn Group

University of Lincoln
LN67TS, Lincoln, UK

{fcaliva, fdesousaribeiro}@lincoln.ac.uk

*Both authors contributed equally.

Antonios Mylonakis,
Christophe Demazière
and Paolo Vinai

Chalmers University of Technology
Division of Subatomic and Plasma Physics
Department of Physics

SE-412 96 Gothenburg, Sweden

{antmyl, demaz, vinai}@chalmers.se

Georgios Leontidis and
Stefanos Kollias
School of Computer Science
MLearn Group

University of Lincoln
LN67TS, Lincoln,

United Kingdom

{gleontidis, skollias}@lincoln.ac.uk

Abstract—In this work, a novel deep learning approach to unfold nuclear power reactor signals is proposed. It includes a combination of convolutional neural networks (CNN), denoising autoencoders (DAE) and k -means clustering of representations. Monitoring nuclear reactors while running at nominal conditions is critical. Based on analysis of the core reactor neutron flux, it is possible to derive useful information for building fault/anomaly detection systems. By leveraging signal and image pre-processing techniques, the high and low energy spectra of the signals were appropriated into a compatible format for CNN training. Firstly, a CNN was employed to unfold the signal into either twelve or forty-eight perturbation location sources, followed by a k -means clustering and k -Nearest Neighbour coarse-to-fine procedure, which significantly increases the unfolding resolution. Secondly, a DAE was utilised to denoise and reconstruct power reactor signals at varying levels of noise and/or corruption. The reconstructed signals were evaluated w.r.t. their original counter parts, by way of normalised cross correlation and unfolding metrics. The results illustrate that the origin of perturbations can be localised with high accuracy, despite limited training data and obscured/noisy signals, across various levels of granularity.

Index Terms—deep learning, convolutional neural networks, clustering trained representations, denoising autoencoders, signal processing, nuclear reactors, unfolding, anomaly detection.

I. INTRODUCTION

The monitoring of nuclear reactors while running at nominal conditions is crucial and advantageous. In fact, by analysing measured fluctuations of process parameters, such as the neutron flux, it is possible to gather valuable insight into the functionality of the core and subsequently the detection of anomalies at an early stage ([1], [2]). These fluctuations are generally referred to as noise and can be denoted as in (1), where $X(\mathbf{r}, t)$ represents the signal and $X_0(\mathbf{r}, t)$ its trend. Both are a function of two variables: \mathbf{r} which is the spatial coordinate, i.e. location in the core, and t , the time.

$$\delta X(\mathbf{r}, t) = X(\mathbf{r}, t) - X_0(\mathbf{r}, t) \quad (1)$$

Causes for these fluctuations are multiple and can relate to mechanical vibrations of internal reactor components, the turbulent character of the flow within the core, the coolant boiling, and to a smaller extent, the stochastic character of

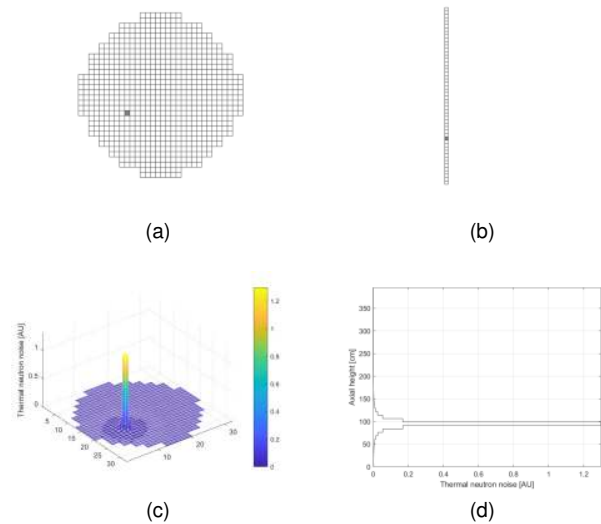


Fig. 1. Illustration of a CORE SIM simulation related to the thermal energy group. The sub-plots at the top are exemplary of the radial (a) and axial (b) positions of the noise source within the reactor core. The sub-plots at the bottom are exemplary of the radial (c) and axial (d) distributions of the amplitude of the corresponding induced neutron noise.

nuclear reactions. In order to model how the fluctuations affect the neutron flux, dedicated core simulators can be employed to perform simulations either in the time or frequency domain. These models accept as input information related to physical perturbations, the probabilities of neutron interactions within the core, along with the description of the geometry of the reactor. Once this data are known/given, the reactor transfer function can be calculated. Consequently, the neutron noise induced by the applied perturbation can be estimated such that the so called *forward problem* can be solved. Conversely, the *backward problem* refers to the localisation of the fluctuation/s origin, and can only be retrieved if the reactor transfer function can also be inverted. The latter is also known as the process of *unfolding*. Nevertheless, solving the *unfolding problem* (hereafter shorted as *unfolding*) is non-trivial as it would

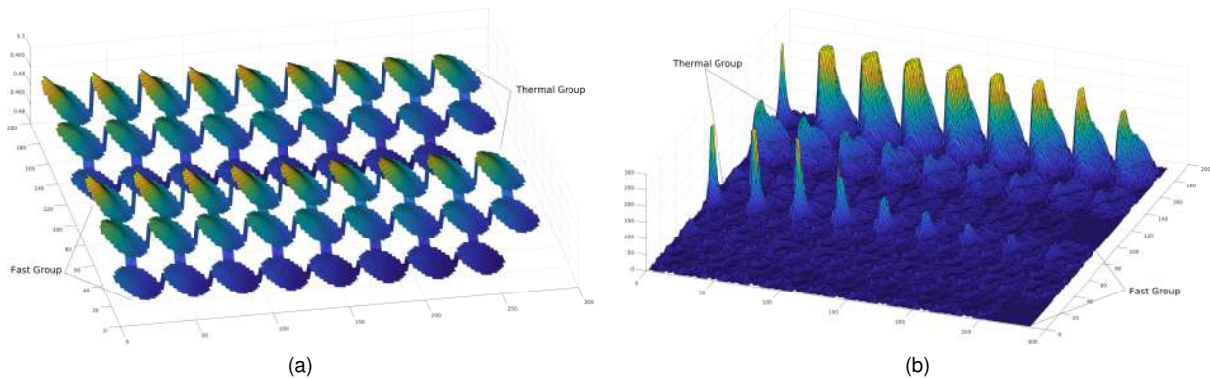


Fig. 2. Thermal (top three rows) and fast (bottom three rows) group response to an in-core Dirac's like perturbation. The twenty-six layers of the reactor are unrolled into a two-dimensional image. For each point, the height of the spike is representative of the induced noise measured in that particular point. **a:** Signals phase. **b:** Signals amplitude. **(a)** is shown in \log_{10} scale.

require measurements of the induced neutron noise at every position inside the reactor core. This is not possible as in reality, reactors have a limited number of in- and out- core sensors able to measure fundamental parameters.

Considering the scarcity of previous research, in this work a novel method to unfold, denoise and reconstruct the signal is proposed. This is achieved by introducing appropriate signal analysis techniques and using Deep Neural Network (DNN) architectures to localise the origin of perturbations in reactors.

II. RELATED WORK

Few studies can be found in academic literature addressing the problem of fault detection in nuclear reactors. Current work follows either model-driven or data-driven approaches. Most notably, in [3], auto-associative kernel regression and sequential probability ratio tests were combined to monitor sensors' conditions. If an anomaly was detected, the system would be able to reconstruct the measurements of faulty sensors. An anomaly detection framework based on symbolic dynamic filtering and associated pattern classification was proposed in [4]. This was optimised by appropriate partitioning of sensor time series. In [5], critical heat flux was predicted by means of an Adaptive Neuro-Fuzzy Inference System. In [6], artificial neural networks were implemented to diagnose transients, based on reactor process parameters. In [7], a combination of principal component analysis and fisher discriminant analysis was proposed for fault detection in nuclear reactors.

Given the outburst in popularity of deep learning, a vast amount of research has recently been published presenting new techniques ([8]–[12]). In [13], a Convolutional Neural Network (CNN) and Naïve Bayes data fusion scheme was proposed for the detection of fractures in plant components by way of the analysis of individual video frames.

To the best of the authors' knowledge, this is the first study in which a deep learning based system is utilised to solve the unfolding, denoising and reconstructing of signals representative of the core responses to perturbations at different frequencies. This is based on the analysis of the thermal and fast groups of the neutron flux responses.

III. THE EXAMINED SCENARIO

Core simulators are able to perform simulations in both time [14] and frequency domains [2]. Former simulations provided a description of how a nuclear core behaves as a function of time, at all possible locations throughout the core and also for the in- and out- core neutron sensors. Although the information extends to a defined but flexible period of time, such tools were not primarily developed for modelling the effect of very small perturbations (i.e. noise). The latter, on the other hand, were specifically designed to model the effect of small stationary fluctuations, and have the ability to describe the distribution of the induced neutron noise within the whole core reactor. In this study, data relating to an absorber of variable strength in the frequency domain have been used. This data, representative of a scenario where the neutron noise is induced in a Pressurised Water Reactor (PWR), was generated by means of CORE SIM [2]. During the forward problem, the reactor transfer function thoroughly captures the response of the neutron flux, which is induced by the known distribution of perturbations. If the perturbations reduce to a Dirac function applied to the point (\mathbf{r}_p) at a given angular frequency (ω), then the transfer function is the Green's function of the system ([1], [15]). Considering that the effect of the perturbation can be assessed from any spatial point \mathbf{r} , the induced neutron noise can be measured as in (2), where V refers to the volume of the reactor core.

$$\partial\phi(\mathbf{r}, \omega) = \int_V G(\mathbf{r}, \mathbf{r}_p, \omega) dS(\mathbf{r}_p) d\mathbf{r}_p \quad (2)$$

From (2), it can be perceived that when the neutron flux is measurable at any single location throughout a reactor core, the Green's function, depicted in (2), gives a one-to-one relationship between every possible location of a Dirac-like perturbation and every single position where the induced neutron noise can be measured. The estimation of the Green's function thus represents an ideal case of unfolding, since there are as many possible locations of the noise source as possible locations of the induced noise.

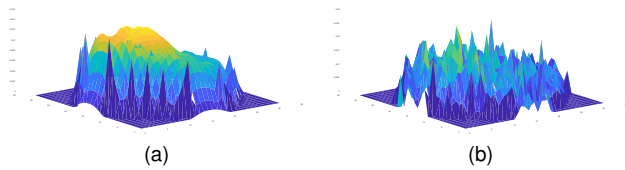


Fig. 3. **a**: Thermal group response to a Dirac’s like perturbation. **b**: The same signal of **(a)** is affected by the noise at $SNR = 1$. For each point, the height of the spike is representative of the induced noise measured in that particular point.

A. Simulated Data Generation

Absorber of Variable Strength in the Frequency Domain:

In this study, CORE SIM was used to estimate a spatially-discretised Green’s function in the frequency domain (2) [16]. CORE SIM applies diffusion theory to perform a low-order approximation of the angular moment of the neutron flux, which the scalar neutron flux and net neutron currents can be determined from. Regarding the discretisation of energy, a two-energy group formulation was used: one with a high energy spectrum, hereafter referred to as the fast group, and the other with a low spectrum, i.e. the thermal group. Moreover, based on linear theory, the calculations of the induced neutron noise were carried out using a first-order approximation of the neutron noise.

Given a spatial discretisation of the reactor core in three dimensions, CORE SIM computed the Green’s function (2). In this scenario, the noise source is defined as the perturbation of the thermal macroscopic absorption cross-section, which characterises the ability of a material to absorb thermal neutrons (see Fig. 1). Further calculations were computed for all possible sources of noise within the core, to determine the spatially-discretised form of the Green’s function. In each set, three different frequencies were used: 0.1, 1 and $10Hz$. The PWR adopted in this work consisted of a radial core of size 15×15 fuel assemblies, in which axial and bottom reflectors were also explicitly taken into account. A volumetric mesh with voxels of dimension $32 \times 32 \times 26$, with nodes of size $\Delta x = 10.75$ cm, $\Delta y = 10.75$ cm and $\Delta z = 15.24$ cm was utilised for calculations. For more details related to CORE SIM, please refer to the official user’s manual ([2], [16]).

B. Data Pre-processing

The output of CORE SIM is a 3D representation of the induced neutron noise. It can be considered as an ideal scenario where a detector (sensor) and thus the noise signal is available at each voxel of the core volume. Moreover, the calculated output can be seen as a clean signal that carries only the information related to the noise produced by a Dirac’s like perturbation. The simulation output consists of the fast and the thermal neutron responses. Specifically, these are complex signals distributed in the form of a three-dimensional array of size $32 \times 32 \times 26$, with each complex signal containing a perturbation at differing coordinate points i, j, k (considered the label) within the volume. The dataset is comprised of

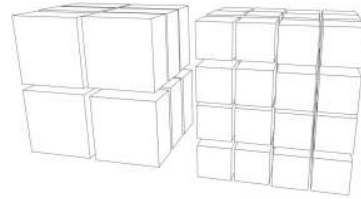


Fig. 4. Volumetric splitting used to generate the twelve and forty-eight unfolding labels. The $32 \times 32 \times 26$ volume was compartmentalised into twelve and forty-eight volumetric subsections by a factor of $2 \times 2 \times 3$ and $4 \times 4 \times 3$ respectively.

19552 instances per frequency (0.1, 1 and $10Hz$). For the purpose of learning a meaningful representation from the data, a conversion procedure was devised to unroll each volume into a two-dimensional form. This conversion was independently repeated for the amplitude and phase of each signal. Lastly, the values were rescaled conforming to a range between 0 and 255. Fig. 2 depicts the thermal and fast group response to an in-core perturbation. The signal measured in each layer of the reactor (from the bottom moving upward in the core) is unrolled in a two-dimensional image where the height of the spikes is representative of the induced noise measured in that particular point. To make the signal more realistic, it was processed and corrupted by adding disturbing noise. Additionally, to be more representative of the fact that in reality, fewer in-core sensors are available, parts of the signal were also obscured.

1) *Noise Addition*: White Gaussian noise (WGN) was added to the signal at two distinct signal-to-noise ratios ($SNR = 1$ and $SNR = 3$) producing two versions of noisy data. To ensure that the perturbation was influenced by the introduction of the noise, this was added individually to each slice (depth-wise) of the core volume. Figure 3 depicts how the signal is affected by the noise ($SNR = 1$) and in fact, the relatively larger amplitude of the induced noise in the vicinity of the perturbation on the left-hand side of the image is no longer easily discernible.

2) *Obscuring Signals*: Two versions of obscured data were produced, and for each of them, three thresholds of data maintenance were adopted (25%, 50% and 75%). In the first version, for each output volume of the CORE SIM simulation, a random 25% (50%, and 75% respectively) of the values were maintained and the remaining were set to zero. In the second version, 25% (50%, and 75% respectively) of the sensors were randomly selected and their signal kept. Conversely to the former, in the latter version, the active sensors were randomly selected only once at the beginning of the experiment, and the measurements from these same sensors were kept all across the dataset. An identical data-obscuring procedure was likewise applied to the signals to which noise of $SNR = 1$ and $SNR = 3$, was previously added.

IV. THE PROPOSED APPROACH

In the following section, the proposed deep learning approach for detection of the induced neutron noise in the above reactor environments is presented. Firstly, the desired network

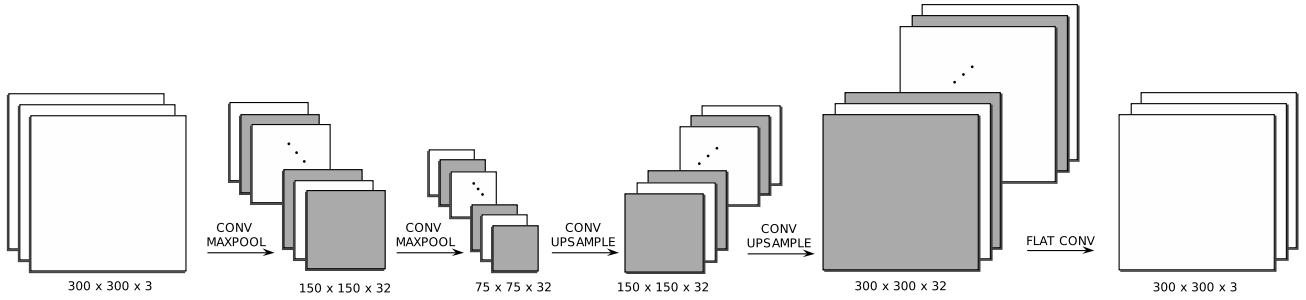


Fig. 5. Depiction of the DAE architecture assembled to solve the corrupted signal reconstruction problem. From left to right, the input image is fed into the first convolutional layer, consisting of 32 (3×3) filters, *same* padding and ReLU activations. The resulting volume of dimensions $300 \times 300 \times 32$, is then fed into a max pooling layer with a kernel size 2×2 and *same* padding, resulting in a $150 \times 150 \times 32$ compressed volume. This process is repeated using the same layer parameters, until the first upsampling layer, where both the volume rows and columns are instead repeated by a factor of 2. The final convolutional layer consists of 3 (3×3) filters, *same* padding and a sigmoid activation function. The output is a volume of the same dimension as the original input image.

outputs are defined and obtained through volumetric splitting of the complex signals. Subsequently, a CNN is trained to perform the unfolding, also introducing a novel hierarchical clustering approach of CNN derived representations. Lastly, a DAE approach to reconstruct and unfold corrupted signals is proposed.

A. Volumetric Splitting

Given the measurements of the induced neutron noise within the reactor, it was possible to localise the source of each perturbation inside a well defined region. Specifically, the $32 \times 32 \times 26$ signal volume array was compartmentalised into either twelve volumetric subsections by a factor of $2 \times 2 \times 3$ or, forty-eight subsections by a factor $4 \times 4 \times 3$. Each subsection was then utilised to generate labels for the experiments (see section V). Fig. 4 is exemplary of the label splits. Through splitting we illustrate the proposed coarse-to-fine unfolding approach, which could also be extended to provide finer unfolding resolutions.

B. Convolutional Neural Networks

CNNs take as input three channelled images and perform automatic feature extraction through a series of volume-wise convolutions and feature routing. To create an appropriate dataset to be fed into a CNN, the two-dimensional transformation of the data, as described in section III-B, was stacked into three channels (RGB). The amplitude and phase of both the thermal and fast groups were concatenated to preserve the integrity of the data, as these groups are the components in which the signal spectrum was discretised by CORE SIM. The first two channels are identical and contain the amplitude of the thermal and fast groups concatenated. The third channel consists of the phase of the two groups concatenated.

The CNN architecture of choice was Inception-V3 [17], due to its high capability to trainable parameters ratio when compared to other architectures such as InceptionResNetV2 [17] or VGG19 [18]. Furthermore, it is important to note that given the modest size of the dataset, a larger network is more likely to overfit as it contains more trainable parameters. For a detailed

description of the Inception-V3 architecture, please refer to its original paper [17].

It was of particular interest to firstly conduct transfer learning and assess the adaptability of pre-trained Inception-V3 Imagenet weights on the dataset. Specifically, each three channelled transformed image was fed through the network up until the last pooling layer, where a 2048 vector representation was extracted for each instance. The 2048 dimensional vectors were then used as input to a new series of fully-connected layers and a softmax classification layer of either twelve or forty-eight classes depending on the experiment. Prior to training, the dataset underwent one more pre-processing step in which the images were zero-padded to the target dimension of the CNN ($299 \times 299 \times 3$). This ensured that the integrity of the signal was preserved whilst accommodating the CNN convolutional layer parameters and arithmetic.

In order to optimise the performance of the new fully-connected layer network to be trained on the problem at hand, a series of architectural decisions were made through experimentation. The best performance was achieved with a fully-connected network consisting of two 2048 unit hidden layers with Rectified Linear Unit ($ReLU \rightarrow f(x) = \max(0, x)$) activations. Furthermore, Dropout [19] was used as an effective regulariser, with the probability of keeping individual neurons ($n \neq 0$) in each hidden fully-connected layer ($l^{[1]}$ to $L - 1$) set to $P(n) = 0.5$. To preserve more information in the input layer ($l^{[0]}$) of the network and thus aid learning [19], the keep probability was instead set to $P(n) = 0.8$.

In view of the unbalance present when splitting the signal volumes into forty-eight classes, it was advantageous to use weighted categorical cross entropy as a loss function (3), to encourage the model to focus on under-represented classes. In (3), the term ω_j (4) is a weight coefficient computed for the j^{th} of all classes J as a function of the proportion of instances N_j compared to the most densely populated class; x and \hat{x} are ground-truth and predicted source respectively.

$$\mathcal{L}(x, \hat{x}) = - \sum_{j=1}^J \omega_j x \log(\hat{x}) \quad (3)$$

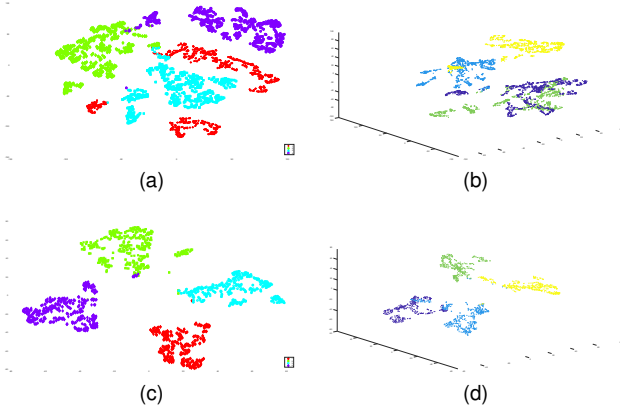


Fig. 6. t -SNE visualisation of k -means ($k = 4$) of the seventh block. The obtained training set clusters are (a-b) and the test set predictions are shown in (c-d). Each point is a lower dimensional projection of 2048 dimensional vector representations of signals, and each colour indicates a different cluster. Images on the left hand side are 2D visualisations, and those on the right the relative 3D visualisations.

$$\omega_j = \frac{\max(\{N_i\}_{i=1:J})}{N_j} \quad (4)$$

C. Increasing Resolution through Clustering

The generation of labels for the dataset, which involved volumetric splitting (see section IV-A), requires a sufficient amount of perturbation examples per class to be trained and classified. Intuitively, by increasing the granularity of the volumetric splitting, one is effectively reducing the number of training instances per class (volumetric subsection). It is therefore prudent to retain as much training data as possible, albeit at the cost of decreased prediction origin granularity. Granting that in a real scenario it is in-feasible to have sufficient measurements per every individual point in the volume, an optimal solution is one which maximises class granularity whilst maintaining an adequate amount of instances per class.

With that in mind, a methodology was devised to artificially increase the resolution of a given prediction by way of clustering instances belonging to individual blocks. Formally, given extracted $N^{[L-1]}$ -dimensional activations $(\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n)$, from the last fully-connected layer $L - 1$ (of L total layers), as latent variable representations of n total input images, the objective function in (5) clusters them into k sets $C = \{C_1, C_2, \dots, C_k\}$ as to minimise within-cluster L^2 norms.

$$\arg \min_C \sum_{i=1}^k \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \quad (5)$$

To achieve this, the first step was to utilise the CNN previously trained on the unfolding, in order to extract 2048 dimensional vector representations from the final average pooling layer. This vector is a compressed, but useful representation obtained through the forward propagation of each image through an already trained network. Therefore, rather than unrolling each original $299 \times 299 \times 3$ image into a long vector, it was

TABLE I
SETTINGS AND RESULTS OF THE UNFOLDING EXPERIMENTS.

Classes	CNN Unfolding				
	Sensors (%)	Signal	Train/Dev/Test (%)	Accuracy	
				Pretrained	Scratch
12	100	clean	75-10-15	97	99.9
	100	SNR=3	75-10-15	88.7	99.9
	100	SNR=1	75-10-15	84.2	98
	25	clean	50-20-30	93.7	99.9
	25	clean	25-15-60	93.4	98.4
48	25	SNR=1	50-20-30	76.6	94.1
	100	clean	75-10-15	92.3	99.9
	100	SNR=1	75-10-15	72.9	92.5
	25	clean	50-20-30	90.3	97.8
	25	clean	25-15-60	85.1	91.1
	25	SNR=1	50-20-30	65.2	82.3

advantageous to utilise the above representations learnt by the network during training, for clustering. Let us consider the task of increasing resolution from twelve to forty-eight classes. Each training image was first fed to the above trained CNN to compute the respective 2048 dimensional representation. Then, each derived representation referring to a corresponding noise perturbation location in one of the twelve original classes, was included in one out of four clusters generated per original class, through the use of the k -means algorithm. Lastly, the centroids of all these forty-eight sub-clusters were calculated. During testing, all data were fed to the trained CNN and their respective representations were classified using a nearest-neighbour method to one of the forty-eight centroids. The result of this classification procedure, is essentially the unfolding at a finer resolution - one fourth of that obtained by the CNN network on twelve classes. This procedure can be extended, or continued, to perform the unfolding at finer resolutions. The implementation of the k -means algorithm followed the k -means++ seeding strategy. Rather than randomly sampling initial centroids from available points, k -means++ employs a heuristic/probabilistic approach which leads to improvements in running times and better final solutions [20]. For visualisation, t -Stochastic Neighbour Embedding (t -SNE) was used as it provides accurate structure revealing maps of high-dimensional data [21] in lower dimensions, such as in 2D or 3D.

D. Denoising Autoencoder

An autoencoder is a type of network designed to copy its input to the output, rather than mapping it to a particular label. Like autoencoders, DEAs are comprised of an encoder and a decoder network. The encoder is responsible for the compression and encoding of a corrupted input $f(\hat{\mathbf{x}})$. The decoder then upsamples the encoding back to the input dimensions, and this procedure forces the network to learn useful properties of the data. During training, a loss function such as mean squared error (6) is minimised by penalising the reconstructed input $g(f(\hat{\mathbf{x}}))$ relative to how similar it is to the original input \mathbf{x} .

$$mse = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - g(f(\hat{\mathbf{x}}_i)))^2 \quad (6)$$

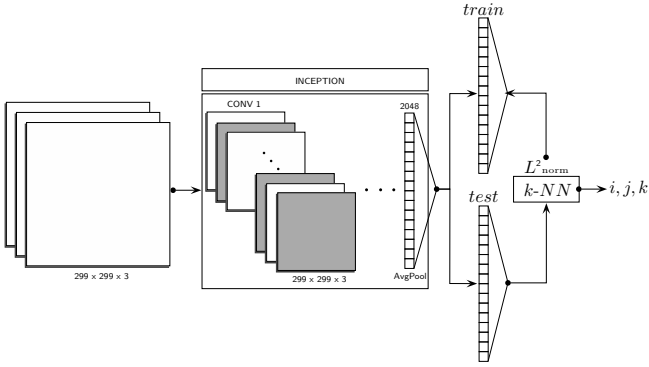


Fig. 7. Architectural depiction of the proposed k -NN procedure to predict perturbation sources (i, j, k) at the original signal resolution of $32 \times 32 \times 26$.

In pursuance of learning useful properties from data, a Denoising Autoencoder (DAE) was utilised as a means of reconstructing a corrupted input signal. The fundamental difference between a traditional autoencoder and a DAE is that the former learns the identity function of the input whereas the latter is forced to learn a denoising function w.r.t the input. It was therefore evident that this property of DAEs is especially useful for learning to reconstruct signals which are either noisy or have been obscured. In all cases, the pre-processing and signal transformation stages discussed in section III-B were used. The signals are treated as three channelled two-dimensional images, as to allow for convolution operations in order to retain valuable spatial information, rather than unrolling each image into a vector.

A concrete depiction of the architectural parameters can be seen in Fig. 5. The network is comprised of five convolutional layers, four of which have 32 (3×3) filters and ReLU activations. The final convolutional layer includes 3 filters of size 3×3 and a sigmoid (σ) activation function in order to reconstruct an image of identical dimensions to the input.

Moreover, two max pooling layers of filter size 2×2 were used to reduce the representation and finally produce a $75 \times 75 \times 32$ encoding layer. The decoder follows the same pattern but in reverse, as to upsample the encoding volume back to the input size. Lastly, *same* padding was implemented throughout to retain the spatial dimension of the volumes after convolving, also known as flat convolution. Lastly, the autoencoder was trained to minimise the mean squared error (mse) presented in (6) along with Adaptive Moment Estimation (Adam) optimisation to include adaptive learning rate, momentum, RMSprop and bias correction in weight updates, which helps to obtain faster convergence rate than normal Stochastic Gradient Descent with momentum [22].

V. EXPERIMENTAL STUDY

Two sets of experiments were carried out. In the first set, the CNN and the proposed clustering methodology were employed to solve the unfolding problem. In the second, the proposed DAE was utilised to reconstruct a complete clean signal starting from obscured signals, and to filter out noise.

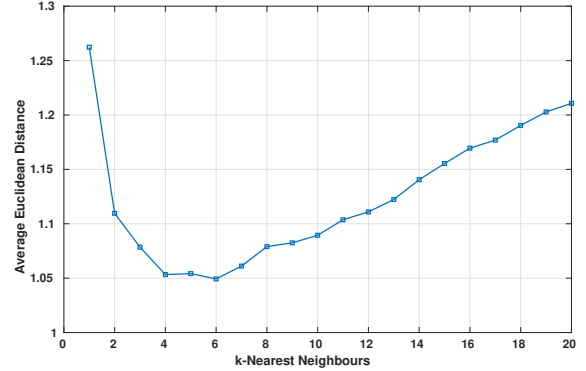


Fig. 8. Reaching a very fine unfolding resolution through a k -Nearest Neighbour ($k = 6$) algorithm utilising the L^2 norm as a distance metric.

Post DAE training, the reconstructed signals were predicted by the CNN for unfolding.

The experiments were run under different train/validation/test data split configurations: 75-10-15%, 50-20-30% and 25-10-15% respectively. The motivation was that in a real scenario, one would never have complete quantitative information regarding the noise induced in the core, and so by limiting the amount of training data, it is possible to mitigate learning from an unrealistic dataset. As a direct implication, the DNN should inherently have the ability to learn from a limited number of recorded instances, and be able to predict the induced noise in the occurrence of unseen scenarios. The implementation was based on MATLAB [23], Keras deep learning framework [24] and Tensorflow numerical computation library [25]. The experiments were conducted using a server with an Intel Xeon(R) E5-2620 v4 CPU, eight GPUs and 96GB of RAM.

A. CNN-based Unfolding

This experiment included two subsets, namely unfolding the signal to identify twelve and forty-eight possible perturbation sources respectively. Several further tests were performed, each of them involving different input data - as reported in Table I. The highest performance achieved on the twelve class test, by utilising pre-trained weights, with clean and complete signal input was 97% accuracy. On the other hand, the CNN trained from scratch performed better, achieving 99.9% in both the twelve and forty-eight classes experiments. Despite the reduction of the signal available (25% of the sensors active), training from scratch proved to be highly accurate in unfolding the signal to forty-eight classes regardless of the size of the training set. It achieved 97.8% and 87.3% accuracy when the training set consisted of half and a fourth of the entire dataset respectively. In the presence of noisy signal ($SNR = 1$), with 25% of active sensors, the accuracy achieved was 94.1% in twelve classes and 82.3% in forty-eight. Conversely, when retaining 100% of active sensors, the performance increased up to 98% and 92.5% for the twelve and forty-eight classes problem respectively. In continuation, as previously discussed in section IV-C a k -means clustering approach was devised,

TABLE II
SETTINGS AND RESULTS OF THE DAE EXPERIMENTS

Deep-CNN Autoencoder				
Sensors	Signal	Train/Test	Normalised Cross-Correlation	
			Clean vs Corrupted	Clean vs Reconstructed
75%	clean	25/75%	0.77	0.995
50%	clean	25/75%	0.57	0.995
25%	clean	25/75%	0.37	0.993
25%	SNR=1	25/75%	0.36	0.991

based on extraction of activations from the last fully-connected layer of the trained CNN. Figure 6 depicts 2D and 3D t -SNE visualisations of k -means ($k = 4$), belonging to the seventh of the twelve blocks. Each point corresponds to a 2048 dimensional vector representation of the original signal and each colour represents a different cluster. A respective test set prediction accuracy of 95.3% was achieved, indicating that very good results were obtained when increasing the unfolding resolution from twelve to forty-eight classes, either through CNN re-training or through the clustering approach.

In an extension of this study, a k -NN based approach was devised to perform the unfolding up to the original resolution of $32 \times 32 \times 26$ (see Fig. 7). Firstly, 2048 dimensional CNN representations were split into two separate sets (train/test) containing no overlapping perturbation locations (labels) between the sets. The perturbation location of each data point in the test set was then predicted by computing the mean μ of each triad of coordinates (w.r.t the original signal volume) belonging to its k nearest neighbour representations within the train set. Figure 8, shows that for $k = 6$, the unfolding procedure produced an excellent perturbation location (i, j, k) estimation accuracy, with the average error of just over 1 coordinate point in the reactor.

B. DAE-based Signal Filtering and Unfolding

In this experiment, signal denoising and reconstruction were performed. Table II lists the input data to the DAE for each test. The percentage of volume-wise maintained sensors accounted for 25-50-75% of the total; whereas the signal was either clean or corrupted with $SNR = 1$.

A further study was carried out to evaluate the performance of a combination of the DAE followed by a CNN classifier as described previously. Starting from the partially obscured and disturbed signals, the aim was to unfold the induced neutron noise to either twelve or forty-eight sources. The work-flow consisted of a denoising and reconstruction step performed by means of the DAE, and subsequent classification of the reconstructed signals. For the latter, a CNN model previously trained on the forty-eight classes problem with clean signals was utilised.

To ensure a superior generalisation of signal reconstruction in the experiments, the DAE training set size was limited to only 25%. In effect, this forced the autoencoder to learn to generalise to a much bigger test size proportion. Fig. 9 is exemplary of the reconstruction of a signal starting from 75

(left column), 50 (middle) and 25 (right column) % of the sensors. These depict the clean (top), corrupted (middle) and reconstructed (bottom) signals. The metric employed to measure the precision of the reconstruction was normalised cross-correlation (ncc , (7)), as it provides sub-pixel image matching evaluation precision [26]. Given two three-channelled images A and B , we can quantify their similarity per channel as

$$ncc = \frac{\sum_{i,j} (a_{i,j} - \mu_A)(b_{i,j} - \mu_B)}{[\sum_{i,j} (a_{i,j} - \mu_A)^2 \sum_{i,j} (b_{i,j} - \mu_B)^2]^{0.5}} \quad (7)$$

where $a_{i,j}$ and $b_{i,j}$ refer to each pixel in A and B with μ_A and μ_B as their mean pixel intensities per channel. The final ncc is the average of the three channels (RGB), and is a $\mathbb{R} \cap [-1, 1]$ computed solely on the portion of the image containing the signal, disregarding zero value padding. Table II reports the similarity of the reconstruction to the original signal. The average ncc of the reconstruction was in the worst case 0.991. In the *cascade* experiment, reconstructed signals were predicted by a previously trained CNN on the original (clean) signals. Given that the ncc coefficients of the reconstructed signals compared to the original were very close to 1, the CNN classification performance on the reconstructed signals was almost identical to the original results, as reported in table I.

VI. CONCLUSIONS AND FUTURE WORK

This paper proposes a novel method to solve the unfolding, denoising and reconstruction of signals with induced neutron noise in a pressurised water reactor. The data consisted of the core thermal and fast group responses to perturbations applied within the reactor, at differing frequencies 0.1, 1 and 10Hz, and comprising the knowledge of the noise signal at each voxel of the core volume.

The proposed solution was based on the coupling of a deep convolutional neural network with clustering of internal representations extracted from the trained CNN, combined with appropriate signal analysis methodologies. A very high accuracy was achieved in the unfolding throughout the experimental study, including the originally generated signals, as well as their respective noisy and obscured counterparts.

Moreover, very good results were also obtained through the proposed clustering of CNN extracted representations method to increase unfolding resolution. k -means based unfolding achieved 95.3% accuracy for four-way subdivisions of blocks belonging to the twelve classes. Furthermore, unfolding up to a very fine resolution was successfully achieved through the proposed k -NN based coarse-to-fine approach, reaching an average error of only 1 neighbouring coordinate point in the original $32 \times 32 \times 26$ reactor dimensions.

A Deep-CNN Denoising Autoencoder was also developed to denoise and reconstruct noisy reactor signals. Several experiments were successfully conducted, and comparatively evaluated using a Normalised Cross-Correlation coefficient criterion. It was shown that the reconstructed signals were very close approximations of the originals, and were thereafter used for unfolding of noisy and obscured data.

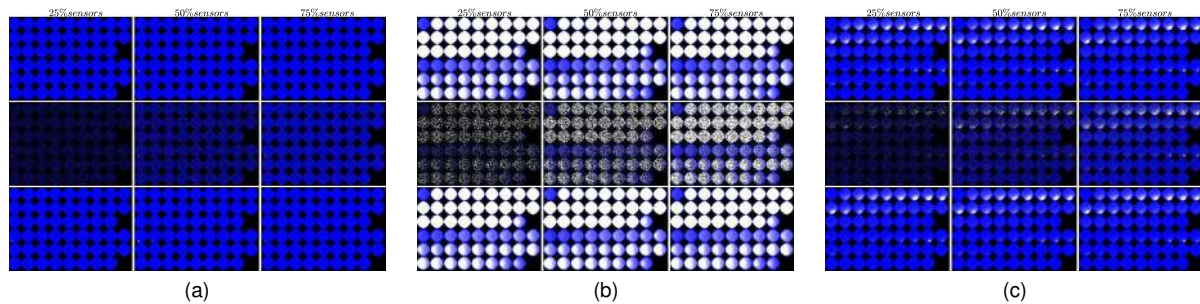


Fig. 9. Three examples (a-c) of the reconstruction performed with the DAE when: 75%, 50% and 25% of the sensors were used. For each of these: **Top:** Original signal. **Middle:** Obscured signal. **Bottom:** Reconstructed signal.

Our future work will extend the experimental study to other types of perturbations and signals generated in either the frequency or time domain, and will ultimately lead application on nuclear reactor data currently generated by the CORTEX EU Horizon 2020 project [27].

ACKNOWLEDGMENT

The research conducted was made possible through funding from the Euratom research and training programme 2014-2018 under grant agreement No 754316 for the 'CORe Monitoring Techniques And EXperimental Validation And Demonstration (CORTEX)' Horizon 2020 project, 2017-2021.

REFERENCES

- [1] Dan Gabriel Cacuci. *Handbook of Nuclear Engineering: Vol. 1: Nuclear Engineering Fundamentals; Vol. 2: Reactor Design; Vol. 3: Reactor Analysis; Vol. 4: Reactors of Generations III and IV; Vol. 5: Fuel Cycles, Decommissioning, Waste Disposal and Safeguards*, volume 2. Springer Science & Business Media, 2010.
- [2] Christophe Demazière. Core sim: a multi-purpose neutronic tool for research and education. *Annals of Nuclear Energy*, 38(12):2698–2718, 2011.
- [3] Wei Li, Min jun Peng, Ming Yang, Geng lei Xia, Hang Wang, Nan Jiang, and Zhan guo Ma. Design of comprehensive diagnosis system in nuclear power plant. *Annals of Nuclear Energy*, 109:92 – 102, 2017.
- [4] X. Jin, Y. Guo, S. Sarkar, A. Ray, and R. M. Edwards. Anomaly detection in nuclear power plants via symbolic dynamic filtering. *IEEE Transactions on Nuclear Science*, 58(1):277–288, Feb 2011.
- [5] Salman Zaferanlouei, Dariush Rostamifard, and Saeed Setayeshi. Prediction of critical heat flux using anfis. *Annals of Nuclear Energy*, 37(6):813 – 821, 2010.
- [6] T.V. Santosh, A. Srivastava, V.V.S. Sanyasi Rao, A.K. Ghosh, and H.S. Kushwaha. Diagnostic system for identification of accident scenarios in nuclear power plants using artificial neural networks. *Reliability Engineering System Safety*, 94(3):759 – 762, 2009.
- [7] Farhan Jamil, Muhammad Abid, Inamul Haq, Abdul Qayyum Khan, and Masood Iqbal. Fault diagnosis of pakistan research reactor-2 with data-driven techniques. *Annals of Nuclear Energy*, 90:433 – 440, 2016.
- [8] Dimitrios Kollias, Miao Yu, Athanasios Tagaris, Georgios Leontidis, Andreas Stafylopatis, and Stefanos Kollias. Adaptation and contextualization of deep neural network models. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 2017.
- [9] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 843–852. IEEE, 2017.
- [10] Fabio De Sousa Ribeiro, Francesco Caliva, Mark Swainson, Kjartan Gudmundsson, Georgios Leontidis, and Stefanos Kollias. An adaptable deep learning system for optical character verification in retail food packaging. In *Evolving and Adaptive Intelligent Systems, IEEE Conference on*, 2018.
- [11] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.
- [12] Dimitrios Kollias, Athanasios Tagaris, Andreas Stafylopatis, Stefanos Kollias, and Georgios Tagaris. Deep neural architectures for prediction in healthcare. *Complex & Intelligent Systems*, pages 1–13, 2018.
- [13] F. C. Chen and M. R. Jahanshahi. Nb-cnn: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion. *IEEE Transactions on Industrial Electronics*, 65(5):4392–4400, May 2018.
- [14] Gerardo Grandi. Validation of casmo5/simulate-3k using the special power excursion test reactor iii e-core. cold start-up, hot start-up, hot standby and full power conditions. Technical report, 2015.
- [15] Imre Pázsit and Christophe Demazière. Noise techniques in nuclear systems. In *Handbook of Nuclear Engineering*, pages 1629–1737. Springer, 2010.
- [16] Christophe Demazière. User's manual of the core sim neutronic tool. Technical report, Chalmers University of Technology, 2011.
- [17] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- [19] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.
- [20] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.
- [21] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.
- [22] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [23] Users Guide Matlab. The mathworks. Inc., Natick, MA, 1992, 1760.
- [24] François Chollet et al. Keras, 2015.
- [25] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.
- [26] Brian B Avants, Charles L Epstein, Murray Grossman, and James C Gee. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Medical image analysis*, 12(1):26–41, 2008.
- [27] Christophe Demazière, Paolo Vinai, Mathieu Hursin, Stefanos Kollias, and Joachim Herb. Noise-based nuclear plant core monitoring and diagnostics. *Proceedings of Advances in Reactor Physics, Mumbai, India*, 2017.