Towards a Deep Unified Framework for Nuclear Reactor Perturbation Analysis

Fabio De Sousa Ribeiro* and Francesco Calivá*
University of Lincoln, UK
{fdesousaribeiro,fcaliva}@lincoln.ac.uk
*Both authors contributed equally

Dionysios Chionis and Abdelhamid Dokhane
Paul Scherrer Institute
Villigen, Switzerland
{dionysios.chionis, abdelhamid.dokhane}@psi.ch

Antonios Mylonakis and Christophe Demazière
Chalmers University of Technology, Sweden
{antmyl,demaz}@chalmers.se

Georgios Leontidis and Stefanos Kollias
University of Lincoln
Lincoln, UK
{gleontidis,skollias}@lincoln.ac.uk

***Abstract*—In this paper, we take the first steps towards a novel unified framework for the analysis of perturbations in both the Time and Frequency domains. The identification of type and source of such perturbations is fundamental for monitoring reactor cores and guarantee safety while running at nominal conditions. A 3D Convolutional Neural Network (3D-CNN) was employed to analyse perturbations happening in the frequency domain, such as an absorber of variable strength or propagating perturbation. Recurrent neural networks (RNN), specifically Long Short-Term Memory (LSTM) networks were used to study signal sequences related to perturbations induced in the time domain, including the vibrations of fuel assemblies and the fluctuations of thermal-hydraulic parameters at the inlet of the reactor coolant loops. 512 dimensional representations were extracted from the 3D-CNN and LSTM architectures, and used as input to a fused multi-sigmoid classification layer to recognise the perturbation type. If the perturbation is in the frequency domain, a separate fully-connected layer utilises said representations to regress the coordinates of its source. The results showed that the perturbation type can be recognised with high accuracy in all cases, and frequency domain scenario sources can be localised with high precision.**

***Index Terms*—deep learning, 3D convolutional neural networks, recurrent neural networks, long short-term memory, multi label classification, regression, signal processing, nuclear reactors, unfolding, anomaly detection.**

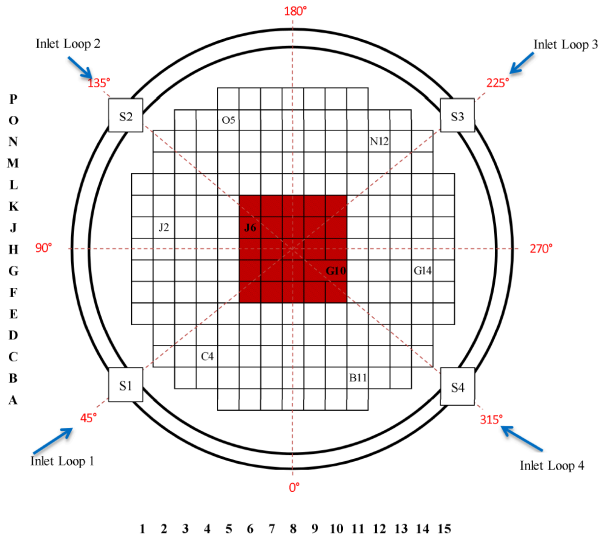

Fig. 1. Illustrative radial view of the nuclear reactor core model utilised in Simulate-3K. Each letter and number pairing denotes an in- or ex-core signal detector, and each grid square represents a fuel assembly. The red central zone represents a 5×5 cluster of fuel assemblies that vibrates synchronously in the x direction. The calculated neutron noise distribution was utilised in our deep learning based analysis of perturbations in the Time Domain.

I. Introduction

For over half a century, the nuclear industry has primarily focused on the technological evolution of reliable nuclear power plants for the production of electricity. By monitoring nuclear reactors while running at nominal conditions, it is possible to gather valuable insight for early detection of anomalies. Various types of fluctuations can be caused by the turbulent nature of flow in the core, mechanical vibrations within the reactor, coolant boiling and stochastic character of nuclear reactions, among other factors. These fluctuations are often referred to as neutron noise $\delta X(\mathbf{r}, t)$, which is measured 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: $\mathbf{r}$ the spatial coordinate within the core, and t time.

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

With detailed descriptions of reactor geometry, physical perturbations and probabilities of neutron interactions within the core – by assuming a particular reactor transfer function (i.e. Green's function) – one can simulate how fluctuations affect the neutron flux in the time or frequency domain. Different types of perturbations can then be applied in order to estimate and study the induced neutron noise, as to solve the *forward problem*. Intuitively, the *backward problem*, also known as *unfolding*, consists of localising the perturbation origin and can only be carried out if the reactor transfer function is inverted. Solving the unfolding problem is therefore non-trivial as measurements of the induced neutron noise are not available at every position inside the reactor core, due to a limited number of in- and ex-core sensors available. In this work, a novel method to unfold nuclear reactor signals pertaining to the localisation of different types of perturbations is proposed. This is achieved by extending and improving previous research on the application of deep learning techniques to detect anomalies in nuclear reactors [1].

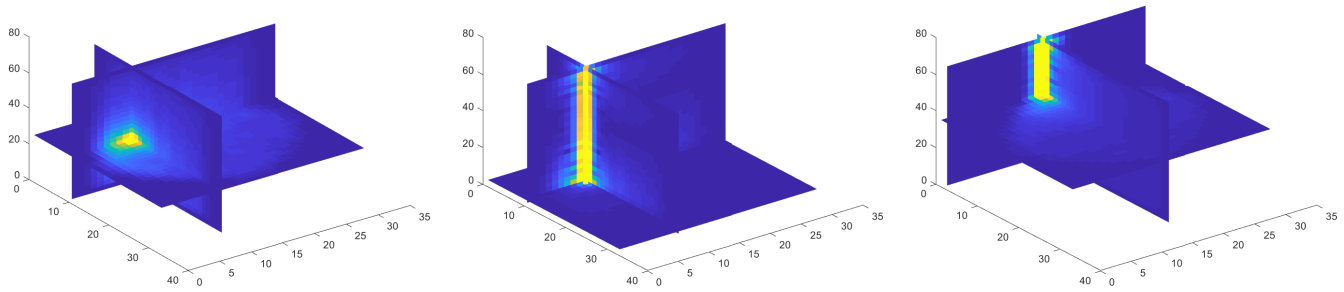

Fig. 2. Examples of induced neutron noise types. From **Left** to **Right**, the responses to *Localised*, *Propagating* type 1 and 2 perturbations are illustrated.

II. Related Work

Fault detection in nuclear reactors has been the focus of a few recent studies. [2] proposed a pattern recognition framework to detect anomalies based on symbolic dynamic filtering of time series data. [3] predicted critical heat flux by ways of Adaptive Neuro-Fuzzy Inference Systems (ANFIS). [4] monitored sensors by utilising auto-associative kernel regression and sequential probability ratio tests. [5] collected reactors parameters and implemented an artificial neural network (NN) based system to diagnose transients. [6] proposed a nuclear reactor fault detector based on the combination of principal component analysis and fisher discriminant analysis. Deep learning has recently shown to be effective in a variety of safety-critical fields spanning from signal analysis, to computer vision applications including medical imaging and text recognition ([7]–[13]). In [14], a Convolutional NN (CNN) and Naïve-Bayes data fusion scheme was proposed to detect fractures in plant components by way of individual video frame analysis. In [15], a dynamic Galerkin Finite Element Method-based simulator was applied to calculate the frequency domain neutron noise distribution of the VVER-1000 core reactor. Subsequently, an ANFIS was employed to localise the induced neutron noise source. Conversely, in a recent work by [1], the induced neutron noise was simulated by using CORE-SIM, at different perturbations strengths and frequencies. A CNN was employed to localise the origin of frequency domain neutron noise perturbations in nuclear reactor signals. This was achieved by spatially splitting the complex signal volumes into 12 or 48 individual blocks, each pertaining to a different class. A classification task was then formulated, followed by a combination of k-means and k-NN based analysis of extracted latent variables, enabling a finer unfolding resolution. Although the results were promising, an unfounded conversion of complex signal volumes for use in conventional CNNs led to unnecessary loss of spatial information. To address this limitation, in this work we propose a new bespoke 3D CNN model for multi-task perturbation unfolding regression and type classification. Additionally, we extend our analysis to time-domain simulated signals regarding vibrating fuel assemblies and/or fluctuations in thermal-hydraulic parameters (e.g. inlet coolant flow/temperature).

III. The Examined Scenarios

A. Frequency Domain

In this study, CORE-SIM [16] was employed to model the induced neutron noise, in a Pressurised Water Reactor (PWR), under two scenario settings: *Absorber of Variable Strength* and *Propagating Perturbation* in the frequency domain. During the forward problem, the reactor transfer function, which is considered to be the Green's function of the system, captures the response of the induced fluctuations in neutron flux. The effect of a perturbation can be assessed from any spatial point within the reactor core, provided that there exists a one-to-one relationship between every possible location where a perturbation is located and the position where the neutron noise is measured. The latter is described as

$$\delta\phi(\mathbf{r},\omega) = \int_V G(\mathbf{r},\mathbf{r_p},\omega)\delta S(\mathbf{r_p})d\mathbf{r_p}, \tag{2}$$

where the core transfer function is integrated across the whole core reactor volume V, whereas $\mathbf{r_p}$ and ω refer to the source and the angular frequency of the perturbation respectively. For more details, please refer to the official CORE-SIM user manual [16], [17]. Diffusion theory was applied to perform a low-order approximation of the angular moment of the neutron flux. The energy of the system was discretised with a two-energy group formulation: one with a high and one low energy spectrum, henceforth referred to as the *Fast* and the *Thermal* groups respectively.

Absorber of Variable Strength: In this scenario (*Localised*, see Fig. 2), the thermal macroscopic absorption cross-section was perturbed at three different frequencies 0.1, 1 and 10 Hz, altering the absorption of thermal neutrons. This perturbation type can be considered as localised at a specific source location. A PWR with a radial core of size 15×15 fuel assemblies (FA) was modelled, using a volumetric mesh with $32 \times 32 \times 26$ voxels.

Propagating Perturbation: In these scenarios (*Propagating* type 1 and 2, see Fig. 2), fuel assemblies were also perturbed at 0.1, 1 and 10 Hz, at which the fluctuations in neutron noise were modelled. Propagating perturbations were located either at the core inlet and transported upwards with the coolant starting from the lowest level of the core (type 1); or within

TABLE I
SYNCHRONISED VIBRATION OF A 5×5 FUEL ASSEMBLIES CENTRAL CLUSTER.

Scenario	Perturbation	Frequency	Amplitude	ID
1	5×5 cluster FAs	WN	$1\ mm$	1 0 0 0
	5×5 cluster FAs	WN	$0.5\ mm$	1 0 0 0
2	5×5 cluster FAs	1 Hz	$1\ mm$	0 1 0 0
	5×5 cluster FAs	1 Hz	$0.5\ mm$	0 1 0 0

TABLE II
SYNCHRONISED PERTURBATION OF COOLANT THERMAL-HYDRAULIC PARAMETERS.

Scenario	Perturbation	Frequency	Amplitude	ID
3	temperature	random	$\pm 1°C$	0 0 1 0
4	flow	random	$\pm 1\%$	0 0 0 1

the core and propagated along the fuel assembly's cross-section, by means of the coolant flow (type 2). See Fig. 8 for intuition. Identical mesh specifications to the *Absorber of Variable Strength* scenario were adopted.

Combined Perturbations: In this scenario, combinations of the aforementioned perturbation types can occur simultaneously at different locations in the reactor. However, no more than one instance per perturbation type can occur at any given time.

Data Pre-processing: The complex signals are a 3D representation of the distribution of the induced neutron noise, including *Fast* and *Thermal* neutron groups. They are distributed in the form of voxels of size $32 \times 32 \times 26$, each containing a perturbation located at a specific coordinate location i, j, k (considered as the label of our regression task). The dataset is comprised of 19552 (*Absorber of Variable Strength*) and 752 (*Propagating* type 1 and 2) instances per frequency (0.1, 1 and 10 Hz). Furthermore, the signal was corrupted by obscuring parts (set values to zero) at random in order to emulate fewer available sensor measurements. Two versions of obscured data were generated with channel-wise repeated masks of size $32 \times 32 \times 26$. Each 32×32 mask was generated by randomly selecting 5% and 20% of measurements respectively, and setting remaining values to zero. As previously alluded to, a given reactor signal is composed of 2 types of responses, *Fast* and *Thermal*, each comprised of amplitude and phase. Resulting in a total of 4 components of size $32 \times 32 \times 26$, which we concatenated into a $64 \times 64 \times 26$ volume, zero-padded to $64 \times 64 \times 32$ for convenience.

B. Time Domain

Simulate-3K (S3K) was utilised to model fuel assemblies cluster vibrations, including fluctuations in thermal-hydraulic parameters between the coolant loops, on a model of the four-loop Westinghouse PWR mixed core, utilised in [18]. The system operating conditions were close to those used in the frequency domain experiments. For more details with regard to S3K, the reader is invited to refer to the manual [19].

TABLE III
COMBINATION OF SYNCHRONISED VIBRATION OF A 5×5 FUEL ASSEMBLIES CENTRAL CLUSTER AND SYNCHRONISED PERTURBATION OF COOLANT THERMAL-HYDRAULIC PARAMETERS.

Scenario	Combined Perturbations	ID
5	Temperature (5) & flow (6)	0 0 1 1
6	5×5 FA (2) & temperature (5)	1 0 1 0
7	5×5 FA (1) & temperature (5)	1 0 1 0
8	5×5 FA (4) & temperature (5)	0 1 1 0
9	5×5 FA (3) & temperature (5)	0 1 1 0
10	5×5 FA (2) & flow (6)	1 0 0 1
11	5×5 FA (1) & flow (6)	1 0 0 1
12	5×5 FA (4) & flow (6)	0 1 0 1
13	5×5 FA (3) & flow (6)	0 1 0 1

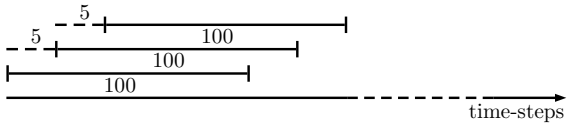

Fig. 3. Signal sampling. Signal windows of 100 time-steps were sampled using sliding windows of stride 5 time-steps

Fig. 1 depicts a cross-sectional view of the utilised core. The cluster of fuel assemblies is highlighted in red, whereas the coordinates (e.g. B11) identify the location of neutron detectors. Detector-wise, the reactor is comprised of six axial levels and a total of fifty-six detectors: eight located ex-core, identically distributed at two axial levels (level 1 ($L1$) and level 6 ($L6$)); forty-eight in-core, equally distributed across the six levels. Every scenario had a duration of $100\ s$, sampled with time steps of $0.01\ s$, and is briefly explained below.

1) Vibrating central cluster of fuel assemblies: This perturbation refers to four perturbation instances (see Table I), in which a cluster of 5×5 fuel assemblies is vibrating synchronously in the x direction, following either a white noise signal or a sine wave function $f = 0.1$ Hz, with varying amplitudes in the range of $0.5\ mm$, and $1\ mm$. "ID" is a label later utilised to classify different perturbation types. It is worth noting that the first and second rows represent the same scenario, since the applied perturbations are the same but with different amplitude; identical consideration applies to the third and fourth rows. Therefore, two individual scenarios were identified out of the four possible perturbations.

2) Perturbation of thermal-hydraulic parameters: This perturbation refers to two scenarios, in which synchronised fluctuations of inlet coolant temperature between the four coolant loops were induced. As reported in Table II, the inlet coolant temperature was forced to fluctuate with amplitude of $\pm 1°C$ over the mean value of $283.8°C$ (third scenario). In the fourth scenario, inlet coolant flow random fluctuations with amplitude of 1% over the relative flow (100%) were simulated.

3) Combined Perturbations: Scenarios five to thirteen refer to combinations of previous perturbations associated to the vibration of a 5×5 fuel assembly and fluctuations of inlet coolant thermal-hydraulic parameters between the four coolant

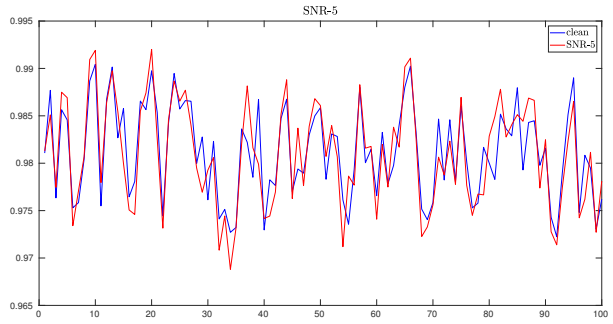

Fig. 4. Example of a signal obtained by means of S3K with noise added at SNR= 5.

loops. A detailed description of these scenarios is provided in Table III. In the column "Combined Perturbations", the number between brackets links to the Scenario ID reported in Table I and II.

Data Pre-processing: Signals produced by S3K are a representation of the neutron flux measured by the in- and ex-core detectors. Taking into account the duration of each applied perturbation and sampling rate, data from each sensor were available in the form of a vector of 10001 elements. Given the limited amount of data available, it was appropriate to perform data augmentation. To this end, each signal was re-sampled by means of sliding windows as shown in Fig. 3. Specifically, with sensor measurements over 100 s at a sampling rate of 0.01 s, we get $\mathbf{x} \in \mathbb{R}^{10001}$ signal vectors. These vectors are augmented by means of 100 time step sliding windows with a stride of 5 to produce $\mathbf{x} \in \mathbb{R}^{1980\times100}$. Furthermore, the signal was corrupted by the addition of White Gaussian Noise at signal-to-noise ratios (SNR) 10 and 5 to study the effect of noisy signals on the performance of our model (Fig. 4).

IV. The Proposed Approach

A. Frequency Domain

Given complex reactor signals in the form of volumetric meshes, it is advantageous to capture spatial information not only in 2D coordinate space (i, j) but also channel-wise through k. This means that knowledge learnt in a particular area of the volume can generalise well to others. The generalisation property of CNNs is crucial, as it allows for a great reduction in the number of parameters when compared to fully-connected (FC) networks, without sacrificing performance. However, it is important to state that the signal volumes are not a measure of induced neutron noise over time, but rather a measured response in every (i, j, k) location within the core reactor, in an instant soon after a perturbation is induced. Therefore, the input signal volumes are more closely related to MRI or CT scans rather than videos in terms of data format. Relatedly, 3D CNNs have been used extensively in the medical field for tumour and lesion segmentation, as well as in action recognition tasks to a very good level of success [20]–[24]. In pursuance of optimal feature extraction in all dimensions of the reactor signal, a bespoke 3D CNN is proposed.

TABLE IV
3D-CNN architecture for Frequency Domain perturbation type classification and source regression.

Input Size: 64×64×32×1		
Conv-BN-ReLU	3×3×3@64	64×64×32×64
MaxPool	2×2×2	32×32×16×64
Conv-BN-ReLU	1×1×1@32	32×32×16×32
Conv-BN-ReLU	3×3×3@128	32×32×16×128
MaxPool	2×2×2	16×16×8×128
Conv-BN-ReLU	1×1×1@64	16×16×8×64
Conv-BN-ReLU	3×3×3@256	16×16×8×256
MaxPool	2×2×2	8×8×4×256
Conv-BN-ReLU	1×1×1@128	8×8×4×128
Conv-BN-ReLU	3×3×3@512	8×8×4×512
MaxPool	2×2×2	4×4×2×512
4×4×2 Global Average Pooling		
3×1 Fully-Connected, Multi-sigmoid		
3×1 Fully-Connected, Linear		

1) Convolutional Neural Networks: Convolutional Neural Networks (CNNs) [25] perform automatic feature extraction through a series of volume-wise convolutions and feature routing. For each convolutional layer, a resulting set of filters are learnt to capture spatial patterns in given inputs. Deeper CNNs are capable of capturing complex hierarchical concepts, whereby more general and abstract concepts initiate from the stem of the network and become increasingly task specific in the final layers. The convolution operation in CNNs is significantly more efficient than dense matrix multiplication through sparse interactions and parameter sharing. Formally, in 3D CNNs one would compute a pre-activated value of a given unit $n^{[\ell]}_{i,j,k}$ at (i, j, k) position in a 3D feature map of layer ℓ, by summing the weighted kernel contributions from the previous layer units in $\mathbf{A}^{[\ell-1]}$ as

$$n^{[\ell]}_{i,j,k} = \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}\sum_{z=0}^{Z-1} \mathbf{W}^{[\ell]}_{x,y,z}\mathbf{A}^{[\ell-1]}_{i+x,j+y,k+z}, \tag{3}$$

where $\mathbf{W}^{[\ell]}_{x,y,z}$ is a single learnt weight pertaining to a kernel $\mathbf{W}^{[\ell]}$ of dimensions $X \times Y \times Z$ in layer ℓ, which is convolved with cells from the previous layer ($\mathbf{W}^{[\ell]} * \mathbf{A}^{[\ell-1]}$). Each feature map f in a given layer ℓ has a learnt bias term $b^{[\ell,f]}$, which is added pre non-linearity as

$$a^{[\ell,f]}_{i,j,k} = \phi(n^{[\ell,f]}_{i,j,k} + b^{[\ell,f]}), \tag{4}$$

where $\phi(\cdot)$ is a non-linear activation function such as ReLU:$\rightarrow$ $f(\cdot) = \max(0, \cdot)$ or the logistic sigmoid.

Table IV depicts the 3D CNN architecture, devised through experimentation, for the classification of perturbation types in the frequency domain and their respective coordinate locations in 3D space. Convolutional layers use $3 \times 3 \times 3$ kernels with stride 1 and are followed by Batch Normalization (BN) [26] and ReLU activations. In order to reduce the number of parameters incurred by 3D convolutions and increase the complexity of the network with more ReLU non-linearities, Bottleneck

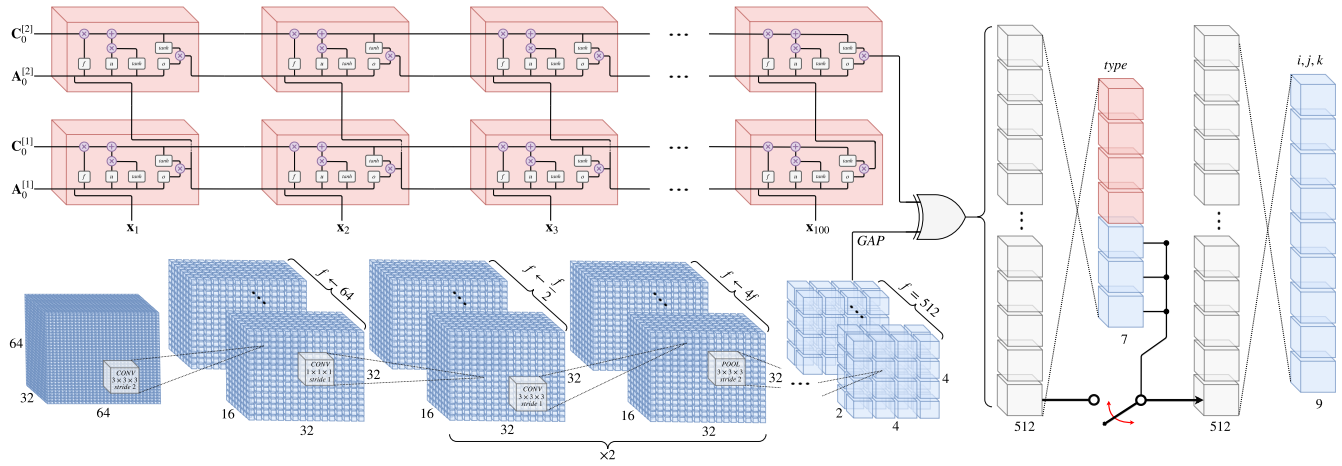

Fig. 5. Unified framework for time and frequency domain perturbation type classification and coordinate regression. An LSTM network at the top for time domain signals, and a 3D CNN below for frequency domain signals. Both networks output 512 dimensional latent variable representations of their respective inputs, and their *flow* is controlled by XOR gate logic and a *switch* is activated for perturbation coordinate regression in the frequency domain.

Layers ($1\times1\times1$ convolution) are introduced between $3\times3\times3$ convolutions. Max Pooling with $2\times2\times2$ kernels down sample inputs and a final Global Average Pooling (GAP) [27] layer produces 512 dimensional vector representations. The representations are then fed to 2 separate FC Layers, one for multi-label classification with 3 sigmoid non-linear units and the other for perturbation coordinate regression (i, j, k) with 3 linear units. In the combined perturbation case, the 3 sigmoid units represent 7 different classes denoted as

$$\mathbf{C} = \{001,\ 010,\ 100,\ 101,\ 011,\ 110,\ 111\}, \tag{5}$$

where $\mathbf{C}$ contains all combinations of Localised (ID 100), Travelling type 1 (ID 010) and type 2 (ID 001) perturbations as described in Section III. In practice, the 3 linear units become 9 units to allow for regression of more than one perturbation location at a time.

When training a CNN on multiple objectives, it is common practice to compute a linear weighted sum of losses per task i of T tasks, where weight coefficients λ_i control the dominance of each loss over the gradient. Formally, the multi-task optimisation objective is minimised with respect to $\mathbf{W}$ parameters given $\mathcal{D}$ input data as

$$\mathscr{L} = \sum_{i}^{T} \lambda_i \ell_i(\mathcal{D}; \mathbf{W}), \tag{6}$$

where ℓ_i represents either the negative log-likelihood loss for perturbation type classification: $\ell_1(y_1, \widehat{y}_1)$, or the $L2$ loss for perturbation coordinate regression: $\ell_2(y_2, \widehat{y}_2)$. Concretely, the 3D CNN is trained by minimising the following criterion $\mathscr{L}(\mathcal{D}; \mathbf{W}, \lambda_1, \lambda_2) =$

$$-\frac{1}{N}\sum_{i=1}^{N}\left[\frac{\lambda_1}{P}\sum_{j=1}^{P}\left[y_1^j \log(\widehat{y}_1^j) + (1-y_1^j)\log(1-\widehat{y}_1^j)\right] + -\frac{\lambda_2}{C}\sum_{c=1}^{C}\left\|y_2^c - \widehat{y}_2^c\right\|^2\right]_i \tag{7}$$

where P and C denote the number of perturbation types and location coordinates respectively, with λ_1, λ_2 as tuned weight coefficients for each loss. The resulting network model $\mathcal{F}(\mathcal{D}; \mathbf{W})$ predicts a continuous vector of outputs $(i, j, k$ coordinates) and discrete outputs for perturbation type classes. Lastly, parameters $\mathbf{W}$ were initialised as proposed in [28].

B. Time Domain

Given the sequential nature of the signals in the perturbation induced in the time domain, it was intuitive to utilise Recurrent Neural Networks (RNN). RNNs are particularly suitable for this type of data as their cells can formulate a non linear output $\mathbf{A}^{[t]}$ based on both the input data $\mathbf{x}^{[t]}$ at the current time step t, and the previous time-step activation $\mathbf{A}^{[t-1]}$. This is described in (8), where $\phi(\cdot)$ is a non-linear activation function of choice such as the hyperbolic tangent.

$$\mathbf{A}^{[t]} = \phi(\mathbf{x}^{[t]}, \mathbf{A}^{[t-1]}) \tag{8}$$

In particular, Long Short-Term Memory (LSTM) was adopted because of its capability of learning long term dependencies on data. This is attained by formulating memory cells. The equations relative to LSTM follow, and the reader is invited to refer to the original paper [29] for further details.

$$\begin{aligned}
\widetilde{\mathbf{C}}^{[t]} &= \tanh(\mathbf{W}_{\tilde{c}} \cdot [\mathbf{A}^{[t-1]}, \mathbf{x}^{[t]}] + \mathbf{b}_{\tilde{c}}) \\
\mathbf{\Gamma}_u &= \sigma(\mathbf{W}_u \cdot [\mathbf{A}^{[t-1]}, \mathbf{x}^{[t]}] + \mathbf{b}_u) \\
\mathbf{\Gamma}_f &= \sigma(\mathbf{W}_f \cdot [\mathbf{A}^{[t-1]}, \mathbf{x}^{[t]}] + \mathbf{b}_f) \\
\mathbf{\Gamma}_o &= \sigma(\mathbf{W}_o \cdot [\mathbf{A}^{[t-1]}, \mathbf{x}^{[t]}] + \mathbf{b}_o) \\
\mathbf{C}^{[t]} &= \mathbf{\Gamma}_u \odot \widetilde{\mathbf{C}}^{[t]} + \mathbf{\Gamma}_f \odot \mathbf{C}^{[t-1]} \\
\mathbf{A}^{[t]} &= \mathbf{\Gamma}_o \odot \tanh(\mathbf{C}^{[t]})
\end{aligned} \tag{9}$$

In (9), $\mathbf{C}$ is the memory cell, $\mathbf{\Gamma}_u$, $\mathbf{\Gamma}_f$ and $\mathbf{\Gamma}_o$ are the update, forget and output gates respectively; $\mathbf{W}$ denotes the model's weights, and $\mathbf{b}$ are the bias vectors. These parameters are all jointly learnt through backpropagation. Essentially, at each time-step, a candidate update of the memory cell $\widetilde{\mathbf{C}}^{[t]}$ is

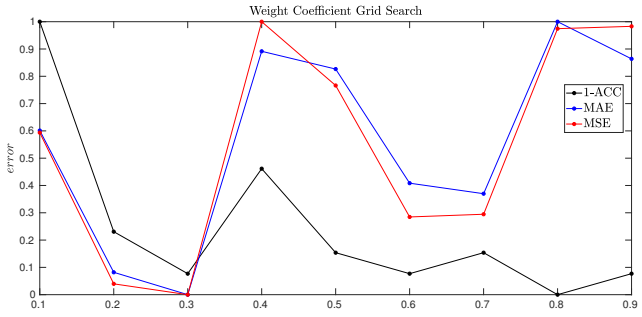

Fig. 6. Weight coefficient grid search for the 3D-CNN classification and regression losses. Coefficient 0.3 for classification and 0.7 for regression yielded the best performance.

proposed, and according to the learnt gates, $\widetilde{\mathbf{C}}^{[t]}$ can be utilised to update the memory cell $\mathbf{C}^{[t]}$, and subsequently provide a non linear activation of the LSTM cell $\mathbf{A}^{[t]}$. In order to improve the representational capacity of our network and therefore learn a meaningful representation of the signal, two LSTM layers were stacked, with each LSTM cell containing 512 neurons.

The problem of recognising which scenario a signal is representative of was tackled as a multi-label classification task. Since four individual perturbation (and their combinations) were identified (see Table I, II and III), in order to classify which of these perturbation was present, 512 dimensional LSTM representations were fully connected to four neurons with sigmoid activation functions. During training the following negative log-likelihood criterion was minimised

$$\mathscr{L}(y,\widehat{y}) = -\frac{1}{PN}\sum_{j=1}^{P}\sum_{i=1}^{N}\Big[y_j \log(\widehat{y}_j) + (1-y_j)\log(1-\widehat{y}_j)\Big]_i \tag{10}$$

where P is the number of sigmoid units used for the multi-label classification task, and N is the number of samples in a batch. The parameters of the resulting model were initiliased as per the scheme proposed in [30].

C. Deep Time-Frequency Framework

As illustrated in Fig. 5, a Deep Neural Network (DNN) framework was formulated for processing both Time and Frequency Domains signals coming from nuclear reactor sensor measurements. It is important to clarify that the simulations were performed using different rector cores in each domain, as per CORE-SIM and S3K specifications. Both the 3D CNN and LSTM network produced 512 dimensional vector representations of their respective inputs. The representations were then fed to a fused classification layer comprised of 7 sigmoid units (3 for Frequency & 4 for Time) to accommodate all scenario combinations as a multi-label classification task. Lastly, whenever a frequency domain perturbation is detected, the red switch in Fig. 5 is triggered and the current 512 dimensional representation is fed to a separate FC layer to regress perturbation coordinates (i, j, k) in 3D space.

TABLE V
RESULTS OF THE FREQUENCY DOMAIN 3D CNN EXPERIMENTS FOR PERTURBATION TYPE CLASSIFICATION AND LOCALISATION REGRESSION. (*) MARKS COMBINED PERTURBATIONS SCENARIOS.

3D CNN Perturbation Classification & Localisation				
Sensors (%)	Train/Valid/ Test (%)	Classification Accuracy (%)	(i,j,k) Regression	
			MAE	MSE
20	60/15/25	**99.75±0.09**	**0.2528±0.03**	**0.1347±0.02**
20	25/15/60	99.12±0.17	0.4221±0.05	0.4152±0.07
20	15/25/60	98.62±0.22	0.5886±0.05	0.8174±0.12
5	60/15/25	99.32±0.18	0.326±0.05	0.2086±0.04
5	25/15/60	98.34±0.22	0.4818±0.05	0.6044±0.08
5	15/25/60	97.27±0.54	0.689±0.1	1.0749±0.25
20*	60/15/25	99.82±0.05	0.5602±0.04	1.6036±0.15
20*	25/15/60	99.56±0.07	0.8942±0.04	3.5739±0.16
20*	15/25/60	99.44±0.08	0.9635±0.06	4.2814±0.19
5*	60/15/25	99.47±0.03	0.8809±0.04	3.4424±0.16
5*	25/15/60	98.33±0.24	0.5001±0.04	0.6381±0.08
5*	15/25/60	**97.15±0.15**	**1.9528±0.11**	**11.902±0.66**

V. EXPERIMENTAL STUDY

A. Frequency Domain

For completeness and more detailed analysis of the results, the performance of the proposed framework in the Time and Frequency domains are kept separate. The implementation was based on MATLAB [31], Keras deep learning framework [32] and Tensorflow numerical computation library [33]. The experiments were conducted using a server with an Intel Xeon(R) E5-2620 v4 CPU, eight GPUs and 96GB of RAM. The results of the experiments conducted on the volumetric signal data are reported in Table V. As explained in greater detail in Subsection III-A, the volumetric signals were corrupted by obscuring parts at random, in order to emulate fewer available sensor measurements and thus increase the complexity of the problem. As shown in Table V, in the first experiment a dataset with 20% of the sensor measurements was generated. Similarly, in the second experiment a different dataset was generated in which only 5% of the sensor measurements were kept. Both of these experiments were conducted to study the effect of sensor measurement sparsity on the performance of our algorithm. Furthermore, different training, validation and test splits were also utilised to study the effect of learning from a smaller pool of possible perturbations in the training set.

In the case of the Combined Perturbation experiments (marked with (*) in Table V), a similar approach was undertaken with regards to the percentage of sensors kept and the dataset splits. Two datasets were generated for training (20% and 5%) in which multiple perturbations are classified and their respective source coordinates regressed simultaneously. Moreover, a hyper-parameter grid search was performed over the loss weight coefficients for each task, and the best results were achieved with $\lambda_1 = 0.3$ and $\lambda_2 = 0.7$ (see Fig. 6). For all experiments, the 3D CNN was trained to minimise the criterion

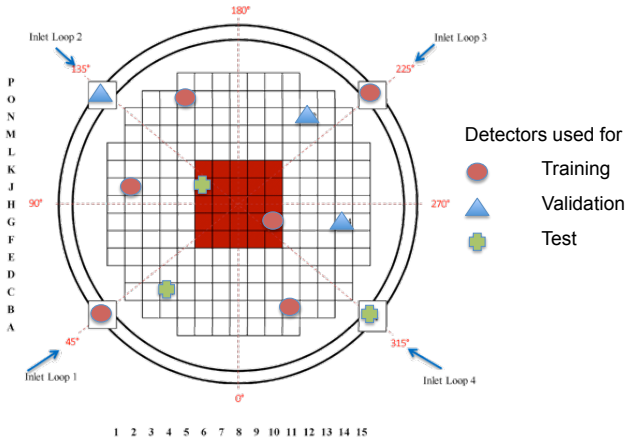

Fig. 7. Description of detector locations for the signals utilised in training, validation and testing of the deep LSTM network model, in the classification of different types and combinations of Time domain perturbations.

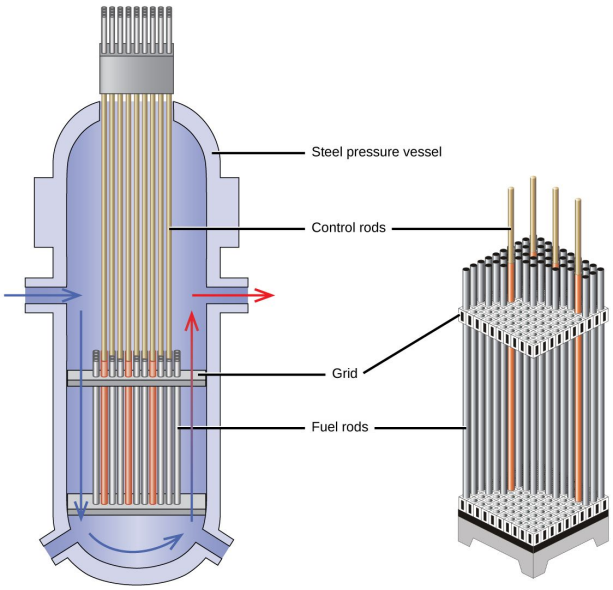

Fig. 8. Illustration of a nuclear reactor core, highlighting its internal components. The core shown on the left hand side contains the fuel and control rod assembly shown on the right hand side. Photo credit to [35], [36].

in Eq. (7) using backpropagation, and the Adaptive Moment Estimation (Adam) optimiser [34] with the default parameters and a batch size of size 32. Each model was trained 10 times and the mean performance was taken as the final result, along with the standard deviation.

As observable in Table V, high classification performance was achieved in all experiments with $99.75\%\pm0.09$ and $97.15\%\pm0.15$ accuracy in the best case and worst case respectively. The mean squared and absolute errors (MSE, MAE) were used as evaluation metrics for the perturbation coordinate regression results, with best case of 0.2528 ± 0.03 (MAE), 0.1347 ± 0.02 (MSE) and worst case of 1.95 ± 0.11 (MAE), 11.90 ± 0.66 (MSE).

Overall, the results show that the classification task achieves better performance across all datasets compared to the regression task. The regression performance deteriorates with the introduction of combined perturbations and limited sensor measurement/training set size, whereas the classification of perturbations types is more resilient to fluctuations in the number of sensors used in the training phase.

B. Time Domain

In this experiment, individual sensor measurements were utilised to detect each of the thirteen scenarios (Table I, II and III). Starting with the data from the thirteen scenarios provided by S3K, each comprised of 56 one-dimensional signals of length 10001 (one signal per detector), after resampling, 17164 samples of size 56×100 were obtained. Subsequently, each 56×100 sample was subdivided into 56 one-dimensional signals of size 100×1. During training, each 100×1 signal from a single sensor was utilised to detect the presence of a scenario. In other words, any given scenario (perturbation) must be detected within one second of monitoring. Fig. 7 shows which sensors were utilised to train, validate and test the LSTM network, at each radial level of the reactor. For better intuition, Fig. 8 provides a depiction of the main components of a core, including fuel assemblies in a typical nuclear reactor. Overall, 480592 (from 28 sensors), 240296 (from 14 sensors) and 240296 (from 14 sensors) signals were used for training, validating and testing.

Hyper-parameters were experimentally tuned, and those utilised provided the best performance. The negative log-likelihood criterion in (10) was minimised with mini-batch (32) stochastic gradient descent (SGD). The Adam optimisation algorithm was used, to include adaptive learning rate, momentum, RMSprop and bias correction in weight updates, offering faster convergence rate than normal SGD with momentum [34]. The classification accuracy (%) achieved by the LSTM network was 97% on the clean signals, 81% with added noise at SNR= 10 and 77% with added noise at SNR= 5.

VI. Conclusion & Future Work

In this paper, the first step towards a unified deep framework was proposed for the classification and regression of perturbations in nuclear reactors. Both Time and Frequency domain data were obtained through inducing perturbations such as an *Absorber of Variable Strength* and *Propagating Perturbation* in the Frequency domain; vibration of fuel assemblies and fluctuations of thermal-hydraulic parameters at the inlet coolant between the 4-loops of a Westinghouse PWR reactor in the Time domain.

The proposed framework is comprised of a 3D CNN and an LSTM architecture that each output 512 dimensional representations of their respective input signals, and combinations of nuclear reactor perturbations are classified with a fused multi-sigmoid layer. A *switch* was introduced to control the *flow* of the frequency domain 512 dimensional representation, which is fed to a regression layer whenever a perturbation is detected in the 3D complex signal volume. Furthermore, the effects of sensor measurement sparsity and noisy signals were

evaluated in a series of experimental studies, demonstrating the capability of our framework to achieve good results in both unfolding and perturbation type classification.

In future work, we plan to extend our studies to other types of data, simulated in the Time and Frequency domains utilising the same/multiple reactor cores, to test the sensitivity of our framework to different reactor characteristics. Furthermore we intend to investigate real data coming from nuclear power plants, in pursuit of a framework suitable for simultaneously handling Time and Frequency domain signals for the localisation and classification of nuclear reactor anomalies.

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. We would like to thank the reviewers for their helpful comments.

REFERENCES

[1] Francesco Calivá, Fabio De Sousa Ribeiro, Antonios Mylonakis, Christophe Demaziere, Paolo Vinai, Georgios Leontidis, Stefanos Kollias, et al. A deep learning approach to anomaly detection in nuclear reactors. In *Proceedings of 2018 International Joint Conference on Neural Networks (IJCNN)*, 2018.

[2] Xin Jin, Yin Guo, Soumik Sarkar, Asok Ray, and Robert M. Edwards. Anomaly detection in nuclear power plants via symbolic dynamic filtering. *IEEE Transactions on Nuclear Science*, 58(1):277–288, Feb 2011.

[3] Salman Zaferanlouei, Dariush Rostamifard, and Saeed Setayeshi. Prediction of critical heat flux using anfis. *Annals of Nuclear Energy*, 37(6):813–821, 2010.

[4] 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.

[5] TV Santosh, Abhinav Srivastava, VVS Sanyasi Rao, AK Ghosh, and HS 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.

[6] 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.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[8] 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.

[9] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[10] Dimitrios Kollias, Miao Yu, Athanasios Tagaris, Georgios Leontidis, Andreas Stafylopatis, and Stefanos Kollias. Adaptation and contextualization of deep neural network models. In *Proceedings of 2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 2017.

[11] 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.

[12] Fabio De Sousa Ribeiro, Francesco Calivá, 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 International Conference on*, 2018.

[13] Fabio De Sousa Ribeiro, Liyun Gong, Francesco Calivá, Kjartan Gudmundsson, Mark Swainson, Miao Yu, Georgios Leontidis, Xujiong Ye, Stefanos Kollias, et al. An end-to-end deep neural architecture for optical character verification and recognition in retail food packaging. In *Image Processing, IEEE International Conference on (ICIP)*, 2018.

[14] Chen Fu-Chen and Jahanshahi Mohammad R. 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.

[15] Seyed Abolfazl Hosseini and Iman Esmaili Paeen Afrakoti. Neutron noise source reconstruction using the adaptive neuro-fuzzy inference system (anfis) in the vver-1000 reactor core. *Annals of Nuclear Energy*, 105:36–44, 2017.

[16] Christophe Demazière. Core sim: a multi-purpose neutronic tool for research and education. *Annals of Nuclear Energy*, 38(12):2698–2718, 2011.

[17] Christophe Demazière. User's manual of the core sim neutronic tool. Technical report, Chalmers University of Technology, 2011.

[18] Tomasz Kozlowski and Thomas J Downar. Oecd/nea and us nrc pwr mox/uo2 core transient benchmark. *Final Specifications, Revision*, 2, 2003.

[19] Gerardo Grandi, Jeffrey A. Borkowsky, and Kord S. Smith. Simulate-3k models and methodology. *SSP-98013, Revision*, 6, 2006.

[20] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017.

[21] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016.

[22] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[23] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE, 2015.

[24] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.

[25] Yann LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, pages 143–155, 1989.

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[27] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[28] 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.

[29] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[30] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[31] MATLAB Users Guide. The mathworks. *Inc., Natick, MA*, 5:333, 1998.

[32] François Chollet et al. Keras, 2015.

[33] 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.

[34] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[35] Generalic Eni. Control rod. croatian-english chemistry dictionary & glossary. https://bit.ly/2PKBYrf., 2017. Accessed: 05-09-2018.

[36] Elizabeth Gordon. Nuclear reactor components. https://bit.ly/2Cpjt9O., 2017. Accessed: 05-09-2018.