



**HAL**  
open science

# Interpreting convolutional neural network decision for earthquake detection with feature map visualization, backward optimization and layer-wise relevance propagation methods

Josipa Majstorović, Sophie Giffard-Roisin, Piero Poli

## ► To cite this version:

Josipa Majstorović, Sophie Giffard-Roisin, Piero Poli. Interpreting convolutional neural network decision for earthquake detection with feature map visualization, backward optimization and layer-wise relevance propagation methods. *Geophysical Journal International*, 2023, 232 (2), pp.923-939. 10.1093/gji/ggac369 . hal-03832942

**HAL Id: hal-03832942**

**<https://hal.science/hal-03832942>**

Submitted on 28 Oct 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1 **Interpreting convolutional neural network decision for**  
2 **earthquake detection with feature map visualisation,**  
3 **backward optimisation and layer-wise relevance**  
4 **propagation methods**

5 Josipa Majstorović<sup>1\*</sup>, Sophie Giffard-Roisin<sup>1</sup>, Piero Poli<sup>1</sup>

<sup>1</sup> *l'Institut des Sciences de la Terre, Université Grenoble Alpes, CNRS (UMR5275), Grenoble, France*

6

7 **SUMMARY**

8 In the recent years, the seismological community has adopted deep learning (DL) models  
9 for many diverse tasks such as discrimination and classification of seismic events, identi-  
10 fication of P- and S- phase wave arrivals or earthquake early warning systems. Numerous  
11 models recently developed are showing high accuracy values, and it has been attested for  
12 several tasks that DL models perform better than the classical seismological state-of-art  
13 models. However, their performances strongly depend on the DL architecture, the training  
14 hyperparameters, and the training datasets. Moreover, due to their complex nature, we are  
15 unable to understand how the model is learning and therefore how it is making a predic-  
16 tion. Thus, DL models are usually referred to as a “black-box”. In this study we propose  
17 to apply three complementary techniques to address the interpretability of a convolutional  
18 neural network (CNN) model for the earthquake detection. The implemented techniques  
19 are: feature map visualisation, backward optimisation and layer-wise relevance propa-  
20 gation. Since our model reaches a good accuracy performance (97%), we can suppose  
21 that the CNN detector model extracts relevant characteristics from the data, however a  
22 question remains: can we identify these characteristics? The proposed techniques help

23 to answer the following questions: How is an earthquake processed by a CNN model?  
24 What is the optimal earthquake signal according to a CNN? Which parts of the earth-  
25 quake signal are more relevant for the model to correctly classify an earthquake sample?  
26 The answer to these questions help understand why the model works and where it might  
27 fail, and whether the model is designed well for the predefined task. The CNN used in this  
28 study had been trained for single-station detection, where an input sample is a 25 seconds  
29 three-component waveform. The model outputs a binary target: earthquake (positive) or  
30 noise (negative) class. The training database contains a balanced number of samples from  
31 both classes. Our results shows that the CNN model correctly learned to recognize where  
32 is the earthquake within the sample window, even though the position of the earthquake  
33 in the window is not explicitly given during the training. Moreover, we give insights on  
34 how a neural network builds its decision process: while some aspects can be linked to  
35 clear physical characteristics, such as the frequency content and the P- and S- waves, we  
36 also see how different a DL detection is compared to a visual expertise or an STA/LTA  
37 detection. On top of improving our model designs, we also think that understanding how  
38 such models work, how they perceive an earthquake, can be useful for the comprehension  
39 of events that are not fully understood yet such as tremors or low frequency earthquakes.

40 **Key words:** Neural networks, Numerical modelling, Time-series analysis, Computa-  
41 tional seismology

## 1 INTRODUCTION

Science community has to a great extent embraced machine learning (ML) algorithms for solving various tasks, and seismology is following this trend. Seismology is an observational data driven research field, and throughout years great number of techniques has been developed to study earthquake and how the seismic wave propagate through Earth. Because of the demand to handle large amount of data with usually computationally expensive techniques, the implementation of ML algorithms in seismology started very early (Dowla et al. 1990; Dai & MacBeth 1995), and nowadays there are numerous applications (Kong et al. 2019; Bergen et al. 2019; Mignan & Broccardo 2020, and references therein). The range of topics include: earthquake detection (Perol et al. 2018; Majstorović et al. 2021), phase picking (Zhu & Beroza 2018; Ross et al. 2018; Mousavi et al. 2020), early warning systems (Kong et al. 2016), real-time seismicity monitoring (Cua & Heaton 2007), ground-motion prediction (Jozinović et al. 2020) for obtaining source, path, and site effects (Alavi & Gandomi 2011), or subsurface geophysical structure in seismic tomography (Elad 2010) (for more detailed list of references see Kong et al. (2019)). ML algorithms, especially deep learning (DL) models, offer a successful framework to tackle all these tasks since DL models can be designed: a) to work with any type of data or even with the combination of different types of data, b) to produce any type of output depending on the task, c) to implement the algorithms in a computationally efficient way. Even though numerous studies provide us with a proof of concept that DL methods can successfully solve traditional seismological problems, some difficulties still remain. Mostly because these models are highly parametric and strongly depend on the DL architecture, the training hyperparameters, and the training dataset. Moreover, in the training process, we lose track of how the prediction is obtained and we cannot examine if models has learned something physically significant from the data itself.

One step closer into understanding how and why DL models work, and why some perform better than the others, can be done by developing tools to understand the so-called "black-box" nature of DL models. Even though the structure of a DL model is explicitly defined, and it is well understood how the mathematical operations are implemented, it can be substantially complex and the number of the operations can be tremendous. Therefore, tracking how data are being transformed within DL model is not feasible. Overall, studying the black-box nature of DL model, implies interpreting how the data are fitted for some predefined task by using a specific DL architecture. In recent years rich set of various techniques has been developed for the purpose of interpreting a prediction process behind DL models (Barredo Arrieta et al. 2020; Roscher et al. 2020; Samek et al. 2021; Ras et al. 2021; Linardatos et al. 2021; Kong et al. 2022). It is highly crucial to recognize if DL model failed to represent training data and sometime sole prediction value is not enough to alert the user of the problem. In the situations

75 where DL outputs have huge impacts on the decision making processes, this lack of interpretability is  
76 highly criticized (Castelvecchi 2016).

77 The various applications of DL models within the seismology are designed to produce only the  
78 output prediction value, that is obtained by maximizing the accuracy of the model performance. For  
79 example, in an earthquake detection task, the DL detectors are developed to recognizing earthquake  
80 signals in continuous seismograms that contain signals of many other geophysical, anthropogenic,  
81 instrumental sources, which we refer to as noise. To declare a detection within the continuous data,  
82 the model has to surpass a certain threshold of the prediction value. Even though the problem of  
83 earthquake detection is quite straightforward, the existing models are usually developed for specific  
84 purposes and/or in specific conditions, and suffer from false detections (Perol et al. 2018; Lomax et al.  
85 2019; Wu et al. 2019; Mousavi et al. 2019; Magrini et al. 2020; Zhu & Beroza 2018; Ross et al. 2018;  
86 Mousavi et al. 2020; Yang et al. 2020; Majstorović et al. 2021; Xiao et al. 2021; Saad et al. 2021).  
87 Consequently, developed detectors once applied on the same continuous data are generating dissimilar  
88 results. While we can quantify that existing detectors reach different accuracy performance values, we  
89 don't know why. In this context of interpretability, EQTransformer (Mousavi et al. 2020), the detector  
90 and seismic phase picking encoder, has already provided some intuition behind the decision process  
91 by implementing the hierarchical attention mechanism (Luong et al. 2015; Yang et al. 2016). Using  
92 attention mechanism we can get a first indication on what the DL model is focused on at different  
93 stages of the network. However, this technique is only able to investigate a specific layer of the network  
94 (the attention layer), thus it does not provide insights on how other layers transform the information  
95 within the network.

96 In this study, we apply interpretation techniques to explore the prediction process behind DL de-  
97 tector model. For this purpose we use the convolutional neural network (CNN) detector developed  
98 in Majstorović et al. (2021). Our main motivation is to explain how the information about the earth-  
99 quakes is embedded in a DL model, i.e. a binary classification model, separating earthquake from  
100 noise signals. If we consider that DL model is interpreting our training data space, this implies that  
101 it has presumably learned some high level characteristics, features, patterns, and is able to general-  
102 ize well to the unseen data that seemingly belong to the training data space. By exploring how our  
103 CNN detector makes prediction and how it classifies the samples from the evaluation dataset, we can  
104 learn which earthquake characteristics are relevant for this task and we can explore how our CNN  
105 architecture is suitable for this predefined scope. To tackle these questions we applied three differ-  
106 ent interpretability techniques, the feature map visualisation (Krizhevsky et al. 2012; Zeiler & Fergus  
107 2014), the backward-optimisation algorithm (Simonyan et al. 2014; Olah et al. 2017) and the layer-  
108 wise relevance propagation algorithm (Bach et al. 2015). Each technique focuses on different aspects

109 of the model. The feature map visualisation reveals how the individual earthquake samples are repre-  
110 sented within the CNN layers. The backward optimisation method answers the question of how the  
111 optimal earthquake signal looks like for our trained CNN model. And finally, the layer-wise relevance  
112 method illustrates which parts of the earthquake sample are important for a good detection.

113 The paper is organized as follows: first, we introduce the data and the CNN earthquake detector  
114 model used in this study. Further, we introduce the theory behind the feature map visualisation, the  
115 backward optimisation and the layer-wise relevance methods. Lastly, we analyze how and why the  
116 CNN earthquake detector model works, and which parts of the model are more or less independent  
117 when we repeat the retraining process. We finish with the discussions and perspectives for future  
118 works.

## 119 **2 MATERIAL AND METHODS**

### 120 **2.1 Data**

121 In this study we use the AQULO dataset introduced in Majstorović et al. (2021). It is based of two  
122 types of samples: positive samples which are the earthquake signals that contain at least P- and S-  
123 waves arrivals, and negative samples that are composed of random geophysical and anthropogenic  
124 signals. The data are collected from the AQU station placed in the city of L'Aquila, in the Abruzzo  
125 region in the Central Apennines of Italy for a period of 30 years. It contains around 123k samples,  
126 from which 48% are positive samples. In this case each positive sample corresponds to one catalogued  
127 earthquake, where 40% originated from the Valoroso catalog (Valoroso et al. 2013). In other words, we  
128 cleaned our dataset from the positive samples that contain multiple events. The duration length of the  
129 samples is set to 25 seconds, the sampling frequency is 20 Hz and within each sample there are three  
130 components (east-west, north-south, vertical) waveforms. Additionally, the dataset is non-filtered and  
131 normalised per sample by the maximum value out of the three components. Due to the Gutenberg-  
132 Richter law (Gutenberg & Richter 1955) the distribution of the earthquakes' epicentral distance and  
133 magnitude is quite imbalanced.

### 134 **2.2 Convolutional neural network**

135 Neural networks (NNs) are a family of techniques within the machine learning domain that are at the  
136 basis of deep learning algorithms (McCulloch & Pitts 1943). They were inspired by the functionality  
137 of human brains, how the biological neurons communicate within each other using complex intercon-  
138 nections. The NNs are algorithms that can process parallel information. Their basic components are  
139 called neurons (or units, nodes) that are organized in layers and are connected with links. There are

140 three types of layers: an input layer, one or more hidden layers, and an output layer. In a standard  
 141 artificial NN called fully connected NN, the values from the input layer are connected to the output  
 142 layer, or prediction layer, through a series of hidden layers that are called fully-connected layers. Each  
 143 neuron from one layer is connected to every neuron of the next layer. In Figure 1 we show the neu-  
 144 ron model, defined as  $a = \sigma(\sum_i w_i x_i + b)$  where  $x_i$  is the input value,  $w_i$  is the weight of  $x_i$ ,  $b$  is  
 145 the bias,  $\sigma$  is an activation function, and  $i$  is the index in the previous layer. The weights represent  
 146 the connection strength between neurons, the biases are constant additive terms, while the activation  
 147 function is used to introduce non-linearity to NNs. In this complex mapping process from the input  
 148 to the output layer, we adjust the weights and the biases, which are optimized by a learning algorithm  
 149 during the training process. There are two phases within the learning process, the feed-forward and  
 150 the backpropagation. In the feed-forward phase the input data are passed through the layers and we  
 151 calculate the output values. In the final step of the feed-forward phase, we calculate the error (loss) be-  
 152 tween the predicted and ground truth value of the output layer for every sample of the training dataset.  
 153 Then, this error is backpropagated to adjust the weights and the biases in the backpropagation phase.  
 154 The iterative optimizing process in which we minimize the error itself is called the stochastic gradient  
 155 descent. These two phases are repeated until an acceptable loss value is reached.

156 [Figure 1 about here.]

Convolutional neural networks (CNNs) are a special kind of NNs able to process data having a  
 grid-like structure such as images (2D or 3D) as well as temporal time series (1D) (LeCun et al. 2015).  
 The main building elements are convolutional, pooling (downsampling operation) and fully-connected  
 layers. Stacking these different layers defines a CNN architecture. The difference between CNNs and  
 traditional NNs are the convolutional layers acting as a variety of filters by using a mathematical  
 operation called convolution defined as

$$f(t) = (x * w)(t) = \int x(a)w(t - a)da, \quad (1)$$

157 where  $x(t)$  is the input (image or time series) or the output of a previous layer,  $w(t)$  is the kernel or  
 158 filter,  $a$  is a dummy variable, and  $f(t)$  is the output feature map. The filter is smaller than the input data,  
 159 thus the multiplication is always applied between a filter-sized patch of the input data and the filter, and  
 160 this operation is repeated over the whole input data. In every convolutional layer, many feature maps  
 161 are estimated from different kernels, and the values of the kernels are the weights being optimized  
 162 during the training. These convolutions help the model to leverage three important concepts: sparse  
 163 interactions, parameter sharing and equivariant representations. The sparse connectivity is achieved by  
 164 making the filter smaller than the input, thus only a small local patch of data is interconnected unlike  
 165 in fully-connected layer where all neurons from one layer are interconnected. This property implies

166 that fewer parameters need to be learned and that the output requires fewer operations to be computed.  
167 Next, parameter sharing refers to the fact that during the convolution the weights within one filter are  
168 used in every position on the input. This, further, reduces the number of the parameters of the model.  
169 The parameter sharing introduces another property which is called equivariance to translation. This  
170 property allows the filter to discover features of objects within the data, while the position of the object  
171 does not need to be fixed in order to be detected by CNN.

172 As in NNs, the values of the filters, the weights, are learned during the training process of the  
173 network. Therefore, the network learns what types of features to extract from the input data by min-  
174 imizing the loss function. It is important to emphasise that the network learns more than one filter at  
175 a time, and multiple convolutional layers are applied in series. This allows decomposing input data to  
176 the features of higher abstractions. Same as in NNs, the (linear) convolution layers are followed by  
177 (non-linear) activation functions in order to obtain activation maps from the feature maps.

### 178 **2.3 CNN earthquake detection model**

179 In this study we use the CNN architecture from Majstorović et al. (2021), also shown in Figure 2A,  
180 where the full process of training and selecting the optimal hyperparameters is explained. The CNN  
181 model is trained to recognise earthquake signals within the continuous recordings that in very high  
182 percentage contain noise signals, which comprise numerous anthropogenic and unknown geophysical  
183 signals. Its architecture consists of seven convolutional and two fully-connected layers. Each convo-  
184 lutional block has 32 outputs that corresponds to a kernel of size 3, a stride of 2 and padding of 1,  
185 and it is followed by the rectified linear (ReLU) activation function. The last fully-connected layer is  
186 followed by the sigmoid activation function. Based on our samples' shape the input layer has a di-  
187 mension  $3 \times 500$ , corresponding to three components and 500 time steps (which is a time series of 25  
188 seconds sampled with frequency of 20 Hz). The output of the CNN is a scalar between 0 and 1 that  
189 represents the CNN estimation whether the input sample belongs to the negative or positive class, i.e.  
190 whether the CNN model recognizes an earthquake or not within the 25 seconds time window. During  
191 the training process the data is split into training (80%), validation (10%) and evaluation (10%) sets.  
192 To train our CNN model we use the stochastic gradient descent optimizing algorithm with a learning  
193 rate of  $10^{-2}$ , a momentum of 0.9 and a batch size of 512 samples. If during the training the model  
194 does not improve for 50 epochs on the validation dataset, the training process is stopped.

### 195 **2.4 Ten CNN training runs**

196 The CNN model, as any other deep learning algorithm, is stochastic by nature. This randomness comes  
197 from the weight initialisation and the training process by changing the order of the training samples.

198 Consequently, retraining a CNN model with the same dataset and hyperparameters yields different  
199 weights and biases, and this might introduce slightly different predictions. If by introducing small  
200 changes in the training set, we obtain small differences in the output of the trained model, we can argue  
201 that the training algorithm is stable (Bousquet & Elisseeff 2002; Charles & Papailiopoulos 2018). To  
202 consider the stochastic nature of the CNN models and to test for the stability, we train our model by  
203 using ten different random orders of the training dataset, thus we store the 10 model instances. The  
204 average accuracy on the evaluation set, having the same number of positive and negative samples,  
205 is 0.9758 and the standard deviation between the model runs is  $8 \cdot 10^{-4}$  (the full evaluation report  
206 is shown in Table S1): the models seems to be very consistent among them. The first trained model  
207 instance is referred to as model A, and it is used as the reference model.

## 208 **2.5 Interpretation methods**

209 When we talk about interpretation, the goals are numerous: checking the limits of the model, finding a  
210 way how to improve the model, better understanding the physics of the system we are modeling, etc.  
211 In this study, we want to tackle these different aspects, so we explore different types of interpretation  
212 techniques. First, in order to check the limits of the model and potentially improve it, we need a way to  
213 understand what is happening inside the model. For this, a first possibility is to extract and visualize the  
214 weights of the model, which in the case of CNNs are the kernel filters. However, in our case the filters  
215 are not visually interpretable as they are small (the filter size within our CNN model for the first layer  
216 is set to 3x3 which corresponds to the 3 components of our input array and the size of the temporal  
217 window (3 time steps), and for other layers the filter size is 3x32). Nonetheless, we can analyze the  
218 decision processes, for targeted samples, by extracting and visualizing the sequential transformation  
219 of the input data through the network, i.e. the feature maps, which was originally done in 2D CNNs by  
220 Krizhevsky et al. (2012); Zeiler & Fergus (2014). The feature maps visualisation (FMV) technique is  
221 developed in Section 2.5.1.

222 However, when visualizing feature maps we are not able to understand how a model relates the  
223 input and the output. In order to analyze the model at a global scale and understand what it does on  
224 average, a good strategy is to determine class prototypes (optimal inputs). For every class, we estimate  
225 what would be the 'mean' sample. Several studies (Simonyan et al. 2014; Olah et al. 2017) proposed  
226 a method, called backward optimisation, able to generate a prototype signal for each class with an  
227 iterative process. We explore the backward optimization (BO) technique and develop it in Section  
228 2.5.2.

229 Moreover, in the scope of analyzing the link between input and output for individual decisions  
230 (by analyzing the individual waveform samples from the evaluation dataset), many methods have

231 proposed ways to interpret why a certain data is classified as a certain class. They almost all rely on  
 232 the generation of a heatmap showing which part of the input plays in favor or against its classification.  
 233 A first family of methods, called sensitivity analysis or gradients/saliency methods (Simonyan et al.  
 234 2014), is proposing to estimate for each pixel (or time step) what is the gradient of the model function  
 235 (how does the output varies when modifying this pixel). A second family of methods, called layerwise-  
 236 relevance propagation (LRP) (Bach et al. 2015), propagates a relevance value from the output to the  
 237 input, expressing how important each pixel (or time step) is for the decision. The LRP methods were  
 238 shown to be more stable than the gradients methods, especially in deep networks (Montúfar et al. 2014;  
 239 Balduzzi et al. 2018). Moreover, we want to explain the decision function and not its variations, so we  
 240 will focus on the LRP technique and give more details in Section 2.5.3. The three identified methods,  
 241 depicted in Figure 2B, are all using a scientific visualisation to explain how a model works: they are  
 242 referred to as 'visualization methods' (Ras et al. 2021). Moreover, as throughout the analysis we do not  
 243 alter the trained weights and biases of the CNN models, these methods belong to the post-modeling  
 244 (post-hoc) methods (Barredo Arrieta et al. 2020).

### 245 2.5.1 Feature map visualisation (FMV)

246 Once a model has been trained, we can have access and visualize the learned weights, which are  
 247 the kernel filters in the case of convolutional layers. Moreover, by inputting a new sample from the  
 248 evaluation dataset into the model, we can also extract the sequential transformations of the input data  
 249 by every filter of every layer: this is what we call 'feature maps' (Krizhevsky et al. 2012; Zeiler  
 250 & Fergus 2014). The schematic representation of the feature maps, belonging to the convolutional  
 251 layers, along with their sizes is shown in Figure 2A with light blue blocks. The feature maps helps us  
 252 understand how the convolutional filters transform the input data at intermediate layers. As mentioned  
 253 previously, having more convolutional layers implies having features of higher abstraction. Therefore,  
 254 it is expected that the features extracted in the first convolutional layer have temporal details while the  
 255 features in the last 7th convolutional layer should be more general. We can define the decomposition  
 256 within the CNN as

$$\begin{aligned}
 f_1 &= C_1(x) \\
 a_1 &= \sigma(f_1) \\
 f_2 &= C_2(a_1), \\
 a_2 &= \sigma(f_2) \\
 &\vdots
 \end{aligned}
 \tag{2}$$

257 where  $x$  is the input,  $C_k$  is the  $k$ -th convolutional layer,  $\sigma$  is the activation function,  $f_k$  is the feature  
258 map of layer  $k$ ,  $a_k$  is the activation map of layer  $k$ . In this study, we visualize the feature maps  $f_k$ .

### 259 2.5.2 *Backward optimisation (BO)*

260 In the following, we give a short overview of the method called backward optimisation (BO) or feature  
261 optimisation (Simonyan et al. 2014; Olah et al. 2017). The schematic representation is shown in Figure  
262 2B. The goal of this technique is to iteratively generate a synthetic input, or prototype, based on the  
263 trained model for a specific class. By maximizing the output likelihood, in our case the likelihood of  
264 the CNN to correctly predict an earthquake signal, we can perceive what is the optimal earthquake  
265 signal for the CNN model.

266 There are several important steps in this method (Figure 2B). First, we freeze the trained CNN  
267 model's weights and biases. This means that during the new iteration process those parameters are  
268 not updated. Next, a vector with the same size of the input sample (here 500x3) is passed through the  
269 network, the prediction is obtained and compared with the desired class (either earthquake or noise  
270 class in case of the CNN earthquake detection model). The misfit is calculated using the same loss  
271 function, and the backward optimisation algorithm is used to update the input sample values. The  
272 network is using the previous knowledge (weights and biases) to update the input sample for a desired  
273 class. This is repeated during several iterations, until convergence. The process is quite similar to how  
274 the model is originally trained, however here the updated variables are the values of the input sample.  
275 The final input is hereinafter referred as the optimal input.

276 The optimal input solution is thus generated iteratively. The input vector can be initialized in dif-  
277 ferent ways: only zero values (the exact "zero input"), an array of random values ("random input")  
278 and an array matching real-dataset sample ("real input"). The zero input generates an invariant syn-  
279 thetic optimal input that is completely novel with respect to the starting initial array. It allows the  
280 trained CNN detector model to generate one perfect earthquake (for positive class) or noise (for neg-  
281 ative class) solution. Here, the term "perfect" relates to having a maximum likelihood for earthquake  
282 solution and minimum likelihood for the noise solution. Unlike the zero input, we can randomly chose  
283 the initialisation values within the input array with a Gaussian process and have as a result a collec-  
284 tion of optimal inputs. The obtained optimal inputs are novel and almost never resemble the original  
285 training samples (McGovern et al. 2019). The random inputs allow us to perform an ensemble study  
286 since we are able to generate an endless possible optimal input solutions. Third, in the case of a real  
287 input initialization, the optimal input solution is physically realistic (McGovern et al. 2019). Hence,  
288 this solution could be understood as an improvement of an existing earthquake signal by finding the  
289 needed changes in order to maximize the detection rate of the actual earthquake signal. It allows us to

290 understand which parts of the training samples need more attention or modification in order to improve  
 291 the actual detection rate.

### 292 2.5.3 Layer-wise relevance propagation (LRP)

293 Layer-wise relevance propagation (LRP) aims at determining which parts of a particular input vector  
 294 contribute most strongly to a NN decision, as a kind of heatmap. Unlike the backward optimisation,  
 295 LRP is a non-iterative method applied to one input real sample at a time, by propagating the relevance  
 296 value backward using purposely designed local propagation rules. The original method was introduced  
 297 by Bach et al. (2015) and has been applied in different scientific fields to unravel the decision making  
 298 process of large variety of deep learning models. Some applications are understanding relevant fea-  
 299 tures in text-based data (Arras et al. 2017), patch-based learning of video data (Anders et al. 2019),  
 300 Alzheimer’s disease patterns (Böhle et al. 2019), climate patterns (Toms et al. 2020). The schematic  
 301 representation is shown in Figure 2C.

302 The goal of LRP is to define a measure called relevance  $R$  over the input vector taking into account  
 303 the model’s decision. This is accomplished by respecting the conservation property, meaning that the  
 304 contribution received by a neuron must be redistributed to the lower layer in equal amount (Montavon  
 305 et al. 2018, 2019). There are three steps: in the first step the weights and the biases of the trained model  
 306 are frozen. Next, we forward pass the input array through the DL model and we collect the activations  
 307 at each layer. In the third step, the prediction of the last output layer is backpropagated using a set  
 308 of propagation rules that satisfies the conservation law. Let’s mark with  $j$  the neurons at layer  $l$ , with  
 309  $k$  the neurons at the lower layer  $l - 1$  and with  $R$  the relevance. Then, the conservation law implies  
 310  $\sum_j R_j = \sum_k R_k$ . The implementation relies on a specific set of propagation rules. Let’s describe the  
 311 neuron activation  $a_k$  by the equation

$$a_k = \sigma\left(\sum_j a_j w_{jk} + b_k\right), \quad (3)$$

312 where  $a_j$  are the activations from the previous layer  $l$ ,  $w_{jk}$ ,  $b_k$  are the weights and biases of the neuron.  
 313 One propagation rule that has shown to work well in practice (Montavon et al. 2018, 2019) is the  $\beta$ -rule  
 314 (or  $\alpha\beta$ -rule, where  $\alpha = 1 + \beta$ ) defined as

$$R_j = \sum_k \left( (1 + \beta) \frac{a_j w_{jk}^+}{\sum_j a_j w_{jk}^+} - \beta \frac{a_j w_{jk}^-}{\sum_j a_j w_{jk}^-} \right) R_k, \quad (4)$$

315 where  $()^+$  and  $()^-$  denote the positive and negative parts, and the constrain  $\beta \geq 0$  is valid. To avoid  
 316 numerical instability by dividing with zero, a stabilizing term  $\varepsilon$  can be introduced. The parameter  $\beta$

317 controls how much weight is given to the positive/negative relevance within the layered graph structure  
318 of DL model. For example, by setting  $\beta = 0$  we only consider the positive relevance, and by setting  
319  $\beta = 1$  we are allowing the negative relevance to have an impact to the final LRP solution. The  
320 positive relevance being propagated to the input layer highlights the relevant parts of the input layer,  
321 and vice versa. To understand which part of the input three-component waveforms are relevant for the  
322 earthquake prediction it might be meaningful to focus on several possible LRP- $\beta$  solutions (Montavon  
323 et al. 2018) by varying the  $\beta$  value.

324 [Figure 2 about here.]

### 325 **3 RESULTS**

326 As explained in Section 2.5, we use the FMV, BO and LRP methods to explore the decision process of  
327 our trained CNN model (see Figure 2). In order to study the CNN model in an objective way and obtain  
328 complementary solutions for the different methods, we select a set of waveforms of earthquakes char-  
329 acterized by different distances and magnitudes and some noise samples, and we use them throughout  
330 the analysis. These samples are extracted from the evaluation dataset, so that the FMV, BO and LRP  
331 solutions are independent from the examined model. The ten chosen earthquakes are shown in Figure  
332 S1.

333 We also use the interpretation methods to explore how our model is stable in terms of the training  
334 process and the existing architecture. For this purpose we analyze ten CNN detector models from  
335 Section 2.4, while the results for the model A are used as a reference (see Table S1).

#### 336 **3.1 Feature map visualisation**

337 Using the FMV we can visualize the feature maps of the intermediate convolutional layers. Based  
338 on our CNN architecture shown in Figure 2A there are 36 output channels for each of the seven  
339 convolutional layers, that are the products of the convolution using the filters of size 3x3 (first layer)  
340 and 3x32 (all other layers). Each output channel is associated with the feature map (light blue blocks  
341 in 2A) and its size is decreasing from 251 down to 4 features.

342 In Figure 3, for the first time to our knowledge, we show how an earthquake and a noise sample  
343 are seen by a DL model trained for earthquake detection by visualizing the feature maps. Clearly, the  
344 earthquake feature map differ from the noise feature maps for all CNN layers. We notice how the bulk  
345 of phases (positive or negative peaks, i.e. activation of the neuron), including P- and S- phase arrivals,  
346 is visually present throughout the first five layers (see several other earthquake samples in Figures  
347 S2-S10), while these characteristics are evidently absent in the noise feature maps.

348 To understand how the CNN model treats earthquake samples of different distance and magnitude  
 349 values, we show in Figure 4 and Figure S11 a comparison between ten earthquake samples (A-J) and  
 350 two noise samples (K-L). For better visual comparison the vertical scale is the same per channel across  
 351 all included samples. We are visualizing only the 5th convolutional layer (L5): in L5, the original  
 352 sample (500 time steps) is already reduced into vectors of size 16, but the bulk related to seismic  
 353 phase arrivals is still visually present. At this point, we distinctly notice that some feature maps are  
 354 common: a) within the layer, b) across different earthquake samples. For example, if we consider the  
 355 earthquake sample A, we notice that all channels numbered with 2, 10 (Figure 4), 16, 18, 19, 22,  
 356 23, 26, 27, 29, 32 (Figure S11) contain a left-sided amplitude peak. Moreover, these channels are  
 357 quite similar and consistent also for earthquakes with different distances and magnitudes (see Figure  
 358 4A-F and Figure S11A-F). Yet, we notice that the pattern is changing, but being consistent, for the  
 359 earthquakes with the epicentral distance larger than 50 km (see Figure 4G-J and and Figure S11G-J).  
 360 These observations could point to the fact that the CNN model defined a relevant highly abstract latent  
 361 space where 'generalized' earthquakes (of different distance and magnitude values) exists.

362 [Figure 3 about here.]

363 [Figure 4 about here.]

364 The aforementioned occurrence of the repetitive feature maps suggests that our CNN model is  
 365 stable in terms of the architecture design, since these maps are not random. To explore this hypothesis  
 366 even further, we proceed to numerically quantify the resemblance between the feature maps by cal-  
 367 culating the Pearson correlation coefficients  $r(x_i, x_j)$  (Freedman et al. 2007), where  $x_i$  is the feature  
 368 map and index  $i$  stands for the channel number and runs from 1 to 32. We quantify the resemblance  
 369 between the feature maps of the 5th layer for earthquake sample B (Figure 4B and Figure S11B) ob-  
 370 tained for the reference CNN detector model A (from Section 2.4 and Table S1). We calculate that  
 371 49% out of 32 channels have at least one or more channel pairs with high correlation coefficient of  $r >$   
 372 0.8 (see Figure S12). Thus, we are finding pairs of channels with quite similar patterns. Beyond quan-  
 373 tify the stability of the network, the observed redundancy indicates that less than 32 channels could  
 374 be sufficient for the CNN detector model to provide correct classification of the positive sample. This  
 375 analysis shows how the FMV could be used to guide the design of CNN architecture. For our case  
 376 a less complex one, might provide similar performance. Yet, the presented analysis cannot provide  
 377 us with straightforward information on how to design the optimal CNN architecture in terms of the  
 378 number of layers or channels without any additional testing.

379 The stability of our CNN detector can be additionally supported by studying the feature maps of  
 380 ten different training runs of the CNN model. We train ten different models by randomly initializing

381 weights and by changing the order of the training samples (while keeping the architecture and the  
 382 training hyperparameters), see Section 2.4. We extract the feature maps of the 5th convolutional layer  
 383 of the earthquake sample B for the ten CNN models. It is important to underline that, due to the  
 384 randomness introduced in the training process, the indices of the channels are not fixed between the  
 385 different runs (Figure S13). By visual comparison, we observe that the feature maps are extremely  
 386 similar, for example we see a feature map with the left-sided amplitude peak repeating consistently  
 387 (see Figure S13). However, as much as we can see some stable patterns among feature maps, we can  
 388 also notice that there is a level of uniqueness related to these feature maps, meaning that some feature  
 389 maps are unique for each training run. We then quantify this visual comparison, by calculating the  
 390 correlation coefficients  $r(x_i, y_{j,i})$ , where  $x_i$  stands for the feature maps of the reference model A,  $y_{j,i}$   
 391 stands for the feature maps of different training runs with index  $j$  running from B to J (see Section 2.4  
 392 and Table S1), and again index  $i$  represents the channel index. Since in this case we are only examining  
 393 whether the feature maps of different training runs are similar, for each channel of  $x_i$  we keep only  
 394 the related correlation coefficient of  $y_{j,i}$  that has the maximum value, and we repeat this for every  $j$ .  
 395 We notice that in average more than 75% of the channels associated with the reference model A have  
 396 high correlation values of  $r > 0.8$  (see Figure S14). The high resemblance among the feature maps  
 397 of different training runs of the CNN detector model are quite high, proving that the existing CNN  
 398 architecture is quite stable, despite the randomness during the training process.

### 399 **3.2 Backward optimisation**

400 The input data in our CNN detector (see Section 2.2) is shaped as  $3 \times 500$  array where 3 stands for the  
 401 number of components (E-W, N-S, Z) and 500 stands for the time steps. Following the details in Sec-  
 402 tion 2.5.2 the optimal input obtained by applying the backward optimisation technique is also of shape  
 403  $3 \times 500$ . When maximizing the output likelihood for the class associated with the earthquake signal,  
 404 we expect to obtain waveforms that resemble the three components of real seismograms. Intuitively,  
 405 they should, to some degree, be similar to the samples of the training dataset.

406 To better evaluate the optimal input solutions for our CNN detector model, we proceed by studying  
 407 the solutions of ten different training runs (Section 2.4). In Figure 5A we show the optimal inputs for  
 408 these ten models, while the input array is initialized with zeros. Each input is updated separately for  
 409 each model for 5000 iterations. While the optimal input training is converging correctly for all ten  
 410 models, we notice that often the first few iterations (2 to 3) are diverging (increase of the loss) before  
 411 converging (see Figure S15). This can be due to the fact that the 'zero initialisation' is out of the  
 412 training sample domain of the CNN models. We notice that the randomness introduced during the

413 training process of the CNN models affects the final optimal inputs, thus they are different for each  
 414 run. However, we can observe some common characteristics present in all these solutions.

415 First, all optimal inputs show consistent local amplitude increase simultaneously on all three chan-  
 416 nels, corresponding to three components of our training sample. Next, we notice that the Z components  
 417 related to the models A, B, E, G, J have their absolute amplitude exceeding 1. Thus, 83% of the total  
 418 components (25/30) is consistent with our normalisation approach. In the time domain, the optimal  
 419 solutions are showing a more complex behavior, which we can interpret as if there were occurrence of  
 420 multiple earthquakes in the time window. This behavior is consistent with the BO results on images  
 421 (Simonyan et al. 2014; Olah et al. 2017) where multiple artifacts of the learned object are repeated on  
 422 the optimal image solution. This might be because the CNN model was trained using a broad range  
 423 of local, regional, and teleseismic events, and this diversity encourages the complex multiple-event  
 424 behaviour we are seeing. We also notice that the P- and S- waves are not really identifiable in the  
 425 simulated inputs. This is interesting as it means that such samples, which would not be identified as  
 426 earthquakes by any specialist, have a perfect detection score for this CNN model. We can see how  
 427 part of the decision process is different between a human and a CNN. In Figure 5B we show the com-  
 428 parison between the optimal inputs for all ten CNN models and the average amplitude spectrum of  
 429 the training positive samples. We notice that the optimal inputs have enhanced high frequency content  
 430 respect to real earthquake samples. By performing the same test for the negative class (noise), we can  
 431 notice a clear difference especially in terms of the amplitude values (see Figure S16): the negative  
 432 class is characterized by very low amplitude values. This does not match with the noise samples used  
 433 for the training, since our noise samples have the amplitude bounded within -1 to 1 (see Figure 3C).  
 434 However, the low amplitude values indicates that for the CNN model, the optimal noise sample is  
 435 supposed to have an amplitude as low as possible.

436 [Figure 5 about here.]

437 We also study how the real earthquake samples are modified when we apply the BO technique.  
 438 This test illustrates the modification of an earthquake sample to improve the confidence of the CNN  
 439 model. We perform 5000 iterations on an input array initialized with a real earthquake sample from  
 440 the evaluation dataset using the reference CNN model A (Section 2.4 and Table S1). In Figure 6 we  
 441 show the raw Z component as well as the optimal input waveform at their full time scale, a zoom of the  
 442 interval between 2.5 and 7.5 seconds, the logarithmic ratio between the two waveforms in the time and  
 443 frequency domain (results for E and N components are shown in Figure S17 and S18, respectively).  
 444 The logarithmic ratio is calculated as  $\log\left(\frac{|A_{in}|}{|A_{out}|} * 100\right)$ , where  $A_{in}$  is the raw waveform and  $A_{out}$   
 445 is the optimal input waveform or the modified one. A ratio value above 2 indicates a decrease of the  
 446 optimal input amplitude with respect to the raw one. We notice that the upgrades are quite small in

447 amplitude, mostly less than one quarter of the magnitude. This is not surprising, since the prediction  
 448 values of these earthquake samples before the BO modification are quite high, meaning that the starting  
 449 loss values are already small. Moreover, we observe that the modifications are occurring within the full  
 450 time window, yet, more often at the beginning of the time window (see Figure 6, S17 and S18 for Z,  
 451 E, N component, respectively). These are not predominately positive or negative, and are earthquake-  
 452 dependent. The modification occurring less than 5 seconds, and the absence of it between 5 to 10  
 453 seconds, might indicate that the BO technique acts in a favor of a noise reduction before the P-arrival  
 454 time. In Figure 6 we also see that the alterations are associated with high frequency content, which  
 455 suggests that CNN model learned what frequencies are relevant for the earthquake detection.

456 [Figure 6 about here.]

457 We also compare the modifications for ten different training runs of the CNN model using the  
 458 same earthquake sample B (see Figure S19). The results support previous observations from the FMV  
 459 technique. The models perform similar alterations to the input signal, supporting the stability of our  
 460 CNN model architecture design.

### 461 3.3 Layer-wise relevance propagation

462 In this section we attempt to understand which parts of the waveform are relevant for its classification  
 463 as an earthquake or noise, by means of the LRP method. The relevance values depend on the used  
 464 LRP propagation rule. In Figure 7 we show the solutions obtained with the LRP- $\beta - 0$  rule defined by  
 465 Eq. 4 for ten earthquakes (same as in Figure 4) and the reference model A (Section 2.4 and Table S1).  
 466 We use  $\varepsilon = 10^{-6}$  to stabilize the solutions. The results show high values of the relevance matching  
 467 with the position of the earthquake in the time window, with peak values associated with P and S time  
 468 arrivals (Figure 7). We further observe a consistent time distribution of the high relevance for the three  
 469 components. Moreover, time distribution of high relevance closely follows the bulk of seismic arrivals.  
 470 In more details, for events close to the station, with short S-P time, the relevance is visually much more  
 471 compact in time (Figure 7A-F), respect to distant events for which the earthquake signature (S-P time,  
 472 or P wave plus coda) is longer (Figure 7G-J).

473 To explore this property in more details, we estimate the spreading of the positive relevance value  
 474 with respect to time as the standard deviation,  $\sigma_{LRP}$ , for almost 6000 samples in the evaluation dataset.  
 475 The standard deviation of the LRP solution is calculated by taking into account the full time span of  
 476 500 time steps, where the relevance values are considered as steps' weights. Thus, this measure is not  
 477 related to specific time, but to a full waveform sample. The  $\sigma_{LRP}$  as a function of epicentral distance  
 478 is reported in Figure 8A. Our analysis confirms that the earthquake samples with larger distances are

479 related to larger  $\sigma_{LRP}$ , meaning that the positive relevance is more spread. This again reflects that  
 480 the duration of an earthquake is related to signals having larger S-P time and/or P wave plus coda.  
 481 The model learns in a general way what is an earthquake, regardless of the position of the earthquake  
 482 related signals (e.g. P, S, coda waves) in the time window. Clearly, this is achieved without explicitly  
 483 pointing to the earthquake position within the corresponding time window. We observe that the high  
 484 relevance values follow the positions of the multiple non-overlapping earthquakes in the time window,  
 485 even though the CNN detector is trained with the positive samples having a single event with a fixed  
 486 P-onset (see Figure S20). However, more relevance is associated with the earlier event. Beyond this  
 487 general property, we can see in Figure 7 that the relevance shows a high frequency pattern. Also, for  
 488 the larger distances sometimes particular time steps are given more relevance without a clear meaning.  
 489 How these patterns are involved in the decision process of the CNN is not fully interpreted, leaving  
 490 room for possible improvement of our earthquake understanding.

491 We also calculate the relevance values for the  $LRP-\beta - 0.5$  (see Figure S21) and  $LRP-\beta - 1$  rules  
 492 (see Figure S22). Unlike the  $LRP-\beta - 0$  rule, these allow negative relevance values to be propagated  
 493 up to the input array. In the context of the detection task, negative relevance is associated with the  
 494 negative (noise) class. Thus, time periods associated with high negative relevance values are those that  
 495 do not help the CNN detector to classify earthquakes in the positive class. The results are consistent  
 496 for both  $LRP-\beta - 0$  and  $LRP-\beta - 0.5$  rules, while the solutions for the  $LRP-\beta - 1$  are less stable. This  
 497 might be because we ask the LRP technique to consider the positive input array thorough an increasing  
 498 impact of negative relevance, which could be in contradiction. Nevertheless, we do not see consistent  
 499 patterns related to the negative relevance for both  $LRP-\beta - 0.5$  and  $LRP-\beta - 1$ . Such patterns, which  
 500 are rare, are mostly associated with the time periods before P-arrivals (first 5 seconds) and the coda  
 501 parts of the earthquakes. When the negative relevance patterns occur during the P and S phase arrivals,  
 502 the associated amplitudes are less notable than those related to the positive relevance. Further, if we  
 503 consider the  $\sigma_{LRP}$  value, unlike the  $LRP-\beta - 0$  solution, both  $LRP-\beta - 0.5$  and  $LRP-\beta - 1$  remain more  
 504 localised. For the  $LRP-\beta - 0.5$  solution the position of the positive relevance seems to be precisely  
 505 linked to the P and S phase arrivals, which is not the case for the  $LRP-\beta - 1$  solution.

506 [Figure 7 about here.]

507 [Figure 8 about here.]

508 As for the FMV and BO technique, we apply the  $LRP-\beta - 0 - 0.5 - 1$  rules for ten different  
 509 training runs of our CNN model (Section 2.4 and Table S1). For this test we use only the earthquake  
 510 sample B (Figure 4 and S1). The LRP results are quite consistent between all models for the  $LRP-$   
 511  $\beta - 0$  and  $\beta - 0.5$  rules, and the results for the  $LRP-\beta - 1$  are varying more (see Figure S23, S24,

512 S25 for the LRP- $\beta - 0$ ,  $\beta - 0.5$ ,  $\beta - 1$  rule, respectively). For the results associated with the LRP-  
 513  $\beta - 0$  and  $\beta - 0.5$ , the main characteristics and the shape of the relevance values are the same across  
 514 different training runs. For the LRP- $\beta - 1$ , the differences across different training runs are more  
 515 prominent. This observation again indicate the relative instability of this rule, that propagates more  
 516 negative relevance, applied on the positive input array.

517 By applying the LRP- $\beta - 0$  rule for the negative (noise) samples from our evaluation dataset (also  
 518 around 6000 samples), we notice that the relevance is negative and fully spread within the whole time  
 519 window (see an example for LRP- $\beta - 0$  in Figure S26). Having a negative relevance for the LRP-  
 520  $\beta - 0$  (that only propagates the positive relevance) is not impossible. Our CNN model is trained to  
 521 recognise earthquakes from noise samples, by providing us with a probability value between 0 and  
 522 1. We should obtain the positive (resp. negative) relevance for the earthquake (resp. noise) samples,  
 523 if those are classified as positive (resp. negative). In a detection task, the positive relevance indicates  
 524 time steps that help the CNN model to classify the sample as positive, while the negative relevance  
 525 indicates the opposite: time steps that help the model to classify the sample as negative. In Figure 8B  
 526 we show  $\sigma_{LRP}$  for the positive and negative samples from the evaluation dataset. The plot shows that  
 527 the CNN detector is indeed more localised for the positive samples, as the positive  $\sigma_{LRP}$  values are  
 528 in average smaller. The LRP- $\beta - 0$  is spread within the whole time window for the negative samples,  
 529 which is reflected by having larger  $\sigma_{LRP}$  values. This implies that each time is equally relevant for the  
 530 detection task.

531 To explore whether the LRP- $\beta$  solutions are robust, we also check the relevance solutions by  
 532 using the LRP- $\varepsilon$  propagation rule (see Figure S27) (Montavon et al. 2018). We can conclude that  
 533 the solutions related to this rule are highly linked with the solutions presented by the LRP- $\beta$  rules: the  
 534 positive relevance values for the earthquake samples are associated with the location of the earthquakes  
 535 within the sample window. This shows that despite the large variety of possible LRP rules, the main  
 536 findings are in agreement.

## 537 **4 DISCUSSION**

538 In this study we introduce three different methods to explain how a CNN detector model makes pre-  
 539 diction between two classes, the earthquake (positive) and the noise (negative) class. The samples  
 540 related to these two classes are constructed as 25 second window that contain either earthquake events  
 541 of different distance and magnitude values (positive samples) or signals of many unidentified sources  
 542 (negative samples). As an output this model gives a probability value, when being closer to one it indi-  
 543 cates that the positive class is predicted. This model reaches quite high accuracy performance (97%).  
 544 At this point of analysis, it is clear that the CNN detector model extracts relevant characteristics from

545 both type of samples to make a good prediction, however a question remains: can we identify these  
546 characteristics? We use three techniques to visualize how the trained model treat the input array, in  
547 order to get a better insight into what are these relevant characteristics.

548 By considering the results obtained with one CNN detector model and several earthquake samples,  
549 we can deduce that the characteristics we are identifying are relevant for the different earthquakes. This  
550 is important since our CNN model should be able to generalize to earthquakes of different distances  
551 and magnitudes, once it is applied on the continuous data (Majstorović et al. 2021). We notice that  
552 earthquake diversity is well captured during training. Despite the waveforms differences of the earth-  
553 quake samples the CNN detector model is able to generalize well by generating feature maps of similar  
554 shape (Figure 4). Therefore, by using the FMV technique we note that the relevant characteristics are  
555 identified as very abstract feature maps that define a high-dimensional space where all earthquakes can  
556 eventually fit. The existence of this abstract space is the reason why the CNN model performs with  
557 high accuracy. Since we can obtain relevant maps for different earthquakes of various signal to noise  
558 ratios, it could indicate that the extracted features have some physical meaning (Figure 4).

559 The results from the BO technique are more difficult to interpret. If we try to answer the question  
560 of how does the optimal earthquake looks like, the answer is that it is not what a human would expect.  
561 This result suggests that the optimal processing for earthquakes detection is perhaps more complex  
562 than some standard methods (such as STA/LTA, etc). The optimal input results reflect the complex  
563 nature of the patterns learned by the CNN model, also shown in previous studies (Olah et al. 2017). Yet,  
564 by comparing the optimal solutions for the earthquake and the noise classes, we might conclude that  
565 the amplitude is quite an important characteristic, while the shape and location of the peaks (events)  
566 seem less relevant (see Figure 5 and 6). Indeed, the earthquake optimal solutions span a much larger  
567 range of amplitude values, while those for the noise optimal solutions are more constrained.

568 From the LRP technique, we observe that the high relevance values follow the position of the  
569 earthquake related signals (e.g. P, S and coda waves) within the sample window (Figure 7). The model  
570 learns in a general way what is an earthquake without being explicitly pointed to the earthquake  
571 position within the corresponding time window during the training process. However, more relevance  
572 is given to times related to P and S phase arrivals than the coda (Figure 7). This does not imply that the  
573 LRP technique could be used as a stable technique for picking P- and S- phase arrivals. We also notice  
574 that some time steps with very high relevance are followed by time steps with low relevance, without a  
575 clear pattern on what differs these time steps (Figure 7). This could be subject to further explorations,  
576 however it is not entirely odd, since LRP relevance values are sample dependent (Montavon, 2019).

577 By comparing the results for different training runs, we can infer whether our CNN model is stable  
578 to the randomness we introduce during the training process. While comparing the interpretability

579 results we conclude that these results are coherent with the low accuracy variability between different  
580 models: the different runs of the models show very similar patterns and features. The feature maps have  
581 similar shape and amplitude values, and high correlation coefficients when compared to each other.  
582 The class prototypes estimated by BO are quite different between each model (indicating randomness  
583 can still play an important role), but many common characteristics are present in all optimal solutions.  
584 Moreover, the modifications found by optimizing a real earthquake sample are similar for the different  
585 runs. Finally, the LRP solutions associated with different model runs show that the relevance is of  
586 the same order of magnitude and the peaks are at the same location when being tested on the same  
587 earthquake sample.

588 In our analysis, the positive and negative classes are represented differently for the studied tech-  
589 niques, which is also a sign of a good model performance. For example, the feature maps of two  
590 classes differ in amplitude and shape (see Figure 4). However, unlike the earthquake feature maps,  
591 the noise maps are quite different from sample to sample. This might imply that the feature space  
592 related to noise samples is more complex and diverse than the earthquake feature space. Certainly, this  
593 is a reason why the detection problem is quite challenging. Even though the earthquakes share some  
594 common characteristics, they are buried into a random wave field whose sources we are not able to  
595 dissociate. When observing the optimal solutions for the noise class (initialized with zero values, see  
596 Figure S16), we notice that the main characteristic is to have low amplitude. When the LRP technique  
597 is applied to the negative class, we learn that the CNN needs all time steps to correctly predict this  
598 class: the relevance values are homogeneously spread within the full sample window. This behavior  
599 differs from the positive class, where the time steps related to the earthquake are more relevant than  
600 the others. It comes probably from the fact that in order to know if a sample is negative, all time steps  
601 should be detected as noise so they are all important.

602 From our analysis we can infer that the CNN detector model is extracting and using some relevant  
603 characteristics from the training samples to make good predictions. These characteristics seem to have  
604 a physical meaning, therefore applying these techniques to related research areas, that have the same  
605 research goal or similar subjects, might be quite interesting. For example, it would be interesting to see  
606 if we can use the feature maps to better understand some less known signals such as tremors, low- and  
607 very low-frequency earthquakes (Peng & Gomberg 2010). Furthermore, we could try to understand  
608 the black-box nature of the models that are used to predict laboratory earthquakes (Rouet-Leduc et al.  
609 2017). Especially since these models are trained for predicting the rupture time. Thus, it would be  
610 interesting to see whether the feature maps change over the time preceding the rupture, or check the  
611 properties of the signals (from FMV or LRP) in those time windows that we can use to predict the  
612 rupture.

613 We show in our study how interpretation techniques might be useful to assess whether a CNN  
614 model is stable with respect to the randomness of the training process. Each technique has a different  
615 indication of how the stability can be assessed based on the output result, and the consistency between  
616 the output results is the first indicator of the stability. Yet, the different interpretation methods are  
617 all additional measures to assess the robustness of a model. Furthermore, we also show how these  
618 techniques might be practical for the architecture or the database design. For example, if the prediction  
619 is not satisfying, using the FVM technique one can inspect whether the feature maps of samples  
620 belonging to different classes are significantly different for the predefined task. Furthermore, the LRP  
621 technique can be used to indicate which parts of the training samples, if any, is helping CNN to make  
622 good prediction. If the high relevance value does not match with the predefined targeted class in the  
623 time window or image, this might be indicative that the training samples need to be better defined.

624 Despite these techniques provide us with valuable insight about our CNN model, all three tech-  
625 niques are evaluated by a visual approach and we do not have a relevant ground truth. To overcome  
626 this lack of a numerical interpretation, it might be interesting to compare the results among different  
627 models that are trained with different datasets but the same architecture, in that case taking an advan-  
628 tage of existing toolbox (Woollam et al. 2022). Based on the performance of, for example, two models  
629 trained with different datasets but the same architecture, we could assess how the accuracy is related  
630 to feature maps, optimal solutions, or relevance values. The possible existing differences should not  
631 affect the general characteristics that we discussed through our study, for example: the feature maps  
632 between negative and positive samples being visually different, the optimal inputs reflecting the fre-  
633 quency content of the training samples, the LRP solutions indicating the meaningful samples within  
634 the time window.

635 It should be noted that it is relatively straightforward to apply these techniques on a standard CNN  
636 architecture like the one used in this analysis. For more complex architectures, like transformers or  
637 recurrent networks, as well as for regression tasks, other methods might be needed (Barredo Arrieta  
638 et al. 2020; Roscher et al. 2020; Samek et al. 2021; Ras et al. 2021; Linardatos et al. 2021).

## 639 5 CONCLUSIONS

640 We applied three different interpretation techniques in order to better understand the decision process  
641 of a DL earthquake detection model. We conclude that it is important to interpret the results of these  
642 techniques jointly, since they define different characteristics which are all important to have a full pic-  
643 ture of how the model perform. From our analysis we infer that the model learns some physical and  
644 meaningful earthquake characteristics without imposing any physical constraints during the training.  
645 These characteristics are different from what an expert seismologist would expect and are not defined

646 as simply as in other standard methods (such as STA/LTA, etc). In other words, the CNN model does  
647 not perceive an earthquake as humans do. The feature maps related to earthquake samples suggest  
648 that the CNN model, at the deepest architecture level, defines very abstract high-dimensional space  
649 where diverse earthquakes can eventually fit. The prototype earthquake signals propose that the model  
650 learned some complex patterns where amplitude is of high importance, while the shape and location  
651 of these patterns seem less relevant. Besides, the frequency content of earthquakes seems to be well  
652 captured, which implies that it is an important earthquake characteristic. The high relevance values as-  
653 sociated with the earthquake related signals (e.g. P, S and coda waves) indicate that the CNN learned  
654 in a general way what is an earthquake, since we never explicitly highlighted these signals during the  
655 training process. We conclude that the important strength of the CNN model is the ability to generalize  
656 well to many types of earthquakes, without having any other prior information about them during the  
657 training process. The CNN is able to perform very complex transformation on the input samples in  
658 order to obtain robust results. Therefore, we believe that this is also an opportunity to discover new rep-  
659 resentations of events that are still poorly understood, such as tremors or low-frequency earthquakes.  
660 As we showed, there are numerous perspectives how this study can be extended, and hopefully the  
661 seismological community will continue the effort in interpreting more DL models.

## 662 **ACKNOWLEDGMENTS**

663 This research received funding from the European Research Council (ERC) under the European Union  
664 Horizon 2020 Research and Innovation Programme (grant agreements 802777-MONIFaults). Some  
665 computations presented in this paper were performed using the GRICAD infrastructure (<https://gricad.univ-grenoble-alpes.fr>), which is supported by Grenoble research communities.

## 667 **DATA AVAILABILITY**

668 The data underlying this article were accessed from <http://mednet.rm.ingv.it/?more=1>, <http://iside.rm.ingv.it/instruments/network/IV> (INGV Seismological Data Centre 2006). Some  
669 parts of the processing algorithm make use of the Python codes available here  
670 <https://github.com/moboehle/Pytorch-LRP>,  
671 <https://github.com/utkuozbulak/pytorch-cnn-visualizations>.

## 673 **REFERENCES**

674 Alavi, A. H. & Gandomi, A. H., 2011. Prediction of principal ground-motion parameters using a hybrid method  
675 coupling artificial neural networks and simulated annealing, *Computers & Structures*, **89**(23-24), 2176–2194.

- 676 Anders, C. J., Montavon, G., Samek, W., & Müller, K.-R., 2019. *Understanding Patch-Based Learning of*  
677 *Video Data by Explaining Predictions*, pp. 297–309, Springer International Publishing, Cham.
- 678 Arras, L., Horn, F., Montavon, G., Müller, K.-R., & Samek, W., 2017. "what is relevant in a text document?":  
679 An interpretable machine learning approach, *PLOS ONE*, **12**(8), 1–23.
- 680 Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W., 2015. On pixel-wise explana-  
681 tions for non-linear classifier decisions by layer-wise relevance propagation, *PLOS ONE*, **10**(7), e0130140.
- 682 Balduzzi, D., Frean, M., Leary, L., Lewis, J., Ma, K. W.-D., & McWilliams, B., 2018. The shattered gradients  
683 problem: If resnets are the answer, then what is the question?
- 684 Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-  
685 Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F., 2020. Explainable artificial intelligence  
686 (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, *Information Fusion*, **58**,  
687 82–115.
- 688 Bergen, K. J., Johnson, P. A., de Hoop, M. V., & Beroza, G. C., 2019. Machine learning for data-driven  
689 discovery in solid Earth geoscience, *Science*, **363**(6433).
- 690 Bousquet, O. & Elisseeff, A., 2002. Stability and generalization, *J. Mach. Learn. Res.*, **2**, 499–526.
- 691 Böhle, M., Eitel, F., Weygandt, M., & Ritter, K., 2019. Layer-wise relevance propagation for explaining deep  
692 neural network decisions in mri-based alzheimer's disease classification, *Frontiers in Aging Neuroscience*,  
693 **11**, 194.
- 694 Castelveccchi, D., 2016. Can we open the black box of ai?, *Nature News*, **538**(7623), 20.
- 695 Charles, Z. & Papailiopoulos, D., 2018. Stability and generalization of learning algorithms that converge  
696 to global optima, in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80 of  
697 **Proceedings of Machine Learning Research**, pp. 745–754, PMLR.
- 698 Cua, G. & Heaton, T., 2007. The virtual seismologist (vs) method: A bayesian approach to earthquake early  
699 warning, in *Earthquake early warning systems*, pp. 97–132, Springer.
- 700 Dai, H. & MacBeth, C., 1995. Automatic picking of seismic arrivals in local earthquake data using an artificial  
701 neural network, *Geophysical Journal International*, **120**(3), 758–774.
- 702 Dowla, F. U., Taylor, S. R., & Anderson, R. W., 1990. Seismic discrimination with artificial neural networks:  
703 Preliminary results with regional spectral data, *Bulletin of the Seismological Society of America*, **80**(5), 1346–  
704 1373.
- 705 Elad, M., 2010. *Sparse and redundant representations: from theory to applications in signal and image pro-*  
706 *cessing*, vol. 2, Springer.
- 707 Freedman, D., Pisani, R., & Purves, R., 2007. Statistics (international student edition), *Pisani, R. Purves, 4th*  
708 *edn. WW Norton & Company, New York*.
- 709 Gutenberg, B. & Richter, C. F., 1955. Magnitude and Energy of Earthquakes, *Nature*, **176**(4486), 795–795.
- 710 INGV Seismological Data Centre, 2006. Rete Sismica Nazionale (RSN). Istituto Nazionale di Geofisica e  
711 Vulcanologia (INGV), Italy, Last accessed: March 2020.
- 712 Jozinović, D., Lomax, A., Štajduhar, I., & Michelini, A., 2020. Rapid prediction of earthquake ground shaking

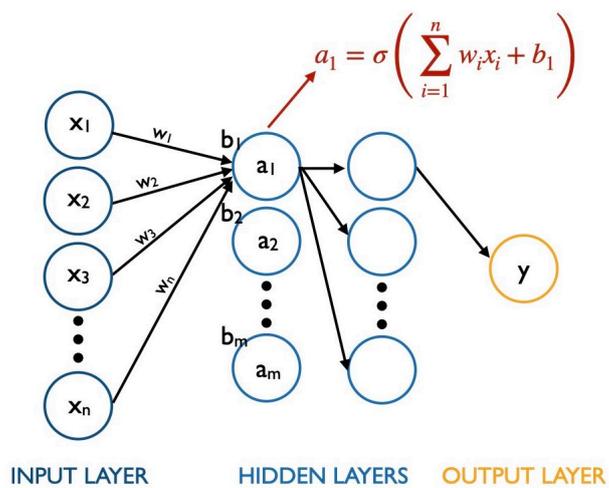
- intensity using raw waveform data and a convolutional neural network, *Geophysical Journal International*, **222**(2), 1379–1389.
- Kong, Q., Allen, R. M., & Schreier, L., 2016. Myshake: Initial observations from a global smartphone seismic network, *Geophysical Research Letters*, **43**(18), 9588–9594.
- Kong, Q., Trugman, D. T., Ross, Z. E., Bianco, M. J., Meade, B. J., & Gerstoft, P., 2019. Machine Learning in Seismology: Turning Data into Insights, *Seismological Research Letters*, **90**(1), 3–14.
- Kong, Q., Wang, R., Walter, W. R., Pyle, M., Koper, K., & Schmandt, B., 2022. Combining deep learning with physics based features in explosion-earthquake discrimination.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems*, vol. 25, Curran Associates, Inc.
- LeCun, Y., Bengio, Y., & Hinton, G., 2015. Deep learning, *Nature*, **521**(7553), 436–444.
- Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S., 2021. Explainable ai: A review of machine learning interpretability methods, *Entropy*, **23**(1).
- Lomax, A., Micheline, A., & Jozinović, D., 2019. An Investigation of Rapid Earthquake Characterization Using Single-Station Waveforms and a Convolutional Neural Network, *Seismological Research Letters*, **90**(2A), 517–529.
- Luong, T., Pham, H., & Manning, C. D., 2015. Effective approaches to attention-based neural machine translation, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Association for Computational Linguistics, Lisbon, Portugal.
- Magrini, F., Jozinović, D., Cammarano, F., Micheline, A., & Boschi, L., 2020. Local earthquakes detection: A benchmark dataset of 3-component seismograms built on a global scale, *Artificial Intelligence in Geosciences*, **1**, 1–10.
- Majstorović, J., Giffard-Roisin, S., & Poli, P., 2021. Designing convolutional neural network pipeline for near-fault earthquake catalog extension using single-station waveforms, *Journal of Geophysical Research: Solid Earth*, **126**(7), e2020JB021566.
- McCulloch, W. S. & Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity, *The bulletin of mathematical biophysics*, **5**(4), 115–133.
- McGovern, A., Lagerquist, R., Gagne, D. J., Jergensen, G. J., Elmore, K. L., Homeyer, C. R., & Smith, T., 2019. Making the black box more transparent: Understanding the physical implications of machine learning, *Bulletin of the American Meteorological Society*, **100**(11), 2175 – 2199.
- Mignan, A. & Broccardo, M., 2020. Neural Network Applications in Earthquake Prediction (1994–2019): Meta-Analytic and Statistical Insights on Their Limitations, *Seismological Research Letters*, **91**(4), 2330–2342.
- Montavon, G., Samek, W., & Müller, K.-R., 2018. Methods for interpreting and understanding deep neural networks, *Digital Signal Processing*, **73**, 1–15.
- Montavon, G., Binder, A., Lapuschkin, S., Samek, W., & Müller, K.-R., 2019. *Layer-Wise Relevance Propagation: An Overview*, pp. 193–209, Springer International Publishing, Cham.

- 750 Montúfar, G., Pascanu, R., Cho, K., & Bengio, Y., 2014. On the number of linear regions of deep neural  
751 networks.
- 752 Mousavi, S. M., Zhu, W., Sheng, Y., & Beroza, G. C., 2019. CRED: A Deep Residual Network of Convolu-  
753 tional and Recurrent Units for Earthquake Signal Detection, *Scientific Reports*, **9**(1), 10267.
- 754 Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y., & Beroza, G. C., 2020. Earthquake transformer—an  
755 attentive deep-learning model for simultaneous earthquake detection and phase picking, *Nature Communica-*  
756 *tions*, **11**(1), 3952.
- 757 Olah, C., Mordvintsev, A., & Schubert, L., 2017. Feature visualization, *Distill*, [https://distill.pub/2017/feature-](https://distill.pub/2017/feature-visualization)  
758 [visualization](https://distill.pub/2017/feature-visualization).
- 759 Peng, Z. & Gomberg, J., 2010. An integrated perspective of the continuum between earthquakes and slow-slip  
760 phenomena, *Nature Geoscience*, **3**(9), 599–607.
- 761 Perol, T., Gharbi, M., & Denolle, M., 2018. Convolutional neural network for earthquake detection and loca-  
762 tion, *Science Advances*, **4**(2), e1700578.
- 763 Ras, G., Xie, N., van Gerven, M., & Doran, D., 2021. Explainable deep learning: A field guide for the  
764 uninitiated.
- 765 Roscher, R., Bohn, B., Duarte, M. F., & Garcke, J., 2020. Explainable machine learning for scientific insights  
766 and discoveries, *IEEE Access*, **8**, 42200–42216.
- 767 Ross, Z. E., Meier, M., Hauksson, E., & Heaton, T. H., 2018. Generalized Seismic Phase Detection with Deep  
768 Learning, *Bulletin of the Seismological Society of America*, **108**(5A), 2894–2901.
- 769 Rouet-Leduc, B., Hulbert, C., Lubbers, N., Barros, K., Humphreys, C. J., & Johnson, P. A., 2017. Machine  
770 learning predicts laboratory earthquakes, *Geophysical Research Letters*, **44**(18), 9276–9282.
- 771 Saad, O. M., Huang, G., Chen, Y., Savvaidis, A., Fomel, S., Pham, N., & Chen, Y., 2021. Scalodeep: A highly  
772 generalized deep learning framework for real-time earthquake detection, *Journal of Geophysical Research:*  
773 *Solid Earth*, **126**(4), e2020JB021473.
- 774 Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J., & Müller, K.-R., 2021. Explaining deep neural  
775 networks and beyond: A review of methods and applications, *Proceedings of the IEEE*, **109**(3), 247–278.
- 776 Simonyan, K., Vedaldi, A., & Zisserman, A., 2014. Deep inside convolutional networks: Visualising image  
777 classification models and saliency maps.
- 778 Toms, B. A., Barnes, E. A., & Ebert-Uphoff, I., 2020. Physically interpretable neural networks for the geo-  
779 sciences: Applications to earth system variability, *Journal of Advances in Modeling Earth Systems*, **12**(9),  
780 e2019MS002002.
- 781 Valoroso, L., Chiaraluce, L., Piccinini, D., Di Stefano, R., Schaff, D., & Waldhauser, F., 2013. Radiography of  
782 a normal fault system by 64,000 high-precision earthquake locations: The 2009 L’Aquila (central Italy) case  
783 study, *Journal of Geophysical Research: Solid Earth*, **118**(3), 1156–1176.
- 784 Woollam, J., Münchmeyer, J., Tilmann, F., Rietbrock, A., Lange, D., Bornstein, T., Diehl, T., Giunchi, C.,  
785 Haslinger, F., Jozinović, D., Michelini, A., Saul, J., & Soto, H., 2022. SeisBench—A Toolbox for Machine  
786 Learning in Seismology, *Seismological Research Letters*, **93**(3), 1695–1709.

- 787 Wu, Y., Lin, Y., Zhou, Z., Bolton, D. C., Liu, J., & Johnson, P., 2019. DeepDetect: A Cascaded Region-Based  
788 Densely Connected Network for Seismic Event Detection, *IEEE Transactions on Geoscience and Remote*  
789 *Sensing*, **57**(1), 62–75.
- 790 Xiao, Z., Wang, J., Liu, C., Li, J., Zhao, L., & Yao, Z., 2021. Siamese earthquake transformer: A pair-input  
791 deep-learning model for earthquake detection and phase picking on a seismic array, *Journal of Geophysical*  
792 *Research: Solid Earth*, **126**(5), e2020JB021444.
- 793 Yang, S., Hu, J., Zhang, H., & Liu, G., 2020. Simultaneous Earthquake Detection on Multiple Stations via a  
794 Convolutional Neural Network, *Seismological Research Letters*, **92**(1), 246–260.
- 795 Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E., 2016. Hierarchical attention networks for docu-  
796 ment classification, in *Proceedings of the 2016 Conference of the North American Chapter of the Association*  
797 *for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, Association for Computa-  
798 tional Linguistics, San Diego, California.
- 799 Zeiler, M. D. & Fergus, R., 2014. Visualizing and understanding convolutional networks, in *Computer Vision*  
800 *– ECCV 2014*, pp. 818–833, Springer International Publishing, Cham.
- 801 Zhu, W. & Beroza, G. C., 2018. PhaseNet: A Deep-Neural-Network-Based Seismic Arrival Time Picking  
802 Method, *Geophysical Journal International*.

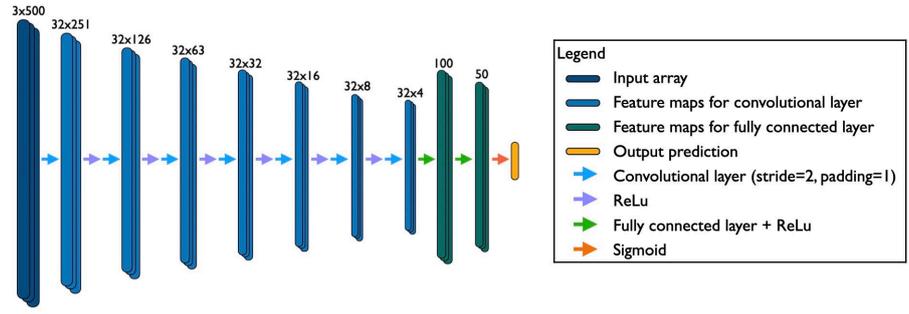
## 803 LIST OF FIGURES

- 804 1 Schematic representation of a fully connected neural network.
- 805 2 A) Schematic representation of the network architecture for the CNN detector model.  
806 The input array size is  $3 \times 500$  corresponding to the three components (E-W, N-S and vertical)  
807 of the 25 s seismic waveforms sampled at 20 Hz. The output prediction is a scalar value  
808 ranging from 0 to 1, where 0 means that the sample belongs to the negative (noise) class and  
809 1 to the positive (earthquake) class. There are nine hidden layers, out of which first seven  
810 are convolutional layers with stride of 2 and padding of 1, and two fully connected layers.  
811 The feature maps from Section 2.5.1 are represented here as light blue blocks. B) Schematic  
812 representation of the backward optimisation method. C) Schematic representation of the  
813 layer-wise relevance propagation method.
- 814 3 A) Earthquake sample from the evaluation dataset with epicentral distance of 5.3 km  
815 and magnitude 0.82. B) Visualisation of the feature maps associated with 32 output channels  
816 (rows) of the seven convolutional layers (L1 - L7) (light blue blocks in 2A) for the earth-  
817 quake sample in A) obtained with the reference CNN detector model A (Section 2.4 and  
818 Table S1). C) Noise sample from the evaluation dataset. D) Same as B) for the noise sample  
819 from C).
- 820 4 Visualisation of the feature maps associated with the first 10 output channels of the  
821 5th convolutional layer of the reference CNN detector model A (Section 2.4 and Table S1)  
822 for 10 earthquake samples (A-J) and 2 noise samples (K and L). First column corresponds  
823 to the Z component of the earthquake samples, while other columns corresponds to the first  
824 10 channel feature maps. The vertical scale is the same per channel for all samples.
- 825 5 A) Optimal inputs obtained for the 10 CNN detector models (Section 2.4) using the  
826 backward optimisation (BO) technique (see Figure 2B) with the input array initialised with  
827 zero values. Columns corresponds to E, N, Z components and the amplitude spectrum of  
828 the associated components. B) Average spectrum of the training positive samples (orange)  
829 compared with the spectrum of 10 optimal inputs (blue).
- 830 6 Optimal inputs for the input array initialised with a real earthquake sample (examples  
831 from Figure 4). Raw Z component (blue) and the optimal input (orange) waveforms are  
832 represented in full time scale, zoomed between 2.5 and 7.5 seconds, the logarithmic ratio of  
833 their amplitude calculated in time and frequency domain.
- 834 7 LRP relevance solutions for the reference CNN detector model A (Section 2.4 and  
835 Table S1) using the LRP- $\beta$  propagation rule with  $\beta = 0$  in Eq. (4) for samples shown in  
836 Figure 4. Raw earthquake E, N, Z components (blue) are compared with the  $\beta = 0$  relevance  
837 value (orange). The time seconds that are associated with the high relevance values are used  
838 by CNN detector model to correctly classify this signal as an earthquake.
- 839 8 Applying the LRP- $\beta = 0$  rule on all samples of the evaluation dataset and calculating  
840 the standard deviation of the LRP over time,  $\sigma_{LRP}$ . A)  $\sigma_{LRP}$  with respect to the epicentral  
841 distance for the positive samples. Yellowish colour indicate higher densities spots. B) His-  
842 togram of  $\sigma_{LRP}$  for the positive (gray) and negative (blue) samples, with the distributions'  
843 average values (dash-dotted line for positive samples and solid line for negative samples).



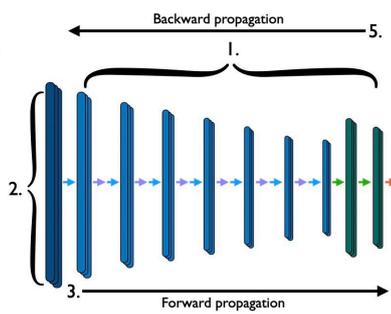
**Figure 1.** Schematic representation of a fully connected neural network.

**A CNN architecture**



**B Backward optimisation**

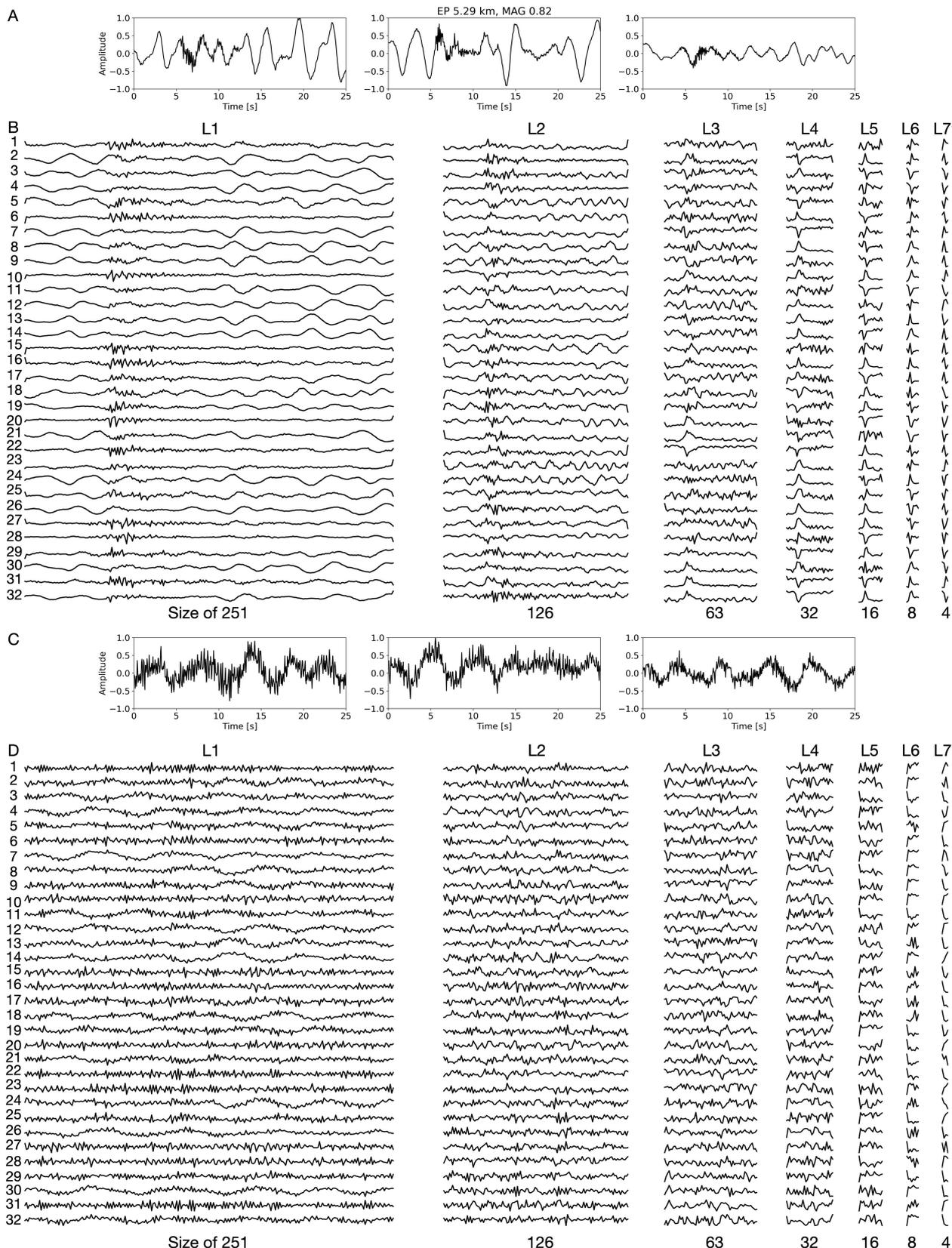
1. Weights and biases of hidden layers are frozen.
2. The input array is initialised (e.g. with zeros).
3. The input array is forward propagated.
4.  $f_{loss}(estimated\ output, desired\ output)$  is calculated by setting the desired output to 1 for investigating the earthquake optimal input.
5.  $f_{loss}$  is backpropagated into the model and the input array is updated.
6. Step 3 to 5 are iterated until  $f_{loss}$  converges.



**C Layer-wise relevance propagation**

1. Weights and biases of hidden layers are frozen.
2. The input array is initialised with a real earthquake sample.
3. The input array is forward propagated and the activations of all layers are stored.
4. The relevance  $R$  of the output is calculated.
5. The relevance  $R$  is backpropagated to the input using the LRP propagation rule.

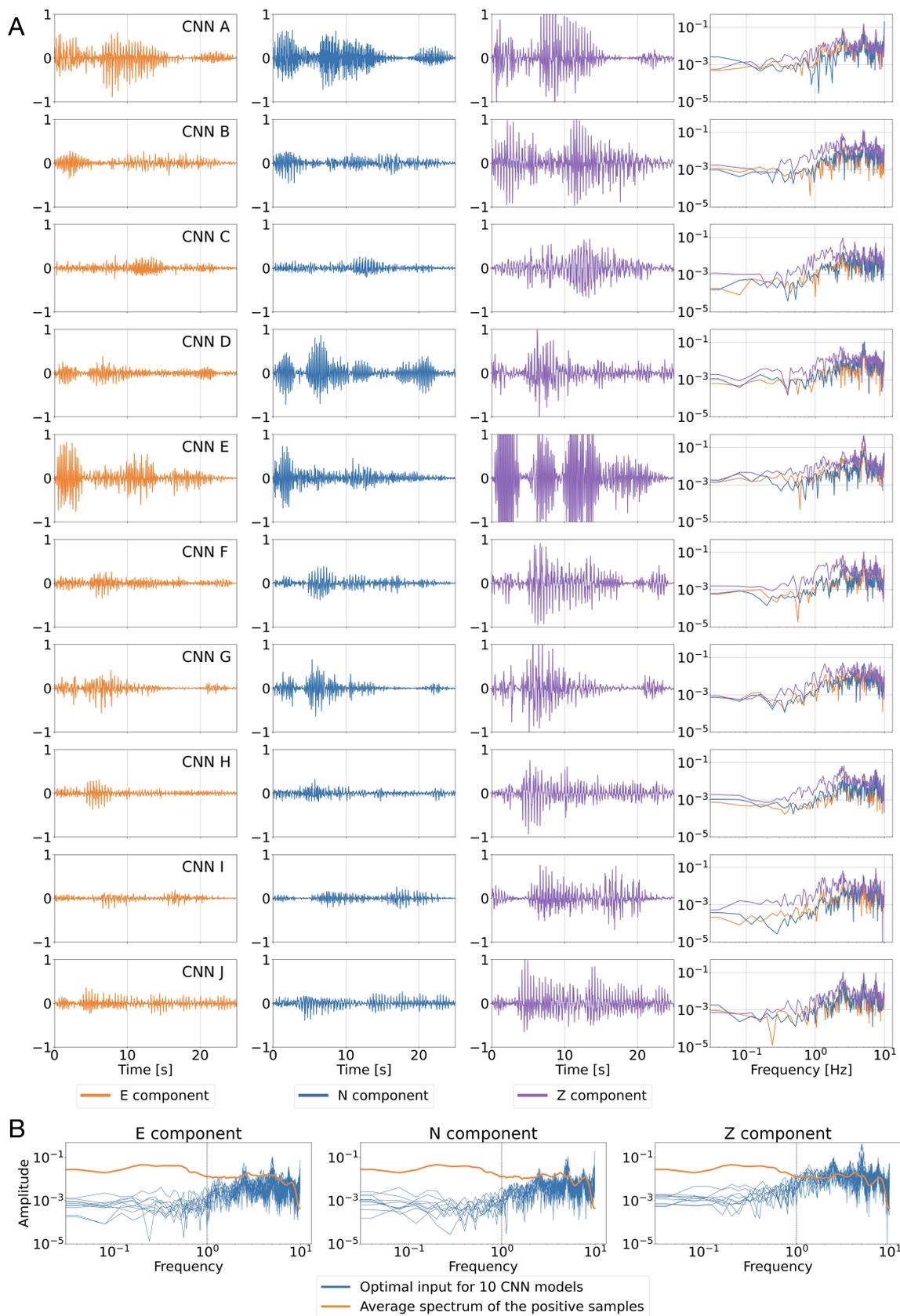
**Figure 2.** A) Schematic representation of the network architecture for the CNN detector model. The input array size is  $3 \times 500$  corresponding to the three components (E-W, N-S and vertical) of the 25 s seismic waveforms sampled at 20 Hz. The output prediction is a scalar value ranging from 0 to 1, where 0 means that the sample belongs to the negative (noise) class and 1 to the positive (earthquake) class. There are nine hidden layers, out of which first seven are convolutional layers with stride of 2 and padding of 1, and two fully connected layers. The feature maps from Section 2.5.1 are represented here as light blue blocks. B) Schematic representation of the backward optimisation method. C) Schematic representation of the layer-wise relevance propagation method.



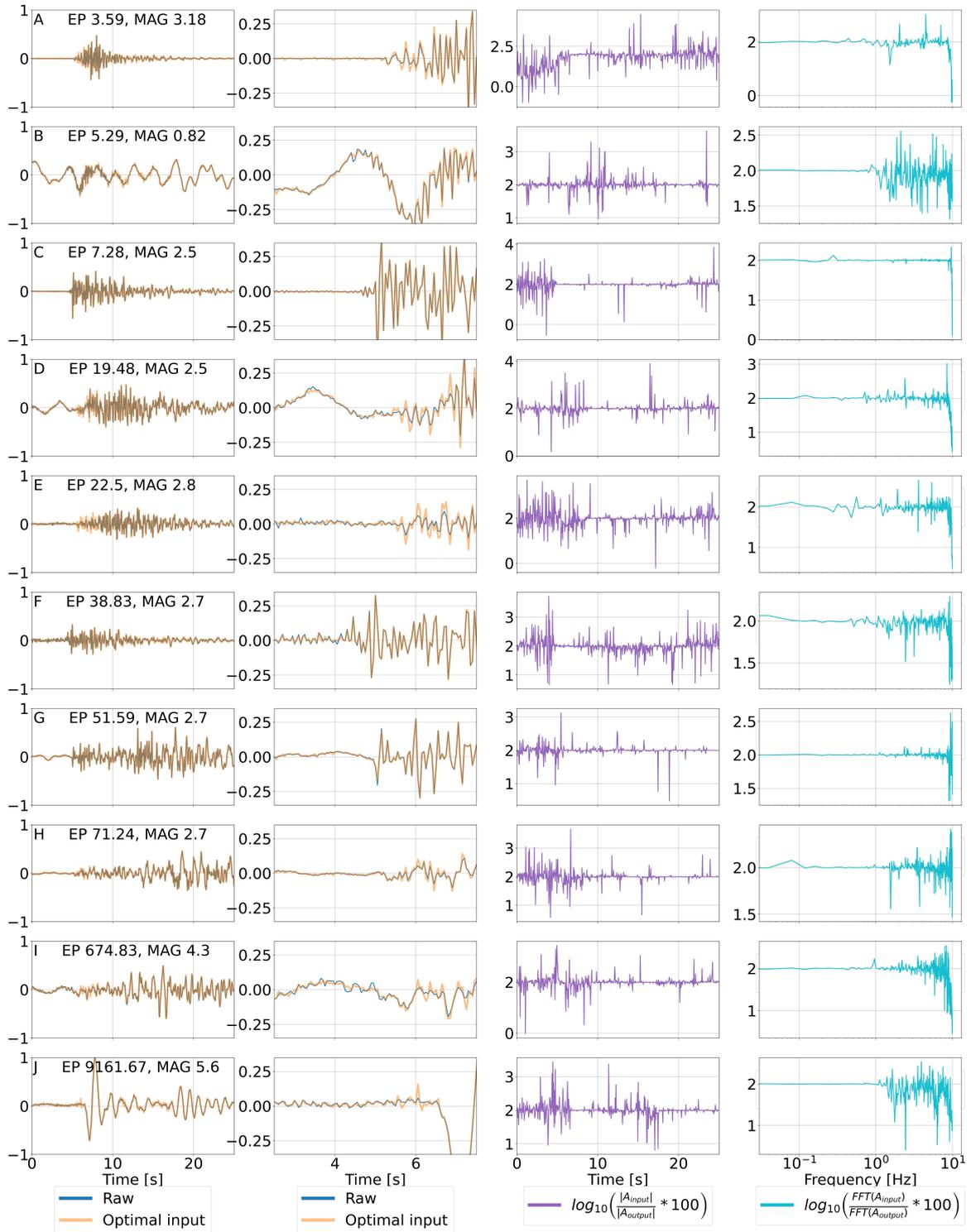
**Figure 3.** A) Earthquake sample from the evaluation dataset with epicentral distance of 5.3 km and magnitude 0.82. B) Visualisation of the feature maps associated with 32 output channels (rows) of the seven convolutional layers (L1 - L7) (light blue blocks in 2A) for the earthquake sample in A) obtained with the reference CNN detector model A (Section 2.4 and Table S1). C) Noise sample from the evaluation dataset. D) Same as B) for the noise sample from C).



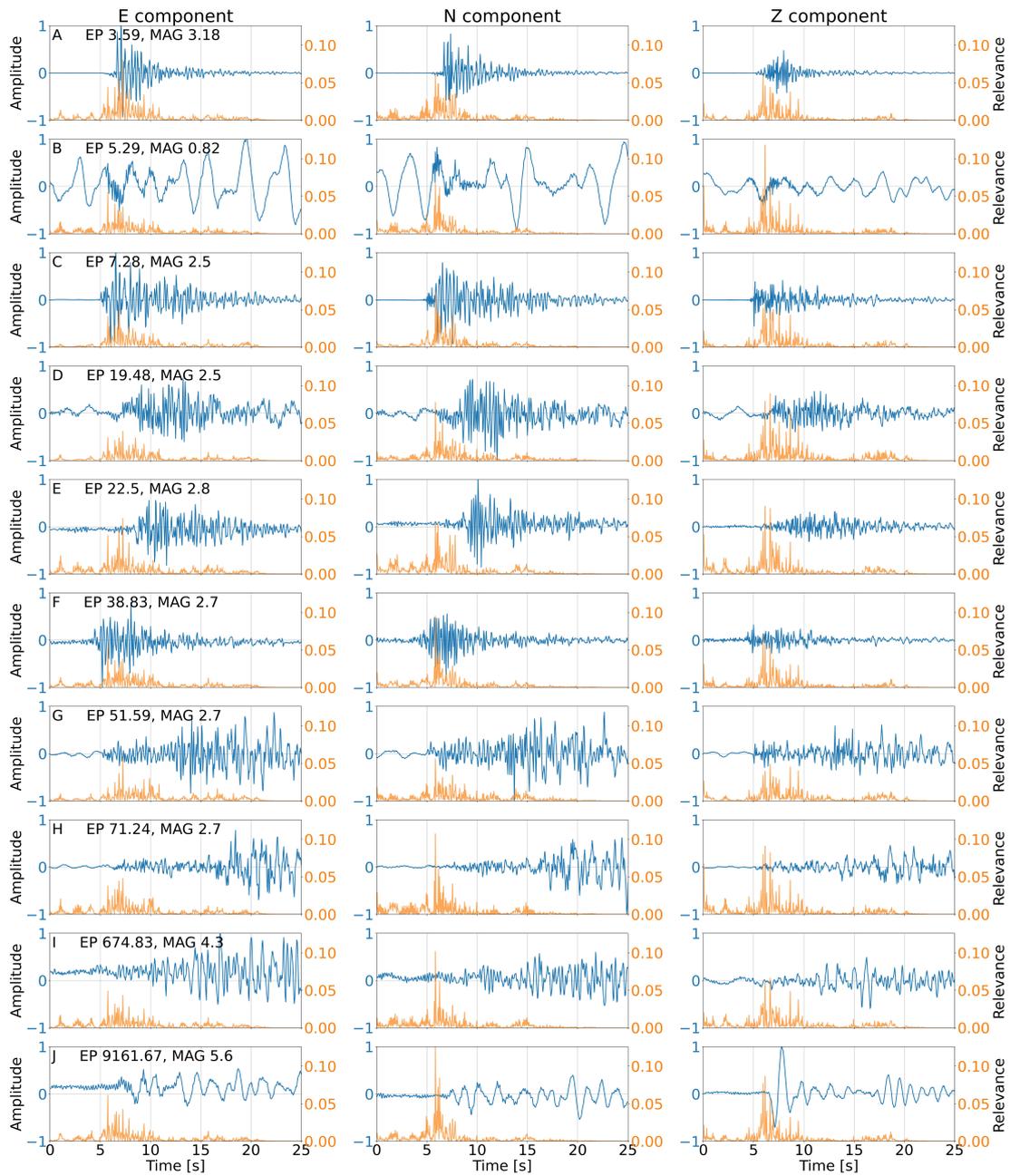
**Figure 4.** Visualisation of the feature maps associated with the first 10 output channels of the 5th convolutional layer of the reference CNN detector model A (Section 2.4 and Table S1) for 10 earthquake samples (A-J) and 2 noise samples (K and L). First column corresponds to the Z component of the earthquake samples, while other columns corresponds to the first 10 channel feature maps. The vertical scale is the same per channel for all samples.



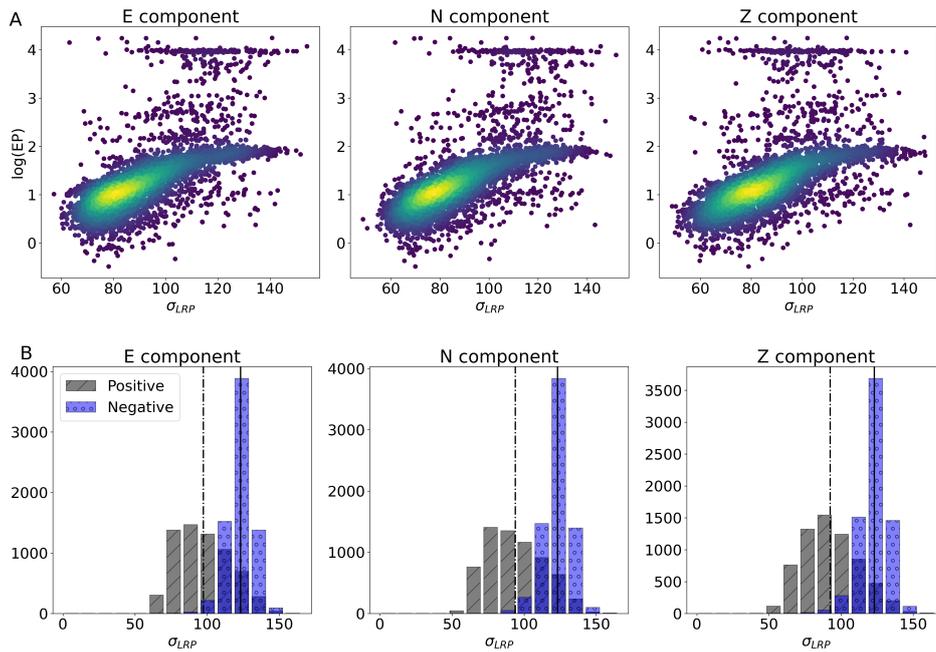
**Figure 5.** A) Optimal inputs obtained for the 10 CNN detector models (Section 2.4) using the backward optimisation (BO) technique (see Figure 2B) with the input array initialised with zero values. Columns corresponds to E, N, Z components and the amplitude spectrum of the associated components. B) Average spectrum of the training positive samples (orange) compared with the spectrum of 10 optimal inputs (blue).



**Figure 6.** Optimal inputs for the input array initialised with a real earthquake sample (examples from Figure 4). Raw Z component (blue) and the optimal input (orange) waveforms are represented in full time scale, zoomed between 2.5 and 7.5 seconds, the logarithmic ratio of their amplitude calculated in time and frequency domain.



**Figure 7.** LRP relevance solutions for the reference CNN detector model A (Section 2.4 and Table S1) using the LRP- $\beta$  propagation rule with  $\beta = 0$  in Eq. (4) for samples shown in Figure 4. Raw earthquake E, N, Z components (blue) are compared with the  $\beta = 0$  relevance value (orange). The time seconds that are associated with the high relevance values are used by CNN detector model to correctly classify this signal as an earthquake.



**Figure 8.** Applying the  $LRP-\beta - 0$  rule on all samples of the evaluation dataset and calculating the standard deviation of the LRP over time,  $\sigma_{LRP}$ . A)  $\sigma_{LRP}$  with respect to the epicentral distance for the positive samples. Yellowish colour indicate higher densities spots. B) Histogram of  $\sigma_{LRP}$  for the positive (gray) and negative (blue) samples, with the distributions' average values (dash-dotted line for positive samples and solid line for negative samples).