

A hybrid residual neural network–Monte Carlo approach to invert surface wave dispersion data

Mattia Aleardi* and Eusebio Stucchi

Earth Sciences Department, University of Pisa, via S. Maria 53, Pisa, 56126, Italy

Received December 2020, revision accepted April 2021

ABSTRACT

Surface-wave inversion is a non-linear and ill-conditioned problem usually solved through deterministic or global optimization approaches. Here, we present an alternative method based on machine learning. Under the assumption of a local one-dimensional model, we train a residual neural network to predict the non-linear mapping between the full dispersion image and the model space, parameterized in terms of shear wave velocity and layer thicknesses. On the one hand, compared to standard convolutional neural networks, the residual network prevents the vanishing gradient problem when training a deep network. On the other hand, the use of the full dispersion image avoids the time-consuming and often ambiguous picking procedure and allows considering higher modes in the inversion framework. One key aspect of any machine learning inversion strategy is the definition of an appropriate training set. In this case, the models forming the training and validation examples are uniformly drawn from previously defined ranges that cover a wide range of possible near-surface layered V_s models. The reflectivity method constitutes the forward modelling operator that converts the model parameters into the observed shot gathers. The inversion also includes a Monte Carlo simulation strategy that propagates onto the model space the uncertainties related to noise in the data and the modelling error introduced by the network approximation. We first discuss synthetic inversions to assess the applicability of the proposed method and to analyse the effect of erroneous model parameterizations. The inversion results are also benchmarked with those provided by a more standard approach in which the particle swarm optimization algorithm inverts the fundamental mode only. Then, we discuss a field data application. Our tests confirm that the residual neural network inversion provides accurate model estimations and reliable uncertainty appraisals. One of the main benefits of the proposed approach is that once the network is trained it provides the near-surface shear wave velocity profile in near real-time.

Key words: Inversion, Surface wave, Near-surface.

INTRODUCTION

Rayleigh wave measurements are highly sensitive to S-wave velocity (V_s) and hence the multichannel analysis of surface waves (MASW) (Park *et al.*, 1999; Socco and Strobbia, 2004;

Foti *et al.*, 2018) is widely employed for near-surface investigations (Xia *et al.*, 2003; Lai, 2005; Strobbia *et al.*, 2010; Serdyukov *et al.*, 2017; Rahman *et al.*, 2018; Cercato *et al.*, 2020) and also for statics corrections in deep seismic exploration (Socco *et al.*, 2010). Well-established inversion methods rely on dispersion curve inversion under the assumption

*Email: mattia.aleardi@unipi.it

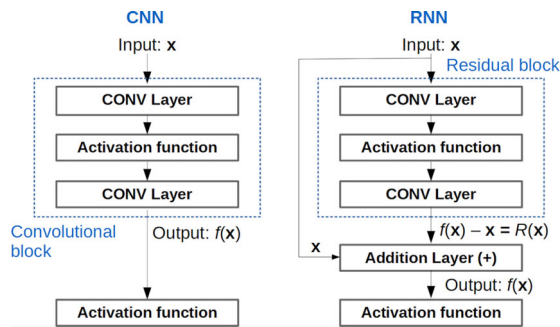


Figure 1 Comparison between a convolutional block in CNN (left) and a residual block in RNN (right).

of a one-dimensional (1D) subsurface structure (Dal Moro *et al.*, 2007; Socco and Boiero, 2008; Maraschini and Foti, 2010; Cercato, 2011; Di Giulio *et al.*, 2019). The Rayleigh wave phase velocity primarily depends on shear wave velocities (V_s), and layer thicknesses (b), while compressional-wave velocities (V_p) and density exert minor roles in determining the observed dispersion pattern (Cox and Teague, 2016). For this reason, the subsurface is usually subdivided into a reasonable number of layers, and the inverse problem is often parameterized in terms of V_s and thickness of the layers, while the V_p and density values are either assumed to be known or related to the V_s values by a simplified relation (e.g., a constant Poisson's ratio in the investigated depth interval).

The extraction of the dispersion curve is a very delicate procedure, especially when higher modes are more energetic than the fundamental mode (Xia *et al.*, 2003; Luo *et al.*, 2009; Boiero *et al.*, 2011). For this reason, common inversion approaches reduce the human effort needed for the picking procedure and avoid modal misinterpretation by limiting the attention to the fundamental mode only, although it is known that higher modes are essential to better constrain the solution in case of shear wave velocity inversions and/or high stiffness contrasts within the soil column (Feng *et al.*, 2005; Luo *et al.*, 2009; Farrugia *et al.*, 2016; Sajeve and Menanno, 2017)

The MASW problem is non-linear and ill-posed. The non-linearity is commonly tackled by adopting Monte Carlo (MC) optimization algorithms to locate the global minimum of the error function to be minimized. These methods ensure a wide exploration of the parameters space, thus avoiding entrapment in local minima. The ill-posedness is mitigated by including appropriate model constraints into the error function. To this respect, also the estimation of the uncertainty affecting the recovered solution is of primary importance and for this reason a Bayesian framework and Markov Chain Monte Carlo methods (MCMC; Sambridge and Mosegaard, 2002;

Tarantola, 2005) can be employed to cast this inverse problem into a solid probabilistic framework (Aleardi *et al.*, 2020) for accurate posterior probability density (PPD) estimations. However, these methods are usually computationally expensive due to the considerable number of samples needed to attain stable PPDs. For this reason, the MCMC approach becomes computationally impractical for inverting large datasets.

Over the past years, machine learning algorithms (Monajemi *et al.*, 2016; Goodfellow *et al.*, 2016) have been increasingly applied to solve geophysical inverse problems. In this context, neural networks are very useful when the forward relation is known, but the inverse mapping is either expensive to compute analytically or to approximate numerically. Therefore, the network is trained to predict the mapping between the data domain and the model parameters. Compared to conventional fully connected networks, convolutional neural networks (CNNs) are regularized networks with two advantages: sparse connectivity and sharing weights among convolutional layers that reduce the computational cost, while improving the generalization ability (LeCun *et al.*, 2015; Schmidhuber, 2015; Krizhevsky *et al.*, 2017). The learning stage is an optimization process that minimizes a difference criterion between predicted and desired output, and a sufficiently large training set is usually needed to adjust the internal network parameters. However, once the network is trained it provides an output from the input in real-time. Some applications of CNN in geophysics can be found in Lewis and Vigh (2017), Araya-Polo *et al.* (2018), Richardson (2018), Waldeland *et al.* (2018), Wang *et al.* (2019), Wu and McMechan (2019), Puzyrev (2019), Park and Sacchi (2020), Aleardi (2020), Moghadas (2020), Aleardi and Salusti (2021).

The number of convolutional layers in a CNN plays a crucial role as a deeper network can potentially better approximate complex non-linear functions. It is generally believed that the deeper the network, the higher is the accuracy (match between desired and actual network outputs). However, one common issue of CNN is related to the vanishing gradient problem that occurs when the network is unable to propagate gradient information from deep layers back to the shallow ones. This usually occurs when the network gets deeper (i.e., formed of many convolutional layers) and is usually referred to as the degradation problem: the accuracy gets saturated for a given number of layers and then starts degrading rapidly if additional layers are added. To solve this issue, Glorot and Bengio (2010) proposed a residual neural network (RNN) in which skip connections (also called shortcuts) are used to

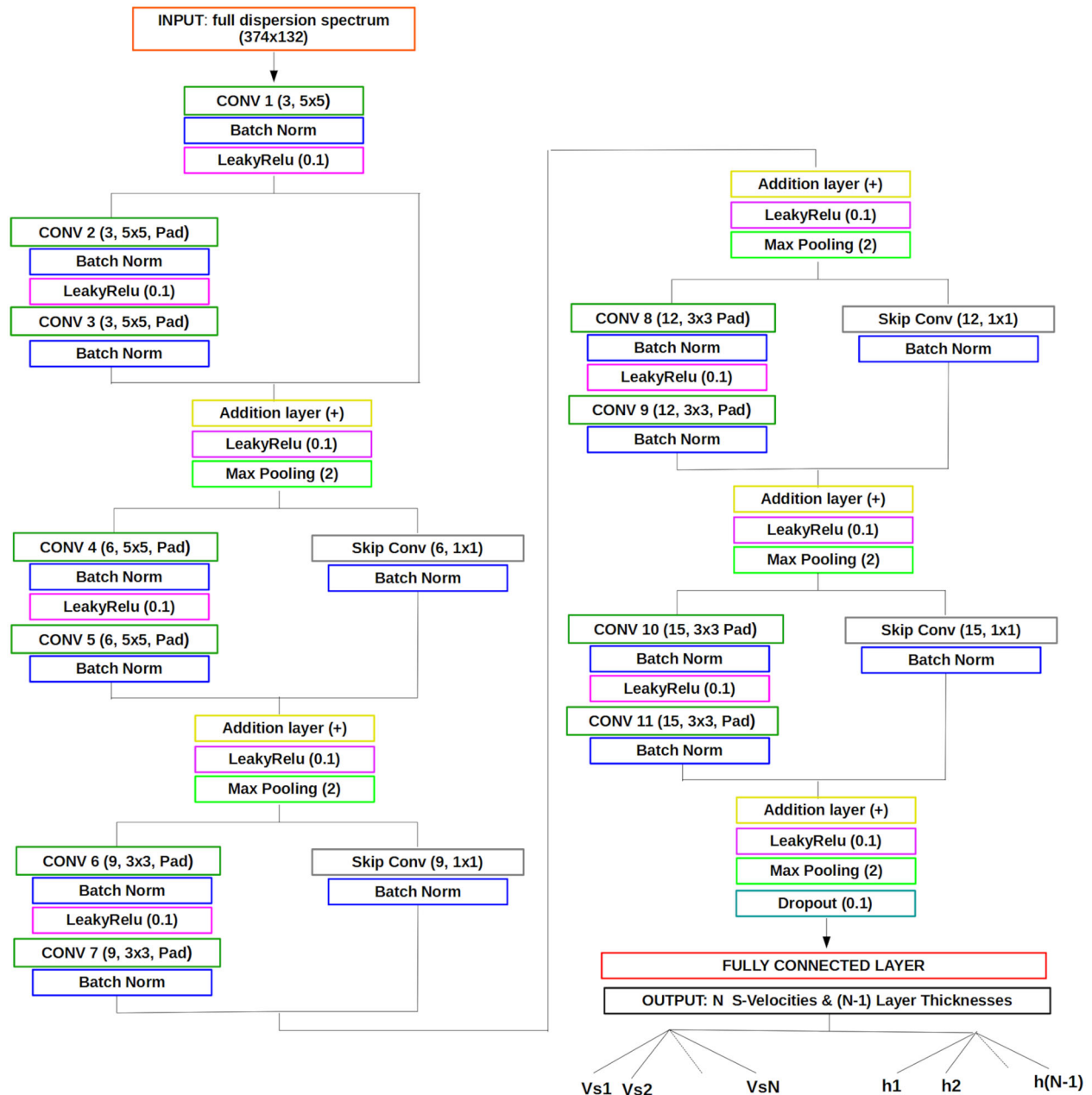


Figure 2 Schematic representation of the adopted RNN architecture annotated with key parameters. For example, in the second convolutional layer “CONV 2” the term within bracket ($3, 5 \times 5, \text{Pad}$) means that we use three convolutional filters with size 5×5 and that zero padding is applied. The dimension of the input (374×132) refers to the synthetic inversion. (See the text for additional explanations). The stride is 1 in all the convolutional layers.

connect shallow and deep layers so that the information is directly passed through the network. This means that the result of a layer is added directly to the corresponding output of a deeper layer. See He *et al.* (2016) for a rigorous mathematical explanation of the vanishing gradient problem and of RNN in terms of function classes.

In the context of MASW inversion, although a fully connected network with a limited number of hidden layers can provide quite accurate inverse mapping when only the fundamental mode of Rayleigh waves is considered (Yablokov *et al.*, 2020), we envisage that a CNN with a significant number of layers is actually needed when the full dispersion image is

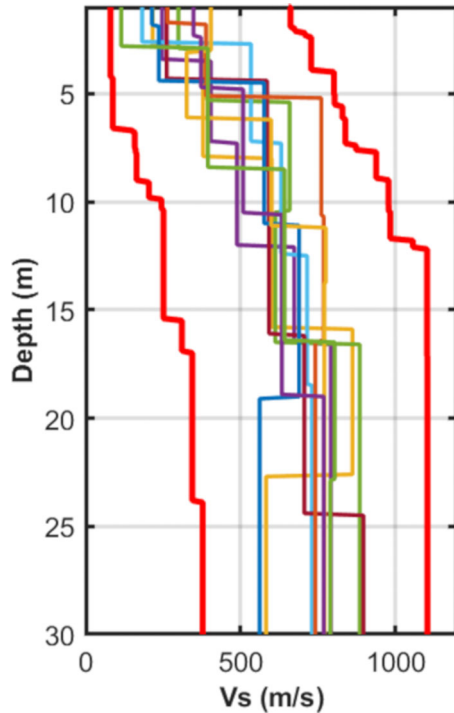


Figure 3 The thick red lines represent the admissible V_s range for generating the training and validation sets. The coloured lines show 10 out of the 20,000 generated models.

considered. For this reason, to avoid the degradation problem, we adopt an RNN approach to MASW inversion. The use of the full dispersion spectrum avoids the time and human effort needed for modal identification and manual picking. For a previously defined number of layers, the unknowns in the inversion are the V_s value and thickness of the layers, while the V_p and density are not inverted due to their limited influence on the Rayleigh waves dispersion pattern.

The RNN–MASW inversion includes three steps: (1) *Generation phase*: define an ensemble of 1D V_s models and compute the associated phase velocity spectra. The V_s models and associated spectra constitute the network output and

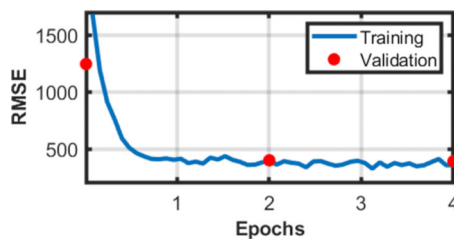


Figure 4 Evolution of the RMSE error on the training and validation sets.

input responses, respectively. (2) *Network design*: define a neural network architecture to approximate the non-linear mapping between the dispersion image and the V_s model. (3) *Training phase*: train the network by minimizing the difference between the predicted and desired output. The training ensemble is built by drawing uniformly at random V_s models within a previously selected velocity range. The defined velocity models should cover a wide range of possible near subsurface scenarios. The reflectivity method (Kennett, 1983) computes the seismic gathers from the velocity profiles under the assumption of a 1D structure. In the forward modelling phase, the density values are kept constant, while the V_p is uniquely determined from the V_s by assuming a fixed Poisson ratio value. The RNN inversion is combined with a subsequent MC simulation approach to estimate the uncertainties affecting the retrieved solution. To this end, we propagate onto the model space the uncertainties related to noise contamination and also the so-called modelling error introduced by the RNN approximation. As such, the network will provide an approximated inverse mapping between the data and the model space and the approximation error must be propagated into the estimated velocity profile.

We first assess the applicability of the proposed approach by inverting velocity spectra derived from synthetic seismic data. In this section, we compare the RNN outcomes with those achieved by a particle swarm optimization (PSO) and we also investigate how erroneous assumptions on the number of layers affect the recovered model. Then, the method is applied to field data. Some studies proposed machine learning approaches for extracting dispersion curves (Dai *et al.*, 2020) or for surface waves inversion (Meier *et al.*, 2007; Cheng *et al.*, 2019; Hu *et al.*, 2020), especially at a seismological scale. Other recent studies used RNN to solve geophysical problems (Wang *et al.*, 2020; Yang *et al.*, 2020; Othman *et al.*, 2021). However, to the best of our knowledge, this is the first time that an RNN and a MC approach are combined to solve the MASW problem in which the near-surface V_s profile and the associated uncertainties are jointly estimated from the full dispersion data.

METHODS

The selected residual neural network architecture

Both convolutional and RNNs can be expressed as a function F that computes the output O from the input I through the internal parameters P :

$$O = F(P, I). \quad (1)$$

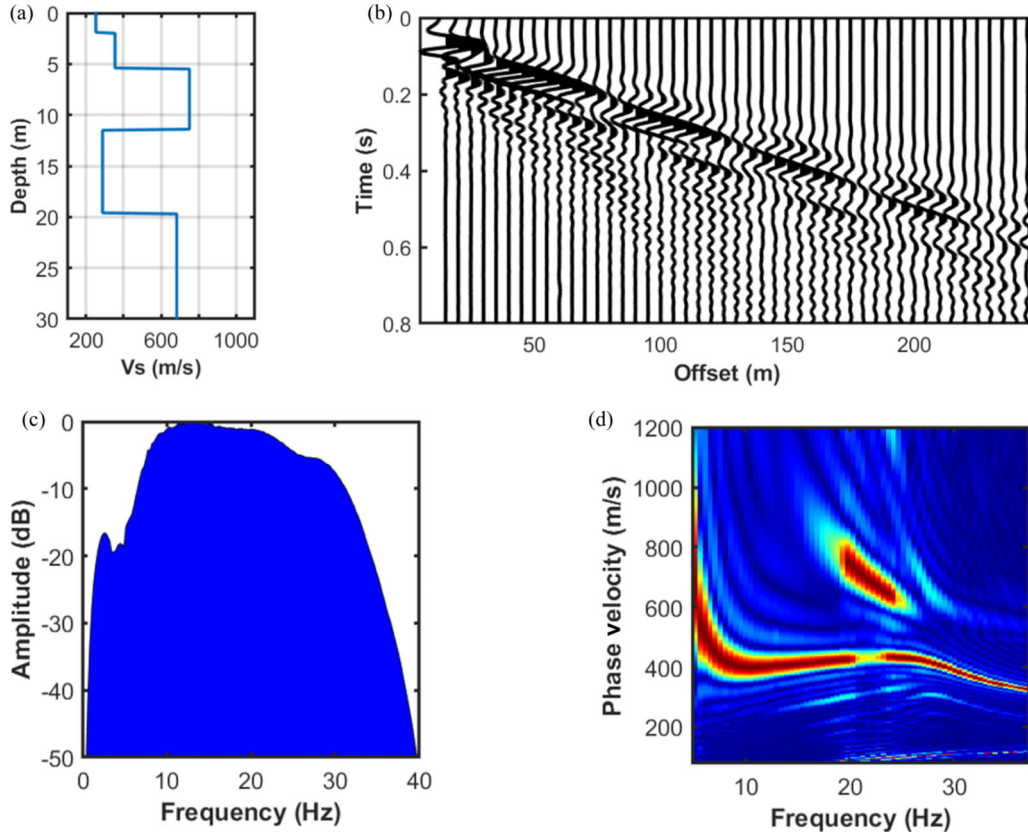


Figure 5 Example of generated Vs model (a), associated shot gather (b), Fourier amplitude spectrum (c) and phase velocity spectrum (d).

Both networks use convolutional filters and fully connected layers to extract features from 1-D, 2-D, or 3-D inputs. The filters are called ‘kernels’ and the extracted features are called

the features maps. In a CNN, the feature mapping from one arbitrary convolutional layer to the next can be generically written as follows (Sun *et al.*, 2020):

$$O_j^p = f \left(B_j + \sum_{i=1}^I O_i^{p-1} * W_j \right), \quad j = 1, 2, \dots, J, \quad (2)$$

where $*$ represents the convolution process, I denotes the number of the feature maps in the $(p - 1)$ th layer, and J is the number of feature maps in the p th layer that corresponds to the number of filters considered in that layer; B_j is a matrix with the same size as O_j^p expressing the biases for the p th layer; O_j^p represents the j th feature map in the p th layer, O_i^{p-1} is the i th feature map in the $(p - 1)$ th layer, and W_j represents the j th filter of the p th layer that is the weight matrix connecting O_j^p with O_i^{p-1} ; $f()$ is the activation function that includes non-linearity in the mapping process. Therefore, in a traditional CNN, each layer feeds into the next layer (Fig. 1, left panel). Differently, RNN uses shortcuts and skip connections to connect shallow and deep layers directly (Fig. 1, right panel), thus preventing the information loss that occurs when backpropagating the gradient. This means that the result of a shallow

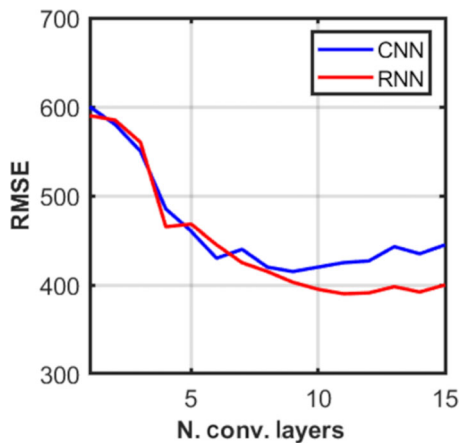


Figure 6 Evolution of the final RMSE error computed on the training set for the CNN and RNN when the number of convolutional layers is varied.

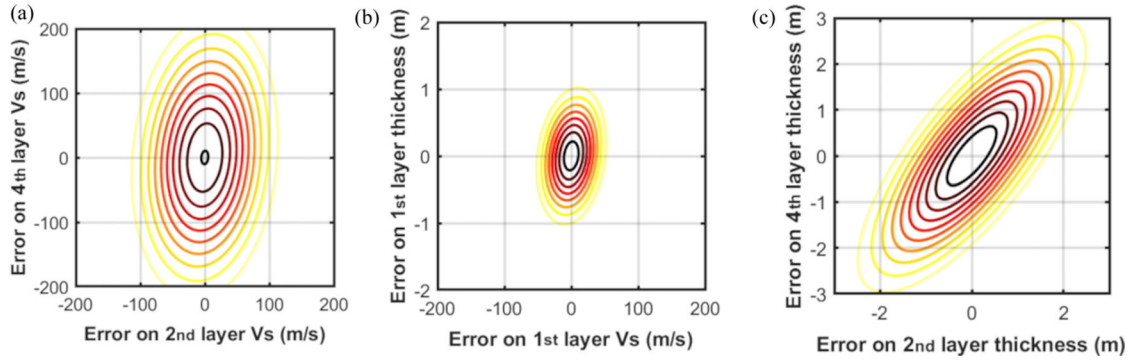


Figure 7 The multivariate Gaussian modelling error projected onto some model space dimensions. (a) Projection onto the directions defined by the Vs of the second and fourth layer. (b) Projection onto the model space directions associated with the Vs and thickness of the first layer. (c) Projection onto the directions defined by the thickness of the second and fourth layer. In (a)–(c) light and dark colours code low and high probability values, respectively.

layer is added directly to the corresponding output of a deeper layer so that the information is passed directly through the network as an identity function. The idea of this approach is to assume that the residual mapping ($R(\mathbf{x})$ in Fig. 1, right panel) is easier to optimize than the mapping $f(\mathbf{x})$ associated with the traditional CNN (Fig. 1, left panel). As an added advantage, an identity mapping for a given layer (i.e., a function that provides an output equal to the input) can be simply obtained by putting $R(\mathbf{x})$ to zero. Many different types of residual blocks have been proposed, but in this work we use the original configuration schematically depicted in Fig. 1.

The final selected RNN architecture is represented in Fig. 2. Note that for some residual blocks we use skip connections to adjust features dimensions before addition layers. In some cases, we apply a zero-padding operation to preserve the dimensions after convolution. We use the leaky ReLU activation function (Hahnloser *et al.*, 2000) with a leakage value of 0.1. After a convolutional layer, a sub-sampling technique is often used to reduce the dimensions of the feature maps. Here we employ the max-pooling operation with a size of 2.

This is the most common sub-sampling strategy in which a sliding window with predefined dimensions extracts the maximum value from a rectangular neighbourhood of pixels forming the features maps. After the last convolutional layer, the feature maps are fed into the fully connected layer, and we use the dropout regularization to avoid overfitting. Dropout is a methodology where randomly selected neurons are ignored during the training phase (i.e., in our case the 10%). The internal network parameters \mathbf{P} (e.g., bias and filter values within the convolutional layers) are first initialized according to the He method (He *et al.*, 2015) and then updated during the learning procedure that minimizes the root-mean-square-error (RMSE) between the desired and the computed output. The updating process is driven by the RMSprop optimizer (i.e., an unpublished, adaptive learning rate method) running for four epochs. The updating of the internal network parameters can be generically written as follows:

$$\mathbf{P}_i = \mathbf{P}_{i-1} - \gamma \frac{\partial \varepsilon}{\partial \mathbf{P}_{i-1}}, \quad (3)$$

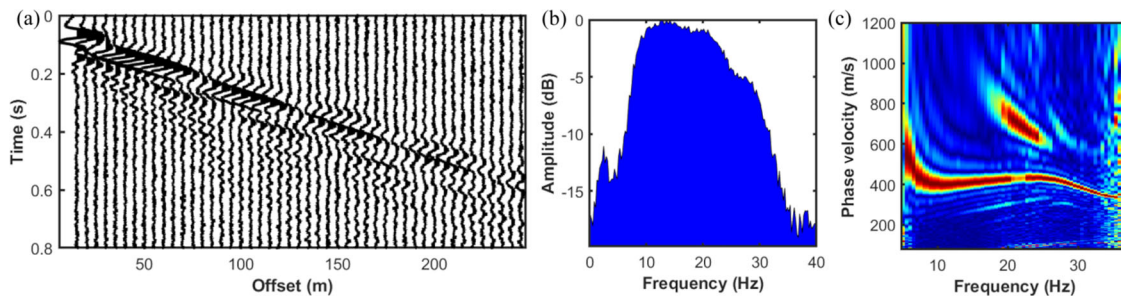


Figure 8 Example of noise-contaminated shot gather (a) and the associated Fourier amplitude and phase velocity spectra (b, and c, respectively). Compare with Figure 5 that shows the same shot and spectra before noise contamination.

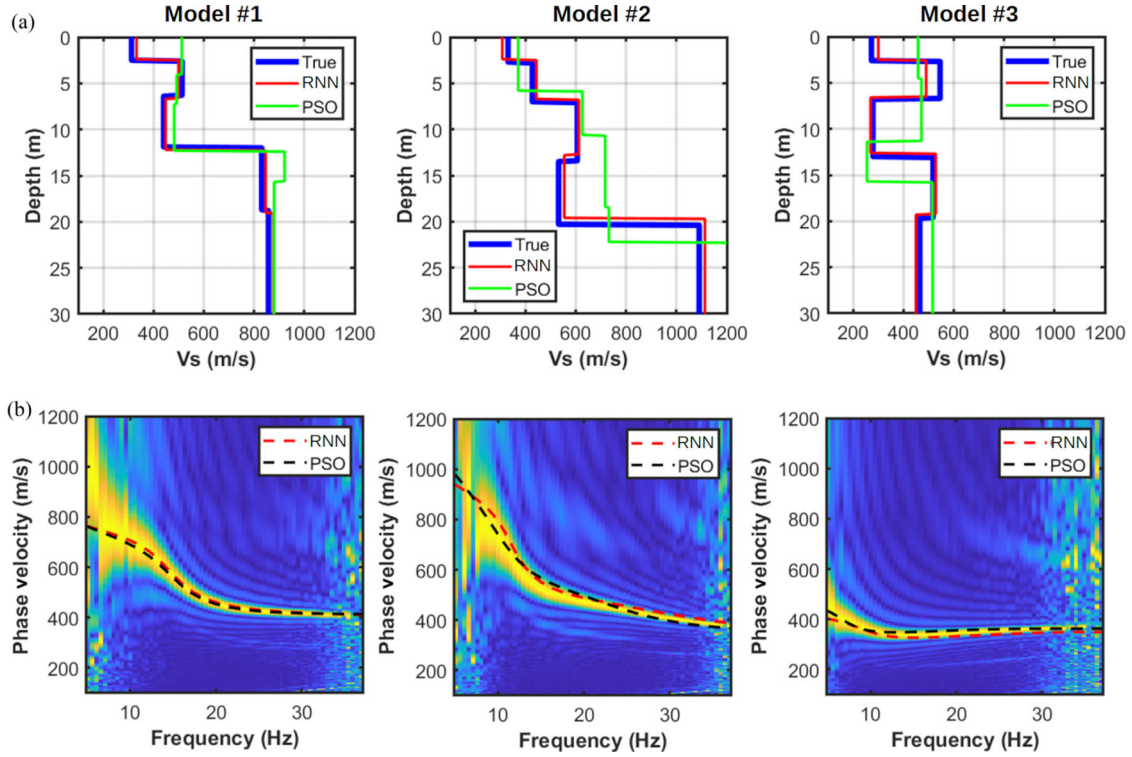


Figure 9 Comparison of the RNN and PSO results on three different subsurface models. (a) Comparison of the true and predicted V_s profiles. (b) Comparison of the observed phase velocity spectrum and the fundamental mode computed on the RNN and PSO results.

where i represents the iteration number, ε is the loss function value and γ is the so-called learning rate. Here we set the initial learning rate to 0.0004, and this value is halved every epoch. We also use batch normalization within each convolutional layer as a regularization operator (Santurkar *et al.*, 2018), and we employ a batch size of 64.

As it can be noticed from Fig. 2, several hyperparameters define the RNN architecture: number of convolutional layers and filters, activation function, stride and size of the convolutional filters, strategy to initialize the weights, number of epochs, learning rate value and a method that minimizes the error function. Personal preferences and experiences usually determine the hyperparameter setting. Here we employ a trial-and-error procedure in which different hyperparameters are modified and the final net architecture has been determined according to its accuracy on the validation set.

Data, model parameters and the Monte Carlo uncertainty propagation

The network input (i.e., the observed data \mathbf{d} of the RNN inversion) is an $L \times P$ matrix representing the full phase-velocity spectrum, with L and P denoting the number of samples along

the velocity and frequency axes, respectively. For example, the input dimension shown in Fig. 2 refers to the synthetic case with $L = 374$ and $P = 132$. The output \mathbf{m} is a vector containing the V_s and thickness (b) values for a previously defined number of layers (N):

$$\mathbf{m} = [V_{s1}, V_{s2}, \dots, V_{sN}, b_1, b_2, \dots, b_{N-1}]. \quad (4)$$

For what concerns the uncertainty assessment, we take into account both the noise affecting the data and the approximation error introduced by the network. To this end, we adopt a MC approach: Let \mathbf{M} denote the models forming the training set, whereas \mathbf{N} represents the associated RNN predictions. According to Hansen and Cordua (2017), a sample of the modelling error can be computed as the difference between desired and predicted models ($\mathbf{E} = \mathbf{M} - \mathbf{N}$). Under a Gaussian assumption, the modelling error distribution can be written as $\mathcal{N}(0, \mathbf{C}_e)$, where \mathbf{C}_e represents the covariance of \mathbf{E} and \mathcal{N} indicates the Gaussian distribution. We also assume that Gaussian noise $\mathcal{N}(0, \mathbf{C}_n)$ contaminates the seismic gather. The following iterative MC approach is used for the uncertainty quantification:

1. Use the trained network to compute the predicted model \mathbf{m}_b from the observed dispersion image \mathbf{d} ;

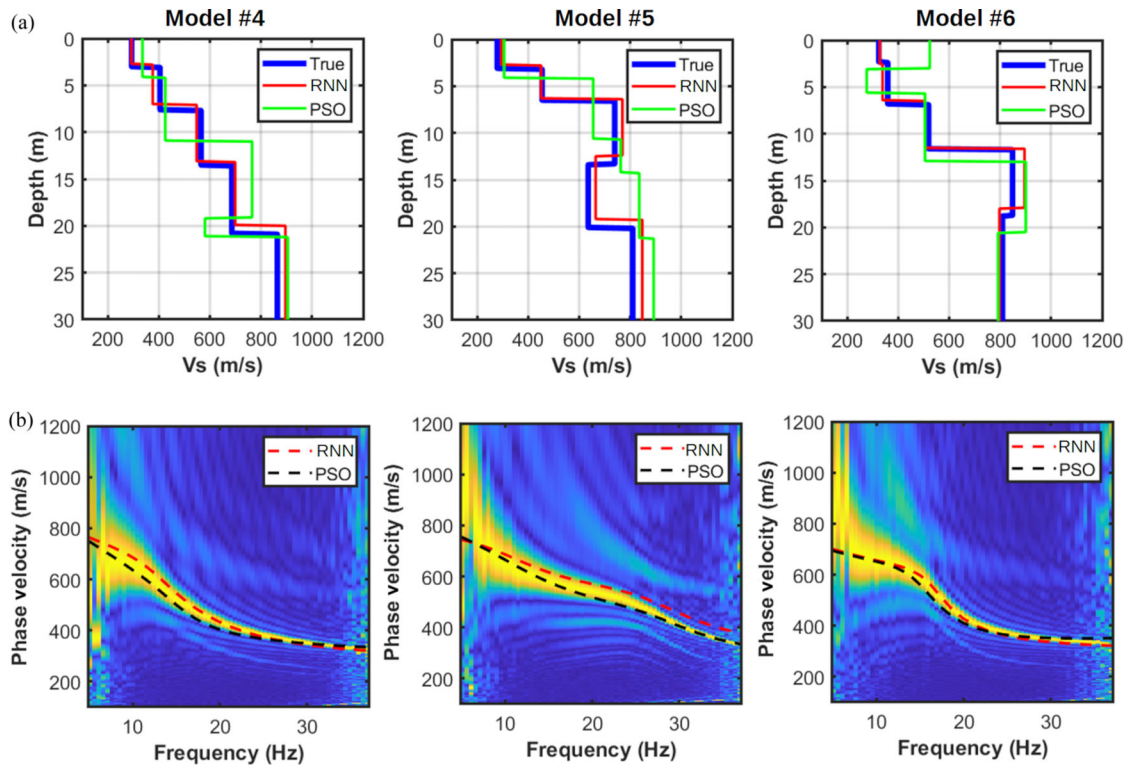


Figure 10 Comparison of the RNN and PSO results on three different subsurface models. (a) Comparison of the true and predicted Vs profiles. (b) Comparison of the observed phase velocity spectrum and the fundamental mode computed on the RNN and PSO results.

2. Run a reflectivity modelling to compute the noise-free seismic data \mathbf{s}_b associated with \mathbf{m}_b ;
3. Draw \mathbf{n} from $\mathcal{N}(0, \mathbf{C}_n)$ and compute $\mathbf{s}_n = \mathbf{s}_b + \mathbf{n}$;
4. Compute the phase velocity spectrum from \mathbf{s}_n , thus obtaining \mathbf{d}_n ;
5. Use the trained RNN to compute the predicted model \mathbf{m}_n from \mathbf{d}_n ;

6. Draw \mathbf{e} from $\mathcal{N}(0, \mathbf{C}_e)$ and compute $\mathbf{m}_e = \mathbf{m}_n + \mathbf{e}$;
7. Store \mathbf{m}_e and repeat from 3) to 7) for q times.

Each vector \mathbf{m}_e can be considered a possible solution in agreement with the observed data and the assumed distributions for noise and modelling errors. An approximated PPD can be numerically derived from the ensemble of q MC

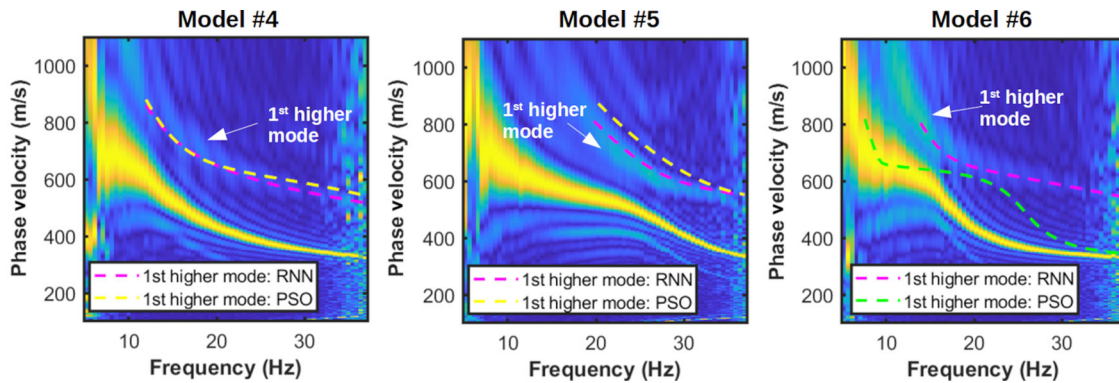


Figure 11 Comparison of phase velocity spectrum associated with models 4, 5, and 6 (shown in Figure 10) and the fundamental and first higher modes computed on the RNN and PSO solutions. The white arrows point towards the first higher mode identified in the dispersion spectrum.

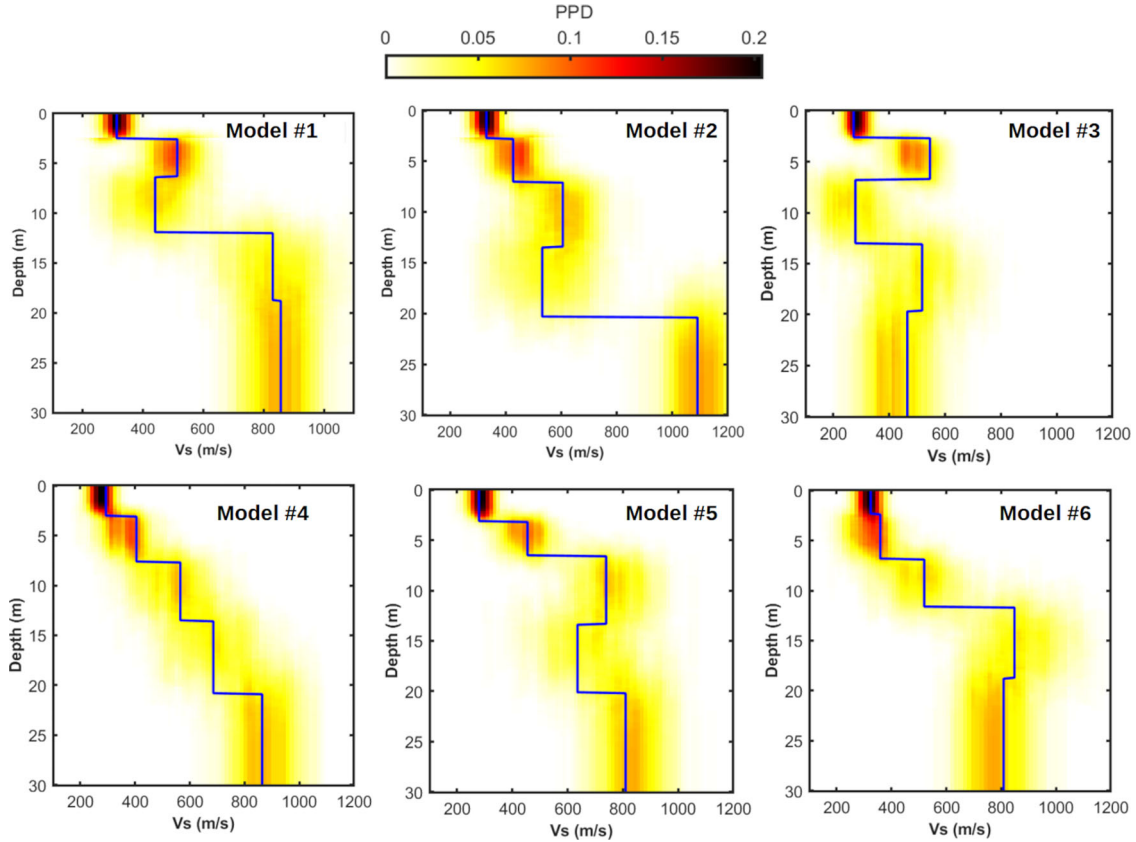


Figure 12 PPD estimated by the MC simulation approach for the six models previously shown in Figures 9 and 10. The blue lines depict the true V_s profile.

simulations. For simplicity, we assume Gaussian-distributed noise and modelling errors, but the MC approach can handle whatever parametric or non-parametric distribution. Note that the MC approach is extremely fast because the network instantaneously predicts a model from the input data.

SYNTHETIC INVERSION

For the synthetic experiments, we use a training set of 19,000 examples including the phase velocity spectra associated with V_s and h values. The validation set is composed of 1000 examples: This results in a 95/5 split. The subsurface V_s models include five layers ($N = 5$ in Fig. 2) and have been generated with uniform probability within a previously defined range (Fig. 3) that covers a wide range of possible subsurface scenarios. We assume a constant, depth-independent density value equal to 1.8 g/cm^3 , whereas the V_p in each layer is related to the V_s by a Poisson's ratio equal to 0.38. As previously mentioned, the density and V_p values are not considered in the inversion

procedure. The phase velocity spectra have been computed on shot gathers generated via the reflectivity method employing a zero-phase source wavelet with a peak frequency of 15 Hz. We assume an off-end acquisition layout with 48 equally spaced receivers with minimum and maximum offsets equal to 10 and 245 m, respectively. The network has also been trained on different numbers of models in the training set (e.g., 5000, 10,000, 20,000, 30,000), but we found that 20,000 constitutes the best compromise between the network accuracy and the computational cost of the generation and training phases. The generation of the 20,000 examples takes 1.5 hours, using a parallel hybrid Matlab-Fortran code running on a two deca-core Intel E5-2630 at 2.2 GHz (128 Gb RAM). For the network training, we use the Matlab implementation of RNN running on a common notebook equipped with a quad-core Intel Core i-7 7700HQ CPU@2.80 GHz, with 16 Gb RAM. The computing time for training the final network configuration is 6 minutes, approximately. Note that the training converges to the final loss value after less than three epochs (Fig. 4). Similar errors on the validation and training sets prove

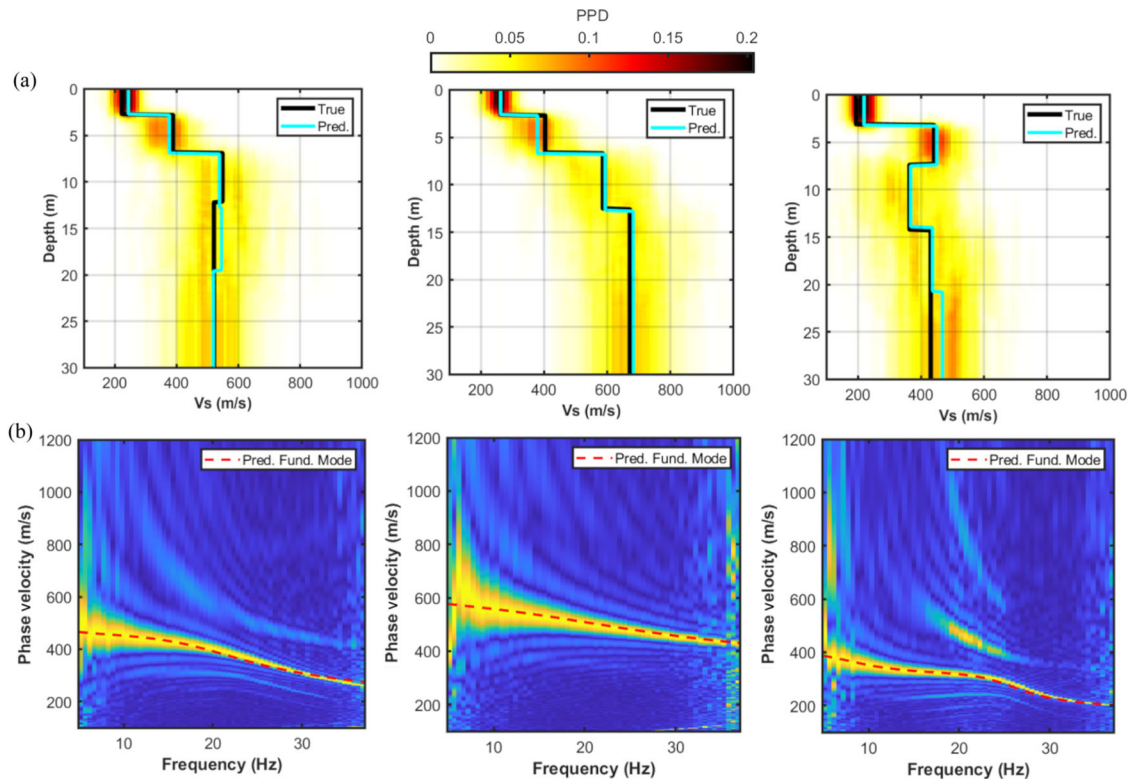


Figure 13 RNN results on four-layer models. (a) Comparison between the true V_s profile, the RNN predictions and the MC PPD estimations. (b) Comparison between the phase velocity spectra associated with the true models shown in (a) and the fundamental mode computed on the RNN results.

that overfitting has been avoided. As an example, Fig. 5 shows a V_s model and associated shot gather, Fourier spectrum and dispersion pattern.

Before describing in detail the RNN results, we show in Fig. 6 a direct comparison between the RMSE values computed on the training set for RNN and CNN when the number of layers is varied. The first 11 RNN layers maintain the hyperparameter setting previously shown in Fig. 2, while the convolutional layers 12–15 use the same configuration of layer 11 (15 filters with dimensions 3×3). The CNN and RNN architectures are the same; the only difference is that skip connections and addition layers are not employed in the CNN. It is clear that for nine convolutional layers the CNN gets saturated and then the accuracy starts decreasing when other layers are added. Conversely, the RNN accuracy steadily increases and eventually reaches a stable value for 11 convolutional layers. We observed this decrease in the accuracy of the CNN predictions also for other tests (not shown here for brevity) that employed different hyperparameter settings (i.e., size of convolutional filters, type of activation function, learning rate). Therefore, we deem that this result is not related to the specific CNN architecture we used.

Figure 7 shows some projections of the multivariate modelling error (assumed to be Gaussian distributed) onto specific model space directions. As expected, the standard deviation of the modelling error increases as the depth of the considered layer increases. For example, a high precision characterizes the estimated velocity and thickness of the first layer, while higher uncertainties affect the estimated V_s and thickness of the fourth layer.

First, we apply the RNN inversion under the assumption that the number of layers is perfectly known. In other words, the RNN is applied to invert phase-velocity spectra generated by five-layer models. We also compare the RNN predictions with those provided by a more standard PSO inversion in which the error function to be minimized is a linear combination of data fitting and a model regularization term:

$$E(\mathbf{m}) = \|\mathbf{d} - G(\mathbf{m})\|_2^2 + \gamma \|\mathbf{m} - \mathbf{m}_{\text{prior}}\|_2^2. \quad (5)$$

Similar, to the RNN inversion the model vector \mathbf{m} only includes the V_s and thickness of the layers, and the number of layers is assumed perfectly known, while the Poisson's ratio and density are kept fixed to their true values. Different from

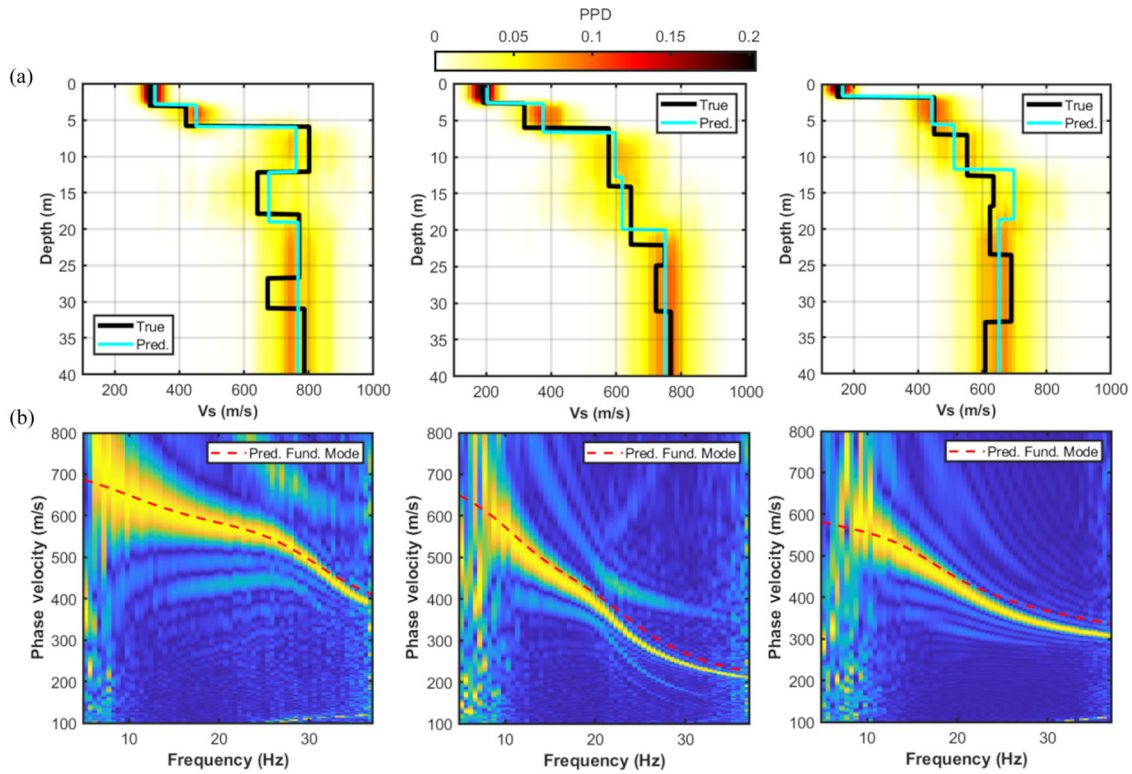


Figure 14 RNN results on seven-layer models. (a) Comparison between the true V_s profile, the RNN predictions, and the MC PPD estimations. (b) Comparison between the phase velocity spectra associated with the true models shown in (a) and the fundamental mode computed on the RNN results.

RNN, in the PSO inversion only the picked fundamental mode is considered as the observed data \mathbf{d} . In equation (5), G constitutes the forward modelling operator that computes the fundamental mode for the considered model \mathbf{m} . In our case, the

Haskell–Thompson method is employed (Haskell, 1953). The prior model $\mathbf{m}_{\text{prior}}$ is equal to the central values of the ranges used to define the training and validation examples (see Fig. 3). The optimal value of the trade-off parameter γ has been set

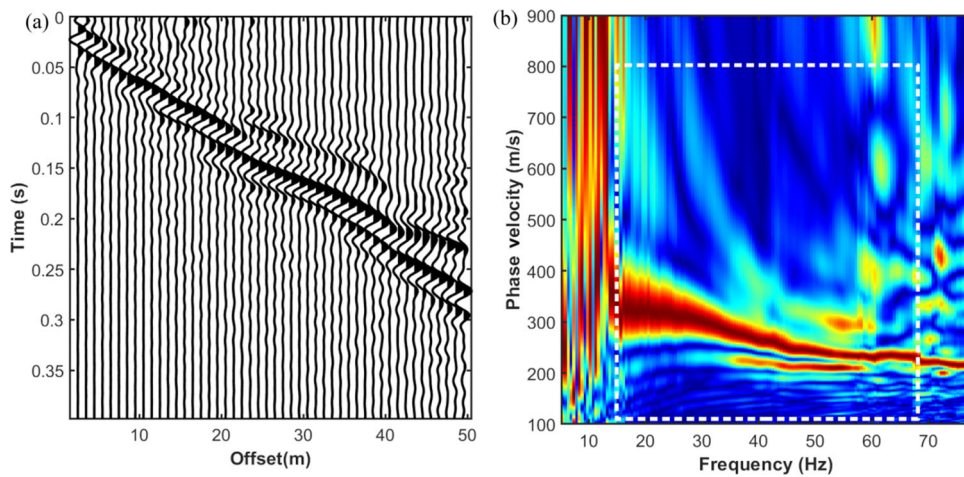


Figure 15 (a) The average and band-pass filtered observed shot gather used to compute the observed phase velocity spectrum in (b). In (b) the dotted white rectangle encloses the portion of the dispersion image considered as observed data in the RNN inversion.

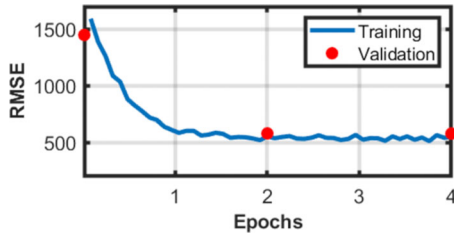


Figure 16 Evolution of the RMSE error on the training and validation sets over epochs.

through the L-curve approach (Aster *et al.*, 2018). The PSO has been run for 150 iterations and employing a population of 100 particles randomly initialized over the ranges shown in Fig. 3. A single PSO run takes 20 minutes, approximately, using a parallel Matlab implementation running on the quad-core i-7 Intel previously mentioned.

For both RNN and PSO, the data input for the inversion has been derived from shot gathers contaminated with uncorrelated Gaussian noise with a standard deviation equal to 50% of the standard deviation of the whole noise-free gather (Fig. 8). As expected the Gaussian noise added to the seismic dataset mainly affects the low and high frequencies (i.e., frequencies lower than 10 Hz and higher than 30 Hz). The noise model is assumed to be known during the inversions.

Figures 9(a) and 10(a) show the RNN and PSO predictions obtained for six different subsurface models extracted from the validation set. In all cases, the RNN predictions well reproduce the true V_s profiles and are much closer to the true model than the PSO results. Note that the RNN inversion recovers the velocity reversals and the significant V_s contrasts in the true models. Figures 9(b) and 10(b) compare the observed phase velocity spectrum and the fundamental mode computed on the PSO and RNN results. These figures illustrate that both approaches provide final data predictions that well reproduce the fundamental mode. Therefore, if we focus the attention on the PSO prediction we note that it provides good data fitting, but poor model accuracy. This suggests the severe ill-conditioning of the inverse problem and that the PSO results can be improved either by including a more informative prior model (e.g., narrower model search ranges and a $\mathbf{m}_{\text{prior}}$ vector closer to the true solution), or by including higher modes in the observed data. Figure 11 compares the first higher modes predicted by the RNN and PSO results, and we observe that the higher mode is better recovered by the RNN inversion. This comparison is limited to models 4–6 previously shown in Fig. 10 because the first higher mode is easily recognizable in the phase velocity spectrum.

Differently from the PSO approach, the implemented RNN inversion also allows for an accurate assessment of the model uncertainties. Figure 12 shows the PPD provided by the MC sampling for the six models considered in Figs. 9 and 10. As expected, we observe that the uncertainty increases moving from the shallow to the deep part of the model, where the data illumination is poor. To numerically compute the PPD, we use 1000 MC simulations that take 2 minutes of computing time to be generated when a parallel Matlab code runs on the quad-core i-7 Intel. Notably, in all cases, the true model lies within the velocity range spanned by the MC simulations. This illustrates the reliability of the implemented inversion workflow.

We now investigate the effects of an erroneous subsurface parameterization on the RNN predictions. To this end, we use the previously shown velocity ranges (see Fig. 3) to randomly generate three models with four and seven layers, respectively. Then we use the previously trained network (that inherently assumes a five-layer parameterization) to invert the phase velocity spectra generated on these new models. For the four-layer case (Fig. 13a), we observe that the RNN estimates include some fictitious layers in the deeper part of the model (i.e., where data illumination is poor), but characterized by V_s values similar to the actual model. The true models always lie in the velocity range spanned by the MC simulations, and, as expected, the accuracy and precision of the results decrease as the depth increases. This test illustrates that despite the overparameterization, the overall velocity profile is well recovered by the trained RNN, especially in the shallowest part of the subsurface. The fundamental mode generated by the predicted RNN model well reproduces the fundamental mode visible in the phase velocity spectrum (Fig. 13b). In contrast, when the number of layers is underestimated (i.e., under parameterization of the inverse problem), some biased predictions start to appear also in the shallow part of the model (i.e., erroneous velocity and thickness estimations; Fig. 14a). In addition, the true model lies sometimes at the extreme edge of the V_s range spanned by the MC simulations (e.g., the leftmost panel of Fig. 14(a) around 30 m depth) and the predicted fundamental modes show larger mismatches with the fundamental modes visible in the observed spectra with respect to the previous test (i.e., compare Fig. 13b with Fig. 14b).

These experiments demonstrated that an overparameterization is preferable to an under parameterization: in the former case, the true velocity profile is well recovered and the dispersion pattern is well reproduced by the estimated model. In the latter, the RNN provides biased predictions and the data match decreases. For this reason, poor data fitting may indicate an under parameterization of the inverse problem. In all

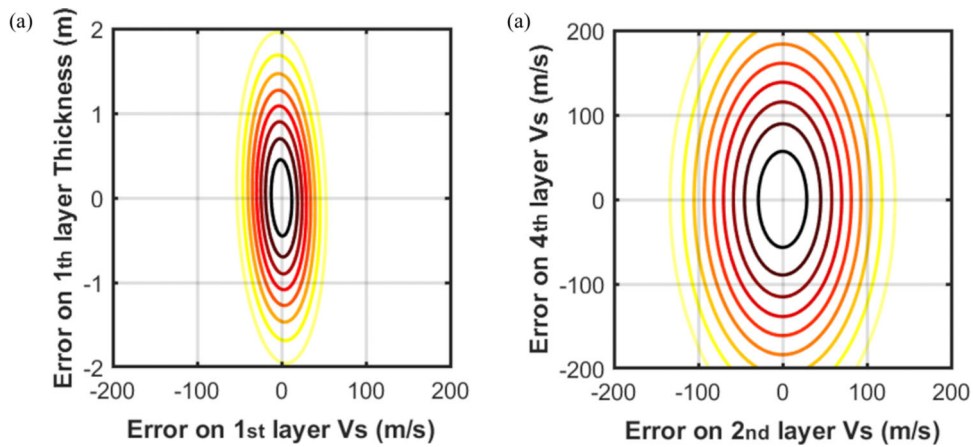


Figure 17 The multivariate Gaussian modelling error projected onto some model space directions. (a) Projection onto the directions pertaining to the Vs and thickness of the first layer. (c) Projection onto the directions defined by the Vs of the second and fourth layers. In (a) and (b) light and dark colours code low and high probability values.

cases, if different parameterizations result in similar data fitting, the model with the minimum number of layers should be preferred.

FIELD DATA INVERSION

We apply the implemented approach to a single seismic gather of the interPACIFIC project (Garofalo *et al.*, 2016). The source is an 8-kg sledgehammer and the minimum and maximum offset are 3 m and 50 m, respectively, for a receiver interval of 1 m and an off-end acquisition layout. For the considered source position, 12 different shots have been recorded. The sample-by-sample covariance estimated on these 12 shots gives an approximated estimate of the data covariance matrix to be used in the MC uncertainty propagation (Aleardi *et al.*, 2018). The observed phase velocity spectrum has been computed on the average seismic record derived from the 12 available shots after band-pass filtering (7–80 Hz). Figure 15 shows the so obtained shot gather and the associated phase velocity spectrum. The main difficulty posed by this dataset is related to the type of source employed and the consequent lack of low frequencies, which are crucial to constrain the Vs of the deep layers. Due to the low signal-to-noise ratio at frequencies below 17 Hz and higher than 70 Hz, the frequency range of interest for the RNN inversion is 17–68 Hz (Fig. 15b). For this reason, we expect that the accuracy and precision of the inversion results significantly decrease as the depth increases.

Due to the very different frequency ranges characterizing this field data with respect to the previous synthetic test, we derive new training and validation sets of 19,000 and 1000 examples, respectively, generated on the same Vs

range depicted in Fig. 3. The training seismograms are generated using a source wavelet estimated from the field data (Xing and Mazzotti, 2019). Based on previous results obtained in the investigated area (Xing and Mazzotti, 2019), we first consider an eight-layer parameterization. The generation of the 20,000 examples takes 1.3 hours, using the parallel hybrid Matlab-Fortran code running on the two deca-core Intel previously described. We maintain the same network architecture previously described (Fig. 2) that we train for four epochs (Fig. 16) for a total computing time of 6 minutes on the quad-core Intel. Again the similar RMSE values computed on the validation and training sets indicate the generalization capability of the network and that overfitting has been avoided. Note that in this case, the final RMSE error is higher than the one previously obtained for the synthetic test (Fig. 4). This decreased accuracy is probably related to the lack of low frequencies in the observed dispersion data.

Similar to the synthetic test, the difference between desired and actual RNN outputs gives the modelling error distribution. As an example, Fig. 17 shows some projections of this multivariate Gaussian error onto specific model space directions. We again observe that the uncertainty increases as the depth of the considered layer increases. Due to the lack of low frequency, we also observe a decreased precision with respect to the synthetic example (see Fig. 7).

Figure 18 shows examples of RNN predictions and the associated uncertainty estimations obtained on two models extracted from the validation set. To simulate a realistic situation close to the one expected for the real dataset, we compute the observed phase velocity spectra after contaminating the

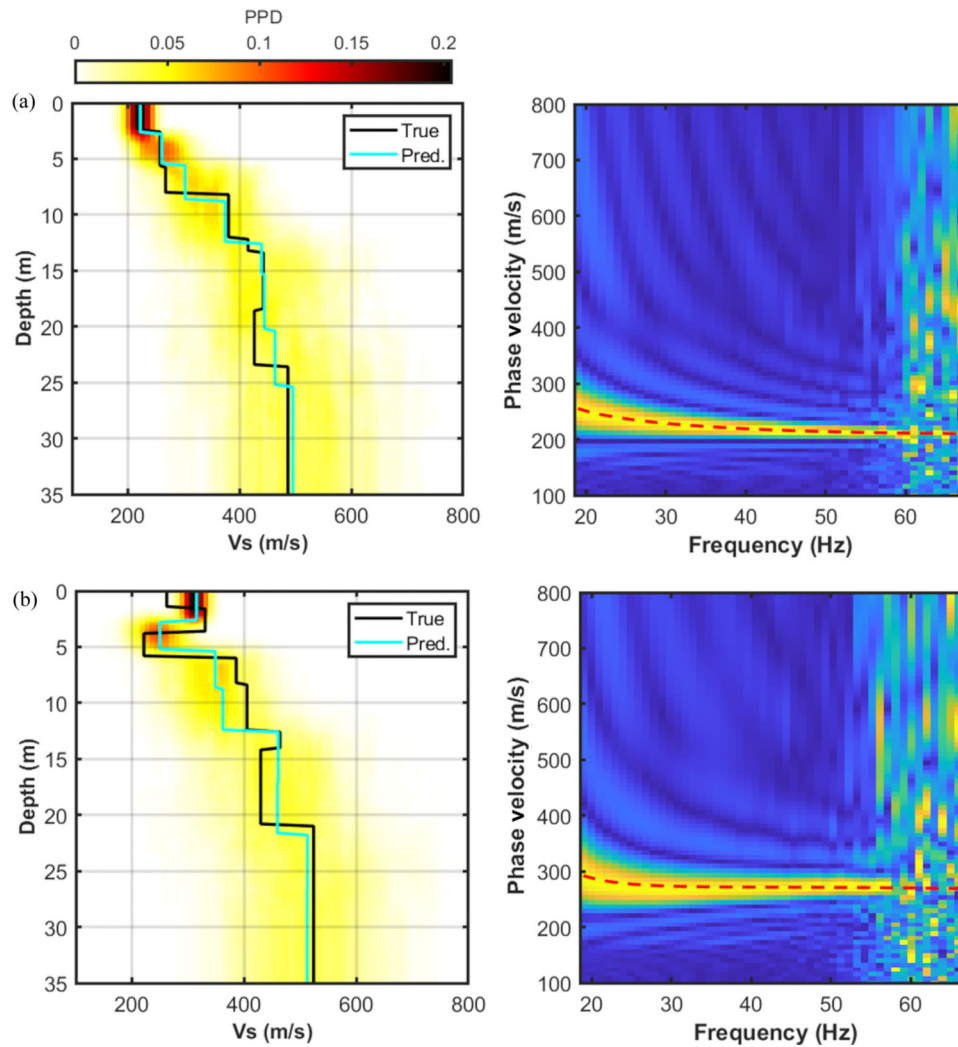


Figure 18 (a) and (b) RNN predictions and MC uncertainty estimations were obtained for two models extracted from the validation set. In the right panels, the dotted red line represents the fundamental mode computed on the RNN prediction.

noise-free synthetic gathers with the noise model estimated for the field data. We observe that the predicted RNN model well reproduces the true V_s profile, especially at shallow depth, although the overall accuracy is decreased with respect to the synthetic experiments. However, the fundamental modes generated by the RNN predictions well match the fundamental modes visible in the observed dispersion patterns for the considered frequency range. As expected, the uncertainty significantly increases as the depth increases and the estimated PPD illustrates that the V_s values below 15–20 m depth lie in the so-called equivalence region of solutions (Fernandez Martinez *et al.*, 2012).

Figure 19(a) shows the inversion results obtained on the field data for an eight-layer parameterization. Again the velocities of the shallow layers are estimated with high preci-

sion, while the uncertainty rapidly increases in the deeper part of the model. In Fig. 19(b), we observe that the RNN estimated model well reproduces the fundamental mode visible in the observed dispersion pattern, but also the interpreted first higher mode. Therefore, we deem that the final predictions are in agreement with the observed data and with the assumed noise and modelling errors.

We repeat the field data inversion, but considering a 6-layer parameterization. Similar to the previous test, we generate new 20,000 six-layer examples and we retrain the network with the same architecture. The inversion results are represented in Fig. 20. The predicted V_s profile in the shallowest part of the model (down to 8 m) is very similar to the results obtained for an eight-layer model. The differences become more significant around 10 m depth where the

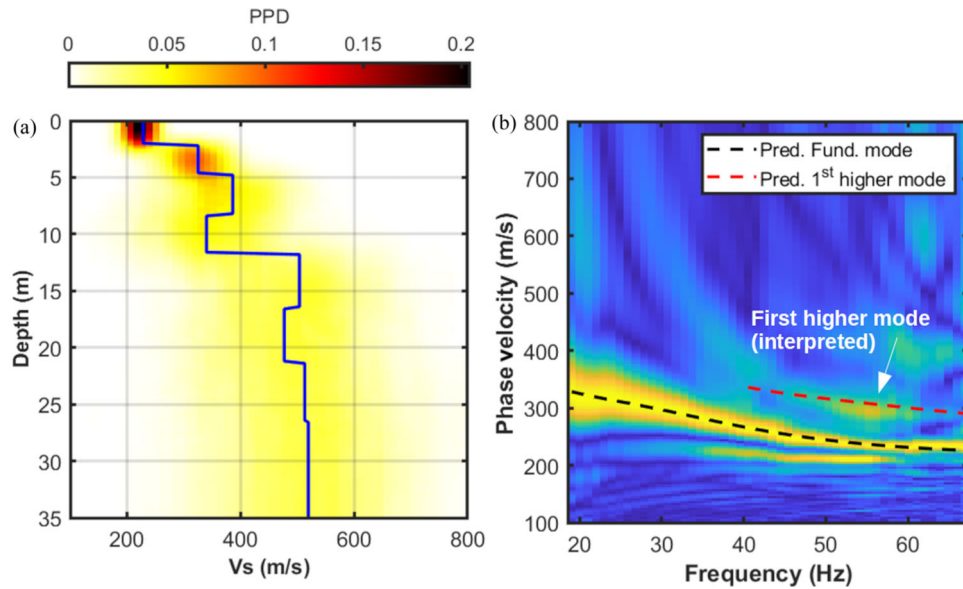


Figure 19 Field inversion result for an eight-layer parameterization. (a) RNN-predicted model (blue line) and associated PPD estimations. (b) Comparison between the observed phase velocity spectrum and the fundamental and first higher modes computed on the RNN prediction. The white arrow points towards the interpreted first higher mode visible in the observed spectrum.

eight-layer parameterization predicts a V_s reversal with higher velocity contrasts with the surrounding layers. The shear wave velocity in the deeper part of the model is higher when the six-layer parameterization is considered, although this part is associated with significant model uncertainties. Compared to Fig. 19(b), the RNN prediction for a six-layer model gener-

ates a fundamental and, especially, a first higher mode with increased differences with the observed dispersion pattern.

DISCUSSION

We proposed a hybrid RNN–MC inversion procedure for the estimation of the V_s profile from surface wave dispersion data

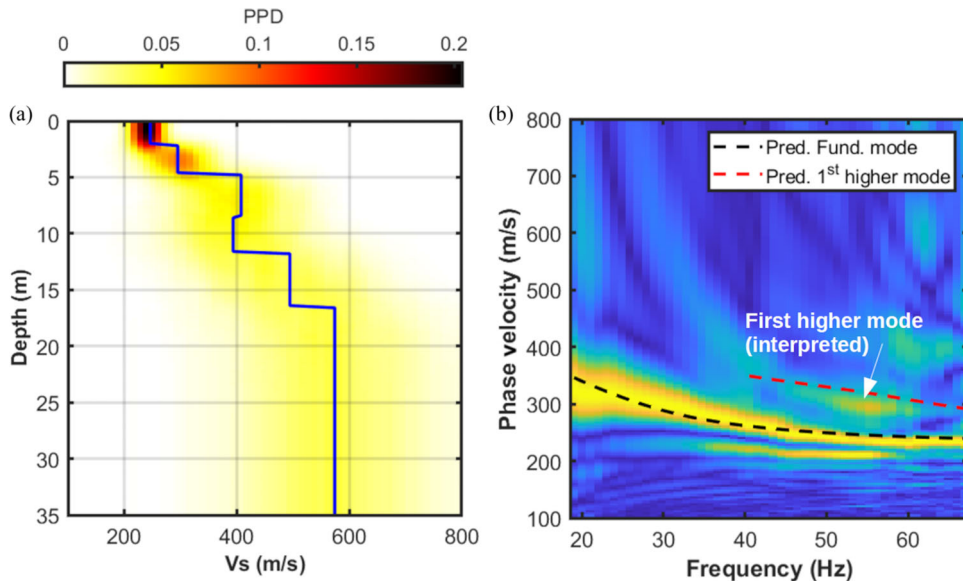


Figure 20 Field inversion result for a six-layer parameterization. (a) RNN-predicted model (blue line) and associated PPD estimations. (b) Comparison between the observed phase velocity spectrum and the fundamental and first higher modes computed on the RNN prediction. The white arrow points towards the interpreted first higher mode visible in the observed spectrum.

under the assumption of a locally 1D layered model. The use of RNN instead of standard CNN allowed us to train a deep network while mitigating the vanishing gradient problem. We exploited the full phase velocity spectrum to avoid both the human effort related to the picking procedure and the ambiguity affecting the modal identification. The inclusion of a MC simulation was aimed at properly propagating onto the RNN prediction both the noise affecting the observed data and the modelling error associated with the network approximation. Different from standard inversion approaches, the implemented method does not include any model constraint into the error function, but the network is trained on a dataset containing realistic subsurface scenarios and learns how to reproduce a similar model to fit the input data.

The first stage of the presented approach is the data generation that is the most computationally demanding although perfectly parallelizable. However, the RNN inversion requires a relatively small dataset for training and indeed the computing time for generating the ensemble of 20,000 models forming the training and validation sets was less than 2 hours in both synthetic and field applications. The computational cost of the generation phase can be drastically reduced by employing optimized codes running on a larger computer cluster. The learning phase is much less computationally demanding and took only 6 minutes in both synthetic and field example cases. This means that different network architectures can also be tested at an affordable computational cost.

The main limitation of the implemented method is that the network can only be trained for a fixed parameterization (i.e. , for a fixed number of layers). This means that if no previous information is available about the investigated area, different training and validation sets with different numbers of layers must be generated and different networks must be trained. Once these networks have been trained and applied to the observed data, the optimal model parameterization can be chosen based on the difference between the observed and predicted data. If different parameterizations give comparable data predictions the one with the minimum number of layers should be preferred. The main benefit of our inversion algorithm is that once the network has been trained it provides V_s predictions in real-time and also the associated uncertainties can be assessed with a very limited extra computational effort. For this reason, the proposed approach is particularly useful when several dispersion data pertaining to two- or three-dimensional seismic acquisitions must be inverted.

Now we are trying to extend the method to invert the full seismic gather, thus overcoming the 1D assumption. In this context, compression techniques could be used to reduce

the dimensions of the input and output of the network and to make the computing time of the learning phase affordable (Aleardi and Salusti, 2021).

CONCLUSIONS

We presented a hybrid RNN–MC approach to invert surface wave dispersion data that also provides the uncertainty affecting the recovered solution. Synthetic and field experiments showed very promising results and demonstrated that a RNN can effectively approximate the inverse of a nonlinear operator that is very difficult and expensive to compute. The implemented approach needs a relatively small training set for the learning process, and once the network has been trained it provides model predictions and associated uncertainties in near real-time.


CONFLICT OF INTEREST


The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

Data are available on request from the corresponding author. The field dataset is freely available at <http://intepacific.geopsy.org/>

ORCID

Mattia Aleardi  <https://orcid.org/0000-0003-1433-0281>

Eusebio Stucchi  <https://orcid.org/0000-0002-7555-7140>

REFERENCES

- Aleardi, M., Salusti, A. and Pierini, S. (2020) Trans-dimensional and Hamiltonian Monte Carlo inversions of Rayleigh-wave dispersion curves: A comparison on synthetic datasets. *Near Surface Geophysics*, 18(5), 515–543.
- Aleardi, M. (2020) Combining discrete cosine transform and convolutional neural networks to speed up the Hamiltonian Monte Carlo inversion of pre-stack seismic data. *Geophysical Prospecting*, 68(9), 2738–2761.
- Aleardi, M., Ciabbari, F. and Gukov, T. (2018) A two-step inversion approach for seismic-reservoir characterization and a comparison with a single-loop Markov-chain Monte Carlo algorithm. *Geophysics*, 83(3), R227–R244.
- Aleardi, M. and Salusti, A. (2021) Elastic pre-stack seismic inversion through discrete cosine transform reparameterization and convolutional neural networks. *Geophysics*, 86(1), R129–R146.

- Aster, R.C., Borchers, B. and Thurber, C.H. (2018) *Parameter Estimation and Inverse Problems*. Elsevier.
- Araya-Polo, M., Jennings, J., Adler, A. and Dahlke, T. (2018) Deep-learning tomography. *The Leading Edge*, 37(1), 58–66.
- Boiero, D., Bergamo, P., Bruno Rege, R. and Socco, L.V. (2011) Estimating surface-wave dispersion curves from 3D seismic acquisition schemes: Part 1 – 1D models. *Geophysics*, 76(6), G85–G93.
- Cercato, M. (2011) Global surface wave inversion with model constraints. *Geophysical Prospecting*, 59(2), 210–226.
- Cercato, M., De Donno, G., Di Giulio, A., Lanzo, G. and Tommasi, P. (2020) Dynamic characterization of the hill of Civita di Bagnoregio (Viterbo, Central Italy) for seismic response analysis. *Engineering Geology*, 266, 105463.
- Cheng, X., Liu, Q., Li, P. and Liu, Y. (2019) Inverting Rayleigh surface wave velocities for crustal thickness in eastern Tibet and the western Yangtze craton based on deep learning neural networks. *Nonlinear Process in Geophysics*, (2), 61–71.
- Cox, B.R. and Teague, D.P. (2016) Layering ratios: A systematic approach to the inversion of surface wave data in the absence of a priori information. *Geophysical Journal International*, 207(1), 422–438.
- Dai, T., Xia, J., Ning, L., Xi, C., Liu, Y. and Xing, H. (2020) Deep learning for extracting dispersion curves. *Surveys in Geophysics*, 42(1), 69–95.
- Dal Moro, G., Pipan, M. and Gabrielli, P. (2007) Rayleigh wave dispersion curve inversion via genetic algorithms and marginal posterior probability density estimation. *Journal of Applied Geophysics*, 61, 39–55.
- Di Giulio, G., Punzo, M., Bruno, P.P., Cara, F. and Rovelli, A., (2019) Using a vibratory source at Mt. Etna (Italy) to investigate the wavefield polarization at Pernicana fault. *Near Surface Geophysics*, 17(4), 313–329.
- Farrugia, D., Paolucci, E., D’Amico, S. and Galea, P. (2016) Inversion of surface wave data for subsurface shear wave velocity profiles characterized by a thick buried low-velocity layer. *Geophysical Journal International*, 206(2), 1221–1231.
- Feng, S., Sugiyama, T. and Yamanaka, H. (2005) Effectiveness of multimode surface wave inversion in shallow engineering site investigations. *Exploration Geophysics*, 36, 26–33.
- Fernandez Martinez, J.L., Fernandez Muniz, M.Z. and Tompkins, M.J. (2012) On the topography of the cost functional in linear and nonlinear inverse problems. *Geophysics*, 77(1), W1–W15.
- Foti, S., Hollender, F., Garofalo, F., Albarello, D., Asten, M., Bard, P.Y., et al. (2018) Guidelines for the good practice of surface wave analysis: A product of the InterPACIFIC project. *Bulletin of Earthquake Engineering*, 16(6), 2367–2420.
- Garofalo, F., Foti, S., Hollender, F., Bard, P.Y., Cornou, C., Cox, B., et al. (2016) InterPACIFIC project: Comparison of invasive and non-invasive methods for seismic site characterization. Part II: Inter-comparison between surface-wave and borehole methods. *Soil Dynamics and Earthquake Engineering*, 82, 241–254.
- Glorot, X. and Bengio, Y. (2010) Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, 249–256.
- Goodfellow, I., Bengio, Y. and Courville, A. (2016), *Deep Learning*. MIT Press.
- Hahnloser, R.H., Sarpeshkar, R., Mahowald, M.A., Douglas, R.J. and Seung, H.S. (2000) Digital selection and analogue amplification co-exist in a cortex-inspired silicon circuit. *Nature*, 405(6789), 947–951.
- Hansen, T.M. and Cordua, K.S. (2017) Efficient Monte Carlo sampling of inverse problems using a neural network-based forward – applied to GPR crosshole traveltime inversion. *Geophysical Journal International*, 211(3), 1524–1533.
- Haskell, N.A. (1953) The dispersion of surface waves on multi-layered media. *Bulletin of the Seismological Society of America*, 43, 17–34.
- He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778.
- He, K., Zhang, X., Ren, S. and Sun, J. (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, 1026–1034.
- Hu, J., Qiu, H., Zhang, H. and Ben-Zion, Y. (2020) Using deep learning to derive shear-wave velocity models from surface-wave dispersion data. *Seismological Research Letters*, 91(3), 1738–1751.
- Kennett, B.L.N., (1983), *Seismic Wave Propagation in Stratified Media*. Cambridge University Press.
- Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2017) Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90.
- Lai, C.G. (2005) Surface waves in dissipative media: Forward and inverse modelling. In *Surface Waves in Geomechanics: Direct and Inverse Modelling for Soils and Rocks*. Springer, Vienna, pp. 73–163.
- LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep learning. *Nature*, 521(7553), 436–444.
- Lewis, W. and Vigh, D. (2017) Deep learning prior models from seismic images for full-waveform inversion. 87th Annual International Meeting, SEG, Expanded Abstracts, 1512–1517.
- Luo, Y., Xia, J., Miller, R.D., Xu, Y., Liu, J. and Liu, Q. (2009) Rayleigh-wave mode separation by high-resolution linear Radon transform. *Geophysical Journal International*, 179(1), 254–264.
- Maraschini, M. and Foti, S. (2010) A Monte Carlo multimodal inversion of surface waves. *Geophysical Journal International*, 182(3), 1557–1566.
- Meier, U., Curtis, A. and Trampert, J. (2007) Global crustal thickness from neural network inversion of surface wave data. *Geophysical Journal International*, 169(2), 706–722.
- Moghadas, D. (2020) One-dimensional deep learning inversion of electromagnetic induction data using convolutional neural network. *Geophysical Journal International*, 222(1), 247–259.
- Monajemi, H., Donoho, D.L. and Stodden, V. (2016) Making massive computational experiments painless. In 2016 IEEE International Conference on Big Data, 2368–2373.
- Othman, A., Iqbal, N., Hanafy, S.M. and Waheed, U.B. (2021) Automated event detection and denoising method for passive seismic data using residual deep convolutional neural

- networks. *IEEE Transactions on Geoscience and Remote Sensing*, <https://doi.org/10.1109/TGRS.2021.3054071>.
- Park, C.B., Miller, R.D. and Xia, J. (1999) Multichannel analysis of surface waves. *Geophysics*, 64(3), 800–808.
- Park, M.J. and Sacchi, M.D. (2020) Automatic velocity analysis using convolutional neural network and transfer learning. *Geophysics*, 85(1), V33–V43.
- Puzryev, V. (2019) Deep learning electromagnetic inversion with convolutional neural networks. *Geophysical Journal International*, 218(2), 817–832.
- Rahman, M.Z., Kamal, A.M. and Siddiqua, S. (2018) Near-surface shear wave velocity estimation and Vs 30 mapping for Dhaka City. *Bangladesh. Natural Hazards*, 92(3), 1687–1715.
- Richardson, A. (2018) Seismic full-waveform inversion using deep learning tools and techniques. arXiv preprint:1801.07232.
- Sambridge, M. and Mosegaard, K. (2002) Monte Carlo methods in geophysical inverse problems. *Reviews of Geophysics*, 40(3), 3–1.
- Santurkar, S., Tsipras, D., Ilyas, A. and Madry, A. (2018) How does batch normalization help optimization?. *Advances in Neural Information Processing Systems*, 31, 2483–2493.
- Sajeva, A. and Menanno, G. (2017) Characterisation and extraction of a Rayleigh-wave mode in vertically heterogeneous media using quaternion SVD. *Signal Processing*, 136, 43–58.
- Schmidhuber, J. (2015) Deep learning in neural networks: An overview. *Neural networks*, 61, 85–117.
- Serdyukov, A.S., Yablokov, A.V., Chernyshov, G.S. and Azarov, A.V. (2017) The surface waves-based seismic exploration of soil and ground water. In *IOP Conference Series: Earth and Environmental Science*, 53(1), 012010.
- Socco, L.V. and Strobba, C. (2004) Surface-wave method for near-surface characterization: A tutorial. *Near-Surface Geophysics*, 2(4), 165–185.
- Socco, L.V., Boiero, D., Foti, S., Maraschini, M., Piatti, C., Bergamo, P., et al. (2010) Surface wave analysis for S-wave static correction computation. In *SEG Technical Program Expanded Abstracts*, 1929–1933.
- Socco, L.V. and Boiero, D. (2008) Improved Monte Carlo inversion of surface wave data. *Geophysical Prospecting*, 56(3), 357–371.
- Strobba, C., Vermeer, P., Laake, A., Glushchenko, A. and Re, S. (2010) Surface waves: Processing, inversion and removal. *First Break*, 28(8), 85–91.
- Sun, J., Slang, S., Elboth, T., Larsen Greiner, T., McDonald, S. and Gelius, L.J. (2020) A convolutional neural network approach to deblending seismic data. *Geophysics*, 85(4), WA13–WA26.
- Tarantola, A. (2005) Inverse problem theory and methods for model parameter estimation. *Society for Industrial and Applied Mathematics*.
- Waldeland, A.U., Jensen, A.C., Gelius, L.J. and Solberg, A.H.S. (2018) Convolutional neural networks for automated seismic interpretation. *The Leading Edge*, 37(7), 529–537.
- Wang, B., Zhang, N., Lu, W. and Wang, J. (2019) Deep-learning-based seismic data interpolation: A preliminary result. *Geophysics*, 84(1), V11–V20.
- Wang, L., Meng, D., Wu, B. and Liu, N. (2020) Seismic inversion via closed-loop fully convolutional residual network and transfer learning. Society of Exploration Geophysicists. In *SEG Technical Program Expanded Abstracts 2020*, 1521–1525.
- Wu, Y. and McMechan, G.A. (2019) Parametric convolutional neural network-domain full-waveform inversion. *Geophysics*, 84(6), R881–R896.
- Xia, J., Miller, R.D., Park, C.B. and Tian, G. (2003) Inversion of high frequency surface waves with fundamental and higher modes. *Journal of Applied Geophysics*, 52(1), 45–57.
- Xing, Z. and Mazzotti, A. (2019) Two-grid full-waveform Rayleigh-wave inversion via a genetic algorithm – Part 2: Application to two actual data sets. *Geophysics*, 84(5), R815–R825.
- Yablokov, A. and Serdyukov, A. (2020) Inversion of surface waves dispersion curves using artificial neural network. In *82nd EAGE Annual Conference & Exhibition*, 1–5.
- Yang, L., Chen, W., Liu, W., Zha, B. and Zhu, L. (2020) Random noise attenuation based on residual convolutional neural network in seismic datasets. *IEEE Access*, 8, 30271–30286.