Elastic properties and litho-fluid facies estimation from pre-stack seismic data through bi-directional long short-term memory networks.

Mattia Aleardi

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

Corresponding author: Mattia Aleardi, mattia.aleardi@unipi.it

ABSTRACT

One of the main goals of pre-stack seismic inversion is the estimation of elastic properties (i.e., P-, S- wave velocities and density) and litho-fluid classes in the investigated area. To this end, many inversion strategies have been proposed but the most popular is based on a two-step inversion approach: first elastic properties are inferred from pre-stack data, then a classification algorithm is used to convert the outcomes of the first stage into litho-fluid facies. In this work, we propose an alternative approach based on recurrent neural networks. We train two bidirectional long short-term memory networks to predict the inverse mappings from pre-stack seismic data to elastic properties, and litho-fluid classes. In the elastic inversion, we also use a Monte Carlo simulation approach to properly propagate onto the model space both the uncertainties related to noise contamination in the data and to the modeling error introduced by the network approximation. One crucial 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 drawn according to a previously defined elastic and facies prior models derived from actual well log recordings. In particular, we assume a Gaussian-mixture elastic prior, and we also take into account the uncertainties affecting the estimation of the transition probabilities of facies. We invert each seismic gather independently, and in this context, the generation of the training set, and the learning process can be accomplished with a very

limited computational effort on a common notebook. Once trained, the networks estimate the elastic properties, the litho-fluid facies, and the related uncertainties from the pre-stack data in near real-time. Synthetic and field data inversions are used to validate the proposed method. The network predictions are also benchmarked against the outcomes of a more standard two-step approach that combines a linear elastic inversion and a subsequent point-wise Bayesian classification. Our results demonstrate that the implemented algorithm guarantees more accurate elastic property estimations and facies predictions than the standard inversion strategy. In particular, the predictions provided by the long short-term memory network are less affected by erroneous assumptions on the noise statistics and prior model, and by errors in the estimated source wavelet.

Running head: LSTM pre-stack inversion and facies classification

Keywords: Seismic inversion; AVA

INTRODUCTION

Pre-stack seismic data are routinely used to infer elastic properties (such as P-, S-wave velocity and density; *Vp*, *Vs*, and ρ) and litho-fluid facies in the subsurface (Avseth et al. 2005; Doyen, 2007; Kemper and Gunning, 2014; Bachrach and Gofer, 2019). This task can be formulated as an inverse problem that is inherently ill-posed meaning that multiple solutions reproduce well the observed data. The inversion is also complicated by the fact that the model space includes both continuous and discrete model parameters (the elastic properties and litho-fluid facies, respectively). Several approaches have been proposed over the last decades, and many of them formulate the inversion into a probabilistic framework (Tarantola, 2005) to properly assess the uncertainties affecting the estimated elastic parameters and litho-fluid classes. The most popular strategy solves the problem using a two-step cascaded workflow in which the computational workload is reduced by inverting each seismic gather separately. In this context, the lateral continuity of the predictions only depends

on the lateral continuity of the seismic data. The first step estimates elastic properties from pre-stack seismic data, then the outcomes of this first stage are converted into fluid classes via a classification algorithm. The first step of seismic inversion usually employs a linear forward modeling operator (e.g., a linearization of the full Zoeppritz equations) under Gaussian assumptions for noise and model parameter distributions (e.g., the popular approach proposed by Buland and Omre, 2003). To mitigate the ill-conditioning, geostatistical vertical constraints are usually employed to correctly preserve plausible vertical variability in the predicted elastic parameters (Bongajum et al. 2013; Bosch et al. 2015; Azevedo and Soares, 2017). To simplify the classification process, standard approaches estimate facies independently at each location, although it is known that the facies model must be vertically correlated to mimic depositional processes and gravity effects. Examples of popular classification methods include Bayesian classification (Doyen, 2007), maximum-likelihood approach (Da Veiga and Le Ravalec, 2012), Monte Carlo classification (Grana et al. 2012), pattern recognition and clustering algorithms (Martinez and Martinez, 2007; Hastie et al., 2009) such as support vector machine algorithms and discriminant analysis (Wang et al. 2014; Negahdari et al. 2014). Fuzzy logic, k-nearest neighbors, and artificial neural network have also been employed (Dubois et al. 2007; Hall, 2016; Aleardi and Ciabarri, 2017). Spatial/temporal facies constraints are usually modeled by including a stationary prior Markov-chain or Markov-random field for the discrete property (Larsen et al. 2006; Ulmoven and Omre, 2010; Rimstad et al. 2012; Lindberg and Grana, 2015; Fjeldstad and Omre, 2017; Talarico et al. 2020), although this increases the overall computational cost of the classification procedure as compared to point-wise methods. The two-step approach although computationally fast has several drawbacks related to error propagation from data space to the facies model, and to the consistency between the observed seismic data and the predicted classes (Doyen, 2007; Zunino et al. 2015). To circumvent these issues several strategies have been proposed; these directly infers litho-fluid facies from seismic data (Rimstad and Omre, 2010; Grana et al. 2020), or integrate the seismic inversion and the classification process into a single workflow (Connolly et al.

2016; Aleardi et al. 2018; de Figueiredo et al. 2018; de Figueiredo et al. 2019; Aleardi and Salusti, 2020a).

Over the last years, the recent advent of high-speed multi-core CPUs and GPUs have extensively promoted the applications of machine learning algorithms to solve geophysical inverse problems (Monajemi et al., 2016; Goodfellow et al. 2016). Machine learning approaches can be used to solve both regression problems (i.e., in the case of continuous model parameters) or classification problems (for discrete properties). They can be very useful when the forward relation is known, but the inverse mapping is either expensive to compute analytically or numerically approximate. Therefore, the network is trained to predict the mapping between the data space and the discrete or continuous parameters. Some applications of machine learning methods to solve geophysical problems 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), Sun (2020), Aleardi (2020), Aleardi and Salusti (2020b), Moghadas (2020).

Many different types of machine learning algorithms exist such as artificial neural networks (ANNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) (Bishop, 2006; Goodfelow et al. 2016; Geron et al. 2019). Artificial Neural Networks are theoretically capable of learning any nonlinear function linking the output and the input response. Hence, these networks are popularly known as universal function approximators. Some of the drawbacks of ANNs are the computational cost, the elevated number of internal parameters to be estimated (that exponentially increases with the dimension of the input and output responses), the overfitting issue, and the vanishing/exploding gradient problems. Besides, in cases of image applications, ANNs lose the spatial relationships between the pixels of the image and hence cannot capture sequential information in the data. CNN uses convolutional kernels to extract the relevant features from the input and hence they are commonly used in image classification problems. Differently to ANN, CNN captures the spatial variability pattern from an image and exploits parameter sharing to decrease the number of

internal parameters to optimize. However, CNN treats each input response separately and hence they cannot capture sequential information hidden in the input data. To overcome this issue RNNs have been proposed to process sequential data (i.e., seismograms). The basic RNN architecture is similar to an ANN but with recurrent connections in the hidden state. Similar to CNN they use parameter sharing to reduce the number of internal parameters. RNNs use an internal state (memory) to share features learned across the different positions within a time series. Hence, they can learn the dynamic temporal relations within a time sequence. However, standard RNNs cannot infer long-term dependencies in the input set and are also severely affected by vanishing/exploding gradient problems. To overcome these issues Hochreiter and Schmidhuber (1997) proposed the Long Short Term Memory networks (LSTM) that are a special kind of RNN, capable of learning long-term dependencies. It has been demonstrated that LSTMs work tremendously well on a large variety of problems and nowadays are routinely employed in handwriting and speech recognition (Graves et al. 2012).

Applications of RNN and LSTM to solve geophysical problems are recent and date back to the very last years. These examples refer to earthquake classification (Kuyuk and Susumu, 2018), detection of earthquake precursors (Cai et al. 2019), earthquake magnitude prediction (Gonzales et al. 2019), facies classification from post-stack seismic data (Grana et al. 2019), seismic velocity analysis (Fabien-Ouellet and Sarkar, 2020), well log generation (Zhang et al. 2018), well production prediction (Jie et al. 2020), seismic data interpolation (Yoon et al. 2020), porosity estimation from well log data (Chen et al. 2020).

In this work, we use a bidirectional LSTM approach to predicts elastic parameters and litho-fluid classes from pre-stack seismic data. A bidirectional LSTM (Bi-LSTM) is a special class of LSTM that simply puts two independent LSTMs together. This structure allows the networks to have both backward and forward information about the sequence at every time step, this means that at any point in time it preserves information from both past and future. We solve the problem under the assumption

of a local 1D subsurface model, or in other words, we maintain the computational cost affordable by inverting each seismic gathers separately. We train two Bi-LSTMs one to solve the regression problem related to elastic inversion and the other to solve the classification problem. Therefore, one Bi-LSTM is used to predict the mapping between the pre-stack domain and the elastic space, the other is trained to infer the facies model from the seismic data. In both cases, the Bi-LSTM learns the intrinsic vertical dependencies of elastic parameters and facies distribution from the training set. Once the network is fully trained it provides the estimated elastic and facies profile in near real-time. Uncertainty assessment in machine-learning predictions using recurrent neural networks has been discussed in recent publications especially focused on computer science and engineering (Cong and Liang, 2009; Graf et al., 2010), but applications to geophysical inversions is still an open research field. In our work, we combine the Bi-LSTM for the elastic inversion with a Monte Carlo (MC) simulation approach that propagates onto the estimated *Vp*, *Vs*, and density models the uncertainties related to both noise contamination and the so-called modeling error introduced by the Bi-LSTM approximation.

The proposed inversion approach includes four steps: (1) Generation phase: define an ensemble of 1D facies and elastic profiles and compute the associated pre-stack response. In our application, a convolutional forward operator based on the full Zoeppritz equations provides the non-linear mapping between the model and the data space. (2) Network Design: define two Bi-LSTM architectures: one to map the data onto the elastic space, and the other to convert the seismic response into facies. (3) Training phase: train the two networks by minimizing the differences between the predicted and desired output. (4) Use the trained network to invert the seismic data.

The facies profiles forming the training ensemble are generated according to a first-order stationary Markov Chain and the training examples properly model also the expected uncertainties affecting the estimated transition matrices. The generated facies models are also used to define the training for the elastic inversion. In this case, the elastic properties in each facies are distributed

according to a Gaussian prior model determined from well log data. This means that we assume a Gaussian mixture prior to properly take into account the facies dependency of the elastic parameters. The vertical variability of the elastic attributes is modeled according to a vertical Gaussian variogram.

We apply the method to both synthetic and field data. The real data have been acquired onshore to investigate a turbiditic sequence in which sand intervals with different porosities alternate to shales. The elastic and facies predictions provided by the trained Bi-LSTMs are also compared with the outcomes of a more standard two-step (inversion+classification) approach. In this case, we employ the probabilistic linear inversion proposed by Buland and Omre (2003) to infer the elastic properties from the seismic data, then a standard point-wise Bayesian classification infers the facies profile from the maximum-a-posteriori prediction provided by the linear approach. In the synthetic experiments, we also assess the stability of the predictions provided by the Bi-LSTM and the standard approach in case of erroneous assumptions on the noise statistics, on the prior model, and in case of errors in the predicted source wavelet. To the best of the author's knowledge, this is the first application in which Bi-LSTM and MC simulations are used to estimate elastic properties, facies profiles and the associated uncertainties from pre-stack seismic data.

THE METHOD

Recurrent neural and long short-term memory networks

Recurrent neural networks (RNNs) are a class of deep neural networks commonly applied to process sequential data such as text and audio signals. RNNs can take one or more input vectors and produce one or more output vectors that are influenced not just by weights applied to the inputs (like a standard ANN), but also by a hidden state vector that through a feedback loop also depends on previous input and output responses (Medsker and Jain, 2001). In other terms, the same input could produce a different output depending on previous inputs in the series. This is well suited for pre-stack inversion, in which a given seismic amplitude can be generated by different combinations of elastic

properties at the reflecting interface. This fact is responsible for the ill-conditioning of the elastic inversion that is commonly mitigated by taking into account not only the observed seismic amplitudes but also the correlation between the elastic properties at neighboring time samples. In standard inversions, this is usually accomplished by infusing a spatial/temporal variogram model into the inversion kernel.

Let us consider the basic RNN architecture as depicted in Figure 1. The RNN is applied to a time series of length *T*, where the vectors $\mathbf{x} = [x_1, x_2, \dots, x_T]$ and $\mathbf{y} = [y_1, y_2, \dots, y_T]$ denote the input and output responses, respectively. For each time step *t*, we can write:

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h), \quad (1)$$

$$y_t = f\big(W_{hy}h_t + b_y\big), \quad (2)$$

where h represents the hidden state, f is the activation function, b_h and b_y are the trainable bias vectors for the hidden and output state, whereas W_{hh}, W_{xh}, W_{hy} are the trainable weighting matrices associated with the hidden-to-hidden, input-to-hidden, and hidden-to-output connections, respectively. In Figure 1 note that the matrices W_{hh}, W_{xh}, W_{hy} are shared within the RNN unit: this reduces the number of trainable parameters and then the time needed for the learning phase. The internal network parameters are first initialized and then updated during the learning procedure to minimize an error (loss) function that quantifies the difference between the desired and the computed output. The updating process is driven by a back-propagation algorithm:

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

where **P** generically represents the internal network parameters (i.e., bias vectors and weighting matrices), *i* is the iteration number, ε denotes the loss function value, and γ is the so-called learning rate. Well-known drawbacks of standard RNN are the difficulties to learn long-term dependencies that give rise to the so-called vanishing/exploding gradient problems. To avoid this issue several advanced RNN architectures have been developed such as Gated Recurrent Unit (GRU) and Long

Short-Term Memory (LSTM) networks, with LSTM being a generalization of GRU (Pudikov and Brovko, 2020).

LSTMs are specifically devoted to learning long-term features in the data and accomplish this task by introducing a cell state with an update gate, a forget gate, and an output gate (Figure 2). The cell state makes long-term dependency flows through the network, while the other gates add or remove information to the cell state. For example, the input gate discovers which information at the current time step should enter the cell state, while the forget gate determines which information should be forgotten from the previous cell state. More in detail, the value of the cell state at each time step (c_t) depends on the candidate value at the current step ($\tilde{c}_t$), the values at the previous time step (c_{t-1}) and on the output of the forget and input gates at the current step (f_t, i_t, respectively):

$$\tilde{c}_t = tanh(W_c h_{t-1} + W_c x_t + b_c), \quad (4)$$

$$f_t = \sigma\big(W_f h_{t-1} + W_f x_t + b_f\big), \quad (5)$$

$$i_t = \sigma(W_i h_{t-1} + W_i x_t + b_i), \quad (6)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \quad (7)$$

$$o_t = \sigma(W_o h_{t-1} + W_o x_t + b_o), \quad (8)$$

$$h_t = o_t \odot tanh(c_t), \quad (9)$$

$$y_t = W_y h_t, \quad (10)$$

where $tanh$ and σ represent the hyperbolic tangent and sigmoid activation functions, respectively; $\odot$ denotes the Hadamard product, and o_t is the output gate and codes the information to be passed to the hidden state at the next time step h_t; the W matrices and b vectors represent the learnable parameters.

However, if we limit the attention to a 1D model, the seismic data and the elastic properties are temporarily correlated forward and backward in time. For example, due to the filter effect introduced

by the seismic wavelet, a given elastic property at time T influences the seismic response at times $t>T$ as well. Moreover, an elastic property value or a facies at time T is related with all the surrounding samples $T+\Delta t<t<T+\Delta t$, where Δt generically represents a correlation distance. For this reason, to solve the facies classification problem and the elastic inversion we use a network that has an internal memory backward and forward in time. A bidirectional LSTM (Bi-LSTM) is an extension of LSTM that preserves information from both past and future time steps so that the output layer can get information from past (backward) and future (forward) states simultaneously. This is accomplished by connecting two hidden layers of opposite directions to the same output, or in other terms by formulating the hidden state as a concatenated matrix of forward $\vec{h}_t$ and backward $\overleftarrow{h}_t$ hidden states (Figure 3). Both hidden states are considered to compute the output:

$$y_t = \sigma(W_y\vec{h}_t + W_y\overleftarrow{h}_t + b_y). \quad (11)$$

To simplify the notation Bi-LSTM is indicated with the acronym LSTM from here on.

The implemented approach

The elastic inversion aims to map the pre-stack input data (partial stacks at different incidence angles) onto the *Vp*, *Vs*, and density space. In this case, the observed data, that constitutes the input $\mathbf{X}$ to the network, can be written in matrix form as:

$$\mathbf{X} = \begin{bmatrix} d_{t_1,\Theta_1} & d_{t_2,\Theta_1} & \dots & d_{t_T,\Theta_1} \\ d_{t_1,\Theta_2} & d_{t_2,\Theta_2} & \dots & d_{t_2,\Theta_2} \\ \vdots & \vdots & \vdots & \vdots \\ d_{t_1,\Theta_N} & d_{t_2,\Theta_N} & \dots & d_{t_T,\Theta_N} \end{bmatrix}, \quad (12)$$

where d denotes the observed seismic sample, T is the total time length of the traces, and N is the total number of incidence angles considered (usually near, mid, and far angles) that corresponds to the number of input features to the network. The output of the network can still be written as a matrix $\mathbf{Y_e}$ with 3 rows and T columns:

$$\mathbf{Y_e} = \begin{bmatrix} Vp_1 & Vp_2 & \dots & Vp_T \\ Vs_1 & Vs_2 & \dots & Vs_T \\ \rho_1 & \rho_2 & \dots & \rho_T \end{bmatrix}. \qquad (13)$$

For the facies classification, the input is the same as for the elastic inversion but the desired output is a vector $\mathbf{y_f}$ with discrete values representing the facies.

$$\mathbf{y_f} = [f_1 \quad f_2 \quad \dots \quad f_T\ ]. \qquad (14)$$

Both networks try to learn the inverse mapping from the data to the model space from a training set (see the next section). The network architectures used for the elastic inversion and facies classification are very similar: They include an input layer, hidden LSTM layers composed of several LSTM units, a fully connected layer, and an output layer. The two networks only differ in the number of hidden LSTM units used (where this number determines the learning capacity of the network) and in the activation function employed by the fully connected layer: The network for the elastic inversion deals with continuous numbers and hence uses a LeakyReLU activation function with a leakage value of 0.1. Instead, the network for facies classification employs a softmax function that converts logits into probabilities. This means that the output is a vector representing the probability distributions of a list of potential outputs (facies class in our case) at each time sample. Therefore, and different from the network used for elastic inversion, the facies classification also provides an assessment of the uncertainties affecting the predicted facies profile.

For uncertainty estimation in the elastic inversion, we must project onto the parameters space both the noise affecting the data and the modeling error introduced by the network approximation. To this end we adopt a Monte Carlo approach: Let **M** represent the examples forming the validation ensemble, and **N** the ensemble of predicted models output of the LSTM. A sample of the modelings error can be obtained as $\mathbf{E} = \mathbf{M} - \mathbf{N}$ (Hansen and Cordua, 2017)**.** Under a Gaussian assumption, the modeling error can be defined as $\mathcal{N}(0, \mathbf{C}_e)$, where $\mathbf{C}_e$ is the covariance of **E.** This error together with the noise affecting the seismic data $\mathcal{N}(0, \mathbf{C}_n)$ (also assumed to be Gaussian-distributed) are propagated onto the final predictions through an iterative MC approach. Now, let **d** be the vector

expressing the observed data input to the network, whereas q represents the number of MC simulations. The implemented MC approach for uncertainty propagation comprises the following steps:

1) Use the trained network to compute the predicted elastic profile $\mathbf{m}_{\mathrm{b}}$ from the observed data $\mathbf{d}$;
2) Run a forward modeling to compute the noise-free seismic data $\mathbf{d}_{\mathrm{b}}$ associated to $\mathbf{m}_{\mathrm{b}}$;
3) Draw $\mathbf{n}$ from $\mathcal{N}(0, \mathbf{C}_n)$ and compute $\mathbf{d}_{\mathrm{n}} = \mathbf{d}_{\mathrm{b}} + \mathbf{n}$;
4) Use the trained LSTM and $\mathbf{d}_{\mathrm{n}}$ to compute the predicted model $\mathbf{m}_{\mathrm{n}}$;
5) Draw $\mathbf{e}$ from $\mathcal{N}(0, \mathbf{C}_e)$ and compute $\mathbf{m}_{\mathrm{e}} = \mathbf{m}_{\mathrm{n}} + \mathbf{e}$;
6) Store $\mathbf{m}_{\mathrm{e}}$ and repeat from 3) to 6) for q times.

Each generated vector $\mathbf{m}_{\mathrm{e}}$ is an elastic realization in agreement with the observed data, the trained network, and the assumed noise and modeling error distributions. An approximated uncertainty estimation can be numerically derived from the ensemble of q MC simulations. For simplicity, we assume that both the error terms (noise and modeling errors) are Gaussian, but the implemented approach can be applied to whatever parametric or non-parametric error distributions. Note that the previous MC approach is extremely fast because the network predicts a model from the input data (steps 1 and 4) in real-time.

RESULTS

Synthetic inversions

In view of the application of the LSTM approach to two field seismic gathers, we directly train the networks using model and data assumptions appropriate for the study area. The field data have been acquired onshore to investigate a clastic reservoir hosted within a turbiditic sequence in which sand formations with high variable porosity and thickness values intercalate to shales intervals. For this reason, we consider three different litho classes: shale, low porosity sand, and high porosity sand (SH,

LPS, and HPS, respectively). We consider a time interval of 0.4 s for a time sampling of 4 ms: This results in $101 \times 3 = 303$ elastic properties to be estimated from the data.

To generate the training and validation sets we first define the transition matrix determining the facies profile together with the prior probability and the vertical variogram for the elastic properties. Seven wells are available in the area: five of them are used to define the prior assumptions for both the continuous and discrete properties, while the remaining two have been used as blind tests in the field inversion.

To simulate the facies profile, usually, a first-order Markov chain is used in which the facies sequence is drawn according to a transition probability matrix. This means that the element $\mathbf{T}_{i,j}$ at the *i*-th row and *j*-th column of the transition matrix represents the probability of a transition from facies *i* located above the interface to facies *j* located below. Usually, the transition matrix is estimated from available well log data and assumed perfectly known and stationary over the investigated zone during the simulation/inversion process. However, this is a simplification because the transition probability could vary in the reservoir zone, and also the estimated $\mathbf{T}_{i,j}$ values could be also biased by available well log data that do not fully sample all the possible facies configurations. For this reason, in simulating the validation and training sets we take into account the uncertainty related to the estimated facies transition probabilities. We use the five wells to derive five transition probability matrices from which we can define for each $\mathbf{T}_{i,j}$ element the associated mean and standard deviation values. Then under the assumption of a Gaussian distribution for each $\mathbf{T}_{i,j}$, we generate a transition matrix for each simulated model in the training and validation set, under the additional constraint that each row of **T** sums to 1, such that the corresponding conditional probability is a valid probability density function. After sampling the transition matrix, we generate the facies profile of *T* samples through a sequential approach. We first draw the sample at the top of the sequence (f_1) from the prior facies probability (obtained from the stationary distribution associated with the considered transition matrix), hence the subsequent facies realizations at each time sample are drawn from the probability $p(f_t|f_{t-1})$. To

derive the elastic model, we distribute *Vp*, *Vs*, and density values along the simulated facies profile under the assumption of a Gaussian mixture prior and a Gaussian vertical variogram. The Gaussian mixture prior assumes the elastic properties to be Gaussian distributed within each facies, so that the facies dependency of the elastic parameters is taken into account (Figure 4). In Figure 4 we observe that the expected *Vp*, *Vs* and density values decrease as we move from shale to low-porosity sand, and high-porosity sand. If needed whatever type of prior elastic model can be considered (i.e., a non-parametric prior), provided that appropriate geostatistical simulation tools are available to generate elastic realizations. The parameters defining the prior assumptions and the vertical variability of the elastic properties have been determined from the five considered well log data after upscaling (Lindsay and Van Koughnet 2001). The upscaling has been applied to take into account the different resolutions of well log and seismic data. After generating the elastic profile, a convolutional 1D forward modeling based on the full Zoeppritz equations is used to compute the observed seismic data for the three incidence angles of 0, 15, and 30 degrees. To avoid overfitting, in the learning phase we also take into account the noise affecting the field data. To this end we contaminate the noise-free seismic gathers forming both the training and validation examples with Gaussian distributed and uncorrelated noise with zero mean and a covariance matrix estimated from the field dataset. This covariance is inferred from neighboring seismic gathers by measuring the variability of the reflected amplitudes at each time sample, thereby assuming that such variability is only related to noise contamination (Aleardi et al. 2018).

The training and validation sets for the elastic inversion and facies classification are composed of 5000, and 200 prior realizations, respectively, and of associated seismic gathers. The best network configurations (i.e., the number of models forming the training set, the optimal number of hidden layers, the number of hidden units, learning rate) have been determined through a trial-and-error procedure and the final settings have been determined as a compromise between the prediction accuracy on the validation set and the computational effort needed for training. The final net selected for the elastic inversion uses 2 hidden layers composed of 40 units each, while the network for the

facies classification employs 2 hidden layers of 50 units. In both cases, we use a minibatch size of 24, whereas the Adam optimizer (Balles and Hennig, 2018) is employed to update the network internal parameters. For both networks, we set the learning rate to 0.0005, and this value is multiplied by 0.98 every epoch. Before the fully connected layer, a dropout regularization of 10% is also used to tune the networks and avoid overfitting. These network architectures are schematized in Figure 5. The training for the network solving the elastic inversion runs for 100 epochs, while 50 epochs are employed for the facies classification. If we consider Matlab codes running on a common notebook equipped with a quad-core intel i-7 7700HQ CPU@2.80 GHz with 16 Gb RAM, the training and validation sets are generated in 1 minute, approximately, while both networks can be trained in less than 6 minutes. Figure 6a shows the evolution of the root-mean-square error (RMSE) during the training phase of the network used for the elastic inversion. Figure 6b represents the evolution of the accuracy during the learning process for the network used for facies classification. In both cases, we observe that convergence has been achieved within the selected number of epochs, whereas the similar RMSE, and accuracy, on the training and validation sets ensure us that overfitting has been prevented.

The LSTM predictions are compared with those provided by a more standard approach in which the elastic properties are inferred from partial angle stacks through a linear inversion (Buland and Omre, 2003) under the assumption of log-Gaussian distributed elastic properties. The log-Gaussian prior model has been derived from the five considered wells, while the vertical constraint infused into the linear inversion is expressed by the same variogram model used to generate the training and validation sets. Then, the maximum-a-posteriori (MAP) solution provided by the linear inversion is the input for a point-wise Bayesian classification in which the prior facies model is derived from the average stationary distribution associated with the transition probability matrices derived from the five wells. We compare the LSTM and standard approach on test models and data simulating different possible scenarios:

- Test 1: The prior model assumptions, the noise statistics, and the source wavelet used to derive the test model and the associated seismic response are the same assumed in the learning phase. This is the optimal situation in which the actual subsurface model and the observed data are in agreement with the prior assumptions.
- Test 2: Similar to Test 1 but now the variance of the noise is twice that assumed in the learning phase but the underlying statistical distribution (uncorrelated Gaussian model) still holds. In this case, we are simulating an underestimation of the noise in the data.
- Test 3: Now we contaminate the data with both correlated and uncorrelated Gaussian noise with a variance twice that assumed in the learning stage. The temporal variogram of the correlated noise is equal to that assumed for the elastic parameters. Now, we are simulating residuals of coherent noise in the data that in field applications can be ascribed to multiple reflections or diffractions not correctly removed during the processing phase.
- Test 4: Similar to Test 3 but now we also simulate an erroneous estimation of the peak frequency and phase of the angle-dependent source wavelet. The errors increase as the incidence angles increases (Figure 7).
- Test 5: Similar to Test 4 but now we also simulate errors in the assumed prior elastic and facies models. To this end, the facies test model has been drawn from a transition matrix that results in a stationary distribution with an increased 10% of the actual shale probability. Similarly, the elastic test model has been drawn from facies-dependent model covariance matrices with 10% of errors with respect to the model covariances used to derive the training examples.

For the LSTM and linear inversion, we assess the quality of the predictions by measuring the percentage difference between predicted and actual elastic properties, and between observed and predicted data. For the LSTM elastic inversion, we also run 1000 MC simulations to compute the 95 % posterior confidence intervals. Confusion matrices, precision, and recall are used to quantitatively compare the facies classification results. The recall quantifies the percentage of samples belonging

to a given class that have been correctly classified. The precision represents the percentage of samples classified in a given facies that actually belong to that facies.

Figure 8 represents the inversion and classification results for Test 1. From a first visual analysis of Figure 8a, we note that both approaches provide elastic predictions that closely follow the vertical variability of the true properties and that satisfactorily predict the actual elastic contrasts at the reflecting interfaces. However, the overall quality of the predictions decreases moving from *Vp*, *Vs*, and density. This is expected since *Vp* and density are the parameters exerting the largest and smallest influence on observed seismic data, respectively. However, we also note that such decreasing in the accuracy is more significant for the standard approach. Both inversion results generate predicted data in close agreement with the observed seismic response. Figure 8b illustrates that the LSTM approach provides more accurate facies classification results in which all the main HPS and LPS intervals are correctly located. The posterior facies probability illustrates that the LSTM usually recovers the correct facies with lower uncertainties than the Bayesian method. Only the high porosity sand located at 0.142 s is misclassified by the LSTM probably because the thickness of this layer is far below the seismic resolution. The Bayesian classification overestimates the occurrence of low porosity sands and erroneously locates an HPS sand around 0.39 s. This qualitative analysis is confirmed by the confusion matrices shown in Figure 9: The LSTM guarantees more precise classification results and, for the SH and HPS, also higher recall values. For the Bayesian classification, the significant overestimation of LPS results in recall values higher than for the LSTM.

Figures 10-17 show the results for Tests 2-5. As expected, erroneous assumptions on the noise statistics, errors in the source wavelet, and errors in the assumed prior model, decrease the overall quality of the predictions. A visual inspection of these results confirms that: both approaches provide estimated models that reproduce well the observed data; the LSTM elastic predictions are closer to the actual property values; the Bayesian classification tends to overestimate the occurrence of LPD and underpredicts the occurrence of HPS; The LSTM recovers more realistic facies profiles in which

the actual layer thickness and depositional trends are better preserved; The combined presence of errors in the wavelet estimation and coherent noise in the data significantly deteriorate the quality of the retrieved elastic and facies models; In all of the tests and for both approaches, erroneous assumptions about the source wavelet and noise particularly affect the density estimates, which is the parameter least informed by the data. The quality of the predictions yielded by the standard approach is particularly low in the most challenging scenario (Tests 5; Figure 16), thus demonstrating the importance of accurate prior assumptions. In all the tests the LSTM outperforms the standard approach. Notably, for all the examples the actual elastic properties are enclosed by the 95 % confidence intervals as estimated from 1000 MC simulations.

For a more quantitative analysis of the previous results we represent in Table 1, and for all the tests, the percentage difference between actual and predicted elastic parameters and predicted and observed data. In all cases, it can be noticed that the accuracy of the elastic inversion decreases moving from Tests 1 to Test 5, but this decreasing accuracy is more significant for the linear inversion. This means that the LSTM predictions are less affected by erroneous assumptions in the statistical noise properties, in the prior model, and also less influenced by errors in the estimated source wavelet, and by unmodelled coherent noise in the data. For Test 1 we observe that the predicted data generated on the MAP linear solution are slightly closer to the observed seismic than the data predicted by the LSTM. This can be explained by taking into consideration that the linear inverse solution corresponds to the maximum of a posterior probability density or in other terms to a minimum of an error function. Differently, the LSTM predictions are not driven by a minimization process, but they are simply obtained by applying the previously trained network to the observed data. For the other tests, we observe that the data mismatch for the linear inversion is significantly lower than for the LSTM inversion. This is probably related to the underprediction of the noise variance and the consequent data overfitting. In other words, for underpredicted noise, the linear inversion is too confident on the quality of the observed seismic gathers. This could indicate that the LSTM is less prone to overfitting the data in case of errors in the assumed noise variance.

The poorer predictions provided by the linear inversion can also be ascribed to the simple Gaussian prior used, which overlooks the facies dependency of the elastic properties and thus oversimplifies the actual distribution of the elastic properties in the subsurface. For what concerns the facies classification, the main limitation of the Bayesian inversion is that it overlooks the vertical correlation of the facies model and classifies each sample independently.

Field data application

We now discuss the results obtained on the two field seismic gathers used as blind tests. The upsampled collocated well log data (hereafter called Well 1 and Well 2) have been used to validate the predictions of the LSTM and standard approach. The true facies profiles have been inferred from the derived petrophysical property curves (not shown here) by imposing appropriate cut-off values (for example a porosity lower than 5% and a clay content higher than 70% define the shale facies). The low-frequency elastic model has been derived from geostatistical interpolation of the available well log data.

The visual inspection of Figures 18-21 illustrates that the overall quality of the results is often lower compared to the previous synthetic examples, although the major elastic contrasts have been correctly predicted. For both seismic gathers, the LSTM outperforms the standard approach as it provides elastic predictions and facies classification usually closer to the actual model, especially for Well 1, while the predictions are more similar for Well 2. In particular, the LSTM correctly locates the main sand layers, while the Bayesian classification still overpredicts the occurrence of low-porosity sand. For Well 1 the thick HPS layer below 0.5 s is interpreted as LPS by the Bayesian approach, while it is correctly classified by the LSTM. For Well 1, the increasing mismatch between the predicted and actual elastic properties below 0.15 s is probably related to coherent noise in the data that cannot be properly modeled by the assumed diagonal data covariance matrix. However, more importantly, for the LSTM approach, the true property values always lie within the 95 % confidence interval

numerically derived from 1000 MC simulations. Table 2 summarizes the field inversion results. It emerges that the LSTM provides more accurate elastic property predictions, while similar to the synthetic test the linear inversion yields final estimates that better reproduce the observed data.

DISCUSSION

Estimation of elastic properties and litho-fluid facies is the primary goal of pre-stack seismic inversion. In this work, we used bilinear LSTMs to solve this mixed discrete-continuous inverse problem. We trained one network to solve the elastic inversion and the other to solve the classification problem, under the assumption of a local 1D subsurface model, and Gaussian-mixture distributed elastic parameters. Standard approaches to elastic inversions must be adapted and reformulated when the prior assumptions or the relation linking the model to the data change (i.e., from linear to non-linear forward operator). Differently, the LSTM inversion can be directly applied to any type of prior elastic model, and forward equation. The implemented method does not require the regularization in its common-sense meaning (i.e., the inclusion of model constraints or prior information into the inversion framework), but the network is trained on a data set containing realistic subsurface scenarios and thus learns how to reproduce a similar model that fits the observed data. Moreover, instead of minimizing an error function or sampling from a posterior probability distribution, the proposed approach employs trained LSTMs to reconstruct the elastic parameters and litho-fluid classes from observed pre-stack seismic gathers.

The generation phase requires a very limited computational effort, and the proposed approach also requires a relatively small data set for training, thus meaning that different network architectures and hyperparameter settings can be evaluated rapidly. This reduces not only the effort required for the network configuration but also makes applications of the approach to different exploration areas with different prior assumptions computationally affordable. The LSTM elastic inversion was combined with a subsequent Monte Carlo simulation for uncertainty appraisals, in which both the noise affecting

the observed data and the modeling error associated with the network approximation are taken into account. Once trained, the networks estimate elastic properties, litho-fluid classes, and associated uncertainties from the data in real-time.

Since the computational cost to generate the training set and to train the network is very limited (a few minutes on a common notebook), we suggest retraining the network from scratch when the LSTM inversion is applied to different scenarios. This is similar to what is commonly done in Bayesian inversion in which the prior must be reestimated for different exploration areas. Another possible approach could be applying a transfer learning strategy in which a small portion of new training examples is employed to adjust the internal network parameters when the prior of the target differs from that assumed during the learning stage (Aleardi and Salusti 2021). However, this method is mainly recommended when the generation and training phases are computationally expensive so it might be not appropriate for the considered application. In any case and similar to any Bayesian inversion, a robust prior model that exhaustively captures the actual distribution of the elastic properties in the study area is a crucial ingredient of the proposed inversion strategy. Other tests we performed (not shown here for brevity) that simulate more challenging scenarios with highly overlapped elastic properties within the different facies, confirmed the robustness and the applicability of the presented LSTM approach.

CONCLUSIONS

We trained two bidirectional long short-term memory networks to infer elastic properties and litho-fluid classes from pre-stack data. A Monte Carlo simulation was also implemented to estimate model uncertainties in the elastic inversion. Synthetic and field data experiments showed promising results and demonstrated that LSTM networks can effectively approximate the inverse of nonlinear operators that project the observed data onto the discrete and continuous parameter space. The trained networks achieved satisfactory predictions when realistic conditions were simulated (e.g., errors in the

estimated source wavelet or erroneous assumptions about the noise and model statistics). In all the tests the LSTM usually provided more accurate predictions than the standard, two-step approach to elastic inversion and facies classification. Notably, the quality of the LSTM predictions is less affected by errors in the assumed noise and prior model, and by errors in the estimated source wavelet.

Conflict of interest

The author declares no conflict of interest.

Data Availability

Codes and synthetic data available on request from the authors.

REFERENCES

Aleardi, M., and Ciabarri, F. (2017). Application of different classification methods for litho-fluid facies prediction: A case study from the offshore Nile Delta. Journal of Geophysics and Engineering, 14(5), 1087-1102.

Aleardi, M., Ciabarri, 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. (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., and Salusti, A. (2020a). Markov chain Monte Carlo algorithms for target-oriented and interval-oriented amplitude versus angle inversions with non-parametric priors and non-linear forward modellings. Geophysical Prospecting, 68(3), 735-760.

Aleardi, M., and Salusti, A. (2020b). Elastic pre-stack seismic inversion through Discrete Cosine Transform reparameterization and Convolutional Neural Networks. Geophysics, 86(1), 1-18.

Araya-Polo, M., Jennings, J., Adler, A., and Dahlke, T. (2018). Deep-learning tomography. The Leading Edge, 37(1), 58-66.

Azevedo, L., and A. Soares, (2017). Geostatistical methods for reservoir geophysics. Springer.

Avseth, P., T. Mukerji, and G. Mavko, (2005). Quantitative seismic interpretation: Cambridge University Press.

Bachrach, R., and Gofer, E. (2019). Non-Linear Anisotropic Litho-Petro-Elastic AVA Inversion: Example from Unconventional Play. In 81st EAGE Conference and Exhibition 2019. European Association of Geoscientists & Engineers, 2019(1), 1-5.

Balles, L., and Hennig, P. (2018). Dissecting adam: The sign, magnitude and variance of stochastic gradients. In International Conference on Machine Learning, 404-413.

Bishop, C. M. (2006). Pattern recognition and machine learning. springer.

Bongajum, E. L., Boisvert, J., and Sacchi, M. D. (2013). Bayesian linearized seismic inversion with locally varying spatial anisotropy. Journal of Applied Geophysics, 88, 31-41.

Bosch, M., Bertorelli, G., Álvarez, G., Moreno, A., and Colmenares, R. (2015). Reservoir uncertainty description via petrophysical inversion of seismic data. The Leading Edge, 34(9), 1018-1026.

Buland, A., and Omre, H. (2003). Bayesian linearized AVO inversion. Geophysics, 68(1), 185-198.

Cai, Y., Shyu, M. L., Tu, Y. X., Teng, Y. T., and Hu, X. X. (2019). Anomaly detection of earthquake precursor data using long short-term memory networks. Applied Geophysics, 1-10.

Chen, W., Yang, L., Zha, B., Zhang, M., and Chen, Y. (2020). Deep learning reservoir porosity prediction based on multilayer long short-term memory network. Geophysics, 85(4), 1-69.

Cong, S., and Liang, Y. (2009). PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems. IEEE Transactions on Industrial Electronics, 56(10), 3872-3879.

Connolly, P. A., and Hughes, M. J. (2016). Stochastic inversion by matching to large numbers of pseudo-wells. Geophysics, 81(2), M7-M22.

Da Veiga, S., and Le Ravalec, M. (2012). Maximum likelihood classification for facies inference from reservoir attributes. Computational Geosciences, 16(3), 709-722.

de Figueiredo, L. P., Grana, D., Bordignon, F. L., Santos, M., Roisenberg, M., and Rodrigues, B. B. (2018). Joint Bayesian inversion based on rock-physics prior modeling for the estimation of spatially correlated reservoir properties. Geophysics, 83(5), M49-M61.

de Figueiredo, L. P., Grana, D., Roisenberg, M., and Rodrigues, B. B. (2019). Multimodal Markov chain Monte Carlo method for nonlinear petrophysical seismic inversion. Geophysics, 84(5), M1-M13.

Doyen, P. M., (2007). Seismic reservoir characterization. EAGE.

Dubois, M. K., Bohling, G. C., and Chakrabarti, S. (2007). Comparison of four approaches to a rock facies classification problem. Computers & Geosciences, 33(5), 599-617.

Fabien-Ouellet, G., and Sarkar, R. (2020). Seismic velocity estimation: A deep recurrent neural-network approach. Geophysics, 85(1), U21-U29.

Fjeldstad, T., and Omre, H. (2017). Bayesian inversion of convolved hidden Markov models with applications in reservoir prediction. arXiv:1710.06613

Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media.

Goodfelow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT press.

González, J., Yu, W., and Telesca, L. (2019). Earthquake Magnitude Prediction Using Recurrent Neural Networks. In Multidisciplinary Digital Publishing Institute Proceedings (Vol. 24, No. 1, p. 22).

Graf, W., Freitag, S., Kaliske, M., and Sickert, J. U. (2010). Recurrent neural networks for uncertain time-dependent structural behavior. Computer-Aided Civil and Infrastructure Engineering, 25(5), 322-323.

Grana, D., Mukerji, T., Dvorkin, J., and Mavko, G. (2012). Stochastic inversion of facies from seismic data based on sequential simulations and probability perturbation method. Geophysics, 77(4), M53-M72.

Grana, D., Azevedo, L., and Liu, M. (2020). A comparison of deep machine learning and Monte Carlo methods for facies classification from seismic data. Geophysics, 85(4), WA41-WA52.

Graves, A. (2012). Long short-term memory. In Supervised sequence labelling with recurrent neural networks. Springer, Berlin, Heidelberg.

Hall, B., (2016). Facies classification using machine learning: The Leading Edge, 35, 906–909,

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.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Kemper, M., and Gunning, J. (2014). Joint impedance and facies inversion–seismic inversion redefined. First Break, 32(9).

Kuyuk, H. S., and Susumu, O. (2018). Real-Time Classification of Earthquake using Deep Learning. Procedia Computer Science, 140, 298-305.

Jie, L., Junxing, C., and Jiachun, Y. (2020). Prediction on daily gas production of single well based on LSTM. In SEG 2019 Workshop: Mathematical Geophysics: Traditional vs Learning, 41-43.

Larsen, A. L., Ulvmoen, M., Omre, H., and Buland, A. (2006). Bayesian lithology/fluid prediction and simulation on the basis of a Markov-chain prior model. Geophysics, 71(5), R69-R78.

Lewis, W., and Vigh, D. (2017). Deep learning prior models from seismic images for full-waveform inversion. In SEG Technical Program Expanded Abstracts, 1512-1517. Society of Exploration Geophysicists.

Lindberg, D. V., and Grana, D. (2015). Petro-elastic log-facies classification using the expectation–maximization algorithm and hidden Markov models. Mathematical Geosciences, 47(6), 719-752.

Lindsay, R., and Van Koughnet, R. (2001). Sequential Backus averaging: Upscaling well logs to seismic wavelengths. The Leading Edge, 20(2), 188-191.

Martinez, W. L., and Martinez, A. (2007). Computational statistics handbook with Matlab: Chapman & Hall.

Medsker, L. R., and Jain, L. C. (2001). Recurrent neural networks. Design and Applications, 5, 64-67.

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.

Negahdari, A., Ziaii, M., and Ghiasi-Freez, J. (2014). Application of discriminant analysis for studying the source rock potential of probable formations in the Lorestan Basin, Iran. International Journal of Mining and Geo-Engineering, 48(1), 31-54.

Park, M. J., and Sacchi, M. D. (2020). Automatic velocity analysis using Convolutional Neural Network and Transfer learning. Geophysics, 85(1), V33-V43.

Pudikov, A., and Brovko, A. (2020). Comparison of LSTM and GRU Recurrent Neural Network Architectures. In International Scientific and Practical Conference in Control Engineering and Decision Making, Springer, 114-124.

Puzyrev, V. (2019). Deep learning electromagnetic inversion with convolutional neural networks. Geophysical Journal International, 218(2), 817-832.

Richardson, A. (2018). Seismic full-waveform inversion using deep learning tools and techniques. arXiv preprint:1801.07232.

Rimstad, K., Avseth, P., and Omre, H. (2012). Hierarchical Bayesian lithology/fluid prediction: A North Sea case study. Geophysics, 77(2), B69-B85.

Rimstad, K., and Omre, H. (2010). Impact of rock-physics depth trends and Markov random fields on hierarchical Bayesian lithology/fluid prediction. Geophysics, 75(4), R93-R108.

Talarico, E. C. E. S., Grana, D., Passos de Figueiredo, L., and Pesco, S. (2020). Uncertainty quantification in seismic facies inversion. Geophysics, 85(4), M43-M56.

Tarantola, A. (2005). Inverse problem theory and methods for model parameter estimation. Society for Industrial and Applied Mathematics.

Ulvmoen, M., and Omre, H. (2010). Improved resolution in Bayesian lithology/fluid inversion from prestack seismic data and well observations: Part 1—Methodology. Geophysics, 75(2), R21-R35.

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.

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, G., Carr, T. R., Ju, Y., and Li, C. (2014). Identifying organic-rich Marcellus Shale lithofacies by support vector machine classifier in the Appalachian basin. Computers & Geosciences, 64, 52-60.

Wang, B., Zhang, N., Lu, W., and Wang, J. (2019). Deep-learning-based seismic data interpolation: A preliminary result. Geophysics, 84(1), V11-V20.

Wu, Y., and McMechan, G. A. (2019). Parametric convolutional neural network-domain full-waveform inversion. Geophysics, 84(6), R881-R896.

Yoon, D., Yeeh, Z., and Byun, J. (2020). Seismic Data Reconstruction Using Deep Bidirectional Long Short-Term Memory With Skip Connections. IEEE Geoscience and Remote Sensing Letters. In press. doi: 10.1109/LGRS.2020.2993847.

Zhang, D., Yuntian, C. H. E. N., and Jin, M. E. N. G. (2018). Synthetic well logs generation via Recurrent Neural Networks. Petroleum Exploration and Development, 45(4), 629-639.

Zunino, A., Mosegaard, K., Lange, K., Melnikova, Y., and Mejer Hansen, T. (2015). Monte Carlo reservoir analysis combining seismic reflection data and informed priors. Geophysics, 80(1), R31-R41.

FIGURES & TABLES

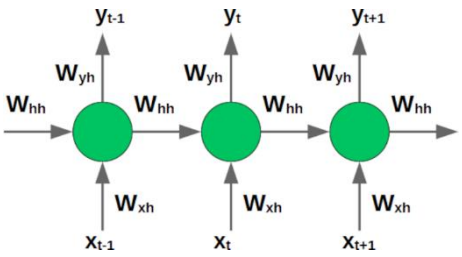

Figure 1: Standard RNN unit.

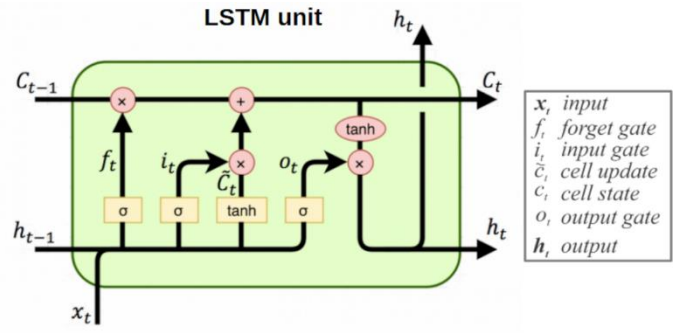

Figure 2: Standard LSTM unit.

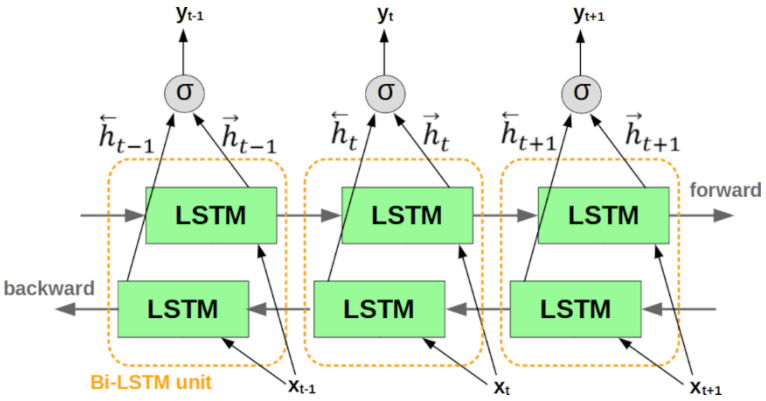

Figure 3: Architecture of bidirectional LSTM.

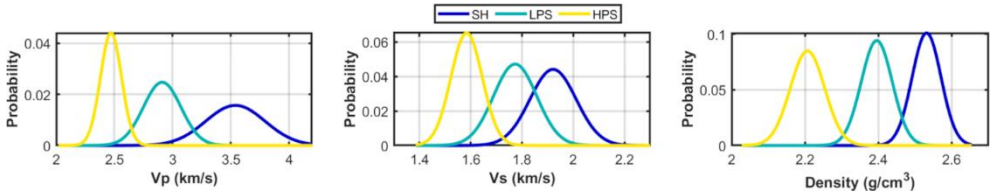

Figure 4: Facies-dependent Gaussian components of the elastic prior model for Vp, Vs, and density.

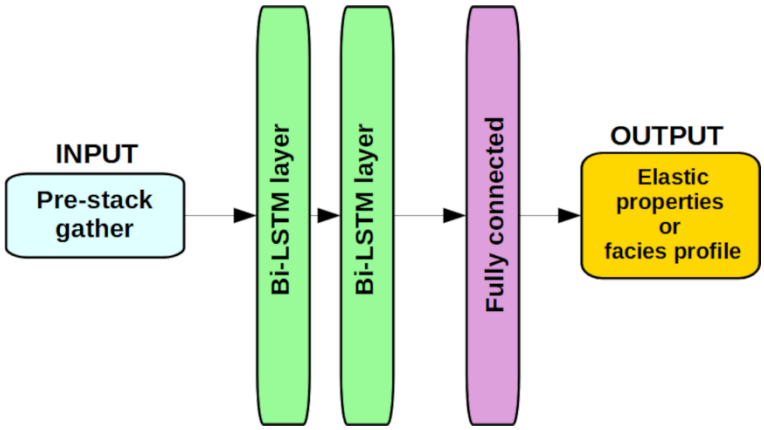

Figure 5: Schematization of the network architectures.

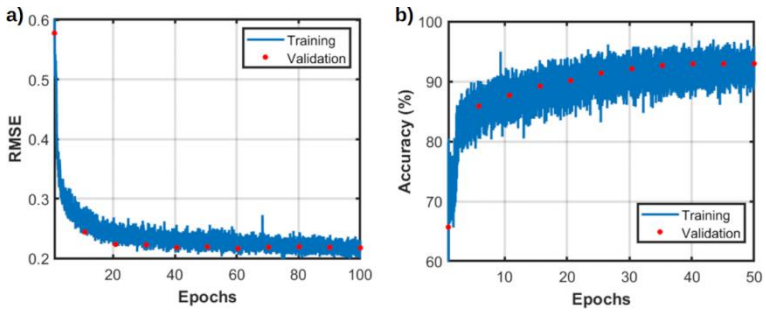

Figure 6: a) RMSE evolution during the training phase of the network used for the elastic inversion. b) Evolution of the accuracy for the network used for facies classification.

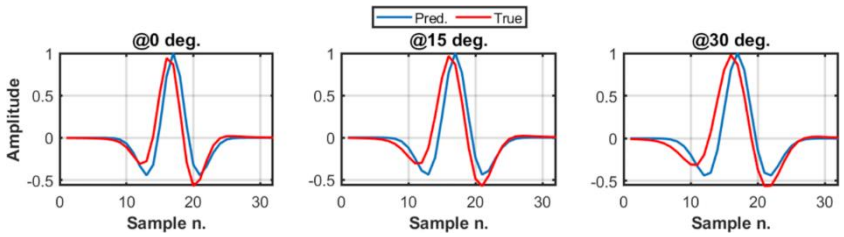

Figure 7: Comparison between the angle-dependent source wavelets assumed in the learning phase (blue curve) and the wavelets used to compute the observed data in Tests 4 and 5 (red curve). We simulate errors in the peak frequency and phase that increase with the incidence angle.

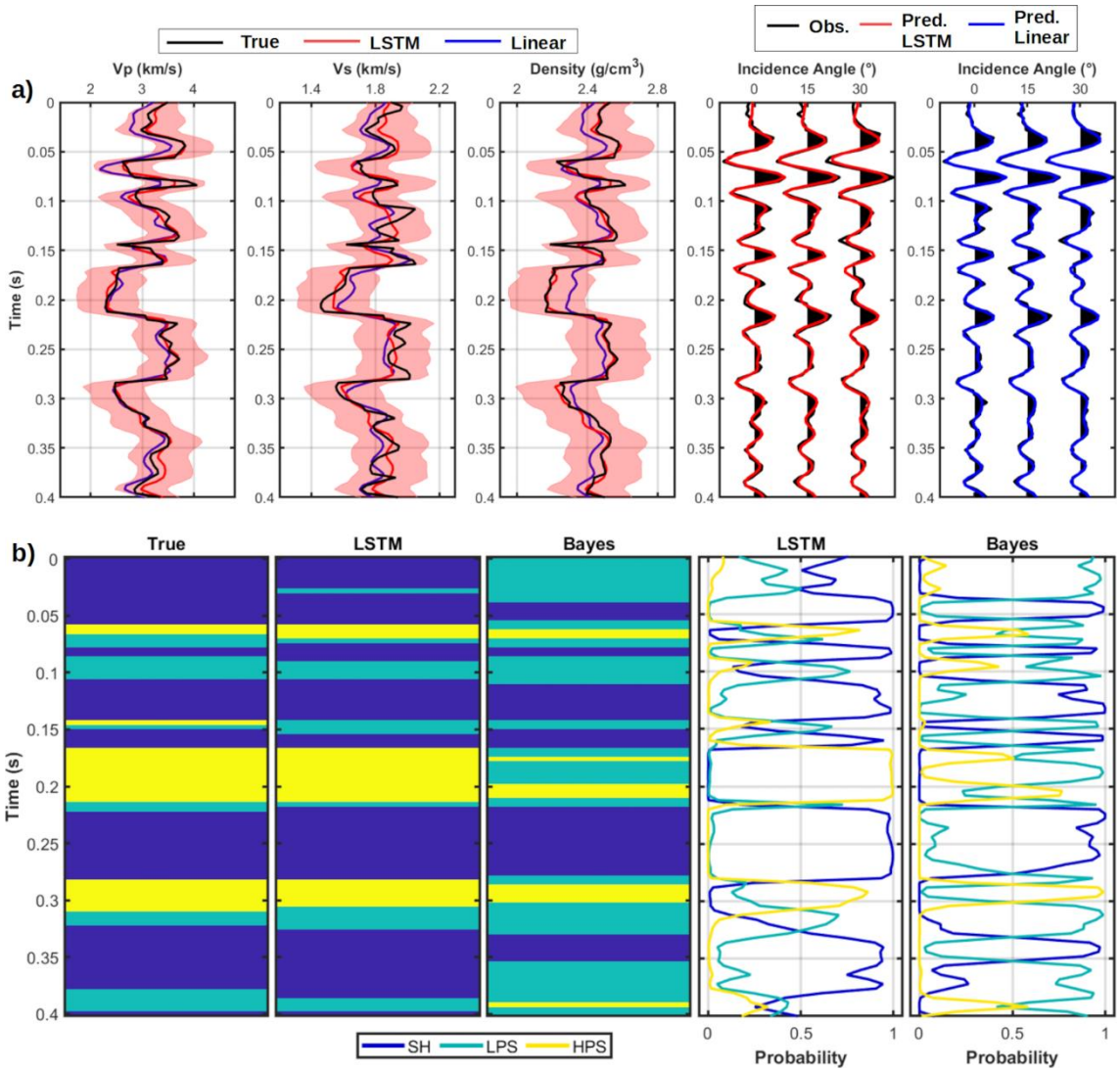

Figure 8: Results for Test 1. a) Comparison between the true elastic profile and the elastic parameters estimated by the LSTM and linear inversion (MAP solution). The shaded red area depicts the 95% confidence interval associated with the LSTM estimates and provided by 1000 MC simulations. The predicted data have been computed on the LSTM predictions and on the MAP solution provided by the linear approach. b) Facies classification results and the facies probability provided by the LSTM and Bayesian classification. The outcomes of the Bayesian classification has been obtained from the MAP solution estimated by the linear inversion.

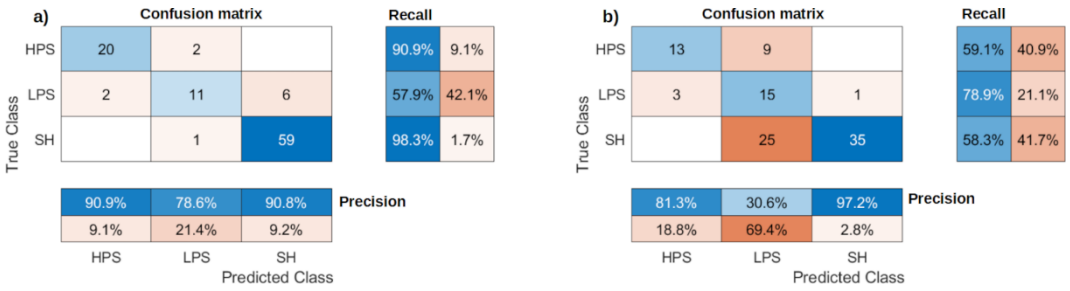

Figure 9: Confusion matrices, precision, and recall associated with the LSTM (a) and Bayesian (b) classifications for Test 1. The values in the confusion matrices indicate the number of samples correctly classified (main diagonal) or those misclassified (off-diagonal terms).

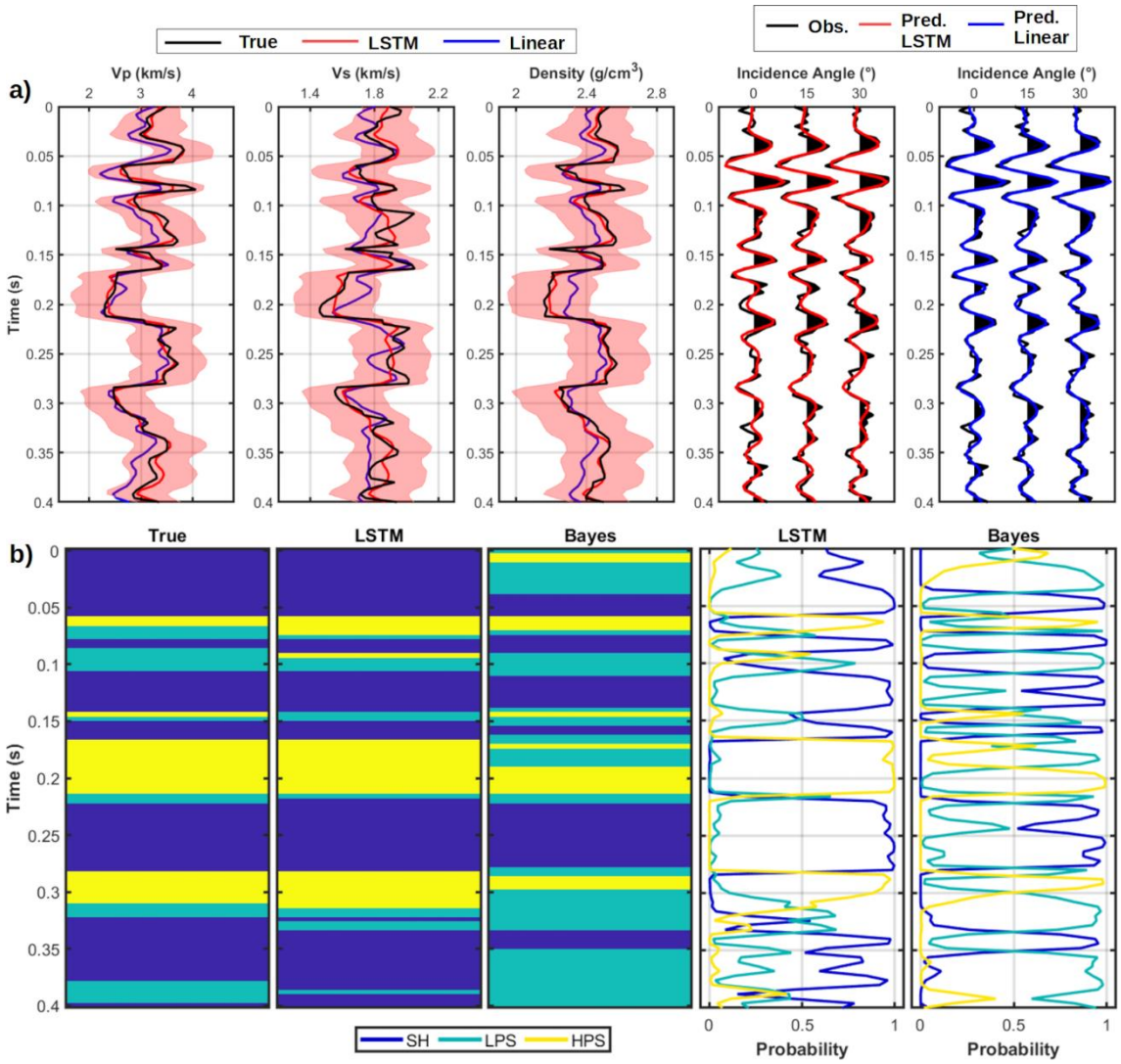

Figure 10: As in Figure 8 but for Test 2.

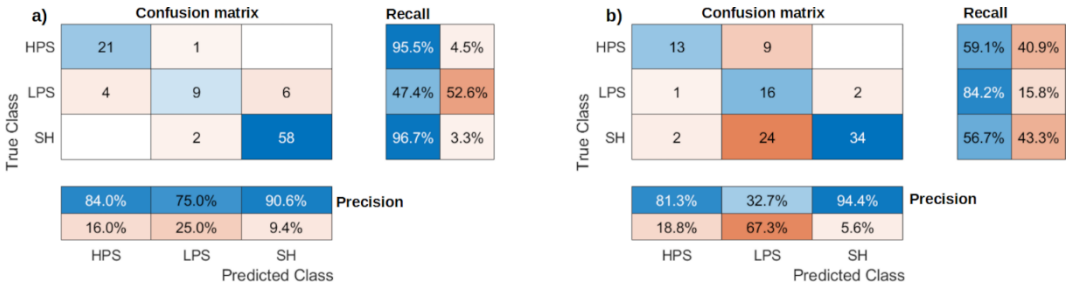

Figure 11: As in Figure 9 but for Test 2.

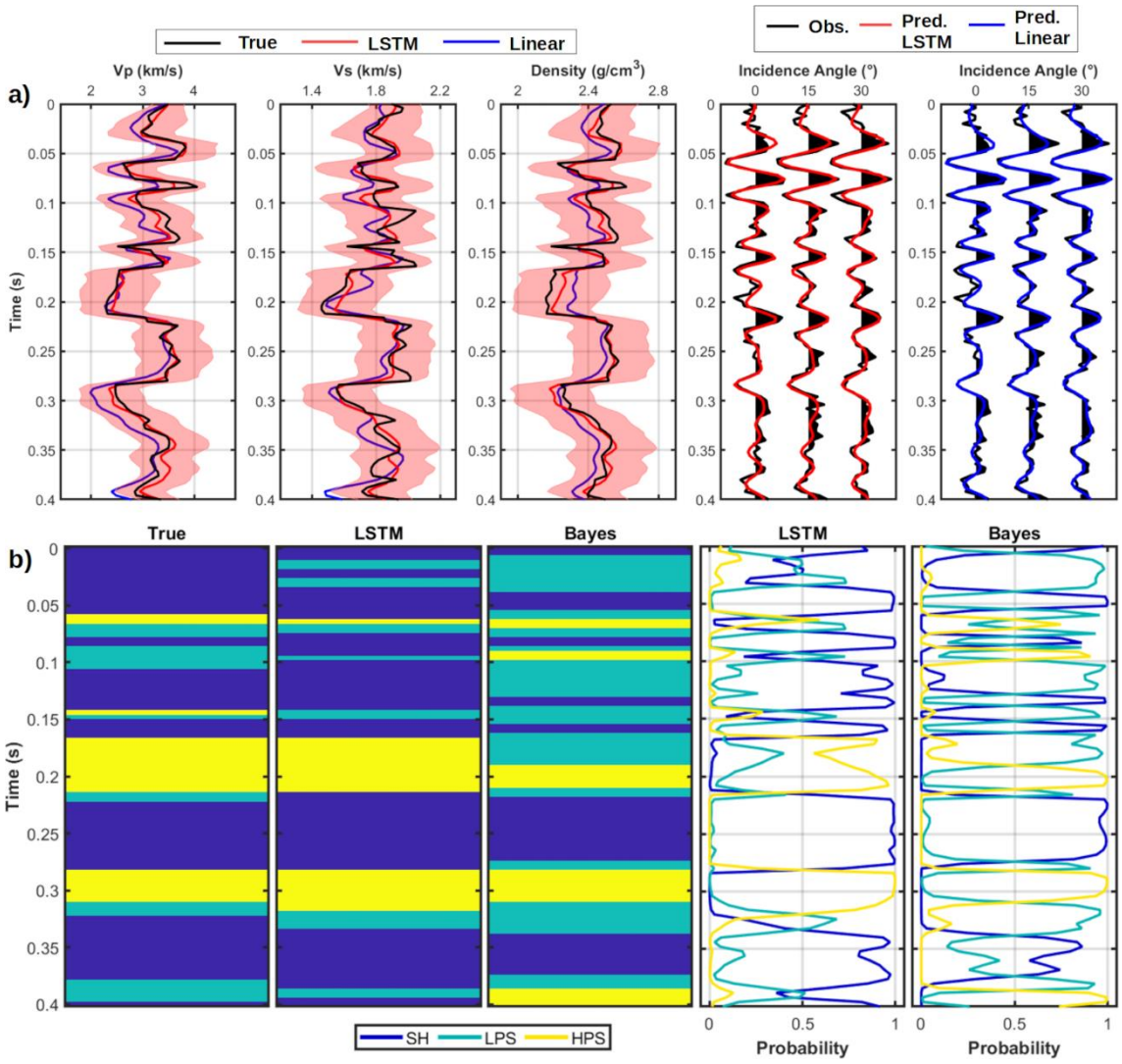

Figure 12: As in Figure 8 but for Test 3.

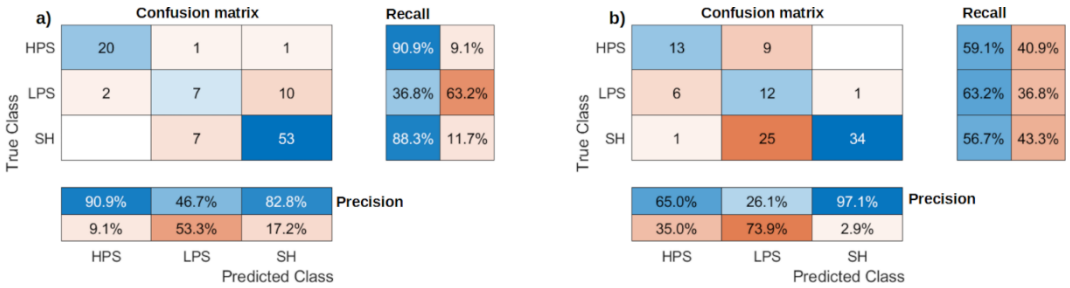

Figure 13: As in Figure 9 but for Test 3.

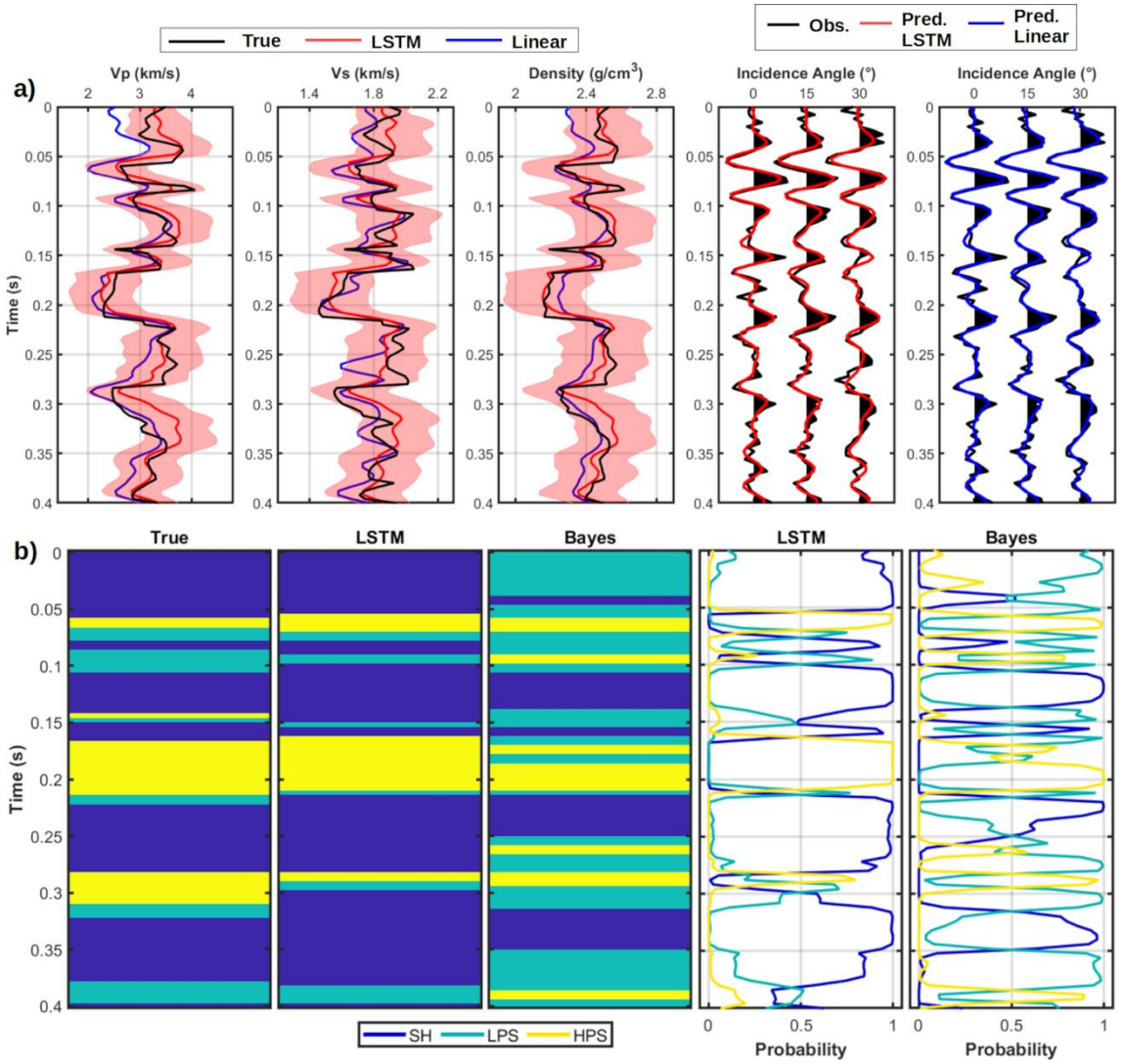

Figure 14: As in Figure 8 but for Test 4.

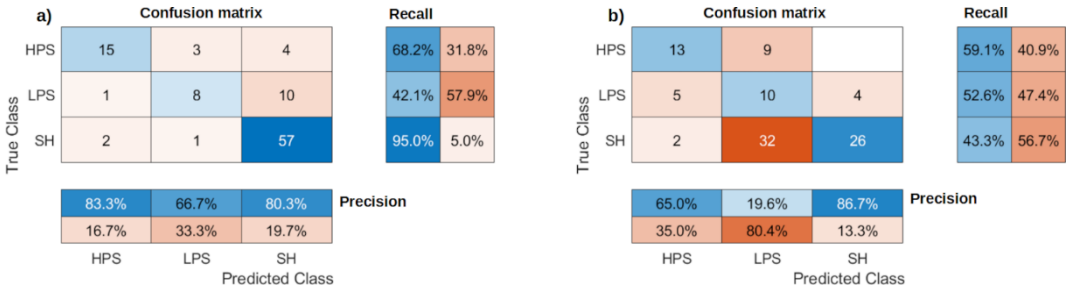

Figure 15: As in Figure 9 but for Test 4.

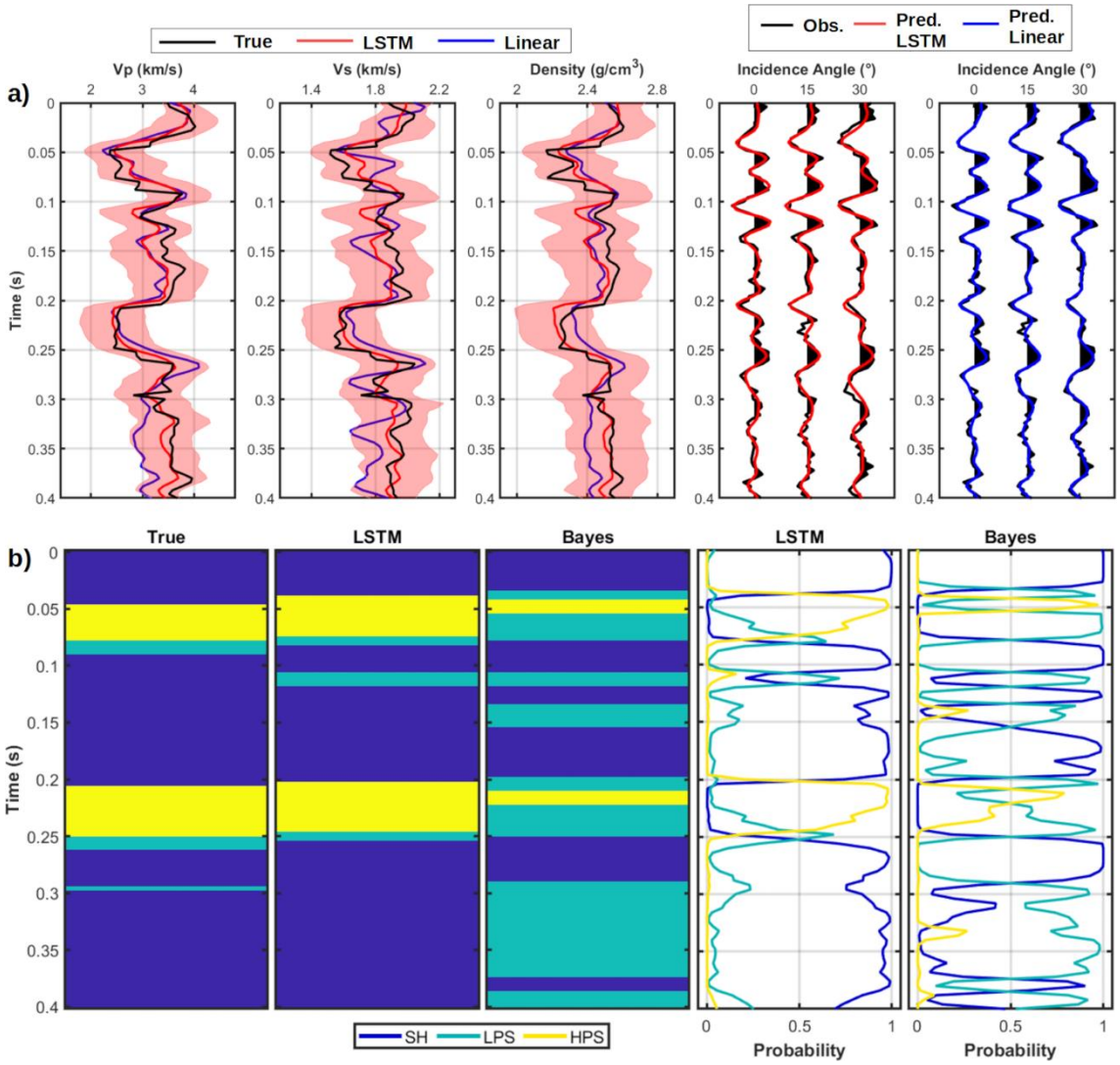

Figure 16: As in Figure 8 but for Test 5.

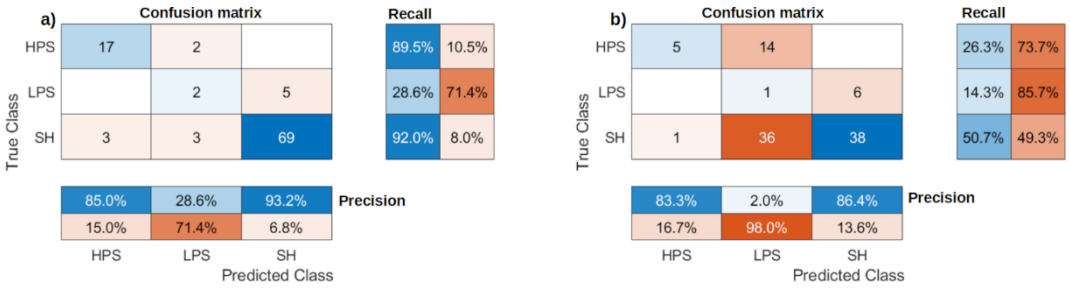

Figure 17: As in Figure 9 but for Test 5.

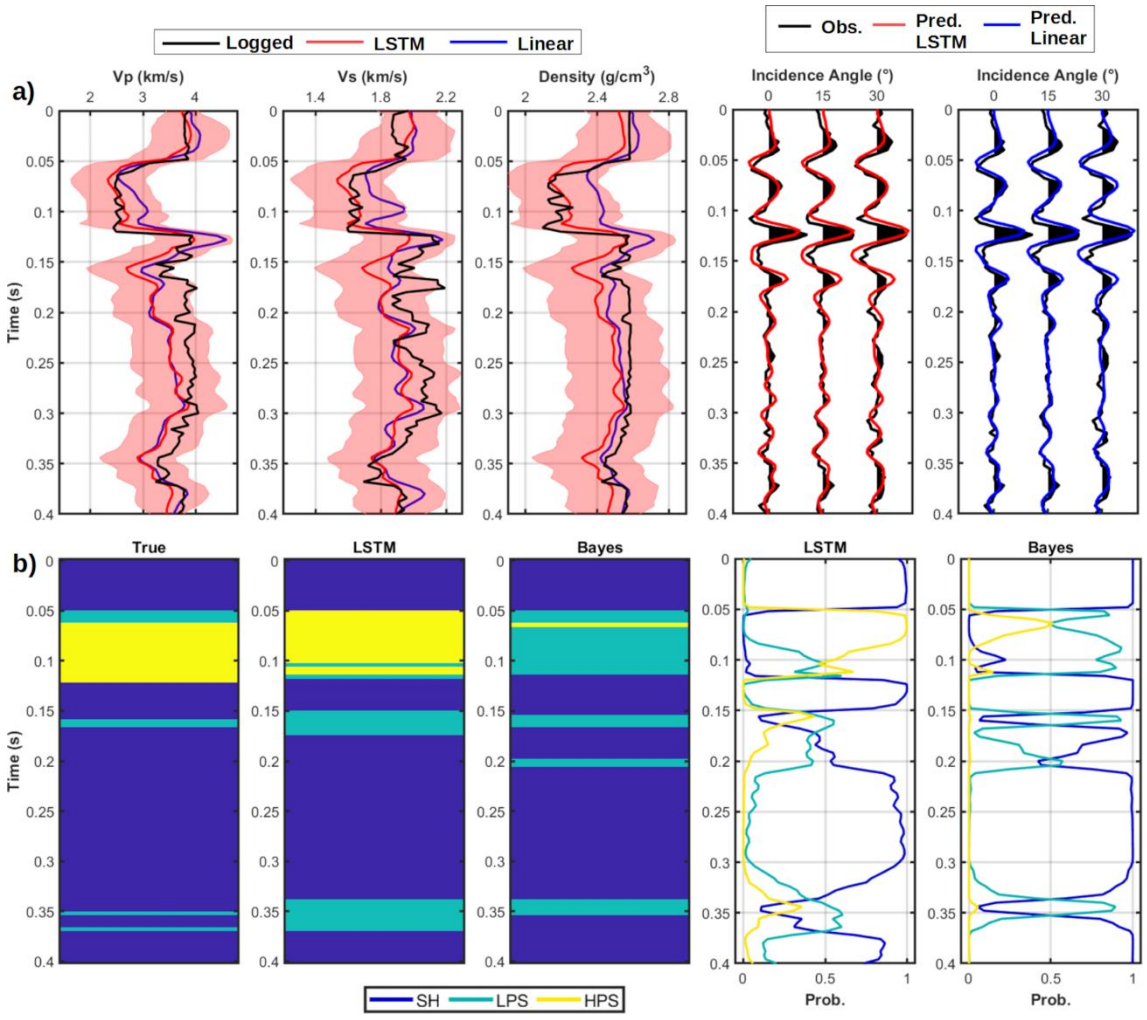

Figure 18: Results for Well 1. a) Comparison between the logged elastic properties and the elastic parameters estimated by the LSTM and linear inversion (MAP solution). The shaded red area depicts the 95% confidence interval associated with the LSTM estimates and provided by 1000 MC simulations. The predicted data have been computed on the LSTM predictions and on the MAP solution provided by the linear approach. b) Facies classification results and the facies probability provided by the LSTM and Bayesian classification. The outcomes of the Bayesian classification has been obtained from the MAP solution estimated by the linear inversion.

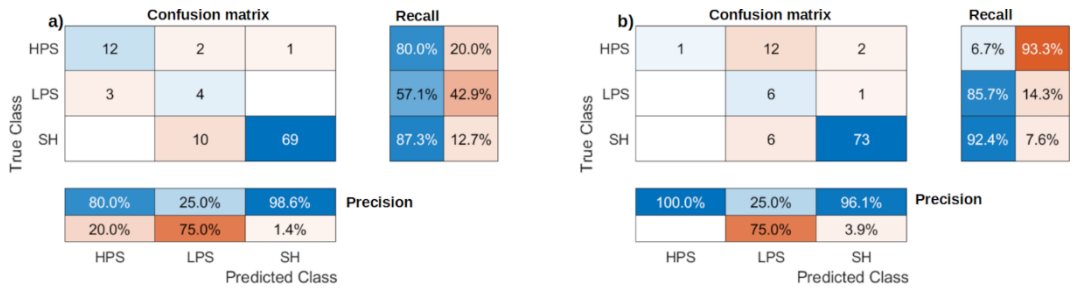

Figure 19: Confusion matrices, precision, and recall associated with the LSTM and Bayesian classification results for Well 1. The values in the confusion matrices indicate the number of samples correctly classified (main diagonal) or those misclassified (off-diagonal terms).

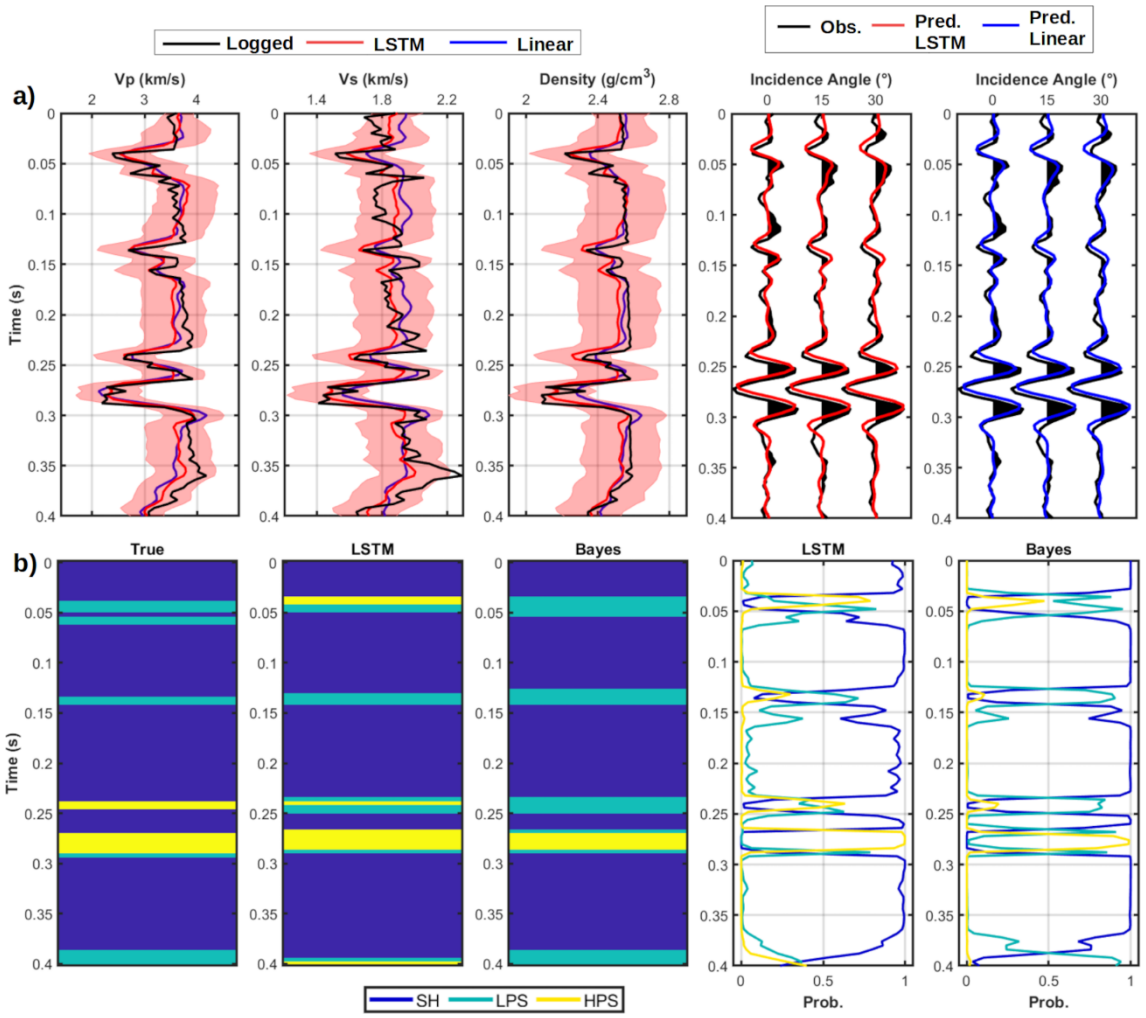

Figure 20: As in Figure 18 but for Well 2.

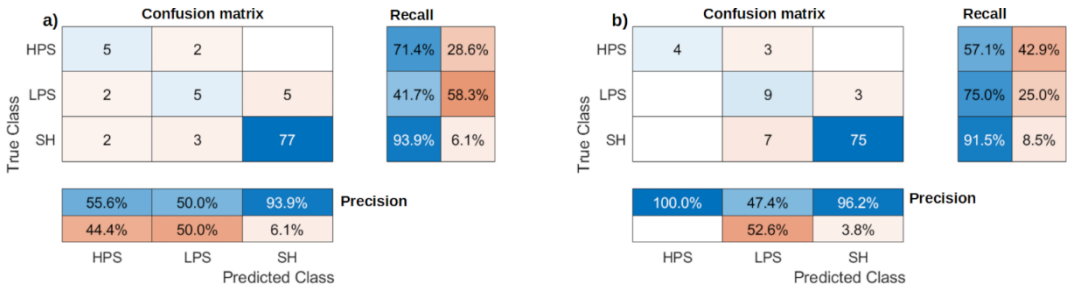

Figure 21: As in Figure 19 but for Well 2.

	LSTM				**Linear inversion**			
	Vp err. (%)	*Vs* err. (%)	*ρ* err. (%)	Data err. (%)	*Vp* err. (%)	*Vs* err. (%)	ρ err. (%)	Data err. (%)
Test 1	3.15	3.44	1.10	14.33	6.40	3.62	2.96	13.65
Test 2	3.18	3.36	1.20	19.71	7.58	5.07	3.25	13.25
Test 3	4.06	3.42	1.58	30.72	8.21	5.15	3.23	24.80
Test 4	7.89	4.86	2.37	34.12	11.32	5.49	3.52	26.99
Test 5	8.01	5.01	2.31	34.29	12.48	6.42	3.43	25.63

Table 1: Summary of the results provided by the LSTM and linear elastic inversion on synthetic inversion tests.

	LSTM				**Linear inversion**			
	Vp err. (%)	*Vs* err. (%)	ρ err. (%)	Data err. (%)	*Vp* err. (%)	*Vs* err. (%)	ρ err. (%)	Data err. (%)
Well 1	8.44	5.49	3.62	32.22	9.23	5.84	3.92	28.35
Well 2	6.09	4.64	2.03	31.03	6.69	5.90	2.15	28.04

Table 2: Summary of the elastic inversion results on the two field seismic gathers.