

UNSUPERVISED SEISMIC ATTRIBUTE INTERPRETATION USING DEEP LEARNING

J. Walda, S. Dell, and D. Gajewski

email: *jan.walda@uni-hamburg.de, jan.walda@uni-hamburg.de, dirk.gajewski@uni-hamburg.de*
keywords: *artificial intelligence, machine learning, seismic attributes, interpretation, deep learning*

ABSTRACT

Machine learning, in particular deep learning, has become a vital factor in pattern recognition and repetitive tasks, often outperforming humans. Seismic interpretation is often associated with finding specific patterns of interest and can depend on the interpreters involved. We aim to provide consistent automatic interpretation of specific data, that assist interpreters. In order to do so, we combine deep learning with traditional machine learning techniques for automatic interpretation of seismic attributes using 3D data of the F3 block, offshore Netherlands and the Volve Field. A major difficulty of seismic interpretation is the way of dealing with the richness of seismic attributes, which results in a multi-dimensional problem in interpretation. Usually, the amount of seismic attributes is reduced, e.g. by principle component analysis, before interpretation. In order to analyze the most important spatial information from two sets of attributes containing six attributes each, we use a 3D convolutional autoencoder. The autoencoder aims to find a reduced representation of the data. To verify, whether the found representation is reasonable, we reconstruct the original data and evaluate the misfit of reconstructed and original data. Once the misfit is sufficiently small, we cluster the reduced representation (encoding) to obtain a feature cube that contains a label for each sample. This process reduces the multidimensional information of multiple seismic attributes and their spatial distribution to one label for each sample in the 3D spatial volume. The found labels can be interpreted instead of the numerous seismic attributes, which eases and accelerates interpretation and reduces cost. Furthermore, human interpreters might overlook features of interest in the seismic attributes, which can be revealed by our unsupervised deep learning approach.

0.2 INTRODUCTION

Seismic attributes describe the specific characteristics of a seismic signal, e.g., a shape or continuity over spatial intervals, and are derived either from a single-trace or by comparing adjacent traces (Chopra and Marfurt, 2005; Brown, 2001). They play an important role in prospecting, as they provide valuable information about seismic data, which in turn, is used for qualitative and quantitative analysis, e.g., facies mapping (Ross and Cole, 2017; Qian et al., 2018) or porosity prediction. There have been hundreds of different seismic attributes reported so far. The usage of a particular attribute is mainly determined by the interpretation need (Roden et al., 2015). Complex trace (instantaneous amplitude) attributes help to interpret lateral continuity of stratigraphic events or bedding geometry. Geometric attributes, such as dip and azimuth, are used to detect faults and fractures (Randen et al., 2001). Spectral decomposition attributes highlight lateral variability of depositional features and faults, and so on (Chopra and Marfurt, 2007).

Seismic attributes might linearly be related to each other and differ only in the resolution (Barnes, 2007) so that deriving useful information from the dependent attributes is a tortuous process, which usually does not result in a significant uplift in classification of structural and stratigraphic features. Several methods were proposed to identify linear relationships between attributes, among other, standard correlation, rank

correlation, and principal component analysis. Rank correlation is computed like standard correlation, except that the attribute values are first sorted, in order of value, followed by a correlation of their ranks in the sort (see p 31. of Isaaks and Srivastava, 1989). Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of correlated observations into a set of linearly uncorrelated variables called principal components. It transforms (de-correlates) a set of linearly related seismic attributes into a new set of independent attributes (see e.g., Duda et al., 2001). PCA has proven to be a robust method to reduce a large set of seismic attributes, to indicate variations in the data and to determine the most salient attributes (e.g., Roden et al., 2015). However, PCA is prone to non-linear dependency, as it is mainly focused on finding hidden linear correlations between seismic attributes. Moreover, PCA assumes that large variance implies low covariance that successively averages a most salient component. There are several cases, where mean and co-variance do not fully describe a statistic distribution, which limits PCA in several seismic applications. To better acknowledge non-linear dependencies in seismic attributes, several attempts have been made to complement PCA with unsupervised neural networks, e.g., self-organizing-maps (Roden et al., 2015). We suggest using another unsupervised network, so-called autoencoder, which is an encoder-decoder network. The network utilizes nonlinear encoder and nonlinear decoder functions, thus can learn a nonlinear generalization of standard PCA. Supervised convolutional neural networks have been successfully applied to salt body classification (Shi et al., 2018; Di et al., 2018) and fault interpretations (Wu et al., 2018) already. Unsupervised approaches using sparse autoencoder and seismic data directly exist (Shafiq et al., 2018b,a). However, they use sparse autoencoder rather than 3D convolutional autoencoder, as we do here. Though, using seismic attributes in an unsupervised scheme is not yet established.

In this paper, we describe a methodology to analyze combinations of seismic attributes as well as the seismic data and extract a set of uncorrelated attributes, which we call a feature cube throughout the paper. We analyze several seismic attributes, aiming at an improved characterization of faults and contours of salt bodies. Additionally, we analyze the seismic data to detect anomalies that indicate areas of interest for further evaluation. We use a combination of traditional machine learning and deep learning, to perform a sample by sample automatic identification of spatial features via an encoder-decoder network. We organize the encoder-decoder network as a so-called autoencoder, which tries to encode information. In a second step the encoded information is fed into a decoder that tries to reconstruct the original image. With sufficient training, the autoencoder will encode spatial information as well as information from several seismic attributes for each sample and reconstruct the original data. If the reconstructed and original data match, we assume the autoencoder found a data representation that contains all important features. These features are classified using an unsupervised clustering algorithm. In our work, we use the k-means algorithm for clustering. As a result, we obtain a volume of clusters (feature cube), which can be labeled during interpretation instead of dealing simultaneously with a multitude of seismic attributes of the data. Our approach automatically delivers the clustered volume just from provided data without human interaction (apart from parameter and seismic attribute selection), which makes it particularly appealing for processing large 3D data. We aim to propose a method that automatically clusters features based, on the input and is not limited to one specific task, such as fault detection. Rather, the choice of input should determine the task. Furthermore, we aim to train a network on one data, that is general enough to be applied to other data-sets, without the necessity of retraining.

METHOD

We attempt to automatically cluster seismic data. In this work, we use geometric attributes as attribute set 1, which can be used to identify faults, fractures, regional stress fields and anisotropy, see e.g. in Roden et al. (2015). However, our method does not make assumptions on the data used nor the interpretative use afterwards. It simply extracts information from given input data. Therefore, this method is usable with any kind of seismic attributes and interpretative aim in mind without changes apart from the input data. This is demonstrated with the choice of another attribute set 2, where two attributes haven been changed as suggested to us by A. E. Barnes (personal communication, September 14, 2019). This attribute set consists of the semblance, the average frequency, trace envelope, most negative and positive curvatures and dip. Furthermore, those attributes are determined from the coherency-weighted stack, while attribute set 1 is determined from the original stack. In principle, it is not necessary to prescribe the seismic attributes,

though the task changes in this case. In a third case, we will use the seismic directly, in order to demonstrate the flexibility of the method in that regard.

In principle, we could classify the seismic attributes of each sample directly using traditional machine learning techniques such as k-means, Gaussian mixture models (Rasmussen, 2000) or self-organizing maps (Kohonen, 1990). If we do so, we use only single pixel values and ignore lateral information, apart from the seismic attribute estimation. This information is crucial when dealing with seismic data and attributes, since structures in the subsurface have extensions and patterns, suitable to find features, such as faults, especially when just the seismic data is used. Though the attributes might be estimated considering local features, most traditional machine learning techniques do not consider lateral behavior of the estimated attributes. Some traditional machine learning techniques like support vector machines (Cortes and Vapnik, 1995) are able to account for lateral information. They can be seen as a neural network of specific architecture with an involved pre-processing. However, deep learning approaches can learn more complex relations due to their advanced architecture (Goodfellow et al., 2016).

Our method naturally considers local features in the estimated attributes due to our deep learning approach. We use a convolutional neural network (CNN) structure as an autoencoder (encoder-decoder) network. This network consists of a series of convolutional filters and subsequent data reductions, called max pooling. The aim of the encoder is to find a representation which describes the given data. The decoder uses the encoded features to, e.g., reconstruct the original input (self-supervised learning) or relate the encoded features to some labeled data (supervised learning). In our case, we want the method to cluster our data without human interaction. Interpreting the clusters occurs after the neural network has organized, reduced and clustered the data. Therefore, we reconstruct our input data using the decoder from the encoded representation of the encoder. If the reconstruction is sufficiently close to our input data, we can assume that the encoding describes the main features of the data. The encoded representation contains the information of all used attributes and their lateral behavior. Furthermore, it is a compressed representation of the most meaningful information in the data. Uncorrelated noise is rarely a main feature of seismic attributes, which usually have a good signal to noise ratio. Therefore, noise is mostly ignored in the encoded data, which benefits the following clustering algorithm tremendously.

Once we found a good encoded representation, we substitute the decoder of our network with an unsupervised clustering algorithm. In this work, we chose k-means, because this is a common and robust algorithm. The seismic attributes used in our work are described in the following section.

Calculation of seismic attributes

Seismic attributes represent subsets of the quantitative information in the seismic data based on measured time, amplitude, frequency, and attenuation. These subsets describe specific measurements of geometric, kinematic, dynamic, and statistical features in seismic data, which allows for grouping attributes into particular classes. For an overview of the categorization of seismic attributes we refer to, e.g. Roden et al. (2015). As one objective is delineation of faults and salt diapirs, we consider those particular attributes which describe, or can be co-rendered to, the lateral continuity of the seismic signal. We use fault likelihood, dip and azimuth, volumetric curvatures (the most positive and most negative), envelope and instantaneous frequency as attribute set 1.

Various methods are available for the computation of the volumetric vector dip, among other complex trace analysis (Barnes, 1996), discrete scans (Marfurt et al., 1998) or gradient structure tensor (Bakker et al., 1999). We apply a modification of the discrete scans method, so-called local parameter search, where we scan for volumetric vector dip and curvatures locally in a 3D aperture, using the following equation, based on CRS (Jäger et al., 2001).

$$t_R^2 = (t_I + p_x x + p_y y)^2 + t_I (K_{xx} x^2 + K_{xy} xy + K_{yy} y^2). \quad (1)$$

Here R stands for time-migrated reflections, t_I is the zero-offset image time, p_x and p_y are components of the volumetric vector dip and K_{xx} , K_{xy} , K_{yy} are volume curvature components (Iversen et al., 2012; Dell et al., 2014). The local parameter search also provides the semblance, which is used to compute the fault likelihood (Hale, 2012). Dip and azimuth as well as the most-positive and negative curvatures are calculated using the standard equations (see e.g., in Roberts, 2001).

The fault likelihood and the first-order attributes, dip and azimuth, allow for a delineation of lineaments. Semblance, as a measure of the similarity of seismic waveforms of neighbouring traces, becomes low, if the seismic waveforms vary due to structural features. These can be faults and fractures, chaotic seismic patterns or changes in seismic facies (Marfurt et al., 1998). As an opposite to the semblance norm, the fault likelihood attribute becomes very high (Hale, 2012). Dip in the time domain represents an apparent slope of a geological formation. This slope can be considered as a special case of the ray illumination, when assuming the rays come vertically from above in a homogeneous medium. The azimuth attribute is complementary to the dip attribute and can help identify both changes in slopes: suddenly occurring changes (spatially high-frequency) caused by discontinuities and also gradual changes caused by wrinkles and flexures (Birkhäuser et al., 2001). The azimuth attribute is sensitive to increased noise in the data. Combining dip and azimuth frequently provides a better mapping of the geological horizons (Birkhäuser et al., 2001).

Curvature attributes, the second-order attributes, contain information relating to the local shape and magnitude of lineaments. The most positive curvature measures the most positive rate of change in reflection dip and the most negative curvature measures the most negative rate of change in reflection dip (Barnes, 1996). This information can be used to highlight either anticlinal tops (reflection bumps) or synclinal bottoms (reflection sags). The mean curvature measures, how the volume locally changes, e.g., a plane is flat and has zero curvatures including the most-positive, most-negative and mean (Guo, 2014). If a surface is curved, it usually has non-zero curvatures. However, for a small spherical inhomogeneity (a point diffractor), the mean curvature is zero, which enables to distinguish between the changes in volumetric dip and the most-positive (negative) curvature caused by faults and embedded inhomogeneities.

We also compute the envelope and the instantaneous frequency. The envelope represents the instantaneous energy of the signal and is proportional in its magnitude to the reflection coefficient. The envelope is useful to highlight discontinuities, changes in lithology, faults, changes in deposition, tuning effect, and sequence boundaries. The instantaneous frequency represents the mean amplitude of the wavelet, i.e., it corresponds to the average frequency (centroid) of the amplitude spectrum of the seismic wavelet. We compute both attributes using a complex trace approach (Taner et al., 1979; Barnes, 1996).

In total, we use six seismic attributes for both attribute sets to extract geometric features, which means dealing with a six-dimensional interpretation space. Therefore, a method is required to automatically find attribute redundancy, in order to create a single dimensional feature cube. This cube is supposed to rather help and guide an interpreter to better track structural elements of interest, e.g., in creating fracture-density maps. In the next section, we will present such a method based on artificial neural networks.

Deep convolutional neural network

In our work, we aim to automatically extract information from seismic attributes using a deep encoder-decoder neural network. The phrase deep means a neural network of many layers, although it is often used for networks composed of more than one layer. The autoencoder we use, consists of seven convolutional layers and three max pooling layers. The network architecture as well as details about convolutional layers are described in the next section of the paper. Prior to that, we describe the strategy.

First, we extract seismic attributes from a volume of time-migrated reflections and convert them into a data tensor (Figure 1). This data tensor has five dimensions. The first contains all the examples used. This size is determined by a hyper-parameter of the neural network called batch size. In this work we use a batch size of 128. The samples in the batch change with each iteration. In our case we determine the samples randomly. The second, third and fourth dimension are the depth, in-line and cross-line coordinates of the extracted samples. For the training, we use a size of 16x16x16. For the actual application, we use 8x8x8 dimensions. Since we perform three times max-pooling, the bottleneck ends up as a vector where the size of the dimensions is 1. This is important since we want to determine the encoded vector for each sample in the data during the application. However, a larger size of the training cubes helps the autoencoder to converge faster. The fifth and last dimension are the channels. Each attribute gets one channel, so we end up with a total of six channels for both attribute sets. If the seismic data is used directly, only one channel is necessary. The autoencoder tries to find the most important aspects (encoding) of the data examples, which leads to a data reduction in the bottleneck layer, shown in Figure 2. The next step is a reconstruction of the seismic attributes from the encoded data (bottleneck) called decoding. If this task is successful, we assume

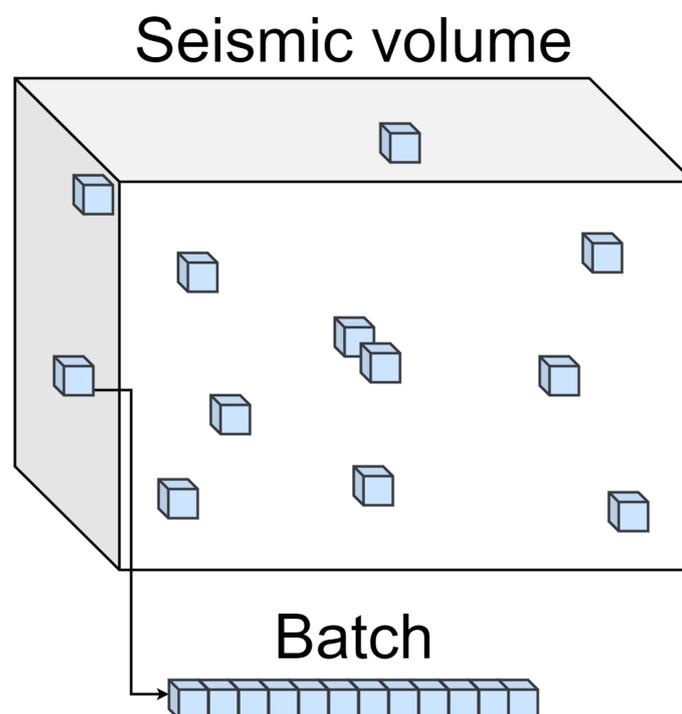


Figure 1: Organization of a training batch for a single channel. We chose small cubes randomly and attach them into a batch, that can be used efficiently by the autoencoder. The batch has the following resulting dimensions: number of sub-cubes in batch, number of depth samples of a sub-cube, number of in-lines samples of a sub-cube, number of cross-lines samples of a sub-cube, number of channels.

that our network learned the most important aspects of our data and we transit to classify the reduced data into clusters, which an interpreter can label as structures, e.g. faults, salt and noise. The seismic attributes used are chosen by humans and can be changed, if other information is desired. In the supervised case, the network needs some example cases of faults and salt to learn which features are important for faults and salt. This approach is called supervised, since the network has some specific information to look for. In the unsupervised case, the neural network will just cluster data to certain groups (Goodfellow et al., 2016). The interpretation of the groups as faults, salt and other events is done by the human interpreter later on. Unsupervised learning results are harder to interpret, since they have no clear label, such as fault, compared to the supervised case. The label is given to the clustered result later on and might not look as expected, since previously hidden information is revealed, which a supervised network might have ignored, since it was not important for the desired task. The unbiased approach of the data is the advantage of unsupervised learning though.

The autoencoder network we use is based on the U-Net (Ronneberger et al., 2015), developed for image segmentation in electron microscope stacks. It was shown in many studies that this network performs particularly well, where limited amounts of data are available. This is also the case for our purpose, as we have limited amounts of samples where areas of interest are present. The majority of the seismic data in the North Sea is marked by sediment layering. However, we simplified the network structure, shown in Figure 3, to relax memory demands and reduce the number of unknowns. We found the simpler version is much easier to train without loss of performance. We use a 3d convolutional autoencoder with seven convolutional layers (light orange in the figure), three max-pooling (dark orange) and up-scaling layers (blue).

A 3D convolutional layer contains of M different filters of $N \times N \times N$ size. In our case, M varies with each layer, according to Figure 3, while the filter size $N = 3$ is kept constant. Some exemplary filters are well known from image processing, such as edge detection and blurring. The max-pooling operation is a

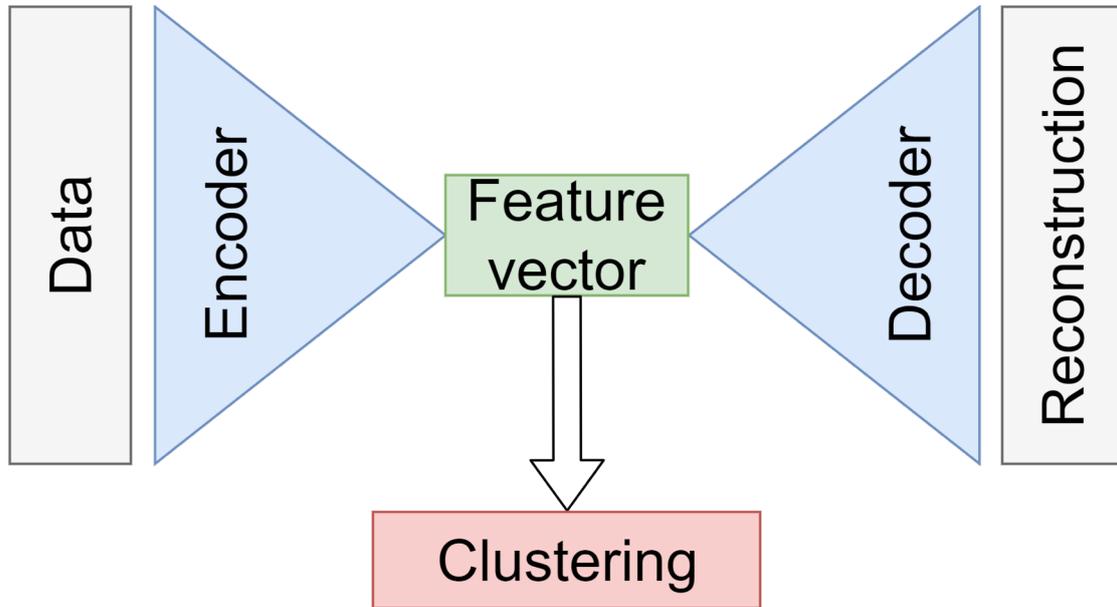


Figure 2: General strategy of the method applied. We feed our data into an autoencoder, which encodes the data into a feature vector, highlighted in green. This feature vector is decoded to reconstruct the original image. Once the reconstruction is sufficiently accurate, we cluster the feature vector of each sample in the data.

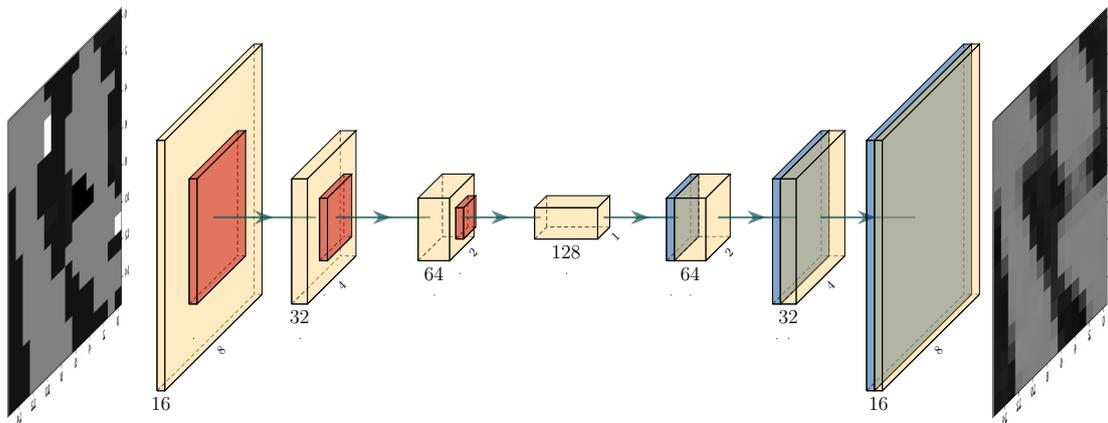


Figure 3: Structure of the autoencoder network used in this work. In the first half (encoder), the data is encoded into a vector of length 128. These 128 encodings contain the most important features of the considered attributes and their spatial behaviour. The second half decodes the encoding to the original attribute sections. The input of the first layer and output of the last layer are compared by a misfit function, in our case mean squared error. The filters are updated reducing the misfit similar to conventional inversions. Bright orange layers represent the 3D convolutional layers where the number of filters per layer M is written beneath. The filter size N of all convolutional layers is $3 \times 3 \times 3$. The darker orange layer represents max-pooling, where the data image size is reduced. The blue layer represents up-scaling layer that reverse the effect of the max-pooling in the decoder. For the training we used batches of size $16 \times 16 \times 16$. For the application we used $8 \times 8 \times 8$ images, indicated by the titled numbers at the convolutional layers. The image on the left side is a 2D slice of an input image and the output of the autoencoder of that image is shown on the right side.

data reduction operation, were only the most significant sample of a small area ($2x2x2$ in our case) is kept. Batch normalization is usually done to make input comparable (e.g. values ranging from 0 to 1) throughout the whole network. Therefore, higher learning rates can be used, which leads to a computational speedup. Furthermore, it reduces one of the premier problems in machine learning, called overfitting. Overfitting refers to the problem that a neural network learned to perform a task for the training data but is not able to generalize to the test data, i.e., data the network has not seen yet (Goodfellow et al., 2016). Identifying and avoiding overfitting is still subject to extensive research and a major reason why networks in seismic are often designed and/or trained for specific data. As we demonstrate with the Volve data, we can avoid overfitting in our applications. This is due to the randomization of the input sample cubes and a low learning rate ($2.5 \cdot 10^{-5}$). A typical learning rate for 2D convolutional layers, such as 10^{-3} lead to stagnation. We also did not split the data regularly into smaller cubes and shuffled them afterward, as is typically done. We chose random samples, where we cut out the surrounding cube. This helped to enable the possibility of intersecting cubes, together with the choice of larger cubes for the training process, compared to the actual application.

In order to deal with very limited amounts of training data, the U-Net copies outputs of different stages of encoding into the decoder as additional examples. In the original U-Net (Ronneberger et al., 2015), this is helpful as it was applied in a supervised learning scheme for image segmentation. In our case, we want to encode data and reconstruct from the encoding. Using the skipping layers in the U-Net makes it impossible for us to train the bottleneck layer of the autoencoder well enough. We can not achieve consistent clusters with skipping layers. Therefore, we do not copy results of different encoding stages into the decoder. This forces each image to be encoded and decoded using the full neural network architecture and ensures that, if the reconstruction is successful, all information is encoded in the deepest layer (bottleneck) of the encoder. This specific layer is fed into a subsequent unsupervised clustering algorithm.

Figure 4 shows the reconstruction of the mean frequency for in-line number 250 of the F3 block. We observe, that the majority of the features are reconstructed. This means the encoder part of the autoencoder found a representation of the data that is condensed, while containing all information, since the decoder could reconstruct all events from the encoded feature vector. For demonstration purpose the figures show a 2D example, though we use 3D images (time, in-line and cross-line) with 3D convolutions for training and application. The main shape of the structures in the image are preserved. However, extreme values are suppressed, which is apparent in the difference plot (Figure 4(c)). This can be avoided by the introduction of skipping layers. In our case, we are not actually interested in a perfect reconstruction, we are after the encoding. Therefore, we accept a slightly worse reconstruction to ensure all reconstructed data is generated from the encoding. Before we consider the clustering, we describe what happens in each layer of the autoencoder.

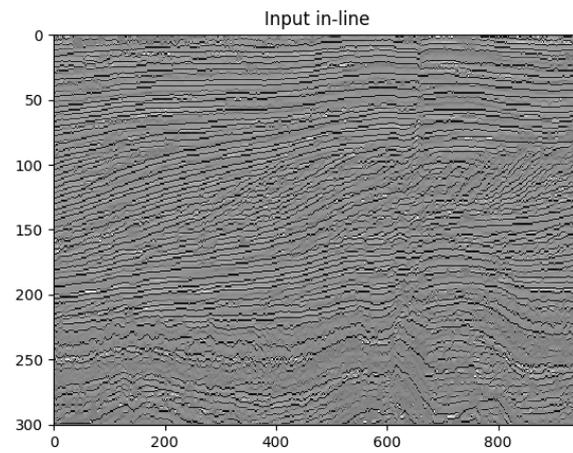
Feature extraction

We apply our method using 3D data as well as 3D convolutions and max-pooling. Nevertheless, the concept behind the approach is the same for 2D and 3D. Since the 2D case is easier to digest, we explain the concept for the 2D case.

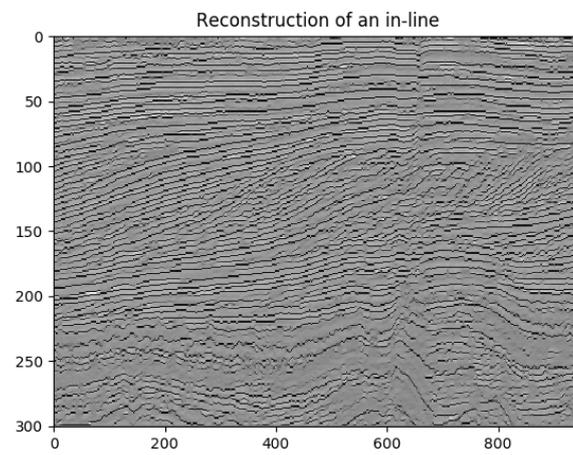
A convolutional network applies filter kernels \mathbf{W} to an input \mathbf{X} to obtain an output $\hat{\mathbf{X}}$ called feature map by applying a cross-correlation. A discrete two-dimensional cross-correlation of an image sample and the filter kernel is given by

$$\hat{X}(i, j) = \sum_n \sum_m I(i + m, j + n)W(m, n), \quad (2)$$

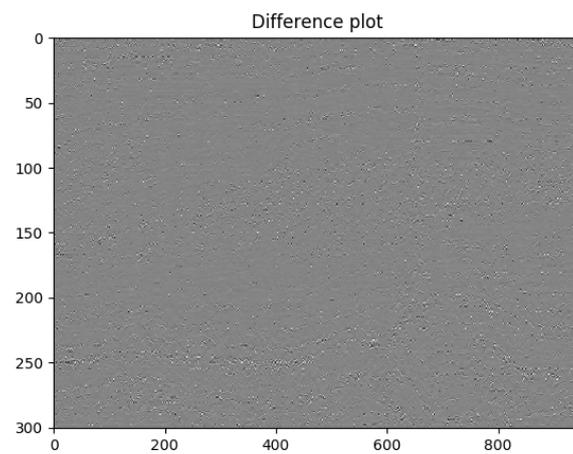
where I is the input to the respective convolutional layer, which is the original image for the first layer and the output of the previous layer in the network for the following. The output of the convolutional layer \hat{I} corresponds to $\hat{\mathbf{X}}$ for the last layer. The dimensions of the filter are denoted by m and n respectively. One filter kernel \mathbf{W} is applied to every sample of an input per feature map. Therefore, usually several feature maps are generated by convolution of different filter kernels with the input at each layer. In the deep learning approaches, several layers are applied to the input data. As a result the encoded representation is a result of many convolutional filters applied to the input data with occasional data reduction. The data reduction is applied to (for the 2D case) $2x2$ pixel subsets of the images, where only the most active



(a) Input data



(b) Reconstruction



(c) Difference

Figure 4: Reconstruction of the mean frequency of in-line 250 from the F3 block. We observe a good reconstruction, where all major features are preserved.

(maximum of the absolute value of the pixels) is used in the next layer. In case the image would be 64×64 pixels, the result of the data reduction would be a 32×32 image, where only the most meaningful pixels were considered. This is the main reason for noise suppression in autoencoder networks, which is also one of their main purposes. The update of the convolutional filter kernels is done using backpropagation (Goodfellow et al., 2016), which uses gradients that are backpropagated from layer to layer to update the corresponding filter kernels of the updated layer. The gradient is given by

$$\frac{\partial J}{\partial \mathbf{W}_i}, \quad (3)$$

where J is the objective function, \mathbf{W} the filter kernels and i the layer index. A frequently used objective function for data regression is the L2 Norm or mean squared error (MSE)

$$J = \frac{1}{3N_b} \sum_{i=1}^{N_b} \|\mathbf{X}_i - \hat{\mathbf{X}}_i\|^2, \quad (4)$$

where \mathbf{X} is the original image, $\hat{\mathbf{X}}$ the reconstructed image and N_b the number of images in a batch (128 in our case). The objective function aims to minimize the misfit of the original image with the reconstructed one. In machine learning, the objective function as well as the output of the last layer of a neural network depend on the goal. Our first problem (training the autoencoder) is a regression problem, where MSE is a common objective function. In case of a classification problem, a cross entropy is used instead, if the ground truth is known. We do not want to rely on a sufficient amount of manually picked ground truth data that might contain presumptions. Therefore, we use the regression to reconstruct the input data through the autoencoder and substitute the decoder by a clustering algorithm. For the training and application of the autoencoder, we use the Tensorflow API developed by Google (Abadi et al., 2016). Once the training is done, we cluster the encoded representation of our data.

We already mentioned one of the main reasons autoencoder are used is noise reduction due to the data reduction. Another import purpose is including lateral information, which is done by the subsequent application of the convolutional filters. The encoded representation we use in this work contains spatial information of the neighboring eight samples around the target sample for all six attributes. This ensures that different structures have significantly different patterns in their encoded representation, which is necessary for the clustering. The encoded data for each sample is a vector of 128 different data representations, also called features. The activity of the 128 features is fed into the clustering algorithm, which means significantly more information is available than just by clustering the six seismic attributes directly. This benefit of the deep learning network helps to cluster the seismic attributes more reliably.

Clustering

In the previous part, we described our method to extract salient features of our seismic attribute cubes. To make use of those features and actually obtain our desired output, we need to cluster the features into different categories. In this work, we are interested in two specific structures for the F3 data: faults and salt. In principle, we could cluster our features in three categories, the third category is used for everything we are not interested in. However, that would require labeled data for training, which we do not want to rely on. We need to add more cluster in order to account for all present features. Those might be our desired faults and salt but also reflections, noise and more. Thus, we assume a larger amount of labels and interpret the labels at the end. This effectively means, we interpret the output of our machine learning framework instead of teaching the machine learning framework our way of interpretation. This has advantages we discuss in the conclusions. A major disadvantage is, that we might get results which we do not expect or do not fit our task. Furthermore, unsupervised approaches are viewed as unstable, which we aim to disprove.

There are many ways to perform clustering, either by neural networks or more traditional methods, such as k-means (MacQueen et al., 1967), among others. The k-means algorithm is our choice in this work. The k-means clustering method is a vector quantization method that partitions the data space into clusters. In our case, we aim to partition our encoded data space into six. The choice of clusters depends on the task and provided data. Since we are interested in the consistency of the method, we chose a small amount of

clusters (six) with significant distinction, which should be reproducible, apart for small areas, with the two chosen attribute sets. In preliminary tests, we found that a high amount of clusters is only necessary when a large amount of attributes are available, particularly if the seismic data is used directly. However, there is no general guideline how many cluster are necessary. The density-based spatial clustering of applications with noise (Ester et al., 1996) determines the number of clusters on the fly and can be used as an alternative to k-means, if in doubt. The k-means algorithm performs clustering in an unsupervised fashion, meaning we do not need to provide sample cases, where the ground-truth is known. However, we also face the risk that the algorithm will not cluster the way we want it to, since the algorithm might find another classification which is more suitable. We can try to circumvent this issue by clustering our data into more clusters and interpret each cluster at the end. The k-means algorithm uses our encoding as input and substitutes the decoder, once the feature extraction has satisfying performance.

The k-means algorithm aims to cluster N_o observations with m features into k cluster. Since we want to cluster each sample of our 3D volume, N_o is the number of samples in our data, m is the number of features of each sample. The encoder we use, delivers a vector of length 128, therefore $m = 128$. Our data vector \mathbf{x} has a length of $m \cdot N_o$. In case of memory issues, one can apply the k-means clustering to batches, similar to neural networks approaches.

We chose $k = 6$ clusters. For this setting, k-means chooses six clusters, such that the variance of observations within a cluster is minimal. Formulated as an equation it reads:

$$\arg_S \min \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg_S \min \sum_{i=1}^k |S_i| \sigma^2(S_i) \quad (5)$$

where S is the center of the cluster, i the cluster index, \mathbf{x} the data, $\boldsymbol{\mu}$ the mean value of S_i and σ^2 the variance. The k-means algorithm is applied to the encoded feature vector, which is the first part, up to the bottleneck layer, of Figure 3 of each sample. During clustering, the filter of the encoder are fixed, and estimated previously from the training of the autoencoder. These two steps are independent of each other but can be refined simultaneously, if needed. The result is the cluster of the considered sample. After the application of k-means to the full data, we obtain a feature cube as shown in Figure 6 for attribute set 1 and Figure 7 for attribute set 2.

NORTH SEA DATA

F3 Block offshore Netherlands

F3 is a block in the Dutch sector of the North Sea. The block is covered by 3D seismic data that was acquired to explore for oil and gas in the Upper-Jurassic - Lower Cretaceous strata. The upper 1200 ms of the data set consists of reflectors belonging to the Miocene, Pliocene, and Pleistocene. The large-scale sigmoidal bending is readily apparent, and consists of the deposits of a large fluviodeltaic system, that drained large parts of the Baltic Sea region (e.g., Sørensen et al., 1997; Overeem et al., 2001).

The deltaic package consists of sand and shale, with an overall high porosity (20 – 33%). Some carbonate-cemented streaks are present. A number of interesting features can be observed in this package. The most striking feature is the large-scale sigmoidal bedding, with text-book quality downlap, toplap, onlap, and truncation structures. Bright spots are also clearly visible, and are caused by biogenic gas pockets. They are not uncommon in this part of the North Sea. Several seismic facies can be distinguished: transparent, chaotic, linear, shingles. Well logs show the transparent facies to consist of a rather uniform lithology, which can be either sand or shale. The chaotic facies likely represents slumped deposits. The shingles at the base of the clinoforms have been shown to consist of sandy turbidites.

Volve reservoir

The subsurface and production data of the Volve Field was opened by Equinor in 2018 (Equinor, 2018). The Volve reservoir is located in block 15/9 in the middle of the North Sea and was discovered in 1993. The production of the reservoir started in 2008 and was shut down in 2016, delivering about 9.5 million barrels of oil with a recovery rate of about 54 %. The reservoir is located in the Hugin Formation in a depth of 2700 m and 3100 m and consists of Jurassic sandstones. The west side of the reservoir is heavily faulted.

RESULTS

The goal of this work is to automatically obtain a 3D feature cube out of several seismic attributes, which should ease and accelerate seismic interpretation. Figure 5 shows three slices of the data set, crossing the salt body. Figure 6 is the clustering result of the proposed method using the geometrical attributes (attribute set 1). We observe that the major faults are well identified. However, the clusters they belong to vary, which can be result of their different origination. The salt body, in the green cluster number 3, is identified consistently and can be automatically extracted for further evaluation. Cluster number 5 in black though, seems to be meaningless and noisy. The slightly different attribute set 2 results in Figure 7 show mostly the same pattern, though the clustering result is less noisy. Furthermore, more faults were found. A major difference is, found faults are consistently clustered in several cluster numbers (between 2, yellow, and 4, blue). It appears to be dependent on the orientation, though, this needs further evaluation. Nevertheless, those details might have been missed by human interpreters or in labeled examples by supervised schemes. Apart from the little human interaction, that reduces cost and saves time, we believe this to be of interest and added value provided by our approach. Please note, that we cannot expect stratigraphy, as we did not use the according attributes.

One of the three slices, in-line 250, is evaluated in more detail in Figure 8, where the seismic data is shown on top (a) and the clustered attribute set 2 result overlain at the bottom (b). We observe that layered structures are mostly covered by cluster 0, which we kept transparent. However, slightly discontinuous structures are clustered into cluster number 2, shown in yellow. The red cluster number 1 includes inclined seismic events with a particular orientation. Cluster 3, in green, consists mostly of the salt body and the main fault. The blue cluster number 4 shows stronger discontinuities, mostly related to faults. The fault has more geometrical similarities to the salt body, where it originates from, than any other cluster. The method was able to find even small discontinuities, when we compare small yellow structures with the corresponding seismic (Figure 8(a)). From a deep learning and image processing point of view, this is no trivial task (Goodfellow et al., 2016) and therefore, emphasizing the use of autoencoder for unsupervised clustering. The highly fractured structure below 1200 ms is labeled in green, which means the neural network found a different pattern for the salt body and the fractures above the salt. However, horizons in and beneath the salt might still be labeled red and yellow, if their extension is larger than the excerpt used for encoding, which was $8 \times 8 \times 8$ samples. Even though the method has never seen the seismic or amplitudes, the extension of found labels correspond well with the seismic signals. Particularly the top of salt is quite accurate. Furthermore, the comparison of our approach (Figure 8) with pure k-means clustering (Figure 9a) and k-means with a PCA prior (Figure 9b) shows, that encoding seismic attributes and their spatial behaviour stabilizes and improves the clustering process significantly, particularly in geologically complex areas, like the salt body. The neural network approach found one more cluster and could differentiate faults into separate clusters, while the conventional machine learning approaches could only identify four clusters (including the transparent cluster 0). Please note, the ordering of the cluster is not the same as in Figure 8, since the autoencoder encoding is not used in the conventional machine learning approaches. Hence, the features the PCA and k-means methods use are different. Though the yellow cluster number 2 of Figure 9 coincides with the red cluster 1 of Figure 8. All faults and stronger discontinuities are labeled in black (cluster 5) in the conventional approaches, while separated in the autoencoder approach. The pure k-means clustering (Figure 9a) differs from k-means clustering with a PCA prior (Figure 9b) in the cluster number 3 and 4 (green and blue). The PCA prior emphasis the average frequency highly, which provides layering in blue. As a result the green cluster number 3 is missing almost entirely, which was used by the pure k-means approach to identify discontinuous horizons. Though the three approaches show differences, the general structures are very similar which indicates we obtained consistent results.

Figure 10 shows the migrated time slice at 640 ms without (Figure 10(a)) and with the labels colored on top (Figure 10(b)). We can follow the extension of the fault starting from in-line 400 and cross-line 800 to in-line 100 and cross-line 1040 (in blue, yellow and green). Furthermore, we observe a fault zone at the bottom right of the image. Those are easily visible, since they are labeled in differently (red and green) compared to the reflections (transparent). The transparent labels from the boundaries are due to the fact that we did neglect samples where the excerpt is larger than the distance to the boundary of the image. In that case, we simply initialized the value with zero, which is the same value as the transparent cluster. The workflow also works for data gaps present in the bottom left of the corner. In this case, data gaps were

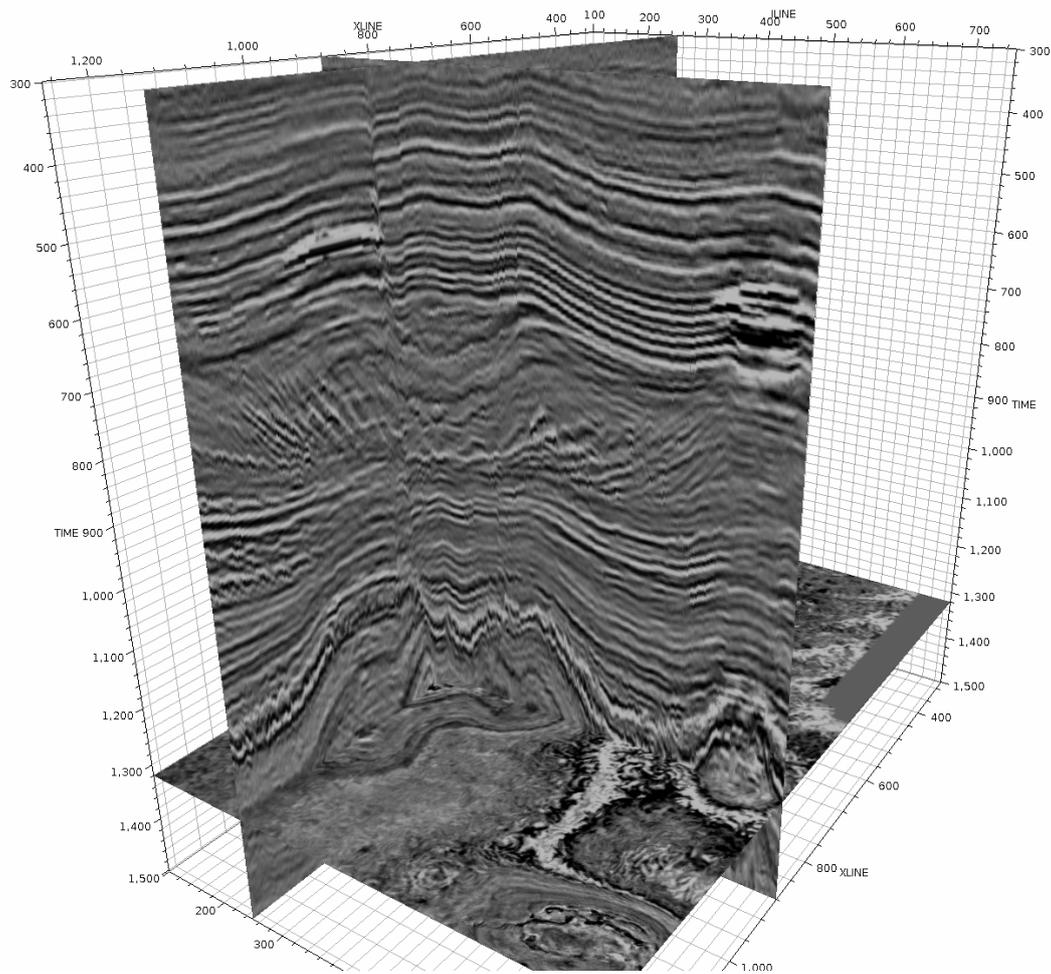


Figure 5: Time migrated slices of the F3 block offshore Netherlands. The slices cross a salt body in the lower part and several faults from the salt body up to the top.

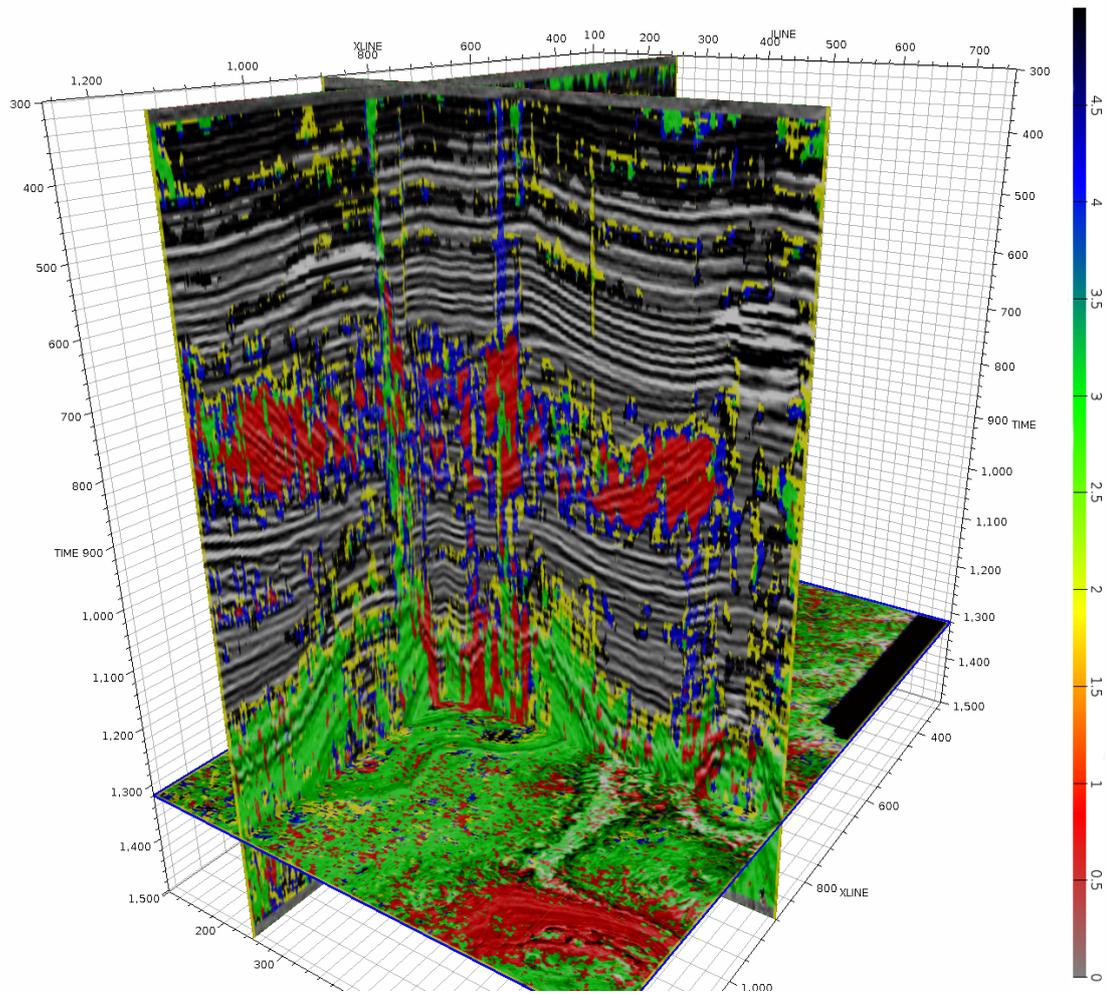


Figure 6: Geometrical feature cube of the F3 data obtained by the proposed method using attribute set 1. The major fault, in green, is clearly visible similar to the salt body, also in green, in the deeper part of the data. Most other faults were labeled in yellow, blue and red colors, showing the significantly different pattern in the geometrical seismic attributes. The boundaries of the data were not encoded and clustered to avoid boundary effects.

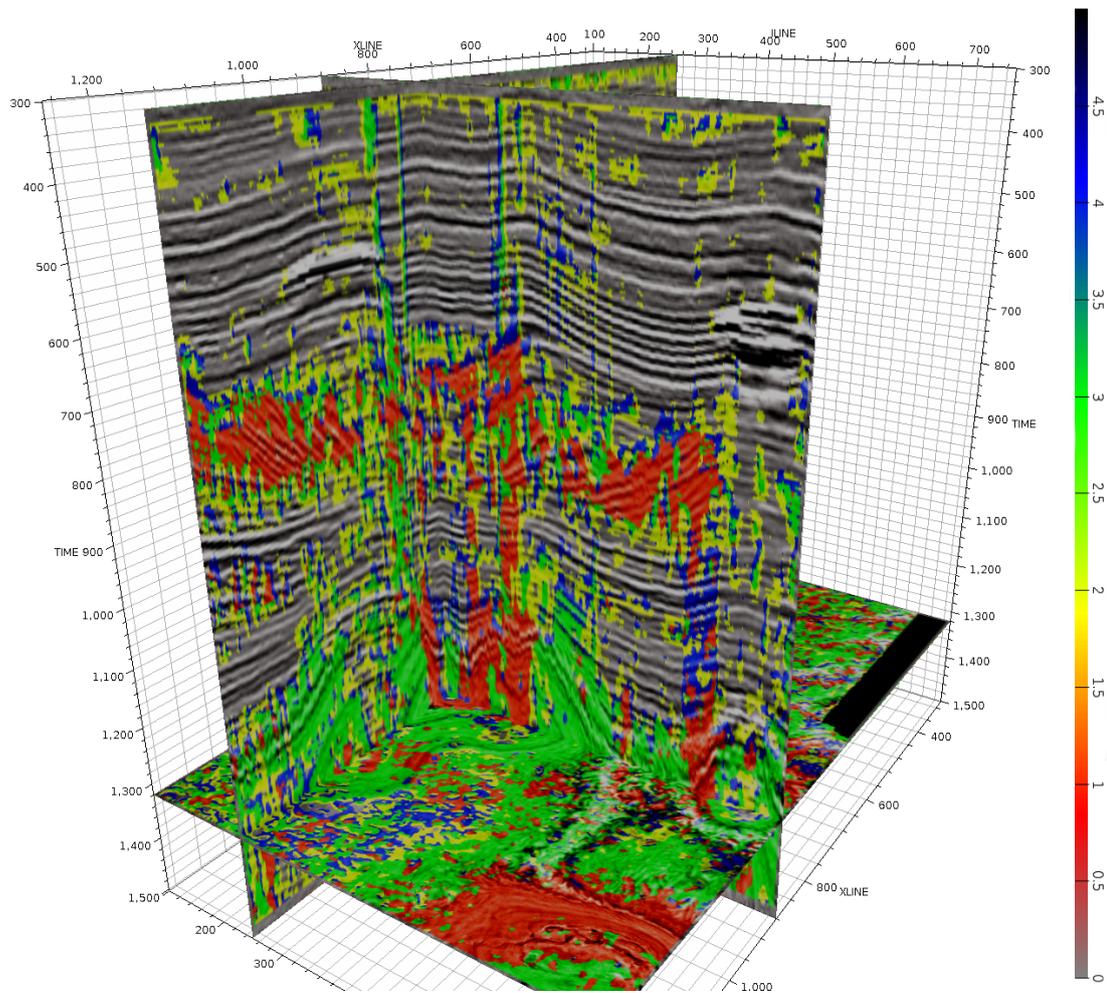
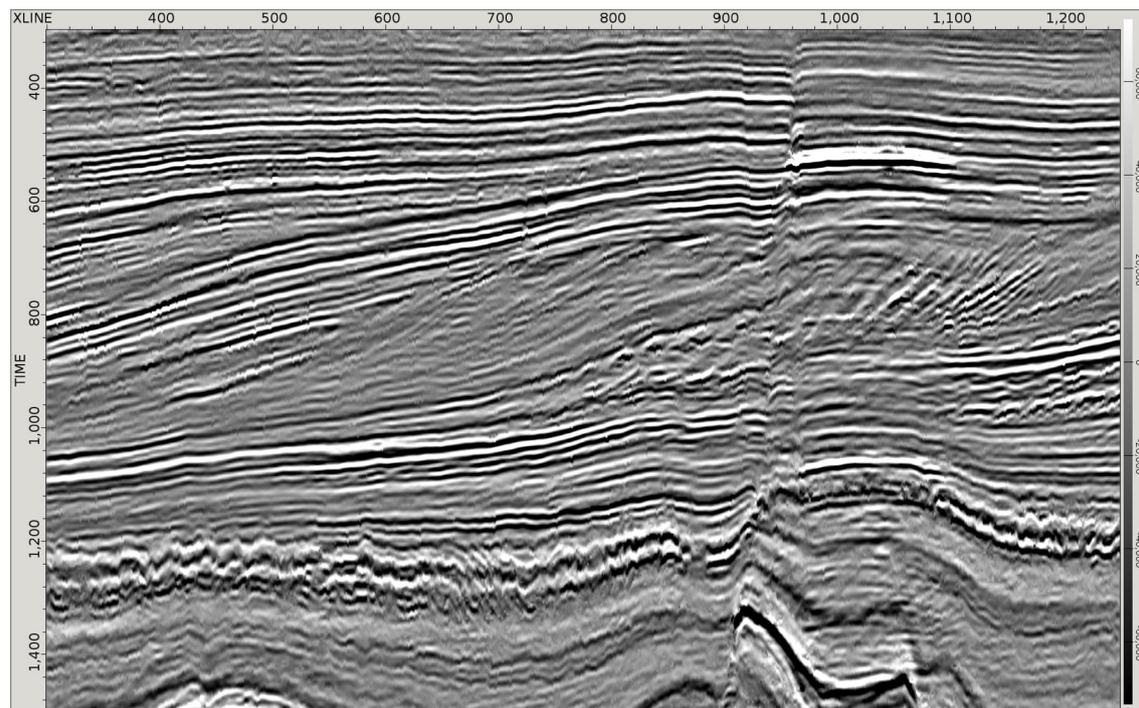
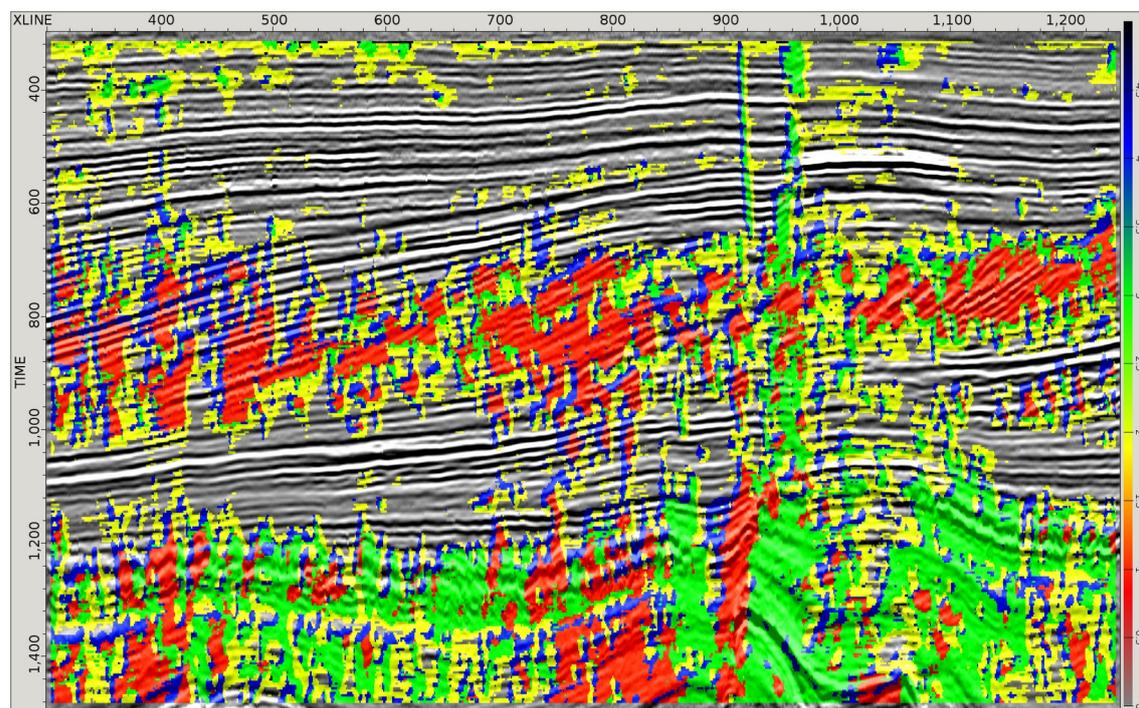


Figure 7: Feature cube of the F3 data obtained by the proposed method using attribute set 2. Similar to Figure 6, the major fault and salt body are labeled in green. Most other faults were labeled in yellow, blue and red colors. Though the main features are the same, we observe some differences, particularly in the upper part of the data. In general, attribute set 2 seems to find more faults. The boundaries of the data were not encoded and clustered to avoid boundary effects.

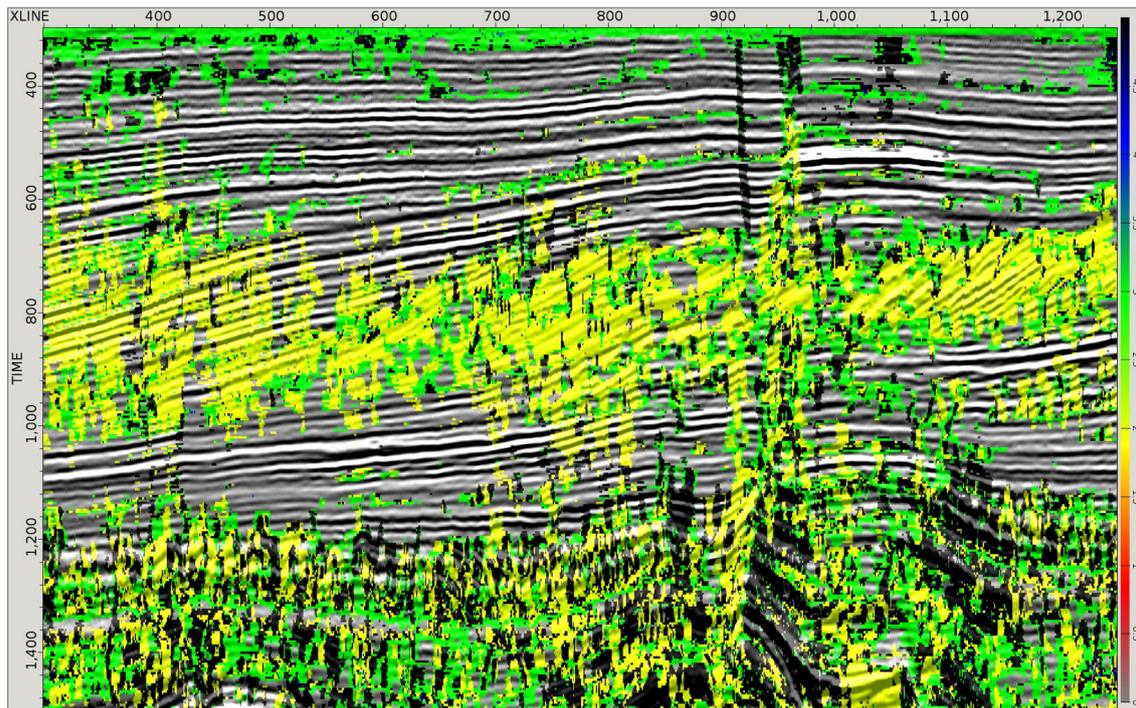


(a) In-line 250

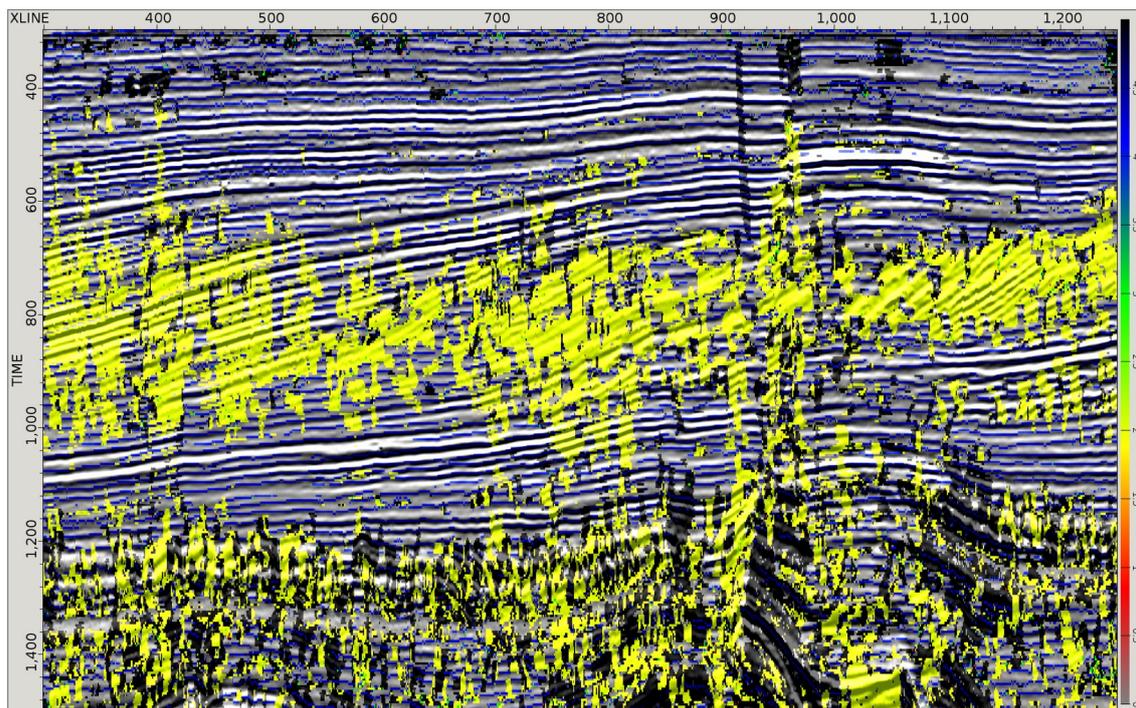


(b) Labeled in-line 250

Figure 8: Overlay of the migrated in-line 250 of the F3 data and its classified results using attribute set 2. The blue label (cluster 4) is present in areas where faults are clearly identified. The major fault is primarily green (cluster 3), similar to the majority of the salt body. Smaller discontinuities are marked in yellow (cluster 2).



(a) K-means



(b) K-means using PCA input

Figure 9: Overlay of the migrated in-line 250 of the F3 data and its classified results of attribute set 2, using k-means (9(a)) and k-means using a prior PCA on the attributes (9(b)). Except for the blue and green cluster (cluster 3 and 4) the results are very similar. However, the PCA helped k-means to better identify horizons, which are partly represented by the green areas (cluster 3) in 9(a). In 9(b), they are clearly identified in blue (cluster 4). However, in both cases a total of four clusters were identified, while the deep learning approach provided five.

labeled into their own cluster number 5 (black) by the autoencoder approach.

Though the seismic and labels correspond well, the network has never seen the seismic. It just has seen seismic attributes such as the discontinuity attribute, shown in Figure 11.

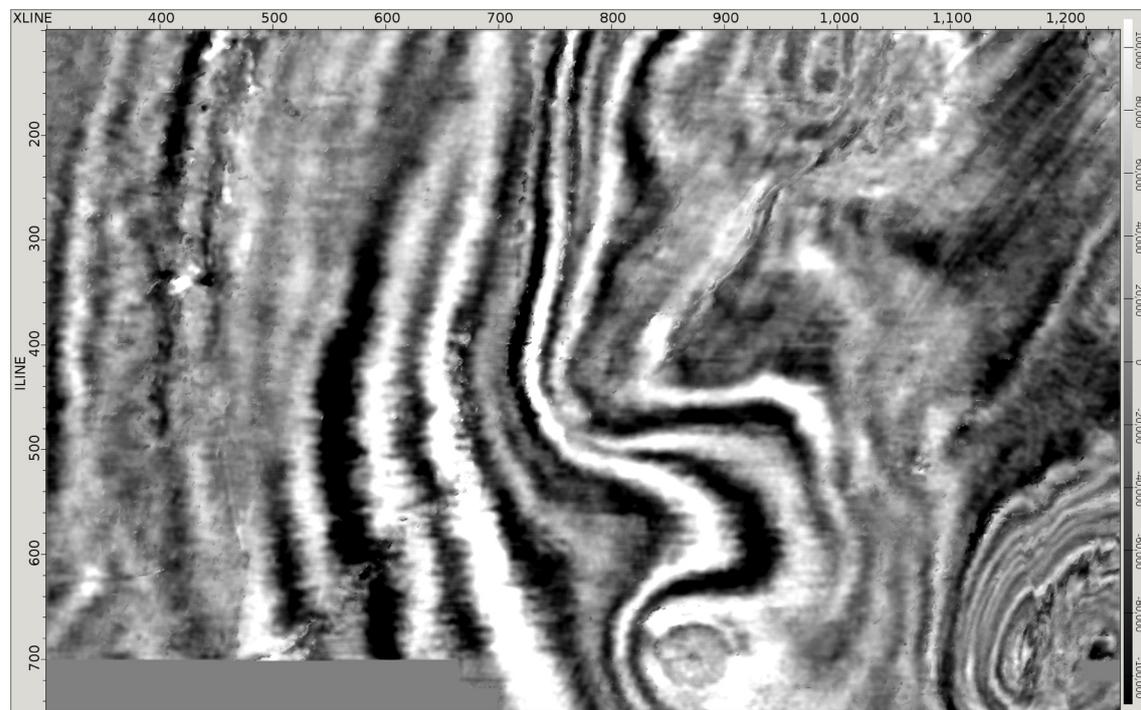
So far, we trained the neural network on the same data as we trained it for. In order to estimate how well the method generalizes, we apply it to the Volve data set described previously. We mentioned the choice of seismic attributes can vary and it is also possible to use the seismic data itself, without the estimation of pre-selected seismic attributes. We want to demonstrate that as well. Hence, we trained another neural network with the exact same architecture as before, except this time, we only have one input channel, the seismic data. The training data was, as before, the F3 offshore Netherlands block. Once the training of the autoencoder has been done, we fixed the according weights (filter) to extract the bottleneck layer and clustered it into 6 clusters and 64 clusters in order to estimate consistencies and differences, though we show only the six clusters here. Figure 12 shows three slices of the 3D volume where the dome-like structure, where the three slices intersect, contains a reservoir. The application of the clustering algorithm to the encoded features of the autoencoder is shown in Figure 13. We observe, that the area of the reservoir was clustered with a distinct color (green) compared to the rest of the data in the surrounding area. Therefore, the method found an anomaly in a particular cluster, which is also the only cluster which is identical when using 64 clusters. Whether this is just the case for this data set or can be observed in other reservoirs needs to be evaluated in future. Nevertheless, we find that most of the clustered structures coincide very well with the seismic. After careful evaluation, we found the gray cluster to be areas of no or low amplitude, black to be weaker events, blue to be events of negative amplitude, red events of positive amplitudes, yellow events of very strong positive amplitude and green events of very strong negative amplitudes.

DISCUSSION

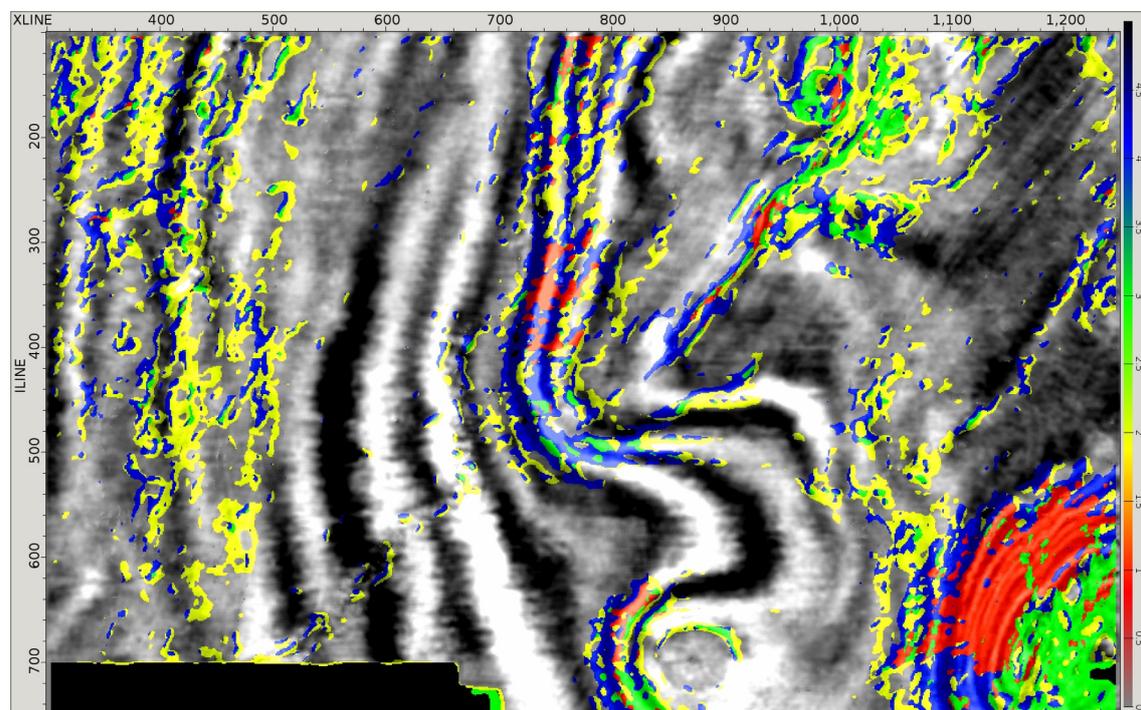
We suggested an automatic unsupervised workflow to label seismic data based on seismic attributes and seismic data directly. Although we limited the attributes involved and did not include the actual seismic itself in the first example, this is not a limitation of the method, as shown for the Volve data. We illustrated possible applications, though many more are possible. Furthermore, we demonstrated that a pre-trained autoencoder can generalize very well to new data sets. Due to the selection of the input, i.e. seismic attributes and seismic itself, our approach can be specialized in the interpretative use of interest. The choice of attributes, or whether to include seismic or not, leads to different clustering results. Therefore, the choice of the input determines which patterns the method aims to find, which is in control of the user. For tasks not shown in this work, a different set of seismic attributes and/or the seismic itself might be more suitable (Roden et al., 2015). We think our approach can save time and cost by accelerating interpretation and remove bias of interpreters (Bond et al., 2007). Furthermore, we believe an unbiased unsupervised scheme can find patterns humans and supervised methods might have missed or are not obvious immediately, like in case of the green structure in the Volve data.

Even though we got reasonable results, there is a lot of room for improvements. We used a single GPU with limited amounts of memory and time for the self training of the autoencoder and clustering algorithm. Our reconstructions were not perfect and might have missed important details due to insufficient training, which might improve our results significantly. However, a near perfect reconstruction at the training step might lead to overfitting and a worse generalization to other data. Furthermore, more sophisticated clustering algorithms might be able to find better relationships of patterns. In this work, we trained the clustering independently from the autoencoder. It is possible to do that simultaneously (Ghasedi Dizaji et al., 2017). Despite these limitations, we still got reasonable results without the effort and cost of providing manually interpreted labels for a supervised approach, as it is common in industry. The training and clustering for the F3 dataset using seismic attributes took approximately two days in total on a Nvidia GeForce GTX 1080 on a conventional workstation. Training just the seismic data clustering the Volve data set took approximately one day. Though the majority of the time is spent training, once the autoencoder is trained sufficiently, the application takes about an hour, in case of the Volve example.

We furthermore demonstrated, that the encoding of seismic attributes helps the clustering algorithm tremendously. Therefore, autoencoder networks can help established methods by providing more suitable input data, i.e. by denoising and incorporation of spatial patterns, as was the case in our example.



(a) Time slice at 640 ms



(b) Labeled time slice at 640 ms

Figure 10: Overlay of the migrated time slice at 640 ms of the F3 data set and its clustered results.

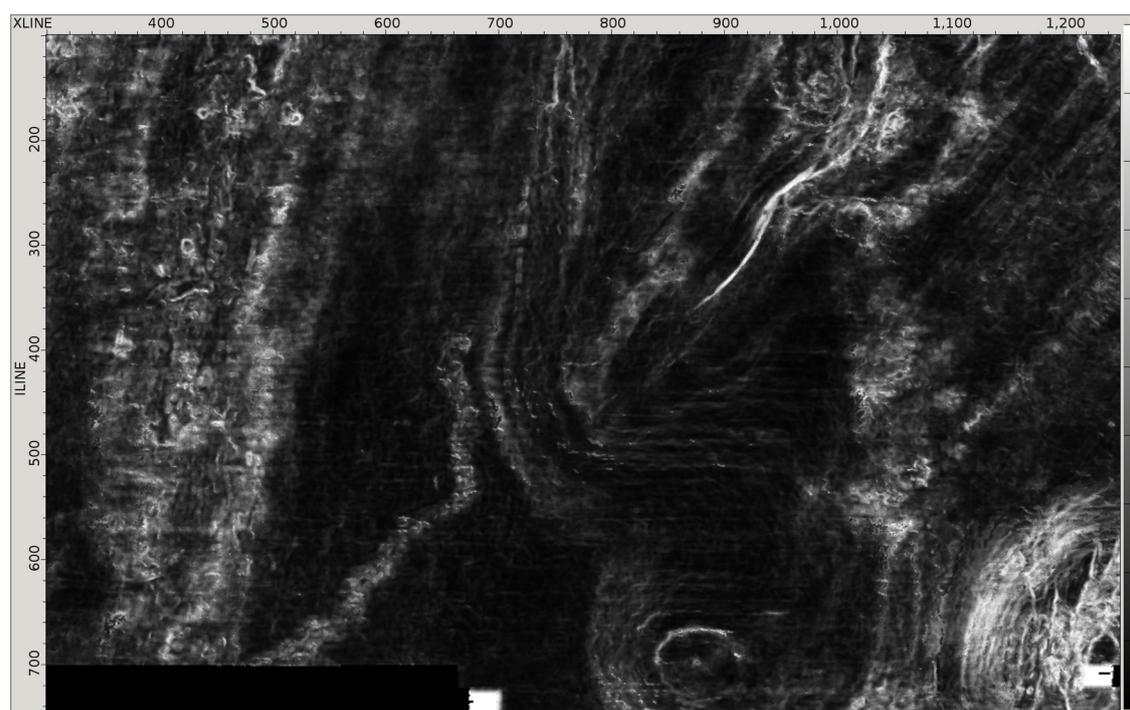


Figure 11: Discontinuity attribute extracted from the migrated data. This is part of the data the neural network actually sees, since we have not included the seismic itself in this example.

CONCLUSIONS

We presented a method to analyze sets of multiple seismic attributes and seismic data and extract a set of uncorrelated features. The method is based on a combination of traditional machine learning, in form of clustering algorithm, where we chose k-means, and deep learning. An unsupervised 3D convolutional autoencoder network is first used to extract the most salient information from the multiple attributes by minimizing the mean-square error between batches of input and reconstructed images. Based on the loss function, we considered the autoencoder found a data representation that contains all important features and stopped the training. We performed a manual quality control after the training on an exemplary in-line, to verify the network performed the expected task. The design of the autoencoder allows us to build a feature vector for each sample. Subsequently, we clustered each sample using the feature vector in six different classes by the k-means algorithm. All the steps, apart from converting the selected seismic attributes under investigation into a data tensor as required by the autoencoder network, were conducted without human interaction in an unsupervised fashion. The subsequent quality control is done manually, though not part of the actual algorithm. Human interpretation is required, when we try to assign labels or meaning to the six clusters, e.g. amplitudes in the Volve example.

We also showed, that seismic sections and labeled features correspond well. Although the network did not see the seismic sections in the first example, the labeled features and seismic show the same lateral pattern. In case of the Volve Field, we used the seismic data. It turns out the approach clusters the area of the reservoir in a cluster that is different from the surrounding area, highlighting the method can find areas of interest. Finally, we gave a rough interpretation of the labelled features. In case of the F3 example, where seismic attributes are clustered, the first class, displayed in green, we elucidated as primarily the salt body and related faults. The second class, displayed in red, we interpreted as tilted structures. The third class, displayed transparently, we interpret as mostly horizontal or slightly titled continuous horizons. Yellow indicates smaller discontinuities, whereas blue contains faults.

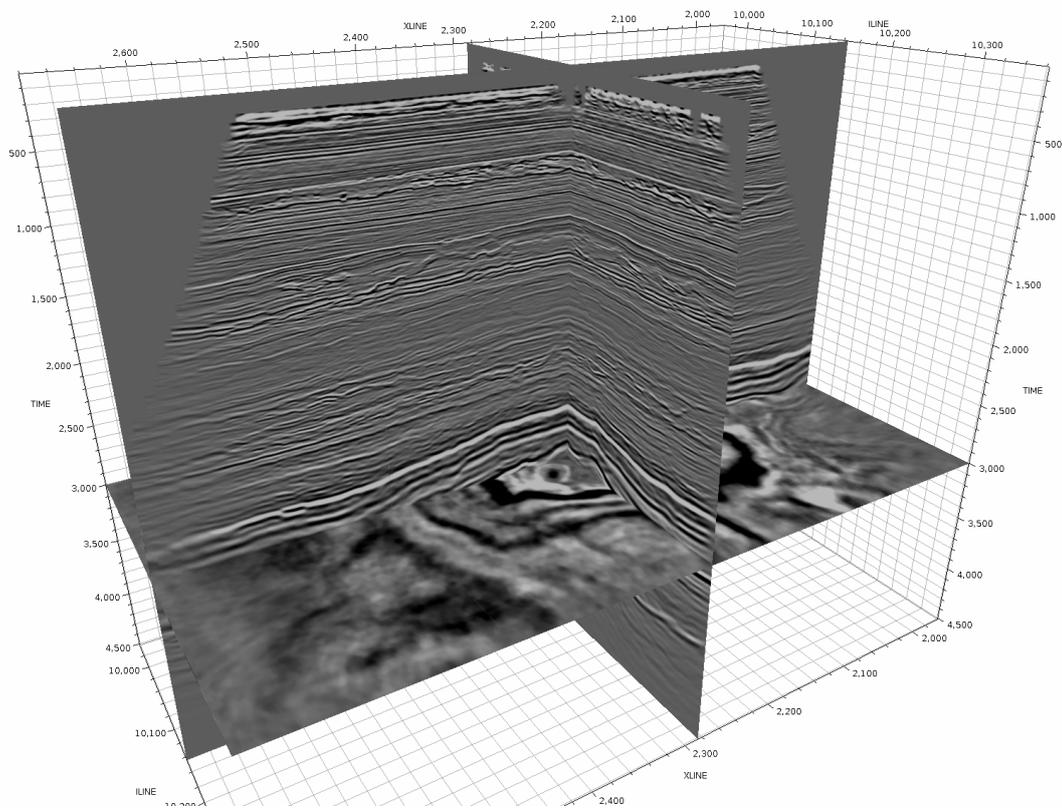


Figure 12: Depth migrated seismic of the Volve data set as provided by Equinor. The reservoir is located in the Hugin formation, in the dome-like structure where the three slices intersect.

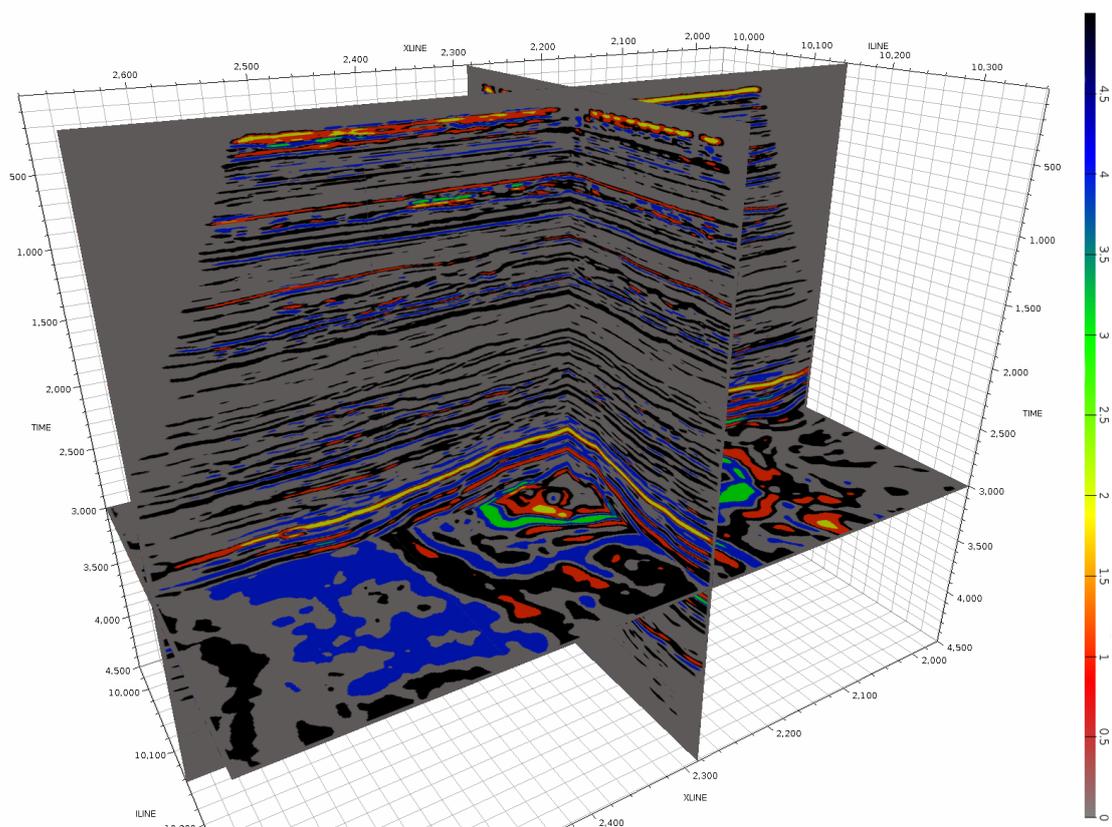


Figure 13: Result of the clustering of the encoded seismic data into six clusters. The clusters 1 and 2 (red and yellow) are positive amplitudes of differing strength, while 3 and 4 (green and blue) are negative amplitudes of varying strength. The cluster 0 (gray) clustered very weak amplitudes while 5 (black) clustered the weaker events. The reservoir is located in the area of the green anomaly in the center of the figure.

ACKNOWLEDGEMENTS

The work was supported by the sponsors of the Wave Inversion Technology (WIT) consortium and the Federal Ministry for Economic Affairs and Energy of Germany (project number 03SX427B). We thank Shearwater GeoServices Software, inc. for providing Reveal used for visualizations and Google for providing the Tensorflow API. Furthermore we thank dGB Earth Sciences for providing the Netherlands Offshore F3 Block data and Equinor for providing the Volve data set. We would like to thank the Applied Seismics Group Hamburg for continuous discussions and the associate editor A. Abubaker, two anonymous reviewer and in particular A. E. Barnes for the very valuable feedback that improved this manuscript tremendously.

REFERENCES

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283.
- Bakker, P., van Vliet, L. J., and Verbeek, P. W. (1999). Edge preserving orientation adaptive filtering. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 1, pages 535–540 Vol. 1.
- Barnes, A. E. (1996). Theory of 2-d complex seismic trace analysis. *GEOPHYSICS*, 61(1):264–272.
- Barnes, A. E. (2007). Redundant and useless seismic attributes. *GEOPHYSICS*, 72(3):P33–P38.
- Birkhäuser, P., Roth, P., Meier, B., and Naef, H. (2001). 3D Seismik: Räumliche Erkundung der mesozoischen Sedimentschichten im Zürcher Weinland. *Nagra technical report*, NTB 00-03:Nagra, CH-5430 Wettingen.
- Bond, C. E., Gibbs, A. D., Shipton, Z. K., and Jones, S. (2007). What do you think this is?"conceptual uncertainty" in geoscience interpretation. *GSA today*, 17(11):4.
- Brown, A. R. (2001). Understanding seismic attributes. *Geophysics*, 66(1):47–48.
- Chopra, S. and Marfurt, K. J. (2005). Seismic attributes — A historical perspective. *Geophysics*, 70(5):3SO–28SO.
- Chopra, S. and Marfurt, K. J. (2007). *Seismic Attributes for Prospect Identification and Reservoir Characterization*. Society of Exploration Geophysics, Tulsa.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.
- Dell, S., Tygel, M., and Gajewski, D. (2014). Image-ray tomography. *Geophysical Prospecting*, 62(3):413–426.
- Di, H., Wang, Z., and AlRegib, G. (2018). Deep convolutional neural networks for seismic salt-body delineation. In *AAPG Annual Convention and Exhibition*.
- Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern classification*. Wiley Interscience, New York.
- Equinor (2018). Volve Data Village. <https://data.equinor.com/dataset/Volvea>.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.
- Ghasedi Dizaji, K., Herandi, A., Deng, C., Cai, W., and Huang, H. (2017). Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5736–5745.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Guo, H. (2014). *Modern Mathematics and Applications in Computer Graphics and Vision*. World Scientific Publishing Company. <https://books.google.de/books?id=vCc8DQAAQBAJ>.
- Hale, D. (2012). Fault surfaces and fault throws from 3d seismic images. In *SEG Technical Program Expanded Abstracts 2012*, pages 1–6.
- Isaaks, E. H. and Srivastava, R. M. (1989). *An Introduction to Applied Geostatistics*. Oxford University Press, New York.
- Iversen, E., Tygel, M., Ursin, B., and de Hoop, M. V. (2012). Kinematic time migration and demigration of reflections in pre-stack seismic data. *Geophysical Journal International*, 189(3):1635–1666.

- Jäger, R., Mann, J., Höcht, G., and Hubral, P. (2001). Common-reflection-surface stack: Image and attributes. *Geophysics*, 66(1):97–109.
- Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480.
- MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.
- Marfurt, K. J., Kirlin, R. L., Farmer, S. L., and Bahorich, M. S. (1998). 3-D seismic attributes using a semblance-based coherency algorithm. *Geophysics*, 63:1150–1165.
- Overeem, I., Weltje, G. J., Bishop-Kay, C., and Kroonenberg, S. B. (2001). The late cenozoic eridasos delta system in the southern north sea basin: a climate signal in sediment supply? *Basin Research*, 13(3):293–312.
- Qian, F., Yin, M., Liu, X.-Y., Wang, Y.-J., Lu, C., and Hu, G.-M. (2018). Unsupervised seismic facies analysis via deep convolutional autoencoders. *Geophysics*, 83(3):A39–A43.
- Randen, T., Pedersen, S. I., and Sønneland, L. (2001). Automatic extraction of fault surfaces from three-dimensional seismic data. In *SEG Technical Program Expanded Abstracts 2001*, pages 551–554. Society of Exploration Geophysicists.
- Rasmussen, C. E. (2000). The infinite gaussian mixture model. In *Advances in neural information processing systems*, pages 554–560.
- Roberts, A. (2001). Curvature attributes and their application to 3d interpreted horizons. *First Break*, 19(2):85–100.
- Roden, R., Smith, T., and Sacrey, D. (2015). Geologic pattern recognition from seismic attributes: Principal component analysis and self-organizing maps. *Interpretation*, 3(4):SAE59–SAE83.
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.
- Ross, C. P. and Cole, D. M. (2017). A comparison of popular neural network facies-classification schemes. *The Leading Edge*, 36(4):340–349.
- Shafiq, M., Prabhushankar, M., and AlRegib, G. (2018a). Leveraging sparse features learned from natural images for seismic understanding. In *80th EAGE Conference and Exhibition 2018*.
- Shafiq, M. A., Prabhushankar, M., Di, H., and AlRegib, G. (2018b). Towards understanding common features between natural and seismic images. In *SEG Technical Program Expanded Abstracts 2018*, pages 2076–2080. Society of Exploration Geophysicists.
- Shi, Y., Wu, X., and Fomel, S. (2018). Automatic salt-body classification using a deep convolutional neural network. In *SEG Technical Program Expanded Abstracts 2018*, pages 1971–1975. Society of Exploration Geophysicists.
- Sørensen, J. C., Gregersen, U., Breiner, M., and Michelsen, O. (1997). High-frequency sequence stratigraphy of upper cenozoic deposits in the central and southeastern north sea areas. *Marine and Petroleum Geology*, 14(2):99 – 123.
- Taner, M. T., Koehler, F., and Sheriff, R. E. (1979). Complex seismic trace analysis. *Geophysics*, 44(6):1041.
- Wu, X., Shi, Y., Fomel, S., and Liang, L. (2018). Convolutional neural networks for fault interpretation in seismic images: 88th annual international meeting. In *SEG, Expanded Abstracts*, volume 2018, pages 1946–1950.