Automatic high-resolution microseismic event detection via supervised machine learning

Qu, Shan; Guan, Zhe; Verschuur, Eric; Chen, Yangkang

DOI
10.1093/gji/ggaa193

Publication date
2020

Document Version
Final published version

Published in
Geophysical Journal International

Geophysical Journal International

Geophys. J. Int. (2020) **222,** 1881–1895 doi: 10.1093/gji/ggaa193
GJI Seismology

Automatic high-resolution microseismic event detection via supervised machine learning

Shan Qu,[1] Zhe Guan,[2] Eric Verschuur[1] and Yangkang Chen[3]

[1] *Delft university of technology, Departments of Imaging Physics, Mekelweg* 2, 2628 *CD Delft, Netherlands*
[2] *Rice University, Applied Physics Program, Houston, TX, USA*
[3] *School of Earth Sciences, Zhejiang University, Hangzhou, Zhejiang Province* 310027*, China. E-mail: chenyk2016@gmail.com*

Accepted 2019 June 7. Received 2019 May 22; in original form 2019 February 4

SUMMARY

Microseismic methods are crucial for real-time monitoring of the hydraulic fracturing dynamic status during the development of unconventional reservoirs. However, unlike the active-source seismic events, the microseismic events usually have low signal-to-noise ratio (SNR), which makes its data processing challenging. To overcome the noise issue of the weak microseismic events, we propose a new workflow for high-resolution microseismic event detection. For the preprocessing, fix-sized segmentation with a length of 2∗*wavelength* is used to divide the data into segments. Later on, 191 features have been extracted and used as the input data to train the support vector machine (SVM) model. These features include 63 1-D time/spectral-domain features, and 128 2-D texture features, which indicate the continuity, smoothness, and irregularity of the events/noise. The proposed feature extraction maximally exploits the limited information of each segment. Afterward, we use a combination of univariate feature selection and random-forest-based recursive feature elimination for feature selection to avoid overfitting. This feature selection strategy not only finds the best features, but also decides the optimal number of features that are needed for the best accuracy. Regarding the training process, SVM with a Gaussian kernel is used. In addition, a cross-validation (CV) process is implemented for automatic parameter setting. In the end, a group of synthetic and field microseismic data with different levels of complexity show that the proposed workflow is much more robust than the state-of-the-art short-term-average over long-term-average ratio (STA/LTA) method and also performs better than the convolutional-neural-networks (CNN), for this case where the amount of training data sets is limited. A demo for the synthetic example is available: https://github.com/shanqu91/ML_event_detection_microseismic.

Key words: Event detection; High-resolution; Machine learning; Microseismic; SVM; 2-D texture features.

INTRODUCTION

It has been well known that microseismic monitoring plays an important role in characterizing physical processes related to fluid injections and extractions in hydrocarbon and geothermal reservoirs (Shapiro *et al.* 2006; Vera Rodriguez *et al.* 2012; Xia *et al.* 2013). In general the microseismic data are recorded by downhole or buried, shallow surface geophone arrays, which offer the significant advantages of being sufficiently close to the fracture and being unaffected by the free surface (Warpinski 2000). However, the energy stimulated from the hydraulic fracturing is usually extremely weak. As a result, the weak signal is easily overwhelmed by the background noise, which may lead to unauthentic arrival time-picks and localization of microseismic events when no proper denoising algorithms or event detection techniques are applied. Therefore, prior to the localization and mechanism analysis of the source, the identification and detection of microseismic events or applying a reliable and effective denoising process become important challenges (Forghani-Arani *et al.* 2013; Mousavi & Langston 2016a,b; Liu *et al.* 2016b; Guan & Niu 2017).

The state-of-the-art denoising algorithms include transforming domain thresholding methods (Candes *et al.* 2006), singular spectrum analysis (Vautard *et al.* 1992), low-rank-approximation-based methods (Huang *et al.* 2016), dictionary-learning-based methods (Elad & Aharon 2006), empirical-mode-decomposition and empirical-mode-decomposition-like methods (Huang *et al.* 1998), etc. Denoising microseismic data will inevitably cause useful small-amplitude signal damage, which degrades the fidelity of the processed data (Li *et al.* 2016; Huang *et al.* 2017a; Zhang *et al.* 2019). Moreover, the damaged waveform amplitude will greatly affect the

subsequent source localization and mechanism analysis (Maxwell *et al.* 2010). Considering the potential disadvantages caused by denoising, a robust event detection method is a strong demand in the microseismic community.

Traditional event detection is based on energy analysis (Hatherly 1982; Allen 1978; Vaezi & Van der Baan 2015), which is a widely used method due to few assumptions about the data based on some statistic criterion. For example, a popular criterion is the short-term average over long-term average (STA/LTA) ratio, in which the ratio of the continuously calculated average energy of the data in two consecutive moving-time windows, a short-term window and a subsequent long-term window, is used as a statistical criterion (Allen 1982, 1978; Vaezi & Van der Baan 2015). However, this method has some disadvantages. It requires a careful setting of parameters (threshold and window lengths, Trnkoczy 1999). Moreover, these algorithms are sensitive to sudden amplitude increases, therefore the noise whose energy is comparable to or greater than microseismic events may be detected as a microseismic event (Withers *et al.* 1998). In order to mitigate this noise issue, another event detection method that is based on template matching was proposed by Gibbons & Ringdal (2006). It takes advantage of pre-determined events, known as the master event, and cross-correlates them with continuous recordings to detect events with high similarities (Gibbons & Ringdal 2006; Song *et al.* 2010; Senkaya & Karsli 2014). The template-matching-based method is sensitive to small amplitude events and therefore a typical way to detect weak events in earthquake seismology (Song *et al.* 2010) even in the presence of high background noise. These detection methods are especially useful to lower the detection threshold and increase the detection sensitivity. Michelet & Toksöz (2007) and Arrowsmith & Eisner (2006) have also shown that these methods can be effective as long as the separation between the master event and target event is less than the dominant wavelength. Gelchinsky & Shtivelman (1983) proposed a hybrid method that combines the benefits of template-matching-based methods and the energy-analysis-based methods. However, the template-matching-based method requires a master (or known) event as an input, which is not always available. In addition, it is limited to detect events that are similar to the master event, which means it might have a high false-negative rate, and is a computationally expensive method.

In recent years, some researchers have already done investigations on supervised machine-learning-based event detection or event picking (McCormack *et al.* 1993; Knapmeyer-Endrun & Hammer 2015; Akram *et al.* 2017; Provost *et al.* 2017; Rouet-Leduc *et al.* 2017; Zheng *et al.* 2017; Zhao & Gross 2017; Chen 2018c; Mousavi *et al.* 2018a,b; Perol *et al.* 2018; Zhu & Beroza 2018; Dokht *et al.* 2019) Zhao & Gross (2017) trained a support vector machine (SVM) model with 1-D features of the segments to distinguish microseismic from noise events. However, these methods require a longer length of segmentation ($\sim 15 * wavelength$) for providing sufficient information for each segment to provide a stable prediction. The results, therefore, have a coarse vertical resolution. Chen () proposed a microseismic picking algorithm based on unsupervised machine learning that utilizes fuzzy clustering to identify signal onsets. As shown in his experiment, this clustering-based method is sensitive to the noise level. When the noise level becomes extremely strong, the clustering method may make some mistakes. Zheng *et al.* (2017), Mousavi *et al.* (2018a,b), Zhu & Beroza (2018), Perol *et al.* (2018), Chen *et al.* (2019c) and Dokht *et al.* (2019) have showed successful and promising performances of deep learning for the purpose of event detection. However, the deep-learning-based seismic event detection methods usually require much larger training data sets compared to the traditional machine-learning-based methods like SVM (LeCun *et al.* 2015; Li *et al.* 2019).

In this work, we propose a new workflow for high-resolution microseismic event detection. Details of the workflow are presented step by step: (1) fix-sized segmentation, with a length of $2 * wavelength$, is used to divide the data into segments; (2) 191 features have been extracted in total, including 63 1-D time/spectral-domain features, and 128 2-D texture features indicating the continuity, smoothness, and irregularity of the events/noise; (3) a combination of univariate feature selection and random-forest-based recursive feature elimination is implemented for feature selection, which not only finds the best features but also the number of features that are needed for the best accuracy; (4) a C-SVM model, where the 'C' represents a coefficient used to control the tolerance of error item, is considered in the essential training process. In addition, a cross-validation (CV) process is implemented for the automatic parameter setting and (5) the trained model is then applied to detect events of the test data. In the end, results obtained on a group of synthetic and real microseismic data with different levels of complexity show that the proposed workflow is more robust than the state-of-the-art STA/LTA method and also performs better than the CNN, for our case when the amount of training data sets is limited. Note that this paper is an extended version of work published in Qu *et al.* (2018).

MICROSEISMIC EVENT DETECTION AS A CLASSIFICATION PROBLEM

The proposed workflow for microseismic event detection can be summarized as the following steps: (1) segmentation and labelling; (2) feature extraction and normalization; (3) feature selection; (4) support vector classification and (5) test on new data. To clearly demonstrate the whole workflow for event detection, a group of synthetic microseismic data sets is used. The synthetic microseismic data sets are simulated from the three-layer velocity model shown in Figs 1(a) and (b). The modelled training data with $SNR = -13$ is shown in Fig. 2(a). The modelled test data sets are displayed in Figs 5(a) and 6(a), including $SNR = -10$ and $SNR = -13$ level of noise energy, respectively. The definition of noise energy is as follows:

$$SNR_{dB} = 10log_{10}\left(\frac{P_{signal}}{P_{noise}}\right), \quad (1)$$

where P is the average power. In all the data sets, the noise level is much stronger than the signal level. The receivers are located along the full surface with a spacing of 7.5 m and a time duration is 3.1 s. In addition, five different traces of clean and noisy test data 2 ($SNR = -10$) are demonstrated in Fig. 7. We can see that the signal is masked by the strong background noise and hard to detect on a single trace.

Segmentation and labelling

Segmentation is a very important preprocessing stage, where the microseismic data are split into segments. Fix-sized segmentation is used in this work and we set the length of each segment as $2 * wavelength$, which is 0.058 s in the synthetic example. In this way, the vertical resolution of event detection results is higher, however, each segment contains very limited information, which makes the problem tougher to solve.

After the segmentation, the training data segments are labelled into two classes: events and noise. For the synthetic example, a total

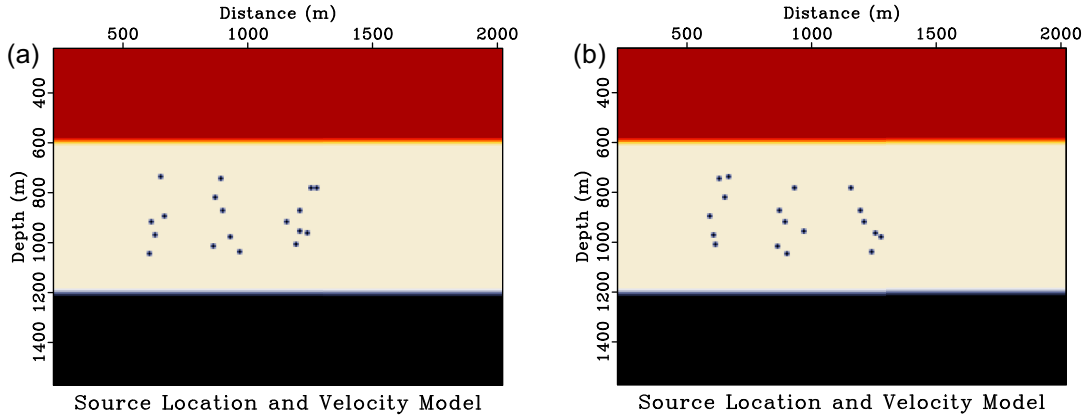

Figure 1. Synthetic example: the geometry and velocity model for the modelling of (a) raw training data with $SNR = -13$ dB, (b) raw test data.

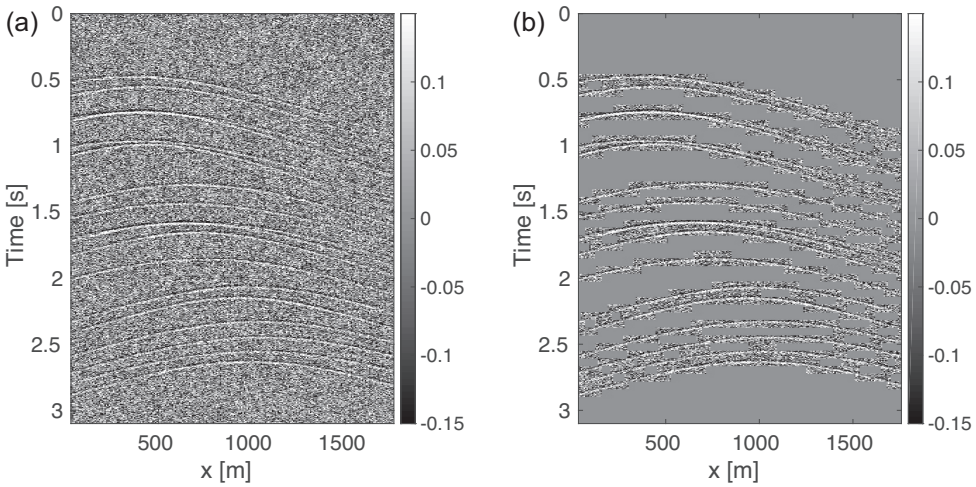

Figure 2. Synthetic example: (a) raw training data, (b) labelled training data.

of 12 960 segments are extracted, including 4853 segments containing a microseismic event and 8107 segments of noise. The labelled training synthetic data is shown in Fig. 2(b). As is well-known, supervised classification is largely dependent on the labelled training data sets, which are usually done based on different criteria from different users. Please note that we will discuss how using different labelling criteria affects the final prediction in the 'Discussion' section.

Feature extraction and normalization

The purpose of feature extraction is to convert all the segments into relevant features, which are served as input training vectors for the classification. The dimension of the data is reduced in the feature extraction step, which improves the classifier's performance. Many researchers have already done investigations on the feature extraction for seismic event detection (Mousavi *et al.* 2016; Zhao & Gross 2017). They extracted 1-D features of the segments in time, frequency and time–frequency domains. However, as we have mentioned, due to the high resolution of segmentation, each segment contains very limited information. As a result, only extracting 1-D features is not enough. In order to maximize the information per segment, we propose to extract both 1-D time/frequency-domain features and 2-D texture features for each segment. 191 features have been extracted, including 63 1-D features, and 128 2-D features. The 1-D features consist of both time-domain features and spectral features and are listed in Table 1 with description. By only considering the 1-D features of the seismic data, the 2-D features (for example, continuity, smoothness and irregularity of the events) are ignored, which is obviously a waste of information. Therefore, 128 extra 2-D texture features are considered in our feature extraction. he microseismic data are first converted into a grey-scale image. After that, local grey-level co-occurrence matrices (GLCM) in a moving window are calculated. The GLCM characterizes the texture of an image by calculating how often pairs of pixel with specific values and in a specified spatial relationship occur in an image (Haralick *et al.* 1973). Certain features that characterize texture properties of the image are then calculated from this matrix, which are *Contrast*, *Correlation*, *Energy* and *Homogeneity*. In addition, orientations 0°, 45°, 135° and 90° and distances of $1-8$ neighbouring voxels are considered. Please note that a range of orientations and distances is considered here to make the feature extraction process more general, however, causes feature redundancy, which will be discussed in more detail in Section 3. Feature selection. The calculated 2-D texture parameters within the moving window are severed as the features of the segment in the centre of the window. Details of the texture features are described in Table 2. Part of the 2-D texture features extracted from the training data are demonstrated in Fig. 3, being the *Contrast*, *Correlation*, *Energy* and *Homogeneity*

Table 1. The list of the extracted 1-D time/spectral-domain feature.

ID	Feature name	Description
1	Mean	
2	Median	
3	STD	Standard deviation
4	MAD	Median absolute deviation
5	25th percentile	The value below which 25 PER CENT of observations fall
6	75th percentile	The value below which 75 PER CENT of observations fall
7	Inter quantile range	The difference between 25th percentile and 75th percentile
8	Skewness	A measure of symmetry relative to a normal distribution
9	Kurtosis	A measure of whether the data is heavy- or light-tailed relative to normal distribution
10	Zero-crossing rate	
11	Energy	The sum of squares of the signal values
12	Entropy of energy	The entropy of normalized energies, a measure of abrupt changes
13–25	MFCC	Mel Freq. Cepstral Coef., form a cepstral representation where the freq. bands are not linear but distributed according to the mel-scale
26	Dominant freq. magnitude	The energy of a spectrum is centred upon
27	Spectral Centroid	Indice of the dominant freq.
28	Spectral Spread	The second central moment of the spectrum
29	Spectral Entropy	Entropy of the normalized spectral energies
30	Spectral Roll-off	The freq. below which 85 PER CENT of the total spectral energy lies
31	RMS energy	Root-mean-square energy
32	Spectral bandwidth	The 2rd order spectral bandwidth
33–36	Polynomial features	Coef. of fitting an 3rd-order polynomial to the spectrum
37–48	Chroma vector	A 12-element feature vector indicating how much energy of each pitch class is present in the data
49	Chroma Deviation	The STD of the 12 chroma coef.
50-56	Spectral contrast	It considers the spectral peak, the spectral valley, and their difference in each freq. sub-band
57	Spectral flatness	A measure to quantify how much noise-like a sound is (High value indicates the spectrum is similar to white noise)
58–63	Tonnetz	The tonal centroid features

Table 2. The list of the extracted 2-D texture features.

ID	Feature name	Description
64–95	Contrast	Measures the local variations in the GLCM, for 0°, 45°, 135° and 90° orientation, with distance of 1−8 of neighbouring voxels
96–127	Correlation	Measures the joint probability occurrence of the specified pixel pairs in the GLCM, for 0°, 45°, 135° and 90° orientation, with distance of 1−8 of neighbouring voxels
128–159	Energy	Provides the sum of squared elements in the GLCM. Also known as uniformity or the angular second moment, for 0°, 45°, 135° and 90° orientation, with distance of 1−8 of neighbouring voxels
160–191	Homogeneity	Measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal, for 0°, 45°, 135° and 90° orientation, with distance of 1−8 of neighbouring voxels

of GLCM, for orientations 0°, 45°, 135° and 90° , with a distance of three neighbouring voxels. In this figure, we can see that the 2-D features can properly indicate most of the events, even when the noise level is high.

After feature extraction, feature normalization, which is used to standardize the range of independent features of the data, is a common requirement for most machine learning estimators. Without standardization, the estimators might behave badly. We normalize the features by removing the mean and scaling to unit variance in this work.

Feature selection

In machine learning, feature selection is the process of selecting a subset of most relevant features for the use in model training. It can reduce overfitting, as well as the training time. Since we extract the 2-D features with a range of orientation and distance for the sake of generalization, there exist highly correlated 2-D feature clusters, as shown in Fig. 4(a). In this simple synthetic scenario, many 2-D features share similar values with each other. Therefore, those clustering features provide redundant information and feature selection is needed to compensate this side-effect.

Univariate feature selection is a simple technique where a statistical test is applied to each feature individually to determine the strength of the relationship of the feature with the outcome variable. One simple criterion is the F-value of ANOVA (Analysis of variance) (Scheffe 1967). We choose the 30 per cent most significant features in this case. The corresponding F-value as a function of feature ID is shown in Fig. 4(b), in which we can see that the 2-D texture features are informative features and show large relevance with respect to different classes.

Univariate feature selection is simple to run and relatively good at gaining a better understanding of data. However, it does not reveal mutual information among features (Chen & Lin 2006). Random forest (RF) is a classification method and it also provides the branch weights that can represent feature importance (Breiman 2001). A

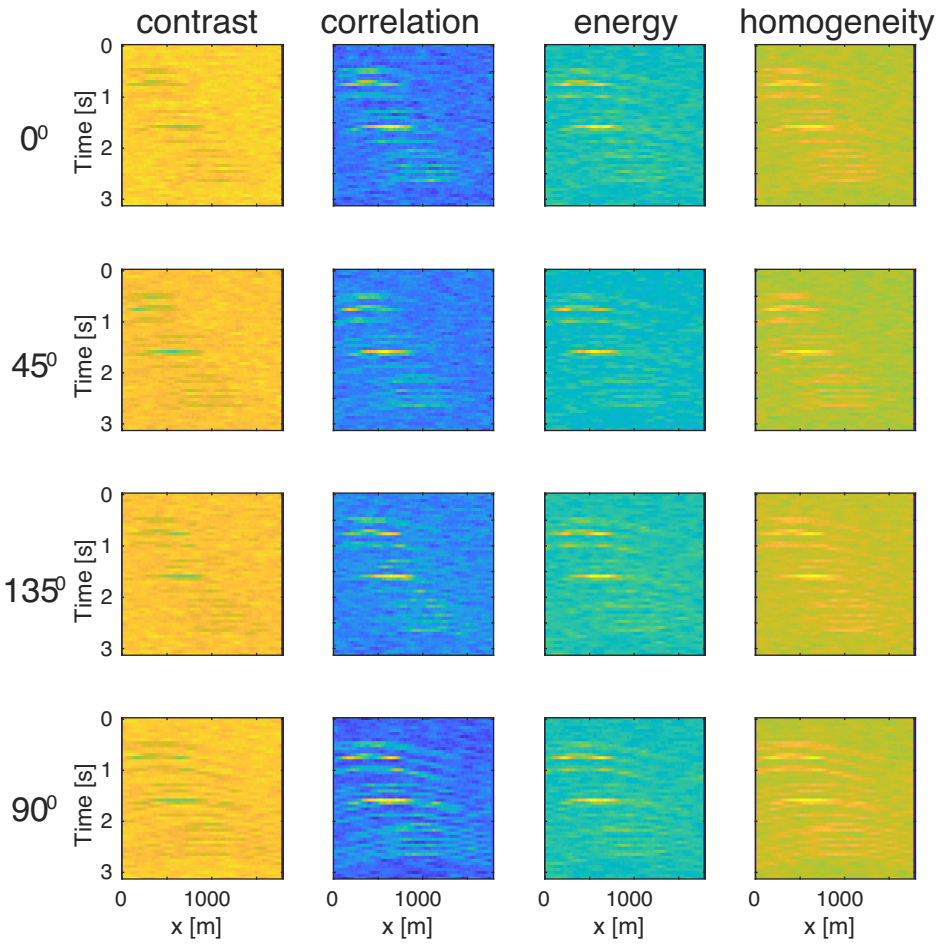

Figure 3. Synthetic example: 2-D texture features of the training data: Contrast, Correlation, Energy and Homogeneity, for orientations 0°, 45°, 90° and 135°, with a distance of three neighbouring voxels.

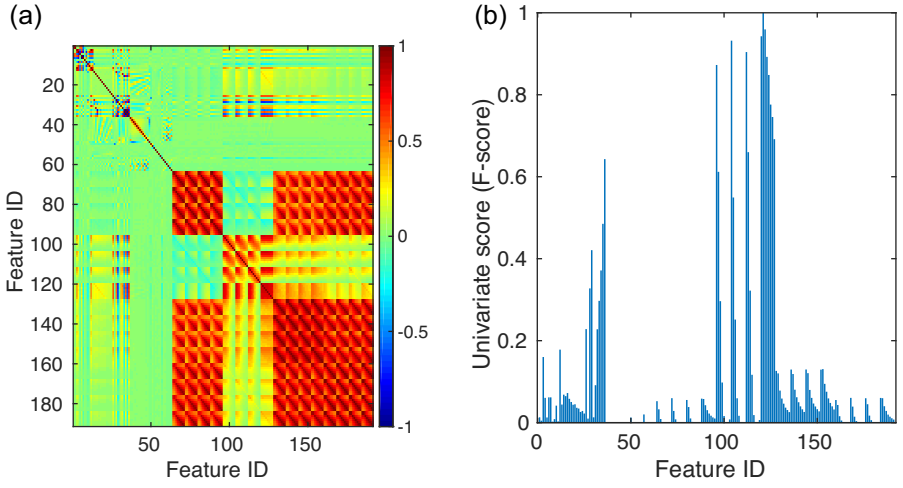

Figure 4. Synthetic example: (a) correlation matrix of the 191 1-D and 2-D features. Note that 2-D features (I-D range 64–191) are correlated due to the nature of the GLCM characteristics, (b) the univariate score (*F*-value) as a function of feature ID. The feature IDs are expained in Tables 1 and 2.

forest consists of a number of decision trees, each of which is constructed with randomly sampled features. Every node in the trees is designed to split the training sets into two parts, therefore similar response values end up in the same set. For one feature, we randomly permute its values in the second data set and obtain another accuracy. The difference between the two numbers can indicate the feature importance. RF is robust easy to use and has relatively good accuracy, which makes it an appealing tool for

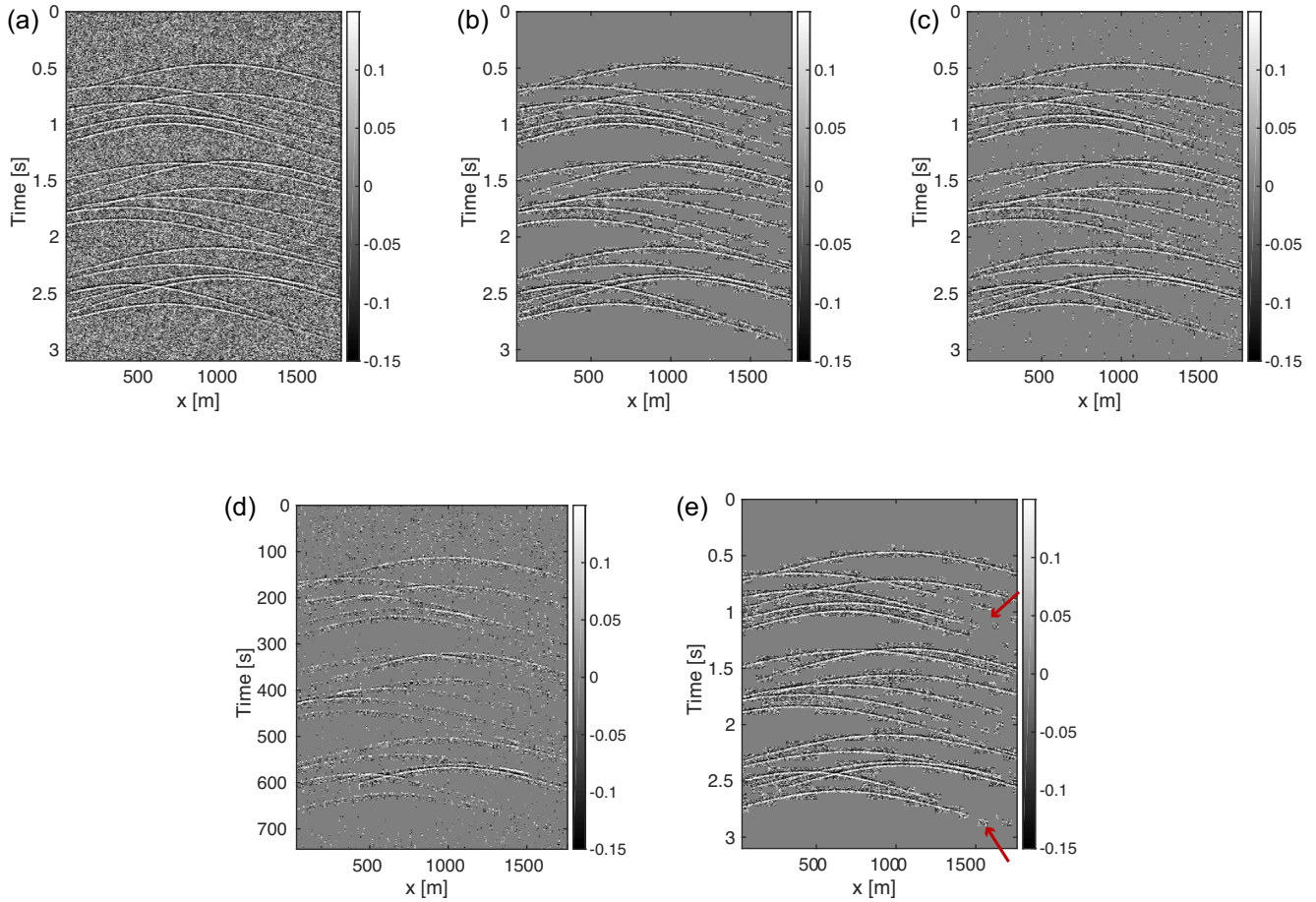

Figure 5. Synthetic example: (a) raw test data 1 ($SNR = -10$ dB); predicted event detection for test data 1 ($SNR = -10$ dB) using (b) both 1-D and 2-D features, (c) only 1-D features, (d) a conventional LTA/STA method and (e) a CNN approach. The arrows point at locations where the CNN fails.

feature selection. However, RF cannot handle too many features (Chen & Lin 2006). Therefore, a combination of univariate feature selection and Random forest is a good choice. In practice, we first use univariate feature selection to reduce the number of features, then apply random-forest-based recursive feature elimination to further find the optimal number of features in a cross-validation loop. In the end, 51 features are selected in the synthetic example.

Support vector classification

Recently, SVM has been an effective classification method by constructing hyperplanes with a maximal margin in a multidimensional space, which separates different cases of different class labels (Boser *et al.* 1992; Cortes & Vapnik 1995). Therefore, we choose SVM as the machine learning algorithm in our proposed workflow.

To construct an optimal hyperplane, SVM employs an iterative training algorithm, which is used to minimize an error function. In this work, given training vectors $x_i \in R^n$, $i = 1, ..., N$ in two classes, and a vector of labels $y_i \in \{1, -1\}$, we use the C-SVM model (Hsu *et al.* 2003), where a coefficient C is used to control the tolerance of the systematic outliers that allows fewer outliers to exist in the opponent class. This model solves a quadratic optimization problem:

$$\min_{\omega,b,\xi} = \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{N}\xi_i \tag{2}$$

subject to the constraints:,

$$y_i\left(\omega^T\phi\left(x_i\right)+b\right) \geq 1-\xi_i \text{ and } \xi_i \geq 0, i = 1, ..., N,$$

where ω represents the normal vector to the hyperplane, b is a constant and C is a penalty parameter on the training error, which is chosen to avoid overfitting. Note that ξ_i is the smallest non-negative number satisfying $y_i\left(\omega^T\phi\left(x_i\right)+b\right) \geq 1-\xi_i$. The kernel ϕ is used to transform the input data into the feature space.

By solving for the Lagrangian dual of the primal problem in eq. (2), a simplified problem is obtained:

$$\max_{\mathbf{a}} = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j K\left(x_i, x_j\right), \tag{3}$$

$$\sum_{i=1}^{N} a_i y_i = 0, C \geq a_i \geq 0, \; i = 1, ..., N, \; j = 1, ..., N,$$

where, $\mathbf{a}$ is introduced by Lagrangian multiplier. This dual formulation only depends on dot-products of the features, which is the kernel function $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$, to map into the higher dimensional feature space by transformation ϕ. The radial basis

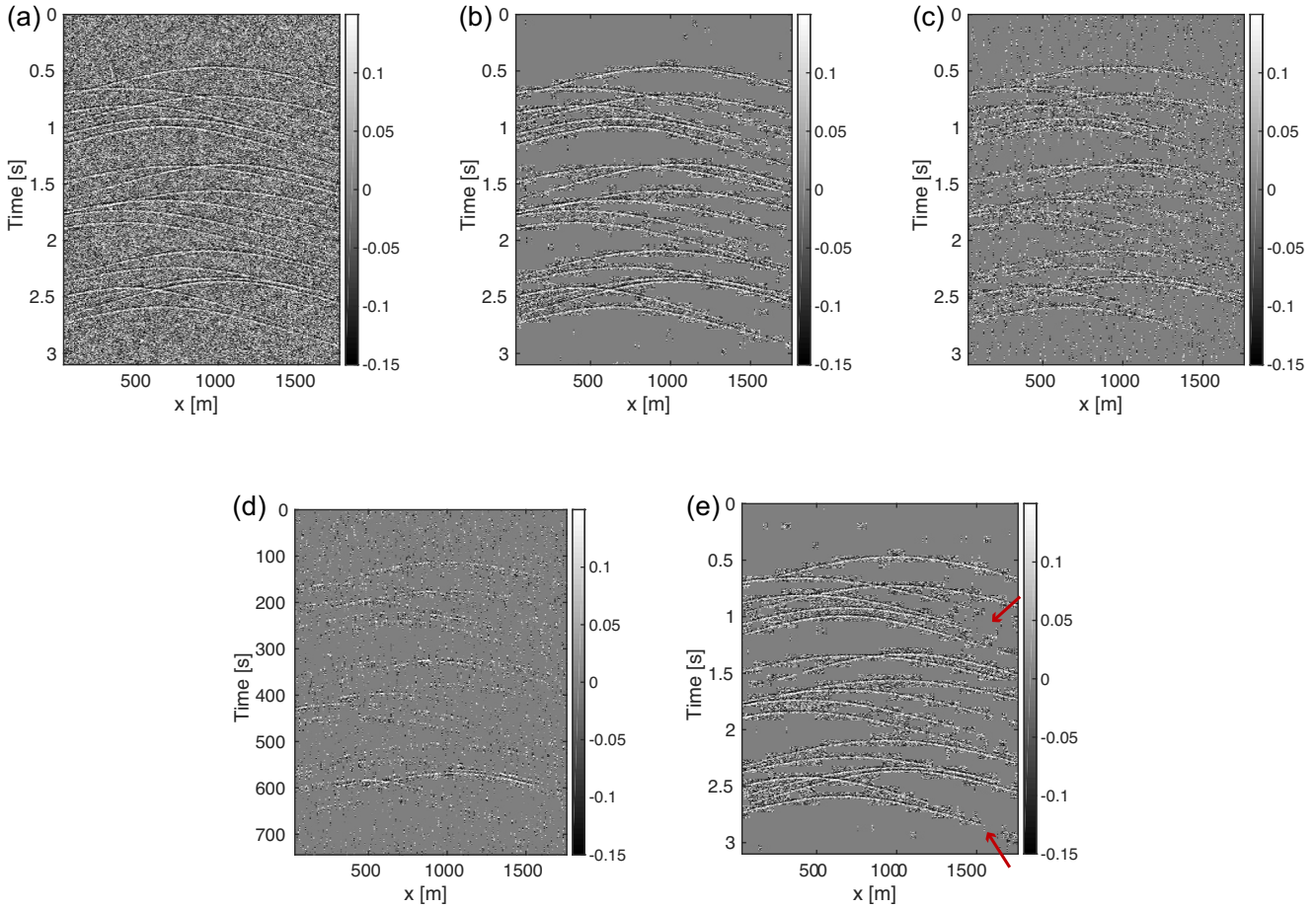

Figure 6. Synthetic example: (a) raw test data 2 ($SNR = -13$ dB); predicted event detection for test data 2 ($SNR = -13$ dB) using (b) both 1-D and 2-D features, (c) only 1-D features, (d) a conventional LTA/STA method and (e) a CNN approach. The arrows point at locations where the CNN fails.

function (RBF) in eq. (3) is used as the kernel function for SVM:

$$K\left(x_i, x_j\right) = \exp\left(-\gamma ||x_i - x_j||^2\right), \ i = 1, ..., N, j = 1, ..., N, \quad (4)$$

where, γ is an adjustable parameter of certain kernel functions. In this case, we set it as $1/N$. With the C-SVM model, there is only one parameter to be determined: C, which tells the SVM optimization how much you want to avoid mis-classifying the data. We conduct a cross-validation (CV) process to decide it. Considering a grid space of $\{C\}$ with $\log_2 C \in \{-3, -2.5, ..., 2.5, 3\}$, we apply fivefold CV on the training data to each C, and then choose the specific C that leads to the lowest CV balanced error. In the synthetic example, $C = 2.1544$ is selected with a score of 0.96. In addition, the size of the noise class is normally different from the size of the event class. This data imbalance could lead to bias. In order to compensate this, we adjust the weights inversely proportional to the class frequencies in the input data.

Test on new data

After obtaining the trained SVM model, we apply it to the test data 1 and 2 in Figs 5(a) and 6(a). The predicted event detection results considering both 1-D and 2-D features are shown in Figs 5(b) and 6(b) and the results considering only 1-D features are shown in Figs 5(c) and 6(c). We can see that, when the noise level is $SNR = -10$ dB, both of them result in a reasonable prediction with 95 per cent (Fig. 5b) and 90 per cent (Fig. 5c) accuracy, respectively. There is an obvious improvement in the prediction accuracy by considering 2-D texture features. When the noise level reaches $SNR = -13$ dB, the predicted result in Fig. 6(c) using only 1-D features is quite noisy with a prediction accuracy of 82 per cent. However, by maximizing the information per segment with extracting extra 2-D texture features, the proposed workflow still ends up with a reasonably good result with 93 per cent accuracy (Fig. 6b). Furthermore, five different traces (at 150, 525, 900, 1275 and 1650 m) of clean, noisy, and predicted detection of test data ($SNR = -10$ dB) are shown in Fig. 7. It can be seen that the events hidden in strong ambient noise can also be detected using our proposed workflow.

Moreover, we also compare the proposed workflow to a state-of-the-art STA/LTA method and a convolutional-neural-network approach (CNN). The STA/LTA parameter is measured in the time domain and defined as follows:

$$\begin{aligned} STA(i) &= \frac{1}{NSTA} \sum_{j=i-NSTA}^{i} d(j), \\ LTA(i) &= \frac{1}{NLTA} \sum_{j=i-NLTA}^{i} d(j), \\ R_{STA/LTA}(i) &= \frac{STA(i)}{LTA(i)}, \end{aligned} \quad (5)$$

where $d(i)$ denotes the input microseismic data and $NSTA$ and $NLTA$ denote short-term and long-term periods, respectively. We use $NSTA = 2{*}wavelength$ and $NLTA = 8{*}wavelength$ in this example. The

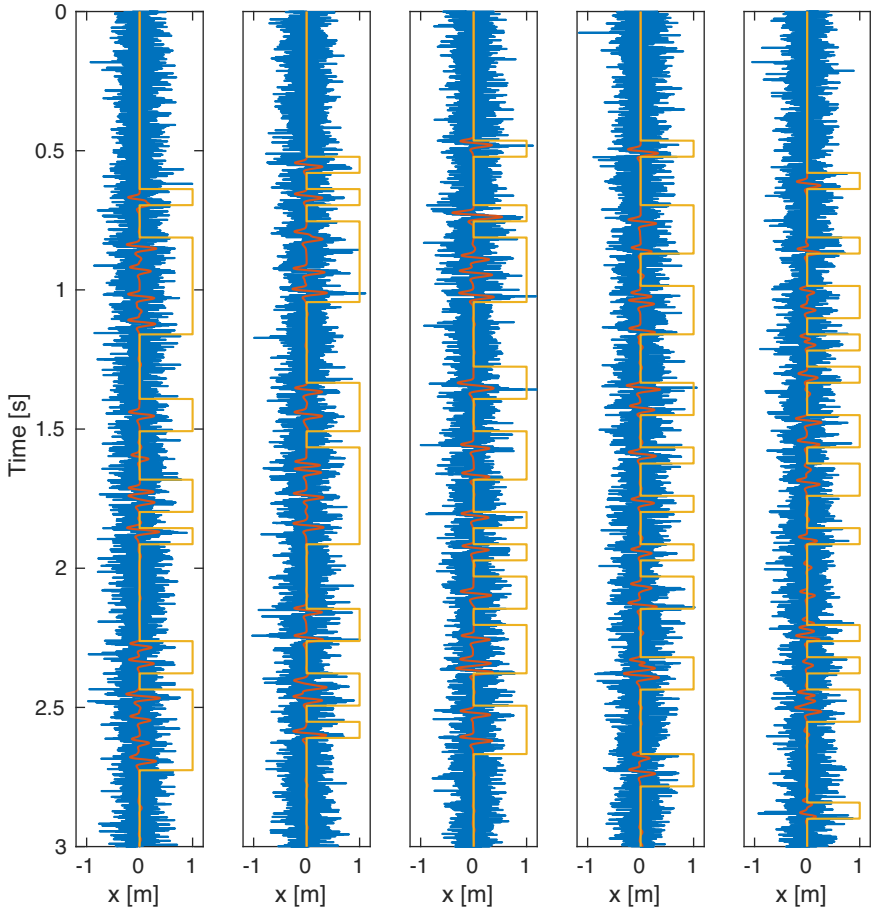

Figure 7. Synthetic example: Five traces (at 150, 525, 900, 1275 and 1650 m) of clean (red) and noisy (blue) test data 2 with $SNR = -13$ dB and the predicted event detection (yellow; 0-noise, 1-event).

Table 3. Classification metrics for test data 2 (Fig. 6a).

Parameters	SVM+1-D/2-D features	SVM+1-D features	CNN
Precision	0.93	0.82	0.91
Recall	0.92	0.82	0.90
F1-score	0.92	0.82	0.90

results using the STA/LTA method are shown in Figs 5(d) and 6(d). It is obvious that the STA/LTA method cannot perform well when strong noise exists. It is worth mentioning that the STA/LTA method is usually implemented after an initial denoising process applied to the raw data.

Regarding the CNN, we design a six-layer architecture, which is adopted from LeNet (LeCun *et al.* 1990): two convolutional layers with 32 kernels (3×3) to learn the local features; followed by one max pooling layer with (10×10) to reduce the number of parameters; after a flattening process, two fully connected layers with 128 kernels are included and a softmaxing layer is added in the end to generate the final classification. The results using this CNN are shown in Figs 5(e) and 6(e). We can see that it works well in detecting events, even though not as good as the proposed workflow in Figs 5(b) and 6(b). Some useful events, which are pointed with red arrows, are damaged. It is well-known that CNN is not able to show its privilege over the traditional machine learning algorithms when very limited training data are available.XXXX

The classification metrics of different strategies for the test data 2 ($SNR = -13$ dB) are written in Table 3

FIELD DATA EXAMPLE

We consider a group of surface-recorded microseismic data in a field data example. The receiver spacing is 7.5 m and the time duration is 2 s. Since the STA/LTA method cannot perform well without a denoising preprocess applied to the raw data, we only compare SVM with only 1-D features, SVM with 1-D and 2-D features, and CNN in this example. For the training, only one raw training data, which is shown in Fig. 8(a), is considered. The corresponding labelled training data is shown in Fig. 8(b). We labelled the data sets based on a relatively strict criterion. With 'strict' we mean that only the very clear events are selected. Afterward, the training data is split into segments with a length of 0.062 s, which is approximately twice the wavelength. There are 2970 segments in total. Part of the extracted 2-D texture features is demonstrated in Fig. 9. We can see that the 2-D features show large relevance with the events. The correlation matrix of the features is shown in Fig. 10(a). Compared to Fig. 4(a), the correlation between 2-D feature clusters is reduced, because in this more complex scenario, different 2-D features (*Contrast*, *Correlation*, *Energy* and *Homogeneity* with orientations 0°, 45°, 135° and 90° and distances of 1−8) start to make a difference, instead of just creating redundancy. Furthermore, the univariate score (F-value) is shown in Fig. 10(b). It is obvious that the 2-D

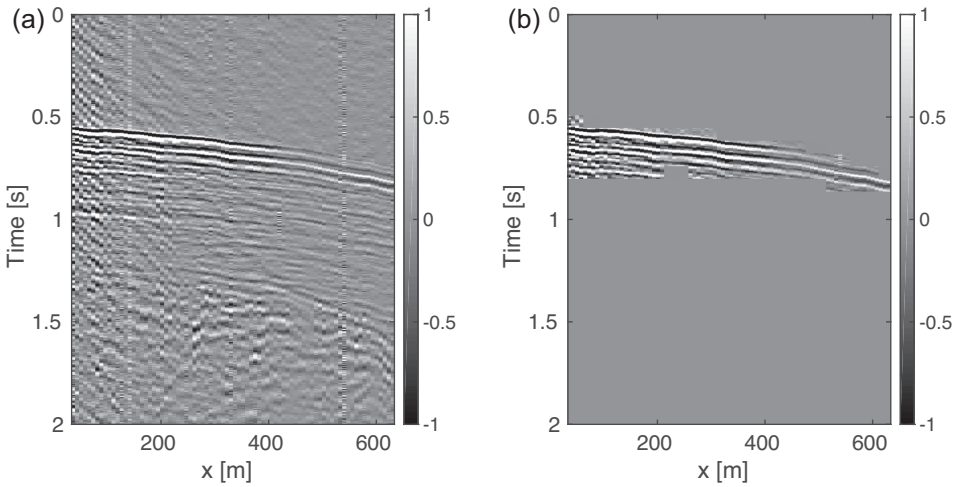

Figure 8. Field data example: (a) raw training data, (b) labelled training data.

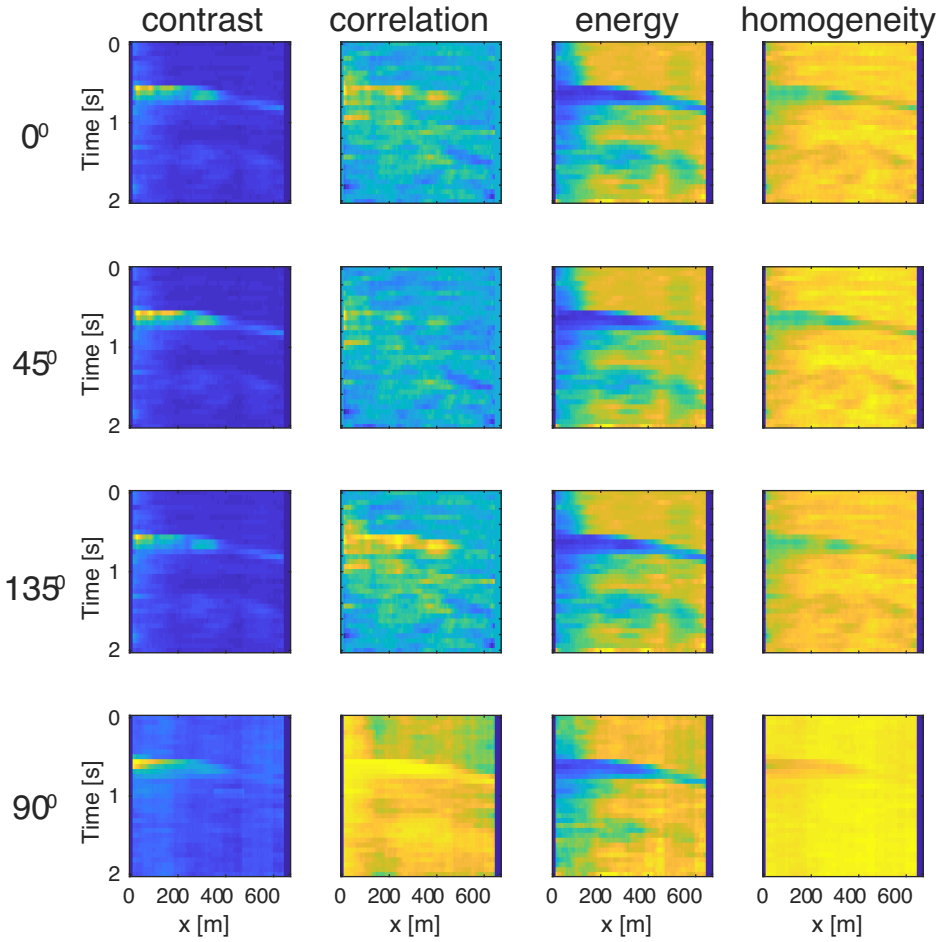

Figure 9. Field data example: 2-D texture features of the training data: Contrast, Correlation, Energy, Homogeneity, for orientations 0°, 45° and 90° and 135°, with a distance of three neighbouring voxels.

texture features play a more important role in the real case, compared to the synthetic example. After feature extraction, 49 features are selected using a combination of univariate feature selection and Random forest. In the SVM classification step, $C = 12.74$ turns out to be the optimal value based on the CV experiment. Finally, the raw test data sets are shown in Figs 11(a), 12(a) and 13(a). The event detection results considering both 1-D and 2-D features are shown in Figs 11(b), 12(b) and 13(b). The event detection results considering only 1-D features are shown in Figs 11(c), 12(c) and 13(c). We can clearly see the improvement of accuracy by using both 1-D and 2-D features, because the continuity, smoothness and regularity of the events are largely emphasized when we label the training data

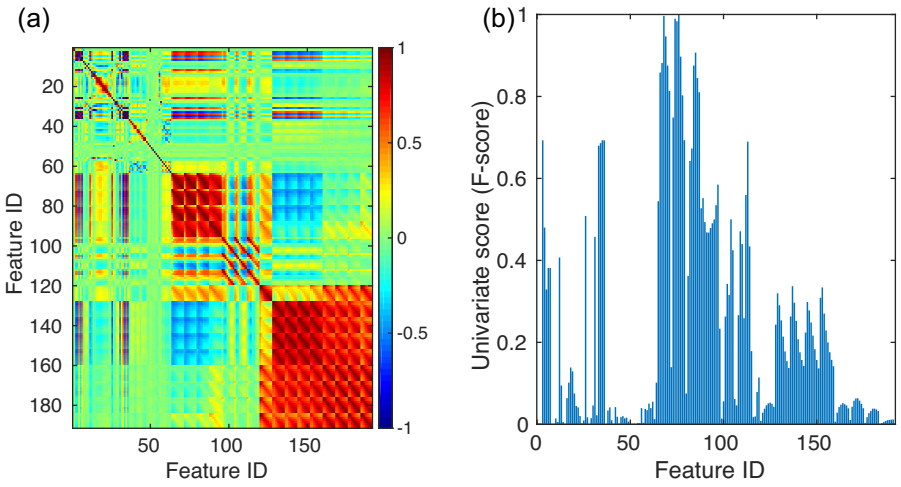

Figure 10. Field data example: (a) correlation matrix of the 191 1-D and 2-D features. Note that 2-D features (ID range 64–191) are correlated due to the nature of the GLCM characteristics, (b) The univariate score (*F*-value) as a function of feature ID.

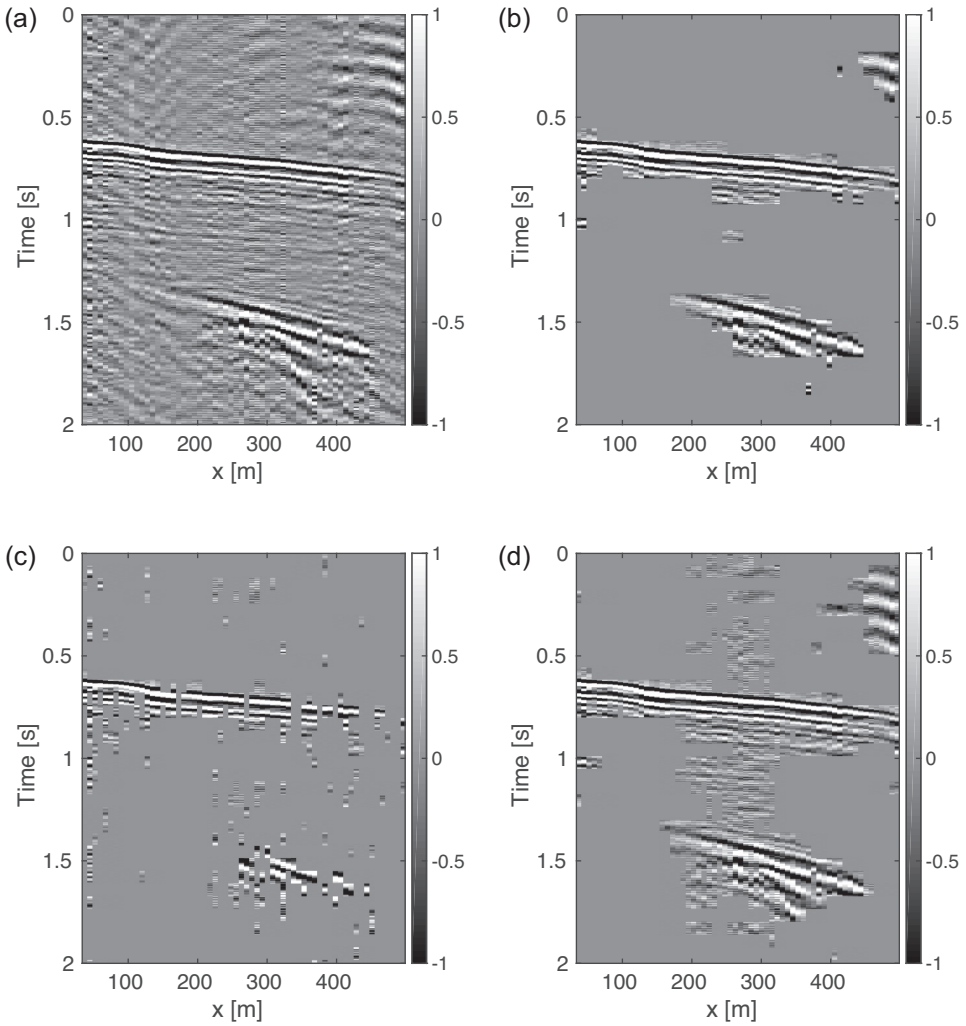

Figure 11. Field data example: (a) raw test data 1, (a) predicted event detection using (b) both 1-D and 2-D features, (c) only 1-D features and (d) a CNN approach.

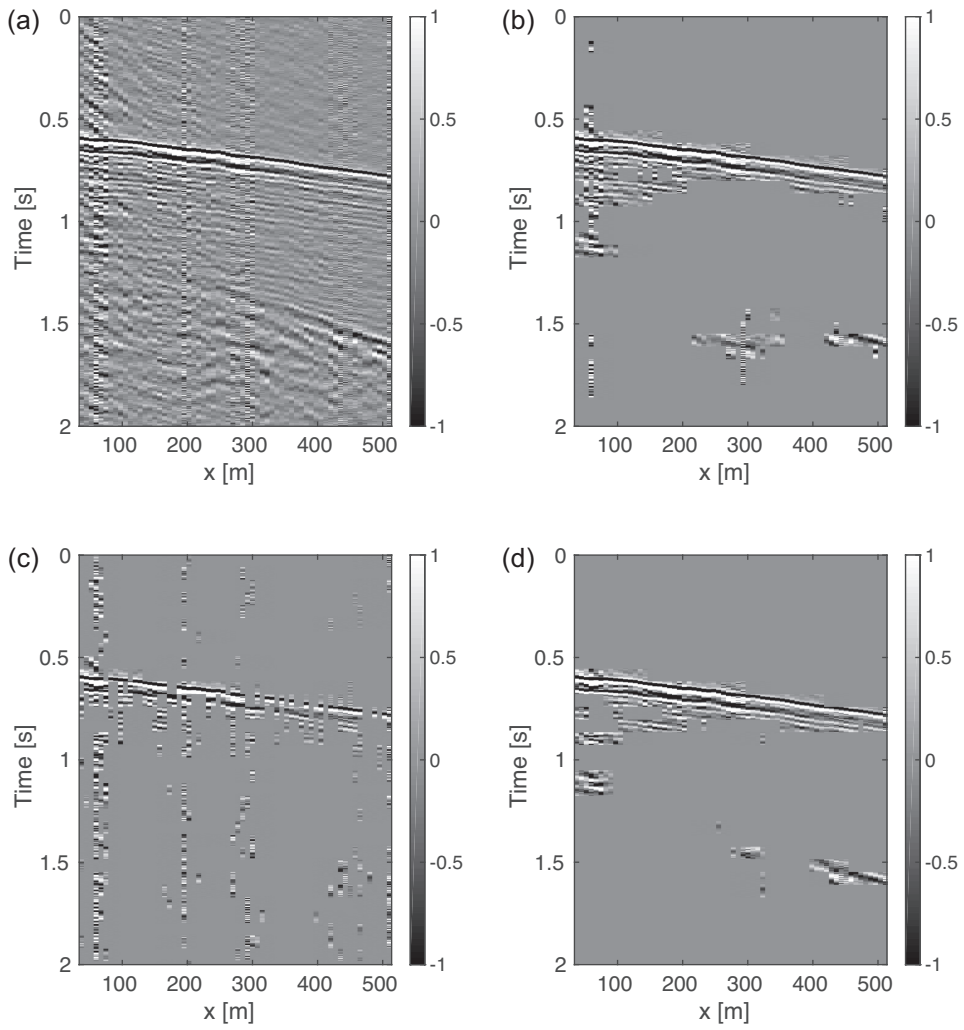

Figure 12. Field data example: (a) raw test data 2, (a) predicted event detection using (b) both 1-D and 2-D features, (c) only 1-D features and (d) a CNN approach.

sets, however, being ignored with only 1-D feature extraction. In addition, the predicted results using the CNN approach are shown in Figs 11(d), 12(d) and 13(d). We can see that using the CNN results in a reasonable prediction, albeit not as good as the proposed workflow, because the CNN is not able to show its privilege over the traditional machine learning algorithms when only one training data set is fed in.

DISCUSSION

Feature importance and sensitivity analysis

With the high classification accuracy achieved by the C-SVM model, we are also interested in the prediction power of the individual features and their corresponding importance. Here, we adopt the best random forest model trained during the recursive feature elimination process in the feature selection section using the synthetic data set. The random forest estimator automatically computes the normalized feature importance metric, and the top 10 most important features are listed in Table 4. We notice that the top nine features are 2-D features, which is in accordance with expectation, since 2-D features carry more information than 1-D features. Furthermore, for the most important feature, being the 135° orientation correlation with distance of two neighbouring voxels (ID 121), we plot its partial dependence in Fig. 14. The partial plot here essentially fixes other features and repeatedly alters the value of feature #121 to make a series of predictions for all of the instances in the test data set. Here the *y*-axis is interpreted as the change in the prediction from what it would be predicted at the baseline value. We see that with a positive feature #121 value it would substantially increase the possibility of detecting a microseismic event and this feature is indeed a robust predictor as the shaded area indicating the confidence level is quite a bit above 0.

Parameter setting and data quality

The implementation of the proposed workflow is quite straightforward. Except for the labelling step, this algorithm is fully automatic and hands-off. Once the feature vectors are fixed, only two parameters are needed: the length of segmentation and regularization parameter C. In this work, $2{*}wavelength$ is used as the segment length in order to obtain high vertical resolution. The regularization parameter C is selected automatically using a cross-validation process to avoid overfitting.

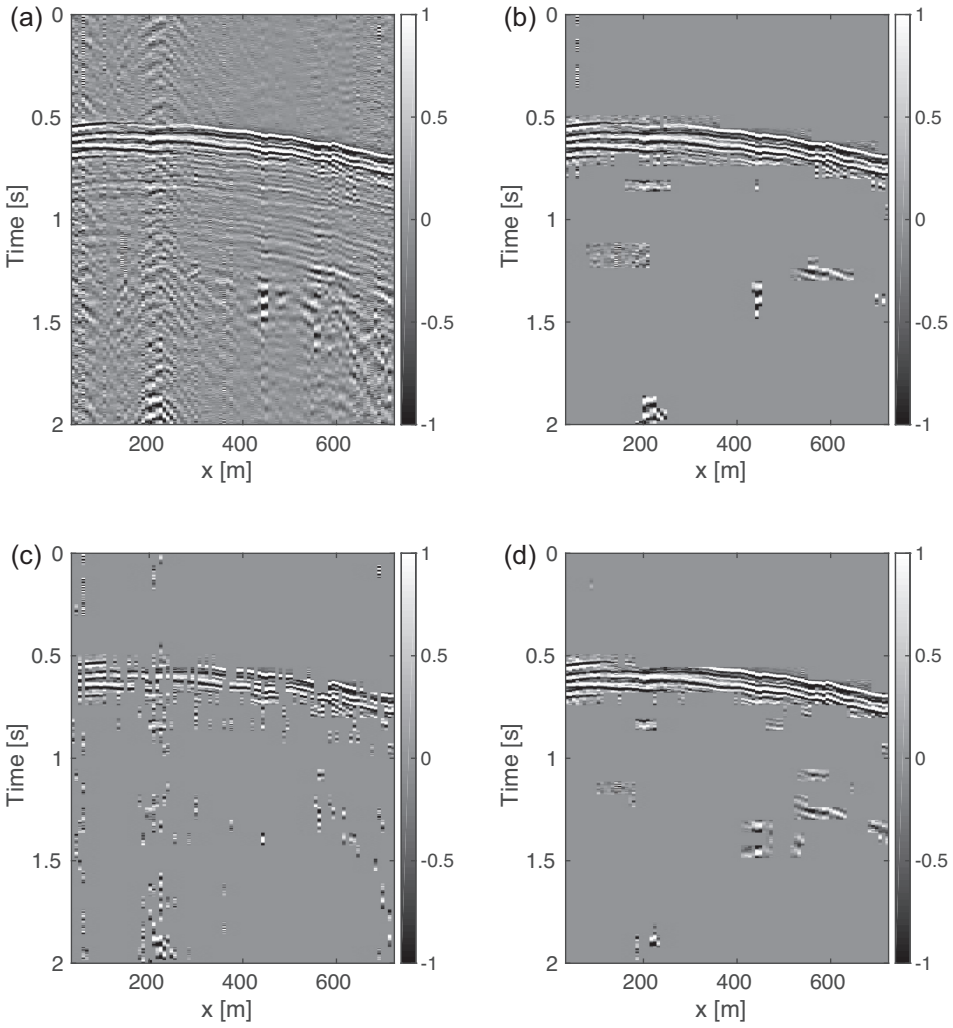

Figure 13. Field data example: (a) raw test data 3, (a) predicted event detection using (b) both 1-D and 2-D features, (c) only 1-D features and (d) a CNN approach.

Table 4. Top 10 most important features, with their normalized importance scores summing up to 0.7440. See Tables 1 and 2 for the explanation of the ID numbers.

ID	121	184	168	122	104	176	127	186	72	36	Sum
Importance	0.3545	0.1255	0.1194	0.0463	0.0345	0.0188	0.0144	0.0124	0.0092	0.0090	0.7440

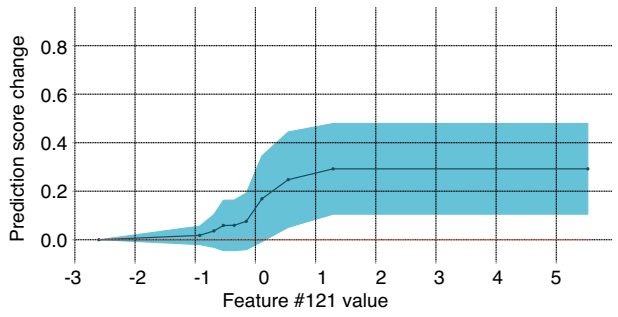

Figure 14. Partial dependence plot for feature #121. A positive feature #121 value would substantially increase the possibility of detecting a microseismic event, and the shaded area denotes the level of confidence.

Regarding the labelling step, the predicted results largely depend on the labelling criterion. The labelled training data based on a relaxed labelling criterion is shown in Fig. 15(a) and the corresponding predicted results are shown in Figs 15(b)–(d). We can see that the results are consistent with the labelling criterion. Therefore, a consistency of the labelling step, which makes sure that the training data sets have a clear classification boundary, is required. In addition, the proposed workflow works well on small data sets, because once a boundary is established during the training, inputting more training data sets are redundant and might even result in over-fitting issue. In this work, only one training data set for both the synthetic and real case is labelled and used to train the model. Using smaller data sets also makes the prediction and labelling process very efficient. The bottom line is that the quality of labelling process is much more important than the quantity.

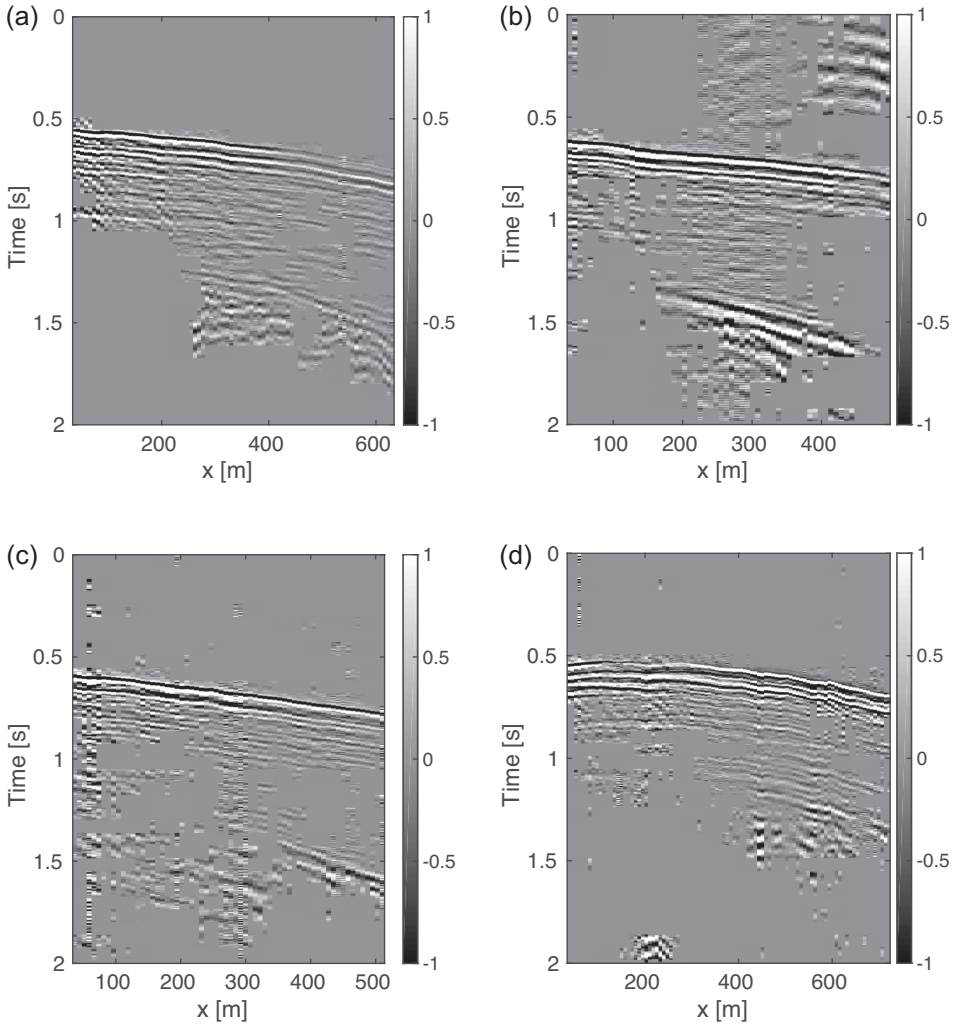

Figure 15. Field data example: (a) labelled training data based on a relaxed criterion; (b–d) predicted event detection of raw test data 1–3 based on this relaxed labelling criterion using both 1-D and 2-D features.

Assuming that a larger number of high-quality labelled training data sets are available, the proposed workflow might not be the optimal choice, since the training time would be too large and an over-fitting issue might pop up. In this case, the use of neural-network-based methods, e.g. convolutional-neural-networks (Krizhevsky *et al.* 2012), is recommended.

Machine learning algorithm

In this work, we choose SVM as the machine learning algorithm in the workflow, due to its high accuracy and nice theoretical guarantees regarding overfitting. Compared to the other popular classification algorithm, like random forests, SVM is more memory-intensive and time-consuming. It is possible to replace SVM with random forests in this workflow for the sake of efficiency. However, in that case the propositions in this paper still stand.

CONCLUSIONS

In order to overcome the noise issue of microseismic data, we proposed a new workflow for high-resolution event detection based on SVM classification with a Gaussian kernel. The proposed workflow is demonstrated in details. For the segmentation step, a length of $2*wavelength$ is used, which then provides the vertical resolution of the event detection. For the feature extraction step, 191 features including both 1-D time/spectral-domain features and 2-D texture features are considered. A combination of univariate feature selection and random-forest-based recursive feature elimination is chosen for feature selection, which finds both the best features and the best number of features needed for the best accuracy. In the training process, the C-SVM model is used and a cross-validation process is conducted for an automatic parameter setting. Finally, a group of synthetic and real microseismic data sets with different levels of complexity showthat the proposed workflow is much more robust than the state-of-the-art STA/LTA method and also performs better than a CNN approach, when the amount of training data sets is limited.

ACKNOWLEDGEMENTS

Shan Qu and Eric Verschuur thank the sponsors of the Delphi consortium for their support. Yangkang Chen is financially supported by the 'Thousand Youth Talents Plan', and the Starting Funds from

Zhejiang University. The authors appreciate Wei Chen for providing the field data sets from a shale play inside the Sichuan Basin. The authors thank Scikit-learn for providing free machine learning library in Python (scikit-learn.org).

REFERENCES

Akram, J., Ovcharenko, O. & Peter, D., 2017. A robust neural network-based approach for microseismic event detection, in *SEG Technical Program Expanded Abstracts 2017,* pp. 2929–2933, Society of Exploration Geophysicists.

Allen, R., 1982. Automatic phase pickers: their present use and future prospects, *Bull. seism. Soc. Am.,* **72**(6B), S225–S242.

Allen, R.V., 1978. Automatic earthquake recognition and timing from single traces, *Bull. seism. Soc. Am.,* **68**(5), 1521–1532.

Arrowsmith, S.J. & Eisner, L., 2006. A technique for identifying microseismic multiplets and application to the valhall field, north seaidentifying microseismic multiplets, *Geophysics,* **71**(2), V31–V40.

Boser, B.E., Guyon, I.M. & Vapnik, V.N., 1992. A training algorithm for optimal margin classifiers, in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory,* pp. 144–152, ACM.

Breiman, L., 2001. Random forests, *Mach. Learn.,* **45**(1), 5–32.

Candes, E., Demanet, L., Donoho, D. & Ying, L., 2006. Fast discrete curvelet transforms, *Multiscale Model. Simulat.,* **5**(3), 861–899.

Chen, Y., 2018c. Fast waveform detection for microseismic imaging using unsupervised machine learning, *Geophys. J. Int.,* **215,** 1185–1199.

Chen, Y., Bai, M., Zu, S., Guan, Z. & Zhang, M., 2019c. Automatic waveform classification and arrival picking based on convolutional neural network, *Earth Space Sci,* doi:10.1029/2018EA000466.

Chen, Y.-W. & Lin, C.-J., 2006. Combining svms with various feature selection strategies, in *Feature Extraction,* pp. 315–324, Springer.

Cortes, C. & Vapnik, V., 1995. Support-vector networks, *Mach. Learn.,* **20**(3), 273–297.

Dokht, R.M., Kao, H., Visser, R. & Smith, B., 2019. Seismic event and phase detection using time–frequency representation and convolutional neural networks, *Seism. Res. Lett.,* **90**(2A), 481–490.

Elad, M. & Aharon, M., 2006. Image denoising via sparse and redundant representations over learned dictionaries, *IEEE Trans. Image Process.,* **15**(12), 3736–3745.

Forghani-Arani, F., Willis, M., Haines, S.S., Batzle, M., Behura, J. & Davidson, M., 2013. An effective noise-suppression technique for surface microseismic data, *Geophysics,* **78**(6), KS85–KS95.

Gelchinsky, B. & Shtivelman, V., 1983. Automatic picking of first arrivals and parameterization of traveltime curves, *Geophys. Prospect.,* **31**(6), 915–928.

Gibbons, S.J. & Ringdal, F., 2006. The detection of low magnitude seismic events using array-based waveform correlation, *Geophys. J. Int.,* **165**(1), 149–166.

Guan, Z. & Niu, F., 2017. An investigation on slowness-weighted ccp stacking and its application to receiver function imaging, *Geophys. Res. Lett.,* **44**(12), 6030–6038.

Haralick, R.M. *et al.*, 1973. Textural features for image classification, *IEEE Trans. Syst., Man, and Cybernet.,* **6,** 610–621.

Hatherly, P., 1982. A computer method for determining seismic first arrival times, *Geophysics,* **47**(10), 1431–1436.

Hsu, C.-W. *et al.*, 2003. A practical guide to support vector classification. Paper available at http://www.csie.ntu.edu.tw/cjlin/papers/guide/guide.pdf.

Huang, N.E. *et al.*, 1998. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis, in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences,* **454,** 903–995, The Royal Society.

Huang, W., Wang, R., Zhou, Y., Chen, Y. & Yang, R., 2016. Improved principal component analysis for 3D seismic data simultaneous reconstruction and denoising, in *SEG Technical Program Expanded Abstracts 2016,* pp. 4102–4106, Society of Exploration Geophysicists.

Huang, W., Wang, R., Chen, X., Zhou, Y., Chen, Y. & You, J., 2017a. Low-frequency noise attenuation of seismic data using mathematical morphological filtering, in *87th Annual International Meeting, SEG, Expanded Abstracts,* pp. 5011–5016.

Knapmeyer-Endrun, B. & Hammer, C., 2015. Identification of new events in apollo 16 lunar seismic data by hidden markov model-based event detection and classification, *J. geophys. Res.: Planets,* **120**(10), 1620–1645.

Krizhevsky, A., Sutskever, I. & Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems,* pp. 1097–1105.

LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E. & Jackel, L.D., 1990. Handwritten digit recognition with a back-propagation network, in *Advances in Neural Information Processing Systems,* pp. 396–404.

LeCun, Y., Bengio, Y. & Hinton, G., 2015. Deep learning, *nature,* **521**(7553), 436.

Li, H., Wang, R., Cao, S., Chen, Y. & Huang, W., 2016. A method for low-frequency noise suppression based on mathematical morphology in microseismic monitoring, *Geophysics,* **81,** V159–V167.

Li, S., Liu, B., Ren, Y., Chen, Y., Yang, S., Wang, Y. & Jiang, P., 2019. Deep learning inversion of seismic data, preprint (arXiv:1901.07733).

Liu, W., Cao, S., Chen, Y. & Zhang, D., 2016b. Application of variational mode decomposition in random noise attenuation and time-frequency analysis of seismic data, in *78th EAGE Conference and Exhibition 2016,* doi: 10.3997/2214-4609.201601249.

Maxwell, S.C., Rutledge, J., Jones, R. & Fehler, M., 2010. Petroleum reservoir characterization using downhole microseismic monitoring, *Geophysics,* **75**(5), 75A129–75A137.

McCormack, M.D., Zaucha, D.E. & Dushek, D.W., 1993. First-break refraction event picking and seismic data trace editing using neural networks, *Geophysics,* **58**(1), 67–78.

Michelet, S. & Toksöz, M.N., 2007. Fracture mapping in the soultz-sous-forêts geothermal field using microearthquake locations, *J. geophys. Res.: Solid Earth,* **112**(B7).

Mousavi, S.M. & Langston, C.A., 2016a. Adaptive noise estimation and suppression for improving microseismic event detection, *J. appl. Geophys.,* **132,** 116–124.

Mousavi, S.M. & Langston, C.A., 2016b. Hybrid seismic denoising using higher-order statistics and improved wavelet block thresholding, *Bull. seism. Soc. Am.,* **106**(4), 1380–1393.

Mousavi, S.M., Horton, S.P., Langston, C.A. & Samei, B., 2016. Seismic features and automatic discrimination of deep and shallow induced-microearthquakes using neural network and logistic regression, *Geophys. J. Int.,* **207**(1), 29–46.

Mousavi, S.M., Beroza, G. & Zhu, W., 2018a. Earthquake signal detection using a deep convolutional-recurrent network, in *AGU Fall Meeting Abstracts*.

Mousavi, S.M., Zhu, W., Sheng, Y. & Beroza, G.C., 2018b. Cred: A deep residual network of convolutional and recurrent units for earthquake signal detection, preprint (arXiv:1810.01965).

Perol, T., Gharbi, M. & Denolle, M., 2018. Convolutional neural network for earthquake detection and location, *Sci. Adv.,* **4**(2), e1700578.

Provost, F., Hibert, C. & Malet, J.-P., 2017. Automatic classification of endogenous landslide seismicity using the random forest supervised classifier, *Geophys. Res. Lett.,* **44**(1), 113–120.

Qu, S., Verschuur, E. & Chen, Y., 2018. Automatic microseismic-event detection via supervised machine learning, in *2018 SEG International Exposition and Annual Meeting,* pp. 2287–2291, Society of Exploration Geophysicists.

Rouet-Leduc, B., Hulbert, C., Lubbers, N., Barros, K., Humphreys, C.J. & Johnson, P.A., 2017. Machine learning predicts laboratory earthquakes, *Geophys. Res. Lett.,* **44**(18), 9276–9282.

Scheffe, H., 1967. *The Analysis of Variance,* Wiley.

Senkaya, M. & Karsli, H., 2014. A semi-automatic approach to identify first arrival time: the cross-correlation technique (CCT), *Earth Sci. Res. J.,* **18**(2), 107–113.

Shapiro, S., Dinske, C. & Rothert, E., 2006. Hydraulic-fracturing controlled dynamics of microseismic clouds, *Geophys. Res. Lett.,* **33**(14).

Song, F., Kuleli, H.S., Toksöz, M.N., Ay, E. & Zhang, H., 2010. An improved method for hydrofracture-induced microseismic event detection and phase picking, *Geophysics,* **75**(6), A47–A52.

Trnkoczy, A., 1999. Topic understanding and parameter setting of sta/lta trigger algorithm, *New Manual Seismol. Observat. Pract.,* **2**.

Vaezi, Y. & Van der Baan, M., 2015. Comparison of the sta/lta and power spectral density methods for microseismic event detection, *Geophys. J. Int.,* **203**(3), 1896–1908.

Vautard, R., Yiou, P. & Ghil, M., 1992. Singular-spectrum analysis: a toolkit for short, noisy chaotic signals, *Phys. D: Nonlin. Phenom.,* **58**(1-4), 95–126.

Vera Rodriguez, I., Bonar, D. & Sacchi, M., 2012. Microseismic data denoising using a 3c group sparsity constrained time-frequency transform, *Geophysics,* **77**(2), V21–V29.

Warpinski, N.R., 2000. Analytic crack solutions for tilt fields around hydraulic fractures, *J. geophys. Res.: Solid Earth,* **105**(B10), 23463–23478.

Withers, M., Aster, R., Young, C., Beiriger, J., Harris, M., Moore, S. & Trujillo, J., 1998. A comparison of select trigger algorithms for automated global seismic phase and event detection, *Bull. seism. Soc. Am.,* **88**(1), 95–106.

Xia, Y., Ni, S. & Zeng, X., 2013. Twin enigmatic microseismic sources in the gulf of guinea observed on intercontinental seismic stations, *Geophys. J. Int.,* **194**(1), 362–366.

Zhang, D., Verschuur, D. & Chen, Y., 2019. Surface multiple leakage extraction using local primary-and-multiple orthogonalization, in *81st Annual International Conference and Exhibition, EAGE, Extended Abstracts,* doi:10.3997/2214–4609.201901199.

Zhao, Z. & Gross, L., 2017. Using supervised machine learning to distinguish microseismic from noise events, in *SEG Technical Program Expanded Abstracts 2017,* pp. 2918–2923, Society of Exploration Geophysicists.

Zheng, J., Lu, J., Peng, S. & Jiang, T., 2017. An automatic microseismic or acoustic emission arrival identification scheme with deep recurrent neural networks, *Geophys. J. Int.,* **212**(2), 1389–1397.

Zhu, W. & Beroza, G.C., 2018. Phasenet: a deep-neural-network-based seismic arrival-time picking method, *Geophys. J. Int.,* **216**(1), 261–273.