Automatic microseismic-event detection via supervised machine learning

Qu, Shan; Verschuur, D.J.; Chen, Yangkang

DOI
10.1190/segam2018-2998279.1

Publication date
2018

Document Version
Final published version

Published in
SEG Technical Program Expanded Abstracts 2018

Automatic microseismic-event detection via supervised machine learning

Conference Paper · August 2018

DOI: 10.1190/segam2018-2998279.1

3 authors, including:

Shan Qu
Delft University of Technology
45 PUBLICATIONS **482** CITATIONS
SEE PROFILE

Yangkang Chen
Zhejiang University
289 PUBLICATIONS **5,190** CITATIONS
SEE PROFILE

Automatic microseismic event detection via supervised machine learning

Shan Qu, Eric Verschuur, Delft University of Technology, Delphi Consortium, and Yangkang Chen, National Center for Computational Sciences, Oak Ridge National Laboratory*

SUMMARY

Microseismic methods are crucial for real-time monitoring of the hydraulic fracturing dynamic status. However, unlike the active-source seismic events, the microseismic events usually have very low signal-to-noise ratio (SNR), which makes its processing challenging. To overcome the noise issue of the weak microseismic events, we propose a novel method for microseismic event detection based on the support vector machine (SVM) classification with a Gaussian kernel. For the preprocessing, fix-sized segmentation with a length of $2*wavelength$ is used to divide the data into segments. 123 features in total, which are used as input data to train the SVM model, have been extracted. These features include 63 1D time/spectral-domain features, and 60 2D texture features. Afterwards, we use a combination of univariate feature selection and random-forest-based recursive feature elimination for feature selection. This feature selection strategy not only finds the best features but also decides the number of features that are needed for the best accuracy. Regarding the essential training process, a C-SVM model, where coefficient C is used to control the tolerance of error item, is considered and a cross-validation (CV) process is implemented for automatic parameter setting. In the end, a group of synthetic and real microseismic data with different levels of complexity show the effectiveness of the proposed method.

INTRODUCTION

It has been well known that microseismic monitoring plays a significant role in characterizing physical processes related to fluid injections and extractions in hydrocarbon and geothermal reservoirs (Shapiro et al., 2006; Huang et al., 2017; Chen, 2018a,b). However, the energy stimulated from the hydraulic fracturing is usually extremely weak. As a result, strong noise in microseismic data may impede the effective usage of the microseismic data for the characterization and monitoring. Prior to the localization and mechanism analysis of the source, the identification and detection of microseismic events become an important challenge.

Traditional event detection is based on phase picking, which is commonly done by energy analysis. The energy analysis is a widely used method due to few assumptions about the data. For example, one criterion is the short-term average over long-term average (STA/LTA) ratio, which can detect the appearance of seismic events when the ratio exceeds the given threshold (Allen, 1982, 1978). However, this method is not sensitive to weak events. Another event detection method is based on template matching, which takes advantage of predetermined events, known as the master event, and cross-correlates them with continuous recordings to detect events with high similarities (Gibbons and Ringdal, 2006; Song et al., 2010). However, template matching requires a master event as the input, which is not always available. In addition, it tends to detect events that are similar to the master event, therefore, lead to biased results. In recent years, some researchers have already done investigations on supervised machine learning based event detection. Maity et al. (2014), Akram et al. (2017), and Chen et al. (2017) proposed to use the neural network model to automatically select microseismic signal arrivals and Zhao et al. (2017) used support vector machine to distinguish microseismic from noise events. However, their methods only consider 1D features for the training and require longer length of segmentation ($\sim 15*wavelength$) for providing stable results. The results, therefore, have coarse resolution.

In this work, we propose a novel method for microseismic event detection based on support vector classification with the Gaussian kernel. After a brief introduction of the SVM, 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; ② 123 features have been extracted in total, including 63 1D time/spectral-domain features, and 60 2D texture features; 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 is considered in the essential training process. In addition, cross-validation (CV) process is implemented for automatic parameter setting; ⑤ the trained model is then applied to detect events of the test data. In the end, a group of synthetic and real microseismic data with different levels of complexity show the effectiveness of the proposed method.

SUPPORT VECTOR CLASSIFICATION

Recently, support vector machine (SVM) has been an effective classification method by constructing hyperplanes with the maximal margin in a multi-dimensional space which separates different cases of different class labels (Cortes and Vapnik, 1995; Boser et al., 1992). 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 coefficient C is used to control the tolerance of the systematic outliers that allows less outliers to exist in the opponent class. This model solves a quadratic optimization problem:

$$\begin{aligned} &\min_{\omega,b,\xi} = \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{N}\xi_i \\ &\text{subject to the constraints:,} \\ &y_i\left(\omega^T\phi(x_i)+b\right) \geq 1-\xi_i \text{ and } \xi_i \geq 0, i = 1,...,N, \end{aligned} \tag{1}$$

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 over-fitting. Note that ξ_i is the smallest non-negative number satisfying $y_i\left(\omega^T\phi(x_i)+b\right) \geq 1-\xi_i$.

10.1190/segam2018-2998279.1

The kernel ϕ is used to transform the input data into the feature space. The kernel function $k\left(x_i, x_j\right) = \phi(x_i) \cdot \phi(x_j)$ represents a dot product of input data points to map into the higher dimensional feature space by transformation ϕ.

MICROSEISMIC EVENT DETECTION AS A SUPPORT VECTOR CLASSIFICATION PROBLEM

The workflow of SVM classification for microseismic event detection can be summarized as the following steps: ① Segmentation and labelling; ② Feature extraction and normalization; ③ feature selection; ④ SVM classification; ⑤ Test on new data. To clearly demonstrate the whole procedure of SVM classification for event detection, a group of synthetic data is used. The raw training and test data are shown in Fig. 1a and 1c, respectively. The receiver spacing is $7.5m$ and the time duration is $3.1s$. $SNR = -10$ for all the datasets, which means the noise level is much stronger than the signal level. In addition, five different traces of clean and noisy data are demonstrated in Fig. 1g. We can see that the signal is masked by the strong background noise and hardly to detect.

1. Segmentation and labelling

Segmentation is a very important preprocessing stage, where the microseismic data are split into segments of signal. Fix-sized segmentation is used in this work and we set the length of each segment as $2 * wavelength$. After the segmentation, the training data segments are labelled into two classes: events and noise. For the synthetic example, a total of 12960 segments are extracted, including 4853 segments of microseismic event and 8107 segments of noise. The labelled training synthetic data is shown in Fig. 1b. As is well-known, supervised classification is largely dependent on the labelled training datasets, which are usually done based on different criterion of different users. Please note that, we will show the predicted results based on different criterion of the labelling step in the **Real data example** section later on.

2. Feature extraction and normalization

The purpose of feature extraction is to convert all the segments into relevant features, which are served as training vectors for SVM classification. 123 features have been extracted, including 63 1D time/frequency-domain features, and 60 2D texture features. The 1D features, consist of both time-domain features and spectral features, are listed in Table 1 with description. However, by only considering the 1D features of the seismic data, the 2D features (for example, continuity, smoothness, and irregularity of the events) are ignored, which is obviously a waste of information. Therefore, we propose to consider 60 extra 2D texture features. Regarding the extraction of texture features, the 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, 0^o, 45^o, and 135^o orientation and distance of $1-5$ of neighbouring voxels are considered. Details of the texture features are shown in Table 2.

Table 1: The list of the implemented 1D time/spectral-domain features

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% of observations fall
6	75th percentile	The value below which 75% 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 are heavy-tailed or light-tailed relative to a 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 Frequency Cepstral Coefficients, form a cepstral representation where the frequency bands are not linear but distributed according to the mel-scale
26	Dominant frequency magnitude	The energy of a spectrum is centered upon
27	Spectral Centroid	Indice of the dominant frequency
28	Spectral Spread	The second central moment of the spectrum
29	Spectral Entropy	Entropy of the normalized spectral energies
30	Spectral Roll-off	The frequency below which 85% 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	Coefficients 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 coefficients
50-56	Spectral contrast	It considers the spectral peak, the spectral valley, and their difference in each frequency subband
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 implemented 2D texture features

ID	Feature Name	Description
64-78	Contrast	Measures the local variations in the GLCM, for 0^o, 45^o, 90^o orientation, and 135^o with distance of $1-5$ of neighbouring voxels
79-93	Correlation	Measures the joint probability occurrence of the specified pixel pairs in the GLCM, for 0^o, 45^o, and 135^o orientation, with distance of $1-5$ of neighbouring voxels
94-108	Energy	Provides the sum of squared elements in the GLCM. Also known as uniformity or the angular second moment, for 0^o, 45^o, and 135^o orientation, with distance of $1-5$ of neighbouring voxels
109-123	Homogeneity	Measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal., for 0^o, 45^o, and 135^o orientation, with distance of $1-5$ of neighbouring voxels

After feature extraction, normalization of the features 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.

3. Feature selection

In machine learning, feature selection is the process of selecting a subset of most relevant features for use in model training. It can save training time and enhance the generalization by reducing over-fitting. Univariate feature selection is a simple technique where a statistical test is applied to each feature individually based on a certain scoring criterion, e.g. F-score. However, it does not reveal mutual information among fea-

10.1190/segam2018-2998279.1

tures (Chen and Lin, 2006). Random forest (RF) is a classification method and it also provides feature importance (Breiman, 2001). However, RF cannot handle too many features (Chen and 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. The univariate feature score as a function of feature ID for synthetic example is shown in Fig. 1f, from where we can see that the 2D texture features show large relevance with respect to different classes. In the end, 11 features are selected in the synthetic example.

4. SVM classification

In this step, the radial basis function (RBF) is used as the kernel function for SVM:

$$k\left(x_i, x_j\right) = \exp\left(-\gamma ||x_i - x_j||^2\right), \tag{2}$$

where, γ is an adjustable parameter of certain kernel functions. In this case, we set it as $1/N_{features}$.

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 5-fold 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 0.91. In addition, the size of the noise class is normally different from the size of the event class. This data imbalance could leads to bias. In order to compensate this, we adjust weights inversely proportional to class frequencies in the input data.

5. Test on new data

After obtaining the trained SVM model, we apply it to the test data in Fig. 1c, which is not part of the training data. The predicted event detection result considering both 1D and 2D features is shown in Fig. 1d and the result considering only 1D features is shown in Fig. 1e. We can see that both of them result in reasonable prediction with 96% and 91% accuracy, respectively. However, by getting use of 2D texture features, there is a huge improvement in the prediction accuracy in Fig. 1d compared to Fig. 1e. Furthermore, five different traces (at 150, 525, 900, 1275, 1650m) of clean, noisy data, and predicted detection are demonstrated in Fig. 1g. It can be seen that the events smeared in the strong ambient noise are also able to be detected by using our proposed method.

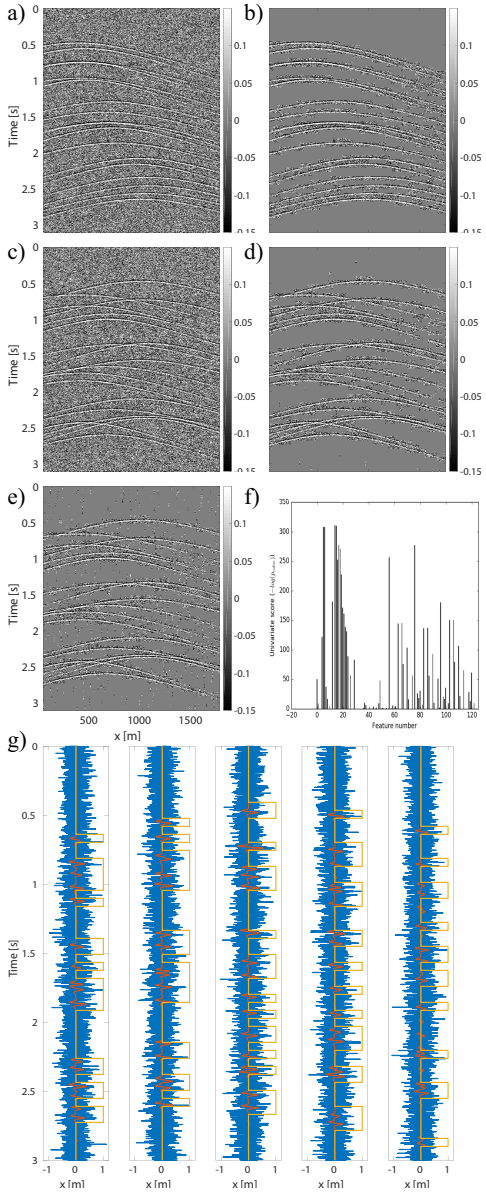

Figure 1: Synthetic example: a) raw training data, b) labelled training data, c) raw test data, d) predicted event detection of the test data using both 1D and 2D features, e) predicted event detection of the test data using only 1D features, f) the univariate score of features, g) five traces (at 150, 525, 900, 1275, 1650m) of clean (red) and noisy (blue) test data and the predicted event detection (yellow; 0-noise, 1-event).

REAL DATA EXAMPLE

Fig. 2 shows the performance of the proposed method for the real data example. This is a group of surface-recorded microseismic data. The receiver spacing is 7.5m and the time duration is 2s. The raw training datasets are shown in Fig. 2a and 2d and the raw test data is shown in Fig. 2g. First, all the microseismic datasets are split into segments with the length of 0.062s, which is around twice of the wavelength. After that, in order to demonstrate how the labelling of the training datasets based on different criterion affects the final prediction, we implement two types of labelling for the training datasets: based on a relaxed criterion (scenario 1) and a strict criterion (scenario 2). The corresponding labelled training datasets are shown in Fig. 2b,2e and 2c,2f, respectively. The univariate score for scenario 1 and 2 are shown in Fig. 2j and 2k, respectively. We can see that the 2D texture features play a more important role in the real case, compared to the synthetic example. After feature extraction, 36 features in the scenario 1 and 32 features in the scenario 2 are selected. In SVM clas-

10.1190/segam2018-2998279.1

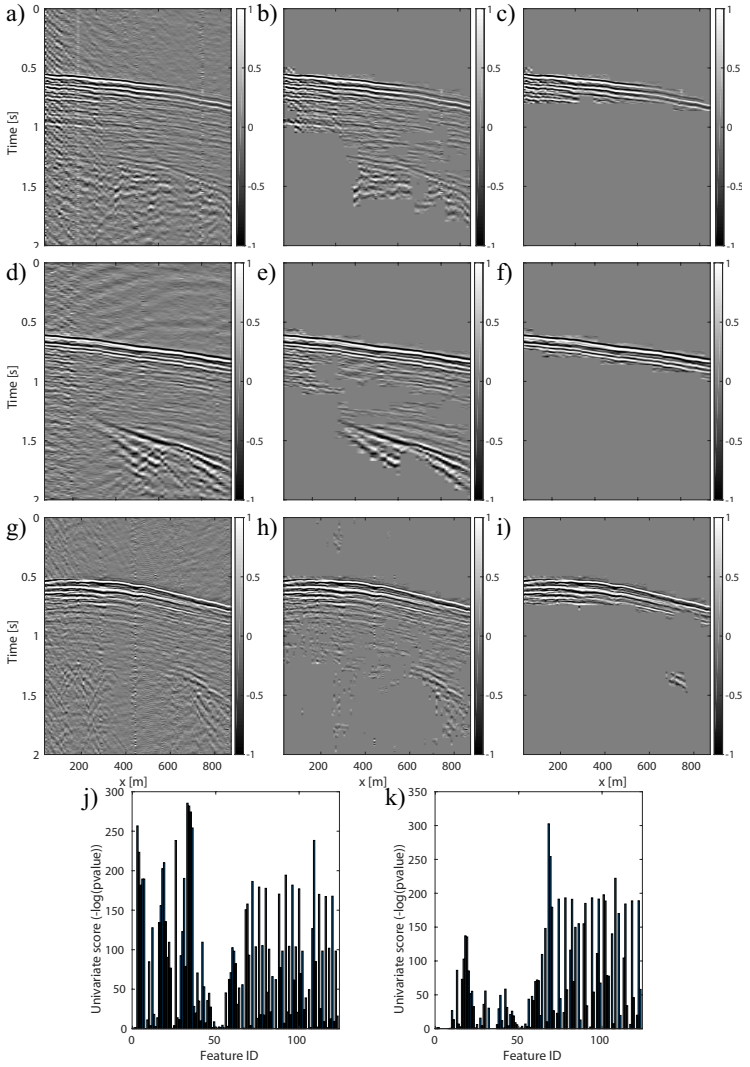

Figure 2: Real data example: a) raw training data 1, b) labelled training data 1 based on a relaxed criterion, c) labelled training data 1 based on a strict criterion, d) raw training data 2, e) labelled training data 2 based on a relaxed criterion, f) labelled training data 2 based on a strict criterion, g) raw test data, h) predicted event detection of the test data based on a relaxed criterion labelling, i) predicted event detection of the test data based on a strict criterion labelling, j) the univariate score of features based on a relaxed criterion labelling, k) the univariate score of features based on a strict criterion labelling.

sification step, $C = 12.74$ in the scenario 1 and $C = 6.16$ in the scenario 2 turn out to be the optimal values based on the CV experiment. Finally, the predicted event detection results of the test data are shown in Fig. 2h and 2j. They are quite reasonable and consistent with the labelling criterion.

CONCLUSIONS

In order to overcome the noise issue of microseismic data, we proposed a novel method for microseismic event detection based on support vector machine classification with a Gaussian kernel. The proposed method 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, 123 features including both 1D time/spectral-domain features and 2D texture features are calculated. 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 automatic parameter setting. Finally, a group of synthetic and real microseismic data with different levels of complexity show the effectiveness of the proposed method.

ACKNOWLEDGEMENTS

Shan Qu and Eric Verschuur thank the sponsors of the Delphi consortium for their support. Yangkang Chen is financially supported by the Distinguished Postdoctoral Fellowship at Oak Ridge National Laboratory. The authors thank Scikit-learn for providing free machine learning library in Python (scikit-learn.org).

REFERENCES

Akram, J., O. Ovcharenko, and D. Peter, 2017, A robust neural network-based approach for microseismic event detection: 87th Annual International Meeting, SEG, Expanded Abstracts, 2929–2933, https://doi.org/10.1190/segam2017-17761195.1.

Allen, R. V., 1978, Automatic earthquake recognition and timing from single traces: Bulletin of the Seismological Society of America, **68**, 1521–1532.

Allen, R. V., 1982, Automatic phase pickers: Their present use and future prospects: Bulletin of the Seismological Society of America, **72**, S225–S242.

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

Breiman, L., 2001, Random forests: Machine Learning, **45**, 5–32, https://doi.org/10.1023/A:1010933404324.

Chen, Y., 2018a, Automatic microseismic event picking via unsupervised machine learning: Geophysical Journal International, **212**, 88–102, https://doi.org/10.1093/gji/ggx420.

Chen, Y., 2018b, Non-stationary least-squares complex decomposition for microseismic noise attenuation: Geophysical Journal International, **213**, 1572–1585, https://doi.org/10.1093/gji/ggy079.

Chen, Y., J. C. Hill, W. Lei, M. Lefebvre, E. Bozdag, D. Komatitsch, and J. Tromp, 2017, Automated time-window selection based on machine learning for full-waveform inversion: 87th Annual International Meeting, SEG, Expanded Abstracts, 1604–1609, https://doi.org/10.1190/segam2017-17734162.1.

Chen, Y.-W., and C.-J. Lin, 2006, Combining SVMS with various feature selection strategies, in I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, eds., Feature extraction: Springer, 315–324.

Cortes, C., and V. Vapnik, 1995, Support-vector networks: Machine Learning, **20**, 273–297, https://doi.org/10.1007/BF00994018.

Gibbons, S. J., and F. Ringdal, 2006, The detection of low magnitude seismic events using array-based waveform correlation: Geophysical Journal International, **165**, 149–166, https://doi.org/10.1111/j.1365-246X.2006.02865.x.

Haralick, R. M., K. Shanmugam, and I. Dinstein, 1973, Textural features for image classification: IEEE Transactions on Systems, Man, and Cybernetics, **SMC-3**, 610–621, https://doi.org/10.1109/TSMC.1973.4309314.

Hsu, C.-W., C.-C. Chang, and C.-J. Lin, 2003, A practical guide to support vector classification.

Huang, W., R. Wang, H. Li, and Y. Chen, 2017, Unveiling the signals from extremely noisy microseismic data for high-resolution hydraulic fracturing monitoring: Scientific Reports, **7**, 11996, https://doi.org/10.1038/s41598-017-09711-2.

Maity, D., F. Aminzadeh, and M. Karrenbach, 2014, Novel hybrid artificial neural network based autopicking workflow for passive seismic data: Geophysical Prospecting, **62**, 834–847, https://doi.org/10.1111/1365-2478.12125.

Shapiro, S. A., C. Dinske, and E. Rothert, 2006, Hydraulic-fracturing controlled dynamics of microseismic clouds: Geophysical Research Letters, **33**, L14312, https://doi.org/10.1029/2006GL026365.

Song, F., H. S. Kuleli, M. N. Toksöz, E. Ay, and H. Zhang, 2010, An improved method for hydrofracture-induced microseismic event detection and phase picking: Geophysics, **75**, no. 6, A47–A52, https://doi.org/10.1190/1.3484716.

Zhao, Z., and L. Gross, 2017, Using supervised machine learning to distinguish microseismic from noise events: 87th Annual International Meeting, SEG, Expanded Abstracts, 2918–2923, https://doi.org/10.1190/segam2017-17727697.1.