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
Accepted author manuscript

Published in
Geophysical Journal International

Automatic high-resolution microseismic event detection via supervised machine learning

Shan Qu*, Zhe Guan†, Eric Verschuur*, Yangkang Chen†

* *Delft University of Technology, Delphi consortium*

† *Rice University, Applied Physics Program*

‡ *Zhejiang university, School of Earth Sciences*

(November 3, 2020)

Running head: **Microseismic event detection**

ABSTRACT

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 1D time/spectral-domain features, and 128 2D 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 over-fitting. 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, support vector machine (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 datasets is limited. A demo for the synthetic example is available: `https://github.com/shanqu91/ML_event_detection_microseismic.`

Key words: machine learning, high-resolution, 2D texture features, microseismic, event detection, SVM

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; 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 and Langston, 2016a,b; Liu et al., 2016).

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 and 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., 2016; 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 and 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 and 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 and Ringdal (2006). It 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; Senkaya and 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 and Toksöz (2007); Arrowsmith and 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 and 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 and Hammer, 2015; Akram et al., 2017; Provost et al., 2017; Rouet-Leduc et al., 2017; Zheng et al., 2017; Zhao and Gross, 2017; Chen et al., 2017; Mousavi et al., 2018a,b; Perol et al., 2018; Zhu and Beroza, 2018; Dokht et al., 2019) Zhao and Gross (2017) trained a support vector machine (SVM) model with 1D 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 (2018) 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 and Beroza (2018), Perol et al. (2018), 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 datasets compared to the traditional machine-learning-based methods like SVM.

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 1D time/spectral-domain features, and 128 2D 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; 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 datasets 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 labeling; ② Feature extraction and normalization; ③ feature selection; ④ Support Vector Classification; ⑤ Test on new data. To clearly demonstrate the whole workflow for event detection, a group of synthetic microseismic datasets is used. The synthetic microseismic datasets are simulated from the three-layer velocity model shown in Figures 1a and 1b. The modeled training data with $SNR = -13$ is shown in Figure 2a. The modeled test datasets are displayed in Figures 5a and 6a, including $SNR = -10$ and $SNR = -13$ level of noise energy, respectively. The definition of noise

energy is as follows:

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

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

1. Segmentation and labeling

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 $0.058s$ 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 labeled into two classes: events and noise. For the synthetic example, a total of 12960 segments are extracted, including 4853 segments containing a microseismic event and 8107 segments of noise. The labeled training synthetic data is shown in Figure 2b. As is well-known, supervised classification is largely dependent on the labeled training datasets, which are usually done based on different criteria from different users. Please note that we will discuss how using different labeling criteria affects the final prediction in the **Discussion** 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 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 and Gross, 2017). They extracted 1D 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 1D features is not enough. In order to maximize the information per segment, we propose to extract both 1D time/frequency-domain features and 2D texture features for each segment. 191 features have been extracted, including 63 1D features, and 128 2D features. The 1D features consist of both time-domain features and spectral features and are listed in Table 1 with description. 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, 128 extra 2D 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^o, 45^o, 135^o, and 90^o and distances of $1 - 8$ neighboring 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 subsection **3. Feature selection**. The calculated 2D texture parameters within the moving window are severed as the features of the segment in the center of the window. Details of the texture features are described in Table 2. Part of the 2D texture features extracted from the training data are demonstrated in Figure 3, being the *Contrast*, *Correlation*, *Energy*, and *Homogeneity* of GLCM, for orientations 0^o, 45^o, 135^o, and 90^o , with a distance of 3 neighboring voxels. In this figure, we can see that the 2D 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.

3. 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 over-fitting, as well as the training time. Since we extract the 2D features with a range of orientation and distance for the sake of generalization, there exist highly correlated 2D feature clusters, as shown in Figure 4a. In this simple synthetic scenario, many 2D 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% most significant features in this case. The corresponding F-value as a function of feature ID is shown in Figure 4b, in which we can see that the 2D 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 and Lin, 2006). Random forest (RF) is a classification method and it also provides the branch weights that can represent feature importance (Breiman, 2001). A 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 dataset 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 feature selection. 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. In the end, 51 features are selected in the synthetic example.

4. Support Vector Classification

Recently, support vector machine (SVM) has been an effective classification method by constructing hyperplanes with a maximal margin in a multi-dimensional space, which separates different cases of different class labels (Boser et al., 1992; Cortes and 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:

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

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 $y_i\left(\omega^T\phi\left(x_i\right) + b\right) \geq 1 - \xi_i$. 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 equation 2, a simplified problem is obtained:

$$\begin{aligned} &\max_{\mathbf{a}} = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j K\left(x_i, x_j\right), \\ &\sum_{i=1}^{N} a_i y_i = 0, C \geq a_i \geq 0, \quad i = 1, ..., N,\ j = 1, ..., N, \end{aligned} \tag{3}$$

where, **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 function (RBF) in equation 3 is used as the kernel function for SVM:

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

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 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 0.96. 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.

5. Test on new data

After obtaining the trained SVM model, we apply it to the test data 1 and 2 in Figures 5a and 6a. The predicted event detection results considering both 1D and 2D features are shown in Figures 5b and 6b and the results considering only 1D features are shown in Figures 5c and 6c. We can see that, when the noise level is $SNR = -10dB$, both of them result in a reasonable prediction with 95% (Figure 5b) and 90% (Figure 5c) accuracy, respectively. There is an obvious improvement in the prediction accuracy by considering

2D texture features. When the noise level reaches $SNR = -13dB$, the predicted result in Figure 6cusing only 1D features is quite noisy with a prediction accuracy of 82%. However, by maximizing the information per segment with extracting extra 2D texture features, the proposed workflow still ends up with a reasonably good result with 93% accuracy (Figure 6b). Furthermore, five different traces (at 150, 525, 900, 1275, $1650m$) of clean, noisy, and predicted detection of test data ($SNR = -10dB$) are shown in Figure 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:

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

where $d\left(i\right)$ 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 results using the STA/LTA method are shown in Figures 5d and 6d. 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 Figures 5e and 6e. We can see that it works well in detecting events, even though not as good as the proposed workflow in Figures 5b and 6b. 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 is available.

The classification metrics of different strategies for the test data 2 ($SNR = -13dB$) 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 $7.5m$ and the time duration is $2s$. Since the STA/LTA method cannot perform well without a denoising preprocess applied to the raw data, we only compare SVM with only 1D features, SVM with 1D and 2D features, and CNN in this example. For the training, only one raw training data, which is shown in Figure 8a, is considered. The corresponding labeled training data is shown in Figure 8b. We labeled the datasets 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 $0.062s$, which is approximately twice the wavelength. There are 2970 segments in total. Part of the extracted 2D texture features is demonstrated in Figure 9. We can see that the 2D features show large relevance with the events. The correlation matrix of the features is shown in

Figure 10a. Compared to Figure 4a, the correlation between 2D feature clusters is reduced, because in this more complex scenario, different 2D features (*Contrast*, *Correlation*, *Energy*, and *Homogeneity* with orientations 0^o, 45^o, 135^o, and 90^o and distances of $1-8$) start to make a difference, instead of just creating redundancy. Furthermore, the univariate score (F-value) is shown in Figure 10b. It is obvious that the 2D 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 datasets are shown in Figures 11a, 12a, and 13a. The event detection results considering both 1D and 2D features are shown in Figures 11b, 12b, and 13b. The event detection results considering only 1D features are shown in Figures 11c, 12c, and 13c. We can clearly see the improvement of accuracy by using both 1D and 2D features, because the continuity, smoothness, and regularity of the events are largely emphasized when we label the training datasets, however, being ignored with only 1D feature extraction. In addition, the predicted results using the CNN approach are shown in Figures 11d, 12d, and 13d. 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 dataset 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 dataset. The random forest estimator automatically computes the normalized feature importance metric, and the top ten most important features are listed in Table 4. We notice that the top nine features are 2D features, which is in accordance with expectation, since 2D features carry more information than 1D features. Furthermore, for the most important feature, being the 135° orientation correlation with distance of 2 neighboring voxels (ID 121), we plot its partial dependence in Figure 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 dataset. 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 labeling 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 over-fitting.

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

Assuming that a larger number of high-quality labeled training datasets 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 over-fitting. 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 support vector machine 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 1D time/spectral-domain features and 2D 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 datasets 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 datasets 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 datasets 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., O. Ovcharenko, and D. Peter, 2017, A robust neural network-based approach for microseismic event detection, *in* SEG Technical Program Expanded Abstracts 2017: Society of Exploration Geophysicists, 2929–2933.

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

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

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

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.

Candes, E., L. Demanet, D. Donoho, and L. Ying, 2006, Fast discrete curvelet transforms: Multiscale Modeling & Simulation, **5**, 861–899.

Chen, Y., 2018, Automatic microseismic event picking via unsupervised machine learning: Geophysical Journal International, **212**, no. 1, 88–102.

Chen, Y., J. C. Hill, W. Lei, M. Lefebvre, E. Bozdağ, 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.

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

Cortes, C., and V. Vapnik, 1995, Support-vector networks: Machine learning, **20**, 273–297.

Dokht, R. M., H. Kao, R. Visser, and B. Smith, 2019, Seismic event and phase detection using time–frequency representation and convolutional neural networks: Seismological Research Letters, **90**, 481–490.

Elad, M., and M. Aharon, 2006, Image denoising via sparse and redundant representations over learned dictionaries: IEEE Transactions on Image processing, **15**, 3736–3745.

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

Gelchinsky, B., and V. Shtivelman, 1983, Automatic picking of first arrivals and parameterization of traveltime curves: Geophysical Prospecting, **31**, 915–928.

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.

Haralick, R. M., K. Shanmugam, et al., 1973, Textural features for image classification: IEEE Transactions on systems, man, and cybernetics, 610–621.

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

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

Huang, N. E., Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, 1998, The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis: Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences, The Royal Society, 903–995.

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

Knapmeyer-Endrun, B., and C. Hammer, 2015, Identification of new events in apollo 16 lunar seismic data by hidden markov model-based event detection and classification: Journal of Geophysical Research: Planets, **120**, 1620–1645.

Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012, Imagenet classification with deep convolutional neural networks: Advances in neural information processing systems, 1097–1105.

LeCun, Y., B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, 1990, Handwritten digit recognition with a back-propagation network: Advances in neural information processing systems, 396–404.

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

Liu, W., S. Cao, Y. Chen, and D. Zhang, 2016, Application of variational mode decomposition in random noise attenuation and time-frequency analysis of seismic data: Presented at the 78th EAGE Conference and Exhibition 2016.

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

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

Michelet, S., and M. N. Toksöz, 2007, Fracture mapping in the soultz-sous-forêts geothermal field using microearthquake locations: Journal of Geophysical Research: Solid Earth, **112**.

Mousavi, S. M., G. Beroza, and W. Zhu, 2018a, Earthquake signal detection using a deep convolutional-recurrent network: Presented at the AGU Fall Meeting Abstracts.

Mousavi, S. M., S. P. Horton, C. A. Langston, and B. Samei, 2016, Seismic features and automatic discrimination of deep and shallow induced-microearthquakes using neural network and logistic regression: Geophysical Journal International, **207**, 29–46.

Mousavi, S. M., and C. A. Langston, 2016a, Adaptive noise estimation and suppression for improving microseismic event detection: Journal of Applied Geophysics, **132**, 116–124.

———, 2016b, Hybrid seismic denoising using higher-order statistics and improved wavelet block thresholding: Bulletin of the Seismological Society of America, **106**, 1380–1393.

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

Perol, T., M. Gharbi, and M. Denolle, 2018, Convolutional neural network for earthquake detection and location: Science Advances, **4**, e1700578.

Provost, F., C. Hibert, and J.-P. Malet, 2017, Automatic classification of endogenous landslide seismicity using the random forest supervised classifier: Geophysical Research Letters, **44**, 113–120.

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

Rouet-Leduc, B., C. Hulbert, N. Lubbers, K. Barros, C. J. Humphreys, and P. A. Johnson, 2017, Machine learning predicts laboratory earthquakes: Geophysical Research Letters, **44**, 9276–9282.

Scheffe, H., 1967, The analysis of variance: Wiley.

Senkaya, M., and H. Karsli, 2014, A semi-automatic approach to identify first arrival time: the cross-correlation technique (cct): Earth Sciences Research Journal, **18**, 107–113.

Shapiro, S., C. Dinske, and E. Rothert, 2006, Hydraulic-fracturing controlled dynamics of microseismic clouds: Geophysical Research Letters, **33**.

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**, A47–A52.

Trnkoczy, A., 1999, Topic understanding and parameter setting of sta/lta trigger algorithm: New manual of seismological observatory practice, **2**.

Vaezi, Y., and M. Van der Baan, 2015, Comparison of the sta/lta and power spectral density methods for microseismic event detection: Geophysical Journal International, **203**, 1896–1908.

Vautard, R., P. Yiou, and M. Ghil, 1992, Singular-spectrum analysis: A toolkit for short, noisy chaotic signals: Physica D: Nonlinear Phenomena, **58**, 95–126.

Warpinski, N. R., 2000, Analytic crack solutions for tilt fields around hydraulic fractures: Journal of Geophysical Research: Solid Earth, **105**, 23463–23478.

Withers, M., R. Aster, C. Young, J. Beiriger, M. Harris, S. Moore, and J. Trujillo, 1998, A comparison of select trigger algorithms for automated global seismic phase and event detection: Bulletin of the Seismological Society of America, **88**, 95–106.

Xia, Y., S. Ni, and X. Zeng, 2013, Twin enigmatic microseismic sources in the gulf of guinea observed on intercontinental seismic stations: Geophysical Journal International, **194**, 362–366.

Zhang, D., E. Verschuur, S. Qu, and Y. Chen, 2019, Multi-domain surface multiple leakage extraction using local primary-and-multiple orthogonalization, *in* SEG Technical Program

Expanded Abstracts 2019: Society of Exploration Geophysicists, 4585–4589.

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

Zheng, J., J. Lu, S. Peng, and T. Jiang, 2017, An automatic microseismic or acoustic emission arrival identification scheme with deep recurrent neural networks: Geophysical Journal International, **212**, 1389–1397.

Zhu, W., and G. C. Beroza, 2018, Phasenet: a deep-neural-network-based seismic arrival-time picking method: Geophysical Journal International, **216**, 261–273.

LIST OF TABLES

LIST OF FIGURES

tion, Energy, Homogeneity, for orientations 0^o, 45^o, and 90^o, and 135^o, with a distance of 3 neighboring voxels.

10 Field data example: (a) correlation matrix of the 191 1D and 2D features. Note that 2D 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.

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

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

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

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.

15 Field data example: (a) labeled training data based on a relaxed criterion; (b-d) predicted event detection of raw test data 1-3 based on this relaxed labeling criterion using both 1D and 2D 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 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 centered 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% 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 1: The list of the extracted 1D time/spectral-domain feature.

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

Table 2: The list of the extracted 2D texture features

Parameters	**SVM+1D/2Dfeatures**	**SVM+1Dfeatures**	**CNN**
precision	0.93	0.82	0.91
recall	0.92	0.82	0.90
F1-score	0.92	0.82	0.90

Table 3: Classification metrics for test data 2 (Figure 6a)

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

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

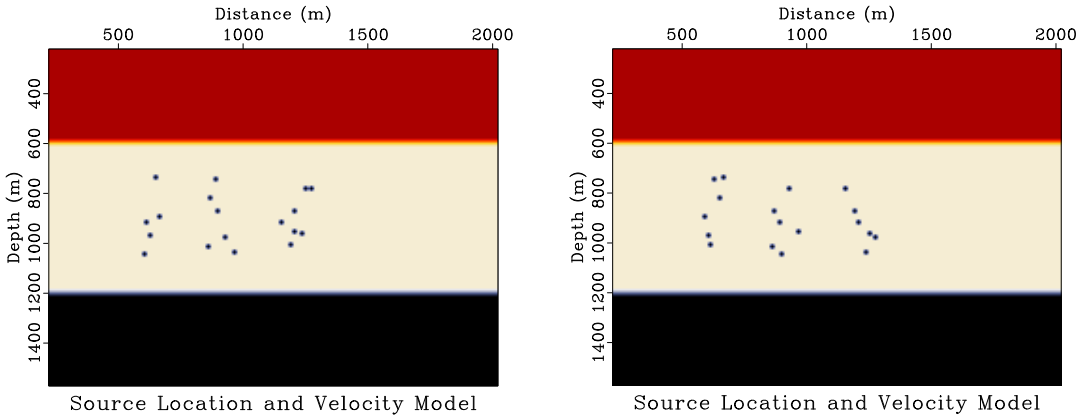


Figure 1: Synthetic example: the geometry and velocity model for the modeling of (a) raw training data with $SNR = -13dB$, (b) raw test data.

Qu et al. –

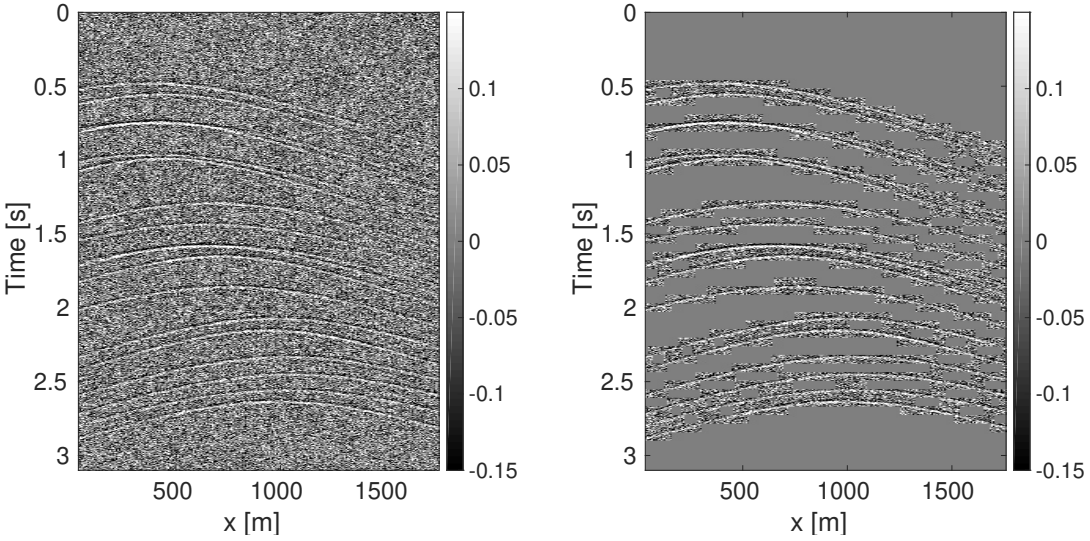


Figure 2: Synthetic example: (a) raw training data, (b) labeled training data.

Qu et al. –

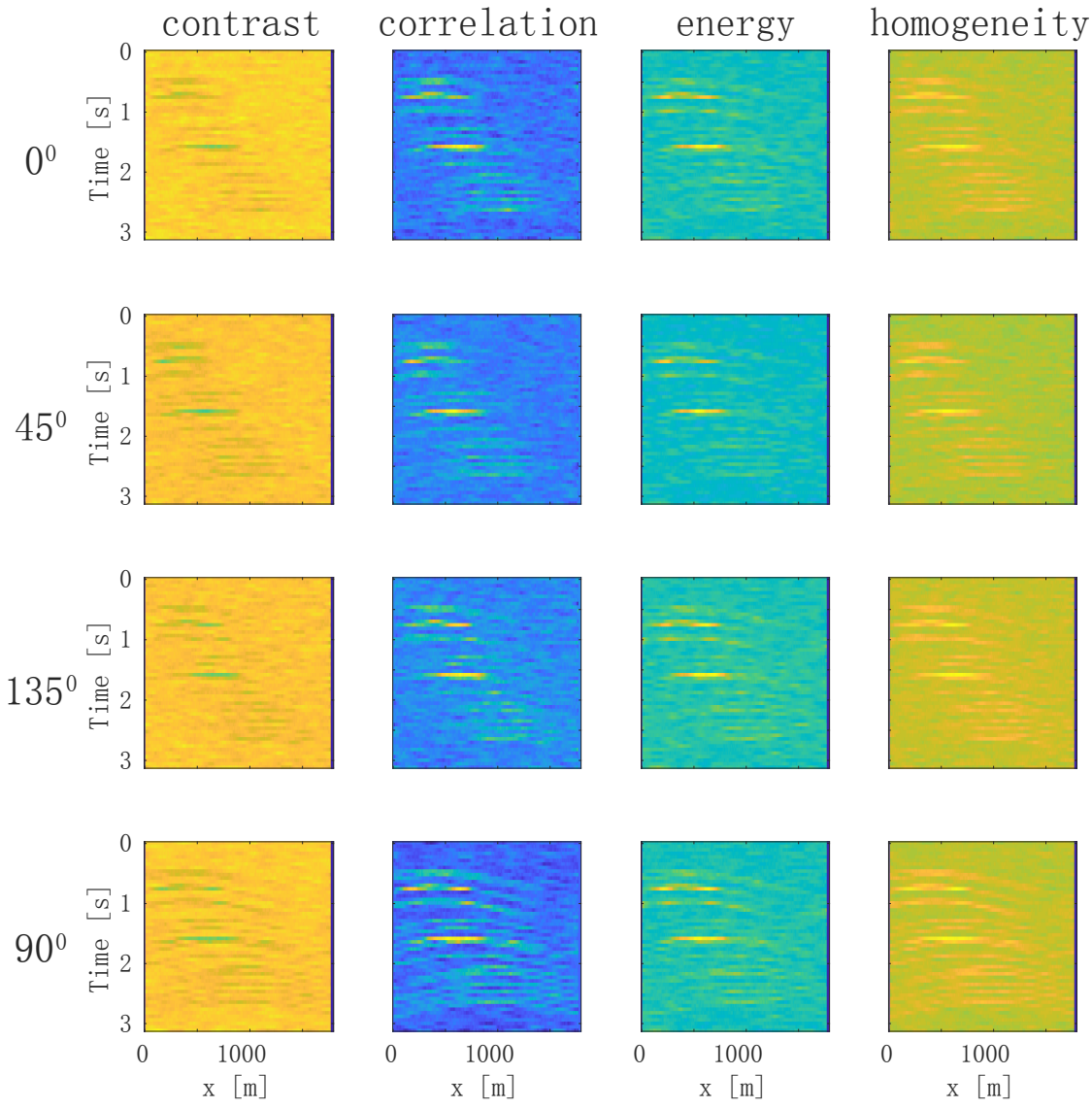
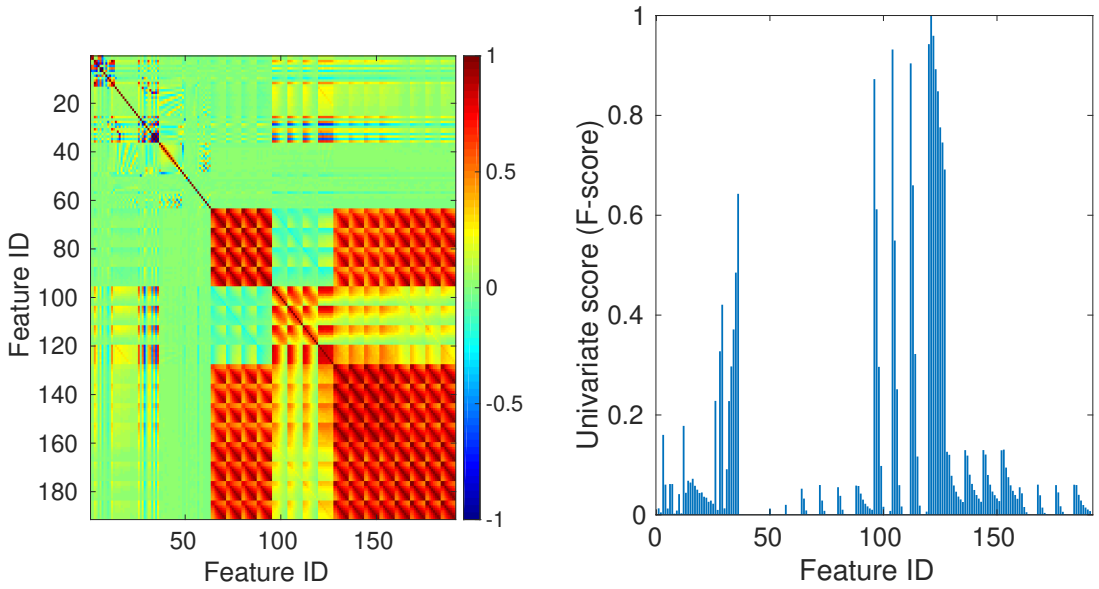


Figure 3: Synthetic example: 2D texture features of the training data: Contrast, Correlation, Energy, Homogeneity, for orientations 0^o, 45^o, 90^o, and 135^o, with a distance of 3 neighboring voxels.



(a) (b)

Figure 4: Synthetic example: (a) correlation matrix of the 191 1D and 2D features. Note that 2D 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. The feature IDs are expained in Table 1 and 2.

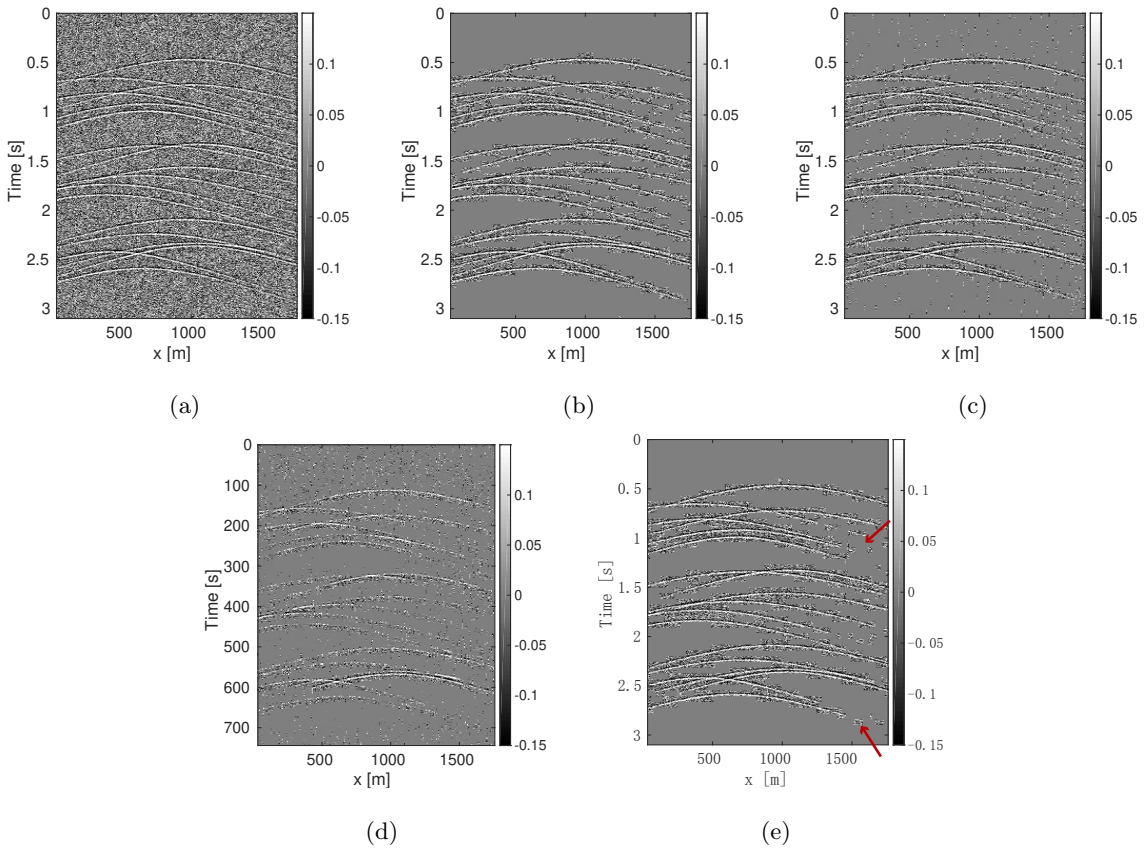


Figure 5: Synthetic example: (a) raw test data 1 ($SNR = -10dB$); predicted event detection for test data 1 ($SNR = -10dB$) using (b) both 1D and 2D features, (c) only 1D features, (d) a conventional LTA/STA method, (e) a CNN approach. The arrows point at locations where the CNN fails.

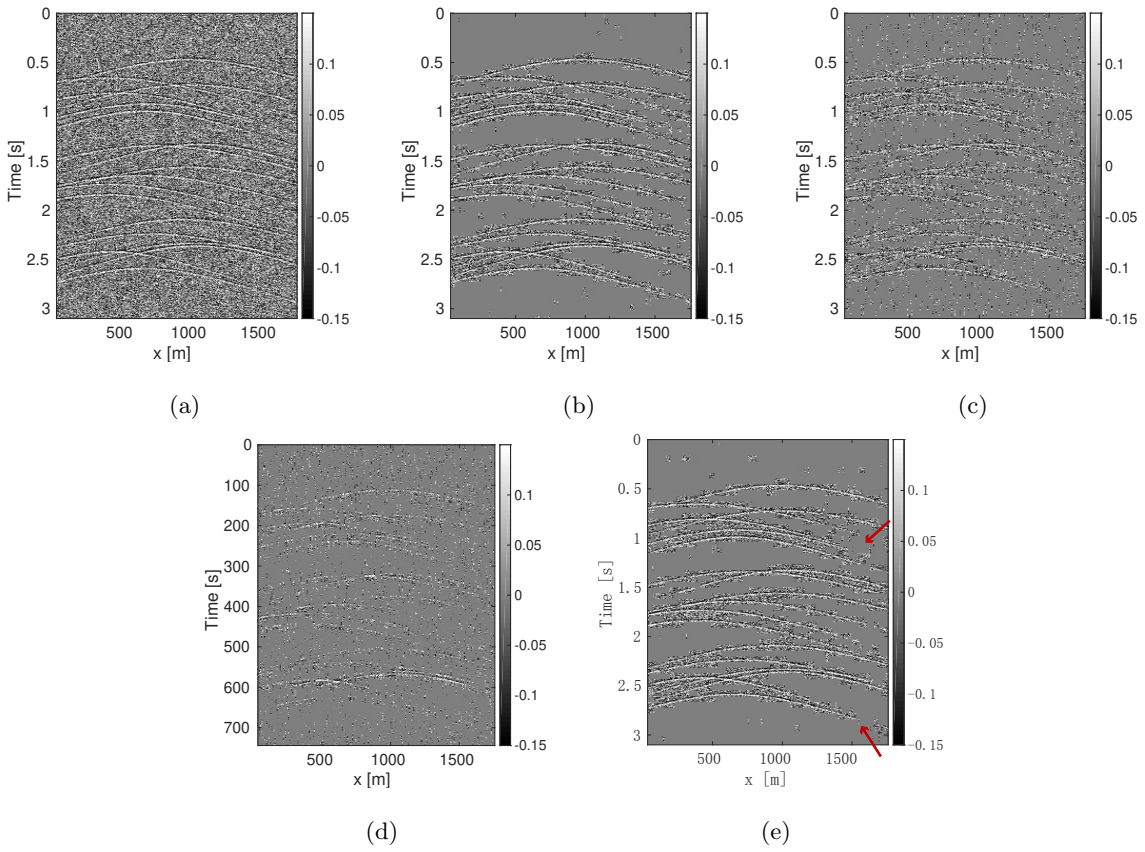


Figure 6: Synthetic example: (a) raw test data 2 ($SNR = -13dB$); predicted event detection for test data 2 ($SNR = -13dB$) using (b) both 1D and 2D features, (c) only 1D features, (d) a conventional LTA/STA method, (e) a CNN approach. The arrows point at locations where the CNN fails.

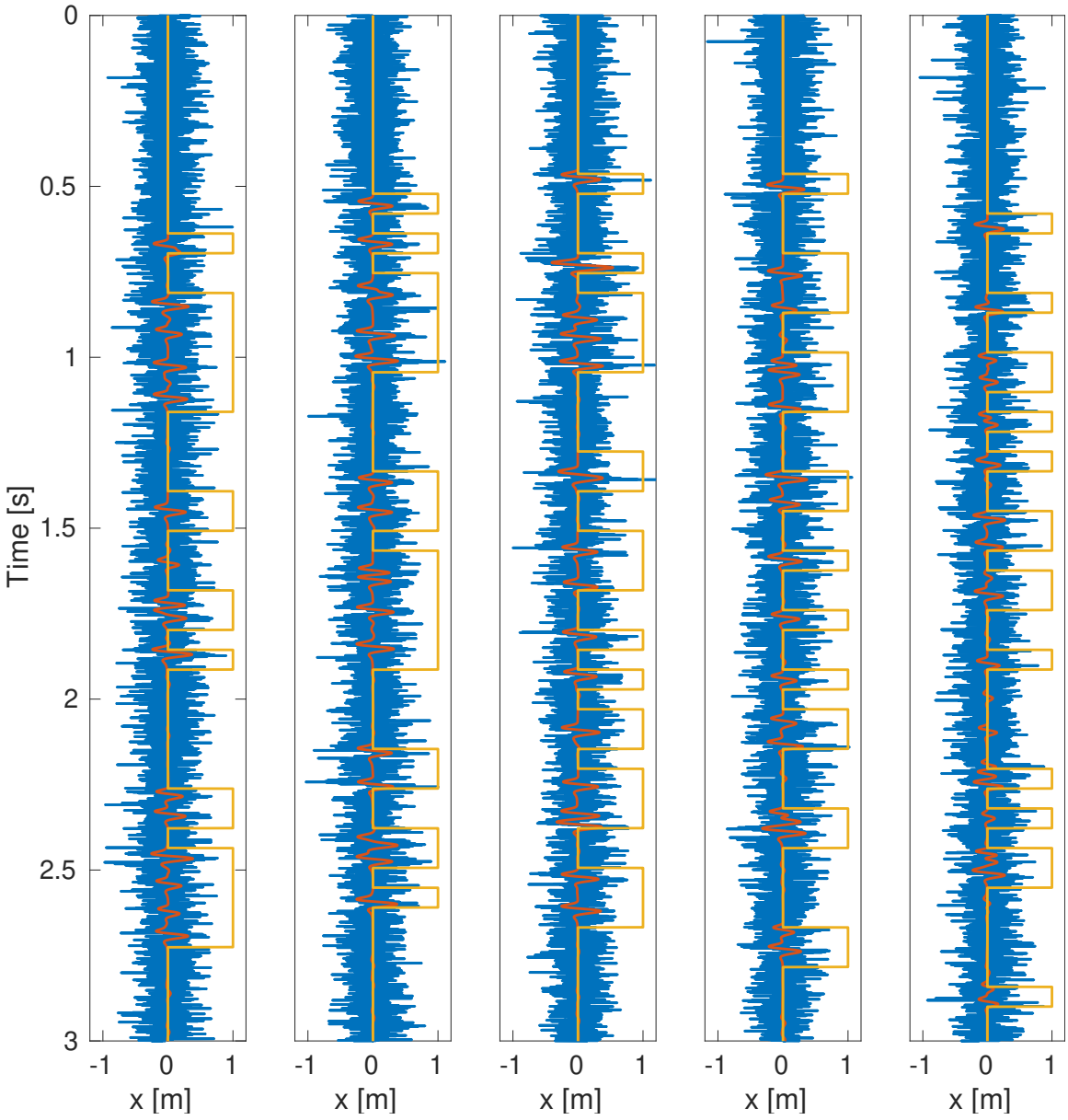


Figure 7: Synthetic example: Five traces (at 150, 525, 900, 1275, 1650m) of clean (red) and noisy (blue) test data 2 with $SNR = -13dB$ and the predicted event detection (yellow; 0-noise, 1-event).

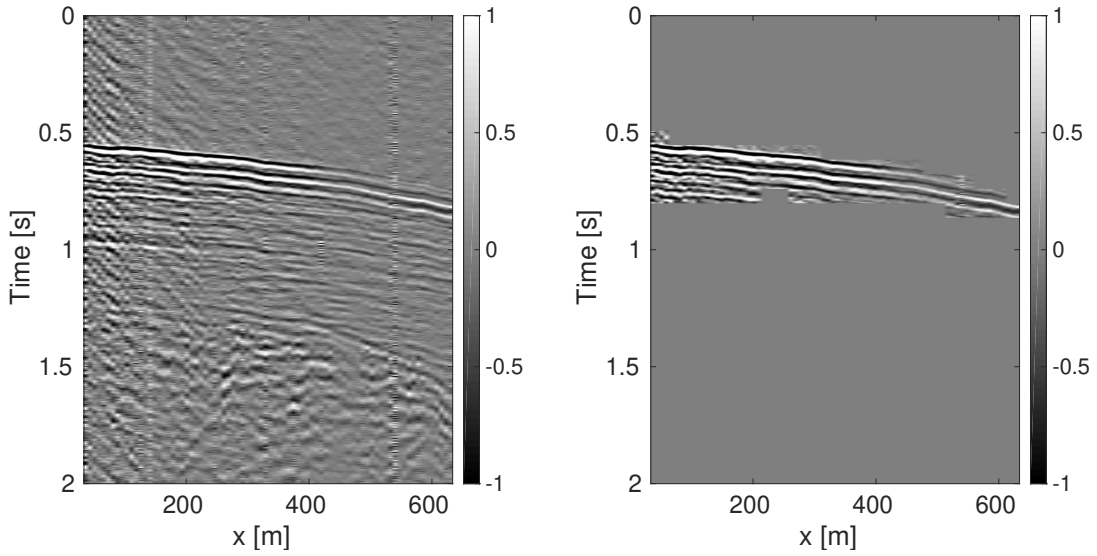


Figure 8: Field data example: (a) raw training data, (b) labeled training data.

Qu et al. –

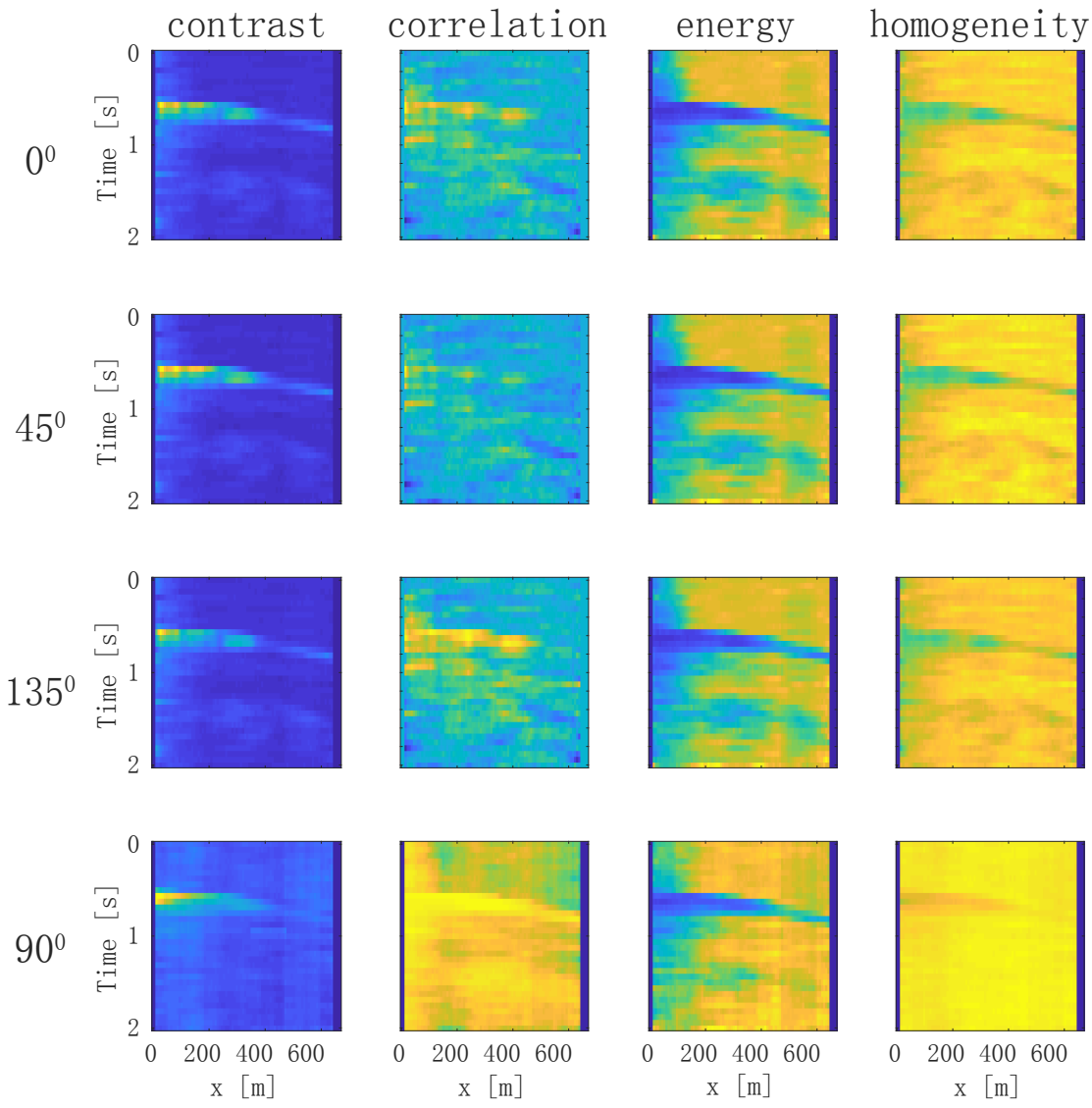
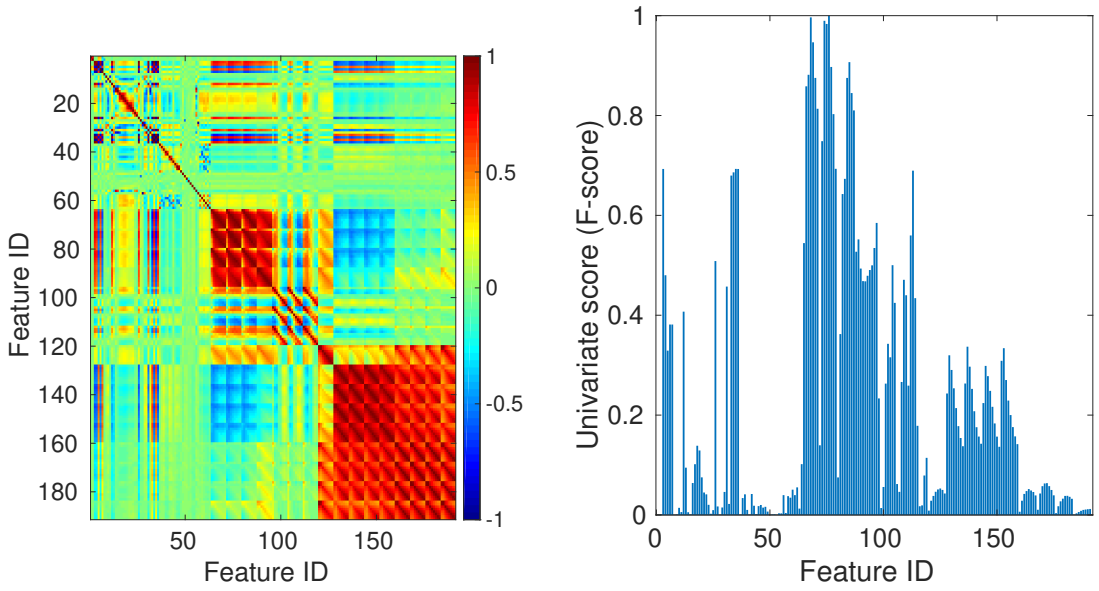


Figure 9: Field data example: 2D texture features of the training data: Contrast, Correlation, Energy, Homogeneity, for orientations 0^o, 45^o, and 90^o, and 135^o, with a distance of 3 neighboring voxels.



(a) (b)

Figure 10: Field data example: (a) correlation matrix of the 191 1D and 2D features. Note that 2D 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.

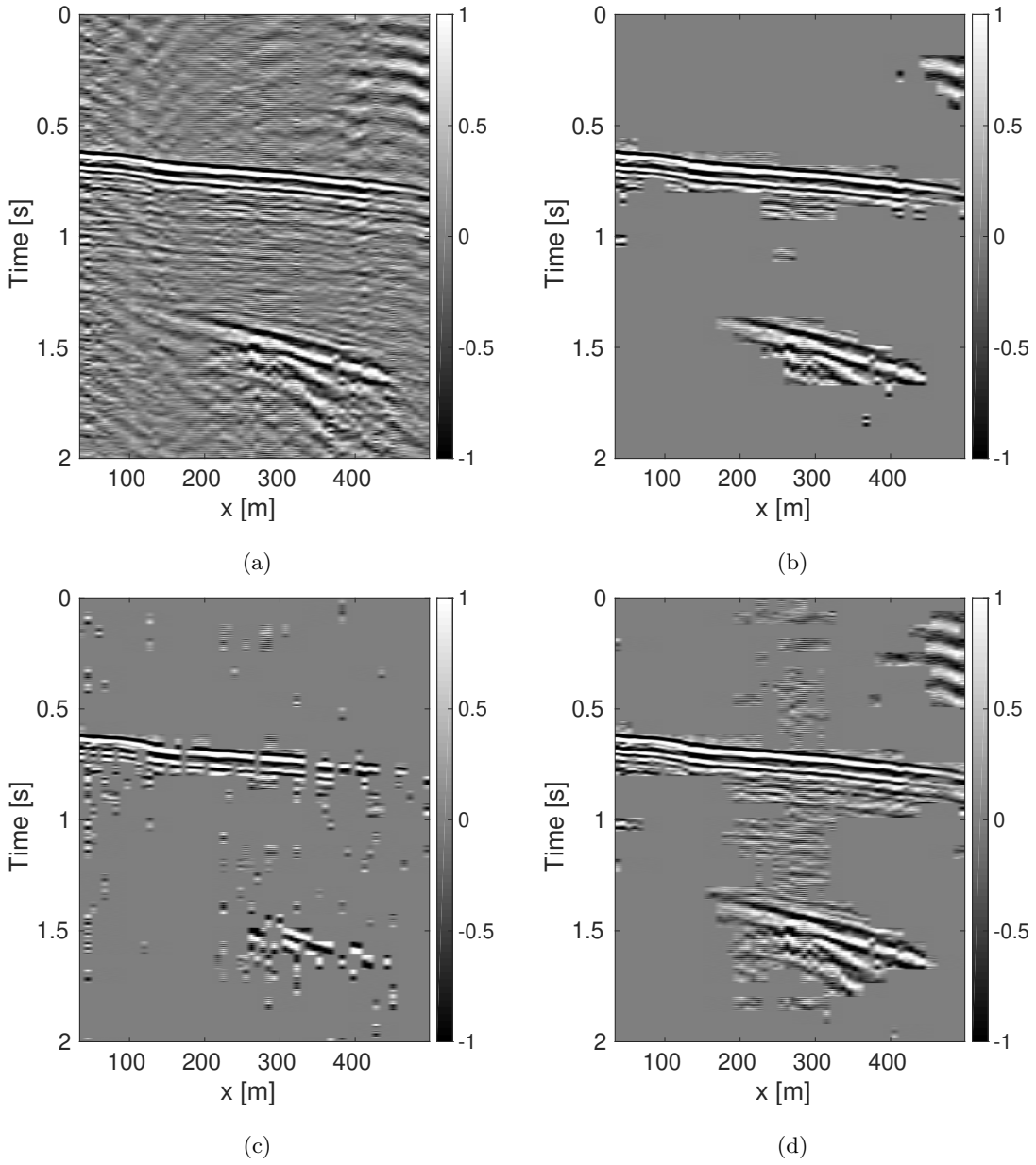


Figure 11: Field data example: (a) raw test data 1, (a) predicted event detection using (b) both 1D and 2D features, (c) only 1D features, (d) a CNN approach.

Qu et al. –

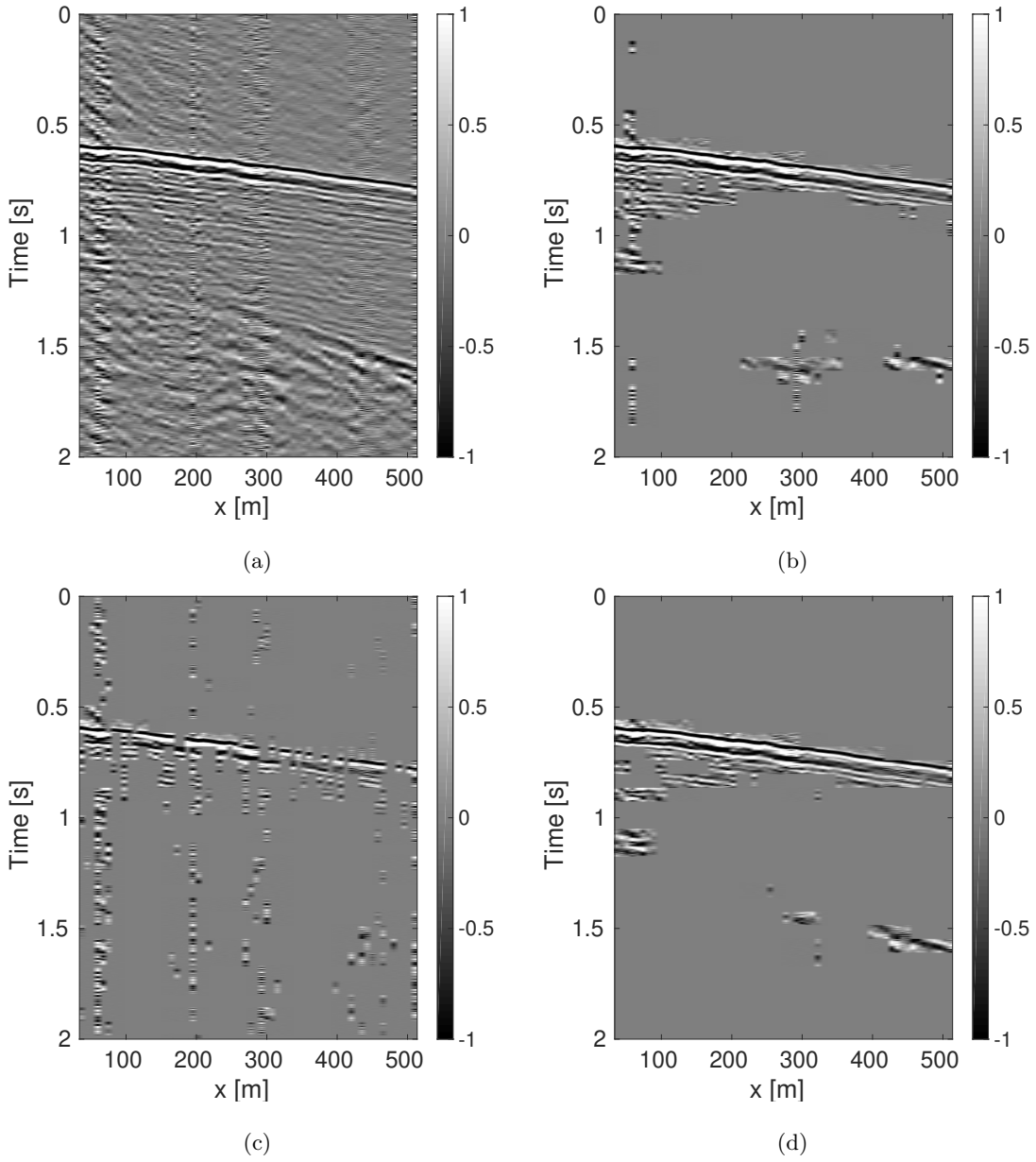


Figure 12: Field data example: (a) raw test data 2, (a) predicted event detection using (b) both 1D and 2D features, (c) only 1D features, (d) a CNN approach.

Qu et al. –

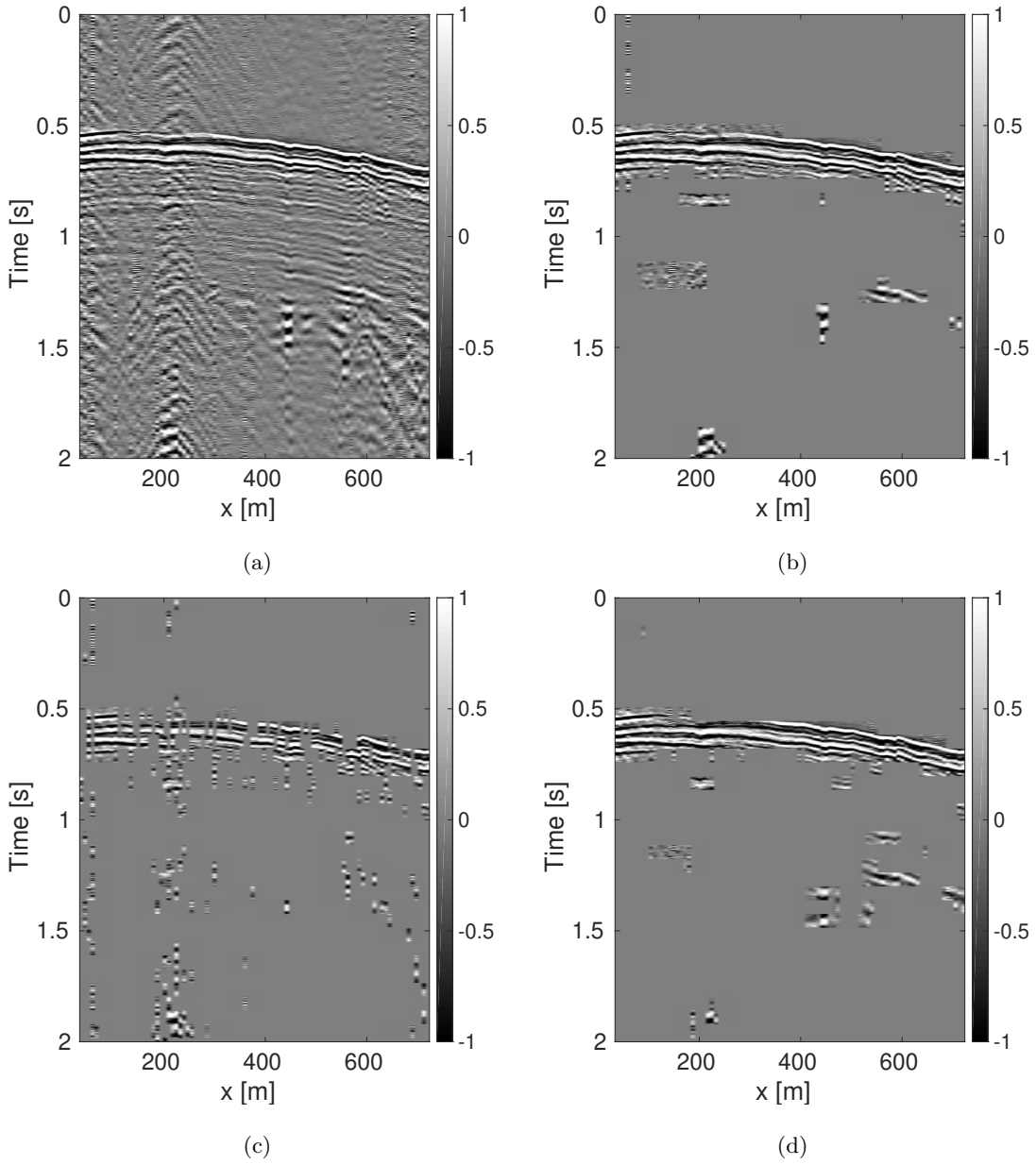


Figure 13: Field data example: (a) raw test data 3, (a) predicted event detection using (b) both 1D and 2D features, (c) only 1D features, (d) a CNN approach.

Qu et al. –

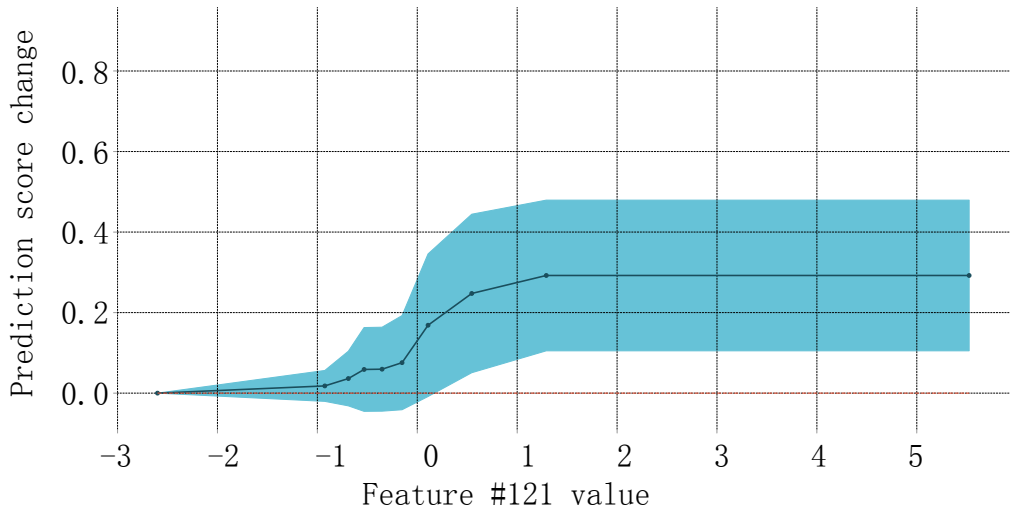


Figure 14: Partial dependence plot for feature #121. A positive feature #121 value would substantially increase the possibility of detecting a microseismic event, and the shaded area denotes the level of confidence.

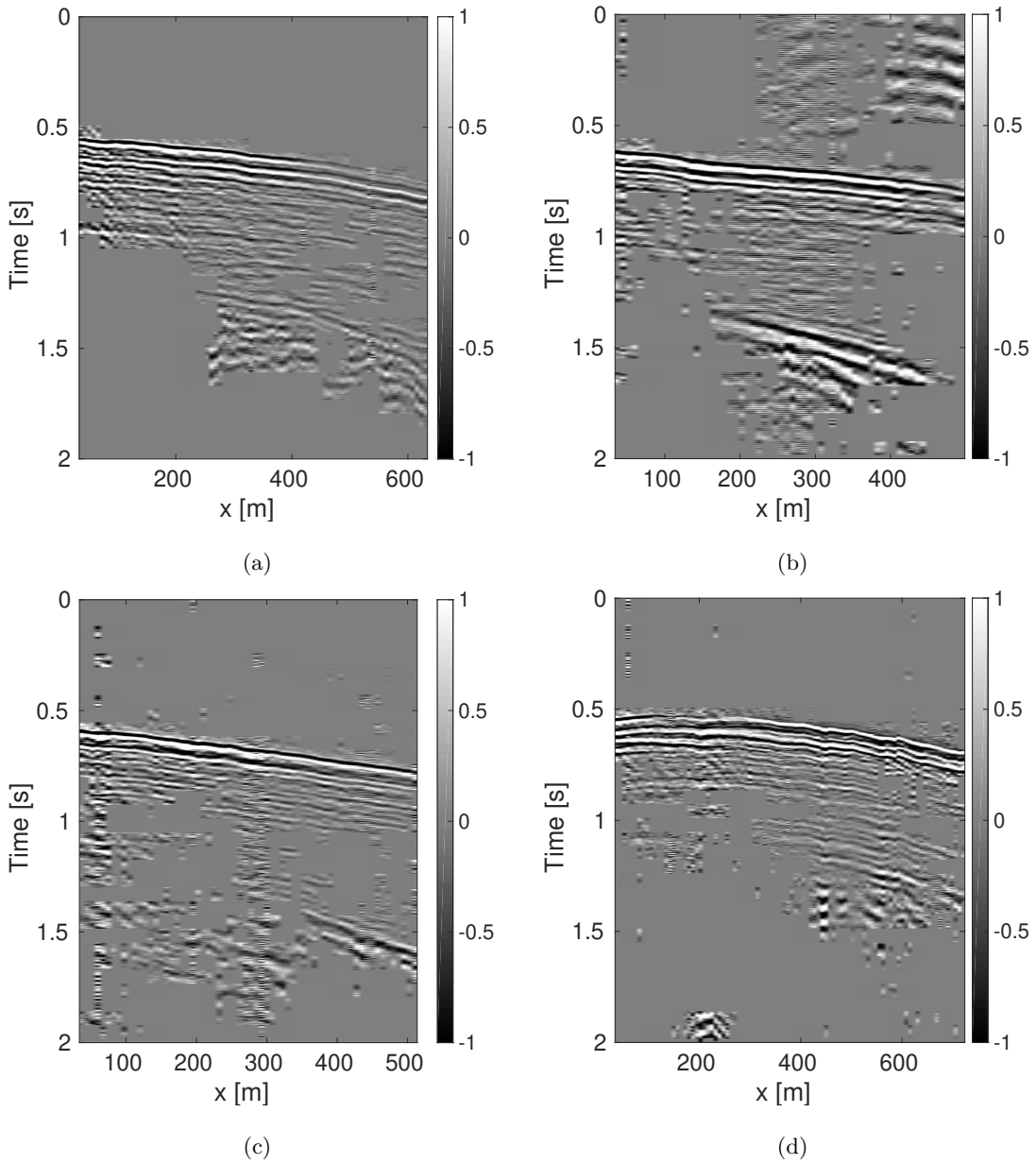


Figure 15: Field data example: (a) labeled training data based on a relaxed criterion; (b-d) predicted event detection of raw test data 1-3 based on this relaxed labeling criterion using both 1D and 2D features.

Qu et al. –