Unsupervised Clustering of Seismic Signals Using Deep Convolutional Autoencoders

S. Mostafa Mousavi, Weiqiang Zhu, William Ellsworth, and Gregory Beroza

Abstract—**In this paper, we use deep neural networks for unsupervised clustering of seismic data. We perform the clustering in a feature space that is simultaneously optimized with the clustering assignment, resulting in learned feature representations that are effective for a specific clustering task. To demonstrate the application of this method in seismic signal processing, we design two different neural networks consisting primarily of full convolutional and pooling layers and apply them to: (1) discriminate waveforms recorded at different hypocentral distances and (2) discriminate waveforms with different first-motion polarities. Our method results in precisions that are comparable to those recently achieved by supervised methods, but without the need for labeled data, manual feature engineering, and large training sets. The applications we present here can be used in standard single-site earthquake early warning systems to reduce the false alerts on an individual station level. However, the presented technique is general and suitable for a variety of applications including quality control of the labeling and classification results of other supervised methods.**

I. Nomenclature

seismic signal, waveform discrimination,unsupervised learning, neural networks, deep learning, clustering.

II. Introduction

Supervised and unsupervised learning form the two main classes of machine learning algorithms. Supervised methods learn a general underlying function, f, that maps an input feature vector, x, to an output label vector y, y = f(x). These algorithms use known training samples ((x(i); y(i)) pairs, where i is the sample number) to optimize the model. On the other hand, unsupervised methods directly learn patterns in data sets based on similarities among samples without relying on known examples.

The promising results of recent studies (e.g [1], [2], [3] [4]) suggest that machine-learning-based algorithms will be powerful substitutes for algorithms currently being used for routine processing of seismic signals in earthquake monitoring. The learning process in most of these studies, however, is supervised and relies on labeled data where the quantity and quality of labeled training sets play an important role in determining the effectiveness of an algorithm. When large, labeled data sets are unavailable, however, unsupervised techniques are important. With some exceptions (e.g. [5] and [6] for event detection, and [7] for picking), unsupervised approaches have received less attention in seismology. In this study, we apply deep neural networks to determine the polarity of the first motion of seismic signals and to discriminate local from teleseismic events in an unsupervised (or self-supervised) fashion. These applications could play important roles in Earthquake early warning (EEW) systems and earthquake monitoring more generally.

EEW systems use the information contained in the early part of local seismic waveforms to detect and characterize earthquakes rapidly and in real time during their early stages to provide advance warnings of impending ground motion [8]. Because the speed of telecommunications exceeds the speed of seismic waves, such alert information can provide invaluable time for both people and automated systems to take actions to mitigate seismic damage and losses. Recently, deep-learning-based models have shown promise for improving the reliability of single-station-based EEW systems (e.g. [9]). These methods take advantage of large labeled datasets of local earthquake and noise seismograms for discriminating between noise and earthquakes. Due to a lack of comparably large datasets of teleseismic events, however, susceptibility to false alarms caused by teleseismic signals remains an issue [9].

After an earthquake signal has been detected, and arrival times of *P* and *S* waves have been picked, the polarity of the first motions needs to be determined. First-motion polarities are used in focal mechanism estimation. [10] proposed a Bayesian approach to polarity picking. [1] showed that deep supervised learning to use convolutional and fully connected layers can be used for determining polarities of first arrivals.

Here, we recast the discrimination problem as a clustering problem. Clustering is a branch of unsupervised methods aiming to separate data into groups of similar data points. Our approach consists of a two-step clustering based on deep neural networks. In the first step, we use under-complete fully convolutional autoencoders to extract features automatically from input seismic data and reduce the feature dimensions. Reducing the dimensionality of the data and performing the clustering in the feature space instead of the data space is known to improve clustering performance. In the second step, the deep network's weights are iteratively fine-tuned by simultaneous optimization of feature learning and clustering assignment such that the network learns features that increase the effectiveness of clustering. This approach can have a variety of applications in seismology. Here we demonstrate its effectiveness for two specific tasks: discrimination of local from teleseismic waveforms and determining the first-motion polarity of local seismograms.

S. Mostafa Mousavi, Weiqiang Zhu, William Ellsworth, Gregory Beroza are with Department of Geophysics, Stanford University, Stanford, California, USA, e-mail: mmousavi@stanford.edu

S. Mostafa Mousavi is the corresponding author.

III. Methodology

In machine learning, clustering usually refers to a subcategory of unsupervised learning methods that are used for partitioning data into groups of similar objects without relying on known examples (labels). It is common to use clustering algorithms, such as k-means [11] to cluster data directly based on distance metrics. However, the effectiveness of these traditional clustering methods decreases as the dimension of input data increases [12]. Hence, dimensionality reduction techniques and feature engineering are used to first transfer input data into a feature space of lower dimension, then clustering is performed in that lower dimensional feature space. Clearly, the choice of this feature space plays a crucial role in clustering performance.

Recently, a group of clustering methods have emerged that employ the capability of deep neural networks in automatic learning of clustering-friendly features (e.g. [13], [14] [15], [16], [17]). These methods improve clustering performance by simultaneously learning the parameters of a deep neural network that maps input data into a lower-dimension feature space, and a set of cluster centroids in that feature space. We follow this approach to construct a deep clustering network for earthquake signal processing. Our network consists primarily of an under-complete fully convolutional autoencoder.

Autoencoders are neural networks that learn to reconstruct their input [18]. They are composed of two parts: an encoder and a decoder. The encoder learns a nonlinear function that maps the input data into a hidden representation (or code). The decoder aims to reconstruct the input using this hidden representation in a way that minimizes a reconstruction loss. One approach to force an autoencoder to learn the most salient features of input data is to constrain the code size to be smaller than the input dimension. These architectures are called under-complete autoencoders and are commonly used for dimensionality reduction and automatic learning of salient representations.

Our network architecture is shown in Figure 1. A clustering layer is connected to the autoencoder's bottleneck. The clustering layer assigns the hidden features of each sample to a cluster. The network's loss function is composed of two parts: reconstruction loss of the autoencoder, Lr, and clustering loss, Lc. The objective is to minimize a loss function that is a weighted combination of these two losses:

$$L = (1-\lambda)L_r + \lambda L_c \ , \qquad (1)$$

where $\lambda \in [0,1]$ is a hyper-parameter that balances Lr and Lc. The clustering is done in two steps. In the initial pre-training step, we train the convolutional autoencoder (by setting λ=0) to learn an initial set of sparse features by minimizing the mean squared error of the reconstruction.

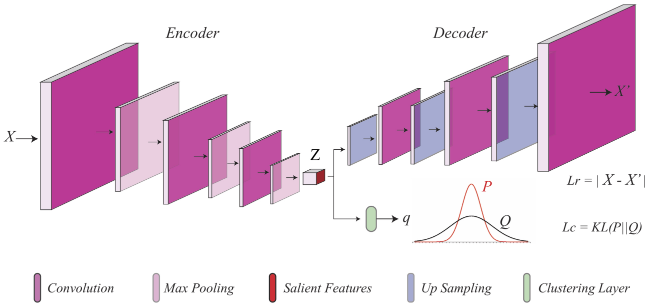

Fig. 1. Network architecture. The encoder and decoderare composed of fully convolutional layers followed by max-pooling and up-sampling layers respectively. Reconstruction loss (Lr) and the clustering loss (Lc) are given in the figure.

After the initial pre-training step, the learned features are used to initialize the cluster centroids (μ_j) in the feature space using k-means. Next, during a fine-tuning stage, cluster assignment and feature learning are jointly performed (by setting λ=0.1). Choosing a large λ would distort the latent feature space and also requires a smaller learning rate that make the training slow. On the other hand, too small λ also eliminate the effect of clustering layer. Hence, we used 0.1 which is a commonly used value by other researchers (e.g. [13]). In this step, first the degree of similarity or the membership probabilities (q_{ij}) between each embedded point, z_i, and cluster centroids, μ_j, are calculated using Student's t-distribution [19]:

$$q_{ij} = \frac{(1+\|z_i-\mu_j\|^2)^{-1}}{\sum_j (1+\|z_i-\mu_j\|^2)^{-1}} \ . \qquad (2)$$

The membership probabilities are used to compute an auxiliary target distribution, p_{ij}:

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})}, \qquad (3)$$

and clustering is performed by minimizing the Kullback–Leibler(KL) divergence between the soft assignments, q_{ij}, and the target distribution, p_{ij}:

$$L_c = KL(P \parallel Q) = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) \ . \qquad (4)$$

During the fine-tuning step, by iteratively refining the cluster's centroids and updating the autoencoder's weights, more clustering-friendly features are learned, and the clustering effectiveness improves as a result. Although this approach can be classified as an unsupervised learning technique (since no labeled data is used), the training process is done in a self-supervised fashion [20] where the targets are generated from the input data.

q_{ij} provide measures of similarities of extracted features from each data point to different cluster centroids. Higher q_{ij}values simply implies higher confidence in assigning a data point to a particular cluster. The auxiliary target distribution, p_{ij}, puts more emphasize on data points assigned with higher confidence while it normalizes the loss contribution of each centroid. Hence, by minimizing the divergence between the membership probabilities (eq-2) and the target distribution (eq-3) during the fine-tuning stage, the network iteratively learns from high-confidence assignments and refines the clusters.

Here we designed the network as a fully convolutional network to reduce the learnable parameters of the network and make it applicable for cases with relatively small training sizes. Moreover, unlike the original paper [13], in the encoder section the feature size is reduced constantly, and we keep the decoder connected to the network during the fine-tuning step. This will help preserveing the data structure of the feature space. We use mini-batch stochastic gradient descent and back-propagation for optimization.

IV. Discriminating Local from Teleseismic Seismograms

We acquired local and teleseismic earthquake waveforms from the data set of [9]. It includes broadband and strong-motion records from the Southern California Seismic Network (SCSN), and records from the Japanese K-NET and KiK-net strong motion networks (surface stations only). Here we used only 9.2k vertical component seismograms (almost equally contain teleseismic and local waveforms). Local seismograms have hypocentral distances less than 150 km while teleseismic ones range from distances of 1000 to 10,000 km. All seismograms were resampled to 100 Hz, de-meaned, detrended, normalized, and band-passed filtered from 1 to 20 Hz. Applying some sorts of normalization to the data is a common practice to improve the performance in machine learning applications. Here, the ban-pass filtering meant to have a similar effect. Teleseismic waveforms are depleted in high-frequency energy, however, even in this frequency band, domain-crafted features struggle to differentiate teleseismic from local signals with sufficient accuracy.

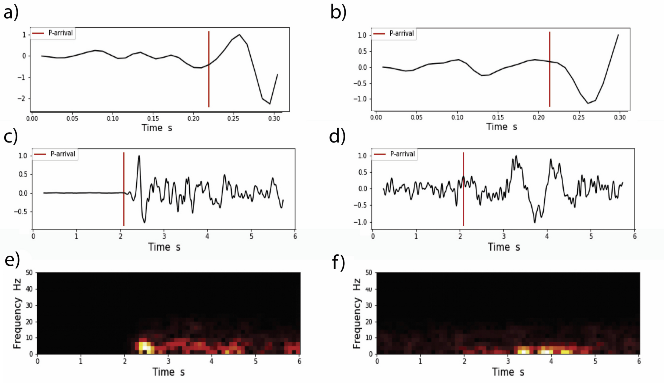

Fig. 2. Samples of short windows time series with upward (a) and downward (b) first polarity motions. c) and d) are two samples of raw time series of local and teleseismic seismograms respectively. Associated short-time Fourier spectrograms of local and teleseismic events are given in (e) and (f) respectively. P arrivals are marked by solid vertical lines.

In our previous study [21] we obtained a better result for discriminating earthquake depth between different seismic signals working in the time-frequency domain. Because of this and the distinct frequency differences between the local and teleseismic data, we also performed the discrimination in the time-frequency domain. We used the short-time Fourier transform to construct spectrograms (Figure 2e-f). The autoencoder used for local/teleseismic discrimination consists of 6 fully convolutional layers (2D convolution) with tangent activation functions (full details of the network architecture is given in the supplementary materials). In the Encoder section each convolutional layer is followed by a Pooling layer that combines the outputs of multiple neurons from the previous layer into a single neuron of the next layer, hence reducing the dimension of learned features as data is fed forward. Upsampling layers in the decoder section perform the opposite. The input and output are a 768 dimensional matrix (16 frequency bins at 48 time points) and the bottleneck dimension is 12.

We continue the fine-tuning process till the clustering accuracy does not improve more than 0.001. The autoencoder is trained for 100 epochs and the fine-tuning step is run for about 6000 iterations. The entire pre-training and fine-tuning took about 25 minutes on a laptop with a 2.7 GHz Intel Core i7 processor and 16 GB of memory which most of this processing time was for the fine-tuning step.

We evaluated the performance of the method using precision, recall, F-score, and clustering accuracy as metrics. Precision for each class is defined as the number of correctly classified samples divided by the total number of assigned samples to each class. The recall is defined as the fraction of instances that are correctly classified, and the F-score combines these two parameters to eliminate effects of unbalanced sample size for different classes. The clustering accuracy measures the overall precision over all classes and is defined as:

$$ACC = \max_{m} \frac{\sum_{i=1}^{n} 1\{y_i = m(c_i)\}}{n}, \tag{5}$$

where c_i is the cluster assignment, y_i is the ground-truth label, and m is a mapping function that ranges over all possible one-to-one mappings between assignments and labels. Intuitively, this determines the most likely group assignment in the ground truth corresponding to the arbitrary cluster labels, 1 or 2. The Hungarian algorithm [22] is used to calculate the mapping function.

The visualization of data and accuracy of clustering using the k-means algorithm are presented in Figure 3. To reduce the dimension of the data to two dimensions for the purpose of the visualization we used t-sne method [24]. Applying the k-means clustering on raw data in the time domain resulted in about 50 % accuracy (Figure 3a). Visualization of the raw seismograms (in reduced dimension) color-coded based on the actual class, help us to understand the very poor performance of k-means on the time series data. Applying k-means to the spectrograms (time-frequency domain) significantly improves the clustering performance to 85% (Figure 3b). After the pre-training of the autoencoder, we used it to transfer the spectrograms into a feature space (Z in Figure 1) with a lower dimension and applied k-means clustering (Figure 3c). This slightly improves the performance to 87 % accuracy. Next, we perform the fine-tuning and update the autoencoder weights every 150 iterations. This gradually increases performance as more useful features are learned and optimized for the clustering task. Figure 3d-f show the improvement of the clustering results at different iterations of the fine-tuning step as a result of extracting better features from the input spectrograms to separate the two types of seismic events more effectively. After 5700 iterations the clustering accuracy increased to 98.41 %. The precision, recall, and F-score for local seismograms are 0.98, 0.99, and 0.98 respectively. While the corresponding values for the teleseismic ones are: 0.99, 0.98, and 0.98. [9] trained a random forest classifier using a set of hand-selected features extracted from ~ 237k samples and were able to classify 95 % of teleseismic records correctly. Our method achieved a better performance, using a fraction of the data, without need for the label, and feature engineering. There were not enough samples at different SNR to perform the noise sensitivity test, however, the feature extraction process should be robust to the background noise.

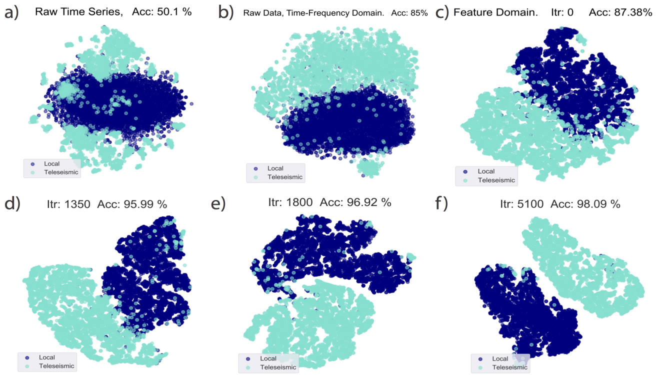

Fig. 3. t-sne visualizations of local and teleseismic data in time domain (a), time-frequency domain (b), feature domain after the pretraining (c), feature domain after 1350 iterations (d), 1800 iterations (e), and 5100 iterations during the fine-tuning (f). The accuracy of the k-mean clustering is given on top of each plot and samples are color-coded based onclass labels. Notice that as the method is trained, the two populations are increasingly divided into well-separated clusters.

V. Application for First-Motion Polarity Determination

To test the ability of the deep-clustering method to determine the first-motion polarity of the seismic signal, we used 250,000 vertical seismograms recorded at local distances (< 120 km) in southern California [1]. The seismograms used here are associated with events with magnitudes from 0 to 5.5. We only used records with picked P arrival time and assigned upward and downward first-motion polarity as determined by analysts. We then used 32 samples (0.32 s) around the P-arrival as the input to our deep-clustering neural network Figure 2. The data preprocessing is the same as previous section however the inputs to the network are time series waveforms. The autoencoder used for polarity determination has 8 fully convolutional layers (1D convolution) with tangent activations where the input and output dimensions are 32 and the bottleneck dimension is 2. The entire process, including pre-training and fine-tuning took about 30 minutes.

We obtain an overall accuracy of 0.97. The precision, recall, and F-score for upward polarities are 0.99, 0.82, and 0.9 respectively. While the corresponding values for the downward polarities are: 0.72, 0.99, and 0.83. [1] achieved an overall precision of 0.97 and 0.93 for classifying upward and downward motions respectively using their supervised approach. Our unsupervised approach achieved higher precision for upward polarities and similar to the [1] we obtained relatively lower precision for the downward motions. This could be due to the larger fraction of mis-classified downward first motions for low snr/small magnitudes documented by [1]. Our method resulted in higher false positive rates for the downward motions compared to the supervised approach; however, this was achieved using only 2% of the data used in the supervised method [1] and with much lower training costs.

To test the sensitivity of the method to the noise level, we divide the data into 5 SNR bins with 50k samples in each bin. The accuracy is 0.91 for the SNR of 0-1 and increases to 0.985 for records with SNR>4. Results for each SNR bin indicate that deep-unsupervised clustering does not require a large number of training samples to achieve good results.

VI. Discussion and Conclusion

Here we used deep neural networks for unsupervised clustering of seismic data. We designed two different networks; one with 2D and the other with 1D fully convolutional layers for discriminating waveforms with different hypocentral distances and first-motion polarities. The networks consist of an autoencoder and a clustering layer and improve the clustering performance by learning salient features from the input data that are the most suitable for the clustering task of interest.

The size of bottleneck layer is important for the clustering performance. Although one of the main function of the proposed network architeture is automatic feature extraction and dimensionality reduction, we found that too small a dimension leads to a lower clustering accuracy due to a higher reconstruction error. The choice of network dimensions depends on the complexity of the problem. Unfortunately there is no rule of thumb for selecting the bottleneck dimension. An empirical approach would be starting with an overcomplete autoencoder and gradually lowering the bottleneck dimension during the pre-training step till the reconstruction loss start increasing significantly.

For the local/teleseismic discrimination problem, the input data are 2D spectrograms with dimensions of 768. The autoencoder reduces the input dimension to 12 features at the bottleneck layer. We plotted the input spectrograms and learned features for four local and four teleseismic samples in Figure 4a and Figure 4b. Although it is difficult to interpret what these features represent and how the 12-dimensional features learned by the network link to the input data, the learned features have far simpler and more distinct patterns than the input spectrograms, which facilitates the clustering objective. The simple and lowdimensional features that have been learned during the pre-training and the fine-tuning enable effective clustering despite the relatively small training sample size. We repeated the whole clustering for a smaller dataset (1000 local and 1000 teleseismic seismograms) and obtained 94 % accuracy.

For the first-motion polarity discrimination task, the network reduces the 32 time series samples of the input waveforms to 2 features. This very-low feature dimension is used to cluster waveforms into two groups of upward and downward motions. In Figure 4c to Figure 4e, we have plotted a zoomed window around the P-arrival time and the values of the associated features in at the network's bottleneck for 12 different samples. It seems that the network has learned a very simple feature pattern that discerns the waveforms with the polarity of the waveform. In Figure 4 values of learned features have been color-coded. Clustering is done based on the similarity of the feature maps (patterns). From Figure 4 c and d it is completely clear that the feature maps for different clusters are very distinct. The clustering is done based on the patterns (variation of the values among two features) not specific value.

For instance, we see that for the upward examples, values of the first feature are always higher than the second feature but for the downward ones values of the first feature are lower than the values for second features. From Figure 4e we see that most of the false positive downward polarities (that are the reason for the lower precision of downward motions) are due to mislabeling. We observed many similar samples in the data set where a tiny downward motion right after the arrival was ignored and the sample was labeled as upward motion by the analyst. However, surprisingly the small features learned by the network are sensitive to these subtle changes in the polarity. This suggests an additional potential application of the clustering method for double checking the labeling and classification results of other supervised methods.

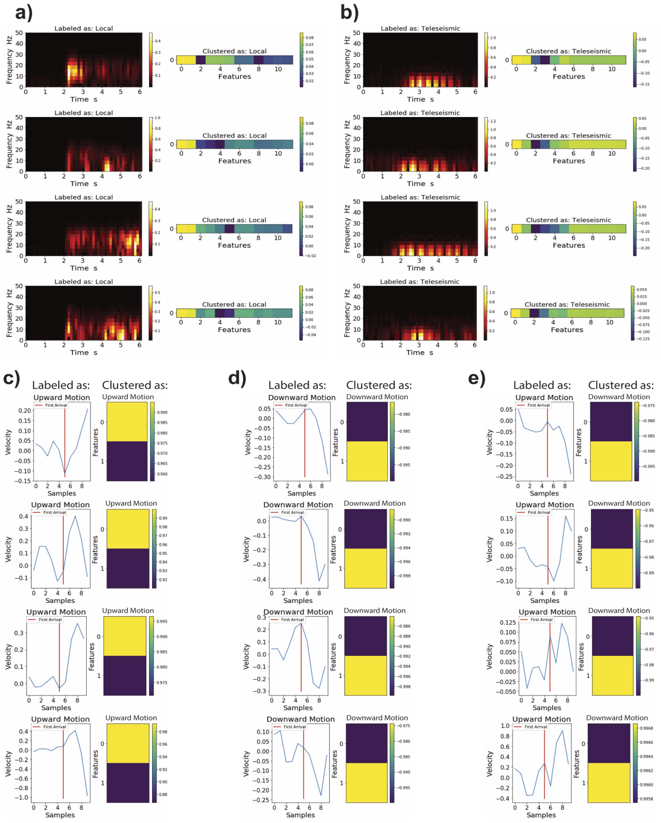

Fig. 4. the network inputs (spectrograms) and visualization of learned features at the network's bottleneck after the fine-tuning stage for four local events (a) and four teleseismic events (b). Zoomed windows of 8 samples around the P-arrival time and the associated learned features for four true positive upward polarity motions (c), four true positive downward polarity motions (d), and four false positive downward motions (e). Feature value have been color codded.

Acknowledgment

Authors would like to thank three anonymous reviewers for their insightful commends and recommendations. This study was supported by the Stanford Center for Induced and Triggered Seismicity. Waveform data and metadata was obtained from the Southern California Earthquake Data Center (http://scedc.caltech.edu/research-tools/deeplearning.html#signal_noise).

References

[1] Z. E. Ross, M. A. Meier, and E. Hauksson, "P-wave arrival picking and first-motionpolarity determination with deep learning," *Journal of Geophysical Research*, vol. 123, pp. 5120–5129, 2018.

[2] W. Zhu and G. C. Beroza, "PhaseNet: A Deep-Neural-Network-Based Seismic Arrival Time Picking Method," *Geophysical Journal of International*, vol. 216, no. 1, pp. 261–273, 2019.

[3] S. M. 3.Mousavi, W. Zhu, Y. Sheng, and G. C. Beroza, "CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection," *scientific report*, 2019. [Online]. Available: arXiv:1810.01965

[4] G. Zhang, Z. Wang, and Y. Chen, "Deep learning for seismic lithology prediction," *Geophysical Journal International*, vol. 215, no. 2, pp. 1368–1387, 2018.

[5] C. E. Yoon, O. O'Reilly, K. J. Bergen, and G. C. Beroza, "Earthquake detection through computationally efficient similarity search," *Science Advances*, vol. 1, pp. 1 501 057–10, 2015.

[6] Y. Chen, "Fast waveform detection for microseismic imaging using unsupervised machine learning," *Geophysical Journal of International*, vol. 10, no. 1093, 2019. [Online]. Available: 10.1093/gji/ggy34

[7] ——, "Automatic microseismic event picking via unsupervised machine learning," *Geophysical Journal of International*, vol. 212, no. 1, pp. 88–102, 2018.

[8] S. Minson, M. Meier, A. Baltay, T. Hanks, and E. Cochran, "The limits of earthquake early warning: Timeliness of ground motion estimates," *Science Advances*, vol. 4, no. 3, p. 0504, 2018.

[9] M. A. Meier, Z. E. Ross, A. Ramachandran, A. Balakrishna, S. Nair, P. Kundzicz, Z. Li, J. Andrews, E. Hauksson, and Y. Yue, "Reliable Real-time Seismic Signal/Noise Discrimination with Machine Learning," *Journal of Geophysical Research*, 2019.

[10] D. J. Pugh, R. S. White, and P. A. F. Christie, "Automatic Bayesian polarity determination," *Geophysical Journal International*, vol. 206, no. 1, pp. 275–291, 2016.

[11] J. . MacQueen, *Some methods for classification and analysis of multivariate observations*. Oakland, CA: USA, 1967, vol. 1.

[12] M. Steinbach, E. Levent, and V. K., "The challenges of clustering high dimensional data," in *New Directions in Statistical Physics*, 2004, pp. 273–309.

[13] J. Xie, R. Girshick, and F. A, "Unsupervised deep embedding for clustering analysis," in *International Conference on Machine Learning*, 2016.

[14] X. Peng, S. Xiao, J. Feng, W. Y. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior," in *International Joint Conference on Artificial Intelligence*, 2016.

[15] K. G. Dizaji, A. Herandi, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. ," 2017. [Online]. Available: arXivpreprintarXiv:1704.06327

[16] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *International Conference on Machine Learning*, 2017, pp. 3861–3870.

[17] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Lomg, "A Survey of Clustering With Deep Learning: From the Perspective of Network Architecture," *IEEE Access*, vol. 6, pp. 2169–3536, 2018.

[18] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," in *JMLR*. JMLR, 2010.

[19] I. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 2579, no. 2605, 2008.

[20] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *international conference on Information and knowledge management*, 2000.

[21] S. M. Mousavi, S. P. Horton, C. A. Langston, and B. Samei, "Seismic Features and Automatic Discrimination of Deep and Shallow Induced-Microearthquakes Using Neural Network and Logistic Regression," *Geophysical Journal of International*, vol. 207, no. 1, pp. 29–46, 2016.

[22] H. W. Kuhn, "The Hungarian Methiod for the assignment problem," *Nav. Res. Logistics*, vol. 52, no. 1, pp. 7–21, 2005.