Unsupervised Deep Clustering of Seismic Data: Monitoring the Ross Ice Shelf, Antarctica

William Frost Jenkins[1,1,1], Peter Gerstoft[2,2,2], Michael Bianco[2,2,2], and Peter D Bromirski[3,3,3]

[1]University of California San Diego
[2]UC San Diego
[3]SIO/UCSD

November 30, 2022

Abstract

Advances in machine learning (ML) techniques and computational capacity have yielded state-of-the-art methodologies for processing, sorting, and analyzing large seismic data sets. In this work, we consider an application of ML for automatically identifying dominant types of impulsive seismicity contained in observations from a 34-station broadband seismic array deployed on the Ross Ice Shelf (RIS), Antarctica from 2014 to 2017. The RIS seismic data contain signals and noise generated by many glaciological processes that are useful for monitoring the integrity and dynamics of ice shelves. Deep clustering was employed to efficiently investigate these signals. Deep clustering automatically groups signals into hypothetical classes without the need for manual labeling, allowing for comparison of their signal characteristics and spatial and temporal distribution with potential source mechanisms. The method uses spectrograms as input and encodes their salient features into a lower-dimensional latent representation using an autoencoder, a type of deep neural network. For comparison, two clustering methods are applied to the latent data: a Gaussian mixture model (GMM) and deep embedded clustering (DEC). Eight classes of dominant seismic signals were identified and compared with environmental data such as temperature, wind speed, tides, and sea ice concentration. The greatest seismicity levels occurred at the RIS front during the 2016 El Niño summer, and near grounding zones near the front throughout the deployment. We demonstrate the spatial and temporal association of certain classes of seismicity with seasonal changes at the RIS front, and with tidally driven seismicity at Roosevelt Island.

Unsupervised Deep Clustering of Seismic Data: Monitoring the Ross Ice Shelf, Antarctica

William F. Jenkins II[1], Peter Gerstoft[1], Michael J. Bianco[1], Peter D. Bromirski[1]

[1]Scripps Institution of Oceanography, University of California San Diego, La Jolla, CA, USA

Key Points:

- Deep clustering identified classes of seismic signals with similar spectral and temporal features.
- Deep clustering can be adapted to various kinds of data sets, enabling rapid exploration of "big data" in seismology.
- Paired with environmental data, deep clustering could provide insights into the causes of seismicity.

Corresponding author: W. F. Jenkins II, `wjenkins@ucsd.edu`

Abstract

Advances in machine learning (ML) techniques and computational capacity have yielded state-of-the-art methodologies for processing, sorting, and analyzing large seismic data sets. In this work, we consider an application of ML for automatically identifying dominant types of impulsive seismicity contained in observations from a 34-station broadband seismic array deployed on the Ross Ice Shelf (RIS), Antarctica from 2014 to 2017. The RIS seismic data contain signals and noise generated by many glaciological processes that are useful for monitoring the integrity and dynamics of ice shelves. Deep clustering was employed to efficiently investigate these signals. Deep clustering automatically groups signals into hypothetical classes without the need for manual labeling, allowing for comparison of their signal characteristics and spatial and temporal distribution with potential source mechanisms. The method uses spectrograms as input and encodes their salient features into a lower-dimensional latent representation using an autoencoder, a type of deep neural network. For comparison, two clustering methods are applied to the latent data: a Gaussian mixture model (GMM) and deep embedded clustering (DEC). Eight classes of dominant seismic signals were identified and compared with environmental data such as temperature, wind speed, tides, and sea ice concentration. The greatest seismicity levels occurred at the RIS front during the 2016 El Niño summer, and near grounding zones near the front throughout the deployment. We demonstrate the spatial and temporal association of certain classes of seismicity with seasonal changes at the RIS front, and with tidally driven seismicity at Roosevelt Island.

Plain Language Summary

We demonstrate the ability of a machine learning technique called deep clustering to automatically identify different types of seismic signals. A neural network encodes spectrograms into simplified representations. Application of a clustering algorithm separates the representations into distinct clusters of signal types. The deep clustering technique was applied to seismic data recorded by an extensive array of broadband seismometers deployed on the Ross Ice Shelf (RIS), Antarctica from 2014 to 2017. In addition to knowing when and where on the RIS signals are detected, clustering enables users to determine the signal characteristics. Paired with environmental data, deep clustering can be used to identify whether certain environmental factors are associated with particular classes of seismicity.

1 Introduction

Ice sheets and ice shelves in West Antarctica are experiencing rapid change. Between 2003 and 2019, the West Antarctic Ice Sheet (WAIS) experienced a net ice loss of 169 billion tons per year, contributing 7.5 mm to sea level rise (Smith et al., 2020). Warming oceans are enhancing basal melting of ice shelves that reduces the buttressing of grounded ice sheets (De Angelis & Skvarca, 2003; Thoma et al., 2008; Pritchard et al., 2012; Paolo et al., 2015), leading to increased discharge of ice into the ocean and raising sea level (Scambos, 2004; Dupont & Alley, 2005; Rignot et al., 2014; Fürst et al., 2016). With West Antarctica alone containing a sea level rise potential of 5.6 m (Smith et al., 2020), monitoring the loss of ice shelves plays a critical role in anticipating future sea level rise and associated societal impacts on coastlines and the environment. Increased seismic activity, such as icequakes resulting from fracturing, can give indications of changes in iceberg calving rates and the integrity of ice shelves and are observable using glacial seismology methods (Aster & Winberry, 2017). However, the prevalence of extensive, continuously recording seismic observing systems has led to an abundance of data which is becoming increasingly difficult to analyze using conventional signal processing. At the same time, advances in computing capabilities and machine learning algorithms have enabled more efficient, data-driven approaches to study natural processes and phenomena. To analyze large seismic data sets more efficiently, we adapt contemporary machine learning techniques to augment existing signal processing and data analysis techniques.

Seismology is a data-intensive field with well-developed signal processing and analytical methods. The recent introduction of machine learning techniques has led to the development of complementary tools that give seismologists novel approaches to traditional analyses, such as earthquake detection and early warning, phase picking, ground-motion prediction, tomography, and geodesy (Kong et al., 2019; Bianco & Gerstoft, 2018; Bianco et al., 2019; Johnson et al., 2019). In this study we present an implementation of *clustering*, a form of unsupervised machine learning used to discover classes of similar signals within a data set (Bishop, 2006; Holtzman et al., 2018; Johnson et al., 2020), and which is commonly used as an exploratory tool for large, unlabeled data sets.

To test the applicability of clustering groups of similar signals for monitoring ice shelves, we focus specifically on the Ross Ice Shelf (RIS), Antarctica, where a 34-station passive seismic array was deployed from November 2014 to January 2017 to observe the response of the RIS to ocean gravity wave impacts and investigate the structural dynamics of the ice shelf (Bromirski et al., 2015). The array, shown in Figure 1, continuously recorded long- and short-period seismic signals that exhibited seasonal and spatial variations related to the shelf's coupling to the ocean, atmosphere, and crust (Baker et al., 2019). Signals and ambient noise of interest on the RIS include tidally-driven stick-slip seismicity at Whillans Ice Stream (Bindschadler, King, et al., 2003; Bindschadler, Vornberger, et al., 2003; D. A. Wiens et al., 2008); basal micro-earthquakes and tremor (Barcheck et al., 2018); tidally and thermally driven rift fractures (Olinger et al., 2019); diurnal seismicity associated with subsurface melting (MacAyeal et al., 2019); wind-generated resonance in the ice (Chaput et al., 2018); flexural and plate waves generated by ocean swell, infragravity waves, and tsunami (Bromirski & Stephen, 2012; Bromirski et al., 2017; Chen et al., 2018); regional and teleseismic earthquakes (Baker et al., 2020); and icequakes generated by ocean gravity waves (Chen et al., 2019). Ambient seismic noise, which can be used to estimate the RIS structure (Diez et al., 2016), also contains spectra from ocean gravity waves, whose dispersion can be used to identify their source distance and origin (Bromirski et al., 2015; Hell et al., 2019).

The seismic data recorded on the RIS are diverse and encompass numerous source mechanisms with a wide range of spatiotemporal variability. In this study, we apply two unsupervised clustering methodologies to the RIS array seismic data to identify classes of seismic events with similar temporal and spectral characteristics. The occurrences and distributions of these signal classes provide information on glaciological processes affecting ice shelf evolution.

2 Background

Grouping seismic signals with similar characteristics (clustering) allows investigation of spatiotemporal variability associated with glaciological processes that result from environmental forcing.

2.1 Clustering

There are numerous methods to cluster data, (Aggarwal & Reddy, 2014), many of which have been adapted for use in seismology and geophysics (Kong et al., 2019). A related approach based on sparse modeling, called dictionary learning, has been applied

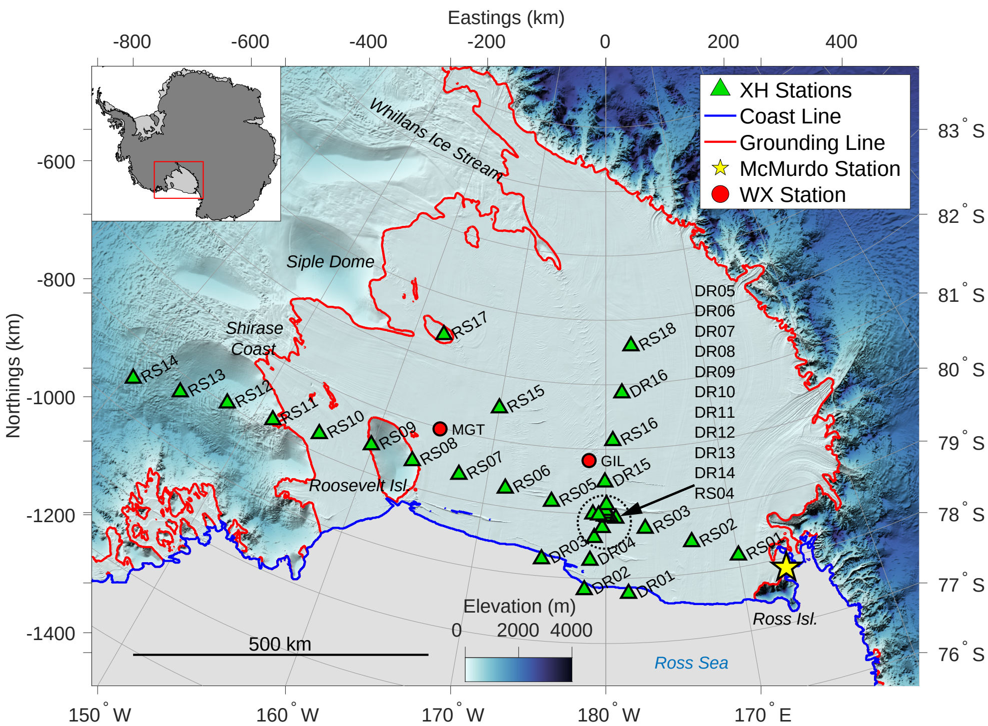

Figure 1. The passive broadband seismic array deployed from November 2014 to January 2017 consisted of 34 seismic stations and was deployed as part of the Ross Ice Shelf Dynamic Response to Wave-Induced Vibrations Project (Bromirski et al., 2015). RIS surface elevation, ice and water layer thicknesses, and grounding and coast lines were obtained from Bedmachine (Morlighem et al., 2017; Greene et al., 2017).

to regularizing seismic inverse problems (Bianco & Gerstoft, 2018; Bianco et al., 2019). Hierarchical clustering has been used by Mousavi et al. (2016) to automatically discriminate between shallow and deep earthquakes, and by Trugman and Shearer (2017) to more precisely localize earthquakes. Graphical clustering has been used to localize sources in a dense seismic array by Riahi and Gerstoft (2017), and by Telesca and Chelidze (2018) to cluster seismic events in time. Distance-based clustering, like the popular k-means algorithm, (MacQueen, 1967; Hartigan & Wong, 1979) has been used by Chamarczuk et al. (2020) to cluster seismicity based on features extracted from seismic data. Perol et al. (2018) used k-means to define probabilistic earthquake locations as part of their convolutional neural network (CNN) detection and localization technique. Wallet and Hardisty (2019) used Gaussian mixture model (GMM) clustering, which assumes clusters in the data exist that can be represented as linearly superimposed Gaussian distributions, enabling identification of seismic facies. Seydoux et al. (2020) detected and clustered seismic signals and background noise with the use of a deep scattering neural network and GMM.

Not all clustering methods involve machine learning. Template matching, in which a matched filter is constructed from a template waveform, is used to scan through continuous recordings to locate similar signals (Gibbons & Ringdal, 2006; Beaucé et al., 2018; Chamberlain et al., 2018). Yoon et al. (2015) and Bergen and Beroza (2018) presented computationally efficient techniques in which locality-sensitive hashing is used to map seismic signals into a hash table, allowing similar signals to be identified by table entry. Hotovec-Ellis and Jeffries (2016) developed an approach that uses correlation-based similarity search to automatically detect and cluster repeating volcanic seismicity in continuous data. Cole (2020) adopted the method of Hotovec-Ellis and Jeffries (2016) to cluster RIS array data at stations RS09, RS10, and RS11 in order to characterize tidal forcing of seismicity at these stations.

2.2 Dimensionality

Data are considered high-dimensional when many features are required to represent or describe the data. Seismic data represented as time series, spectrograms, scalograms, or energy envelopes can contain thousands of features (e.g., discrete samples in a time series, or bins in a spectrogram). Clustering performed directly on such input data is vulnerable to the "curse of dimensionality" (Bellman, 1961; Bishop, 2006; Murphy, 2012;

Aggarwal & Reddy, 2014), i.e., as the dimensionality of the input data increases, the number of data points required to maintain sufficient sampling density increases exponentially. A further consideration is that clustering error metrics can give less meaningful results as dimensionality increases.

As high-dimensional data are difficult to cluster (Aggarwal et al., 2001; Steinbach et al., 2004), dimensionality reduction remains a major focus of development (Yang et al., 2017). It is often desirable to transform the input data to a lower-dimensional representation described by fewer, more salient features. A popular approach is to use principal component analysis (PCA), which projects higher dimensional data into lower dimensional space (Goodfellow et al., 2016) and was used by Reddy et al. (2012) to compress seismic data to maximize feature variance.

The approach to reducing dimensionality in this study employs an autoencoder, a model whose output aims to reproduce its input via a series of non-linear transformations employing a deep neural network (DNN) (Hinton, 2006; Murphy, 2012; Yang et al., 2017). These non-linear transformations provide greater capacity in dimension reduction, and can better model data with low-dimensional representations than, for example, PCA. The autoencoder first encodes input data such as an image—in our case, a spectrogram—into a latent feature vector. Next, the autoencoder decodes the latent features and reconstructs the original image. Since the autoencoder provides a non-linear transformation of the data, it must be trained using gradient descent. In this iterative training, the error between the input and output is minimized. In doing so, the salient features of the data are learned by the network weights. With the dimensionality of the input data reduced in the latent feature space, clustering algorithms can be applied to the data's latent feature space.

2.3 Deep Embedded Clustering

In deep clustering, a DNN such as an autoencoder is used to reduce the dimensionality of the data. A recent deep clustering method that has shown improvement over traditional clustering techniques was developed by Xie et al. (2016), whose *deep embedded clustering* (DEC) consists of two processes: (1) An autoencoder is trained to represent the data's salient features; and (2) the encoding layers and clustering layer are jointly optimized. Yang et al. (2017) extended the approach in DEC by jointly optimizing the clustering step with training the entire autoencoder, not just the encoder layers. Additional variations of DEC have been proposed: Xie et al. (2016) used a stacked denoising autoencoder (Vincent et al., 2010) in their original implementation, but Min et al. (2018) employed autoencoders composed of CNN layers and other architectures. More recently, Chazan et al. (2019) developed an approach in which joint clustering is performed with a mixture of autoencoders, each representing a cluster, and Boubekki et al. (2021) demonstrated improved performance using a clustering algorithm that is jointly optimized with the embeddings of the autoencoder.

Mousavi et al. (2019) used DEC to predict whether seismic detections were local or teleseismic, and Snover et al. (2021) demonstrated the ability of DEC to cluster anthropogenically generated seismic noise. In a similar signal processing and clustering workflow to ours, Ozanich et al. (2021) compared DEC and GMM on spectrograms of acoustic data collected on a coral reef, but in their case found GMM performed better than DEC.

In this study, we implement GMM clustering in the latent feature space and compare its performance with DEC. Using RIS seismic data from December 2014 to November 2016, we identify several different classes of signals, and further demonstrate the utility of deep clustering as an exploratory tool for large, real-world seismic data sets by associating the clustering results with observed environmental factors.

3 Ross Ice Shelf (RIS) Seismic Array and Data

Each station in the RIS seismic array consisted of 3-component Nanometrics Trillium 120 PHQ seismometers emplaced 1 m below the surface of the ice, powered by solar panels during the austral summers, and lithium-ion batteries during the austral winters. Two subarrays comprised the array. The larger subarray consisted of 18 stations spaced approximately 80 km apart (prefix RS), primarily oriented parallel to the RIS front. The RS stations sampled short-period orthogonal components of ground velocity at a sampling rate of 100 Hz, except for two stations that sampled at 200 Hz. The smaller subarray consisted of 16 stations (prefix DR) arranged approximately orthogonal to the ice shelf front along the international date line, sampling ground velocity with a sampling rate of 200 Hz. For this study, we were primarily interested in the detection and classification of icequakes and local/regional earthquakes, using only vertical com-

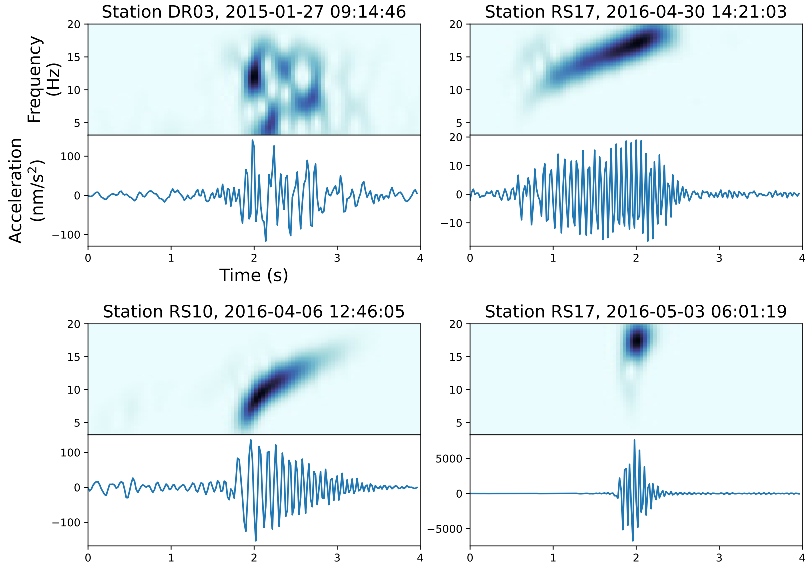

Figure 2. Seismic signals detected on the Ross Ice Shelf exhibited diverse characteristics with variation in time, space, and source mechanism. Shown are examples of acceleration response seismograms and their respective normalized spectrograms spanning the 3-20 Hz band that were typical for the data set. The normalized spectrograms were used as input to the deep clustering analysis.

ponent observations with frequencies of interest occurring between 3 and 20 Hz. This passband was selected to preserve impulsive signals, eliminate high-energy noise prevalent at low frequencies, and exclude resonances generated by wind at frequencies above 20 Hz. Representative types of signals detected are shown in Figure 2.

Seismic data from each station were processed in 24-hour segments as follows: 1) Data were linearly de-trended and tapered with a Hann window. 2) Instrument responses for all stations were removed, giving acceleration in m/s^2. 3) Since the bandwidth of interest was from 3 to 20 Hz, data were decimated to 50 Hz, using low-pass filtering followed-by downsampling. 4) A band-pass filter with cutoff frequencies at 3 and 20 Hz was applied to remove long-period signals originating from tides, tsunamis, infragravity waves, ocean swell, and teleseisms. 5) A short-term average/long-term average (STA/LTA) detection algorithm (Allen, 1982) was used to detect impulsive signals, particularly icequakes and local earthquakes, employing an STA window of 0.5 s, LTA window of 30 s, trigger threshold of 15, and de-trigger threshold of 10. The detector was applied to data from each station from 3 December 2014 to 21 November 2016 for a total of 719 days of array data, yielding 531,407 detections.

Upon detection, a 4 s trace centered on the spectral peak of each triggered event was saved for processing. Centering the trace at the spectral peak yielded more unique clusters by preventing the clustering algorithm from labeling similar signals as different classes based only on their relation to the trigger time. For each seismic trace saved, a spectrogram was computed using the short-time Fourier transform with a 0.4 s Kaiser window, NFFT=256, and 90% overlap. Spectrograms (samples) contained one channel of amplitude information, 87 frequency bins, and 100 time bins for a total of 8,700 features per spectrogram. To improve DNN learning, sample-wise normalization was performed by dividing each spectrogram by its vector norm (LeCun et al., 2012).

4 Deep Clustering Implementation

The objective of deep clustering models is to first encode the input data—in this case, spectrograms of seismic signals—into a layer containing latent (lower-dimensional) features, called the *embedded* layer, and to then apply a clustering algorithm in this latent feature space. In the implementation that follows, the 8,700 features of an input spectrogram are reduced to a latent feature space of just 9 embedded features with the use of a convolutional autoencoder, a type of DNN composed of convolutional and transposed convolutional layers. We then describe the GMM and DEC clustering algorithms that are used in the clustering analysis.

4.1 Dimensionality Reduction with a Convolutional Autoencoder

Autoencoders provide a useful means of data approximation using a lower-dimensional representation via a sequence of non-linear transformations. The autoencoder model consists of three components: an *encoder*, a *bottleneck*, and a *decoder* (Murphy, 2012). First, the encoder maps input data from a data space X into a latent feature space Z, which is contained within the bottleneck of the model. Next, the decoder attempts to reconstruct X from Z. This process is performed iteratively with the objective of minimizing the error between X and the decoder output, X'. In minimizing the error, the autoencoder learns the salient features of X and accurately encodes them in Z, thus reducing the dimensionality of the clustering task.

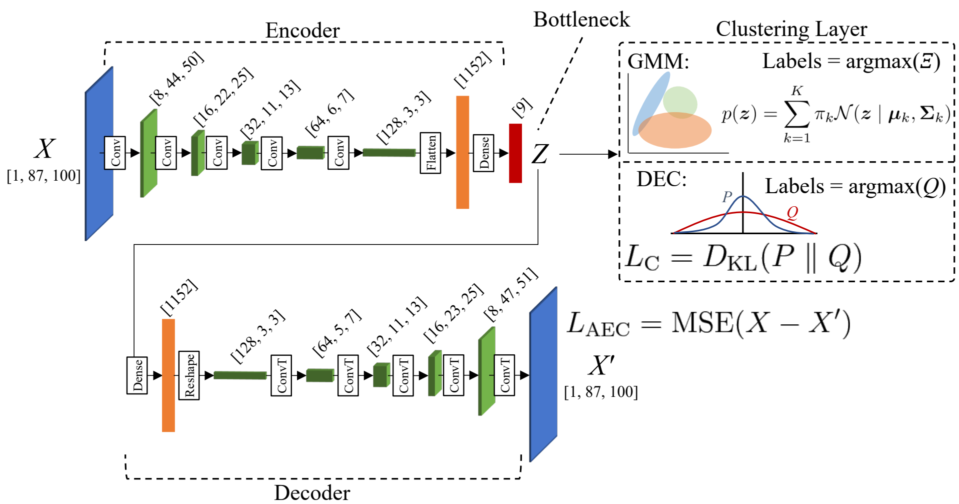

Figure 3. The deep clustering framework in this study uses a convolutional autoencoder that encodes the data space X into the latent feature space Z, and a decoder that recovers the original input X from Z. The mean squared error (MSE) between the input X and the reconstruction X' is used as the autoencoder loss function. The latent feature space Z lies at the bottleneck between the encoder and decoder, providing the input to the clustering layer. Gaussian mixture model (GMM) clustering labels each data sample according to its most likely cluster membership using an expectation-maximization algorithm. Deep embedded clustering (DEC) provides label assignments, and also outputs a clustering loss function that is combined with the MSE to further train the parameters that map $X \to Z \to X'$.

Consider a data set of spectrograms $\mathcal{D} = \{\boldsymbol{x}_n \in X^M\}_{n=1}^N$, where $\boldsymbol{x}_n$ is a vector representation of the n^{th} spectrogram in a data set containing N spectrograms, and the number of features in $\boldsymbol{x}_n$, M, is the spectrogram size (the product of the number of frequency bins and time bins). In the encoder stage, the mapping of X to Z is described by $f_\theta : X \to Z$, where θ are parameters that are learned through iterative model training. The decoder stage is a mirror operation of the encoder and seeks to map the latent feature space Z to the reconstruction X' by $g_\theta : Z \to X'$. The overall mapping of the autoencoder can be described as $F_\theta : X \to Z \to X'$, where $F_\theta = g_\theta \circ f_\theta$. Input spectrograms $\boldsymbol{x}_n$ map to their corresponding latent feature vectors by $\boldsymbol{z}_n = f_\theta(\boldsymbol{x}_n) \in Z^D$, where D is the number of embedded features, and to their reconstructions by $\boldsymbol{x'}_n = F_\theta(\boldsymbol{x}_n) \in X'$.

As the autoencoder is composed of convolutional and transposed convolutional layers, F_θ is a nonlinear mapping that must be appropriately parameterized. This is accomplished by iteratively learning the parameters θ in order to minimize the error between the input and reconstructed data. The mean squared error (MSE) between an input spectrogram with M features and its reconstruction, defined as

$$\ell(\boldsymbol{x}, \boldsymbol{x'}) = \frac{1}{M} \sum_{m=1}^{M} (x_m - x'_m)^2, \tag{1}$$

is averaged over the N samples in the data set to obtain the autoencoder loss function:

$$L_{\text{AEC}} = \frac{1}{N} \sum_{n=1}^{N} \ell(\boldsymbol{x}_n, \boldsymbol{x'}_n). \tag{2}$$

Performing this calculation over the entire data set at once is computationally expensive, memory intensive, and can lead to poor convergence. Instead, the loss is calculated in mini-batch subsets of the data space. For each mini-batch loss, stochastic gradient descent (Goodfellow et al., 2016) is used to update the weights. When all mini-batches have been processed, the next training epoch begins and the process is repeated. After each epoch, a subset of the data separate from the training data is used to validate the model's performance without updating the weights, yielding a validation MSE. Training is performed until a specified maximum number of epochs is reached, or stopped early if the validation MSE fails to decrease below its minimum value after ten epochs. The early stopping criterion prevents the autoencoder from overfitting the training data.

The design choice of autoencoder architecture can be informed by prior knowledge of a data set and its features, as well as practical considerations such as computational resources available. Our DNN architecture, detailed in Table 1, is designed to be computationally efficient, simple to construct, and robust enough to learn salient features

Table 1. *Convolutional Autoencoder Architecture*

Layer Name	Type	Input Shape	Filters	Activation	Output Shape	**Trainable Parameters**
Input	-	-	-	-	[1, 87, 100]	-
Conv1	Convolution	[1, 87, 100]	8	ReLU	[8, 44, 50]	80
Conv2	Convolution	[8, 44, 50]	16	ReLU	[16, 22, 25]	1,168
Conv3	Convolution	[16, 22, 25]	32	ReLU	[32, 11, 13]	4,640
Conv4	Convolution	[32, 11, 13]	64	ReLU	[64, 6, 7]	18,496
Conv5	Convolution	[64, 6, 7]	128	ReLU	[128, 3, 3]	73,856
Flat	Flatten	[128, 3, 3]	-	-	[1152]	0
Encoded	Fully Connected	[1152]	-	ReLU	[9]	10,377
FC	Fully Connected	[9]	-	ReLU	[1152]	11,520
Reshape	Reshape	[1,152]	-	-	[128, 3, 3]	0
ConvT1	Transposed Conv	[128, 3, 3]	64	ReLU	[64, 5, 7]	73,792
ConvT2	Transposed Conv	[64, 5, 7]	32	ReLU	[32, 11, 13]	18,464
ConvT3	Transposed Conv	[32, 11, 13]	16	ReLU	[16, 23, 25]	4,624
ConvT4	Transposed Conv	[16, 23, 25]	8	ReLU	[8, 47, 51]	1,160
Decoded	Transposed Conv	[8, 47, 51]	1	Linear	[1, 95, 101]	73
Output	Crop	[1, 95, 101]	-	-	[1, 87, 100]	-
					Total	**218,250**

Table 2. *Sample Sizes and Hyperparameters used to Train the Autoencoder and Deep Embedded Clustering Model*

Samples			**Hyperparameters**				
Total (N)	Training (N_{train})	Validation (N_{val})	Initial learning rate	Mini-batch size	Classes (K)	Clustering loss factor (λ)	Updates per epoch
531,407	40,000	10,000	10^{-3}	64	8	10^{-4}	10

from a noisy seismic data set. In total, θ contains 218,250 trainable parameters under this DNN architecture.

Autoencoder training is implemented using 50,000 spectrograms randomly selected without replacement from the 531,407 detections. Of the selected spectrograms, 80% are used for training and 20% for validation. The trainable parameters are optimized using the Adaptive Moment Estimation (Adam) algorithm (Kingma & Ba, 2017). In training, there are two principal hyperparameters to address. First is the initial learning rate, which controls the initial step size used by Adam to step down the gradient of the loss. The second hyperparameter is the mini-batch size, which sets the number of spectrograms to be passed through the model at one time. The optimal configuration is found through a grid search of the hyperparameters. A summary of the optimal hyperparameters and the number of spectrograms used are listed in Table 2. As seen in Figure 4a, training and validation losses fall off exponentially with each training epoch until the early stopping criterion is met; in this case, at 48 epochs. The effectiveness of the autoencoder's ability to reconstruct the input spectrogram is illustrated in Figure 5. Though some loss of resolution in time and frequency is expected due to the convolutional and transposed convolutional layers, the structure of the spectrogram is largely preserved, with the salient information of the input encoded to the latent feature space. To test that the autoencoder adequately generalized the entire data set, all spectrograms were fed through the model, yielding an average MSE of 5.9381×10^{-6}, which is consistent with the validation MSE at the early stopping point.

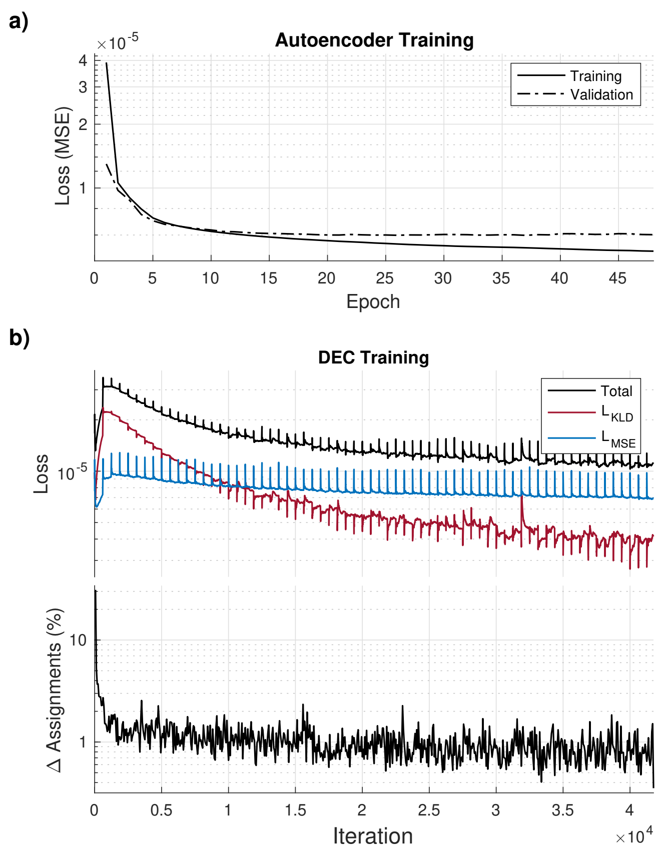

Figure 4. (a) Training and validation losses during autoencoder training. To avoid overfitting the model, training is stopped when the early stopping criterion is met (in this case, at 48 epochs). (b) In the upper plot, loss curves are shown for deep embedded clustering (DEC). In the lower plot, the percentage of samples which undergo class reassignment at each update interval is shown; training is stopped once the change is less than 0.4%

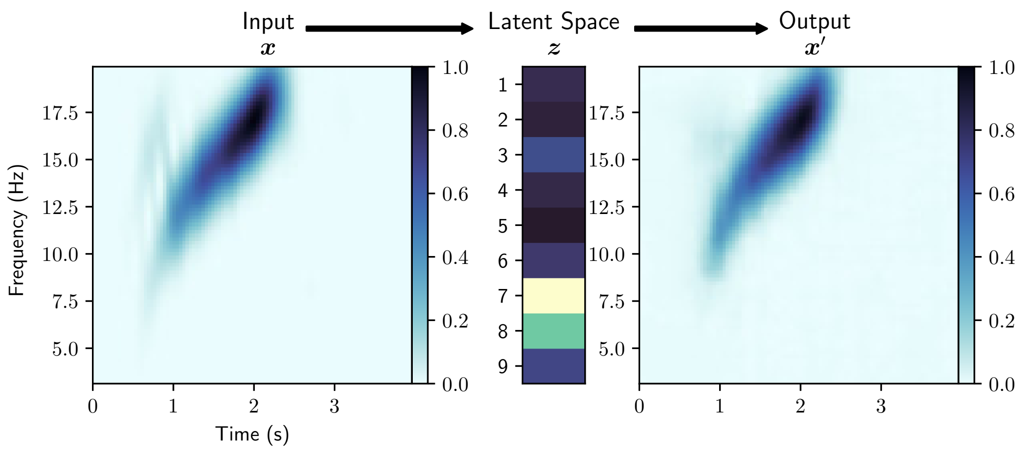

Figure 5. A trained autoencoder takes an input spectrogram $\boldsymbol{x}$, encodes it to a 9-dimensional latent feature vector $\boldsymbol{z}$, then reconstructs the input as $\boldsymbol{x}'$. The autoencoder preserves features correlated within a given cluster and discards the remaining signal, which can help with signal identification.

4.2 Clustering Methodologies

In our deep clustering framework, clustering is performed in the latent feature space, Z, to find K distinct classes of signals within the data. We assume that the data form clusters which are separable in Z space, and that these clusters coalesce around unique locations $\{\boldsymbol{\mu}_k \in Z\}_{k=1}^{K}$, i.e., centroids around which other similar signals may be found. We use Euclidean distance between a centroid and a latent feature vector to measure similarity:

$$d_{n,k} = \|\boldsymbol{z}_n - \boldsymbol{\mu}_k\|_2. \tag{3}$$

$d_{n,k}$ is a measure of the similarity between features indexed by n and k.

4.2.1 Gaussian Mixture Model (GMM)

In GMM clustering, the latent feature vectors $\boldsymbol{z}$ are described by a mixture of K Gaussian distributions that are linearly superimposed in the latent space Z, where each Gaussian model has its own centroid $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$. We follow the methods of Bishop (2006, p. 430) and Murphy (2012, p. 339). The overall distribution of the mixture model is given by the convex combination of their distributions,

$$p(\boldsymbol{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{4}$$

Consider the latent feature vectors $\boldsymbol{z}_n$ as rows of a matrix $\mathbf{Z} \in \mathbb{R}^{N \times D}$ with N samples and D features. To estimate the parameters of each Gaussian distribution, an expectation-maximization (EM) algorithm is used to maximize the Gaussian mixture model's likelihood function of $\mathbf{Z}$ with respect to the parameters $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and π_k (Bishop, 2006, p. 433):

$$\ln p(\mathbf{Z} \mid \{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K\}, \{\boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K\}, \{\pi_1, ..., \pi_K\}) = \sum_{n=1}^{N} \ln \left[\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{z}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right]. \tag{5}$$

For every sample $\boldsymbol{z}_n$, a binary K-dimensional random variable $\xi_k \in \{0, 1\}$ is introduced that has one element equal to one and all others to zero. The marginal distribution over $\boldsymbol{\xi}$ is $p(\xi_k = 1) = \pi_k$, where the mixing coefficients π_k satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$ in order to be valid probabilities. Since $\boldsymbol{\xi}$ is a 1-of-K (categorical) representation, this distribution is written as

$$p(\boldsymbol{\xi}) = \prod_{k=1}^{K} \pi_k^{\xi_k}, \tag{6}$$

and the conditional distribution of $\boldsymbol{z}_n$ given $\boldsymbol{\xi}$ as

$$p(\boldsymbol{z}_n \mid \boldsymbol{\xi}) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{z}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{\xi_k}. \tag{7}$$

Equation (4) is then rewritten in terms of the factored joint distribution $p(\boldsymbol{z}_n, \boldsymbol{\xi}) = p(\boldsymbol{\xi})p(\boldsymbol{z}_n \mid \boldsymbol{\xi})$:

$$p(\boldsymbol{z}_n) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{z}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_{\boldsymbol{\xi}} p(\boldsymbol{\xi}) p(\boldsymbol{z}_n \mid \boldsymbol{\xi}). \tag{8}$$

Using Bayes' theorem and equations (4) and (8), the conditional probability of $\boldsymbol{\xi}$ given $\boldsymbol{z}_n$ is:

$$\gamma(\xi_k) \equiv p(\xi_k = 1 \mid \boldsymbol{z}_n) = \frac{p(\xi_k = 1) p(\boldsymbol{z}_n \mid \xi_k = 1)}{\sum_{j=1}^{K} p(\xi_j = 1) p(\boldsymbol{z}_n \mid \xi_j = 1)} = \frac{\pi_k \mathcal{N}(\boldsymbol{z}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{z}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}, \tag{9}$$

where π_k is the prior probability of $\xi_k = 1$, and $\gamma(\xi_k)$ is the posterior probability having observed $\boldsymbol{z}_n$. As with $\mathbf{Z}$, we construct a matrix $\boldsymbol{\Xi} \in \mathbb{R}^{N \times K}$ whose rows consist of the binary random variables $\boldsymbol{\xi}_n$ for each sample $\boldsymbol{z}_n$. Thus indexed, $\gamma(\xi_{nk})$ is defined as the *responsibility* that distribution k has for *explaining* sample $\boldsymbol{z}_n$, and is analogous to soft clustering, where the probability that sample $\boldsymbol{z}_n$ belongs to distribution k is determined for each of the K distributions. In practice, each latent feature vector $\boldsymbol{z}_n$ is assigned to one of K Gaussian distributions by $\arg\max_{\xi} [\gamma(\xi_{nk})]$.

Using superscript t to denote the iteration index, the EM algorithm for a Gaussian mixture is:

1. Initialization of parameters $\boldsymbol{\mu}_k^{t-1}$, $\boldsymbol{\Sigma}_k^{t-1}$, and π_k^{t-1}.

2. Expectation step. This step encodes the samples' probability of assignment to each Gaussian distribution by evaluating responsibilities $\gamma(\xi_{nk})$ using $\boldsymbol{\mu}_k^{t-1}$, $\boldsymbol{\Sigma}_k^{t-1}$, and π_k^{t-1} (equation (9)).

3. Maximization step. Using the responsibilities $\gamma(\xi_{nk})$, this step updates the centroid location ($\boldsymbol{\mu}_k^t$), shape ($\boldsymbol{\Sigma}_k^t$), and normalization (π_k^t) of each distribution in the latent space Z by:

$$\begin{aligned}
\boldsymbol{\mu}_k^t &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma(\xi_{nk})\boldsymbol{z}_n \\
\boldsymbol{\Sigma}_k^t &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma(\xi_{nk})(\boldsymbol{z}_n-\boldsymbol{\mu}_k^t)(\boldsymbol{z}_n-\boldsymbol{\mu}_k^t)^{\mathrm{T}} \\
\pi_k^t &= \frac{N_k}{N}
\end{aligned} \tag{10}$$

where

$$N_k = \sum_{n=1}^{N}\gamma(\xi_{nk}).$$

4. Convergence check. The log likelihood of $\mathbf{Z}$ is evaluated with respect to the parameters $\boldsymbol{\mu}_k^t$, $\boldsymbol{\Sigma}_k^t$, and π_k^t (equation 5). If convergence occurs in the log likelihood or in the parameters $\boldsymbol{\mu}_k^t$, $\boldsymbol{\Sigma}_k^t$, and π_k^t, the EM algorithm has reached a local maximum and terminates; otherwise, the algorithm returns to step 2.

To accelerate EM convergence, k-means clustering is used to initialize the GMM clustering algorithm (Bishop, 2006, p. 438). EM stops after 1,000 iterations have elapsed or when the change in log likelihood from equation (5) is less than 0.001. To avoid converging on local maxima, the initialization is run 100 times and the initialization with the best log likelihood is retained.

4.2.2 Deep Embedded Clustering (DEC)

In DEC, clustering is performed in conjunction with continued training of the autoencoder, with the clustering layer attached to the bottleneck providing an additional loss function that is backpropagated through the autoencoder layers (Figure 3). The DEC model DNN parameters are initialized using the parameters of the trained autoencoder, and clustering layer parameters are initialized using the centroids from GMM clustering. DEC seeks to improve the GMM clustering by using the Euclidean distance between embedded spectrograms and cluster centroids (equation (3)) as an additional loss function for updating model parameters. Because the input data is unlabeled, a self-supervised method is required. We implement the method developed by Xie et al. (2016), who, drawing from the t-distributed stochastic neighbor embedding (t-SNE) algorithm (van der Maaten & Hinton, 2008), propose measuring the difference between a Student's t-distribution kernel of the latent feature vectors $\boldsymbol{z}$ and an auxiliary target distribution. A simplified Student's t-distribution is used to measure the similarity between embedded spectrograms $\boldsymbol{z}_n$ and the cluster centroids $\boldsymbol{\mu}_k$:

$$q_{nk} = \frac{(1+ \parallel \boldsymbol{z}_n - \boldsymbol{\mu}_k \parallel^2)^{-1}}{\sum_k (1+ \parallel \boldsymbol{z}_n - \boldsymbol{\mu}_k \parallel^2)^{-1}}. \tag{11}$$

Equation (11) results in a set of soft class assignments, i.e., the probability that embedded spectrogram n will be assigned to class k. Latent feature vectors $\boldsymbol{z}_n$ are assigned to one of K classes by $\arg\max_q [q_{nk}]$. The soft class assignments q_{nk} are then used to compute the auxiliary target distribution, p, whose form is designed to improve clustering performance, emphasize embeddings with high-confidence assignments, and normalize each cluster centroid's contribution to the loss function so that large clusters minimally distort Z (Xie et al., 2016):

$$p_{nk} = \frac{q_{nk}^2 / \sum_n q_{nk}}{\sum_k (q_{nk}^2 / \sum_n q_{nk})}. \tag{12}$$

The dissimilarity between the distributions given by equations (11) and (12) is measured using the Kullback-Leibler divergence (Kullback & Leibler, 1951). From the divergence the clustering layer's loss function is obtained:

$$L_{\mathrm{C}} = D_{\mathrm{KL}}(P \parallel Q) = \sum_n \sum_k p_{nk} \log \frac{p_{nk}}{q_{nk}}. \tag{13}$$

In DEC, the clustering layer is attached to the trained autoencoder's bottleneck. During training of the DEC model, the loss functions from equations (2) and (13) are combined into a total loss function,

$$L = L_{\mathrm{AEC}} + \lambda L_{\mathrm{C}}, \tag{14}$$

where λ is a hyperparameter that balances the contributions of the two losses, since they are of differing magnitudes. λ must be tuned: if it is too large, the clustering loss will cause model instability and lead to distortion of the latent space, in which case the latent space will no longer represent the salient features of the data. If λ is too small, the effect on clustering performance will be minimal. We found that $\lambda = 10^{-4}$ yielded optimal performance for model training and clustering.

Two constituent processes occur simultaneously during DEC model training. First, the full loss from equation (14) is backpropagated through the DEC model parameters, which include the autoencoder as well as the cluster centroids. Second, to account for the cluster centroids changing as training progresses, the distributions q_{nk} and p_{nk} are updated at intervals. The update interval is a hyperparameter that must be tuned. Through hyperparameter tuning, an update interval of 10 per training epoch was found to be optimal for clustering performance, minimizing DEC loss, and training within a reasonable time frame. Training is stopped after the number of samples changing assignments after every update interval reaches less than 0.4% of the total number of training samples. The same mini-batch size and initial learning rate are used to train both the autoencoder and DEC model (Table 2). Figure 4b shows how losses decrease over time and the percent change in label assignments for every mini-batch training iteration. Though the overall trends in the loss curves show exponential decay, periodic spikes occur at every update interval, when q_{nk} and p_{nk} are recalculated, and are visible since the losses are recorded after every mini-batch rather than every epoch.

4.3 Selecting Optimal Number of Clusters

Determining the optimal number of clusters, K, is a major challenge in unsupervised machine learning. In this study we treat K as a hyperparameter, iterating the deep clustering workflow over a range of values for K and evaluating the results to choose the best value. Results are evaluated both quantitatively and qualitatively. Quantitative evaluation is performed for each class by examining cumulative distribution functions and probability density functions as functions of distance to each class centroid, $d_{n,k}$ (equation (3)). Additionally, traditional statistical methods for choosing the optimal number of clusters, such as the gap statistic (Tibshirani et al., 2001) and silhouette score (Rousseeuw, 1987), are consulted. The qualitative approach is to visually inspect the similarity of the latent feature vectors $\boldsymbol{z}_n$ to their respective class centroids $\boldsymbol{\mu}_k$, and to see if the spectrograms and seismograms assigned to each class likewise exhibit similarity. In general, the formation of two or more similar classes may indicate that too many classes were initialized, and the data in those classes can be grouped into a single class in post-processing. Too much variance among the spectrograms within a class may indicate the need for one or more additional classes. We found that $K = 8$ was the optimal number of classes for the RIS data set.

5 Results

The following analysis of GMM and DEC performance focuses on how the clustering algorithms affect the latent space Z and whether the methods yield meaningful results in the data space X. Since the samples in the data set are unlabeled and there is no "ground truth" against which to compare results, measurements of intra-class similarity among spectrograms and latent feature vectors are examined. We conclude that neither GMM nor DEC provides a clear advantage in clustering performance. Accordingly, we recommend implementation of GMM for deep clustering of RIS seismic data. The statistical and mathematical underpinnings of GMM are well understood, and the complexity of implementation and interpretation of DEC is difficult to justify in the absence of compelling performance improvement. Furthermore, in practice GMM clustering on a graphics processing unit takes approximately one minute to cluster the entire data set, whereas one DEC hyperparameter tuning run can take several hours.

In the analyses that follow, results are presented for the entire data set of 531,407 spectrograms, including the training and validation data subsets. We mitigate the risk of the DNN in the DEC model overfitting on the training data (Murphy, 2012, p. 23) by using less than 10% of the data set for training and validation, and by drawing training samples randomly without replacement to achieve a training subset representative of the entire data set.

5.1 Clustering Performance

Deep clustering performance is qualitatively checked by comparing centroids to their respective assigned latent data samples. Results for GMM are shown in Figure 6. Each class k is represented by the columns in Figure 6, with each centroid $\boldsymbol{\mu}_k$ and its reconstruction $g_\theta(\boldsymbol{\mu}_k)$ plotted along the top row. Although the centroid is not a member of the data set, because the centroid represents the salient features of its class, its reconstruction is expected to resemble the spectrograms $\boldsymbol{x}_n$ assigned to its class. Subsequent rows show the latent feature vectors $\boldsymbol{z}_n$, spectrograms $\boldsymbol{x}_n$, and associated seismograms of the data samples assigned to the respective classes. To inspect whether intra-class similarity holds with increasing distance from the centroid, samples $\boldsymbol{z}_n$ and $\boldsymbol{x}_n$ are shown for $n = \{1, 1000, 5000, 10000, 15000, 20000, 25000\}$. Near the centroid, latent feature vectors $\boldsymbol{z}_n$ generally exhibit similar values to their class centroid $\boldsymbol{\mu}_k$, indicating that GMM has successfully grouped similar latent data samples into the class, and that the centroid is representative of the data in its class. The spectrograms in each class are likewise similar to each other and to the centroid reconstruction $g_\theta(\boldsymbol{\mu}_k)$, confirming that the latent features embedded in the centroids are representative of the spectrograms in the class. Finally, the similarity in the latent space and time-frequency domain extends to the time domain, where seismograms in each class are similar to one another. As distance increases (i.e., with increasing n), cases of dissimilarity begin to arise as samples overlap with adjacent clusters.

In addition to checking the efficacy of the clustering, visual examination of the results in Figure 6 gives indication of whether or not an appropriate number of clusters was chosen. For example, classes 4 and 8 exhibit similar characteristics in time and frequency, distinct from each other primarily in peak amplitude characteristics. If such distinctions are not useful or if similarities are redundant, classes can be combined in post-processing. If too few clusters are selected, classes may contain widely differing signals, indicating the need to increase the number of clusters.

Clustering with DEC involves two steps: first, the GMM clustering algorithm initializes the centroids, but the latent data are left unmodified. Second, during DEC, centroids are further refined while the latent data are moved much closer to their respective centroids, with some data reassigned to different classes altogether. To determine to what extent this occurs, t-SNE is used to visualize the 9-dimensional latent space in two dimensions (van der Maaten & Hinton, 2008). t-SNE can illuminate possible clusters within data in an unsupervised manner by displaying data in geometrically separated clusters. In Figure 7a, t-SNE results of the latent feature space clustered with GMM show that the data are largely contiguous with few exceptions. Applying the labels assigned by GMM clustering to the data points shows that, while there is some geometric separation between the clusters, the embedding is characterized by overlapping and dispersed class members, indicating poor separation in the latent space. Contrast this with Figure 7b, in which t-SNE results at the conclusion of DEC show both geometric separation as well as nearly homogeneous class assignments.

While t-SNE offers an intuitively visual way to look for clusters in data, results are sometimes difficult to interpret and are impossible to reproduce exactly due to the inherent randomness of the algorithm. Running t-SNE iteratively and with the same ran-

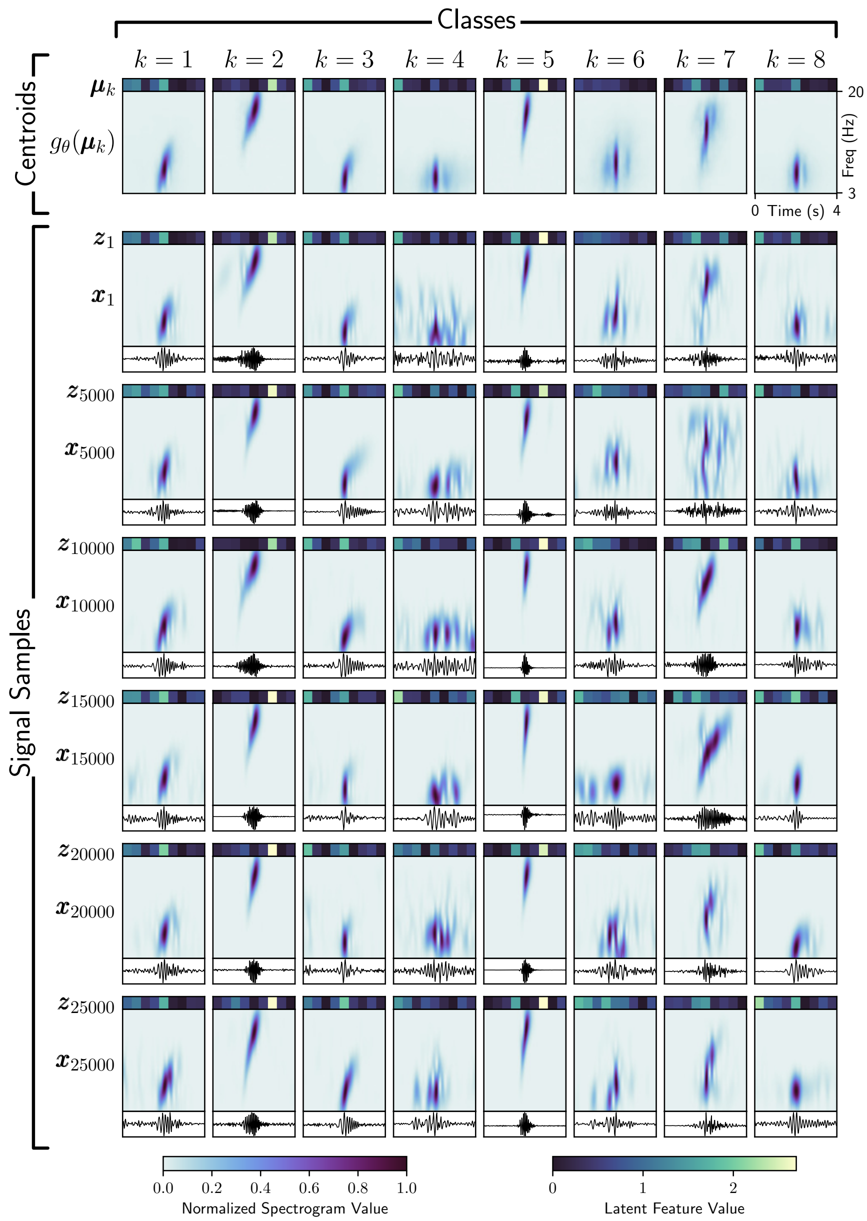

Figure 6. Gaussian mixture model (GMM) clustering results are shown, with samples $\boldsymbol{z}_n$ and $\boldsymbol{x}_n$ the n^{th} closest to their respective centroids. Within a given class k, the cluster centroids $\boldsymbol{\mu}_k$ are similar to the latent feature vectors $\boldsymbol{z}_n$, whose nine elements are shown above each spectrogram. Though the centroids are not members of the data set, their reconstructions $g_\theta(\boldsymbol{\mu}_k)$ exhibit similar characteristics to the spectrograms $\boldsymbol{x}_n$ assigned to each class. Seismograms plotted below each spectrogram also exhibit similarity within each class. With increasing distance from the centroid (i.e., as n increases), dissimilarity and potential cases of mis-assignment are visible in latent feature vectors, spectrograms, and seismograms, e.g for $k = 7$, $n = 15000$.

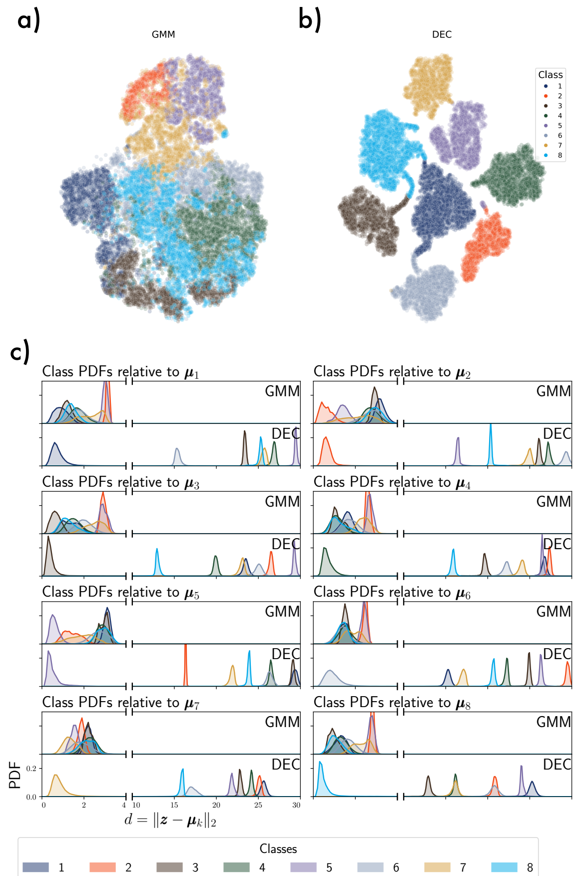

Figure 7. (a) Visualization of the 9-dimensional latent data space is shown in two dimensions using the t-distributed stochastic neighbor embedding (t-SNE) plot for Gaussian mixture model (GMM) clustering. GMM exhibits limited separation within the data and overlapping classes. (b) t-SNE plot for deep embedded clustering (DEC), whose clusters are well separated and contain nearly homogeneous class members. (c) The effects of DEC in the latent feature space are evident for each class probability density function (PDF) with respect to the distance from the centroids. In addition to moving the assigned class members closer to the centroid, DEC increases the distance between the other class centroids and PDFs.

dom seed can mitigate these limitations, but examination of the effects of deep clustering on the densities of the clusters provides a more concrete visualization. Of interest to the ability for the clustering algorithms to identify clusters is the distance of each cluster to the others. In Figure 7c, the probability density functions (PDF) of all clusters are shown as functions of distance to each centroid. Before DEC, though GMM clustering usually results in the PDF of each class being closest to its centroid, there is significant overlap with other clusters, and the clusters themselves are not particularly dense. With DEC, the PDF of each class is closer to its centroid, denser, and farther removed from the other clusters. Thus, DEC effectively separates each cluster from the others, allowing for better distinction between clusters in the latent space.

The effects of DEC become readily apparent when the latent feature vectors are stacked and sorted according to their distance from each centroid, as shown in Figure 8. By sorting the latent space by sample index n such that $d_{n+1,k} > d_{n,k}$, cluster separation can be visualized directly in the latent space. Before DEC, centroids are initialized with the GMM clustering algorithm without modification to the latent data. Closest to each class centroid, the latent feature vectors are similar in appearance to the centroid, but transition continuously to different patterns as the sorted index n increases. The contrast with the latent feature space after DEC is stark: because DEC moves latent data assigned to a particular class closer to the centroid, the effect is that the latent feature vectors take on similar values, and therefore appearance, to the centroid. The result is that the latent space appears more sharply segmented after DEC, with the samples closest to the centroid of nearly uniform appearance to the centroid itself. For reference, the relative location of the other class centroids are marked with white vertical lines. With GMM, the latent feature vectors belonging to the other classes are not readily apparent, whereas after DEC, most of the other centroid locations are associated with their distinctive latent feature vectors.

While DEC effectively transforms the latent feature space Z by moving latent feature vectors closer to their centroids, less clear is whether this transformation causes a corresponding improvement in clustering quality in the data space X. To evaluate intra-class similarity among spectrograms, four pairwise metrics are used to compare the clustering assignments obtained from GMM and DEC.

Figure 8. For each class k, latent data samples z_n are shown stacked according to their distance kz_n $_k$k from the centroid $_k$ (shown to the left). Distance of the other cluster centroids relative to the selected class k are indicated with vertical dotted lines. Deep embedded clustering (DEC) brings assigned data z_n closer to the class centroid, resulting in homogeneity among the latent feature vectors assigned to that class.

Figure 12. Two years of (a) temperature and (b) wind speed observations at Margaret automated weather station (MGT, approximately 122 km southwest of RS09, Figure 1), c) model-derived tides calculated at station RS10, and (d-k) icequake detection statistics for each signal class. Interannual timescale is shown at left with vertical red lines indicating the subset weekly time-scale at right. The diurnal tidal signal correlates with seismicity for classes 2, 3, and 6. Tidal model from (Padman et al., 2002); weather station data from AMRC, SSEC, UW{Madison.

References

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the Surprising Behavior of Distance Metrics in High Dimensional Space. In G. Goos, J. Hartmanis, J. van Leeuwen, J. Van den Bussche, & V. Vianu (Eds.), *Database Theory — ICDT 2001* (Vol. 1973, pp. 420–434). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/3-540-44503-X_27

Aggarwal, C. C., & Reddy, C. K. (Eds.). (2014). *Data clustering: Algorithms and applications*. Boca Raton: Chapman and Hall/CRC.

Allen, R. (1982, December). Automatic phase pickers: Their present use and future prospects. *Bulletin of the Seismological Society of America*, *72*(6B), S225-S242. doi: 10.1785/BSSA07206B0225

Aster, R. C., & Winberry, J. P. (2017, December). Glacial seismology. *Reports on Progress in Physics*, *80*(12), 126801. doi: 10.1088/1361-6633/aa8473

Baker, M. G., Aster, R. C., Anthony, R. E., Chaput, J., Wiens, D. A., Nyblade, A., ... Stephen, R. A. (2019, December). Seasonal and spatial variations in the ocean-coupled ambient wavefield of the Ross Ice Shelf. *Journal of Glaciology*, *65*(254), 912–925. doi: 10.1017/jog.2019.64

Baker, M. G., Aster, R. C., Wiens, D. A., Nyblade, A., Bromirski, P. D., Gerstoft, P., & Stephen, R. A. (2020, October). Teleseismic earthquake wavefields observed on the Ross Ice Shelf. *Journal of Glaciology*, 1–17. doi: 10.1017/jog.2020.83

Barcheck, C. G., Tulaczyk, S., Schwartz, S. Y., Walter, J. I., & Winberry, J. P. (2018, March). Implications of basal micro-earthquakes and tremor for ice stream mechanics: Stick-slip basal sliding and till erosion. *Earth and Planetary Science Letters*, *486*, 54–60. doi: 10.1016/j.epsl.2017.12.046

Beaucé, E., Frank, W. B., & Romanenko, A. (2018, January). Fast Matched Filter (FMF): An Efficient Seismic Matched-Filter Search for Both CPU and GPU Architectures. *Seismological Research Letters*, *89*(1), 165–172. doi: 10.1785/0220170181

Bellman, R. E. (1961). *Adaptive Control Processes: A Guided Tour*. Rand Corporation.

Bergen, K. J., & Beroza, G. C. (2018, June). Detecting earthquakes over a seismic network using single-station similarity measures. *Geophysical Journal Interna-*

tional, *213*(3), 1984–1998. doi: 10.1093/gji/ggy100

Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y., & Wassermann, J. (2010, May). ObsPy: A Python Toolbox for Seismology. *Seismological Research Letters*, *81*(3), 530–533. doi: 10.1785/gssrl.81.3.530

Bianco, M. J., & Gerstoft, P. (2018, December). Travel Time Tomography With Adaptive Dictionaries. *IEEE Transactions on Computational Imaging*, *4*(4), 499–511. doi: 10.1109/TCI.2018.2862644

Bianco, M. J., Gerstoft, P., Olsen, K. B., & Lin, F.-C. (2019, December). High-resolution seismic tomography of Long Beach, CA using machine learning. *Scientific Reports*, *9*(1), 14987. doi: 10.1038/s41598-019-50381-z

Bindschadler, R. A., King, M. A., Alley, R. B., Anandakrishnan, S., & Padman, L. (2003, August). Tidally Controlled Stick-Slip Discharge of a West Antarctic Ice Stream. *Science*, *301*(5636), 1087–1089. doi: 10.1126/science.1087231

Bindschadler, R. A., Vornberger, P. L., King, M. A., & Padman, L. (2003). Tidally driven stick–slip motion in the mouth of Whillans Ice Stream, Antarctica. *Annals of Glaciology*, *36*, 263–272. doi: 10.3189/172756403781816284

Bishop, C. (2006). *Pattern Recognition and Machine Learning* (First ed.). Springer-Verlag New York.

Boubekki, A., Kampffmeyer, M., Brefeld, U., & Jenssen, R. (2021, July). Joint optimization of an autoencoder for clustering and embedding. *Machine Learning*, *110*(7), 1901–1937. doi: 10.1007/s10994-021-06015-5

Bromirski, P. D., Chen, Z., Stephen, R. A., Gerstoft, P., Arcas, D., Diez, A., ... Nyblade, A. (2017, July). Tsunami and infragravity waves impacting A ntarctic ice shelves. *Journal of Geophysical Research: Oceans*, *122*(7), 5786–5801. doi: 10.1002/2017JC012913

Bromirski, P. D., Diez, A., Gerstoft, P., Stephen, R. A., Bolmer, T., Wiens, D. A., ... Nyblade, A. (2015, September). Ross ice shelf vibrations. *Geophysical Research Letters*, *42*(18), 7589–7597. doi: 10.1002/2015GL065284

Bromirski, P. D., & Stephen, R. A. (2012). Response of the Ross Ice Shelf, Antarctica, to ocean gravity-wave forcing. *Annals of Glaciology*, *53*(60), 163–172. doi: 10.3189/2012AoG60A058

Cavalieri, D. J., Parkinson, C. L., Gloersen, P., & Zwally, H. J. (1996, updated yearly). *Sea Ice Concentrations from Nimbus-7 SMMR and DMSP SSM/I-*

SSMIS Passive Microwave Data, Version 1. Boulder, Colorado USA. NASA National Snow and Ice Data Center Distributed Active Archive Center.

Chamarczuk, M., Nishitsuji, Y., Malinowski, M., & Draganov, D. (2020, January). Unsupervised Learning Used in Automatic Detection and Classification of Ambient-Noise Recordings from a Large-N Array. *Seismological Research Letters*, *91*(1), 370–389. doi: 10.1785/0220190063

Chamberlain, C. J., Hopp, C. J., Boese, C. M., Warren-Smith, E., Chambers, D., Chu, S. X., ... Townend, J. (2018, January). EQcorrscan: Repeating and Near-Repeating Earthquake Detection and Analysis in Python. *Seismological Research Letters*, *89*(1), 173–181. doi: 10.1785/0220170151

Chaput, J., Aster, R. C., McGrath, D., Baker, M., Anthony, R. E., Gerstoft, P., ... Stevens, L. A. (2018, October). Near-Surface Environmentally Forced Changes in the Ross Ice Shelf Observed With Ambient Seismic Noise. *Geophysical Research Letters*, *45*(20). doi: 10.1029/2018GL079665

Chazan, S. E., Gannot, S., & Goldberger, J. (2019, March). Deep Clustering Based on a Mixture of Autoencoders. *arXiv:1812.06535 [cs, stat]*.

Chen, Z., Bromirski, P. D., Gerstoft, P., Stephen, R. A., Lee, W. S., Yun, S., ... Nyblade, A. A. (2019, August). Ross Ice Shelf Icequakes Associated With Ocean Gravity Wave Activity. *Geophysical Research Letters*, *46*(15), 8893–8902. doi: 10.1029/2019GL084123

Chen, Z., Bromirski, P. D., Gerstoft, P., Stephen, R. A., Wiens, D. A., Aster, R. C., & Nyblade, A. A. (2018, October). Ocean-excited plate waves in the Ross and Pine Island Glacier ice shelves. *Journal of Glaciology*, *64*(247), 730–744. doi: 10.1017/jog.2018.66

Cole, H. M. (2020). *Tidally Induced Seismicity at the Grounded Margins of the Ross Ice Shelf, Antarctica* (Master's Thesis). Colorado State University, Fort Collins, Colorado.

De Angelis, H., & Skvarca, P. (2003, March). Glacier Surge After Ice Shelf Collapse. *Science*, *299*(5612), 1560–1562. doi: 10.1126/science.1077987

Diez, A., Bromirski, P., Gerstoft, P., Stephen, R., Anthony, R., Aster, R., ... Wiens, D. (2016, May). Ice shelf structure derived from dispersion curve analysis of ambient seismic noise, Ross Ice Shelf, Antarctica. *Geophysical Journal International*, *205*(2), 785–795. doi: 10.1093/gji/ggw036

Dupont, T. K., & Alley, R. B. (2005). Assessment of the importance of ice-shelf buttressing to ice-sheet flow. *Geophysical Research Letters*, *32*(4). doi: 10.1029/2004GL022024

Fürst, J. J., Durand, G., Gillet-Chaulet, F., Tavard, L., Rankl, M., Braun, M., & Gagliardini, O. (2016, May). The safety band of Antarctic ice shelves. *Nature Climate Change*, *6*(5), 479–482. doi: 10.1038/nclimate2912

Gibbons, S. J., & Ringdal, F. (2006, April). The detection of low magnitude seismic events using array-based waveform correlation. *Geophysical Journal International*, *165*(1), 149–166. doi: 10.1111/j.1365-246X.2006.02865.x

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Greene, C. A., Gwyther, D. E., & Blankenship, D. D. (2017, July). Antarctic Mapping Tools for Matlab. *Computers & Geosciences*, *104*, 151–157. doi: 10.1016/j.cageo.2016.08.003

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. *Applied Statistics*, *28*(1), 100. doi: 10.2307/2346830

Hell, M. C., Cornelle, B. D., Gille, S. T., Miller, A. J., & Bromirski, P. D. (2019, November). Identifying Ocean Swell Generation Events from Ross Ice Shelf Seismic Data. *Journal of Atmospheric and Oceanic Technology*, *36*(11), 2171–2189. doi: 10.1175/JTECH-D-19-0093.1

Hinton, G. E. (2006, July). Reducing the Dimensionality of Data with Neural Networks. *Science*, *313*(5786), 504–507. doi: 10.1126/science.1127647

Holtzman, B. K., Paté, A., Paisley, J., Waldhauser, F., & Repetto, D. (2018, May). Machine learning reveals cyclic changes in seismic source spectra in Geysers geothermal field. *Science Advances*, *4*(5), eaao2929. doi: 10.1126/sciadv.aao2929

Hotovec-Ellis, A. J., & Jeffries, C. (2016, April). *Near Real-time Detection, Clustering, and Analysis of Repeating Earthquakes: Application to Mount St. Helens and Redoubt Volcanoes* [Invited]. Reno, NV, USA.

Johnson, C. W., Ben-Zion, Y., Meng, H., & Vernon, F. (2020, August). Identifying Different Classes of Seismic Noise Signals Using Unsupervised Learning. *Geophysical Research Letters*, *47*(15). doi: 10.1029/2020GL088353

Johnson, C. W., Meng, H., Vernon, F., & Ben-Zion, Y. (2019, August). Characteristics of Ground Motion Generated by Wind Interaction With Trees, Structures, and Other Surface Obstacles. *Journal of Geophysical Research: Solid Earth*, *124* (8), 8519–8539. doi: 10.1029/2018JB017151

Kingma, D. P., & Ba, J. (2017, January). Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*.

Klein, E., Mosbeux, C., Bromirski, P. D., Padman, L., Bock, Y., Springer, S. R., & Fricker, H. A. (2020, October). Annual cycle in flow of Ross Ice Shelf, Antarctica: Contribution of variable basal melting. *Journal of Glaciology*, *66* (259), 861–875. doi: 10.1017/jog.2020.61

Kong, Q., Trugman, D. T., Ross, Z. E., Bianco, M. J., Meade, B. J., & Gerstoft, P. (2019, January). Machine Learning in Seismology: Turning Data into Insights. *Seismological Research Letters*, *90* (1), 3–14. doi: 10.1785/0220180259

Kullback, S., & Leibler, R. A. (1951, March). On Information and Sufficiency. *The Annals of Mathematical Statistics*, *22* (1), 79–86. doi: 10.1214/aoms/1177729694

LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K.-R. (2012). Efficient BackProp. In G. Montavon, G. B. Orr, & K.-R. Müller (Eds.), *Neural Networks: Tricks of the Trade: Second Edition* (pp. 9–48). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-35289-8_3

MacAyeal, D. R., Banwell, A. F., Okal, E. A., Lin, J., Willis, I. C., Goodsell, B., & MacDonald, G. J. (2019, September). Diurnal seismicity cycle linked to subsurface melting on an ice shelf. *Annals of Glaciology*, *60* (79), 137–157. doi: 10.1017/aog.2018.29

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth berkeley symposium on mathematical statistics and probability, volume 1: Statistics* (pp. 281–297). Berkeley, Calif.: University of California Press.

Min, E., Guo, X., Liu, Q., Zhang, G., Cui, J., & Long, J. (2018). A Survey of Clustering With Deep Learning: From the Perspective of Network Architecture. *IEEE Access*, *6*, 39501–39514. doi: 10.1109/ACCESS.2018.2855437

Morlighem, M., Williams, C. N., Rignot, E., An, L., Arndt, J. E., Bamber, J. L., ... Zinglersen, K. B. (2017, November). BedMachine v3: Complete Bed Topography and Ocean Bathymetry Mapping of Greenland From Multibeam Echo Sounding Combined With Mass Conservation. *Geophysical Research Letters*, *44*(21). doi: 10.1002/2017GL074954

Mousavi, S. M., Horton, S. P., Langston, C. A., & Samei, B. (2016, October). Seismic features and automatic discrimination of deep and shallow induced-microearthquakes using neural network and logistic regression. *Geophysical Journal International*, *207*(1), 29–46. doi: 10.1093/gji/ggw258

Mousavi, S. M., Zhu, W., Ellsworth, W., & Beroza, G. (2019, November). Unsupervised Clustering of Seismic Signals Using Deep Convolutional Autoencoders. *IEEE Geoscience and Remote Sensing Letters*, *16*(11), 1693–1697. doi: 10.1109/LGRS.2019.2909218

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective.* Cambridge, MA: MIT Press.

Nicolas, J. P., Vogelmann, A. M., Scott, R. C., Wilson, A. B., Cadeddu, M. P., Bromwich, D. H., ... Wille, J. D. (2017, August). January 2016 extensive summer melt in West Antarctica favoured by strong El Niño. *Nature Communications*, *8*(1), 15799. doi: 10.1038/ncomms15799

Olinger, S. D., Lipovsky, B. P., Wiens, D. A., Aster, R. C., Bromirski, P. D., Chen, Z., ... Stephen, R. A. (2019, June). Tidal and Thermal Stresses Drive Seismicity Along a Major Ross Ice Shelf Rift. *Geophysical Research Letters*, *46*(12), 6644–6652. doi: 10.1029/2019GL082842

Ozanich, E., Thode, A., Gerstoft, P., Freeman, L. A., & Freeman, S. (2021). Deep embedded clustering of coral reef bioacoustics. *J. Acoust. Soc. Am.*, 16.

Padman, L., Fricker, H. A., Coleman, R., Howard, S., & Erofeeva, L. (2002). A new tide model for the Antarctic ice shelves and seas. *Annals of Glaciology*, *34*, 247–254. doi: 10.3189/172756402781817752

Paolo, F. S., Fricker, H. A., & Padman, L. (2015). Volume loss from Antarctic ice shelves is accelerating. *Science*, *348*(6232), 327–331. doi: 10.1126/science.aaa0940

Perol, T., Gharbi, M., & Denolle, M. (2018, February). Convolutional neural network for earthquake detection and location. *Science Advances*, *4*(2), e1700578. doi: 10.1126/sciadv.1700578

Pritchard, H. D., Ligtenberg, S. R. M., Fricker, H. A., Vaughan, D. G., van den Broeke, M. R., & Padman, L. (2012, April). Antarctic ice-sheet loss driven by basal melting of ice shelves. *Nature*, *484*(7395), 502–505. doi:

10.1038/nature10968

Reddy, T. A., Devi, K. R., & Gangashetty, S. V. (2012, March). Nonlinear principal component analysis for seismic data compression. In *2012 1st International Conference on Recent Advances in Information Technology (RAIT)* (pp. 927–932). Dhanbad, India: IEEE. doi: 10.1109/RAIT.2012.6194558

Riahi, N., & Gerstoft, P. (2017, March). Using graph clustering to locate sources within a dense sensor array. *Signal Processing*, *132*, 110–120. doi: 10.1016/j.sigpro.2016.10.001

Rignot, E., Mouginot, J., Morlighem, M., Seroussi, H., & Scheuchl, B. (2014, May). Widespread, rapid grounding line retreat of Pine Island, Thwaites, Smith, and Kohler glaciers, West Antarctica, from 1992 to 2011. *Geophysical Research Letters*, *41*(10), 3502–3509. doi: 10.1002/2014GL060140

Rousseeuw, P. J. (1987, November). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. doi: 10.1016/0377-0427(87)90125-7

Scambos, T. A. (2004). Glacier acceleration and thinning after ice shelf collapse in the Larsen B embayment, Antarctica. *Geophysical Research Letters*, *31*(18), L18402. doi: 10.1029/2004GL020670

Seydoux, L., Balestriero, R., Poli, P., de Hoop, M., Campillo, M., & Baraniuk, R. (2020, December). Clustering earthquake signals and background noises in continuous seismic data with unsupervised deep learning. *Nature Communications*, *11*(1), 3972. doi: 10.1038/s41467-020-17841-x

Smith, B., Fricker, H. A., Gardner, A. S., Medley, B., Nilsson, J., Paolo, F. S., ... Zwally, H. J. (2020, June). Pervasive ice sheet mass loss reflects competing ocean and atmosphere processes. *Science*, *368*(6496), 1239–1242. doi: 10.1126/science.aaz5845

Snover, D., Johnson, C. W., Bianco, M. J., & Gerstoft, P. (2021, March). Deep Clustering to Identify Sources of Urban Seismic Noise in Long Beach, California. *Seismological Research Letters*, *92*(2A), 1011–1022. doi: 10.1785/0220200164

Steinbach, M., Ertöz, L., & Kumar, V. (2004). The Challenges of Clustering High Dimensional Data. In L. T. Wille (Ed.), *New Directions in Statistical Physics: Econophysics, Bioinformatics, and Pattern Recognition* (pp. 273–309). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-662-08968-2_16

Telesca, L., & Chelidze, T. (2018, November). Visibility Graph Analysis of Seismicity around Enguri High Arch Dam, Caucasus. *Bulletin of the Seismological Society of America*, *108*(5B), 3141–3147. doi: 10.1785/0120170370

Thoma, M., Jenkins, A., Holland, D., & Jacobs, S. (2008, September). Modelling Circumpolar Deep Water intrusions on the Amundsen Sea continental shelf, Antarctica. *Geophysical Research Letters*, *35*(18), L18602. doi: 10.1029/2008GL034939

Tibshirani, R., Walther, G., & Hastie, T. (2001, May). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(2), 411–423. doi: 10.1111/1467-9868.00293

Trugman, D. T., & Shearer, P. M. (2017, March). GrowClust: A Hierarchical Clustering Algorithm for Relative Earthquake Relocation, with Application to the Spanish Springs and Sheldon, Nevada, Earthquake Sequences. *Seismological Research Letters*, *88*(2A), 379–391. doi: 10.1785/0220160188

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010, December). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research*, 38.

Wallet, B. C., & Hardisty, R. (2019, August). Unsupervised seismic facies using Gaussian mixture models. *Interpretation*, *7*(3), 19.

Wiens, D., & Bromirski, P. (2014). *Collaborative Research: Dynamic Response of the Ross Ice Shelf to Wave-Induced Vibrations, and Collaborative Research: Mantle Structure and Dynamics of the Ross Sea from a Passive Seismic Deployment on the Ross Ice Shelf.* International Federation of Digital Seismograph Networks.

Wiens, D. A., Anandakrishnan, S., Winberry, J. P., & King, M. A. (2008, June). Simultaneous teleseismic and geodetic observations of the stick–slip motion of an Antarctic ice stream. *Nature*, *453*(7196), 770–774. doi: 10.1038/nature06990

Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised Deep Embedding for Clustering Analysis. *Proceedings of the 33rd international conference on machine learning*, 10.

Yang, B., Fu, X., Sidiropoulos, N. D., & Hong, M. (2017, June). Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering. *arXiv:1610.04794 [cs]*.

Yoon, C. E., O'Reilly, O., Bergen, K. J., & Beroza, G. C. (2015, December). Earthquake detection through computationally efficient similarity search. *Science Advances*, *1*(11), e1501057. doi: 10.1126/sciadv.1501057