UNIVERSITY OF CALIFORNIA SAN DIEGO

# Machine learning for localizing and characterizing underwater passive acoustics

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Oceanography

by

Emma Ozanich

Committee in charge:

Peter Gerstoft, Chair
Michael Buckingham
William Hodgkiss
Bhaskar D. Rao
Peter Shearer
Heechun Song
Aaron Thode

2020

The dissertation of Emma Ozanich is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2020

# DEDICATION

To the women in underwater acoustics:
those who led before me
and those yet to follow.

# EPIGRAPH

*Progress is rarely a straight line. There are always bumps in the road, but you can make the choice to keep looking ahead.* —Kara Goucher, Olympic long-distance runner

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGEMENTS

Thank you to my advisor Peter Gerstoft for guiding me to complete the PhD and for sharing your experiences in underwater acoustics. Sincere thanks to Aaron Thode for your insightful feedback and for willingly stepping in as a committee member and coauthor. Thanks to Peter Worcester, Matt Dzieciuch, Lauren Freeman, and Simon Freeman for supporting my research and inviting me to join experiments on multiple occasions.

Thanks to Haiqiang Niu for your hard work with the Noiselab and for inviting me to be a coauthor on many papers. It was an honor to collaborate with you.

Thanks to all my Noiselab labmates for making UCSD fun and memorable. I hope we cross paths again.

Thank you to Kerri Seger for your ongoing mentorship. You're a role model and a friend.

Thanks to my family for encouraging my education and career. Nick, I will always be grateful for your dedication to supporting me throughout graduate school and beyond.

This work was supported by the Office of Naval Research, with Award Numbers N00014-13-1-0632 and N00014-12-1-0226 for the research in Chapter 2 and Award Number N00014-18-1-2065 for Chapters 3-5.

This dissertation is a collection of papers that were published or have been prepared for publication. The text of Chapter Two is in full a reprint of the material as it appears in Emma Ozanich, Peter Gerstoft, Peter F. Worcester, Matthew A. Dzieciuch, and Aaron Thode, "Eastern Arctic ambient noise recorded on a drifting vertical array," *Journal of the Acoustical Society of America*, 142(3):1997–2006, 2017. The dissertation author was the primary researcher and author of Chapter Two. The coauthors listed in this publication directed and supervised the research.

The text of Chapter Three is in part and under some rearrangements a reprint of the material as it appears in Emma Ozanich, Aaron Thode, Peter Gerstoft, Lauren A. Freeman, and Simon Freeman, "Unsupervised clustering of coral reef biacoustics," *Journal of the Acoustical Society of America*, submitted for 2021. The dissertation author was the primary researcher and author in Chapter Three. The coauthors listed in this publication directed and supervised the research.

The text of Chapter Four is in full a reprint of the material as it appears in Haiqiang Niu, Emma Reeves, and Peter Gerstoft, “Source localization in an ocean waveguide using supervised machine learning,” *Journal of the Acoustical Society of America*, 142(3):1176–1188, 2017. The dissertation author was a major research contributor in Chapter Four. The contributions of the dissertation author in Chapter Four include significant content writing and research direction. Coauthors listed in this publications were the primary researcher and supporting researcher that directed and supervised the research.

The text of Chapter Five is in full a reprint of the material as it appears in Emma Ozanich, Peter Gerstoft, and Haiqiang Niu, “Feedforward neural network for direction-of-arrival estimation,” *Journal of the Acoustical Society of America*, 147(3):2035–2048, 2020. The dissertation author was the primary researcher and author in Chapter Five. The coauthors listed in this publication directed and supervised the research.

## VITA

| | |
|---|---|
| 2014 | B. S. in Physics *cum laude*, Hamline University, St. Paul, Minnesota |
| 2017 | M. S. in Oceanography, University of California San Diego |
| 2020 | Ph. D. in Oceanography, University of California San Diego |
| 2013 | REU Intern, Center for Remote Sensing of Ice Sheets,<br>Elizabeth City State University, North Carolina |
| 2011-2014 | Research Assistant to Prof. Kevin Stanley and Prof. JiaJia Dong,<br>Hamline University |
| 2020 | Teaching Assistant to Prof. Peter Gerstoft,<br>ECE 228, Machine Learning for Physical Applications |
| 2014-2020 | Research Assistant to Prof. Peter Gerstoft, Noiselab,<br>University of California, San Diego |

## PUBLICATIONS

Niu, H., Reeves, E., and Gerstoft, P. "Source localization in an ocean waveguide using supervised machine learning", *J. Acoust. Soc. Am.*, 142, 2017.

Ozanich, E. Gerstoft, P., Worcester, P. F., Dzieciuch, M. A., and Thode, A. "Eastern Arctic ambient noise on a drifting vertical array", *J. Acoust. Soc. Am.*, 142, 2017.

Niu, H., Ozanich, E., and Gerstoft, P. "Ship localization in Santa Barbara Channel using machine learning classifiers", *J. Acoust. Soc. Am.*, 142, 2017.

Niu, H., Gong, Z., Ozanich, E., Gerstoft, P., Wang, H., and Li, Z. "Deep-learning source localization using multi-frequency magnitude-only data", *J. Acoust. Soc. Am.*, 146, 2019.

Ozanich, E., Gerstoft, P., and Niu, H. "A deep network for single-snapshot direction-of-arrival estimation", *2019 IEEE Int. Work. Mach. Learn. Sig. Proc.*, 2019.

Bianco, M. J., Gerstoft, P., Traer, J., Ozanich, E., Roch, M.A., Gannot, S., Deledalle, C.-A. "Machine learning in acoustics: Theory and applications", *J. Acoust. Soc. Am.*, 146, 2019.

Ozanich, E., Gerstoft, P., and Niu, H. "Feedforward neural network for direction-of-arrival estimation", *J. Acoust. Soc. Am.*, 147, 2020.

Niu, H., Gerstoft, P., Ozanich, E., Li, Z., Zhang, R., Gong, Z., and Wang, H. "Block sparse Bayesian learning for broadband mode extraction in shallow water from a vertical array", *J. Acoust. Soc. Am.*, 147, 2020.

Ozanich, E., Thode, A., Gerstoft, P., Freeman, L. A., and Freeman, S., "Unsupervised clustering of coral reef bioacoustics," *submitted, J. Acoust. Soc. Am.*, (2021).

## ABSTRACT OF THE DISSERTATION

### Machine learning for localizing and characterizing underwater passive acoustics

by

Emma Ozanich

Doctor of Philosophy in Oceanography

University of California San Diego, 2020

Peter Gerstoft, Chair

Passive acoustics, or the recording of pressure signals from uncontrolled sound sources, is a powerful tool for monitoring man-made and natural sounds in the ocean. Passive acoustics can be used to detect changes in physical processes within the environment, study behavior and movement of marine animals, or observe presence and motion of ocean vessels and vehicles. Advances in ocean instrumentation and data storage have improved the availability and quality of ambient noise recordings, but there is an ongoing effort to improve signal processing algorithms for extracting useful information from the ambient noise. This dissertation uses machine learning as a framework to address problems in underwater passive acoustic signal processing. The chapters within this dissertation cover two types of problems: characterization and classification of ambient noise, and localization of passive acoustic sources.

First, ambient noise and passive acoustic signals were characterized in two locations. In the eastern Arctic, ambient noise was studied from April to September 2013 using a vertical hydrophone array as it drifted from near the North Pole to north of Fram Strait. Median power spectral estimates and empirical probability density functions (PDFs) along the array transit show a change in the ambient noise levels corresponding to seismic survey airgun occurrence and received level at low frequencies and transient ice noises at high frequencies. Noise contributors were manually identified and included broadband and tonal ice noises, bowhead whale calling, seismic airgun surveys, and earthquake *T* phases. The bowhead whale or whales detected were believed to belong to the endangered Spitsbergen population and were recorded when the array was as far north as 86°24’N. Then, ambient noise recorded in a Hawaiian coral reef was analyzed for classification of whale song and fish calls. Using automatically detected acoustic events, two clustering processes were proposed: clustering handpicked acoustic metrics using unsupervised machine learning, and deep embedded clustering (DEC) to learn latent features and clusters from fixed-length power spectrograms. When compared on simulated signals of fish calls and whale song, the unsupervised clustering methods were confounded by overlap in the handpicked features while DEC identified clusters with fish calls, whale song, and events with simultaneous fish calls and whale song. Both clustering approaches were applied to recordings from directional autonomous seafloor acoustic recorder (DASAR) sensors on a Hawaiian coral reef in February 2020.

Next, source localization in ocean acoustics was posed as a machine learning problem in which data-driven methods learned source ranges or direction-of-arrival directly from observed acoustic data. The pressure received by a vertical linear array was preprocessed by constructing a normalized sample covariance matrix (SCM) and used as the input for three machine learning methods: feed-forward neural networks (FNN), support vector machines (SVM) and random forests (RF). The effect of data preprocessing, including frequency bandwidth and snapshot averaging, were examined. Machine learning was implemented as a classification and regression problem. The FNN, SVM, RF and conventional matched-field processing were applied to recordings from ships in the Noise09 experiment to demonstrate the potential of machine learning for underwater source localization. The source localization problem was extended by examining the relationship between conventional beamforming and linear supervised learning. Then, a nonlinear deep feedforward neural network (FNN) was developed for direction-of-arrival (DOA) estimation for two-source DOA and for $K$-source DOA, where $K$ is unknown. $K$-source FNN achieved resolution and accuracy similar to Sparse Bayesian Learning (SBL) for single-snapshot and multiple Signal Classification (MUSIC) for multi-snapshot data with an unknown number of sources. The practicality of the deep FNN model was demonstrated on ships in the Swellex96 experimental data.

# Chapter 1

# Introduction

Ambient noise measurements have been relied on for decades to study under-ice conditions in the Arctic [11, 16, 20, 24, 32] and the behavior and movements of sound-producing animals. [34, 36, 42, 60] To better make sense of these passive acoustic data, automated signal processing methods have been developed for detecting and classifying sounds. [21, 31, 33, 38, 59] The problem of localizing and tracking passive acoustic sources has been addressed through advancements in array processing methods that better characterize the ocean environment [14] or utilize persistent waveguide physics. [17, 56]

This goal of this dissertation is to develop and apply machine learning methods to better analyze and localize passive underwater acoustic sources, particularly focusing on recent advances in machine learning methods and softwares. [1, 26, 52] The purpose is twofold: first, to examine machine learning performance in modeled and real-world scenarios, and second, to address considerations for posing an underwater acoustics problem using machine learning frameworks. Two underwater acoustic problems are considered:

1. *Characterizing and classifying unlabeled passive acoustic data* in the Eastern Arctic [48] and Hawaiian coral reef, [49] two regions that merit increased experimental observation due to their relevance to climate change. [4, 62] Machine learning has become a valuable tool

for classification of passively recorded bioacoustic signals (for a summary, see Sec. VIII in [5]). However, challenges remain in data processing and classification in acoustically noisy environments. We first demonstrate methods of spectral analysis for ambient noise in the Eastern Arctic and present a data-driven approach for removing unwanted non-acoustic noise. Then, we consider unsupervised machine learning clustering approaches for Hawaiian coral reef ambient noise that utilize both handpicked spectral and temporal features as well as deep feature learning, building from existing studies of fish call classification using supervised learning. [21, 31, 33]

2. *Localizing seagoing vessels in passive acoustic recordings.* [44–47] The chapters of this dissertation focus on source ranging [46] and direction-of-arrival (DOA). [47] Motivated by success in other domains, [10, 25, 53, 55] this dissertation leverages recent machine learning software and algorithms. In addition to utilizing current computational capability, our approach differs from existing studies on neural networks for underwater acoustics by using experimental observations for training as well as model-generated fields [57]$^{,}$ [50]$^{,}$ [8]$^{-}$ [3] and considering both classification and regression. [9] We use normalized sample covariance matrix inputs that include phase and amplitude information across an array, instead of complex pressure, phase-only, amplitude-only, transmission loss, eigenvalues, or backscatter. [3, 8, 9, 29, 40, 57, 58] For DOA, we propose using feedforward neural network to handle nonlinearities previously addressed using nonlinear kernels in SVM, [30, 35, 54] similar to the spectral estimation problem recently studied. [22]

## 1.1 Statistical analysis of ambient noise spectra

Statistical spectral methods are common used for analyzing longterm ambient noise with unknown signal content. A sliding window of length $N$ may be applied to the pressure timeseries

$x$ at $T$ time steps $t_i$, $i = 1, \ldots, T$ to compute a set of frequency spectra, [51]

$$p(f, t_i) = \frac{C \cdot B}{F_s N^2} \left| \sum_{n=0}^{N-1} w[n] x[t_i + n] e^{-i \cdot 2\pi \frac{f \cdot n}{F_s}} \right|^2 \tag{1.1}$$

for frequencies $f \in [0, \frac{F_s}{N}, \ldots, (N-1)\frac{F_s}{N}]$. $w[n]$ is the Hamming window, $B$ the window normalization factor, and $C$ is the sensor calibration coefficient for a sensor with sampling rate $F_s$. The $p(f, t_i)$ have units of power density ($\mu$Pa$^2$ per Hz) and are computed using the Fast Fourier transform algorithm.

The matrix $p(f, t_i)$, $f = 0, \ldots, (N-1)\frac{F_s}{N}$, $t_i = 1, \ldots, T$ is often expressed in the decibel scale, $S(f, t_i) = 10 \log_{10} p(f, t_i)$, where $S(f, t_i)$ is called the spectrogram. In Chapter 3, the spectrograms were computed for events lasting less than 2 s and used for extracting spectral features or for deep learning inputs.

Another approach is to estimate the empirical cumulative distribution function (ECDF) at each frequency across all $T$ windows. [51] The 50th percentile of the ECDF, or median, improves ambient noise characterization compared to the mean spectra. [39] In Chapter 2, spectral estimates across 3 and 4 day periods were used to compute the 10th, 50th, and 90th percentiles of the ECDF. The $P$th percentile is the spectral level that exceeds $P$ percent of all time windows. For example, the 90th percentile is exceeded in only 10% of time windows and corresponded to loud and short-lasting ice creaks in [48]. In Chapter 3, the ECDF was computed for individual acoustic features of detected events instead of spectra [49]. For that Hawaiian coral reef soundscape, the 90th percentile of the feature measuring number of impulses corresponded to fish calling during the dusk spawning chorus.

## 1.2 Localization in underwater acoustic waveguides

In shallow water ocean acoustics, localization of a source can be explicitly or implicitly linked to the measured pressure field through the channel propagation characteristics. The pressure field generated by a point source can be well-estimated by solving the separable Helmholtz equation. [23] In theory, the ocean can be expressed as a flat-bottomed waveguide of depth $D$, with no sound speed variation (Pekeris waveguide, see Ch. 2.4.5 [23]). The solution for this environment is well studied and reasonably accurate in many scenarios. It can be expressed as a set of propagating modes measured at depth $z$ generated by a source at range $r$ and depth $z_s$.

The exact relation between the source location parameters and the received field are specific to each ocean environment. In the simplified case of Pekeris waveguide, the pressure field depends on ocean sound speed (SSP) and sediment properties. In practice, the ocean environment is more complex: SSP is influenced by ocean processes, the ocean bottom is sloping and contains local bathymetric features, sediment is layered and spatially variable, and surface and volume scattering may have a non-negligle effect.

A number of existing studies on source localization rely on measured environmental parameters to compare modeled to measured pressure fields. Matched-field processing seeks to estimate the location parameters by finding a maximum likelihood solution between a propagation model and the measured ocean. [2, 41, 61] Other methods have considered signal processing improvements for matching the replicas and measurements. [12, 17] Statistical methods, including genetic algorithms, [19] Bayesian [13, 37] and trans-dimensional Bayesian inversion [15] seek to jointly estimate the ocean environment and the source location parameters.

In this dissertation, the problem of source localization was treated as a pattern matching problem between similar environments. Consistent measured patterns in the received field may be used to infer source location parameters without requiring explicit knowledge of the ocean environment. The research in Chapters 4 and 5 introduces recent developments in machine learning algorithms and implementation to existing research on neural networks and regression for underwater acoustic localization [29, 50, 57] and geoacoustic inversion. [3, 8, 9, 40, 58] In our work, machine learning was used to implicitly determine the relationship between the measured pressure fields and the source location parameters, including range [45, 46] and angle [47]. Related research has also considered source depth . [44]

## 1.3 Supervised machine learning

In supervised machine learning, the objective is to learn a mapping from an input, $\mathbf{x}$, to an output target $y$, given a set of $N$ labeled pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i\}_{i=1}^{N}$. [43] The labeled set $\mathcal{D}$ used to infer the mapping is called the training set. Supervised learning performance is measured on the capability of the trained model to perform on a labeled test set, $\mathcal{D}' = \{(\mathbf{x}_j, y_j)\}_{j=1}^{M}$, that is similar to but does not overlap with the training set.

In the following, $\mathbf{x}_i \in \mathbb{R}^D$ is a $D$–dimensional vector representing $D$ input features. The features may have physical meaning, as in [45–47] where the features were derived from the sample covariance matrix, or they may be statistically informative features derived from a feature learning process as in [47]. The target output is either discrete (classification) with $y_i \in \{1, \ldots, C\}$ or continuous (regression) with $y_i \in \mathbb{R}$. The three supervised machine learning methods applied within this dissertation are briefly described.

### 1.3.1 Neural networks

The feed-forward neural network (FNN), also known as multi-layer perceptron, is constructed using a feed-forward directed acyclic architecture. The outputs are formed through a series of functional transformations of the weighted inputs, [6] where for an input layer comprised of $D$ input variables $\mathbf{x}_t = [x^1, \cdots, x^D]^T$, the output is

$$\hat{y}_t^m = f\left(\sum_{i=1}^{D} w^{i,m} x_t^i\right) = f(\mathbf{w}^{mT}\mathbf{x}_t), \quad m = 1, ..., M \tag{1.2}$$

where $f(\cdot)$ is an arbitrary function and $\mathbf{w}^m$ a weight vector. The locally optimal set of weights $\mathbf{w}^m$, $m = 1, ..., M$, is estimated through inversion using gradient backpropagation. [28]

In convolutional neural networks (CNN), the input $\mathbf{x}_t$ becomes a 2D or 3D image $\mathbf{X}_t \in \mathbb{R}^3$ and the weight vector $\mathbf{w}^m$ is replaced by a 2D filter, with $\mathbf{W}^m \in \mathbb{R}^2$. The primary difference between FNN and CNN is that CNN uses weight sharing and downsampling by convolving a single filter across the entire input image whereas FNN typically weights each input feature separately. [6] This property leads to translational invariance in CNNs making them particularly useful for image feature extraction. [27]

Both FNN and CNN can be posed as classification or regression models. For classification, the training labels are expressed as $M$-dimensional binary vector $\mathbf{y}_t \in \{0, 1\}^M$ representing $M$ classes, with $y_t^m = \delta(m, m_{\text{true}})$, for $\delta()$ the Kronecker delta. The likelihood over the $M$ classes is estimated using the softmax function at the output, [6]

$$\hat{y}_t^m(\mathbf{x}, \mathbf{w}) = \frac{\exp(a_m(\mathbf{x}, \mathbf{w}))}{\sum_{j=1}^{M} \exp(a_m(\mathbf{x}, \mathbf{w}))}, \quad m = 1, \cdots, M \tag{1.3}$$

where $\mathbf{w}$ is the set of all weight and bias parameters, $a_m$ is an activation at an intermediate network layer, and $y_t^m$ satisfies $0 \leq y_t^m \leq 1$ and $\sum_m y_t^m = 1$. For regression, there is only one continuous-valued neuron in the output layer with $\hat{y}_t^m \in \mathbb{R}$. Details of the neural network models used in this dissertation are given in Chapters 3, 4, and 5.

### 1.3.2 Support vector machine

Support vector machines (SVM) use a linear hyperplane to separate the inputs $\mathbf{x}_i$ into two classes given the true class labels $t_i$ [46, 47]

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2, \tag{1.4}$$

$$\text{subject to } s_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1,\ldots,N \tag{1.5}$$

where $\mathbf{w}$ and $b$ are the weights and bias, and $\mathbf{w}^T\mathbf{x} + b = 0$ defines the hyperplane that separates the classes. Minimizing $\frac{1}{2}\|\mathbf{w}\|^2$ is equivalent to maximizing the margin between the nearest point and the separating hyperplane. To extend the SVM to nonlinear problems, $\mathbf{x}_i$ is replaced by $\phi(\mathbf{x}_i)$, [6] where $\phi$ is a nonlinear function. Slack variables are incorporated in (1.4) to penalize misclassified points. [6] Last, the SVM can be used to classify $K$ classes by learning $K(K-1)/2$ hyperplanes, one for each pair of classes, in a method known as "one-versus-one." [6] Additional details of the SVM are discussed in Chapters 4 and 5.

### 1.3.3 Random forest

The random forest (RF) uses statistical bagging on a set of $B$ randomly initialized decision trees to develop a robust classification model, [7]

$$\hat{f}^{\text{bag}}(\mathbf{x}_i) = \arg\max_{t_i} \sum_{b=1}^{B} I(\hat{f}^{\text{tree},b}(\mathbf{x}_i), t_i) \tag{1.6}$$

where $\hat{f}^{\text{tree},b}(\mathbf{x}_i)$ predicts the class of $\mathbf{x}_i$ for the $b$th tree. Each tree is constructed by iteratively partitioning the data. For example, if $\mathbf{x}_i$ is partitioned along the $m$th dimension at cutoff level $c$, then

$$\mathbf{x}_i \in \mathbf{x}_{\text{left}} \quad \text{if} \quad x_{im} > c, \quad \mathbf{x}_i \in \mathbf{x}_{\text{right}} \quad \text{if} \quad x_{im} \leq c \tag{1.7}$$

$$c^* = \arg\min_c G(c), \quad G(c) = \frac{n_{\text{left}}}{N} H(\mathbf{x}_{\text{left}}) + \frac{n_{\text{right}}}{N} H(\mathbf{x}_{\text{right}}), \tag{1.8}$$

where $c^*$ is the optimal cutoff level for each partition, $n_{\text{left}}$ and $n_{\text{right}}$ are the number of points in the left and right regions, and $H()$ is the error, also called impurity function. The partitioning process is repeated until a stop criterion is met, often when the number of points in a region falls below a threshold. Additional details of random forest are discussed in Chapter 4.

## 1.4 Dissertation Overview

This dissertation applies data-driven approaches and machine learning to a diverse set of problems in passive acoustics. Our results demonstrate that while environmental considerations are necessary for passive acoustics, machine learning can be used to enhance feature learning in soundscapes and as a potentially model-free localization method for passive acoustic sources of opportunity.

Chapters 2 and 3 address source characterization and classification in passive acoustics. In Chapter 2, the Arctic soundscape was analyzed using spectral analyses and manual signal analysis and compared to previous Arctic ambient noise studies. An automated method based on the background pressure identified non-acoustic noise for removal. In Chapter 3, sound sources from a Hawaiian coral reef were recorded on vector sensors and automatically detected. Handpicked spectral and temporal features were automatically extracted and clustered using K-means and hierarchical agglomerative unsupervised clustering methods. Then, normalized spectrograms were used in deep embedded clustering, a variant of the convolutional autoencoder neural network modified to clusters its learned features. Simulated signals based on observed signal characteristics were used to examine the performance of both clustering algorithms before implementing on a set of experimental detections.

The localization of passive acoustic sources is addressed in Chapters 4 and 5. Source and ship ranging was studied in Chapter 4 using FNN, SVM, and RF, with the normalized sample covariance matrix inputs. Simulations were used to examine the model performance under varying environmental and preprocessing conditions. Then, machine learning source ranging was conducted on acoustic data from a shallow-water ocean channel. Chapter 5 compares conventional beamforming (CBF) to linear machine learning methods for single-source DOA on a perturbed array. A deep FNN was developed for two or $K$ sources, where $K \leq 10$ here. DOA using FNN, sparse Bayesian learning (SBL) [18], and MUSIC (adaptive CBF) were demonstrated on simulated multi-source data and on experimental data.

# Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, G. S. Corrado C. Citro, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, M. Isard G. Irving, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *Software available from tensorflow.org*, 2015.

[2] A. B. Baggeroer, W. A. Kuperman, and P. N. Mikhalevsky. An overview of matched field methods in ocean acoustics. *IEEE J. Ocean. Eng.*, 18:401–424, 1993.

[3] J. Benson, N. R. Chapman, and A. Antoniou. Geoacoustic model inversion using artificial neural networks. *Inverse Problems*, 16:1627–1639, 2000.

[4] F. Bertucci, E. Parmentier, G. Lecellier, A. D. Hawkins, and D. Lecchini. Acoustic indices provide information on the status of coral reefs: an example from Moorea Island in the south pacific. *Scientific Reports*, 6:33326, 2016.

[5] M. J. Bianco, P. Gerstoft, J. Traer, E. Ozanich, M. A. Roch, S. Gannot, and C.-A. Deledalle. Machine learning in acoustics: Theory and applications. *J.Acoust. Soc. Am.*, 146:3590–3628, November 2019.

[6] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Chaps. 4, 5, and 7, 2006.

[7] L. Breiman. Random forests. *Mach. Learn.*, 45:5–32, 2001.

[8] A. Caiti and S. Jesus. Acoustic estimation of seafloor parameters: A radial basis functions approach. *J. Acoust. Soc. Am*, 100:1473–1481, 1996.

[9] A. Caiti and T. Parisini. Mapping ocean sediments by rbf networks. *IEEE J. Ocean. Eng.*, 19:577–582, 1994.

[10] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, 2011.

[11] W. C. Cummings, O. I. Diachok, and J. D. Shaffer. Acoustic transients of the marginal sea ice zone: A provisional catalog. *Naval Research Laboratory Memorandum Report 6408*, 2141, 1989.

[12] C. Debever and W. A. Kuperman. Robust matched-field processing using a coherent broadband white noise constraint processor. *J. Acoust. Soc. Am*, 122:1979–1986, 2007.

[13] S. E. Dosso and M. J. Wilmut. Bayesian multiple source localization in an uncertain environment. *J. Acoust. Soc. Am*, 129:3577–3589, 2011.

[14] S. E. Dosso and M. J. Wilmut. Bayesian tracking of multiple acoustic sources in an uncertain ocean environment. *J. Acoust. Soc. Am*, 133, 2013.

[15] Stan E Dosso, Jan Dettmer, Gavin Steininger, and Charles W Holland. Efficient trans-dimensional bayesian inversion for geoacoustic profile estimation. *Inverse Problems*, 30(11):114018, 2014.

[16] I. Dyer. The song of sea ice and other arctic ocean melodies. In *Arctic Technology and Policy*, pages 11–37. I. Dyer and C. Chryssostomidis (Hemisphere, New York, 1984.

[17] L. T. Fialkowski, M. D. Collins, W. A. Kuperman, J. S. Perkins, L. J. Kelly, A. Larsson, J. A. Fawcett, and L. H. Hall. Matched-field processing using measured replica fields. *J. Acoust. Soc. Am*, 107:739–746, 2000.

[18] K. L. Gemba, S. Nannuru, and P. Gerstoft. Robust Ocean Acoustic Localization With Sparse Bayesian Learning. *IEEE J. Sel. Top. Sig. Proc.*, 13(1):49–60, 2019.

[19] D. F. Gingras and P. Gerstoft. Inversion for geometric and geoacoustic parameters in shallow water: Experimental results. *J. Acoust. Soc. Am*, 97:3589–3598, 1995.

[20] C. R. Greene and B. M. Buck. Arctic ocean ambient noise. *J. Acoust. Soc. Am*, 36:1218, 1964.

[21] A. K. Ibrahim, H. Zhuang, L. M. Chérubin, M. T. Schärer-Umpierre, and N. Erdol. Automatic classification of grouper species by their sounds using deep neural networks. *J. Acoust. Soc. Am.*, 144(3):EL196–EL202, September 2018.

[22] G. Izacard, B. Bernstein, and C. Fernandez-Granda. A learning-based framework for line-spectra super-resolution. *CoRR*, abs/1811.05844, 2018.

[23] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt. *Computational Ocean Acoustics*. Springer Science & Business Media, NewYork, $2^{nd}$ edition, 2011.

[24] G. B. Kinda, Y. Simard, C. Gervaise, J. I. Mars, and L. Fortier. Arctic underwater noise transients from sea ice deformation: Characteristics, annual time series, and forcing in beaufort sea. *J. Acoust. Soc. Am*, 138:2034–2045, 2015.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst*, pages 1097–1105, 2012.

[26] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[28] Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pages 9–50, London, UK, UK, 1998. Springer-Verlag.

[29] R. Lefort, G. Real, and A. Drémeau. Direct regressions for underwater acoustic source localization in fluctuating oceans. *Appl. Acoust.*, 116:303–310, 2017.

[30] G. Lin, Y. Li, and B. Jin. Research on support vector machines framework for uniform arrays beamforming. In *2010 International Conference on Intelligent Computation Technology and Automation*, volume 3, pages 124–127, May 2010.

[31] T.-H. Lin, Y. Tsao, and T. Akamatsu. Comparison of passive acoustic soniferous fish monitoring with supervised and unsupervised approaches. *J. Acoust. Soc. Am.*, 143(4):EL278–EL284, April 2018.

[32] N. C. Makris and I. Dyer. Environmental correlates of pack ice noise. *J. Acoust. Soc. Am*, 79:1434–1440, 1986.

[33] M. Malfante, J. I. Mars, M. D. Mura, and C. Gervaise. Automatic fish sounds classification. *J. Acoust. Soc. Am.*, 143(5):2834–2846, May 2018.

[34] D. A. Mann and P. S. Lobel. Propagation of damselfish (*pomacentridae*) courtship sounds. *J. Acoust. Soc. Am.*, 101(6):3783–3791, February 1997.

[35] M. Martinez-Ramon, J. L. Rojo-Alvarez, G. Camps-Valls, and C. G. Christodoulou. Kernel antenna array processing. *IEEE Trans. Antennas Propag.*, 55(3):642–650, March 2007.

[36] K. P. Maruska, K. S. Boyle, L. R. Dewan, and T. C. Tricas. Sound production and spectral hearing sensitivity in the Hawaiian sergeant damselfish, *abudefduf abdominalis*. *J. Exp. Biol.*, 210:3990–4004, 2007.

[37] C. F. Mecklenbraüker and P. Gerstoft. Objective functions for ocean acoustic inversion derived by likelihood methods. *J. Comput. Acoust.*, 8:259–270, 2000.

[38] D. K. Mellinger and C. W. Clark. Methods for automatic detection of mysticete sounds. *Marine Freshw. Behav. Phys.*, 29(1-4):163–181, 1997.

[39] Nathan D. Merchant, Tim R. Barton, Paul M. Thompson, Enrico Pirotta, D. Tom Dakin, and John Dorocicz. Spectral probability density as a tool for ambient noise analysis. *The Journal of the Acoustical Society of America*, 133(4):EL262–EL267, 2013.

[40] Z. H. Michalopoulou, D. Alexandrou, and C. Moustier. Application of neural and statistical classifiers to the problem of seafloor characterization. *IEEE J. Ocean. Eng.*, 20:190–197, 1995.

[41] Z. H. Michalopoulou and M. B. Porter. Matched-field processing for broad-band source localization. *IEEE J. Ocean*, 21:384–392, 1996.

[42] S. E. Moore and R. R. Reeves. Distribution and movement,. In The Society for Marine Mammalogy, editor, *The bowhead whale*, pages 313–386. Allen Press, Lawrence, KS, 1993. Spec. Publ 2.

[43] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*, pages 389–397, 897–900. Massachusetts Institute of Technology, 2012.

[44] H. Niu, Z. Gong, E. Ozanich, P. Gerstoft, H. Wang, and Z. Li. Deep-learning source localization using multi-frequency magnitude-only data. *J.Acoust. Soc. Am.*, 146:211–222, July 2019.

[45] H. Niu, E. Ozanich, and P. Gerstoft. Ship localization in Santa Barbara Channel using machine learning classifiers. *J. Acoust. Soc. Am.*, 142:EL455–EL460, November 2017.

[46] H. Niu, E. Reeves, and P. Gerstoft. Source localization in an ocean waveguide using supervised machine learning. *Journal of the Acoustical Society of America*, 142(3):1176–1188, 2017.

[47] E. Ozanich, P. Gerstoft, and H. Niu. Feedforward neural network for direction-of-arrival estimation. *Journal of the Acoustical Society of America*, 147(3):2035–2048, 2020.

[48] E. Ozanich, P. Gerstoft, P. F. Worcester, M. A. Dzieciuch, and A. Thode. Eastern Arctic ambient noise recorded on a drifting vertical array. *Journal of the Acoustical Society of America*, 142(4):1997–2006, 2017.

[49] E. Ozanich, A. Thode, P. Gerstoft, L. A. Freeman, and S. Freeman. (na): Unsupervised clustering of coral reef bioacoustics. In Prep.

[50] J. M. Ozard, P. Zakarauskas, and P. Ko. An artificial neural network for range and depth discrimination in matched field processing. *J. Acoust. Soc. Am*, 90:2658–2663, 1991.

[51] S. E. Parks, I. Urazghildiiev, and C. W. Clark. Variability in ambient noise levels and call parameters of north atlantic right whales in three habitat areas. *The Journal of the Acoustical Society of America*, 125(2):1230–1239, 2009.

[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res*, 12:2825–2830, 2011.

[53] N. Riahi and P. Gerstoft. Using graph clustering to locate sources within a dense sensor array. *Signal Processing*, 132:110–120, 2017.

[54] D. Salvati, C. Drioli, and G. L. Foresti. A weighted MVDR beamformer based on SVM learning for sound source localization. *Pattern Recognition Letters*, 84:15–21, 2016.

[55] M. L. Sharma and M. K. Arora. Prediction of seismicity cycles in the himalayas using artificial neural networks. *Acta Geophysica Polonica.*, 53:299–309, 2005.

[56] H. C. Song and Chomgun Cho. Array invariant-based source localization in shallow water using a sparse vertical array. *J. Acoust. Soc. Am*, 141:183–188, 2017.

[57] B. Z. Steinberg, M. J. Beran, S. H. Chin, and J. H. Howard. A neural network approach to source localization. *J. Acoust. Soc. Am*, 90:2081–2090, 1991.

[58] Y. Stephan, X. Demoulin, and O. Sarzeaud. Neural direct approaches for geoacoustic inversion. *J. Comput. Acoust.*, 6:151–166, 1998.

[59] A. M. Thode, K. H. Kim, S. B. Blackwell, C. R. Greene, C. S. Nations, T. L. McDonald, and A. M. Macrander. Automated detection and localization of bowhead whale sounds in the presence of seismic airgun surveys. *J. Acoust. Soc. Am*, 131:3726–3747, 2012.

[60] T. C. Tricas and K. S. Boyle. Acoustic behaviors in Hawaiian coral reef fish communities. *Mar Ecol Prog Ser*, 511:1–16, September 2014.

[61] E. K. Westwood. Broadband matched-field source localization. *J. Acoust. Soc. Am*, 91:2777–2789, 1992.

[62] P. F. Worcester, M. A. Dzieciuch, and H. Sagen. Ocean acoustics in the rapidly changing arctic. *Acoustics Today*, 16(1):55–64, 2020.

# Chapter 2

# Eastern arctic ambient noise on a drifting vertical array

Ambient noise in the eastern Arctic was studied from April to September 2013 using a 22 element vertical hydrophone array as it drifted from near the North Pole (89°23’N, 62°35’W) to north of Fram Strait (83°45’N 4°28’ W). The hydrophones recorded for 108 min/day on six days per week with a sampling rate of 1953.125 Hz. After removal of data corrupted by non–acoustic transients, 19 days throughout the transit period were analyzed. Noise contributors identified include broadband and tonal ice noises, bowhead whale calling, seismic airgun surveys, and earthquake $T$ phases. The bowhead whale or whales detected are believed to belong to the endangered Spitsbergen population and were recorded when the array was as far north as 86°24’N. Median power spectral estimates and empirical probability density functions (PDFs) along the array transit show a change in the ambient noise levels corresponding to seismic survey airgun occurrence and received level at low frequencies and transient ice noises at high frequencies. Median power for the same periods across the array show that this change is consistent in depth. The median ambient noise for May 2013 was among the lowest of the sparse reported observations in the eastern Arctic but comparable to the more numerous observations of western Arctic noise.

## 2.1 Introduction

Ambient noise in the Arctic Ocean is strongly influenced by its sea ice cover and upward refracting sound speed profile. Internal frictional shearing, thermal stress fracturing, and interaction within leads in the ice generate distinct sounds that are received acoustically at levels exceeding 100 dB re 1$\mu$ Pa$^2$ Hz$^{-1}$. The widespread ice cover deters many animal species from venturing far north, but attracts species capable of seeking ice leads or generating their own breathing holes, such as bowhead whales. [1] At the same time, the upward–refracting sound speed profile and nearly year–round ice cover allow low frequency signals to propagate long distances while attenuating higher frequency components. This unique environment depends strongly on the properties of the Arctic sea ice, including percentage of areal cover, thickness (age), under–ice roughness, and lateral extent. Over the past decade, the Arctic sea ice has dramatically reduced in thickness as well as annual extent, [2] resulting in unknown changes to the ambient noise environment that this study investigates through use of recent data and analysis.

Sea ice noise and the Arctic ambient noise properties have historically been an area of interest in underwater acoustics. [3], [4] Measurements of transient ice noises have shown that they are highly non–Gaussian, [5] varying in frequency, bandwidth, length, and received sound level according to the sea ice properties and environmental conditions, [6] but are often more prevalent near ice ridges. [7], [8] The cumulative ambient noise levels generated by ice noise have been shown to correlate with environmental variables like wind, air pressure, and temperature. [9], [10], [11], [12] Near the Marginal Ice Zone (MIZ), where the ice is subject to increased wave forcing, noise levels have been shown to be as much as 10 dB higher than those further away from the MIZ. [13], [14] Sea ice is a strong scatterer that attenuates high frequencies at a much higher rate than the open ocean, [15] although the exact attenuation coefficients depend on the local sea ice structure in ways that have yet to be determined. [16] Due in large part to biological activity and experimental accessibility, the Western Arctic ambient noise near the

Beaufort Sea [12], [17], [18], [19] has been studied more extensively than the eastern Arctic ambient noise (defined here as areas east of 60°W). Studies north of 85°N are extremely rare. [1]

In April 2013, a bottom–moored vertical hydrophone array was deployed at Ice Camp Barneo near 89°N, 62°W. The experiment was designed to study acoustic propagation and ambient noise under the sea ice. Around April 15 the mooring cable failed. The subsurface float rose to the surface and remained there, with the array hanging unweighted below. It drifted southward with the Transpolar Current toward the Fram Strait, recording ambient noise as scheduled. MicroCAT pressure measurements (see Sec. 2.2.2) showed that the array was vertical under its own weight during much of the transit. The resulting data record the spatiotemporal variation of the far northern Arctic ambient noise ($>$ 85 °N). In this study, the dataset is analyzed and the observations are interpreted in terms of previous studies of this ambient noise.

This paper is organized as follows. In Sec. 2.2, the acoustic experiment is described, data processing methods are explained, and the collection of supplementary environmental data is discussed. Sec. 2.3 discusses select noise events. Sec. 2.4 presents the results of statistical ambient noise analyses in both time and depth, and Arctic ambient noise power estimates from previous studies are compared with the results. The goal of this paper is to establish an understanding of ambient noise contributors and sound levels in the northeastern Arctic during summer 2013.

## 2.2 Methods

### 2.2.1 Acoustic measurements

A 600 m long bottom-moored acoustic receiving array was deployed at Ice Camp Barneo, 89°23'N, 62°35'W, on April 14. Twenty-two omnidirectional hydrophone modules (H.M.) were spaced along the array, with H.M. 1–10 separated by 14.5 m and H.M. 11–22 separated by logarithmically increasing spacing starting at 16.5 m (Table 2.1, Fig. 2.1). The topmost hydrophone was 11.6 m below the subsurface float. The hydrophones recorded underwater sound for 108 min/day six days per week, starting at 1200 UTC each day, with a sampling frequency of 1,953.125 Hz. The hydrophone recording schedule was constrained by the amount of data storage available in the hydrophone modules. Acoustic recordings are available for 119 days between April 29 and September 20.

**Table 2.1**: Instrument spacing, numbering, and MicroCAT sampling periods for the instruments on the VLA during its drifting period. The depth estimates assume that the subsurface buoy was floating at 0 m, an assumption confirmed by the MicroCAT measured depths.

| H.M. # | H.M. Depth (m) | MicroCAT Depth (m) | MicroCAT # | MicroCAT Sampling Period (s) |
|---|---|---|---|---|
| 1 | 11.6 | 4.6 | 1 | 480 |
| 2 | 26.1 | 24.6 | 2 | 480 |
| 3 | 40.6 | | | |
| 4 | 55.1 | 49.6 | 3 | 480 |
| 5 | 69.6 | | | |
| 6 | 84.1 | | | |
| 7 | 98.6 | 99.6 | 4 | 480 |
| 8 | 113.1 | | | |
| 9 | 127.6 | | | |
| 10 | 142.1 | | | |
| 11 | 158.7 | 149.6 | 5 | 380 |
| 12 | 177.7 | | | |
| 13 | 199.5 | 200.6 | 6 | 380 |
| 14 | 224.4 | | | |
| 15 | 253 | 249.6 | 7 | 380 |
| 16 | 285.7 | | | |
| 17 | 323.2 | | | |
| 18 | 366.1 | 349.6 | 8 | 380 |
| 19 | 415.2 | | | |
| 20 | 471.4 | 449.6 | 9 | 380 |
| 21 | 535.8 | | | |
| 22 | 609.6 | 599.6 | 10 | 300 |

The raw acoustic recordings were scaled to be in units of instantaneous sound pressure using the analog-to-digital conversion parameters, the gain, and the hydrophone receiving sensitivity given by the manufacturer. The hydrophone receiving sensitivity was nearly constant above 50 Hz but highly frequency dependent below 50 Hz. The system noise floor was computed using a model that combines the known self–noise of its individual components. The system was experimentally tested in a Faraday cage and by calculating the coherence between multiple sensors recording noise in a quiet room. Both tests fit the modeled system noise floor well.

Median (50%) spectral estimates were created by segmenting three or four day periods of data (see Sec. 2.4.1) into 4096-point windows ($\sim$ 2 s), taking a 16,384-point Fast Fourier Transform to interpolate to high resolution frequency bins of 0.12 Hz, and sorting the individual spectral estimates by power level at each frequency bin. The probability density (PDF) was estimated from these spectral estimates using 100 power bins of equal width at each frequency. The PDF for a target frequency was obtained by averaging three PDFs closest to the target frequency. Spectrograms were estimated using shorter, 512-point windowed segments ($\sim$ 0.25 s) zero-padded to 2048 points (df $\approx$ 1 Hz) in order to capture transients of length $<$ 1 s. Unless otherwise noted, the data were recorded at 84.1 m depth (hydrophone # 6) for comparability to other ambient noise studies in the eastern Arctic. [10]

### 2.2.2 MicroCATs

Ten Sea-Bird SBE 37–SM/SMP MicroCAT instruments, measuring temperature, conductivity, and pressure (dBars) were co-located with the hydrophones, spaced 25, 50, 50, 50, 50, 100, 100, and 150 m apart. The topmost MicroCAT was located 4.6 m below the subsurface float (Table 2.1, Fig. 2.1). The MicroCATs began recording on April 28 and sampled continuously until September 19. The sampling period for each MicroCAT is shown in Table 2.1.

### 2.2.3 GPS coordinates

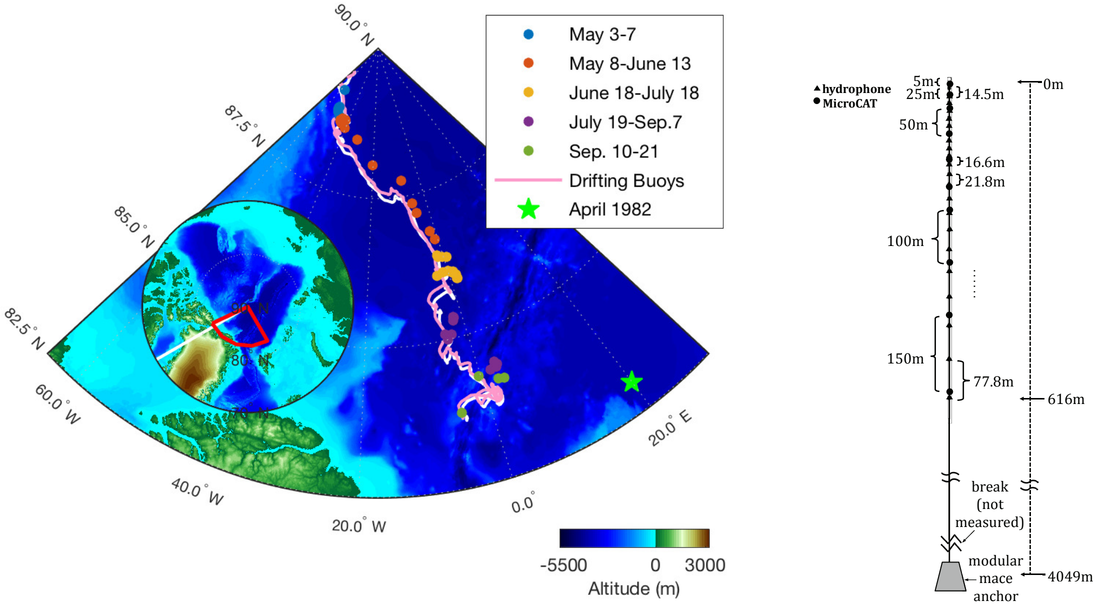

**Figure 2.1**: A bathymetric relief map displaying the location of the receiving array divided according to hydrophone processing period (see Sec. 2.4) along with the location of two concurrently deployed ice–moored buoys with daily GPS and the April 1982 FRAM IV ice camp (★). A map inset shows the location of the array path relative to the Arctic and a line indicating the 60°W longitude. The moored array design is shown to the right of the map.

A Xeos Technologies Kilo Iridium-GPS mooring location beacon located on top of the subsurface float began transmitting ALARM messages on May 3, indicating that the mooring had prematurely surfaced. The reported position at the time of surfacing was 88°50'N, 51°17'W, 63 km from the deployment location. Analysis of an acoustic survey on April 14, following deployment of the mooring, revealed that the acoustic release was significantly shallower than expected. The implication is that the mooring failed shortly after deployment, but the subsurface float was trapped beneath sea ice, preventing the location beacon from obtaining GPS positions or transmitting ALARM messages until it was exposed on May 3. The float drifted southward in the Transpolar Drift. There were frequent gaps in transmissions from the location beacon which are presumed to coincide with periods when the subsurface float was covered by sea ice. The buoy was recovered on September 21, at 84°03’N, 03°05’W. The mooring line was found to have parted immediately above the anchor (Fig. 2.1).

### 2.2.4 Bathymetry

The International Bathymetric Chart of the Arctic Ocean from the National Centers for Environmental Information was used to construct a map of the ocean depth relative to the array location (Fig. 2.1).

The measured depths varied between 2.5 km and 4.7 km during the drift period. The Gakkel Ridge was the shallowest area crossed by the array, and it is possible that the array interacted with the bottom there or in other shallow regions. Without instrumentation on the lower array, the presence of array–bottom interaction cannot be determined.

### 2.2.5 Sea Ice Concentration

Daily sea ice concentration, defined as the areal percentage of satellite imagery above a certain brightness level, was obtained from the Advanced Microwave Scanning Radiometer-2 (AMSR-2) 89-GHz channel satellite dataset, [20] provided in a 4 km X 4 km gridded format from the Institute of Environmental Physics, University of Bremen, Germany. The sea ice concentration ranges from 0 (no ice) to 100 (solid ice). The georeferenced latitude and longitude grids were transformed into regular latitude and longitude grids with 0.1° resolution with the ice concentration interpolated to the array location.

In addition, the AMSR-2 satellite data were used to determine the daily distance from the array to the ice edge. This distance was about 1000 km in April and 200 km in September, decreasing steadily as the array drifted closer to the MIZ.

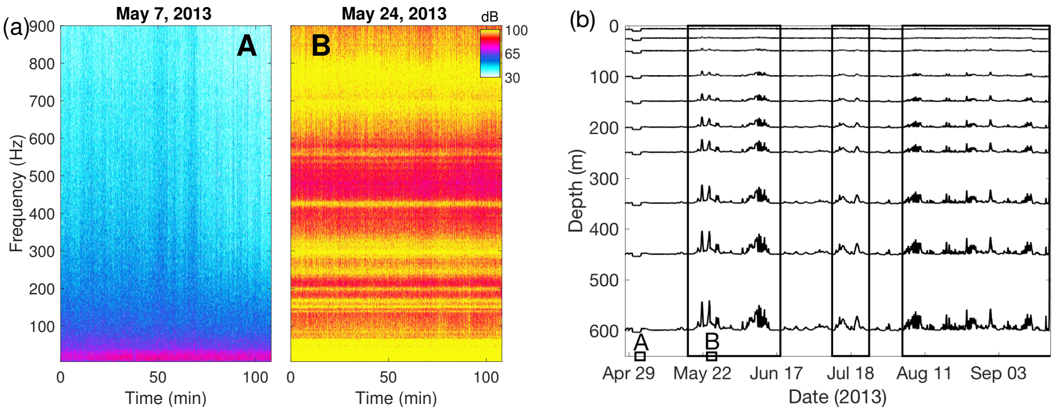

**Figure 2.2**: (a) Spectrograms generated from hydrophone recordings at 609.6 m depth during periods containing typical ambient noise (**A**) and strong spectral bands considered non–acoustic artifacts (**B**). (b) MicroCAT pressure measurements at ten depths exhibit periods of shallowing (rectangles) that correspond to artifacts in (a).

### 2.2.6 Filtering/Noise Removal

The drifting array was heavily contaminated by self–noise at certain times. Low frequency ($f < 5$ Hz) cable strum was observed. Strong spectral bands were also observed, exceeding 100 dB re 1 $\mu$Pa$^2$ Hz$^{-1}$ and extending to the Nyquist frequency (976.56 Hz). These elevated spectral levels, predominant in the frequency bands 0–50 Hz, 250–325 Hz, and 600–900 Hz (Fig. 2.2(a)), were found to correspond with periods of unexpectedly low pressures (depths) on the MicroCATs (Fig. 2.2(b)), making them unlikely to be caused by propagating acoustic noise and more likely to be noise artifacts. With the buoyant subsurface float constrained to the surface, flow past the mooring lifts and thus tilts the array and reduces the MicroCAT pressures (depths). Potential non-acoustic noise sources on the mooring, which lacked fairing, include strumming–induced vibration, flow noise, and/or bottom interaction. The noise artifacts could not be removed by a ω–k beamforming filter indicating that the instruments were directly affected.

Environmental variables including wind, temperature, or ocean waves may be related to the acoustic artifacts, but the lack of meteorological stations or oceanographic buoys near the drifting array makes drawing conclusions about these relationships difficult. For example, daily estimates of wind speed from a reanalysis model are correlated with the daily median acoustic power at 400 Hz, after filtering, with 95% confidence. However, measurements of temperature and pressure at the northerly

To remove affected data, the median MicroCAT pressure for each day was computed. The pressure on MicroCAT #10 (599.6 m) had the largest variation between days and was used as an indicator of flow-related noise. By comparing the good and bad spectrograms with the median pressures on MicroCAT #10 (Fig. 2.2) , it was found that most corrupted data had a median MicroCAT pressure of less than 604.9 dBars. Therefore days with $p_{\mathrm{MicroCAT,10}} < 604.9$ dBars were not used. This method selected 19 days for further analysis: April 30, May 1, 2, 7, 8, 9, 12, 14, June 16, 18, July 3, 14, 19, 24, August 2, and September 10, 18, 19, 20. There is evidence that the noise artifacts were not completely removed for one or two periods (see Sec. 2.4).

## 2.3 Arctic ambient noise source effects

### 2.3.1 Underwater Sound Propagation

Eastern Arctic ambient noise is influenced by the characteristics of sound propagation which are affected by the oceanographic water masses and sea ice cover in the region. [21] Much of this propagation is over long distances due to the intermittent nature of nearby ice noise events (see Sec. 2.3.2), the infrequency of biological activity (see Sec. 2.3.3), and the locations of regular anthropogenic activity (see Sec. 2.3.4).

The sound speed profile in the eastern Arctic is strongly upward refracting with a minimum at the ocean–ice interface (Fig. 2.3(a)). The relevant water masses include Polar Water (0–200m), Arctic Intermediate Water (AIW, 200–1000m), and Deep Polar Water (>1000m). [21] Profiles in the eastern Arctic differ from western Arctic in that the depth of the AIW temperature maximum is considerably shallower in the eastern Arctic.

In completely ice–covered environments, the sea ice acts as a low–pass filter. [21] Higher

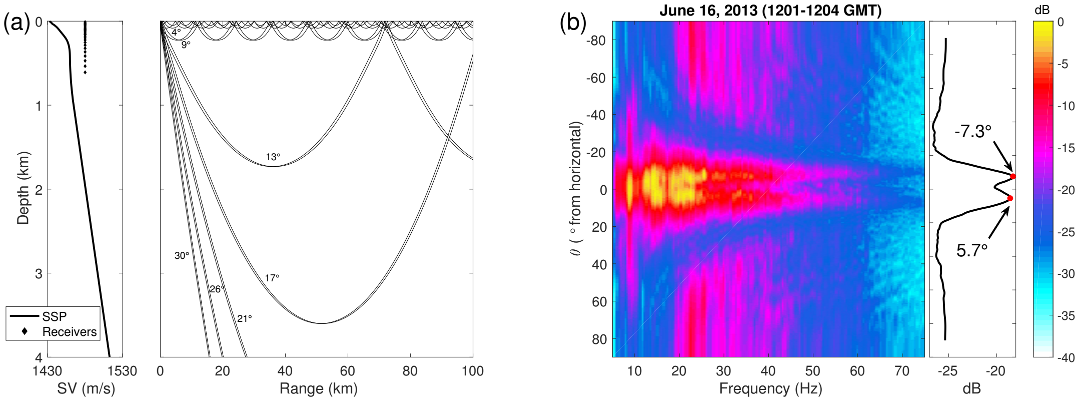

**Figure 2.3**: (a) Bellhop ray propagation model [22] for a near–surface source using the sound speed profile measured at Ice Camp Barneo demonstrates the strongly upward refracting profile. Rays were launched between ±30°from horizontal. (b) Bartlett beamformer at received airgun pulse frequencies, averaged across 1201–1204 GMT on June 16. The arrivals at $-7^\circ$ and $5^\circ$ indicate the preservation of intermediate ray angles over long range propagation ($\theta < 0^\circ$ is upward–looking).

frequency sound ($f > 30$ Hz and $\lambda < 50$ m) is strongly scattered at the water–ice interface. In addition, the number of reflections from the sea ice per kilometer increases as a propagating ray's angle decreases (<5°, Fig. 2.3(b)).

On the other hand, steeper rays (> about 13–15°) experience fewer reflections per kilometer but will interact with bathymetric features, especially at the Gakkel Ridge where the ocean depth shallows to nearly 2 km (Fig. 2.1(a)). At low frequencies (Fig. 2.4 at 5 Hz), even the lowest modes interact with and scatter from bathymetric features, leading to lower ambient noise levels below 10 Hz.

### 2.3.2 Ice–generated noise

Ice noises were observed to be either broadband or tonal in nature. Broadband noise generated by sea ice [6] appears as periods of elevated sound level, here ranging from 5–20 dB above the median level at 500 Hz (Fig. 2.5(a)) and lasting from 10–500 s. Broadband ice noise extended across the frequency band (Fig. 2.5(a)). Tonal ice noises are single–frequency or

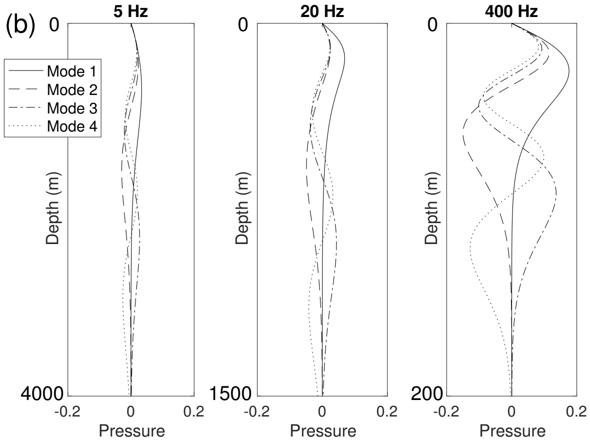

**Figure 2.4**: Modal structure of the eastern Arctic environment at three frequencies. Each panel has a depth scale appropriate for the vertical scale of the modes at that frequency.

harmonic signatures modulated in time (Figs. 2.5(b)-2.5(d)).

Xie and Farmer [23] demonstrated that constant–frequency ice tonals could be modeled as resonances in an infinitely long sea ice block of uniform height, density, and velocity generated by frictional shear stress on its edge. The non–constant tonals observed here may indicate anomalies in the local height or composition of the sea ice or a frictional stress that is velocity–dependent (Fig. 2.5(b)). The slope and curvature of the tonals varies between hydrophone recordings (Fig. 2.5(c)), indicating that significant changes in ice properties and dynamics may occur within the spatiotemporal span of 2–3 array drift days.

Another interesting case are sets of modulated harmonics, ranging from 200–900 Hz, that are 8–10 dB louder than the background spectrum and last about 4 s, recurring with a period of about 9 s (Fig. 2.5(d)). These tonals may be due to ocean waves impinging on the sea ice edge, generating seismic or flexural waves that propagate within the sea ice if the product of the noise frequency and the sea ice thickness is less than about 300 Hz–m [24] and couple into the water column as periodically modulated harmonics. The observation of these tonals on the receiving array suggest that these effects can be seen at least as far as 230 km from the ice edge.

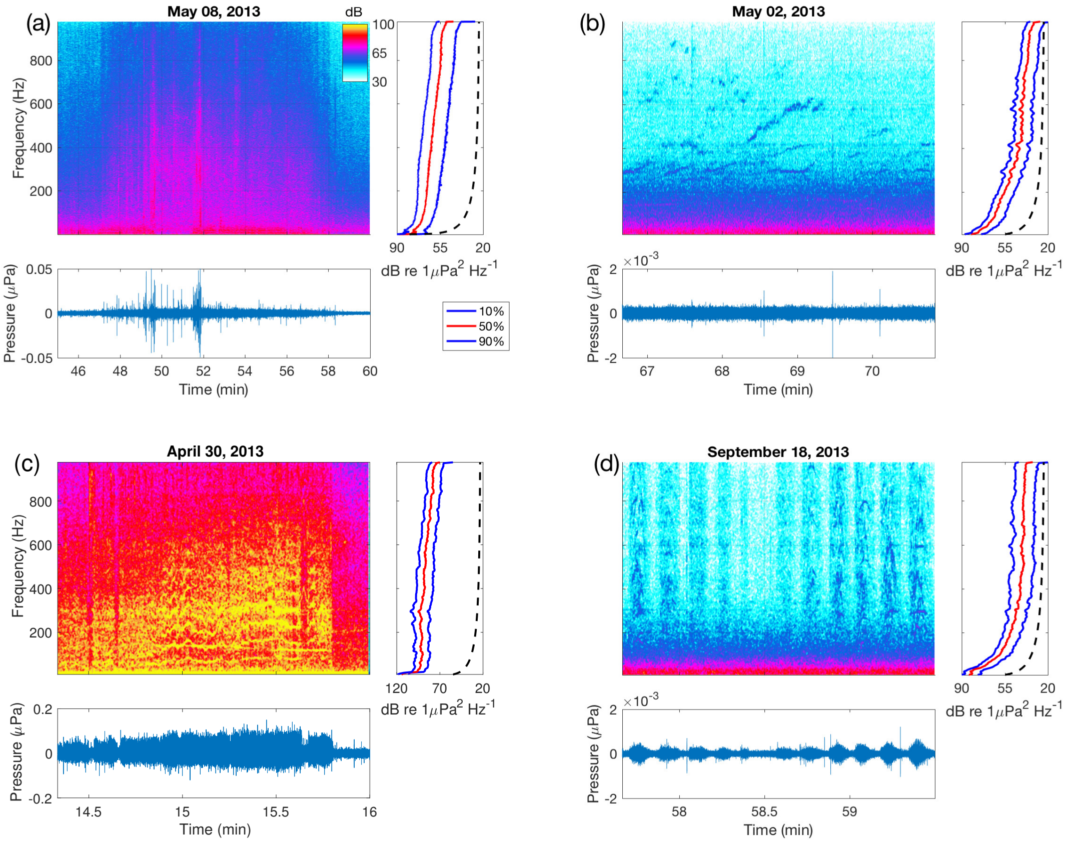

**Figure 2.5**: Spectrograms of ice noises including (a) a broadband event or events lasting up to 10 min, recorded on May 8, (b) non–constant tonals without harmonics lasting up to 1 min, recorded on May 2, (c) near-constant harmonic tonals lasting 2 min, recorded on April 30, and (d) non–constant, modulated harmonic tonals lasting for 5 s with a recurrent period on the order of ocean swell (9 s), recorded 230 km from the sea ice edge on September 18. The recording system noise is shown by the dashed black line.

### 2.3.3 Biological Sources

Bowhead whale calls were observed during the summer 2013 array transit (Fig. 2.6). The length of the call series lasted between 30 s and 7 min. The identification of the sound as a bowhead whale call was conducted by a manual analyst who led the team that identified thousands of bowhead whale calls in passive acoustic datasets recorded by instruments deployed during bowhead whale migrations along the North Slope of Alaska between 2008–2014. [25]' [26] Calls were observed on June 18, July 3, 19, and 24. These calls were recorded when the array was

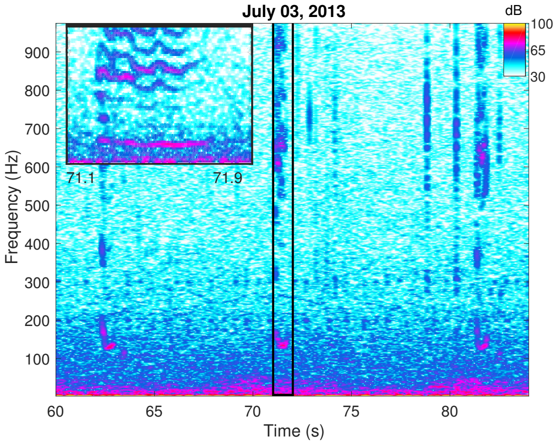

**Figure 2.6**: A series of calls from what is believed to be a Spitsbergen bowhead whale. The calling periodicity is about 10 s. These three calls were taken from a series lasting 55 s. The rectangle corresponds to the inset figure and shows a single call with harmonics from 150–976 Hz. The time axis in both figures is relative to 7 min in the recording on July 3.

northward of 85°N, at least 290 km north of other recordings in the region. [27] Sea ice cover from AMSR2 satellite data [20] was estimated to be higher than 90% locally at the array for these days (Fig. 2.7).

Previous observations of bowhead whales have occurred southward of 82°30'N. Before the year 1818, the prolific species was fished in the region about 200 km west of Spitsbergen, between 76°N and 80°N. By 1818, this group had been depleted nearly to extinction. [27] More recently, individuals or small groups have been acoustically detected as far north as 82°30'N. [1] Satellite–tagged whales in western Greenland spent most of their time in 90% to 100% ice cover far (>100km) inside the ice edge. [28] A recent study of Spitsbergen bowhead whale calling near 78°50'N, 0°W recorded no calls between April 30 and September 1 in 2009.

Measurements of the relative timing of the whale call across the array aperture reveal that the animal was at least 50 km distant. However, placing an upper bound on the range is difficult. Using received levels to estimate source range is imprecise for two reasons: the bowhead whales are capable of calling across a broad spread of source levels [26], and uncertainties arise rise when modeling transmission loss due to scattering of signals from ice. Using timing measurements of

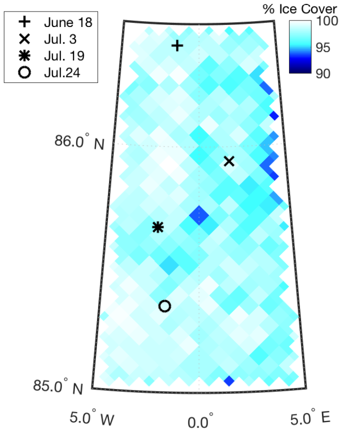

**Figure 2.7**: AMSR2 satellite ice coverage averaged over the days when bowhead whale calls were recorded along with the location of the array on those days. Ice cover was close to 100% at the array on these days.

signal arrivals across the array for localization is feasible, but requires that the vertical array tilt and sound speed profile be modeled or inverted correctly, a topic beyond the scope of the present paper.

### 2.3.4 Seismic Survey Signals

Broadband pressure pulses generated by airguns are used to image the geological structure beneath the seafloor during seismic surveys. At long distances, frequencies higher than about 100 Hz are attenuated. The resulting pulses are observed on hydrophone receivers at frequencies below 50 Hz. Distant noise from seismic surveys can be observed almost daily in the Fram Strait during summer months. E.g. in a previous dataset in the Fram Strait, airgun surveys were observed on 90–95% of days between July and September 2009. [29]

In this dataset, airgun pulses were observed between May 7 and Sep. 19 and were present on 11 of the 19 recording days (Fig. 2.8(a)), with nearly continuous pulses detected during the 108 min recording period whenever observed. Location, type and date of surveys in Norwegian territory were obtained from the Norwegian Petroleum Directorate. According to these data, the array was 1800–3500 km distant from seismic surveys at the start in April and 1000–3000 km

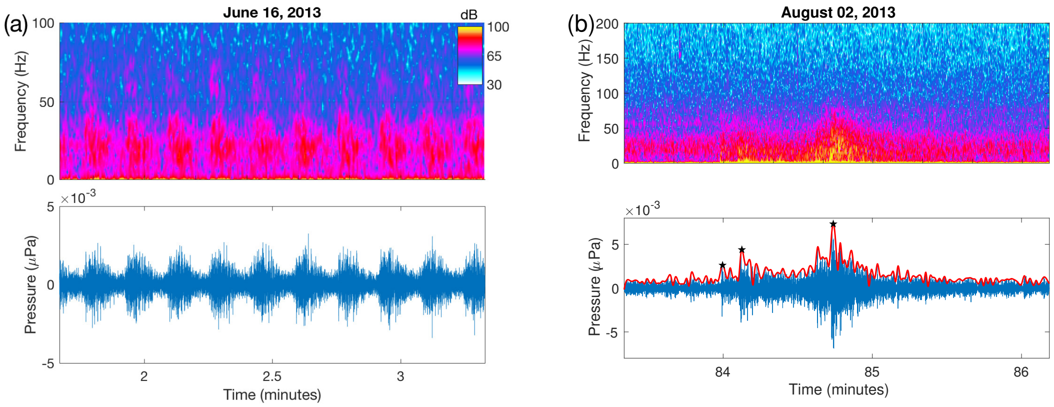

**Figure 2.8**: Spectrograms and time series of (a) low–frequency pulses generated by a distant airgun survey recorded on June 16 and (b) an earthquake recorded on August 2, where the wave arrival delays are used to estimate source range.

distant at the end in September. Seismic surveys conducted in the Canadian Arctic during summer 2013 may have been detected, but survey details were not publicly available.

Transmission loss estimates across the MIZ near the Fram Strait, extending as far as 150 km into the ice, have demonstrated that the under–ice transmission loss is smaller than previously proposed at low frequencies. [16], [30] The observations here also suggest that the change in transmission loss far into the compact ice is small, but uncertainties in source spectrum and distance make quantitative transmission loss estimates unreliable.

### 2.3.5 Arctic Basin Earthquakes

Hydrophone arrays are valuable earthquake monitoring tools. The acoustic $T$–phase pressure wave (see Fig. 2.8(b)) is coupled into the water column at a seamount or down–sloping bathymetric feature near the earthquake. The versatility of hydrophone arrays enables them to be deployed in difficult areas such as the active Gakkel Ridge in the ice–covered Arctic, where ocean bottom seismometers are challenging to deploy. [31]

Time difference of arrival between the $T$, $P$, and $S$ arrivals can be used on a hydrophone array to estimate the earthquake distance:

$$R = \frac{\Delta\tau}{\left(\frac{1}{v_T} - \frac{1}{v_P}\right)} \tag{2.1}$$

where $R$ is the range to the earthquake, $\Delta\tau$ is the arrival time difference, $v_T$ is the group velocity of the $T$–phase, and $v_P$ is the group velocity of the $P$ wave (or $S$ wave).

Three $T$–phase arrivals were observed during the array transit along with occasional $P$ and $S$ wave arrivals (Fig. 2.8(b)). Overall, three $T$–phase events were identified in the data, each lasting 1 min. The arrivals in Fig. 2.8(b) are applied to the time difference method in Eq. 2.1 with $v_T$ from the CTD measurement (1.44 km/s) at deployment and $v_P$, $v_S$ (6.1 and 3.1 km/s) estimated from the IASPEI seismic catalogue and adjusted to achieve agreement between estimates. Although the travel time of the $T$–phase may be biased depending on where it couples into the water column, the estimated earthquake distances of 90 km for the $P$–$T$ difference and 100 km for $S$–$T$ difference agree well here.

The earthquake distance estimate indicates that the event originated at the Gakkel Ridge. The earthquake was not registered in the Global Seismic Network catalogue which only records events with $m_b > 4$. The detection of $T$, $P$, and $S$ arrivals on a single hydrophone for an unregistered earthquake demonstrates the potential for underwater acoustic monitoring of low magnitude seismic activity near the Gakkel Ridge.

## 2.4 Arctic ambient noise levels

### 2.4.1 Eastern Arctic Ambient Noise, Summer 2013

Statistical analyses were conducted for three and four day periods across the array drift path: May 1, 2, 7; May 8, 9, 12, 14; June 16, 18, July 3, 14; July 19, 24, August 2; and September 10, 18, 19. Using three and four day averages reduces the inter–period variance observed among

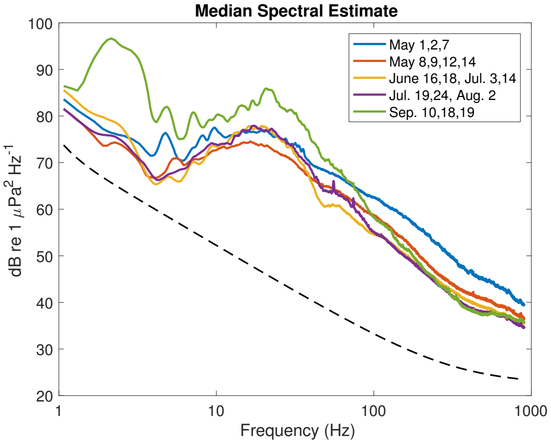

**Figure 2.9**: Median power spectral estimates for three and four day periods in summer 2013. The recording system noise is shown by the dashed black line.

daily estimates while demonstrating the same frequency–dependent ambient noise trends. April 30 and September 20 contain anomalous ice and ship noise events and are excluded from the statistical analyses.

The median power spectra show characteristics of Arctic ambient noise and its sources (Fig. 2.9). The broad peak at 15–20 Hz is attributed to the ice–scattered propagation characteristics of distant sources, [10] as higher frequencies are more attenuated and lower frequencies have bottom interacting modes (see Sec. 2.3.1). Seismic airgun surveys increase the median power at frequencies between about 10 Hz and 100 Hz (Fig. 2.9) due to the dispersive quality of the pulse arrivals. Observations of the spectrogram estimates confirm that the increase in low frequency power for September results from an increase in the received levels of airgun pulses. Likewise, decreased low frequency power in the May 8, 9, 12, 14 period results from lulls in the presence of airgun noise. Transient ice noises result in elevated power levels for frequencies above 100 Hz (Fig. 2.9). Transient ice noises were observed in the spectrograms estimates most frequently and at the highest received levels during May 1, 2, 7 and May 8, 9, 12, 14.

The empirical probability density functions (PDFs) were estimated at 20 Hz and 400 Hz (Figs. 2.10(a) and Fig. 2.10(b), see Sec. 2.2.1 for details). At 20 Hz, the variation in the median

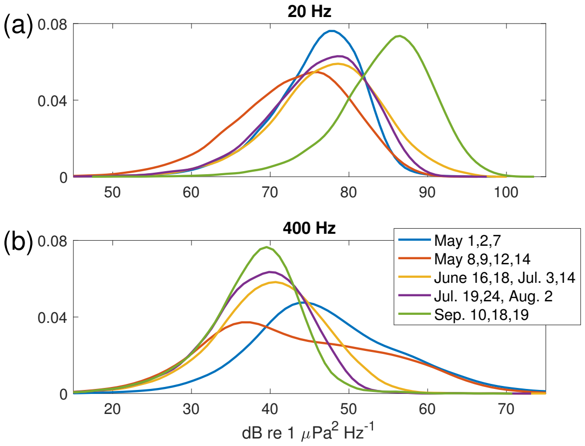

**Figure 2.10**: Empirical probability density functions (PDFs) estimated for the three and four day periods in summer 2013 at (a) 20 Hz and (b) 400 Hz. The 20 Hz estimate is predominantly effected by presence and strength of airgun pulse noise while the 400 Hz estimate corresponds to transient ice noises.

power level corresponds to changes in the received level of seismic airgun noise. May 8, 9, 12, 14 also exhibits a broader distribution as a result of the lull in airgun noise during this period (Fig. 2.10(a)). At 400 Hz, the distributions for May 1, 2, 7 and May 8, 9, 12, 14 are highly non–Gaussian as a result of numerous, loud transient ice noise events (Fig. 2.10(b)). During the remaining periods, ice noises were received at lower and more consistent power levels, resulting in more peaked distributions.

Median estimates for all hydrophones on the array show that the effect of noise sources is consistent with depth. At 20 Hz (Fig. 2.11(a)) the median estimates in depth reflect the shapes of the first and second mode (see Sec. 2.3.1, Fig. 2.4). The 400 Hz median estimates are nearly constant in depth (Fig. 2.11(b)), with the May 1, 2, 7 and May 8, 9, 12, 14 estimates at elevated power levels. Increased power levels below 300 m at both frequencies (Fig. 2.11) may be evidence that the effort to eliminate flow–related noise artifacts was not completely successful for all hydrophones and periods.

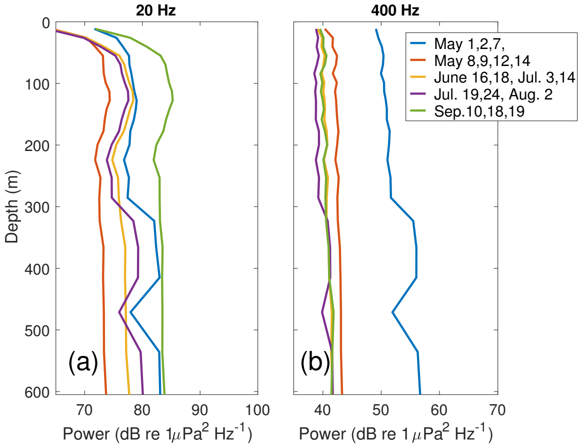

**Figure 2.11**: Depth dependence of median spectral power for three and four day periods in summer 2013 for (a) 20 Hz and (b) 400 Hz indicates that the effect of noise sources is consistent with depth.

### 2.4.2 Comparison of Arctic Ambient Noise

The median spectral power across the period including April 30, May 1, 2, 7–9, 12, and 14 at 84.5 m depth are compared with historical estimates from both western and eastern Arctic stations in Fig. 2.12. The estimated median spectral power for May 2013 was below, but similarly structured to, a composite spectral estimate from April 1982 (Fig. 2.12). [10] The peak at 15 Hz appears less prominent at lower frequencies in 2013 than in 1982. In comparison, a spectral estimate recorded in the Beaufort Sea in April 1975 shows comparable ambient noise levels and structure to 2013 but does not extend to lower frequencies (Fig. 2.12). [8] The differences in these spectra may be caused by environmental factors or by experimental factors, including recording length and post–processing methods, which were not published alongside the 1982 results.

Fig. 2.13 demonstrates the wide variability in Arctic ambient noise estimates across frequency, year, and study. This variability arises from a complex relationship between the Arctic ambient noise and both environmental and anthropogenic factors, such as sea ice percent cover, sea ice age/thickness, barometric conditions and wind patterns, local subsurface currents, seismic survey activity, and marine biologic activity. The studies shown indicate that, without correction for environmental factors, there is not a significant trend in the Arctic ambient noise power levels

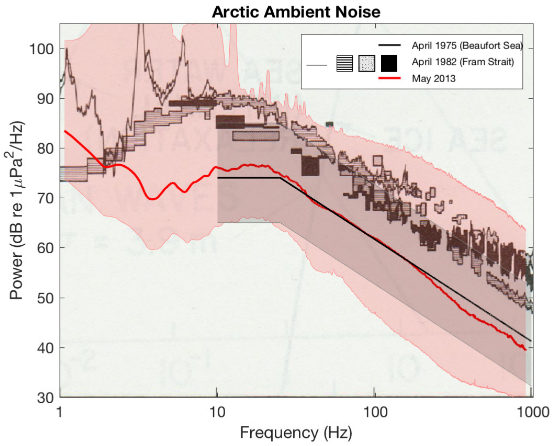

**Figure 2.12**: Median spectral estimate for April 1975 Polar Research Laboratory (Beaufort Sea, depth not published), [8] April 1982 (FRAM IV, Beaufort Sea) [10] at 99 m depth, and May 2013 at 84.5 m depth (see Table 2.2). The FRAM IV data were taken at various times and averaged over different time periods and frequencies, to represent the primary generation mechanisms: cable strum (line, $<$10 Hz), ongoing ice cracking events (striped boxes), and transient ice cracking due to ice cooling (dotted and black boxes). [10] The 10% and 90% spectral levels for May 2013 are shaded; 5% and 95% are given in a smaller shaded region for April 1975.

between 1960 and 2013, but that frequency–dependent ambient noise levels are within a 30–40 dB range for both regions of the ice covered Arctic.

## 2.5 Conclusions

Between April and September 2013, a twenty–two element vertical hydrophone array recorded the eastern Arctic ambient noise for 108 min/day while drifting between 89°N, 62°W and Svalbard.

These data were processed into spectrograms and a number of noise sources were observed, including ice noise, bowhead whale calling, airgun survey pulses, and earthquake $T$–phases. The bowhead whale calls were received between 86 and 87° N in June and July.

The data were also processed into three and four day median spectral estimates. The spectral estimates and corresponding PDFs demonstrate the variation in the occurrence and

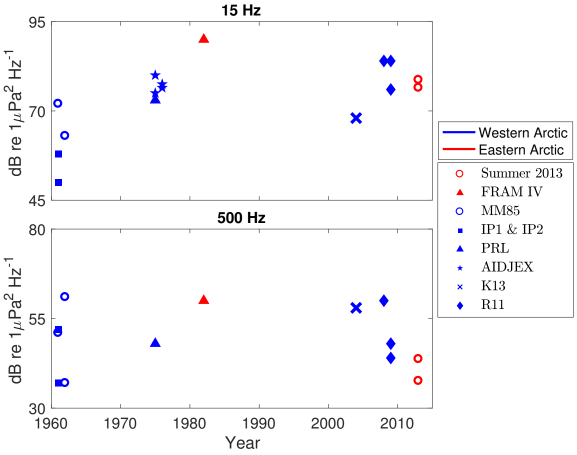

**Figure 2.13**: Scatter plot of median ambient noise level results for 15 Hz and 500 Hz from various studies in both the eastern and western Arctic (see Table 2.2).

received level of seismic airgun survey pulses at low frequencies and ice transients at high frequencies.

The median spectral estimate for May 2013 was compared to historical power spectral estimates, one recorded in a nearby region in April 1982 [10] and another from an ice–covered region in the Beaufort Sea in April 1975. [8] The May 2013 estimate is below the 1982 estimate but close to the 1975 estimate, indicating that local ice source effects may be as significant as regional effects in determining ambient noise levels in the Arctic. A multi–decadal summary of Arctic ambient noise studies displays a lack of change in power levels with time and further demonstrates the variability in Arctic ambient noise level estimates resulting from local experimental variations.

## 2.6 Acknowledgments

We would like to thank John Colosi for providing the MicroCAT data, John Kemp and the WHOI Mooring Operations and Engineering Group for their assistance, and Hanne Sagen and the Norwegian Coast Guard for assistance in recovering the array mooring. This work is supported by the Office of Naval Research under Award Numbers N00014-13-1-0632 and N00014-12-1-0226. Any opinions, findings, and conclusions or recommendations expressed in this publication are

**Table 2.2**: Ambient noise noise level estimates in the Arctic Ocean.

| Location (lat, lon) | Experiment | Dates | 15 Hz | 50 Hz | 100 Hz | 500 Hz | 1 kHz |
|---|---|---|---|---|---|---|---|
| 86°N 56.9°W – 89°N 1°E | May–June 2013 | 05/2013 – 06/2013 | 76.5 | 66 | 60.2 | 43.7 | - |
| 86°N 1.3°E – 83.8°N 4.5°E | July–Sep. 2013 | 07/2013 – 09/2013 | 78.7 | 64.9 | 55.6 | 37.6 | - |
| 83°N 20°E | [10] | 04/1982 | 90 | 79.5 | 73 | 60 | 53 |
| 82°N 168°E | (MM85) [32] | 09–10/1961 | 72 | 70 | 61 | 51 | 40 |
| 75°N 168°W |  | 05–09/1962 | 63 | 64 | 49 | 37 | 32 |
|  |  |  | - | 75 | 72 | 61 | 52 |
| 78.5°N 105.25°W | (IP1) [33] | 27/04/1961 | 50 | 42 | 38 | 37 | 20 |
|  |  | 28/04/1961 | 58 | 52 | 51 | 52 | 51 |
| 74.5°N 115.1°W | (IP2) [33] | 9/2–3/1961 | - | 57 | 56 | 52 | 43 |
| Beaufort Sea | PRL [34] | April 1975 | 73 | 68 | 62 | 48 | 43 |
|  |  |  | (10 Hz) | (32 Hz) |  |  |  |
| ∼72°N 142°W | [19] | 08/1975 | 65–85 | 65–75 | - | - | 38–55 |
|  |  | 11/1975 | 70–90 | 65–88 | - | - | 40–70 |
|  |  | 02/1976 | 65–90 | 60–90 | - | - | 35–70 |
|  |  | 05/1976 | 65–88 | 60–90 | - | - | 37–68 |
| 71°N 126.07°W | (K13) [12] | 11/2004 – 06/2005 | 68 | 69 | 66 | 58 | 54 |
| 72.46°N 157.4°W | (R11) [17] | 09/2008 | 84 | 80 | 74 | 60 | 56 |
|  |  | 03/2009 | 84 | 70 | 62 | 48 | 48 |
|  |  | 05/2009 | 76 | 61 | 56 | 44 | 44 |

those of the authors and do not necessarily reflect the views of the Office of Naval Research.

The text of Chapter Two is in full a reprint of the material as it appears in Emma Ozanich, Peter Gerstoft, Peter F. Worcester, Matthew A. Dzieciuch, and Aaron Thode, “Eastern Arctic ambient noise recorded on a drifting vertical array,” *Journal of the Acoustical Society of America*, 142(3):1997–2006, 2017. The dissertation author was the primary researcher and author of Chapter Two. The coauthors listed in this publication directed and supervised the research.

# Bibliography

[1] S. E. Moore and R. R. Reeves, “Distribution and movement,,” in *The bowhead whale*, edited by T. S. for Marine Mammalogy (Allen Press, Lawrence, KS, 1993), pp. 313–386, spec. Publ 2.

[2] R. Lindsay and A. Schweiger, “Arctic sea ice thickness loss determined using subsurface, aircraft, and satellite observations,” T.C **9**, 269–283 (2015).

[3] I. Dyer, “The song of sea ice and other arctic ocean melodies,” in *Arctic Technology and Policy* (I. Dyer and C. Chryssostomidis (Hemisphere, New York, 1984), pp. 11–37.

[4] W. C. Cummings, O. I. Diachok, and J. D. Shaffer, “Acoustic transients of the marginal sea ice zone: A provisional catalog,” Naval Research Laboratory Memorandum Report 6408 **2141** (1989).

[5] J. G. Veitch and A. R. Wilks, “A characterization of arctic undersea noise,” J. Acoust. Soc. Am **77**, 989–999 (1985).

[6] G. B. Kinda, Y. Simard, C. Gervaise, J. I. Mars, and L. Fortier, “Arctic underwater noise transients from sea ice deformation: Characteristics, annual time series, and forcing in beaufort sea,” J. Acoust. Soc. Am **138**, 2034–2045 (2015).

[7] O. I. Diachok, “Effects of sea ice ridges on sound propagation in the arctic ocean,” J. Acoust. Soc. Am **59**, 1110–1120 (1976).

[8] B. M. Buck and J. H. Wilson, “Nearfield noise measurements from an arctic pressure ridge,” J. Acoust. Soc. Am **80**, 256–264 (1986).

[9] C. R. Greene and B. M. Buck, “Arctic ocean ambient noise,” J. Acoust. Soc. Am **36**, 1218 (1964).

[10] N. C. Makris and I. Dyer, “Environmental correlates of pack ice noise,” J. Acoust. Soc. Am **79**, 1434–1440 (1986).

[11] N. C. Makris and I. Dyer, “Environmental correlates of arctic ice-edge noise,” J. Acoust. Soc. Am **90**, 3288–3298 (1991).

[12] G. B. Kinda, Y. Simard, C. Gervaise, J. I. Mars, and L. Fortier, “Under-ice ambient noise in the eastern beaufort sea, canadian arctic, and its relation to environmental forcing,” J. Acoust. Soc. Am **134**, 77–87 (2013).

[13] O. I. Diachok and R. S. Winokur, “Spatial variability of underwater ambient noise at the arctic ice-water boundary,” J. Acoust. Soc. Am **55**, 750–753 (1974).

[14] F. Geyer, H. Sagen, G. Hope, M. Babiker, and P. F. Worcester, “Identification and quantification of soundscape components in the marginal ice zone,” J. Acoust. Soc. Am **139**, 1873–1885 (2016).

[15] O. Diachok, “Arctic hydroacoustics,” Cold Reg. Sci. Technol **2**, 186–200 (1980).

[16] D. Tollefsen and H. Sagen, “Seismic exploration noise reduction in the marginal ice zone,” J. Acoust. Soc. Am **136**, L47–52 (2014).

[17] E. H. Roth, J. A. Hildebrand, S. M. Wiggins, and D. Ross, "Underwater ambient noise on the chuckchi sea continental slope from 2006-2009," J. Acoust. Soc. Am **131**, 104–110 (2012).

[18] C. L. Berchok, P. J. Clapham, J. Crance, S. E. Moore, J. Napp, J. Overland, M. Wang, P. Stabeno, M. Guerra, and C. Clark, "Passive acoustic detection and monitoring of endangered whales in the arctic (beaufort, chukchi) and ecosystem observations in the chukchi sea: Biophysical moorings and climate modeling," , Annual Report contract M09PC00016 (AKC 083), Bureau of Ocean Energy Management, Regulation, and Enforcement, Anchorage, Alaska (2012).

[19] J. K. Lewis and W. W. Denner, "Arctic ambient noise in the beaufort sea: Seasonal space and time scales," J. Acoust. Soc. Am **82**, 988–997 (1987).

[20] G. Spreen, L. Kaleschke, and G. Heygster, "Sea ice remote sensing using amsr-e 89 ghz channels," J. Geophys. Res. **113**, C02S03 (2008).

[21] P. N. Mikhalevsky, "Acoustics, arctic," in *Encyclopedia of Ocean Sciences*, edited by J. H. Steele, S. A. Thorpe, and K. K. Turekian (Academic Press, London, 2001), pp. 1–8.

[22] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt, *Computational Ocean Acoustics*, $2^{nd}$ ed. (Springer Science & Business Media, NewYork, 2011).

[23] Y. Xie and D. M. Farmer, "The sound of ice break-up and floe interaction," J. Acoust. Soc. Am **91**, 2215–2231 (1992).

[24] P. J. Stein, "Interpretation of a few ice event transients," J. Acoust. Soc. Am **83**, 617–622 (1988).

[25] S. B. Blackwell, S. C. Nations, T. L. McDonald, A. M. Thode, D. Mathias, K. H. Kim, J. C. R. Greene, and A. M. Macrander, "The effects of airgun sounds on bowhead whale calling rates: evidence for two behavioral thresholds," PLoS One 10 e0125720 (2015).

[26] A. M. Thode, S. B. Blackwell, K. D. Seger, and A. S. Conrad, "Source level and calling depth distributions of migrating bowhead whale calls in the shallow beaufort sea," J. Acoust. Soc. Am **140**, 4288–4297 (2016).

[27] K. M. Stafford, S. E. Moore, C. L. Berchok, O. Wiig, C. Lydersen, E. Hansen, D. Kalmbach, and K. Kovacs, "Spitsbergen's endangered bowhead whales sing through the polar night," Endanger. Species Res. **18**, 95–103 (2012).

[28] S. H. Ferguson, L. Dueck, L. L. Loseto, and S. P. Luque, "Bowhead whale (balaena mysticetus) seasonal selection of sea ice," Mar. Ecol. Prof. Ser. **411**, 285–297 (2010).

[29] S. E. Moore, K. M. Stafford, H. Melling, C. Berchok, O. Wiig, K. Kovacs, C. Lydersen, and J. Richter-Menge, "Comparing marine mammal acoustic habitats in atlantic and pacific sectors of the high arctic: year–long records from fram strait and the chukchi plateau," Polar Biol. **35**, 475–480 (2012).

[30] K. LePage and H. Schmidt, “Modeling of low–frequency transmission loss in the central arctic,” J. Acoust. Soc. Am **96**, 1783–1795 (1994).

[31] R. A. Sohn and J. A. Hildebrand, “Hydroacoustic earthquake detection in the arctic basin with the spinnaker array,” B. Seismol. Soc. Am **91**, 572–579 (2001).

[32] R. H. Mellen and H. W. Marsh, *Underwater sound in the Arctic Ocean* (U.S. Navy Underwater Sound Laboratory, New London, Connecticut, 1965), accession Number AD718140.

[33] A. R. Milne and J. H. Ganton, “Ambient noise under arctic sea ice,” J. Acoust. Soc. Am **36**, 855–863 (1964).

[34] B. Buck, “Preliminary under–ice propagation models based on synoptic ice roughness,” , PRL TR–30 Seattle, WA (1981).

# Chapter 3

# Unsupervised clustering of coral reef bioacoustics

An unsupervised process is described for clustering automatic detections in an acoustically active coral reef soundscape. First, acoustic metrics were extracted from spectrograms and timeseries of each detection based on observed properties of signal types and classified using unsupservised clustering methods. Then, deep embedded clustering (DEC)was applied to fixed-length power spectrograms of each detection to learn features and clusters. The clustering methods were compared on simulated bioacoustic signals for fish calls and whale song units with randomly varied signal parameters and additive white noise. Overlap and density of the handpicked features led to reduced accuracy for unsupervised clustering methods. DEC clustering identified clusters with fish calls, whale song, and events with simultaneous fish calls and whale song, but accuracy was reduced when the class sizes were imbalanced. Both clustering approaches were applied to acoustic events detected on directional autonomous seafloor acoustic recorder (DASAR) sensors on a Hawaiian coral reef in February–March 2020. Unsupervised clustering of handpicked features did not distinguish fish calls from whale song. DEC had high recall and correctly classified a majority of whale song. Manual labels indicated a class imbalance between

fish calls and whale song at a 3-to-1 ratio, likely leading to reduced DEC clustering accuracy.

Machine learning has become commonly used within the acoustics community and the ocean bioacoustics community in particular (Sec VIII in Ref. [4]) where automatic detection and classification methods have been under development for marine mammals calls for decades and continue to expand. [3, 8, 15, 31, 32, 34–36, 38] Recently, a smaller set of studies have also considered unsupervised machine learning techniques for analyzing large, unlabelled bioacoustic soundscape data. [10, 11, 20, 39]

Acoustic classification of marine fishes, such as damselfish (family *Pomacentridae*), has been improved through passive acoustic field experiments that have characterized the calls and calling behavior [25, 29, 42] but lacks established terminology across studies and a universally accepted correspondence between calls and behavior. [2] Recently, a few studies have considered automatic classification of fish calls by utilizing machine learning tools. Malfante et al. 2018 extracted time, spectral, and ceptstral features for use in supervised classifiers of fish calls in a seagrass meadow. Based on four call types defined by the authors, the machine learning classifiers achieved up to 95% accuracy. [24] Lin et al. 2018 compared the detection performance of a rule-based energy detector to the machine learning methods of periodicity-coded nonnegative matrix factorization and Gaussian mixture models. The machine learning methods were applied to power spectra of croaker calls recorded in shallow water (10–25 m) and then qualitatively compared to the energy detector results. [22] Then, Ibrahim et al. 2018 compared long short-term neural networks (LSTM) and convolutional neural networks (CNN) for supervised classification of grouper croaks in the time-frequency domain. They found that machine learning outperformed weighted mel-freqeuncy cepstral coefficients, with LSTM achieving over 90% correct classification accuracy on all species tested. [17]

In this paper, the problem of identifying bioacoustic signals in an acoustically active coral reef is addressed using unsupervised machine learning. We consider two approaches for extracting features for clustering:

1. Spectral and time domain features were manually chosen, or handpicked, based on their observed relation to coral reef fish calling and on studies of fish calling spectral and temporal properties. [25, 29, 42]
2. Fixed-length spectrograms were used in a deep embedded clustering (DEC) algorithm, [44] a deep–learning image compression algorithm that ensures accurate image reconstruction from the latent feature vector while encouraging cluster formation among the latent features. [13, 37]

For handpicked features, the statistical properties of different features may vary, and the features are not guaranteed to be separable by unsupervised algorithms. The DEC algorithm aims to address feature separability by jointly learning the features and clusters from the spectrogram. However, DEC on a single-channel spectrogram may have reduced performance when signals overlap or if there are is consistently low signal-to-noise ratio (SNR) within the training data.

Simulated signals were used to compare the limitations of the handpicked feature clustering and DEC. The signals were designed to mimic whale song and fish call pulses recorded on a Hawaiian coral reef, with the SNR and call parameters randomly varied to simulate experimental variation. Accuracy, recall, and precision were used to measure the classification success of both methods. Then, using transient events detected from an automatic directional detector, [40] both methods were applied to acoustic data recorded in February–March 2020.

In Sec. 3.1, unsupervised clustering theory is overviewed for Gaussian-distributed features with known mean and covariance. A method for visualizing high-dimensional data is also discussed. The handpicked features and their extraction procedure are covered in Sec. 3.2A and the DEC theory is reviewed in Sec. 3.2B. Section 3.3 details clustering results for simulated coral reef bioacoustic signals: fish calls and whale song units, represented by Gaussian pulses and frequency-modulated (FM) sweeps. Experimental data collection from a Hawaiian coral reef in Februrary 2020 and the directional detection algorithm are outlined in Sec. 3.4. Last, Sec. 3.5 presents the experimental detection and clustering results. Section 3.6 summarizes the approach and discusses challenges associated with the methods and dataset.

## 3.1 Unsupervised clustering

Unsupervised clustering methods are frameworks for categorizing data according to their similarities. [5] The performance of each algorithm depends on the validity of its underlying assumptions for a given feature set. This section discusses the maximum likelihood class boundaries for Gaussian clusters, then describes the K-means and hierarchical agglomerative clustering algorithms.

### 3.1.1 Maximum likelihood of Gaussian clusters

When the generative distribution of the data are known, an exact solution for the optimal clusters can be derived. Assume a $P$–dimensional vector, $\mathbf{x}_n \in \mathbb{R}^P$, is drawn from one of $K$ Gaussian distributions with mean $\boldsymbol{\mu}_k \in \mathbb{R}^P$ and covariance $\boldsymbol{\Sigma}_k \in \mathbb{R}^{P\times P}$. The analytic solution to the cluster boundaries can be computed using the posterior given by Bayes' theorem, [6]

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_j p(\mathbf{x}|C_j)p(C_j)} \tag{3.1}$$

$$p(\mathbf{x}|C_k) = \frac{1}{(2\pi)^{\frac{P}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)}, \tag{3.2}$$

Where $C_k$ and $C_j$ represent two class labels and each point $\mathbf{x}_n$ belongs to only one class. The optimal boundary between the two classes occurs when the probability of the classes are equal,

$$\begin{gathered} \log p(C_k|\mathbf{x}) = \log p(C_j|\mathbf{x}) \qquad (3.3) \\ \log p(\mathbf{x}|C_k) + \log C_k = \log p(\mathbf{x}|C_j) + \log C_j. \end{gathered}$$

Combining (3.2) and (3.3) and setting equal to zero:

$$0 = -\frac{1}{2}\mathbf{x}^T(\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\Sigma}_j^{-1})\mathbf{x} + \mathbf{w}^T\mathbf{x} + \frac{C}{2} \tag{3.4}$$

$$C = -\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k + \boldsymbol{\mu}_j^T\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\mu}_j - \log\frac{|\boldsymbol{\Sigma}_k|}{|\boldsymbol{\Sigma}_j|} + 2\log\frac{C_k}{C_j} \tag{3.5}$$

$$\mathbf{w} = \boldsymbol{\mu}_k\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\mu}_j\boldsymbol{\Sigma}_j^{-1} \tag{3.6}$$

The general solution (5.14) is a $P$–dimensional parabola, which simplifies to a linear boundary if the distributions have a shared covariance such that $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}_j \ \forall j,k$. In a similar manner, (5.14) is extensible to $K > 2$ classes. [6]

When $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\mu}_k$ are unknown, the Expectation-Maximization (EM) algorithm can be used to estimate them from a set of data $\mathbf{x}$, $n = 1,\ldots,N$ via the complete data log-likelihood. [33]

This can be solved using an alternating algorithm to update the weighted posterior probability (responsibility) and the class prior, mean, and covariance [33]

$$\text{E step:} \quad r_{nk} = \frac{\pi_k p(\mathbf{x}_n|C_k^{t-1})}{\sum_{k'}\pi_{k'}p(\mathbf{x}_n|C_{k'}^{t-1})} \tag{3.7}$$

$$\text{M step:} \quad \pi_k = \frac{1}{N}\sum_n r_{nk} \tag{3.8}$$

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}, \quad \boldsymbol{\Sigma}_k = \frac{\sum_n r_{nk}\mathbf{x}_n\mathbf{x}_n^T}{\sum_n r_{nk}} \tag{3.9}$$

where $C_k^{t-1}$ represents the $k$th cluster at step $t-1$.

EM is iterated until both steps converge for $K$ classes. EM requires that the number of clusters, $K$, be assumed *a priori*, and the estimated covariance matrix can become ill-conditioned if there are fewer than $K$ clusters.

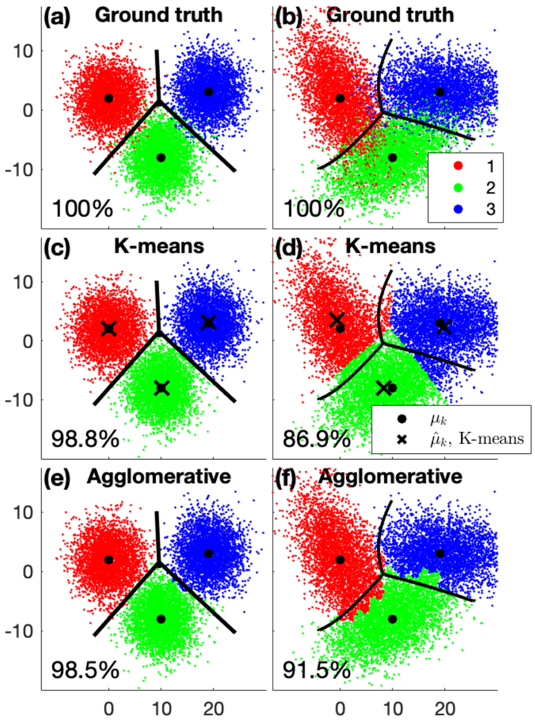

**Figure 3.1**: Clustering on two data distributions using *(c,d)* K-means and *(e,f)* agglomerative clustering with Ward's method, with maximum likelihood boundaries shown by black lines. In the first dataset in *(a)* , the clusters have shared covariances of the form $\sigma^2\mathbf{I}$ ( $\sigma_x^2 = \sigma_y^2 = 3$). The clusters of the second dataset *(b)* each have different covariances resulting from cluster rotations of $\theta = 120°, 25°$, and $0°$ counterclockwise ($\sigma_x^2 = 6$, $\sigma_y^2 = 3$.)

### 3.1.2 K-Means

K-means [16] is an approximation to the EM algorithm that partitions $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ into the $K$ clusters. The K-means algorithm requires that $K$ is set by the practitioner and assumes that all clusters are Gaussians with covariance $\mathbf{\Sigma} = \sigma^2\mathbb{I}$ and prior probability $\pi_k = 1/K$, where $\mathbb{I}$ is the identity matrix.

K-means is also called hard EM because it assigns each point to a cluster rather than computing the cluster likelihood. The EM steps are simplified using the K-means assumptions to solve for the optimal clusters,: [6]

$$1. \quad r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise,} \end{cases} , \forall n \tag{3.10}$$

$$2. \quad \boldsymbol{\mu}_k = \frac{1}{|C_k|} \sum_{\mathbf{x}_n \in C_k} \mathbf{x}_n = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}} \tag{3.11}$$

where $C_k$ is the $k$th cluster and $|C_k|$ denotes its cardinality or size.

K-means will result in the maximum likelihood solution in (5.14) when the classes are Gaussian with shared covariance, $\boldsymbol{\Sigma}_k = \sigma^2\mathbb{I}\ \forall k$. [6] If the true number of classes differs from $K$, the classes will be incorrectly estimated. In the following, an underlying knowledge of the signal content is assumed for specifying $K$.

### 3.1.3 Agglomerative hierarchical clustering

Agglomerative hierarchical clustering, [5, 33] also called bottom-up clustering, partitions a set of $N$ data points, $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, into $K$ clusters by grouping the most similar data at each step. Hierarchical clustering successively merges nearby clusters until the stop criterion is achieved. In this case, the stop criterion is met when $K$ or fewer classes remain, where $K$ must be set by the practitioner.

Agglomerative methods preferentially cluster dense points, improving robustness to outliers. The tradeoff is decreased performance when clusters are dense and close. A related bottom-up approach has been demonstrated to work well for clustering dolphin echolocation clicks. [10, 11]

To initialize from bottom-up, each point begins as its own cluster, $C_{k_0} = \mathbf{x}_n$, $k_0 = 1, \ldots, N$.

Then, clusters that satisfy the minimum distance requirement are merged

$$j,k = \arg\min_{i,i'} d(i,i') \tag{3.12}$$

$$C_{k_1} = \{C_j \cup C_k\}, \tag{3.13}$$

where $d(i,i')$ is the distance between clusters $i$ and $i'$. The agglomerative process is repeated $M$ times, until there are at most $K$ clusters remaining with $C_{k_M}$, $k_M = 1,\ldots,K$.

The distance metric chosen here is *Ward's method.* [18] Ward's method measures the within-cluster variance of two merged clusters. The variance introduced by merging two sub-clusters, $C_1$ and $C_2$ with means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, is measured by the increase in the incremental sum-of-squares, [30]

$$\begin{aligned} d(C_1,C_2) &= \sum_{i\in(C_1\cup C_2)} (\mathbf{x}_i - (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2))^2 - \sum_{j\in C_1} (\mathbf{x}_j - \boldsymbol{\mu}_1)^2 - \sum_{k\in C_2} (\mathbf{x}_k - \boldsymbol{\mu}_2)^2 \\ &= (|C_1| + |C_2|)(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)^2 - |C_1|\boldsymbol{\mu}_1^2 - |C_2|\boldsymbol{\mu}_2^2 \\ &= \frac{2|C_1||C_2|}{|C_1| + |C_2|} \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2^2. \end{aligned} \tag{3.14}$$

where $\boldsymbol{\mu}_1 = \frac{1}{|C_1|}\sum_{j\in C_1}\mathbf{x}_j$ and $\boldsymbol{\mu}_2 = \frac{1}{|C_2|}\sum_{k\in C_2}\mathbf{x}_k$. $|C_1|$ and $|C_2|$ are the cardinality of cluster $C_1$ and $C_2$. In practice, $\sqrt{d}$ from (3.14) was used.

If the number of true classes differs from $K$, the final output may be misinterpreted. However, the hierarchical agglomerative clustering cost function is agnostic of $K$, and the history of clusters can be retrieved to improve understanding of the feature similarities.

### 3.1.4 Clustering simulations

Three 2D Gaussian distributions, each with $N = 6,666$ points, were used to simulate overlapping clusters (Fig. 3.1). The true cluster means were $\boldsymbol{\mu}_1$ = (0,2), $\boldsymbol{\mu}_2$ = (10,-8), and

$\boldsymbol{\mu}_3 = (21,3)$. The covariance of the first dataset was

$$\Sigma_{C_1} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{3.15}$$

with $\sigma_x^2 = \sigma_y^2 = 3$. For the first dataset (Fig. 3.1a), there were no off-diagonal covariance terms.

The second dataset (Fig. 3.1b) was generated by rotating the data counterclockwise at $\theta$, with

$$\Sigma_{C_2} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}^T \tag{3.16}$$

with $\sigma_x^2 = 6$, $\sigma_y^2 = 3$. The three clusters were rotated by $\theta = 120°, 25°$, and $0°$. Off-diagonal covariance terms were introduced by the rotation. In realistic data, a strongly rotated cluster such as Class 1 may represent two highly correlated features.

The K-means algorithm assumes that clusters are spatially distributed around a mean with diagonal covariance. Thus, K-means performs best for the first dataset with identical, diagonal covariance for all clusters. By incorporating intercluster distance, the Ward metric is also able to identify 3 classes but is susceptible to misclassifying data where dense clusters overlap. When the cluster covariances are not of the form $\sigma^2\mathbb{I}$, K-means is no longer a valid approximation to the maximum likelihood solution.

### 3.1.5 Visualization of high-dimensional data

For data with more than two dimensions, $\mathbf{x}_n \in \mathbb{R}^P$ for $P > 2$, clusters may be visualized by applying dimensionality reduction. In this study, 2D t-Stochastic Neighbor Embedding (t-SNE) [14] was used to visualize $P$–dimensional features.

The similarity of one point, $\mathbf{x}_i \in \mathbb{R}^P$, to another point, $\mathbf{x}_j \in \mathbb{R}^P$, is found by computing the

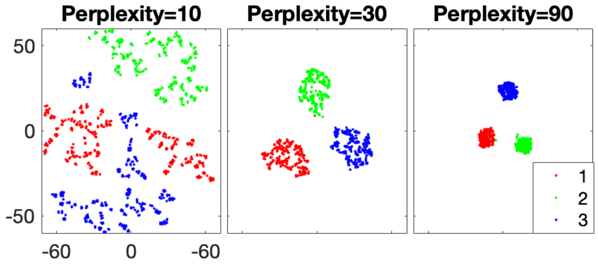

**Figure 3.2**: Varying values of perplexity for t-SNE on $N = 1000$ randomly drawn points $\mathbf{x}_n \in \mathbb{R}^3$, $n = 1, \ldots, N$. The default perplexity value was 30.

conditional probability that the points will be neighbors within a Gaussian density centered at $\mathbf{x}_i$, [14]

$$p_{j|i} = \frac{e^{-\frac{1}{2\sigma_i^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2}}{\sum_{k \neq l} e^{-\frac{1}{2\sigma_i^2}\|\mathbf{x}_k - \mathbf{x}_l\|^2}}, \quad p_{i|i} = 0 \tag{3.17}$$

where $i, j$=$1, \ldots, N$. The neighborhood of $\mathbf{x}_i$, as determined by $\sigma_i$, $i = 1, \ldots, N$ is defined implicitly in terms of the perplexity (Fig. 3.2), [14, 23]

$$\text{perplexity}(P_i) = 2^{H(P_i)} \tag{3.18}$$

$$H(P_i) = -\sum_{j=1}^{N} p_{j|i} \log_2 p_{j|i} \tag{3.19}$$

where $P_i = \sum_j p_{j|i}$, and $H$ is the Shannon entropy. The optimal value of $\sigma_i$ in (3.17) for each point is solved with a binary search for a given value of perplexity. [23]

Then, a set of two-dimensional point projections is randomly initialized with zero-mean Gaussians of low variance, [14] $\mathbf{y}_i \in \mathcal{N}(\mathbf{0}, 10^{-4}\mathbb{I})$. The optimal point projections are found by minimizing the Kullback-Leibler (KL) divergence between the original distribution and the

Student's t-distribution of the proposed data,

$$q_{j|i} = \frac{(1+\|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l}(1+\|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}, \quad q_{i|i} = 0 \tag{3.20}$$

$$KL(P\|Q) = -\sum_{i \neq j}\sum_{j=1}^{N} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}. \tag{3.21}$$

In contrast to classic SNE which uses only Gaussians, t-SNE's use of the Student t-distribution further penalizes outliers. [14]

As shown in Fig. 3.2, the value of perplexity should be varied according to user preference to obtain the desired visualization. Although the default perplexity value in the MATLAB implementation [23] is 30, larger datasets require higher perplexity.

## 3.2 Feature extraction

Feature extraction may be considered the most important step for unsupervised clustering, as the feature properties determine cluster performance. Here, we examine two feature extractions approaches: handpicked features and deep embedded clustering. The handpicked feature vectors were stacked to form a feature matrix to be used with unsupervised clustering methods. Deep embedded clustering jointly learns and clusters feature vectors using a convolutional autoencoder neural network. [44]

### 3.2.1 Handpicked features

The handpicked features (Table 3.1, Fig. 3.3) are time-frequency properties known to relate to fish call type including power, duration, and peak frequency [25, 29, 42] as well as timeseries estimates of impulsive noise. [24, 28] The spectrogram parameters were 256–point FFT with 90% overlap, d$t$=0.0256 s, and d$f$=3.9 Hz.

If $t_1$ and $t_2$ represent the absolute start and end times for a detected event, the event

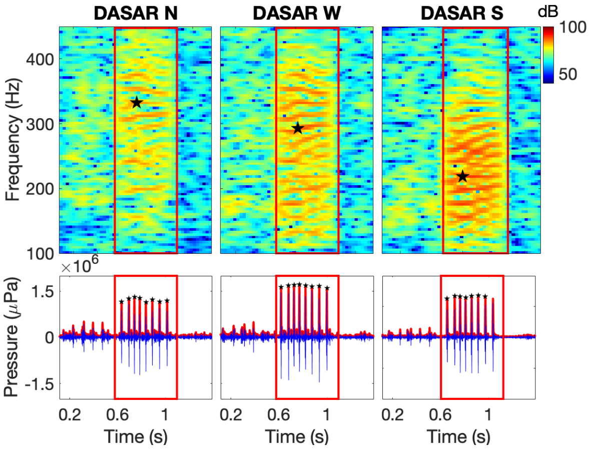

**Figure 3.3**: Handpicked features of a fish call event on February 25, 07:12 HST, measured on directional autonomous seafloor acoustic recordings (DASARs) N, W, and S. A call duration of 0.52 s was determined during the detection process. The spectrograms *(a–c)* were used to extract peak frequencies (black star) from 219–332 Hz and median PSD from 72.8–76.3 dB. The timeseries envelope *(d–f)* was used to extract the kurtosis values of 18–21, cross-sensor coherences of 0.72–0.74, and 8 temporal peaks. DASAR S was the closest to the call.

duration is

$$\text{Duration(s)} = \Delta t = t_2 - t_1. \tag{3.22}$$

Between $t_1$ and $t_2$, the event power spectrogram $|\mathbf{S}|^2 \in \mathbb{R}^{N_f \times N_t}$ was computed. Across the power spectrogram, the median power and peak frequency were extracted (Fig. 3.3a–c), with

$$\begin{aligned} \text{Median PSD} &= \underset{i,j}{\text{median}}\, S^{(i,j)} \\ \text{Peak freq.} &= \arg\max_i \left( \max_j S^{(i,j)} \right), \\ & \qquad i = 1, \ldots, N_f, \quad j = 1, \ldots, N_t. \end{aligned}$$

In simulation, the duration was fixed and the signal-to-noise ratio was randomized. Neither feature was used for clustering.

The pressure timeseries $\mathbf{y} \in \mathbb{R}^N$ was extracted between $t_1$ and $t_2$ (Fig. 3.3). For the

**Table 3.1**: Event features estimated for automatically detected pulse sounds.

| Feature Name (units) | Description |
|---|---|
| Kurtosis | Fourth moment normalized by the squared variance |
| Npeaks (count) | Number of peaks with at least 5 dB prominence *re* the standard deviation |
| Peak frequency (Hz) | Frequency of the peak power spectral density |
| *Features for experimental data only* | |
| Duration (s) | Length of detected event |
| Coherence | Normalized time coherence between DASARs |
| Median PSD (dB) | Median power spectral density (PSD) across event mask |

experimental data, the vector sensor x– and y– velocity channels, $\mathbf{v}_x$ and $\mathbf{v}_y$, were used to create a beamformed pressure timeseries, $\mathbf{y}_b \in \mathbb{R}^N$, for improved detection SNR,

$$\mathbf{y}_b = \mathbf{y} + Z_0 \left[\mathbf{v}_x \sin(\hat{\theta}) + \mathbf{v}_y \cos(\hat{\theta})\right], \tag{3.23}$$

where $Z_0 = \rho c$ is the impedance in water with density $\rho$ ($kg \cdot m^3$) and sound speed $c$ ($m \cdot s^{-1}$), a scaling term to ensure all terms were in pressure units of $kg \cdot m^{-1} s^{-2}$ ($N \cdot m^{-2}$). [41] $\hat{\theta}$ is the estimated azimuth of the detected signal. For the simulated data, $\mathbf{v}_x = \mathbf{v}_y = \mathbf{0}$ and $\mathbf{y}_b \equiv \mathbf{y}$.

Three metrics were chosen for timeseries extraction: kurtosis, number of peaks, and cross-sensor coherence. The kurtosis is a ratio of moments and has recently been applied to the task of differentiating impulsive from non-impulsive sounds, [24, 28]

$$\text{Kurtosis} = \frac{\mu_4}{\sigma^2}, \quad \mu_4 = \frac{1}{N}\sum_{i=1}^{N}[y_b[i] - \overline{\mathbf{y}_b}]^4 \tag{3.24}$$

$\sigma^2$ is the variance and  is the arithmetic mean.

The number of peaks and cross-sensor coherence were extracted from the Hilbert transform of the beamformed pressure timeseries (Fig. 3.3)

$$\tilde{\mathbf{y}}_b = \mathcal{H}(\mathbf{y}_b)$$

where $\mathcal{H}()$ is the Hilbert transform. The number of peaks in the timeseries may be an indicator of fish species and call context for Hawaiian reef fish. [29, 42] Here, the number of peaks was defined as the number of local maxima with at least 5dB prominence relative to the standard deviation,

$$\begin{aligned} \text{Npeaks} &= \sum_{j\in(a,b)} \mathbb{I}(\max_j \tilde{y}_b[j] > C + \max\min_j \tilde{y}_b[j]), \\ C &= \sigma \cdot 10^{1/2} \end{aligned} \tag{3.25}$$

where $(a,b)$ is an interval in $\tilde{\mathbf{y}_b}$, $\sigma$ is the standard deviation, and $\mathbb{I}(x) = 1$ if $x$=True, 0 if $x$=False.

Last, for the experimental detections, the normalized correlation coefficient of the timeseries envelope across DASARs was computed to measure the spatial coherence of the signal propagation,

$$\begin{aligned} \text{Coherence} &= \max_i \frac{1}{C}\sum_{m=i}^{N-1} \tilde{y}_{b,N}[m]\tilde{y}_{b,S}[m-i] \\ C &= \sqrt{\|\tilde{\mathbf{y}}_{b,N}\|_2^2 + \|\tilde{\mathbf{y}}_{b,S}\|_2^2}, \end{aligned} \tag{3.26}$$

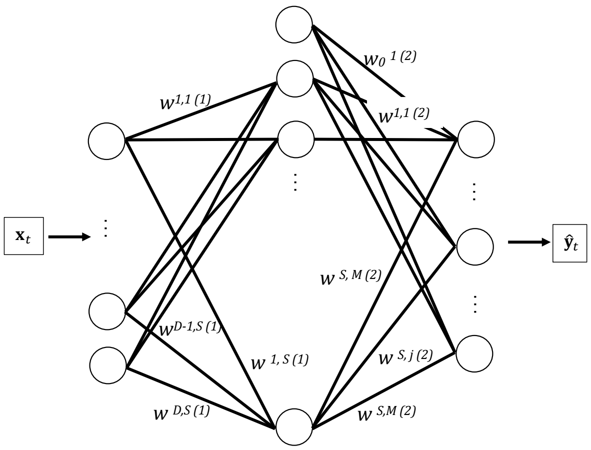

**Figure 3.4**: Architecture of the deep embedded encoding (DEC) convolutional model. Convolutional filters were sized 3x3 and used the rectified linear unit (ReLU) activation function.

where $N-1$ is the number of samples per event, which varies for each detection. $\tilde{\mathbf{y}}_{b,N}$ and $\tilde{\mathbf{y}}_{b,S}$ represent the timeseries extracted on the North and South DASARs.

### 3.2.2 Deep embedded clustering

Deep embedded clustering (DEC) is a modified convolutional autoencoder, a neural network-based feature learning method, [12] that encourages separability of its learned feature space (Fig. 3.4). The structure consists of two stacked networks: the encoder network, which maps input data into a lower-dimensional or latent space, and the decoder network, which reconstructs an approximation of the input from the latent space. Here, the architecture was compressive, with the input image downsampled by the network with 2D convolutions of stride length 2. The Rectified Linear Unit (ReLU) was used to transform the outputs at each layer,

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0. \end{cases} \tag{3.27}$$

**Table 3.2**: Network architecture used in deep embedded clustering. The input and output shapes are given as [height, width, depth]. The kernel size is the shape of the two dimensional convolutional filters in [height, width].

| Layer Name | Layer Type | Input shape | Filters | Kernel size | Stride | Activation | Output shape | Parameters |
|---|---|---|---|---|---|---|---|---|
| Conv1 | 2D Convolution | [90,20,1] | 8 | [3,3] | [2,2] | ReLU | [45,10,8] | 80 |
| | 2D Conv. | [45,10,8] | 8 | [2,1] | [1,1] | ReLU | [44,10,8] | 136 |
| Conv2 | 2D Conv. | [44,10,8] | 16 | [3,3] | [2,2] | ReLU | [22,5,16] | 1168 |
| | 2D Conv. | [22,5,16] | 16 | [1,2] | [1,1] | ReLU | [22,4,16] | 528 |
| Conv3 | 2D Conv. | [22,4,16] | 32 | [2,1] | [2,1] | ReLU | [11,4,32] | 1056 |
| | 2D Conv. | [11,4,32] | 64 | [2,1] | [1,1] | ReLU | [10,4,64] | 4160 |
| Conv4 | 2D Conv. | [10,4,64] | 64 | [2,1] | [2,1] | ReLU | [5,4,64] | 8256 |
| Flatten | Flatten | [5,4,64] | - | - | - | - | [1280] | 0 |
| Encoded | Fully Connected | [1280] | - | - | - | ReLU | [10] | 6405 |
| Dense | Fully Connected | [15] | - | - | - | ReLU | [1280] | 7680 |
| Reshape | | [1280] | - | - | - | - | [5,4,64] | 0 |
| TConv4 | Transposed Convolution | [5,4,64] | 32 | [2,1] | [2,1] | ReLu | [10,4,32] | 4128 |
| | T. Conv. | [10,4,32] | 32 | [2,1] | [1,1] | ReLu | [11,4,32] | 2080 |
| TConv3 | T. Conv. | [11,4,32] | 16 | [2,1] | [2,1] | ReLu | [22,4,16] | 1040 |
| | T. Conv. | [22,4,16] | 16 | [1,2] | [1,1] | ReLu | [22,5,16] | 528 |
| TConv2 | T. Conv. | [22,5,16] | 8 | [3,3] | [2,2] | ReLu | [44,10,8] | 1160 |
| | T. Conv. | [44,10,8] | 8 | [2,1] | [1,1] | ReLu | [45,10,8] | 136 |
| TConv1 | T. Conv. | [45,10,8] | 1 | [3,3] | [2,2] | Linear | [90,20,1] | 73 |

The use of this model for clustering of bioacoustic coral reef signals was inspired by its recent application to unlabeled seismic events. [37] Additional details of the model are given in Fig. 3.4 and Table 3.2. In this study, the input images are the same as the output labels to encourage accurate image reconstruction.

The latent feature separation is accomplished by incorporating two loss functions at different stages of training: mean squared reconstruction error (MSE) between the input and output spectrogram, and Kullback-Leibler divergence, [13, 44]

$$
\begin{aligned}
KL(P\|Q) &= \sum_{n}^{N}\sum_{k}^{K} p_{nk}\log\left(\frac{p_{nk}}{q_{nk}}\right) && (3.28)\\
q_{nk} &= \frac{(1+\|\mathbf{z}_n-\boldsymbol{\mu}_k\|^2)^{-1}}{\sum_j(1+\|\mathbf{z}_n-\boldsymbol{\mu}_j^2\|^{-1}} && (3.29)\\
p_{nk} &= \frac{q_{nk}^2/\sum_m q_{mk}}{\sum_j(q_{nj}^2/\sum_m q_{mk})}, && (3.30)
\end{aligned}
$$

where $\mathbf{z}_n \in \mathbb{R}^P$ is the latent feature vector of the $n$th spectrogram input and $\boldsymbol{\mu}_k$ is the $k$th cluster mean. (3.29) is the empirically estimated Student's t-distribution and (3.30) further penalizes points that are distant from a cluster center. [13]

Unlike supervised machine learning, training labels are not available in unsupervised learning and DEC. Instead, the DEC was trained with fixed-length spectrogram images $\log_{10}|\mathbf{S}_n|^2 \in \mathbb{R}^{N_f\times N_t}$ of 0.5 s$\,=N_t\cdot dT$ at its input and output, where events longer than 0.5 s were clipped to length. This feature-learning approach is analogous to principal component analysis.

First, the DEC (Fig. 3.4) was pretrained to learn latent features using mean squared reconstruction error loss for 1000 epochs with the Adam optimizer [19] and learning rate of $10^{-3}$. The pretrained latent features $\mathbf{z}_n$ were then clustered with K-means to initialize the deep clustering, with means $\boldsymbol{\mu}_k$, $k=1,\ldots,K$. The DEC was trained for an additional 20 epochs using the joint clustering/reconstruction loss function,

$$
L = 0.1\cdot KL + 0.9\cdot MSE, \tag{3.31}
$$

with $KL$ from (3.28).

The DEC was written in Keras [7] using Tensorflow [1].

**Table 3.3**: Signal parameters were drawn from random distributions for each simulated event.

| | FM Sweep | Pulse train |
|---|---|---|
| Duration/Width (s) | $T \in \mathcal{U}(0.2, 0.4)$ | $\tau = 0.005$ |
| Delay (s) | $t_0 \in \mathcal{U}(-0.1, 0.1)$ | $t_0 \in \mathcal{U}(-0.1, 0.1)$ |
| Frequency (Hz) | $f_0 \in \mathcal{U}(100, 400)$ | $f_c = 200$ |
| | $\Delta f \in \mathcal{U}(-150, 150)$ | |
| Peak spacing (s) | | $\Delta t \in 0.47 * beta(4, 23)$ |
| Number of peaks | | $N \in \lfloor 13 * beta(3.5, 8) \rfloor$ |
| SNR (dB) | $SNR \in \mathcal{U}(15, 30)$ | $SNR \in \mathcal{U}(0, 30)$ |

### 3.2.3 Feature matrix

An $N \times P$ feature matrix was constructed by concatenating the feature vectors for each event,

$$\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]^T \in \mathbb{R}^{N \times P}, \tag{3.32}$$

with $\mathbf{z}_n \in \mathbb{R}^P$ ($N \gg P$). The feature vectors $\mathbf{z}_n$ were automatically extracted handpicked features, with $P = 6$ (Sec. 3.2a). The clustering algorithms in Sec. 3.1 were used to find $K$ $P$–dimensional cluster means. All $N$ points were assigned to a unique cluster.

## 3.3 Simulations

A set of coral reef bioacoustic events was simulated to compare handpicked features and deep embedded clustering under varying conditions. A total of 10,000 events were simulated to mimic recorded whale song and fish pulses or pulse trains (Fig. 3.5). The signals were simulated as timeseries sampled at 1kHz, preprocessed as spectrogram images using a 256–pt FFT with 90% overlap, and then clipped to 0.5 s.

The DEC latent feature dimension was optimized over a range of values. The optimal DEC

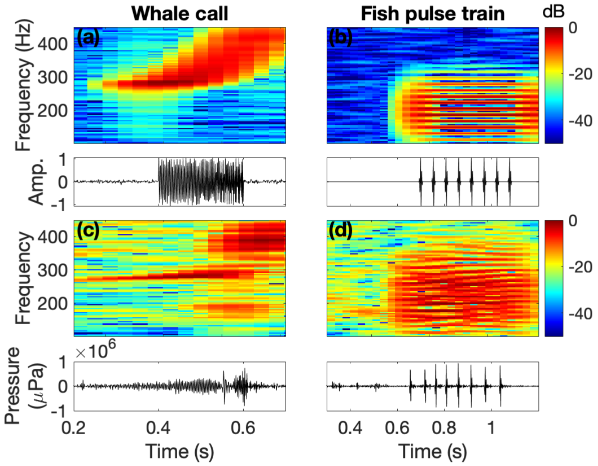

**Figure 3.5**: Simulated (a–b) and measured (c–d) coral reef bioacoustic signals before clipping. Whale calls were simulated with an FM upsweep (a,c), and fish pulse trains were simulated with superimposed Gaussian pulses (b,d). White noise was added to both signals at 25 dB SNR relative to the mean signal power.

performance was compared to unsupervised clustering with automatically extracted handpicked features. Then, a third class containing an overlapping fish call and whale song was included, with 33% of the total samples belonging to each class.

Quadratic FM sweeps mimic parts of humpack whale song. The equation for the instantaneous frequency of a quadratic sweep is [26]

$$f(t) = \beta t^2 + 2\pi f_0, \quad \beta = \frac{2\pi\Delta f}{T} \tag{3.33}$$

where $f_0$ is the initial frequency, $\Delta f$ is the total bandwidth, and $T$ is the duration of the signal. The signal is an FM upsweep when $\beta > 0$, an FM downsweep when $\beta < 0$ and a tonal when $\beta = 0$. The phase of the time domain signal can be found by integrating the instantaneous frequency [26]

$$\Phi(t) = \int_0^{t-t_0} (\beta t^2 + 2\pi f_0) dt \tag{3.34}$$

$$y(t) = \sin(\Phi(t)) = \sin(\frac{\beta}{3}(t-t_0)^3 + 2\pi f_0(t-t_0)), \tag{3.35}$$

where $t_0$ is the time delay of the signal start.

Timeseries of impulses, or pulse trains, were used to simulate fish calls. The pulses were a set of $N$ superimposed Gaussian-modulated sinusoids [43] spaced $\Delta t$ apart

$$\begin{aligned} y(t) &= \sum_{i=0}^{N-1} e^{-a|t-i\Delta t-t_0|^2} \sin(2\pi f_c(t-i\Delta t-t_0)) \\ a &= \tau^{-2} 2\log(2), \end{aligned} \tag{3.36}$$

where $\tau$ is the half-power pulse width and $f_c$ is the center frequency.

The signal parameters were varied randomly for each sample (Table 3.3). Pulse width and center frequency were fixed to achieve a representative pulse structure. The number of pulses and spacing were drawn from experimentally estimated distributions. The duration, initial frequency, and total bandwidth of the FM sweep were drawn at uniform random from a range of realistically observed values. All signals were centered within the 0.5 s spectrogram and assigned a random delay of within ±0.1 s.

White noise was added to the simulated signals using a fixed signal-to-noise ratio (SNR),

$$SNR = 10\log_{10}\frac{\sigma_s^2}{\sigma_n^2}, \quad \sigma_n^2 = \sigma_s^2 10^{-SNR/10}, \tag{3.37}$$

$$y(t) = y(t) + \mathcal{N}(0, \sigma_n^2 \mathbb{I}), \tag{3.38}$$

where $\sigma_s^2$ is the signal power and $\sigma_n^2$ is the noise power. The SNR range of each signal was determined from the experimental spectrograms during manual labeling. The SNR was estimated as the difference of the peak signal power to the median power of the background.

The signal power was estimated as the bandwidth-normalized mean power over the signal

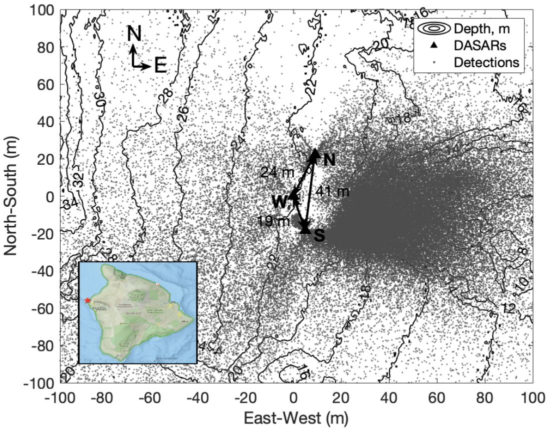

**Figure 3.6**: Diagram of the DASAR array deployed adjacent to a coral reef on the island of Hawaii. [9] The estimated detection locations are shown as gray dots. The majority of the reef is located due east of the array. The sensor positions were measured on the seafloor relative to DASAR W.

duration, [27]

$$\sigma_{s,FM}^2 = \frac{1}{\Delta f \cdot T} \int_{t_0}^{t_0+T} |y(t)|^2 dt, \tag{3.39}$$

$$\sigma_{s,pulse}^2 = \frac{1}{\Delta f \cdot 4\tau} \int_{t_0-2\tau}^{t_0+2\tau} |y(t)|^2 dt. \tag{3.40}$$

Following Sec. 3.2A, three handpicked features were extracted: peak frequency (3.23), kurtosis (3.24), and number of timeseries peaks (3.25). Duration, median power, and cross-sensor coherence were excluded due to limitations of the fixed simulation parameters. Then, K-means and hierarchical clustering were applied to the feature matrix to discover $K=2$ classes (fish or whale) or $K=3$ (fish, whale, or both).

Deep embedded clustering was applied directly to the spectrogram images according to Sec. 3.2B.

## 3.4 Data collection and Processing

Three directional autonomous seafloor acoustic recorders (DASARs) were deployed adjacent to a coral reef westward of the island of Hawaii. The DASARs, labeled N, W, and S from north to south, measured pressure and lateral particle velocity with *x*– and *y*– components oriented at orthogonal compass directions. The array was roughly oriented N-S with inter-sensor spacing about 15 m (Fig. 3.6).

The DASARs recorded continuously for 7 days with a sampling rate of 1 kHz. This study considers a 24-hour period on February 25, 2020. During this period, the dominant soundscape contributors below 500 Hz were reef fish, humpback whales, and motor noise from transiting surface boats, with boat noise occurring predominantly during daylight hours and fish calling most pronounced during the dusk hours.

The data were processed in 5 minute chunks to account for DASAR clock drift. First, a 256–point FFT with 90% overlap and Hanning window was used to generate the complex pressure spectrogram, matrix $\mathbf{S} \in \mathbb{C}^{N_f \times N_t}$ with units μPa·Hz$^{-1}$, with $\mathrm{d}t = 0.026$ s and $\mathrm{d}f = 3.91$ Hz. The complex spectrograms of the *x*– and *y*– particle velocity, matrices $\mathbf{V}_x \in \mathbb{C}^{N_f \times N_t}$ and $\mathbf{V}_y \in \mathbb{C}^{N_f \times N_t}$, were generated identically to $\mathbf{S}$ with units $m \cdot s^{-1}$. Then, the spectrograms from two sensors were cross-correlated along time to find the relative clock delay, assumed constant across 5 minutes.

The active intensity, a measure of in-plane energy, was used to determine the noise directionality:

$$\mathbf{A} = \text{atan2}\left(\Re\{\mathbf{S} \odot \mathbf{V}_y^*\}, \Re\{\mathbf{S} \odot \mathbf{V}_x^*\}\right), \tag{3.41}$$

$^*$ is the complex conjugate and $\Re$ the real component. Atan2 is an elementwise operation with

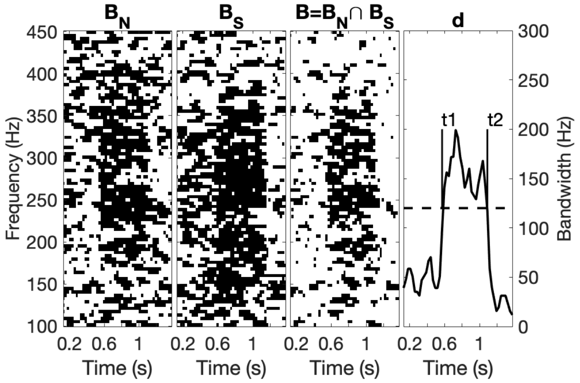

**Figure 3.7**: Directional detector for a fish call event on February 25, 07:12 HST, with DASAR N looking between 135°–225° and DASAR S looking between 45°–135° (clockwise from north). The overlap of the binary masks, summed across frequency, defines the detection timeseries.

domain $(0^\circ, 359^\circ)$, defined counterclockwise from the $y$-axis (0° = North), where for each $(x,y)$,

$$\text{atan2}(y,x) = \begin{cases} \arctan\left(\frac{x}{y}\right) & y > 0 \\ 180^\circ + \arctan\left(\frac{x}{y}\right) & y < 0. \end{cases} \tag{3.42}$$

**A** is therefore the time-frequency representation of compass directionality. In the following, matrix $\mathbf{A}_{\mathrm{N}}$ is called the *azigram* for DASAR N, likewise for $\mathbf{A}_{\mathrm{W}}$ and $\mathbf{A}_{\mathrm{S}}$.

### 3.4.1 Event detection

A directional event detector, [40] was developed to utilize the DASARs' directional capability by combining two DASARs, based on the assumptions:

1. An event arrives from a constant azimuthal sector for each DASAR.
2. Target events are broadband below 500 Hz. The minimum required bandwidth was set with an empirical threshold.

The detection algorithm is demonstrated in Fig. 3.7. First, the azigrams for the north- and southmost DASARs, $\mathbf{A}_N$ and $\mathbf{A}_S$, were used to create binary maps $\mathbf{B}_N$ and $\mathbf{B}_S$ of time-frequency points within a fixed azimuthal sector,

$$\mathbf{B}_\mathrm{N} = \mathbb{I}(\boldsymbol{\theta}_\mathrm{N}^{(1)} < \mathbf{A}_\mathrm{N} \leq \boldsymbol{\theta}_\mathrm{N}^{(2)}), \tag{3.43}$$

and likewise for $\mathbf{B}_S$ (Fig. 3.7a,b). $\mathbb{I}$ is the elementwise identity function, with $\mathbb{I}(\textbf{true}) = \mathbf{1}$. Binary maps were generated for all combinations of azimuthal sectors $\boldsymbol{\theta}_\mathrm{N}, \boldsymbol{\theta}_\mathrm{S} \in ([0^\circ, \Delta\theta^\circ]^T, [\frac{\Delta\theta}{2}^\circ, \frac{3\Delta\theta}{2}^\circ]^T, \ldots, [(360-\Delta\theta)^\circ, 360^\circ])$.

Next, overlapping signals on both DASARs were discovered by creating a combined map (Fig. 3.7c),

$$\mathbf{B} = \mathbf{B}_\mathrm{N} \cap \mathbf{B}_\mathrm{S}. \tag{3.44}$$

The detection timeseries was generated by summing across frequency of events,

$$\mathbf{d} = df * \sum_i \mathbf{B}(i,:) \tag{3.45}$$

$\mathbf{d}$ measures the bandwidth of an event. Event start and end times were determined for $d_j > T$, $j = 1, \ldots, N_t$ for threshold $T$. Events separated by less than $M_\mathrm{sep} \cdot dt$ were merged and events longer than $T_\mathrm{max}$ were removed.

For this study, the detector parameters were $\Delta\theta = 90^\circ$, $T = 120$ Hz, $M_\mathrm{sep} = 1$ ($M_\mathrm{sep} \cdot \mathrm{d}t = 0.0256$ s), and $T_\mathrm{max} = 2$ s. Detected events were localized [21] to ensure physicality and that their signal had sufficient bandwidth for feature extraction.

Detections for which the localization algorithm failed to converge were discarded. The remaining events were spatially filtered within a 100 m by 100 m box from DASAR S (Fig. 3.6). 92,736 localizable detections within 100 m were kept for further analysis, on average about 1 detection per second.

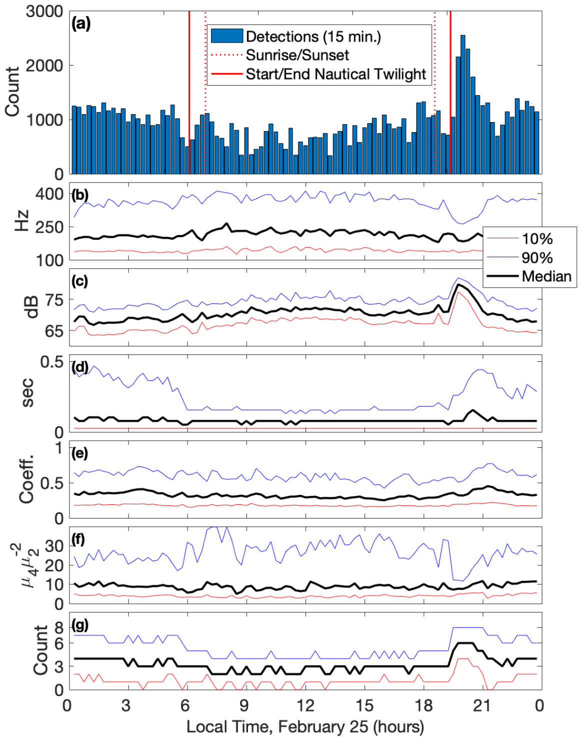

**Figure 3.8**: *(a)* Number of detected events per 15 minutes on February 25 and *(b–g)* 10%, 50%, and 90% levels per 15 minutes for each feature: *(b)* peak frequency, *(c)* median time-frequency power, *(d)* event duration, *(e)* normalized time coherence between sensors, *(f)* kurtosis, and *(g)* number of time peaks. All features were measured on the DASAR S. The coherence is the normalized correlation lag coefficient between DASARs N and S.

Fig. 3.8 shows the number of events detected along with extracted feature median, 10%, and 90% levels for every 15 minutes. Fish calls were most common during nighttime, with pulse trains peaking during the evening chorus after nautical twilight (19:15 HST). The evening chorus corresponded to a visible increase in median power, event duration, and number of time peaks and a visible decrease in the 90th percentiles of peak frequency and kurtosis.

## 3.5 Clustering Analysis

### 3.5.1 Metrics

The performance of the handpicked feature clustering and DEC was measured by their accuracy, precision, and recall,

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(t_i,\hat{t}_i), \quad t_i,\hat{t}_i \in \{0,1,2\} \tag{3.46}$$

$$\text{Precision} = \frac{\sum_{i=1}^{N}\mathbb{I}(t_i,\hat{t}_i)}{\sum_{j=1}^{N}\mathbb{I}(\hat{t}_j,0)}, \quad \text{Recall} = \frac{\sum_{i=1}^{N}\mathbb{I}(t_i,\hat{t}_i)}{\sum_{j=1}^{N}\mathbb{I}(t_j,0)} \tag{3.47}$$

where $t_i$ is the true label and $\hat{t}_i$ is the class prediction. Here, whale song (label 0) was the true class. Precision (positive predictive value or PPV) measured the ratio of correctly predicted whale song events to total predicted whale song. Recall (hit rate or true positive rate) measured the ratio of correctly classified whale song events to the true total. Higher metrics correspond to improved performance, with perfect performance given by accuracy, precision, and recall all equal to 1.

### 3.5.2 Simulations

First, the handpicked features were examined for separability for $K=2$ equal-sized classes (fish, whale) and $K=3$ classes (fish, whale, both) (Fig. 3.9). Whale song and fish call overlapped in peak frequency and in the number of automatically extracted temporal peaks. The signals were most strongly separated by kurtosis, with whale song having very low kurtosis. The optimal clustering was found when all features were included. The handpicked features were clustered using the unsupervised clustering methods K-means and hierarchical agglomerative clustering (Table 3.4), with $K=2$ and $K=3$ assumed for each case. The known simulation labels were used for post-clustering comparison of the methods.

For 2 equal-sized classes, K-means had marginally higher accuracy and recall than

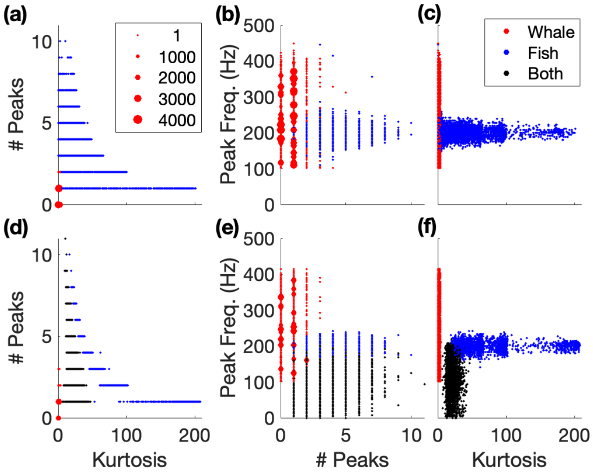

**Figure 3.9**: Handpicked features demonstrate some separability by kurtosis, number of time-series peaks, and peak frequency but with overlapping clusters for *(a-c)* two classes (whale, fish) and *(d-f)* (bottom) three classes (whale, fish, both). The dots are scaled to indicate density of feature pairs, with each dot increased by 1 pt for every 100 samples.

hierarchical clustering, but lower precision (Table 3.4), indicating that the signal clusters were not equidistant in feature space. For $K = 2$ with imbalanced classes, only 25% of simulated events were whale song, with the remaining 75% fish calls (2500/7500). In this case, hierarchical agglomerative clustering had slightly higher accuracy and recall, but slightly lower precision than K-means. The accuracy, recall, and precision were overall higher for the imbalanced classes (Table 3.4), which is likely reflective of an imbalance in feature densities for the different classes.

The classification accuracy for 3 equal-sized classes was higher than for 2 equal-sized classes due to the choice of simulation SNR, which was selected as the higher of the two signals (15–30 dB) for the combined features. The handpicked feature values for the combined fish and whale class were between those of whale song or fish call alone. Overall, low recall and accuracy values for the handpicked feature clustering indicate that many of the whale song events were misclassified, but high precision indicates that most events classified as whale song were correct.

Model parameters cannot be determined by cross-validation in unsupervised clustering due to its inherently label-free nature. [44] To achieve reasonable classification accuracy, a previously successful model architecture was employed, [37] and the simulation accuracy was

**Table 3.4**: Classification accuracy, precision, and recall on simulations for unsupervised clustering with hand-crafted features (K-means, hierarchical clustering) vs deep learning (DEC). Whale song was defined as the positive class.

| Method | Accuracy | Recall | Precision |
|---|---|---|---|
| | $K = 2$ | | |
| K-means | 0.73 | 0.47 | 0.99 |
| Hierarchical | 0.66 | 0.32 | 1.0 |
| DEC ($P = 10$) | 0.99 | 0.99 | 0.99 |
| | $K = 2$ (imbalanced) | | |
| K-means | 0.86 | 0.46 | 0.98 |
| Hierarchical | 0.87 | 0.50 | 0.97 |
| DEC ($P = 10$) | 0.75 | 0.99 | 0.50 |
| | $K = 3$ | | |
| K-means | 0.77 | 0.49 | 1.0 |
| Hierarchical | 0.71 | 0.45 | 1.0 |
| DEC ($P = 15$) | 0.86 | 0.99 | 0.96 |

examined for DEC with $K = 2$ and $K = 3$ equal-sized classes at varying latent feature vector dimensionality (Fig. 3.10). When the latent dimension was low, eg $P < 8$, all features were clustered in a single class. For 2 classes, accuracy was consistent when $P > 8$. For 3 classes, accuracy was unstable. The instability was likely caused by weak convergence and sensitivity to the weight initialization, but the minimum latent dimension was consistent across random seeds. In general, higher accuracy was achieved with higher latent dimension when $K = 3$.

DEC clustering for 2 equal-sized classes is shown in Fig. 3.11(a,b) using a t-SNE repre-

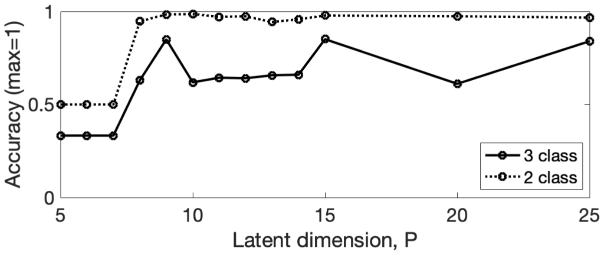

**Figure 3.10**: Accuracy of DEC for 10000 simulated signals with varying latent dimension ($P$). $K = 3$ classes contained fish, whale, or fish and whale, and $K = 2$ classes had fish or whale. In both cases, the method fails at low $P$.

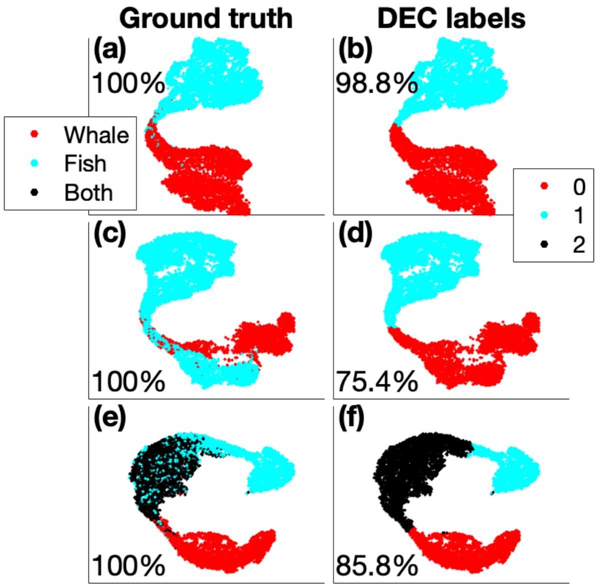

**Figure 3.11**: t-SNE representation of the deep embedded feature vectors for 10000 simulated examples with *(a,b)* fish call and whale song classes, and *(c,d)* fish call, whale song, and fish/whale classes. The plots are colored by *(a,c)* ground truth labels and *(b,d)* DEC predicted labels. The perplexity value was 200.

sentation of the $P = 10$–dimensional latent feature vector with perplexity 200. The high perplexity improved visualization, likely due to the large number of samples relative to the feature dimension. The point colors represent the two class labels. Signals that were spectrally similar were most likely to be misclassified, such as low-frequency FM sweeps and closely spaced fish pulse sequences lasting from 0.2 s to 0.4 s. In most cases, DEC successfully separated fish calls and whale song into separate classes, with high accuracy, precision, and recall (Table 3.4).

For 2 imbalanced classes (2500 whale/7500 fish calls), DEC had reduced accuracy likely due to its convergence to equal-sized clusters as shown in Fig. 3.11(c,d). The reduction in precision and accuracy was directly proportional to the reduction in the class sizes (Table 3.4), while recall was unchanged because the whale song class was overestimated (Fig. 3.11(c,d)).

The addition of a combined signal class, using $K = 3$ equal-sized classes, demonstrated that overlapping signals were difficult to differentiate in the spectral domain (Fig. 3.11(e,f)). Most whale song events were correctly classified, as indicated by a high recall value. The largest

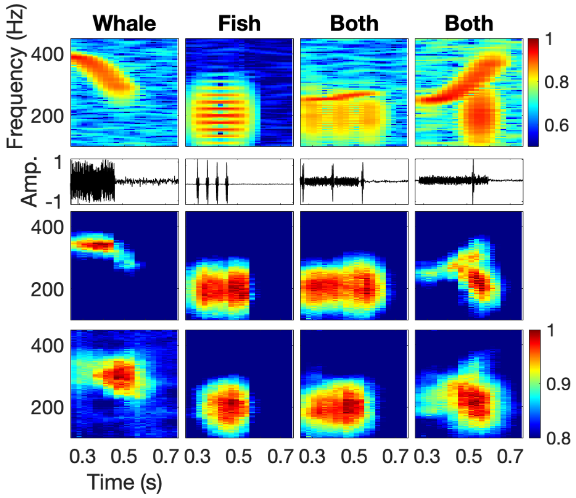

**Figure 3.12**: Simulated coral reef bioacoustic signals successfully classified using DEC. From top to bottom rows: input spectrograms, simulated timeseries, initial DEC reconstruction, and DEC reconstruction after adding clustering loss.

classification overlap was between fish calls and combined class. Figure 3.12 shows that fish calls with large bandwidth and duration may dominate the spectral signature, which may lead to increased misclassification.

### 3.5.3 Experiment

The clustering methods from Sec. 3.3 were applied to a subset of 10,000 unlabeled detections randomly selected from the Hawaii 2020 experiment. Each detection contained one or more directional signals with unknown SNR. Then, labels of whale song/no whale song (fish calls only) were manually assigned for 4000 samples or 40%, based on the signal within the detection window. About two-thirds were labeled as no whale song and contained only fish calls.

The results of clustering the $P = 6$–dimensional handpicked feature vectors with K-means and $K = 2$ are shown using t-SNE in Fig. 3.13b. The overall accuracy, precision, and recall were low (Table 3.5), indicating that the clustering methods did not align with the manual labels. Similar results were found for hierarchical clustering. These results were in line with the simulation results but suggest that feature extraction was less reliable in the experimental data.

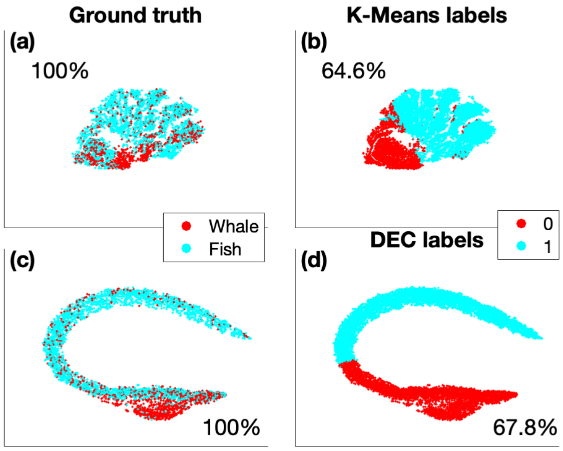

**Figure 3.13**: Experimental data from a Hawaiian coral reef shown as a t-SNE representation of *(a, b)* $P = 6$ handpicked features and *(c,d)* DEC learned features for 10,000 random detections, colored by *(a,c)* hand-labeled classes and *(b,d)* K-means clustering with $K = 2$. The perplexity was 200 for DEC and 300 for the physical features.

**Table 3.5**: Classification accuracy determined from manually labeled experimental detections for unsupervised clustering with handpicked features (K-means, hierarchical clustering) vs deep learning (DEC).

| Method | Accuracy | Recall | Precision |
|---|---|---|---|
| K-means ($K = 2$) | 0.65 | 0.41 | 0.44 |
| Hierarchical ($K = 2$) | 0.51 | 0.56 | 0.34 |
| DEC ($P = 10$, $K = 2$) | 0.68 | 0.83 | 0.60 |

Figure 3.13d shows the result of DEC with $P = 15$ using the normalized input spectrogram estimated from the detected signals. The DEC was pretrained for 5000 epochs in order to account for increased variability in the experimental data before updating clusters for 20 epochs. DEC accuracy was slightly higher than handpicked feature clustering, but the significantly higher recall value demonstrates that DEC correctly classified many of the whale song events. The class imbalance between whale song and fish calls, evident in Fig. 3.13c, was not well captured by the DEC algorithm used in this study, which converged towards well-balanced classes. The reduced accuracy and precision were a result of this imbalance.

Successfully classified spectrogram reconstructions and their corresponding timeseries

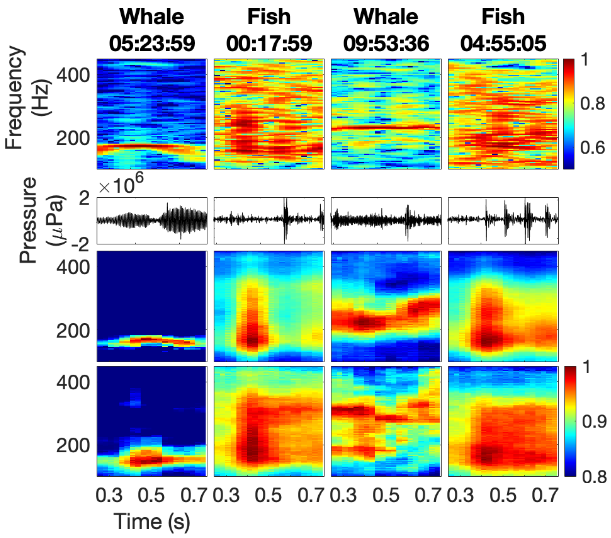

**Figure 3.14**: Experimentally detected coral reef bioacoustic signals successfully classified using DEC. From top to bottom rows: input spectrograms, simulated timeseries, initial DEC reconstruction, and DEC reconstruction after adding clustering loss. The start time of each detection on February 25, 2020 is formatted as HH:MM:SS relative to midnight.

are shown in Fig. 3.14. These demonstrate that whale song was primarily identified by its narrow bandwidth and temporal extent, whereas fish call pulse sequences were identified as broadband. The unscaled timeseries in Fig. 3.14 demonstrate the magnitude variation between events that was not attributed to signal type, motivating the normalization of the spectrograms.

## 3.6 Discussion

An unsupervised machine learning approach was presented for interpreting unlabeled coral reef bioacoustic detections. This approach considers and expands upon methods from recently proposed automatic fish call classifiers. [17, 22, 24]

Given the complex nature of the coral reef soundscape, two approaches were proposed to separate whale song from fish calls. First, handpicked features known to be correlated with coral reef fish species [25, 29, 42] and other relevant acoustic metrics [24, 28] were extracted. These features were clustered using hierarchical clustering and K-means. Then, the deep clustering approach DEC was used to jointly learn features and cluster labels directly from the spectrograms.

Clustering of simulated fish calls (Gaussian pulses) and whale song units (FM sweeps) demonstrated that handpicked features overlapped enough to reduce unsupervised clustering accuracy. By jointly learning features and clusters, DEC was successful in separating fish calls and whale song directly from spectrograms. However, the DEC algorithm [13, 44] with parameters implemented in this study was observed to converge to equal-sized classes, resulting in higher misclassifications. A combined class with overlapping whale song and fish call also reduced DEC performance due to its inability to distinguish separate signals within the spectrograms. Handpicked feature clustering performed similarly with or without the inclusion of a combined class and demonstrated improved accuracy on imbalanced classes, indicating that the handpicked features were not evenly distributed in feature space. In all scenarios except the simulated imbalanced case, DEC had higher accuracy and recall than the handpicked clustering and was more likely to correctly classify existing whale song events.

A directional detector was used to identify potentially localizable broadband bioacoustic events on a Hawaiian coral reef in February–March 2020. A labeled subset of these detections with whale song/no whale song indicated that about two-thirds of the detections contained primarily fish calls and one-third contained a whale song segment.

Unsupervised K-means clustering of handpicked features with $K = 2$ on the experimental, manually labeled data achieved low accuracy, precision, and recall. DEC with $K = 2$ achieved similar accuracy, but its higher recall demonstrated its ability to correctly classify many whale song events. A class imbalance between whale song and fish calls likely led the DEC algorithm to define incorrect class boundaries (Fig. 3.13c,d), as evidenced in its lower precision. DEC reconstructions of the input spectrograms demonstrate that the learned features are representative of spectral features identified by manual labelers.

These results demonstrate that DEC is a promising method for clustering unlabeled bioacoustic signals with distinct spectral signatures. Our results indicate that the feature extraction process is key for unsupervised clustering of hand-picked features and that the feature distributions may be as or more important than their physical relations to the signal. Finally, class imbalance is an important consideration, particularly for unlabeled data where the class priors are unknown. As class imbalance is a common occurrence in ambient noise acoustics and geophysics, DEC clustering algorithms should be considered that jointly learn or regularize the weighted class priors as well as the cluster distributions.

## 3.7 Acknowledgements

Thank you to Greeneridge Sciences for providing the DASAR sensors and to Alex Conrad for assisting with DASAR post-processing. Thank you to Richard Walsh for assistance with hardware deployment and recovery. This work was supported by the Office of Naval research under award N00014-18-1-2065.

# Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] M. C. P. Amorim. Diversity of sound production in fish. *Commun. Fish*, 1:71–104, 2006.

[3] P. C. Bermant, M. M. Bronstein, R. J. Wood, S. Gero, and D. F Gruber. Deep machine learning techniques for the detection and classification of sperm whale bioacoustics. *Scientific Reports*, 9(1):12588, 2019.

[4] M. J. Bianco, P. Gerstoft, J. Traer, E. Ozanich, M. A. Roch, S. Gannot, and C.-A. Deledalle. Machine learning in acoustics: Theory and applications. *J. Acoust. Soc. Am*, 146(5):3590–3628, November 2019.

[5] C. Biemann. *Structure Discovery in Natural Language: Theory and Applications of Natural Language Processing*, pages 73–75. Springer-Verlag, Berlin Heidelberg, 2012.

[6] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Chaps. 4, 5, and 7, 2006.

[7] F. Chollet et al. Keras. https://keras.io, 2015.

[8] Volker B Deecke and Vincent M Janik. Automated categorization of bioacoustic signals: avoiding perceptual pitfalls. *J. Acoust. Soc. Am.*, 119(1):645–653, 2006.

[9] Esri. *National Geographic Style Map [basemap]*, November 23, 2020. https://www.arcgis.com/home/webmap/.

[10] K. E. Frasier, M. A. Roch, M. S. Soldevilla, S. M. Wiggins, L. P. Garrison, and J. A. Hildebrand. Automated classification of dolphin echolocation click types from the Gulf of Mexico. *PLOS Computational Biology*, 13(12):1–23, 12 2017.

[11] Kaitlin E Frasier, E Elizabeth Henderson, Hannah R Bassett, and Marie A Roch. Automated identification and clustering of subunits within delphinid vocalizations. *Marine Mammal Sci.*, 32(3):911–930, 2016.

[12] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, chapter 14, pages 163–215, 493–495, 499–500. Massachusetts Institute of Technology, 2016.

[13] X. Guo, L. Gao, X. Lui, and J. Yin. Improved deep embedded clustering with local structure preservation. *Proc. 26th Int. Joint Conf. Art. Intel. (IJCAI)*, pages 1753–1759, 2017.

[14] L. V. D. Haaten and G. Hinton. J. mach. learn. res. *Visualizing data using t-SNE*, pages 2579–2605, 2008.

[15] X. C. Halkias, S. Paris, and H. Glotin. Classification of mysticete sounds using machine learning techniques. *J. Acous. Soc. Am.*, 134(5):3496–3505, 2013.

[16] T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning*, chapter 13.2.1, page 460. Springer, New York, second edition, 2009.

[17] A. K. Ibrahim, H. Zhuang, L. M. Chérubin, M. T. Schärer-Umpierre, and N. Erdol. Automatic classification of grouper species by their sounds using deep neural networks. *J. Acoust. Soc. Am.*, 144(3):EL196–EL202, September 2018.

[18] Jr. J. H. Ward. Hierarchical grouping to optimize an objective function. *Am. Stat. Ass. J.*, pages 236–244, March 1963.

[19] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *Proc. of the 3rd ICLR*, 2014.

[20] W. Lee and V. Staneva. Compact representation of temporal processes in echosounder time series via matrix decomposition, 2020.

[21] R. V. Lenth. On Finding the Source of a Signal. *Technometrics*, 23(2):149–154, May 1981.

[22] T.-H. Lin, Y. Tsao, and T. Akamatsu. Comparison of passive acoustic soniferous fish monitoring with supervised and unsupervised approaches. *J. Acoust. Soc. Am.*, 143(4):EL278–EL284, April 2018.

[23] L. V. D. Maaten. Barnes-hut-sne. *CoRR*, abs/1301.3342, 2013.

[24] M. Malfante, J. I. Mars, M. D. Mura, and C. Gervaise. Automatic fish sounds classification. *J. Acoust. Soc. Am.*, 143(5):2834–2846, May 2018.

[25] D. A. Mann and P. S. Lobel. Propagation of damselfish (*pomacentridae*) courtship sounds. *J. Acoust. Soc. Am.*, 101(6):3783–3791, February 1997.

[26] S. Mann and S. Haykin. The Chirplet Transform: A Generalization of Gabor's Logon Ttransform. *Vision Interface 1991*, pages 205–212, 1991.

[27] D. Manolakis and J. G. Proakis. In *Digital Signal Processing: Principles, Algorithms, and Applications*, chapter 2.1.2, pages 47–52. Prentice-Hall International, Inc., 3rd edition, 1996.

[28] S. B. Martin, K. Lucke, and D. R. Barclay. Techniques for distinguishing between impulsive and non-impulsive sound in the context of regulating sound exposure for marine mammals. *J. Acoust. Soc. Am.*, 147(4):2159–2176, April 2020.

[29] K. P. Maruska, K. S. Boyle, L. R. Dewan, and T. C. Tricas. Sound production and spectral hearing sensitivity in the Hawaiian sergeant damselfish, *abudefduf abdominalis*. *J. Exp. Biol.*, 210:3990–4004, 2007.

[30] Mathworks. Statistics and Machine Learning Toolbox: User's Guide (R2019b), 2019. Access online 28 February 2020 at `https://www.mathworks.com/help/pdf_doc/stats/stats.pdf`.

[31] B. McCowan. A new quantitative technique for categorizing whistles using simulated signal and whistles from captive bottlenose dolphins (delphinidae, *tursiops truncatus*. *Ethology*, 100(3):177–193, 1995.

[32] D. K. Mellinger and C. W. Clark. Methods for automatic detection of mysticete sounds. *Marine Freshw. Behav. Phys.*, 29(1-4):163–181, 1997.

[33] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*, pages 389–397, 897–900. Massachusetts Institute of Technology, 2012.

[34] M. A. Roch, H. Klinck, S. Baumann-Pickering, D. K. Mellinger, S. Qui, M. S. Soldevilla, and J. A. Hildebrand. Classification of echolocation clicks from odontocetes in the southern california bight. *The Journal of the Acoustical Society of America*, 129(1):467–475, 2011.

[35] Y. Shiu, K. J. Palmer, M. Roch, E. Fleishman, X. Liu, E.-M. Nosal, T. Helble, D. Cholewiak, D. Gillespie, and H. Klinck. Deep neural networks for automated detection of marine mammal species. *Scientific Reports*, 10(1):607, 2020.

[36] Evgeny Smirnov. North atlantic right whale call detection with convolutional neural networks. In *Int. Conf. on Mach. Learn.*, pages 78–79. Citeseer, 2013.

[37] D. Snover, C. W. Johnson, M. J. Bianco, and P. Gerstoft. Deep Clustering to Identify Sources of Urban Seismic Noise in Long Beach, California. *Seismol. Res. Lett.*, pages 1–12, 2020.

[38] W. W. Steiner. Species-specific differences in pure tonal whistle vocalizations of five western North Atlantic dolphin species. *Behav. Ecol. Sociobiol.*, 9(4):241–246, 1981.

[39] D. Stowell and M. D. Plumbley. Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning. *PeerJ*, 2:e488, 2014.

[40] A. Thode, A.S. Conrad, R. Ozanich, E. King, S. E. Freeman, L. A. Freeman, B. Zgliczynski, P. Gerstoft, and K. H. Kim. Automated two-dimensional localization of underwater acoustic transient impulses using vector sensor image processing. *J. Acoust. Soc. Am.*, *In Review*.

[41] A. M. Thode, K. H. Kim, R. G. Norman, S. B. Blackwell, and C. R. Greene. Acoustic vector sensor beamforming reduces masking from underwater industrial noise during passive monitoring. *J. Acoust. Soc. Am.*, 139(4):EL105–EL111, 2016.

[42] T. C. Tricas and K. S. Boyle. Acoustic behaviors in Hawaiian coral reef fish communities. *Mar Ecol Prog Ser*, 511:1–16, September 2014.

[43] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, J. K. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Ä. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and ScipPy 1.0 Contributors. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, 2020.

[44] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedded for clustering analysis. *Proc. 33rd Int. Conf. Mach. Learn.*, 2016.

# Chapter 4

# Source localization in an ocean waveguide using supervised machine learning

Source localization in ocean acoustics is posed as a machine learning problem in which data-driven methods learn source ranges directly from observed acoustic data. The pressure received by a vertical linear array is preprocessed by constructing a normalized sample covariance matrix (SCM) and used as the input for three machine learning methods: feed-forward neural networks (FNN), support vector machines (SVM) and random forests (RF). The range estimation problem is solved both as a classification problem and as a regression problem by these three machine learning algorithms. The results of range estimation for the Noise09 experiment are compared for FNN, SVM, RF and conventional matched-field processing and demonstrate the potential of machine learning for underwater source localization.

## 4.1 Introduction

Acoustic source localization in ocean waveguides is often solved with matched-field processing (MFP). [1]$^{-}$ [2] Despite the success of MFP, it is limited in some practical applications due to its sensitivity to the mismatch between model-generated replica fields and measurements. MFP gives reasonable predictions only if the ocean environment can be accurately modeled. Unfortunately, this is difficult because the realistic ocean environment is complicated and unstable.

An alternative approach to the source localization problem is to find features directly from data. [3]$^{-}$ [4] Interest in machine learning techniques has been revived thanks to increased computational resources as well as their ability to learn nonlinear relationships. A notable recent example in ocean acoustics is the application of nonlinear regression to source localization. [5] Other machine learning methods have obtained remarkable results when applied to areas such as speech recognition, [6] image processing, [7] natural language processing, [8] and seismology. [9]$^{-}$ [10] Most underwater acoustics research in machine learning is based on 1990s neural networks. Previous research has applied neutral networks to determine the source location in a homogeneous medium, [11] simulated range and depth discrimination using artificial neural networks in matched-field processing, [12] estimated ocean-sediment properties using radial basis functions in regression and neural networks, [13]$^{,}$ [14] applied artificial neural networks to estimation of geoacoustic model parameters, [15]$^{,}$ [16] classification of seafloor [17] and whale sounds. [18]

This paper explores the use of current machine learning methods for source range localization. The feed-forward neural network (FNN), support vector machine (SVM) and random forest (RF) methods are investigated. There are several main differences between our work and previous studies of source localization and inversion: [5]$^{,}$ [11]$^{-}$ [18]

1. Acoustic observations are used to train the machine learning models instead of using model-generated fields. [11]$^{,}$ [12]$^{,}$ [14]$^{-}$ [16]

2. For input data, normalized sample covariance matrices, including amplitude and phase information, are used. Other alternatives include the complex pressure, [5] phase difference, [11] eigenvalues, [12] amplitude of the pressure field, [14] transmission loss, [15]$^{,}$ [16] angular dependence of backscatter, [17] or features extracted from spectrograms. [18] This preprocessing

procedure is known as feature extraction in machine learning.

3. Under machine learning framework, source localization can be solved as a classification or a regression problem. This work focuses on classification in addition to the regression approach used in previous studies. [5]' [11]' [13]$^{-}$ [16]

4. Well-developed machine learning libraries are used. Presently, there are numerous efficient open source machine learning libraries available, including TensorFlow, [19] Scikit-learn, [20] Theano, [21] Caffe, [22] and Torch, [23] all of which solve typical machine learning tasks with comparable efficiency. Here, TensorFlow is used to implement FNN because of its simple architecture and wide user base. Scikit-learn is used to implement SVM and RF as they are not included in the current TensorFlow version. Compared to older neural network implementations, Tensorflow includes improved optimization algorithms [24] with better convergence, more robust model with dropout [25] technique and high computational efficiency.

The paper is organized as follows. The input data preprocessing and source range mapping are discussed in Secs. 4.2.1 and 4.2.2. The theoretical basis of FNN, SVM and RF is given in Secs. 4.2.3–4.2.5. Simulations and experimental results in Secs. 4.3 and 4.4 demonstrate the performance of FNN, SVM and RF. In Sec. 4.5, the effect of varying the model parameters is discussed. The conclusion is given in Sec. 4.6.

## 4.2 Localization based on machine learning

The dynamics of the ocean and its boundary cause a stochastic relationship between the received pressure phase and amplitude at the array and the source range. After preprocessing we assume a deterministic relationship between ship range and sample covariance matrix. The pressure–range relationship is in general unknown but may be discovered using machine learning methods. The received pressure is preprocessed and used as the input of the machine learning models (Sec. 4.2.1). The desired output may be either discrete (classification) or continuous (regression) corresponding to the estimated source range (Sec. 4.2.2). The theory of FNN, SVM, and RF are described in Secs. 4.2.3–4.2.5.

### 4.2.1 Input data preprocessing

To make the processing independent of the complex source spectra, the received array pressure is transformed to a normalized sample covariance matrix. The complex pressure at frequency $f$ obtained by taking the DFT of the input pressure data at $L$ sensors is denoted by $\mathbf{p}(f) = [p_1(f), \cdots, p_L(f)]^T$. The sound pressure is modeled as

$$\mathbf{p}(f) = S(f)\mathbf{g}(f,\mathbf{r}) + \varepsilon, \tag{4.1}$$

where $\varepsilon$ is the noise, $S(f)$ is the source term, and $\mathbf{g}$ is the Green's function. To reduce the effect of the source amplitude $|S(f)|$, this complex pressure is normalized according to

$$\tilde{\mathbf{p}}(f) = \frac{\mathbf{p}(f)}{\sqrt{\sum\limits_{l=1}^{L} |p_l(f)|^2}} = \frac{\mathbf{p}(f)}{\|\mathbf{p}(f)\|_2}. \tag{4.2}$$

The normalized sample covariance matrices (SCMs) are averaged over $N_s$ snapshots to form the conjugate symmetric matrix

$$\mathbf{C}(f) = \frac{1}{N_s} \sum_{s=1}^{N_s} \tilde{\mathbf{p}}_s(f)\tilde{\mathbf{p}}_s^H(f), \tag{4.3}$$

where $H$ denotes conjugate transpose operator and $\tilde{\mathbf{p}}_s$ represents the sound pressure over the $s$th snapshot. The product $\tilde{\mathbf{p}}_s(f)\tilde{\mathbf{p}}_s^H(f)$ contains an $S(f)S(f)^H$ term, which for large SNR is dominant and thus reduces the effect of the source phase. Preprocessing the data according to Eqs. (4.2) and (4.3) ensures that the Green's function is used for localization. Only the real and imaginary parts of the complex valued entries of diagonal and upper triangular matrix in $\mathbf{C}(f)$ are used as input to save memory and improve calculation speed. These entries are vectorized to form the real-valued input **x** of size $L \times (L+1)$ to the FNN, SVM and RF.

### 4.2.2 Source range mapping

In the classification problem, a set of source ranges is discretized into $K$ bins, $r_1, ..., r_K$, of equal width $\Delta r$. Each input vector, $\mathbf{x}_n, n = 1, .., N$, is labeled by $t_n$, where $t_n \in r_k, k = 1, ..., K$; this label represents the true source range class and is the target output for the model. SVM and RF use this classification scheme to train and predict the source range for each sample.

For the FNN, the range class $t_n$ is mapped to a $1 \times K$ binary vector, $\mathbf{t}_n$, such that:

$$t_{nk} = \begin{cases} 1 & \text{if } |t_n - r_k| \leq \frac{\Delta r}{2}, \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

$\mathbf{t}_n = t_{n,1}, ..., t_{n,K}$ therefore represents the expected output probability of the neural network, i.e. the probability that the source is at range $r_k$ for input $\mathbf{x}_n$. These target vectors are used to train the FNN. The FNN output predictions are given as a softmax distribution with maximum at the predicted range (see Sec. 4.2.3).

In the regression problem, the target output $r_n \in [0, \infty)$ is a continuous range variable for all three models.

### 4.2.3 Feed-forward neural networks

The feed-forward neural network (FNN), also known as multi-layer perceptron, is constructed using a feed-forward directed acyclic architecture, see Fig. 4.1(a). The outputs are formed through a series of functional transformations of the weighted inputs. In the FNN, the outputs are deterministic functions of the inputs. [26]

Here, a three layer model (input layer $L_1$, hidden layer $L_2$ and output layer $L_3$) is used to

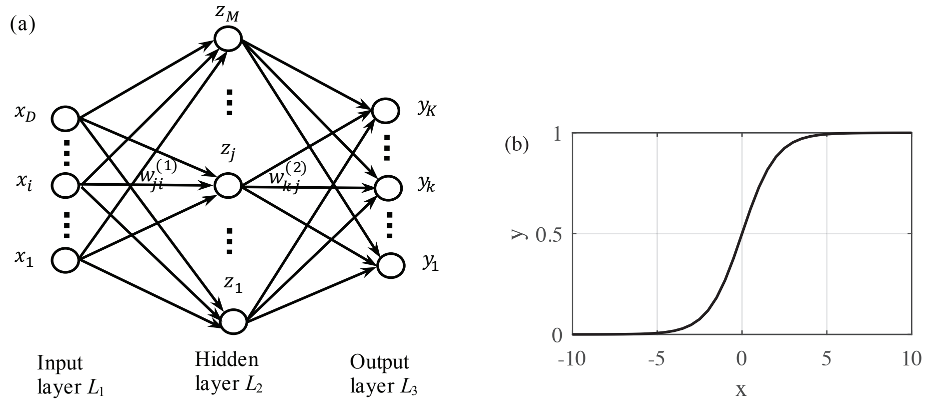

**Figure 4.1**: (a) Diagram of a feed-forward neural network and (b) Sigmoid function.

construct the FNN. The input layer $L_1$ is comprised of $D$ input variables $\mathbf{x} = [x_1, \cdots, x_D]^T$. The $j$th linear combination of the input variables is given by

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}, \qquad j = 1, \cdots, M, \tag{4.5}$$

where $M$ is the number of neurons in $L_2$ and the superscript indicates that the corresponding parameters are in the first layer of the network. The parameters $w_{ji}^{(1)}$ and $w_{j0}^{(1)}$ are called the weights and biases and their linear combinations $a_j$ are called activations. In $L_2$, the activations are transformed using an activation function $f(\cdot)$,

$$z_j = f(a_j). \tag{4.6}$$

The logistic sigmoid was chosen as the intermediate activation function for this study, see Fig. 4.1(b):

$$f(a) = \sigma(a) = \frac{1}{1+e^{-a}}. \tag{4.7}$$

Similarly, for output layer $L_3$, the $K$ output unit activations are expressed as linear combinations of $z_j$

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)}, \qquad k = 1, \cdots, K \tag{4.8}$$

where $w_{kj}^{(2)}$ and $w_{k0}^{(2)}$ represent weights and biases for the second layer.

In the output layer, the softmax function is used as the activation function. The softmax is a common choice for multi-class classification problems. [26] Here, it constrains the output class, $y_k(\mathbf{x}, \mathbf{w})$, to be the probability that the source is at range $r_k$: [26]

$$y_k(\mathbf{x}, \mathbf{w}) = \frac{\exp(a_k(\mathbf{x}, \mathbf{w}))}{\sum_{j=1}^{K} \exp(a_j(\mathbf{x}, \mathbf{w}))}, \quad k = 1, \cdots, K \tag{4.9}$$

where $\mathbf{w}$ is the set of all weight and bias parameters and $y_k$ satisfies $0 \leq y_k \leq 1$ and $\sum_k y_k = 1$.

Before applying the FNN to unlabeled data, the weights and biases $\mathbf{w}$ are determined by training the model on labeled data. Recall that in the FNN case, $\mathbf{t}_n$ is the binary target vector, or true probability distribution (see Sec. 4.2.2), and $y_k(\mathbf{x}_n, \mathbf{w})$ is the estimated probability distribution, for the input $\mathbf{x}_n$ (see Sec. 4.2.1).

During training, the Kullback–Leibler (KL) divergence

$$D_{\mathrm{KL}}(\mathbf{t}_n || \mathbf{y}(\mathbf{x}_n, \mathbf{w})) = \sum_k t_{nk} \left[\ln t_{nk} - \ln y_{nk}\right], \tag{4.10}$$

represents the dissimilarity between $y_{nk} = y_k(\mathbf{x}_n, \mathbf{w})$ and $t_{nk}$, where $\mathbf{t}_n = [t_{n1}, ..., t_{nk}]$, $k = 1, ..., K$. Minimizing the KL divergence $D_{\mathrm{KL}}$ is equivalent to minimizing the cross entropy function $E_n$

$$E_n(\mathbf{t}_n, \mathbf{y}(\mathbf{x}_n, \mathbf{w})) = -\sum_k t_{nk} \ln y_{nk}, \tag{4.11}$$

since the desired output $\mathbf{t}_n$ is constant (independent of $\mathbf{w}$).

For $N$ observation vectors, the averaged cross entropy and resulting weights and biases are

$$E(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk}, \tag{4.12}$$

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \left[ -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk} \right]. \tag{4.13}$$

For the regression problem, there is only one neuron in the output layer representing the continuous range variable. Instead of using Eq. (4.12), a sum-of-squares error function [26] is minimized

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} |y(\mathbf{x}_n, \mathbf{w}) - r_n|^2, \tag{4.14}$$

where $r_n$ is the true source range at sample $n$.

Several optimization methods are provided in the TensorFlow software. In this paper, Adam [24](Adaptive Moment estimation) is used.

### 4.2.4 Support Vector Machine

Unlike neural networks, support vector machines (SVM) are decision machines that do not provide a posterior probability. [26] Instead, the data is divided into two (or more) classes by defining a hyperplane that maximally separates the classes.

First, for simplicity, assume the input $\mathbf{x}_n, n = 1, \cdots, N$ are linearly separable (see Fig. 4.2) and can be divided into two classes, $s_n \in \{1, -1\}$. The class of each input point $\mathbf{x}_n$ is determined by the form [26]

$$\mathrm{y}_n = \mathbf{w}^T \mathbf{x}_n + b, \tag{4.15}$$

where $\mathbf{w}$ and $b$ are the unknown weights and bias. A hyperplane satisfying $\mathbf{w}^T\mathbf{x} + b = 0$ is used to separate the classes. If $\mathrm{y}_n$ is above the hyperplane ($\mathrm{y}_n > 0$), estimated class label $\hat{s}_n = 1$ , whereas if $\mathrm{y}_n$ is below ($\mathrm{y}_n < 0$), $\hat{s}_n = -1$. The perpendicular distance $d$ of a point $\mathbf{x}_n$ to the hyperplane is

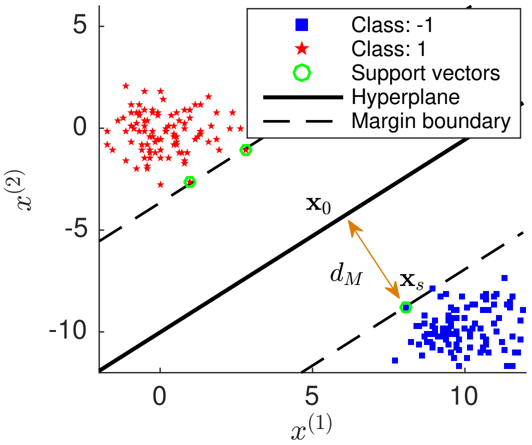

**Figure 4.2**: A linear hyperplane learned by training an SVM in two dimensions ($D = 2$).

the distance between the point $\mathbf{x}_n$ and its projection $\mathbf{x}_0$ on the hyperplane, satisfying

$$\begin{aligned} \mathbf{x}_n &= \mathbf{x}_0 + d\frac{\mathbf{w}}{||\mathbf{w}||}, \\ \mathbf{w}^T\mathbf{x}_0 + b &= 0, \end{aligned} \tag{4.16}$$

where $||\cdot||$ is the $l_2$ norm. From Eq. (4.16), the distance $d$ is obtained:

$$d(\mathbf{x}_n) = s_n \frac{\mathbf{w}^T\mathbf{x}_n + b}{||\mathbf{w}||}, \tag{4.17}$$

where $s_n$ is added in Eq. (4.17) to guarantee $d > 0$. The margin $d_M$ is defined as the distance from the hyperplane to the closest points $\mathbf{x}_s$ on the margin boundary (support vectors, see Fig. 4.2). The optimal $\mathbf{w}$ and $b$ are solved by maximizing the margin $d_M$:

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{argmax}} \quad d_M, \\ &\text{subject to } \frac{s_n(\mathbf{w}^T\mathbf{x}_n + b)}{||\mathbf{w}||} \geq d_M, \quad n = 1, \cdots, N. \end{aligned} \tag{4.18}$$

The Eq. (4.18) is equivalent to this optimization problem: [26]

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{argmin}} \quad \frac{1}{2}||\mathbf{w}||^2, \\ &\text{subject to } s_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, \quad n = 1, \cdots, N. \end{aligned} \tag{4.19}$$

If the training set is linearly non-separable (class overlapping), slack variables [26] $\xi_n \geq 0$ are introduced to allow some of the training points to be miclassified, corresponding the optimization problem:

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{argmin}} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{n=1}^{N}\xi_n, \\ &\text{subject to } s_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 - \xi_n, \quad n = 1, \cdots, N. \end{aligned} \tag{4.20}$$

The parameter $C > 0$ controls the trade-off between the slack variable penalty and the margin.

For the non-linear classification problems, the kernel trick [26] is used to allow data linearly separable in feature space. For this study, we use the Gaussian radial basis function (RBF) kernel: [20]

$$k_\phi(\mathbf{x}, \mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2). \tag{4.21}$$

$\gamma$ is a parameter that controls the kernel shape.

Support vector regression (SVR) is similar to SVM, but it minimizes the $\varepsilon$–sensitive error function

$$\mathcal{E}_\varepsilon(\mathrm{y}_n - r_n) = \begin{cases} 0, & \text{if } |\mathrm{y}_n - r_n| < \varepsilon, \\ |\mathrm{y}_n - r_n| - \varepsilon, & \text{otherwise,} \end{cases} \tag{4.22}$$

where $r_n$ is the true source range at sample $n$ and $\varepsilon$ defines a region on either side of the hyperplane. In SVR, the support vectors are points outside the $\varepsilon$ region.

Because the SVM and SVR models are a two-class models, multi-class SVM with $K$ classes is created by training $K(K-1)/2$ models on all possible pairs of classes. The points

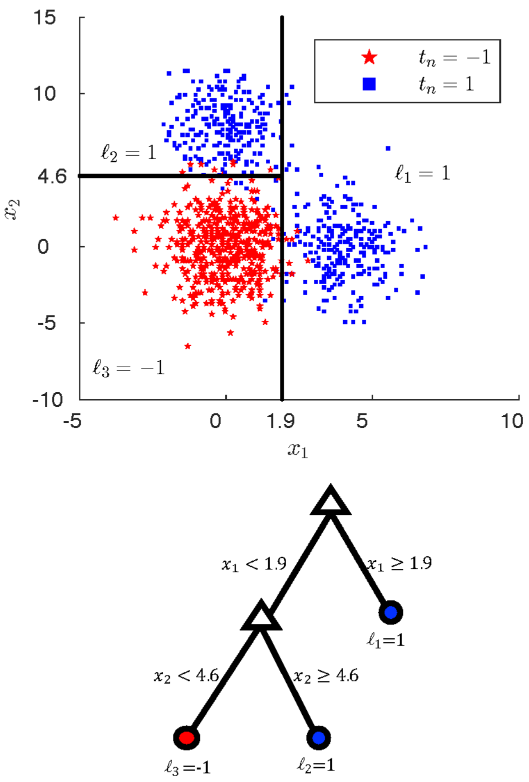

**Figure 4.3**: (Color online) Decision tree classifier and corresponding rectangular regions shown for two–dimensional data with $K = 2$ classes ($D = 2$, $M = 3$) and 1000 training points.

that are assigned to the same class most frequently are considered to comprise a single class, and so on until all points are assigned a class from 1 to $K$. This approach is known at the “one-versus-one” scheme, [26] although slight modifications have been introduced to reduced computational complexity. [27]

### 4.2.5 Random forests

The random forest (RF) [28] classifier is a generalization of the decision tree model, which greedily segments the input data into a predefined number of regions. The simple decision tree model is made robust by randomly training subsets of the input data and averaging over multiple models in RF.

Consider a decision tree (see Fig. 4.3) trained on all the input data. Each input sample,

$\mathbf{x}_n, n = 1, ..., N$, represents a point in $D$ dimensions. The input data can be partitioned into two regions by defining a cutoff along the $i$th dimension, where $i$ is the same for all input samples $\mathbf{x}_n, n = 1, ..., N$:

$$\begin{aligned} \mathbf{x}_n &\in \boldsymbol{x}_{\text{left}} && \text{if } x_{ni} > c, \\ \mathbf{x}_n &\in \boldsymbol{x}_{\text{right}} && \text{if } x_{ni} \leq c. \end{aligned} \tag{4.23}$$

$c$ is the cutoff value, and $\boldsymbol{x}_{\text{left}}$ and $\boldsymbol{x}_{\text{right}}$ are the left and right regions, respectively. The cost function, $G$, that is minimized in the decision tree at each branch is [20]

$$\begin{aligned} c^* &= \underset{c}{\text{argmin}}\, G(c), \\ G(c) &= \frac{n_{\text{left}}}{N} H(\boldsymbol{x}_{\text{left}}) + \frac{n_{\text{right}}}{N} H(\boldsymbol{x}_{\text{right}}), \end{aligned} \tag{4.24}$$

where $n_{\text{left}}$ and $n_{\text{right}}$ are the numbers of points in the regions $\boldsymbol{x}_{\text{left}}$ and $\boldsymbol{x}_{\text{right}}$. $H(\cdot)$ is an impurity function chosen based on the problem.

For the classification problem, the Gini index [20] is chosen as the impurity function

$$H(\boldsymbol{x}_m) = \frac{1}{n_m} \sum_{\mathbf{x}_n \in \boldsymbol{x}_m} I(t_n, \ell_m) \left[ 1 - \frac{1}{n_m} I(t_n, \ell_m) \right], \tag{4.25}$$

where $n_m$ is the number of points in region $\boldsymbol{x}_m$ and $\ell_m$ represents the assigned label for each region, corresponding to the most common class in the region: [20]

$$\ell_m = \underset{r_k}{\text{argmax}} \sum_{\mathbf{x}_n \in \boldsymbol{x}_m} I(t_n, r_k). \tag{4.26}$$

In Eq. (4.26), $r_k, k = 1, ..., K$ are the source range classes and $t_n$ is the label of point $\mathbf{x}_n$ in region $m$, and

$$I(t_n, r_k) = \begin{cases} 1 & \text{if } t_n = r_k, \\ 0 & \text{otherwise.} \end{cases} \tag{4.27}$$

The remaining regions are partitioned iteratively until regions $\boldsymbol{x}_1, ..., \boldsymbol{x}_M$ are defined. In this paper, the number of regions, $M$, is determined by the minimum number of points allowed in a region. A diagram of the decision tree classifier is shown in Fig. 4.3. The samples are partitioned into $M = 3$ regions with the cutoff values 1.9 and 4.6.

For RF regression, there are two differences from classification: the estimated class for each region is defined as the mean of the true class for all points in the region, and the mean squared error is used as the impurity function

$$\begin{aligned} \ell_m &= \frac{1}{n_m} \sum_{\mathbf{x}_n \in \boldsymbol{x}_m} r_n, \\ H(\boldsymbol{x}_m) &= \sum_{\mathbf{x}_n \in \boldsymbol{x}_m} (\ell_m - r_n)^2, \end{aligned} \tag{4.28}$$

where $r_n$ is source range at sample $n$.

As the decision tree model may overfit the data, statistical bootstrap and bagging are used to create a more robust model, a random forest. [29] In a given draw, the input data, $\mathbf{x}_i, i = 1, \cdots, Q$, is selected uniformly at random from the full training set, where $Q \leq N$. $B$ such draws are conducted with replacement and a new decision tree is fitted to each subset of data. Each point, $\mathbf{x}_n$, is assigned to its most frequent class among all draws:

$$\hat{f}^{\text{bag}}(\mathbf{x}_n) = \underset{t_n}{\operatorname{argmax}} \sum_{b=1}^{B} I(\hat{f}^{\text{tree},b}(\mathbf{x}_n), t_n), \tag{4.29}$$

where $\hat{f}^{\text{tree},b}(\mathbf{x}_i)$ is the class of $\mathbf{x}_i$ for the $b$th tree.

### 4.2.6 Performance metric

To quantify the prediction performance of the range estimation methods, the mean absolute percentage error (MAPE) over $N$ samples is defined as

$$E_{\text{MAPE}} = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{Rp_i - Rg_i}{Rg_i} \right|, \tag{4.30}$$

where $Rp_i$ and $Rg_i$ are the predicted range and the ground truth range, respectively. MAPE is preferred as an error measure because it accounts for the magnitude of error in faulty range estimates as well as the frequency of correct estimates. MAPE is known to be an asymmetric error measure [30] but is adequate for the small range of outputs considered.

### 4.2.7 Source localization algorithm

The localization problem solved by machine learning is implemented as follows:

1. Data preprocessing. The recorded pressure signals are Fourier transformed and $N_s$ snapshots form the SCM from which the input $\mathbf{x}$ is formed.

2. Division of preprocessed data into training and test data sets. For the training data, the labels are prepared based on different machine learning algorithms.

3. Training the machine learning models. $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_N]$ are used as the training input and the corresponding labels as the desired output.

4. Prediction on unlabeled data. The model parameters trained in step 3 are used to predict the source range for test data. The resulting output is mapped back to range, and the prediction error is reported by the mean absolute percentage error.

## 4.3 Simulations

In this section, the performance of machine learning on simulated data is discussed. For brevity, only the FNN classifier is examined here, although the conclusions apply to SVM and RF. Further discussion of SVM and RF performance is included in Secs. 4.4 and 4.5.

### 4.3.1 Environmental model and source-receiver configuration

Acoustic data is simulated using KRAKEN [31] with environmental parameters simulating the Noise09 experiment [32], see Fig. 4.4(a). The source frequency is 300 Hz. The source depth

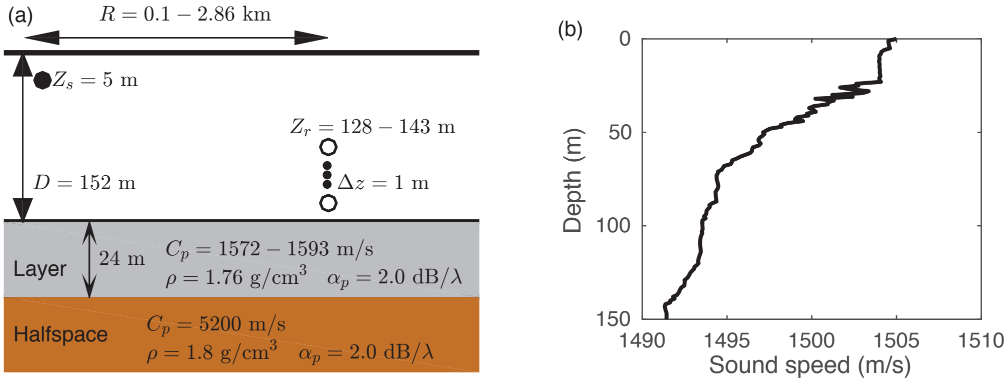

**Figure 4.4**: (Color online) (a) Waveguide parameters and source-receiver configuration. (b) Sound speed profile of water column.

is 5 m in a 152 m waveguide, with a 24 m sediment layer (sound speed 1572–1593 m/s, density 1.76 $\mathrm{g/cm}^3$, attenuation coefficient 2.0 $\mathrm{dB}/\lambda$) and a fluid halfspace bottom (sound speed 5200 m/s, density 1.8 $\mathrm{g/cm}^3$, and attenuation coefficient 2.0 $\mathrm{dB}/\lambda$). The sound speed profile of water column is shown in Fig. 4.4(b). The vertical array consists of 16 receivers spanning 128–143 m depth with inter-sensor spacing 1 m.

A source traveling away from the receiver at 2 m/s is simulated by varying the range from 0.1 to 2.86 km at 2 m intervals. Realizations with different SNRs are generated by adding appropriate complex Gaussian noise to the simulated received complex pressure signals.

Since the source moves in range and the source level is assumed constant, SNR is defined at the most distant range bin

$$\mathrm{SNR} = 10\log_{10}\frac{\sum_{l=1}^{L}|\hat{p}_l|^2/L}{\sigma^2}(\mathrm{dB}), \tag{4.31}$$

where $\hat{p}_l$ is sound pressure signal received by the $l$th sensor at the longest source-receiver distance and $\sigma^2$ represents the noise variance.

### 4.3.2 Input preprocessing and learning parameters

The SCM for a 16–element vertical array is formed at each range point by averaging over $N_s = 10$ successive snapshots (9 snapshots overlapped) according to Eq. (4.3). The number of neurons in the input layer is therefore $D = 16 \times (16+1) = 272$. The range sample interval is 2 m, with 1380 total range samples (1 s duration per snapshot). Thus, a total of $N = 1380$ input matrices constitute the sample set spanning the whole range 0.1–2.86 km.

For each SNR, two realizations of noisy measurements are generated. One realization of size 1380 × 272 is used for the training set. For the test set, the range sample interval is changed to 20 m, and a realization of size 138 × 272 is used as input.

In the test set, $K = 138$ output neurons represent ranges from 0.1–2.86 km incremented by 20 m. The number of neurons in the hidden layer is $M = 128$. To prevent overfitting, the ”keep dropout” technique, [25] with probability 0.5, is used. The initial learning rate for the Adam optimizer [24] is 0.01 and the maximum number of iterations is 1000.

### 4.3.3 Results

The prediction performance is examined for four SNRs ($-10$, $-5$, 0, 5 dB). Figure 4.5 compares range predictions by FNN and the true ranges on test data. For the four SNRs tested, the MAPE for the FNN predictions is 20.6, 6.5, 0.2 and 0.0%, respectively.

As described in Sec. 4.2, the output $y_{nk}$ of FNN represents the probability distribution over a discrete set of possible ranges. To demonstrate the evolution of the probability distribution as the FNN is trained, $y_{nk}$ versus training steps is plotted in Fig. 4.6 for the signal with SNR 5 dB at range 1.5 km. After 300 training steps, the FNN output probability distribution resembles the target output.

In Fig. 4.7, the convergence of the FNN algorithm is investigated by plotting the cross entropy Eq. (4.12) versus the optimization step on training and test data. It shows that the FNN

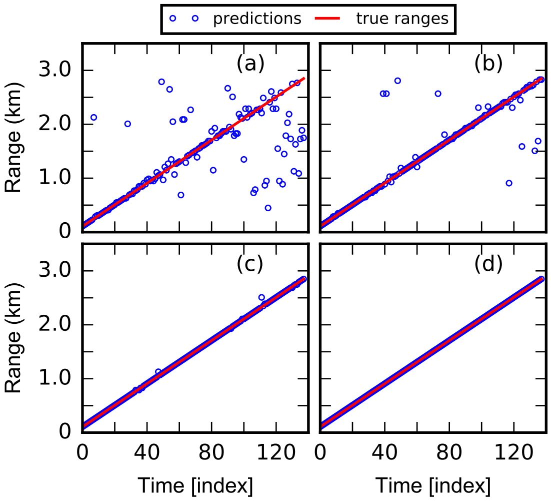

**Figure 4.5**: (Color online) Range predictions by FNN on test data set with SNR of (a) −10, (b) −5, (c) 0, and (d) 5 dB. The time index increment is 10 s.

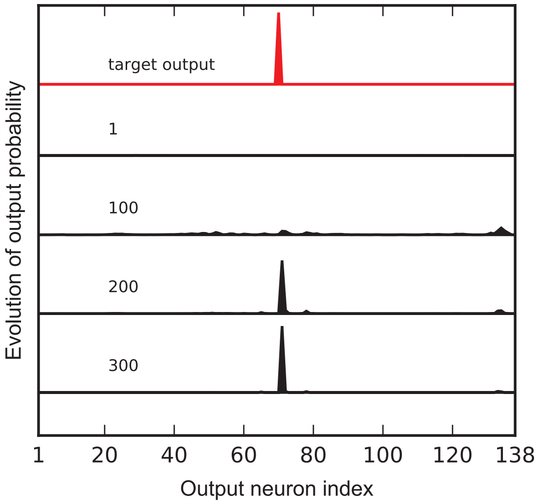

**Figure 4.6**: (Color online) Output probability for range 0.1–2.86 km (the true range is 1.5 km) after training steps (1, 100, 200, 300). The top line represents the target output.

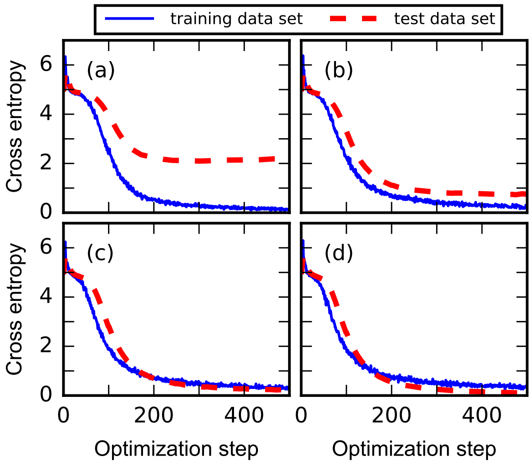

**Figure 4.7**: Cross entropy Eq. (4.11) versus optimization steps on training (solid) and test (dashed) data with SNR of (a) $-10$, (b) $-5$, (c) 0, and (d) 5 dB.

converges after about 300 steps at all SNRs. For low SNRs ($< 0$ dB), the FNN classifier generates poor predictions on test data while performing well on training data, which indicates overfitting.

Increasing the training set size can reduce overfitting but additional data may be not available due to experimental or computational constraints. For higher SNRs (e.g., 0 and 5 dB), both test and training errors converge to low cross entropy, indicating good performance. Therefore, best performance of machine learning methods is expected for high SNR.

## 4.4 Experimental results

Shipping noise data radiated by R/V New Horizon during the Noise09 experiment are used to demonstrate the performance of the FNN, SVM and RF localization. The experiment geometry is shown in Fig. 4.8, with bottom-moored vertical linear arrays (VLAs) indicated by triangles and the three ship tracks used for range estimation. The hydrophone sampling rate was 25 kHz.

The data from VLA2, consisting of 16 hydrophones at 1 m spacing, are used for range estimation. The frequency spectra of shipping noise recorded on the top hydrophone during the

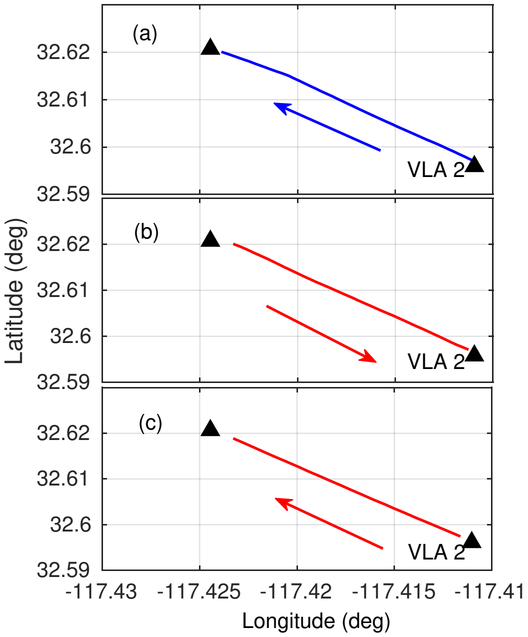

**Figure 4.8**: Ship tracks for Noise09 experiment during the periods (a) 01/31/2009, 01:43–02:05 (training data, ship speed 2 m/s), (b) 01/31/2009, 01:05–01:24 (Test-Data-1, ship speed −2 m/s), and (c) 02/04/2009, 13:41–13:51 (Test-Data-2, ship speed 4 m/s).

three periods are shown in Fig. 4.9. The striations indicate that the source was moving. The SNR decreases with increasing source-receiver distance.

Data from period 01:43–02:05 on January 31, 2009 are used as the training set and 01:05–01:24 on January 31 and 13:41–13:51 on February 4 are used as the test sets (Test-Data-1 and Test-Data-2).

The GPS antenna on the New Horizon is separated from the noise–generating propeller by a distance $L_d$. To account for this difference we use the range between the propeller and VLA2 as the ground truth range $R_g$:

$$R_g = \begin{cases} R_{\mathrm{GPS}} - L_d & \text{for training data and Test-Data-2,} \\ R_{\mathrm{GPS}} + L_d & \text{for Test-Data-1,} \end{cases} \tag{4.32}$$

where $R_{GPS}$ represents the range between the GPS antenna and VLA2. According to the R/V New

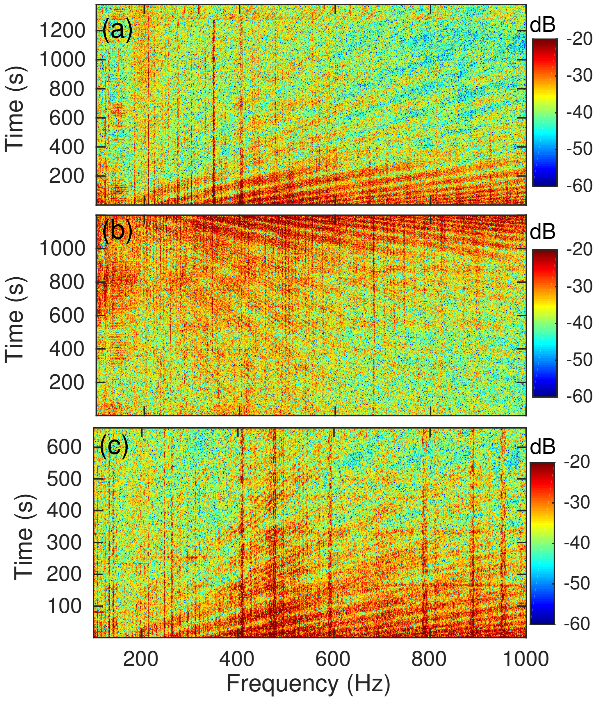

**Figure 4.9**: Spectra of shipping noise during periods (a) 01/31/2009, 01:43–02:05, (b) 01/31/2009, 01:05–01:24, and (c) 02/04/2009, 13:41–13:51.

Horizon handbook, $L_d = 24.5$ m. In the following, the ranges have been corrected by Eq. (4.32).

### 4.4.1 Input preprocessing and learning parameters

For the training set and both test sets, the $16 \times 16$ SCM at each range and frequency, averaged over 10 successive 1-s snapshots, is used as input. There are 1380 samples in the training data set and 120 samples in each of the test data sets (samples are drawn every 10 s for Test-Data-1 and 5 s for Test-Data-2). The source-receiver range 0.1–3 km is divided into $K = 138$ discrete range points.

As in the simulations in Sec. 4.3.2, the keep probability for training dropout of the FNN is 0.5, the initial learning rate is 0.01 and the maximum iteration step is 1000. The number of neurons in the hidden layer is chosen as $M = 128$ for 1 frequency and $M = 1024$ for 66 frequencies ( see Sec. 4.4.2).

For the SVM classifier, Gaussian radial basis function kernel is used. The parameters $\gamma$ (Eq. (4.21)) and $C$ (Eq. (4.20)) were tested over $[10^{-3} \quad 10^{-1}]$ and $[10 \quad 10^{3}]$, respectively. Values of $\gamma = 10^{-2}$ and $C = 10$ are found to be optimal.

For the RF method, the number of trees bagged is 500, with a minimum of 50 samples required for each leaf.

The performance of all test cases for the FNN, SVM, RF and conventional MFP is summarized in Tables 4.1 and 4.2.

### 4.4.2 SCM inputs

Because the shipping noise has a wide frequency band as seen from Fig. 4.9, the performance of the machine learning with the single and multi-frequency inputs is investigated. The FNN classifier is again used an example to illustrate the benefit of using multiple frequencies.

Input SCMs are formed at 550, 950, and 300–950 Hz with 10 Hz increments (66 frequencies). For the multi-frequency input, the SCMs are formed by concatenating multiple single-frequency SCM input vectors. For example, the dimension of a single frequency input sample is 272, whereas the multi-frequency input has a dimension $272 \times N_f$ for $N_f$ frequencies. The FNN is trained separately for each case and the source-receiver range is then predicted at the selected frequencies.

The prediction results on the two test data sets are shown in Figs. 4.10(a–f) along with $R_g$. For single frequency inputs, the minimum error is 12% (Fig. 4.10(d)) at 550 Hz and the highest error is 18% at 950 Hz (Fig. 4.10(e)), both on Test-Data-2. For multi-frequency inputs, the minimum prediction error is 6% on Test-Data-2. In general, the FNN predictions are better at close ranges due to higher SNR, as expected from the simulation results. However, the FNN with multi-frequency inputs performs well regardless of source range.

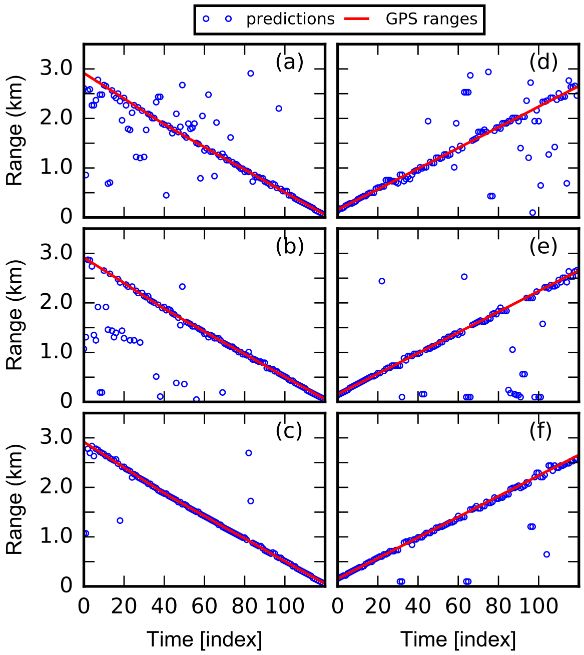

**Figure 4.10**: Range predictions on Test-Data-1 (a, b, c) and Test-Data-2 (d, e, f) by FNN. (a)(d) 550 Hz, (b)(e) 950 Hz, (c)(f) 300–950 Hz with 10 Hz increment, i.e. 66 frequencies. The time index increment is 10 s for Test-Data-1, and 5 s for Test-Data-2.

### 4.4.3 Source localization as a classification problem

Source localization is first solved as a classification problem. Only the best MAPE obtained by FNN (Sec. 4.2.3), SVM (Sec. 4.2.4) and RF (Sec. 4.2.5) is shown here (Fig. 4.11). These results are summarized in Table 4.1.

The lowest MAPE is achieved by the SVM, with 2% on both data sets. RF also reaches 2% MAPE for Test-Data-2 and 3% for Test-Data-1. FNN has 3% MAPE for both test sets. The performance of these three machine learning algorithms is comparable when solving range estimation as a classification problem.

The performance of these machine learning algorithms with various parameters (e.g. number of classes, number of snapshots and model hyper-parameters) is examined in Sec. 4.5.

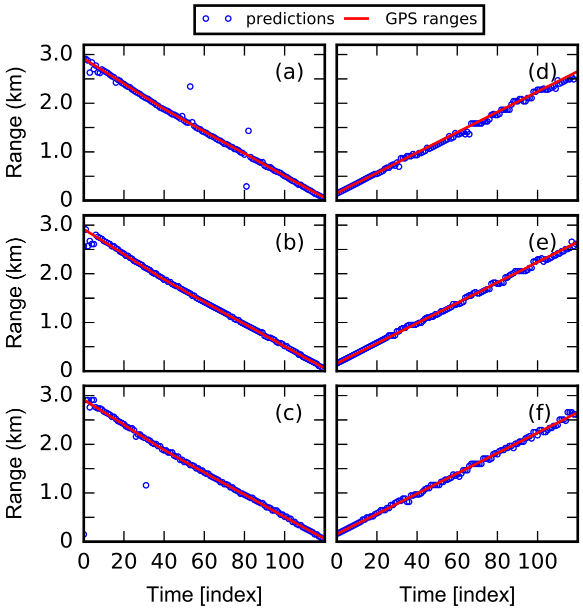

**Figure 4.11**: Source localization as a classification problem. Range predictions on Test-Data-1 (a, b, c) and Test-Data-2 (d, e, f) by FNN, SVM and RF for 300–950 Hz with 10 Hz increment, i.e. 66 frequencies. (a)(d) FNN classifier, (b)(e) SVM classifier, (c)(f) RF classifier.

### 4.4.4 Source localization as a regression problem

Source localization can be also solved as a regression problem. For this problem, the output represents the continuous range. In the training process, the input data remain the same, the labels are direct GPS ranges, and the weights and biases are trained using least-squares objective functions.

The range predictions by FNN with different number of hidden layers along with the GPS ranges are given in Fig. 4.12 showing that increasing number of hidden layers significantly reduces the error of FNN regression. Figure 4.13 shows the results of SVM (Fig. 4.13(a)(c)) and RF regressors (Fig. 4.13(b)(d)) on two data sets. For these methods, since additional layers cannot be added to increase the algorithmic complexity, the performance lags FNN. The best MAPE values for each regressor are shown in Table 4.1. Compared with classifiers, the FNN, SVM and RF degrade significantly for solving regression tasks.

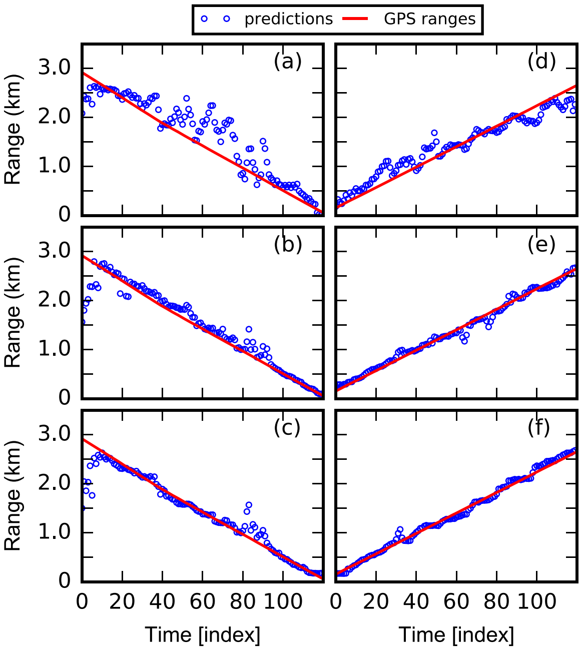

**Figure 4.12**: Source localization as a regression problem. Range predictions on Test-Data-1 (a, b, c) and Test-Data-2 (d, e, f) by FNN for 300–950 Hz with 10 Hz increment, i.e. 66 frequencies. (a)(d) 1 hidden layer, (b)(e) 2 hidden layers, (c)(f) 3 hidden layers. Each hidden layer consists of 512 neurons.

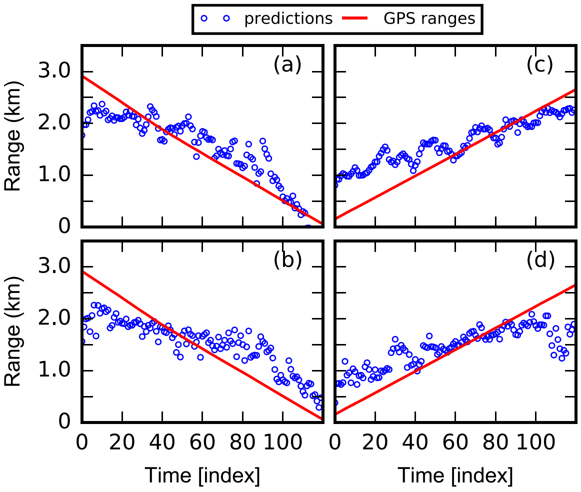

**Figure 4.13**: Source localization as a regression problem. Range predictions on Test-Data-1 (a, b) and Test-Data-2 (c, d) by SVM and RF for 300–950 Hz with 10 Hz increment, i.e. 66 frequencies. (a)(c) SVM for regression, (b)(d) RF for regression.

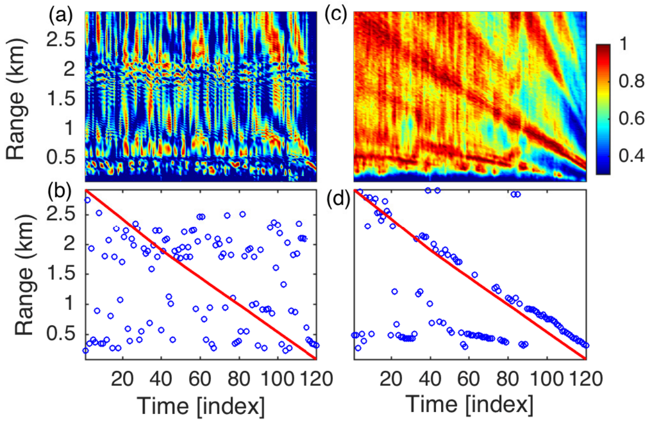

**Figure 4.14**: Localization using Bartlett matched-field processing based on synthetic replica fields on Test-Data-1. (a) ambiguity surface and (b) maximum peaks for 550 Hz, (c) ambiguity surface and (d) maximum peaks for 300–950 Hz with 10 Hz increment. Circles and solid lines denote predictions and GPS ranges respectively.

### 4.4.5 Conventional matched-field processing

The Bartlett MFP [1] is applied to Noise09 data for comparison. Two kinds of replica fields are used in the Bartlett processor. The first is generated by KRAKEN using the Noise09 environment in Fig. 4.4, with the corresponding ambiguity surfaces and maximum peaks shown in Fig. 4.14. We use the measured data (i.e. training data, 01/31/2009, 01:43–02:05) as the second group of replica fields as proposed in Ref. [3]. The results are shown in Fig. 4.15. For each case, both single frequency (550 Hz) and broadband (300–950 Hz) are considered.

From Figs. 4.14 and 4.15, the Bartlett MFP fails to determine source positions using a single frequency, while the FNN still generates a number of reasonable predictions (see Fig. 4.10(a)). Despite improved performance using broadband MFP, there are some errors due to sidelobes. The MAPE of MFP predictions is shown in Table 4.1. The minimum MAPE of Bartlett MFP is 19% on Test-Data-1 and 30% on Test-Data-2, which is much larger than the machine learning classifiers.

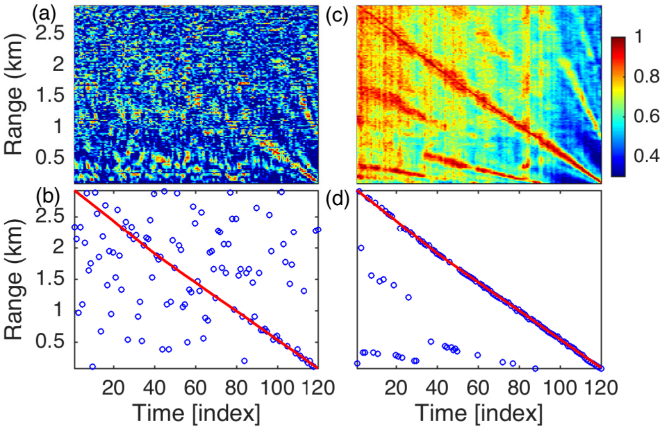

**Figure 4.15**: Localization using Bartlett matched-field processing (training data as replica fields) on Test-Data-1. (a) ambiguity surface and (b) maximum peaks for 550 Hz, (c) ambiguity surface and (d) maximum peaks for 300–950 Hz with 10 Hz increment. Circles and solid lines denote predictions and GPS ranges respectively.

**Table 4.1**: Best MAPE rate of FNN, SVM, RF and MFP predictions.

| Model | MAPE | |
|---|---|---|
| | Test-Data-1 (%) | Test-Data-2 (%) |
| FNN classifier | 3 | 3 |
| SVM classifier | 2 | 2 |
| RF classifier | 3 | 2 |
| FNN regressor | 10 | 5 |
| SVM regressor | 42 | 59 |
| RF regressor | 55 | 48 |
| MFP | 55 | 36 |
| MFP with measured replica | 19 | 30 |

## 4.5 Discussions

### 4.5.1 Range resolution

The number of classes, corresponding to the resolution of range steps, was varied to determine its effect on range estimation results. Previously (see Sec. 4.4) $K = 138$ classes were used, corresponding to a range resolution of 20 m. The MAPE for predictions with different numbers of output classes by FNN, SVM and RF classifiers is given in Table 4.2 with 10 snapshots averaged for each sample. These three classifiers perform well for all tested range resolutions.

### 4.5.2 Snapshots

The number of snapshots averaged to create the SCMs may also affect the performance. Increasing the number of snapshots makes the input more robust to noise, but could introduce mismatch if the source is moving or the environment is evolving across the averaging period. The range estimation methods are tested using 1, 5, and 20 snapshots and the corresponding MAPE is shown in Table 4.2. All of the three models degrade with 1 snapshot due to low SNR and become robust with more snapshots.

### 4.5.3 Number of hidden neurons and layers for FNN

The MAPE of FNN with different numbers of hidden neurons and layers is given in Table 4.2. FNN has the minimum error when the number of hidden neurons is chosen as 128 or 2048 for Test-Data-1 (7%) and 512 for Test-Data-2 (4%). From Table 4.2, the FNN with two hidden layers did not improve the prediction performance.

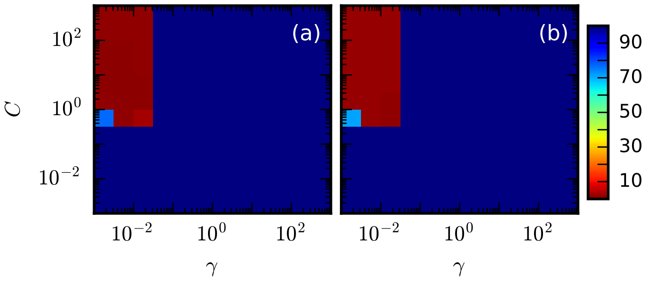

**Figure 4.16**: MAPE of the SVM classifier on (a) Test-Data-1 and (b) Test-Data-2. $\gamma$ is the kernel parameter and $C$ is the regularization parameter. There are 138 output classes and 10 snapshots averaged for each sample.

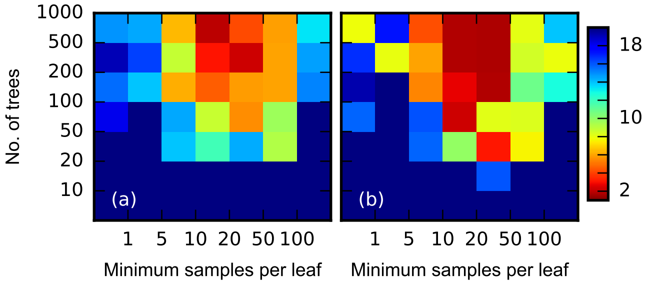

**Figure 4.17**: MAPE of the RF classifier versus the number of trees and the minimum samples per leaf on (a) Test-Data-1 and (b) Test-Data-2. There are 138 output classes and 10 snapshots averaged for each sample.

### 4.5.4 Kernel and regularization parameters for SVM

When using a Gaussian radial basis function kernel, the parameters $\gamma$ in Eq. (4.21) and the regularization parameter $C$ in Eq. (4.20) determine the best separation of the data by SVM. The MAPE versus these two parameters on two data sets is shown in Fig. 4.16. As seen from the result, there exists an optimal interval for these two parameters (i.e. $10 < C < 10^3$ and $10^{-3} < \gamma < 10^{-1}$). The SVM fails when $\gamma$ and $C$ are out of this interval, but is robust when $\gamma$ and $C$ are within the appropriate range.

### 4.5.5 Number of trees and minimum samples per leaf for RF

The number of decision trees and the minimum samples per leaf [20] are the most sensitive parameters for the RF. Figure 4.17 shows the MAPE versus these two parameters. The RF parameters have a smaller range of possible values than SVM, but the RF classifier will not fail for any of these choices. The RF classifier has the best performance for more than 500 trees and 20 to 50 minimum samples per leaf.

### 4.5.6 Multiple sources and deep learning

In our study, only one source is considered. The simultaneous multiple source localization problem is more challenging, especially for sources close to each other. Solving this problem with FNN is a multiple binary classification problem and will require additional training data.

Although the FNN with one hidden layer works well for the data sets in this paper, more complicated machine learning algorithms, e.g. deep learning, may be necessary for more complicated experimental geometries or ocean environments.

## 4.6 Conclusion

This paper presents an approach for source localization in ocean waveguides within a machine learning framework. The localization is posed as a supervised learning problem and solved by the feed-forward neural networks, support vector machines and random forests separately. Taking advantage of the modern machine learning library such as TensorFlow and Scikit-learn, the machine learning models are trained efficiently. Normalized sample covariance matrices are fed as input to the models. Simulations show that FNN achieves a good prediction performance for signals with SNR above 0 dB even with deficient training samples. Noise09 experimental data further demonstrates the validity of the machine learning algorithms.

**Table 4.2**: Parameter sensitivity of FNN, SVM and RF classifiers.

| Part I: FNN classifier | | | | | |
|---|---|---|---|---|---|
| # layers | Hidden nodes | # Classes | Snapshots | MAPE (%) | |
| 1 | 1024 | 1380 | 10 | 7 | 5 |
| 1 | 1024 | 690 | 10 | 3 | 6 |
| 1 | 1024 | 276 | 10 | 6 | 8 |
| 1 | 1024 | 138 | 10 | 8 | 6 |
| 1 | 1024 | 56 | 10 | 7 | 4 |
| 1 | 1024 | 28 | 10 | 10 | 4 |
| 1 | 1024 | 14 | 10 | 16 | 7 |
| 1 | 1024 | 138 | 1 | 10 | 5 |
| 1 | 1024 | 138 | 5 | 6 | 3 |
| 1 | 1024 | 138 | 20 | 8 | 3 |
| 1 | 64 | 138 | 10 | 9 | 9 |
| 1 | 128 | 138 | 10 | 7 | 7 |
| 1 | 256 | 138 | 10 | 8 | 6 |
| 1 | 512 | 138 | 10 | 8 | 4 |
| 1 | 2048 | 138 | 10 | 7 | 5 |
| 2 | 128 | 138 | 10 | 9 | 8 |
| 2 | 256 | 138 | 10 | 9 | 9 |
| 2 | 512 | 138 | 10 | 6 | 8 |
| Part II: SVM classifier | | | | | |
| $\gamma$ | $C$ | # Classes | Snapshots | MAPE (%) | |
| $10^{-2}$ | 10 | 1380 | 10 | 2 | 3 |
| | | 690 | 10 | 2 | 3 |
| | | 276 | 10 | 4 | 3 |
| | | 138 | 10 | 2 | 2 |
| | | 56 | 10 | 3 | 3 |
| | | 28 | 10 | 5 | 3 |
| | | 138 | 1 | 17 | 5 |
| | | 138 | 5 | 2 | 3 |
| | | 138 | 20 | 3 | 2 |
| Part III: RF classifier | | | | | |
| # Trees | Samples | # Classes | Snapshots | MAPE (%) | |
| 500 | 50 | 1380 | 10 | 4 | 10 |
| | | 690 | 10 | 3 | 4 |
| | | 276 | 10 | 3 | 3 |
| | | 138 | 10 | 3 | 2 |
| | | 56 | 10 | 9 | 5 |
| | | 28 | 10 | 13 | 9 |
| | | 138 | 1 | 20 | 15 |
| | | 138 | 5 | 6 | 5 |
| | | 138 | 20 | 3 | 2 |

The experimental results show that multi-frequency input generates more accurate predictions than single frequency (based on FNN). In addition, it shows that classification methods perform better than regression and MFP methods. We tested three classification methods (FNN, SVM, RF), all of which performed good with the best MAPE rate 2–3%. In the current study, the training and test data were from the same ship. In a realistic application, data from multiple ships of opportunity can be used as training data by taking advantage of the Automatic Identification System (AIS), a GPS system required on all cargo carriers. The tracks were quite similar and it would be interesting to test the performance as tracks deviate.

Machine learning is an attractive method for locating ocean sources because of its ability to learn features from data, without requiring sound propagation modeling. It can be used for unknown environments.

## 4.7 Acknowledgments

This work was supported by the Office of Naval Research under Grant No. N00014–1110439 and the China Scholarship Council.

The text of Chapter Four is in full a reprint of the material as it appears in Haiqiang Niu, Emma Reeves, and Peter Gerstoft, "Source localization in an ocean waveguide using supervised machine learning," *Journal of the Acoustical Society of America*, 142(3):1176–1188, 2017. The dissertation author was a major research contributor in Chapter 4. The contributions of the dissertation author in Chapter 4 include significant content writing and research direction. Coauthors listed in this publications were the primary researcher and supporting researcher that directed and supervised the research.

# Bibliography

[1] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt, *Computational Ocean Acoustics*, $2^{nd}$ ed. (Springer Science & Business Media, NewYork, 2011).

[2] S. E. Dosso and M. J. Wilmut, "Bayesian tracking of multiple acoustic sources in an uncertain ocean environment," J. Acoust. Soc. Am **133** (2013).

[3] L. T. Fialkowski, M. D. Collins, W. A. Kuperman, J. S. Perkins, L. J. Kelly, A. Larsson, J. A. Fawcett, and L. H. Hall, "Matched-field processing using measured replica fields," J. Acoust. Soc. Am **107**, 739–746 (2000).

[4] H. C. Song and C. Cho, "Array invariant-based source localization in shallow water using a sparse vertical array," J. Acoust. Soc. Am **141**, 183–188 (2017).

[5] R. Lefort, G. Real, and A. Drémeau, "Direct regressions for underwater acoustic source localization in fluctuating oceans," Appl. Acoust. **116**, 303–310 (2017).

[6] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal Proc. Mag **29**, 82–97 (2012).

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Adv. Neural Inf. Process. Syst 1097–1105 (2012).

[8] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," J. Mach. Learn. Res. **12**, 2493–2537 (2011).

[9] M. L. Sharma and M. K. Arora, "Prediction of seismicity cycles in the himalayas using artificial neural networks," Acta Geophysica Polonica. **53**, 299–309 (2005).

[10] N. Riahi and P. Gerstoft, "Using graph clustering to locate sources within a dense sensor array," Signal Processing **132**, 110–120 (2017).

[11] B. Z. Steinberg, M. J. Beran, S. H. Chin, and J. H. Howard, "A neural network approach to source localization," J. Acoust. Soc. Am **90**, 2081–2090 (1991).

[12] J. M. Ozard, P. Zakarauskas, and P. Ko, "An artificial neural network for range and depth discrimination in matched field processing," J. Acoust. Soc. Am **90**, 2658–2663 (1991).

[13] A. Caiti and T. Parisini, "Mapping ocean sediments by rbf networks," IEEE J. Ocean. Eng. **19**, 577–582 (1994).

[14] A. Caiti and S. Jesus, "Acoustic estimation of seafloor parameters: A radial basis functions approach," J. Acoust. Soc. Am **100**, 1473–1481 (1996).

[15] Y. Stephan, X. Demoulin, and O. Sarzeaud, “Neural direct approaches for geoacoustic inversion,” J. Comput. Acoust. **6**, 151–166 (1998).

[16] J. Benson, N. R. Chapman, and A. Antoniou, “Geoacoustic model inversion using artificial neural networks,” Inverse Problems **16**, 1627–1639 (2000).

[17] Z. H. Michalopoulou, D. Alexandrou, and C. Moustier, “Application of neural and statistical classifiers to the problem of seafloor characterization,” IEEE J. Ocean. Eng. **20**, 190–197 (1995).

[18] A. M. Thode, K. H. Kim, S. B. Blackwell, C. R. Greene, C. S. Nations, T. L. McDonald, and A. M. Macrander, “Automated detection and localization of bowhead whale sounds in the presence of seismic airgun surveys,” J. Acoust. Soc. Am **131**, 3726–3747 (2012).

[19] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, G. S. C. C. Citro, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, M. I. G. Irving, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” Software available from tensorflow.org (2015).

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in python,” J. Mach. Learn. Res **12**, 2825–2830 (2011).

[21] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, “Theano: a cpu and gpu math compiler in python,” in *Proc. 9th Python in Science Conf* (2010), pp. 1–7.

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in Proceedings of 22nd ACM international conference on Multimedia (2014), pp. 675–678.

[23] R. Collobert, S. Bengio, and J. Mariéthoz, “Torch: a modular machine learning software library,” , No. EPFL-REPORT-82802, Idiap (2002).

[24] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” Proc. of the 3rd ICLR (2014).

[25] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res. **15**, 1929–1958 (2014).

[26] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer Chaps. 4, 5, and 7, 2006).

[27] R. Fan, P. Chen, and C. Lin, “Working set selection using second order information for training support vector machines,” J. Mach. Learn Res. **6**, 1889–1918 (2005).

[28] L. Breiman, “Random forests,” Mach. Learn. **45**, 5–32 (2001).

[29] L. Breiman, “Bagging predictors,” Mach. Learn **24**, 123–140 (1996).

[30] P. Goodwin and R. Lawton, “On the asymmetry of the symmetric mape,” Int. J. Forecasting **15**, 405–408 (1999).

[31] M. B. Porter, “The kraken normal mode program,” (2009) `http://oalib.hlsresearch.com/Modes/AcousticsToolbox/manualtml/kraken.html`.

[32] S. H. Byun, C. M. A. Verlinden, and K. G. Sabra, “Blind deconvolution of shipping sources in an ocean waveguide,” J. Acoust. Soc. Am **141**, 797–807 (2017).

[33] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature **521**, 436–444 (2015).

[34] M. J. Bianco, P. Gerstoft, J. Traer, E. Ozanich, M. A. Roch, S. Gannot, and C.-A. Deledalle, “Machine learning in acoustics: Theory and applications,” J.Acoust. Soc. Am. **146**, 3590–3628 (2019).

[35] H. Niu, E. Reeves, and P. Gerstoft, “Source localization in an ocean waveguide using supervised machine learning,” Journal of the Acoustical Society of America **142(3)**, 1176–1188 (2017).

[36] H. Niu, E. Ozanich, and P. Gerstoft, “Ship localization in Santa Barbara Channel using machine learning classifiers,” J. Acoust. Soc. Am. **142**, EL455–EL460 (2017).

[37] Y. Wang and H. Peng, “Underwater acoustic source localization using generalized regression neural network,” J. Acoust. Soc. Am. **143**, 2321–2331 (2018).

[38] Z. Huang, J. Xu, Z. Gong, H. Wang, and Y. Yan, “Source localization using deep neural networks in a shallow water environment,” J. Acoust. Soc. Am. **143**, 2922–2932 (2018).

[39] R. Lefort, G. Real, and A. Drémeau, “Direct regressions for underwater acoustic source localization in fluctuating oceans,” App. Acoustics **116**, 303–310 (2017).

[40] H. Niu, Z. Gong, E. Ozanich, P. Gerstoft, H. Wang, and Z. Li, “Deep-learning source localization using multi-frequency magnitude-only data,” J.Acoust. Soc. Am. **146**, 211–222 (2019).

[41] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, “A learning-based approach to direction of arrival estimation in noisy and reverberant environments,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015), pp. 2814–2818.

[42] S. Chakrabarty and E. A. P. Habets, “Multi-speaker DOA estimation using deep convolutional networks trained with noise signals,” IEEE Journal of Selected Topics in Signal Processing **13**(1), 8–21 (2019).

[43] G. Izacard, B. Bernstein, and C. Fernandez-Granda, “A learning-based framework for line-spectra super-resolution,” CoRR **abs/1811.05844** (2018) `http://arxiv.org/abs/1811.05844`.

[44] G. Izacard, B. Bernstein, and C. Fernandez-Granda, “A learning-based framework for line-spectra super-resolution,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), pp. 3632–3636.

[45] W. Wang, H. Ni, L. Su, T. Hu, Q. Ren, P. Gerstoft, and L. Ma, “Deep transfer learning for source ranging: Deep-sea experiment results,” J. Acoust. Soc. Am. **146**(4), EL317–EL322 (2019).

[46] F. Chollet *et al.*, “Keras,” https://keras.io (2015).

[47] N. C. Raj, P. V. Aswathy, and K. V. Sagar, “Determination of angle of arrival using nonlinear support vector machine regressors,” in *2007 International Conference on Signal Processing, Communications and Networking* (2007), pp. 512–515, doi: `10.1109/ICSCN.2007.350652`.

[48] D. Salvati, C. Drioli, and G. L. Foresti, “A weighted MVDR beamformer based on SVM learning for sound source localization,” Pattern Recognition Letters **84**, 15–21 (2016).

[49] G. Lin, Y. Li, and B. Jin, “Research on support vector machines framework for uniform arrays beamforming,” in *2010 International Conference on Intelligent Computation Technology and Automation* (2010), Vol. 3, pp. 124–127, doi: `10.1109/ICICTA.2010.215`.

[50] M. M. Ramon, N. Xu, and C. G. Christodoulou, “Beamforming using support vector machines,” IEEE Antennas Wirel. Propag. Lett. **4**, 439–442 (2005) doi: `10.1109/LAWP.2005.860196`.

[51] M. Martinez-Ramon, J. L. Rojo-Alvarez, G. Camps-Valls, and C. G. Christodoulou, “Kernel antenna array processing,” IEEE Trans. Antennas Propag. **55**(3), 642–650 (2007) doi: `10.1109/TAP.2007.891550`.

[52] S. Nannuru, A. Koochakzadeh, K. L. Gemba, P. Pal, and P. Gerstoft, “Sparse Bayesian learning for beamforming using sparse linear arrays,” J. Acoust. Soc. Am. **144**(5), 2719–2729 (2018).

# Chapter 5

# A feedforward neural network for direction-of-arrival estimation

This chapter examines the relationship between conventional beamforming and linear supervised learning, then develops a nonlinear deep feedforward neural network (FNN) for direction-of-arrival (DOA) estimation. First, conventional beamforming is reformulated as a real-valued, linear inverse problem in the weight space, which is compared to support vector machine and a linear FNN model. In the linear formulation, DOA is quickly and accurately estimated for a realistic array calibration example. Then, a nonlinear feed-forward neural network (FNN) is developed for two-source DOA and for $K$-source DOA, where $K$ is unknown. Two training methodologies are used: exhaustive training for controlled accuracy, and random training for flexibility. The number of FNN model hidden layers, hidden nodes, and activation function are selected using a hyperparameter search. In plane wave simulations, the 2-source FNN resolved incoherent sources with $1^\circ$ resolution using a single snapshot, similar to Sparse Bayesian Learning (SBL). With multiple snapshots, $K$-source FNN achieved resolution and accuracy similar to Multiple Signal Classification (MUSIC) and SBL for an unknown number of sources. The practicality of the deep FNN model is demonstrated on Swellex96 experimental data for multiple source DOA on a horizontal acoustic array.

## 5.1 Introduction

Motivated by its notable success, [1] machine learning and deep learning have been used within a variety of acoustics domains [2], including source localization in underwater acoustics. [3–8]

The feed-forward neural network (FNN) has shown potential for direction-of-arrival (DOA) estimation in the time domain for single speaker noisy environments [9] and in the frequency domain for multiple speakers in noisy environments. [10] FNN and convolutional networks (CNN) have also been used for multiple-frequency spectral estimation. [11, 12] Underwater acoustic DOA estimation is challenged by array position uncertainty and source phase ambiguity. To overcome this problem, spectral covariance matrix estimates from a linear array were used to train an FNN for range localization. [3, 4, 8, 13]

This paper describes a feedforward network for DOA estimation of an arbitrary number of plane wave sources. First, Conventional Beamforming (CBF) is rewritten as linear in the covariance matrix of the replicas and the sample covariance matrix (SCM). The real, linear formulation of CBF can be solved directly for the replicas. Machine learning algorithms including support vector machine (SVM) and FNN can solve this linear problem. The method is applied to a perturbed array calibration example and solved using open-source software, [14] thus demonstrating that machine learning can be used as well as CBF by relying on representative data instead of an assumed physical model.

Second, motivated by nonlinear kernels used in SVM beamforming applications, [15–19] deep FNN is developed for DOA estimation. The FNN is trained with SCMs, allowing for the addition of array preprocessing. For two sources, the performance of the network is compared on coherent and incoherent sources for varying signal-to-noise ratio (SNR). The number of hidden nodes and hidden layers is found by hyperparameter search. Then, a $K$-source FNN is trained by randomly selecting a large training set. The performance of both methods are shown on simulations against Sparse Bayesian Learning [20] and Mulitple Signal Classification (MUSIC) subspace method. The DOA performance of the two-source and $K$-source FNN models are demonstrated on sources in the Swellex96 experiment.

In Section 5.2, CBF is rewritten as a real-valued, linear problem. In Section 5.3, the theory of linear SVM, linear FNN, and deep FNN are introduced, with an example application for a perturbed array. In Section 5.4, an FNN for the two-source DOA estimation problem is presented, then extended to the $K$-source problem, and simulations are conducted to examine its performance compared to SBL and MUSIC. Section 5.5 demonstrates the real-world performance of both deep FNNs, SBL, and MUSIC (or CBF) using the Swellex96 data. Last, Section 5.6 mentions ongoing work using Convolutional Neural Networks (CNN) and the tradeoffs for selecting deep learning models.

## 5.2 Background and notation

In this section, the linearity of Conventional Beamforming (CBF) is expressed in terms of the SCM real and imaginary components using algebraic properties of the matrix trace. With this formulation, the spatial filters $\mathbf{w}$ can be solved by inversion techniques. This formulation also allows the creation of an SCM feature vector that is used with nonlinear machine learning techniques for high-resolution DOA estimation.

CBF is a spatial filtering method that estimates source angles by assuming plane wave propagation (for details, see e.g. DeFatta et al. [21]). Figure 5.1 shows an example of ocean waveguide propagation for a single underwater source, where the sea surface-generated noise and diffuse shipping noise is modeled as additive Gaussian noise.

For measured data, $\mathbf{p} \in \mathbb{C}^{N\times 1}$, the plane wave replicas $\mathbf{w} \in \mathbb{C}^{N\times 1}$ are often assumed, such

**Table 5.1**: Symbolic and mathematical notation. $()^T$ is the matrix transpose and $()^H$ the Hermitian transpose, Tr$\{\}$ refers to the trace of a square matrix, a$||\cdot||_2^2$ the $\ell 2$ norm and $||\cdot||_F^2$ the Frobenius norm.

| Symbol | Cardinality | Definition |
|---|---|---|
| **CBF** | | |
| $\mathbf{a}(\theta_m)$ | $\mathbb{C}^{N\times 1}$ | Replica vector for a 1D array with $N$ sensors |
| $\mathbf{A}_{\text{cov}} = \mathbf{a}(\theta_m)\mathbf{a}(\theta_m)^H$ | $\mathbb{C}^{N\times N}$ | Covariance matrix of the replica vectors |
| $\mathbf{a}_{\text{cov}}$ | $\mathbb{R}^{2N^2\times 1}$ | Vectorized replica covariance matrix |
| $B(\theta_m)$ | | Scalar CBF output at candidate angle $\theta_m$ |
| $l = [1,...,L]$ | | Snapshot index, the $l$th spectral estimate |
| $\mathbf{n} \sim \mathcal{CN}(\mathbf{0},\sigma^2)$ | $\mathbb{C}^{N\times 1}$ | Complex Gaussian noise with variance $\sigma^2$ on an $N$-sensor array |
| $\mathbf{p}_l(f)$ | $\mathbb{C}^{N\times 1}$ | $N\times 1$ complex pressure vector at frequency $f$ and snapshot $l$ |
| $\mathbf{P} = \frac{1}{L}\sum_{l=1}^{L}\mathbf{p}_l\mathbf{p}_l^H$ | $\mathbb{C}^{N\times N}$ | SCM at time $t$, estimated over $L$ snapshots |
| $s_k(f)$ | | Source amplitude at frequency $f$ |
| $\mathbf{x}, \mathbf{x}_t$ | $\mathbb{R}^{2N^2\times 1}$ | Vectorized SCM or vectorized SCM at time $t$ (for samples in a data set) |
| $\mathbf{x}^u, \mathbf{b}$ | $\mathbb{R}^{N(N+1)\times 1}$ | Vectorized upper triangular and diagonal SCM at time $t$ |
| $x^n, y^n$ | | Horizontal and vertical positions of the $n$th array sensor |
| $\theta_k$ | | DOA of the $k$th source, $k = 1,...,K$ |
| $\theta_m$ | | Candidate DOA, where $m = 1,...,M$ |
| $\theta_{t,k}, \hat{\theta}_t^k$ | | True and estimated angle of the k$th$ source at time $t$ |
| **SVM, FNN** | | |
| $a^s$ | | Activation of an FNN hidden layer |
| $\alpha^m, \mu^m$ | | Lagrange constants for the $m$th class at sample $t$ |
| $D = N^2 + N$ | | Dimension of the feature vector (number of features) |
| $\xi^m$ | | SVM slack variable for the $m$th class at sample $t$ |
| $S$ | | Number of hidden nodes in a hidden layer |
| t | $\{-1,1\}$ | Target label for the SVM |
| $\mathbf{w}^m$ | $\mathbb{R}^{D\times 1}$ | SVM hyperplane for class $m$, or FNN weight vector for the $m$th output class |
| $\mathbf{W}^M$ | $\mathbb{R}^{S\times M}$ | Matrix of FNN weight vectors from $S$ hidden nodes to $M$ outputs |
| $\mathbf{y}, \mathbf{y}_t$ | $\{0,1\}^{M\times 1}$ | True label across $M$ classes and true label at time $t$ (for samples in a data set) |
| $\hat{\mathbf{y}}_t$ | $\mathbb{R}^{M\times 1}$ | Estimated label across $M$ classes at time $t$ |
| **SBL** | | |
| $\mathbf{A} = [\mathbf{a}(\theta_1),...,\mathbf{a}(\theta_M)]$ | $\mathbb{C}^{N\times M}$ | Replica matrix. |
| $\mathbf{\Gamma} = \text{diag}(\gamma)$ | $\mathbb{C}^{N\times N}$ | SBL signal covariance matrix |
| $\sigma^2$ | | Gaussian noise variance |
| $\mathbf{Y} = [\mathbf{p}_1,...,\mathbf{p}_L]$ | $\mathbb{C}^{N\times L}$ | Matrix of observation vectors at $L$ snapshots |

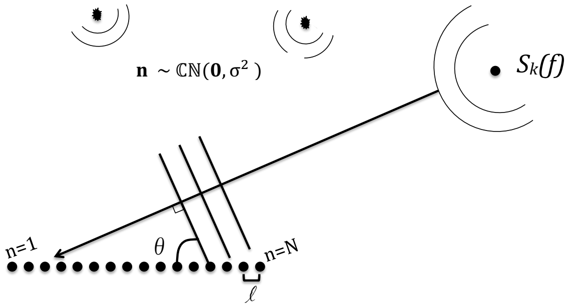

**Figure 5.1**: Single source received on a horizontal line array. The direct sound propagation is planar. Noise and distant sources are modeled as additive Gaussian noise.

that for $K$ sources each with frequency $f$, angle $\theta_k$, and phase speed $c$ on an $N$-element array spaced at $\ell$,

$$a^n(\theta_k) = \frac{1}{\sqrt{N}} e^{i\frac{2\pi f}{c}(n-1)\ell\sin(\theta_k)},\ n = 1,...,N. \tag{5.1}$$

$$\mathbf{a}(\theta_k) = [a^1(\theta_k),...,a^N(\theta_k)]^T, \mathbf{n} \sim \mathcal{CN}(0,\sigma^2),$$

$$\mathbf{p}_l = \sum_{k=1}^{K} s_k(f)\mathbf{a}(\theta_k) + \mathbf{n}, \tag{5.2}$$

where $n$ is the element index ($n = 1$ at first element) and $s_k(f)$ is the $k$th complex-valued source amplitude at frequency $f$. The noise is assumed complex Gaussian with variance $\sigma^2$.

For data collected at the $l$th snapshot, $\mathbf{p}_l$, the SCM across $L$ snapshots is written as $\mathbf{P}$, see (5.5). In Sec. 5.4, the effect of varying $L$ is studied by simulation. The plane wave CBF result is

$$\begin{aligned} B(\theta_m) =& \frac{1}{L}\sum_{l=1}^{L} |\mathbf{a}^H(\theta_m)\mathbf{p}_l|^2 \\ =& \frac{1}{L}\sum_{l=1}^{L} \mathrm{Tr}\{\mathbf{a}^H(\theta_m)\mathbf{p}_l\mathbf{p}_l^H\mathbf{a}(\theta_m)\} \\ =& \mathrm{Tr}\{\mathbf{a}^H\mathbf{P}\mathbf{a}\} = \mathrm{Tr}\{\mathbf{a}\mathbf{a}^H\mathbf{P}\} = \mathrm{Tr}\{\mathbf{A}_{\text{cov}}^H\mathbf{P}\}, \end{aligned} \tag{5.3}$$

where $\mathbf{a}$, equivalent to $\mathbf{a}(\theta_m)$, is the plane wave replica vector at a candidate angle $\theta_m$.

$$\mathbf{A}_{\text{cov}} = \mathbf{A}^{\text{R}}_{\text{cov}} + i\mathbf{A}^{\text{I}}_{\text{cov}} = \mathbf{a}(\theta_m)\mathbf{a}^H(\theta_m), \tag{5.4}$$

$$\begin{aligned}\mathbf{P} &= \frac{1}{L}\sum_{l=1}^{L}\mathbf{p}_l\mathbf{p}_l^H = \mathbf{P}^{\text{R}} + i\mathbf{P}^{\text{I}} \\ &= \mathbf{P}^{R}_{\text{u}} + \mathbf{P}^{\text{R}\,T}_{\text{u}} + i(\mathbf{P}^{\text{I}}_{\text{u}} - \mathbf{P}^{\text{I}\,T}_{\text{u}}) + \mathbf{P}^{\text{R}}_{\text{d}}.\end{aligned} \tag{5.5}$$

$\mathbf{A}^{\text{R}}_{\text{cov}}$ and $\mathbf{A}^{\text{I}}_{\text{cov}}$ are the real and imaginary components of (5.4), likewise for $\mathbf{P}$. $\mathbf{P}^{\text{R}}_{\text{u}}$ is an upper triangular matrix and $\mathbf{P}^{\text{R}}_{\text{d}}$ a diagonal matrix. Defining $B(\theta_m)$ in terms of $\mathbf{P}$ and $\mathbf{A}_{\text{cov}}$ allows their Hermitian properties to be exploited:

$$\begin{aligned}\text{Tr}\{\mathbf{A}^{\text{R}\;T}_{\text{cov}}\mathbf{P}^{\text{I}}\} &= \text{Tr}\{\mathbf{A}^{\text{R}\;T}_{\text{cov}}(\mathbf{P}^{\text{I}}_{\text{u}} - \mathbf{P}^{\text{I}\,T}_{\text{u}}) + \mathbf{A}^{\text{R}}_{\text{cov}}\mathbf{P}^{\text{I}}_{\text{d}}\} \\ &= \text{Tr}\{\mathbf{A}^{\text{R}\;T}_{\text{cov}}(\mathbf{P}^{\text{I}}_{\text{u}} - \mathbf{P}^{\text{I}\,T}_{\text{u}})\} + 0 = 0.\end{aligned} \tag{5.6}$$

Likewise, $\text{Tr}\{(\mathbf{A}^{\text{I}}_{\text{cov}})^T\mathbf{P}^{\text{R}}\} = 0$. Rewriting (5.3),

$$\text{Tr}\{\mathbf{A}^H_{\text{cov}}\mathbf{P}\} = \text{Tr}\{\mathbf{A}^{\text{R}\;T}_{\text{cov}}\mathbf{P}^{\text{R}}\} + \text{Tr}\{\mathbf{A}^{\text{I}\;T}_{\text{cov}}\mathbf{P}^{\text{I}}\} \tag{5.7}$$

Defining vectorization of $\mathbf{A}$ as $\text{vec}(\mathbf{A}) = [A_{11}, ..., A_{N1}, A_{12}, ..., A_{NN}]$. Then,

$$\begin{aligned}B(\theta_m) &= \text{vec}(\mathbf{A}^{\text{R}}_{\text{cov}})^T\text{vec}(\mathbf{P}^{\text{R}}) + \text{vec}(\mathbf{A}^{\text{I}}_{\text{cov}})^T\text{vec}(\mathbf{P}^{\text{I}}) \\ &= \mathbf{a}^T_{\text{cov}}\mathbf{x},\end{aligned} \tag{5.8}$$

$$\begin{aligned}\mathbf{a}_{\text{cov}} &= [\text{vec}(\mathbf{A}^{\text{R}}_{\text{cov}}), \text{vec}(\mathbf{A}^{\text{I}}_{\text{cov}})] \in \mathbb{R}^{N^2\times 1} \\ \mathbf{x} &= [\text{vec}(\mathbf{P}^{\text{R}}), \text{vec}(\mathbf{P}^{\text{I}})] \in \mathbb{R}^{N^2\times 1}\end{aligned} \tag{5.9}$$

A secondary benefit of expressing CBF in terms of $\mathbf{P}$ and $\mathbf{A}_{\text{cov}}$ is that, keeping only the upper diagonal terms, the number of features in the input vector is reduced from $2N^2$ to $N^2$. This reduced formulation has been used for FNN range estimation. [3]

For CBF, the estimated DOA is found by

$$\hat{\theta}_k = \arg\max_{\theta_m} B(\theta_m). \tag{5.10}$$

If data $\mathbf{x}$ ((5.9), derived from $\mathbf{P}$) is known to correspond to the true angle $\theta_m$, then the optimal replicas vectors can be obtained from Eq. (5.8) by replacing $\mathbf{a}_{\text{cov}}$ with a variable $\mathbf{w} \in \mathbb{C}^{N^2 \times 1}$. A Lagrange multiplier $\lambda$ is introduced to satisfy $\|\mathbf{w}\|_2^2 = 1$, based on the CBF weight vectors in (5.1),

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left\{ -\mathbf{w}^T \mathbf{x} + \lambda(\|\mathbf{w}\|_2^2 - 1) \right\}. \tag{5.11}$$

The minimum is found by differentiating (5.11) with respect to $\mathbf{w}$ and $\lambda$, and finding the stationary points:

$$\frac{\partial}{\partial \mathbf{w}} \left[ -\mathbf{w}^T \mathbf{x} + \lambda(\|\mathbf{w}\|_2^2 - 1) \right] = -\mathbf{x} + 2\lambda \mathbf{w} = \mathbf{0} \tag{5.12}$$

$$\frac{\partial}{\partial \lambda} \left[ -\mathbf{w}^T \mathbf{x} + \lambda(\||\mathbf{w}\|_2^2 - 1) \right] = \|\mathbf{w}\|_2^2 - 1 = 0. \tag{5.13}$$

Combining (5.12) and (5.13) and solving for $\mathbf{w}$, an estimate of the covariance weight vector, $\hat{\mathbf{w}}$, is obtained,

$$\hat{\mathbf{w}} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}. \tag{5.14}$$

Under the plane wave assumption, the optimal covariance weights (5.14) for CBF for a single incoming plane wave are the normalized covariance data $\mathbf{x}$ corresponding to the plane wave direction, as in (5.9).

## 5.3 Machine learning Theory

The data-driven methods of Support Vector Machine (SVM) and Feed-forward Neural Network (FNN) are here introduced as architectures for estimating DOA. First, the linear formulations of SVM and FNN are presented for solving the linear CBF in Sec. 5.2. Then, the nonlinear formulation of FNN is developed.

### 5.3.1 Feed-forward neural network

The FNN is an inference model [22] that transforms its inputs using a set of weights and activation functions. The weights are first trained on a set of $T$ examples, $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_t, ..., \mathbf{x}_T]$, whose training labels, or classes, are known. The training labels for $M$ classes are $M$-dimensional binary vectors, $\mathbf{y}_t = [y_t^1, ..., y_t^M]$, where

$$y_t^m = \begin{cases} 1 & m = m_{\text{true}} \\ \\ 0 & m \neq m_{\text{true}} \end{cases}, \quad m = 1, ..., M \tag{5.15}$$

for an input class $m_{\text{true}}$. The output of FNN is a likelihood-like distribution, at each time $t$, over $M$ classes, $\hat{\mathbf{y}}_t = [\hat{y}_t^1, ..., \hat{y}_t^M]$, with

$$\hat{y}_t^m = f\left(\sum_{i=1}^{D} w^{i,m} x^i\right) = f(\mathbf{w}^{m^T} \mathbf{x}_t), \quad m = 1, ..., M \tag{5.16}$$

where $f(\cdot)$ is an arbitrary function and $\mathbf{w}^m$ a weight vector. The locally optimal set of weights $\mathbf{w}^m$, $m = 1, ..., M$, is estimated through inversion by minimizing a cost function, $J$, over $t = 1, ..., T$ input samples,

$$\hat{\mathbf{w}}^m = \underset{\mathbf{w}^m}{\arg\min} \left\{ -\sum_{t=1}^{T} (J(\mathbf{y}_t, \hat{\mathbf{y}}_t(\mathbf{w}^m, \mathbf{x}_t))) \right\}. \tag{5.17}$$

### Linear FNN

For a single set of weights, $\mathbf{w}^m$, with linear activation $f(z)$, e.g. unity, $f(z) = z$, FNN becomes a linear regression. Using the inputs $\mathbf{x}_t$ from (5.9), this becomes

$$\hat{y}_t^m = \mathbf{w}^{mT}\mathbf{x}_t, \quad m = 1, ..., M. \tag{5.18}$$

For comparison to (5.8), the linear FNN cost function is applied:

$$J(\mathbf{y}_t, \hat{\mathbf{y}}_t(\mathbf{x}_t)) = \mathbf{y}_t^T \hat{\mathbf{y}}_t(\mathbf{w}^m, \mathbf{x}_t). \tag{5.19}$$

Setting a constraint on the weight matrix, $\mathbf{W}^M = [\mathbf{w}^1, ..., \mathbf{w}^M]^T$, the solution for the $m$th weight class is

$$\hat{\mathbf{w}}^m = \operatorname*{arg\,min}_{\mathbf{w}^m} \left\{ -\sum_{t=1}^{T} \mathbf{w}^{mT}\mathbf{x}_t + \frac{\mu}{M} \left\| \mathbf{W}^M \right\|_F^2 \right\} \tag{5.20}$$

Assume that one example for each class may be used to train the FNN such that $m$ corresponds to $t$, $m \mathrel{\widehat{=}} t$ and $\mathbf{w}^m = \mathbf{w}^t$. Differentiating with respect to the $m$th weight,

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}^m} \Big( -\sum_{t=1}^{T} \mathbf{w}^{t\,\mathrm{T}}\mathbf{x}_t + \frac{\mu}{M} \left\| \mathbf{W}^T \right\|_F^2 \Big) &= -\mathbf{x}_m + \frac{2\mu}{M}\mathbf{w}^m \\ \hat{\mathbf{w}}^m &= \frac{M}{2\mu}\mathbf{x}_m, \end{aligned} \tag{5.21}$$

where $\mu$ controls the regularization.

For the linear FNN model, if the training data $\mathbf{x}_t$ from (5.9) have one training example at each angle such that $\mathbf{x}_t = \mathbf{x}_m$, then the optimal FNN weights from (5.21) are identical to the CBF covariance weights in (5.14) when $\mu = \frac{M}{2} \left\| \mathbf{p} \right\|_2$.

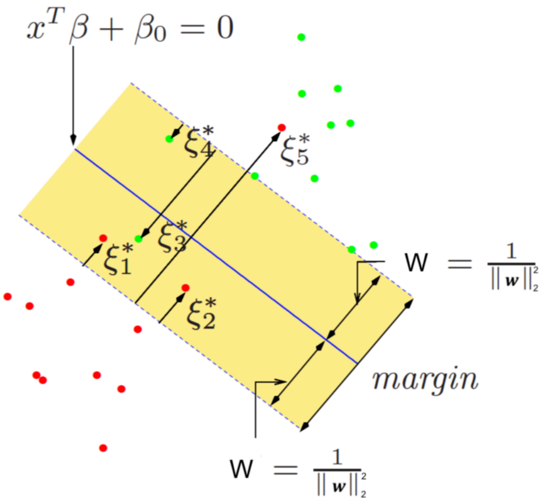

**Figure 5.2**: Example of SVM hyperplane **w** and slack variables $\xi_t^*$ ($\xi_t^* = \frac{\xi_t}{||\mathbf{w}||_2^2}$) in two dimensions for the non-separable case, from Hastie et al. (2008). [23]

### 5.3.2 Support Vector Machine

SVM optimally separates a set of training data into two labeled classes, [23]

$$t_t \in \{-1, 1\}, \tag{5.22}$$

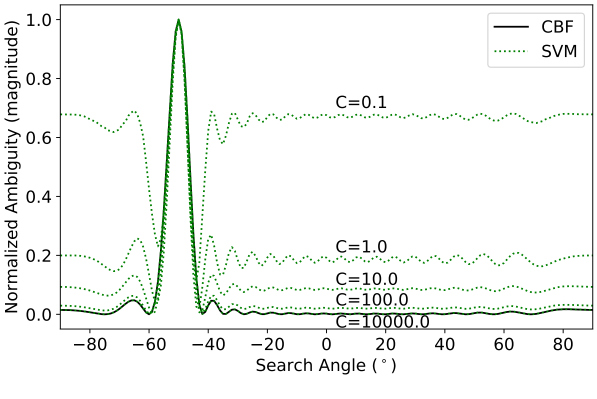

**Figure 5.3**: Ambiguity surfaces from (5.35), (5.36) for CBF and linear SVM at $\theta_t = 50°$, with the constraint value of $C$ varied for SVM.

where $\mathrm{t}_t$ is the target for input sample $\mathbf{x}_t$ from (5.9). $\mathrm{t}_t = 1$ (positive) if $\mathbf{x}_t$ is in the desired class. Figure 5.2 shows an example of the SVM classifier in two dimensions for two data classes.

The SVM is a soft margin classifier. Its solves the tradeoff between a linear classifier, $\sum_{t=1}^{T} \alpha_t^m \mathrm{t}_t \mathbf{x}_t^T \mathbf{w}^m$, and a penalty term on the number of misclassified samples, $C\sum_{t=1}^{T} \xi_t^m$, [23] where $C$ is an empirical parameter,

$$\min_{\mathbf{w}^m, \xi_t^m} \left\{ \frac{1}{2} ||\mathbf{w}^m||_2^2 + C \sum_{t=1}^{T} \xi_t^m - \sum_{t=1}^{T} \mu_t \xi_t^m - \alpha_t^m [\mathrm{t}_t \mathbf{x}_t^T \mathbf{w}^m - 1 + \xi_t^m] \right\} \tag{5.23}$$

where $\xi_t^m \geq 0$, and $\mu_t^m$ and $\alpha_t^m$ are Lagrange multipliers. $\mathbf{w}^m$ is a hyperplane that divides the data $\mathbf{x}_t$ into classes. For the $m$th class, the hyperplane $\mathbf{w}^m \in \mathbb{C}^{N^2 \times 1}$ separates $\mathbf{x}_t \in m$ from all other data. The solution for $\mathbf{w}^m$ is [23]

$$\mathbf{w}^m = \sum_{t=1}^{T} \alpha_t^m \mathrm{t}_t \mathbf{x}_t. \tag{5.24}$$

Thus $\mathbf{w}^m$ is a linear combination of the inputs $\mathbf{x}_t$, whose contribution $\alpha_t^m$ is a result of the choice of $C$, which controls the tradeoff between correct classification and margin width.

As $C \to \infty$, the penalty on $\xi_t^m$ requires that $\xi_t^m = 0 \ \forall m$ in order to satisfy (5.23). SVM becomes a linear classifier,

$$\min_{\mathbf{w}^m} \{ \frac{1}{2} ||\mathbf{w}^m||_2^2 - \sum_{t=1}^{T} \alpha_t^m [\mathrm{t}_t (\mathbf{x}_t^T \mathbf{w}^m) - 1] \} \tag{5.25}$$

Again assuming a single training example for each class such that $\mathbf{w}^m = \mathbf{w}^t$, and differentiating

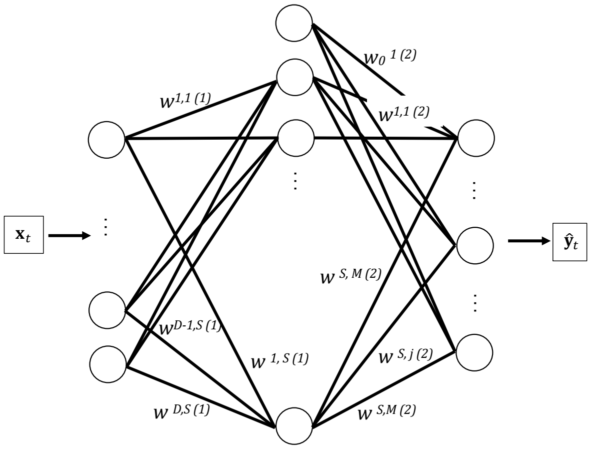

**Figure 5.4**: Model of the FNN with one hidden layer with $S$ nodes. Input nodes D = $N^2$, and output nodes $M = 180$, where $M$ is the number of arrival angles.

w.r.t $\mathbf{w}^m$,

$$\frac{\partial}{\partial \mathbf{w}^m}\Big(\frac{1}{2}||\mathbf{w}^t||_2^2 - \sum_{t=1}^{T}\alpha_t^t[\mathrm{t}_t(\mathbf{x}_t^T\mathbf{w}^t)-1]\Big) = \mathbf{w}^m - \alpha_m^m\mathbf{x}_m^T$$

$$\hat{\mathbf{w}}^m = \alpha_m^m\mathbf{x}_m \tag{5.26}$$

As with linear FNN (5.21) and CBF (5.14), the SVM weight solution for the $m$th class when trained on unique examples is proportional to the CBF covariance weights.

Figure 5.3 shows the normalized ambiguities for SVM and CBF at $\theta_m = -50°$ for $C = [0.1, 1, 10, 100, 10000]$, demonstrating the convergence of linear SVM to the linear CBF as $C \to \infty$. SVM was trained on a set of noiseless plane waves with one example at each angle, $\theta_m \in [-90°, 90°)$, $M$=180, using Scikit-learn software; [24] the SVM and CBF ambiguities are shown for the training data.

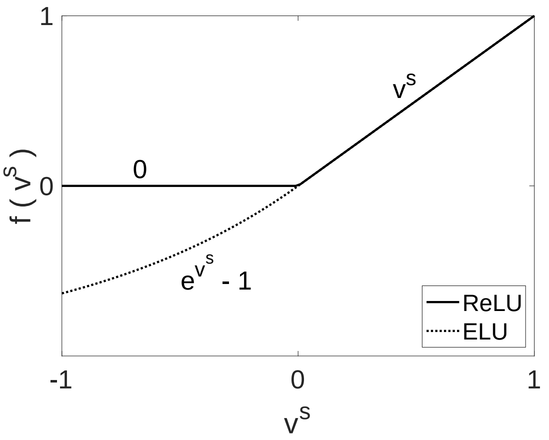

**Figure 5.5**: The recitified linear unit (ReLU) and exponential linear unit (ELU) activation functions introduce nonlinearity in the hidden layers of the FNN.

### 5.3.3 Nonlinear FNN

With hidden layers, the FNN model (Fig. 5.4) is expressed as a series of weighted functional transformations. For a two layer network, [25]

$$\hat{y}_{\mathrm{t}}^{m} = g(\mathbf{w}^{m(2)}\mathbf{a} + w_0^{m(2)}), \tag{5.27}$$

$$\mathbf{a} = [a^1, ..., a^S], \quad a^s = f(\mathbf{w}^{s(1)^T}\mathbf{x}_t + w_0^{s(1)}),$$

where $\mathbf{w}^{s(1)}$ and $w_0^{s(1)}$ are the weights and bias constants of layer 1, similarly for $\mathbf{w}^{m(2)}$ and $w_0^{m(2)}$ in layer 2. $a^s$ is the output of $s$th hidden node. $S$ is the number of nodes in layer 1, the hidden layer. $^{(1)}$ and $^{(2)}$ denote the first (hidden) or second (output) layer. Each hidden layer introduces $S(S+1)$ unknown weight and bias parameters.

A common choice for the activation function $f()$, used here, is the rectified linear unit (ReLU), see Fig. 5.5:

$$a^s = f(v^s) = \max(0, v^s), \qquad v^s = \mathbf{w}^{s^T}\mathbf{x}_t + w_0^s \tag{5.28}$$

where $s = 1, ..., S$.

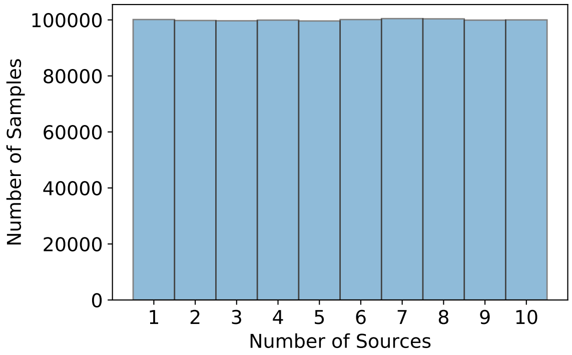

**Figure 5.6**: Distribution of the number of sources in the training set with $10^6$ total samples. For each $K \in [1,10]$, there were $10^5 \pm 450$ training samples.

A second activation function, the exponential linear unit (ELU), [26] was compared to ReLU. ELU prevents the problem of ”dying ReLU”, where the gradients reduce to zero and prevent weight updates during training, by allowing a small negative component of the activation:

$$a^s = f(v^s) = \begin{cases} \exp\{v^s\} - 1 & v^s \leq 0, \\ v^s & v^s > 0. \end{cases} \tag{5.29}$$

The softmax function [14, 22] was used for the output function, $g()$, and the cross-entropy error was used as the training cost function,

$$\hat{y}_{\mathrm{t}}^m = g(z^m) = \frac{e^{z^m}}{\sum_{j=1}^M e^{z^j}}, \quad z^m = \mathbf{w}^{m(2)}\mathbf{a} + w_0^{m(2)}, \tag{5.30}$$

$$\underset{\mathbf{w}^m}{\arg\min}\left\{-\sum_{t=1}^T \sum_{j=1}^M y_{t,\mathrm{true}}^j \ln[y_{\mathrm{pred}}^j(a^s)]\right\}. \tag{5.31}$$

with $a^s$ from (5.28), and $m = 1, ..., M$. The softmax sums all outputs to 1, with $\sum_{m=1}^M \hat{y}_t^m$. This property gives the output resemble a probability for $M$ classes.

### 5.3.4 Methods

The machine learning models were trained with simulated plane wave data according to (5.2), $|s_k| = 1$ and $f = 200$ Hz. Waves were simulated on an $N = 20$ element uniform linear array. A sound speed of $c = 1500$ m/s was assumed, with array element spacing of $\lambda/3$ (2.5 m).

For the linear FNN and 2-source FNN in Secs. 5.3.5 and 5.4, the training data included all scenarios, with $\theta_m \in [-90^\circ, 90^\circ)$, $\Delta\theta = 1^\circ$, $M = 180$, for each source. Thus, the single source case contains $T = 180$ samples and the two source case contains $T = 16110$ unique samples. For the $K$-source FNN in Sec. 5.4, the training set was generated with $T = 10^6$ random samples (Fig. 5.6), where $K \in \mathcal{U}\{1, 10\}$ sources with $\theta_k \in [-90^\circ, 90^\circ)$ and $L \in \mathcal{U}\{1, 10\}$ snapshots for each sample. $\mathcal{U}()$ is the uniform distribution.

Keras [14] software with Tensorflow backend was used to train the FNN weights. The Adam [27] optimizer was used with an initial learning rate of $10^{-3}$, then the learning rate was reduced by 0.5 per 100 epochs for smooth convergence. The FNN was trained on the training set with 1000 epochs (100 epochs for the linear example). One epoch is one cycle through the entire training set.

For SVM, Scikit-Learn [24] with the LinearSVC module with one-versus-rest formulation was used to solve (5.23), where $C = 10^4$ (i.e. linear).

A validation set with $10^4$ random samples was used in Sec. 5.4 to choose the FNN parameters: number of hidden layers, hidden nodes per layer, and activation function. For each sample, the number of sources was randomly generated was $K \in \mathcal{U}\{1, 2\}$ (2 source) or $K \in \mathcal{U}(1, 10\}$ ($K$-source) sources with $\theta_k \in \mathcal{U}[-90^\circ, 90^\circ)$ and $L = 1$ or $L = 10$ snapshots, as specified in the results.

The performance of FNN models was compared using a test set with $10^4$ random samples generated identically to the validation set. In addition, Gaussian random noise was added with variance $\sigma^2$ according to the signal-to-noise ratio (SNR)

$$\sigma^2 = \|\mathbf{p}_1\|_2^2 \times 10^{-\frac{\mathrm{SNR}}{10}}. \tag{5.32}$$

The SCM $\mathbf{P}$ was averaged for $L$ snapshots according to (5.5) before generating the FNN inputs.

The error between the estimated $\hat{\theta}_t^k$ and the true DOA $\theta_{t,k}$ is

$$\text{Error} = \frac{1}{\hat{K}T} \sum_{t=1}^{T} \sum_{k=1}^{\widehat{K}} \left| \theta_{t,k} - \hat{\theta}_t^k \right|, \tag{5.33}$$

$$\text{Accuracy} = \frac{1}{KT} \sum_{t=1}^{T} \sum_{k=1}^{K} \mathbf{1}\left[ \left| \theta_{t,k} - \hat{\theta}_t^k \right| \leq 1^\circ \right], \tag{5.34}$$

where the error is measured only for detected sources, $\hat{K} \leq K$. $\mathbf{1}[x]$ is the indicator function, which is 1 if $x =$True and False otherwise. Error is a measure of prevision for correctly identified sources. Unresolved sources lead to lower accuracy but do not contribute to the error measure in (5.33).

### 5.3.5 Array Calibration Example

Linear machine learning methods SVM and FNN are used as a perturbed array calibration technique. The linear models are quickly trained on a source with known DOA, then the learned weights act like measured replicas for the perturbed array. In an experimental scenario, linear machine learning could be rapidly applied to generate accurate replicas for an array with unknown or perturbed sensor positions.

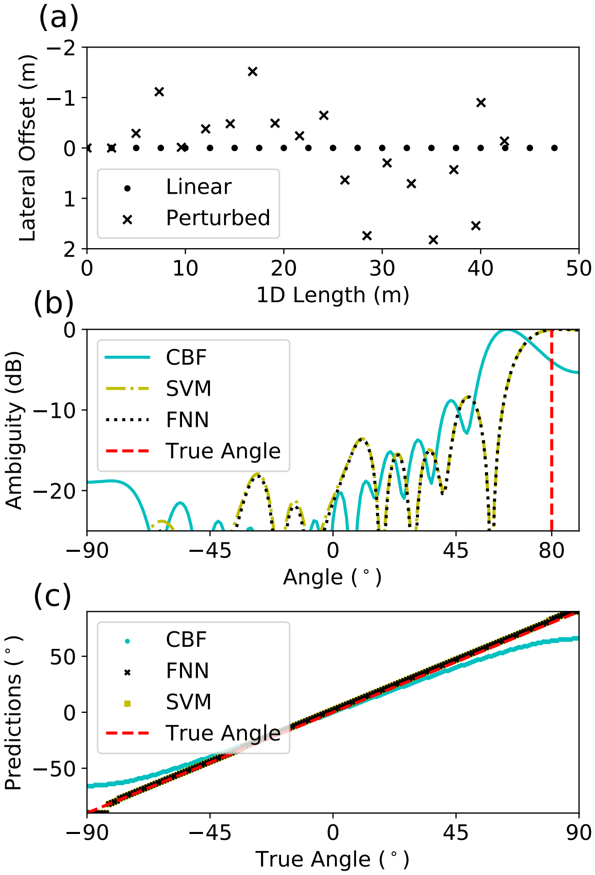

**Figure 5.7**: *(a)* Perturbed array element locations, *(b)* ambiguity surface from (5.35)–(5.37) at $\theta_m = 80°$ for CBF, linear FNN and SVM with $C = 10^4$, and *(c)* predictions at all angles. Linear FNN is identical to SVM in this example.

$$\text{CBF}: \quad B(\theta_m) = \mathbf{a}_{\text{cov}}^T(\theta_m)\mathbf{x}_t, \tag{5.35}$$

$$\text{Linear FNN}: \quad \hat{y}_t^m = \mathbf{w}^{mT}\mathbf{x}_t \tag{5.36}$$

$$\text{SVM}: \quad d^m = \mathbf{w}^{mT}\mathbf{x}_t, \quad m = 1, \dots, M. \tag{5.37}$$

The weights for CBF are given in (5.1). The weights for FNN and SVM are given according to (5.21) and (5.24). For an ideal vertical linear array, the element positions $x, y$ are $x = [0, \dots, 0]$ and $y = [0, \dots, (N-1)\ell]$, with $y$ positive downward and $x^n$ perpendicular to $y$ on a

Cartesian coordinate system. If the elements are randomly perturbed in $x$, then the new sensor positions are (Fig. 5.7a)

$$\begin{aligned} &x^n \sim \mathcal{N}(0,1^2),\ x^1 = 0 \\ &y^n = y^{n-1} + \sqrt{\ell^2 - (x^n - x^{n-1})^2},\ y^1 = 0. \end{aligned} \tag{5.38}$$

This assumes the distance between adjacent elements remains fixed at $\ell$. A plane wave of frequency $f$ is expressed by a projection onto the $n$th sensor,

$$w^n(\theta) = e^{i\frac{2\pi f}{c}(y^n \sin(\theta) - x^n \cos(\theta))}, \quad n = 1, ..., N. \tag{5.39}$$

If CBF is applied assuming the linear array, the element displacement introduces a phase offset according to (5.38) and (5.39). For an array fixed at one end and free at the other, the $y$-axis errors will be compounded along the array. Near endfire (±90°), the $y$ errors are heavily weighted and cause systematic bias in the ambiguity surface as seen in Fig. 5.7b and c. The supervised machine learning method removes ambiguity bias by calibration the weights to the true element positions.

## 5.4 Deep learning for DOA estimation

Nonlinear machine learning methods are necessary for inferring nonlinear solutions. For example, the binary XOR function does not have a linear solution but can be mapped by a simple neural network into a new feature space where it can be replicated using a linear model. [28]

The nonlinear FNN was used to estimate DOA. For the two source problem, the FNN was trained on all combinations, with $M = 180$ and C($M$,2) = 16,110 training samples. This problem can also be solved using an exhaustive search. Then, the method was extended to $K$ sources ($K \leq 10$ in these examples). In this case, exhuastive search is impractical. Instead, uniform

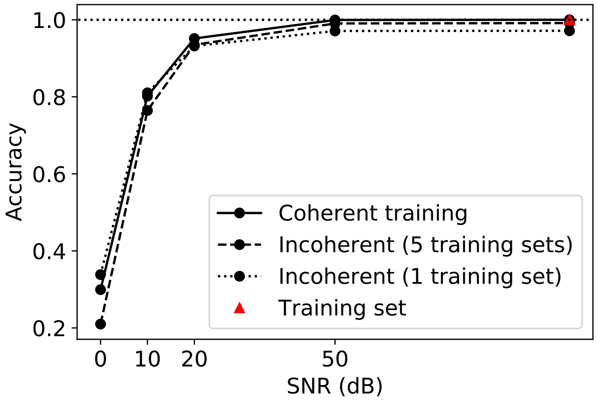

**Figure 5.8**: Validation accuracy vs SNR for FNN trained on coherent (solid) and incoherent (dashed) sources. The validation set was generated from 1000 random simulations of 2 sources with $L = 1$ snapshot. The FNN input features depend on the relative source phases.

random sampling was employed to select a training set with $10^6$ samples, with $K$ sources and $L$ snapshots for each sample.

The predictions are compared to CBF (5.35), MUSIC (5.41), and SBL, with the ambiguity surfaces given by

$$\text{FNN}: \; y_t^m = \frac{e^{z^m}}{\sum_{j=1}^M e^{z^j}}, \quad m = 1, ..., M. \tag{5.40}$$

$$\text{MUSIC}: \; B(\theta_m) = (\mathbf{a}^H(\theta_m)\mathbf{P}_n\mathbf{P}_n^H\mathbf{a}(\theta_m))^{-1} \tag{5.41}$$

$$\text{SBL}: \; \gamma_m, \;\; \mathbf{\Gamma} = \text{diag}(\gamma_1, ..., \gamma_m, ..., \gamma_M). \tag{5.42}$$

where $\mathbf{P}_n \in \mathbb{C}^{N\times(N-K)}$ is the matrix of the noise eigenvectors of the SCM, $\mathbf{P}$ in (5.5). SBL is summarized in Appendix A.

### 5.4.1 Source coherence

Consider two plane wave sources with random phases at each snapshot $l$, $\phi_k^l \in \mathcal{U}(-\pi,\pi)$, then from (5.2)

$$\mathbf{p}_l = \sum_{k=1}^{2} s_k \mathbf{a}(\theta_k) = \sum_{k=1}^{2} \frac{|s_k|}{\sqrt{N}} e^{i\phi_k^l} e^{i\frac{2\pi f}{c}(n-1)\ell\sin(\theta_k)} \tag{5.43}$$

for $n = 1,...,N$. The sources will be coherent if, over $L$, samples

$$\mathbb{E}_L(s_1 s_2 e^{i(\phi_1^l - \phi_2^l)}) = \rho\mathbb{E}_L(s_1\phi_1^l)\mathbb{E}_L(s_2\phi_2^l) \tag{5.44}$$

when $|\rho| = 1$. If $\rho < 1$, the sources will be correlated; $\rho = 0$ corresponds to incoherent sources (see (6.586-6.588) in *Optimal Signal Processing* [29]). From (5.5), the terms of the SCM are

$$P_{n_1,n_2} = P^R_{n_1,n_2} + iP^I_{n_1,n_2} = (p_{n1}p^H_{n2})^R + i(p_{n1}p^H_{n2})^I \tag{5.45}$$

where the indices $n_1, n_2 = 1,...,N$. Combining (5.43) and (5.45) and applying the identity $e^{i\Theta} = \cos(\Theta) + i\sin(\Theta)$, each element in the SCM is

$$\begin{aligned} P^{\mathrm{R}}_{n_1,n_2} = \frac{1}{N}\Big[\sum_{k=1}^{2} |s_k|^2 \cos\Big(\frac{2\pi f}{c}(n_1 - n_2)\ell\sin(\theta_k)\Big) + \\ 2 s_1 s_2 \cos\Big(\frac{2\pi f}{c}(n_1\sin(\theta_1) - n_2\sin(\theta_2))\ell + \Delta\phi\Big)\Big] \end{aligned} \tag{5.46}$$

$$P^{\mathrm{I}}_{n_1,n_2} = \frac{1}{N}\Big[\sum_{k=1}^{2} |s_k|^2 \sin\Big(\frac{2\pi f}{c}(n_1 - n_2)\ell\sin(\theta_k)\Big)\Big] \tag{5.47}$$

where $\Delta\phi^l = \phi^l_1 - \phi^l_2$. If $\Delta\phi^l \in [-\pi,\pi]$, the expected value of $\mathbf{P}$ across $L \to \infty$ snapshots is

$$
\begin{aligned}
<P^R_{n_1,n_2}> = \frac{1}{N}\Big[\sum_{k=1}^{2} |s_k|^2 \cos\Big(\frac{2\pi f}{c}(n_1 - n_2)\ell \sin(\theta_k)\Big) + \\
2 s_2 s_2 \int_0^{2\pi}\int_0^{2\pi} \cos\Big(\frac{2\pi f}{c}(n_1 \sin(\theta_1) - n_2 \sin(\theta_2))\ell + \Delta\phi^l\Big) \mathrm{d}\phi_1 \mathrm{d}\phi_2\Big] \\
= \frac{1}{N}\sum_{k=1}^{2} |s_k|^2 \cos\Big(\frac{2\pi f}{c}(n_1 - n_2)\ell \sin(\theta_k)\Big) \qquad (5.48)
\end{aligned}
$$

since the cosine is integrated over a full period. The limit of large snapshot averaging corresponds to $\rho \to 0$ and estimates the SCM for two incoherent sources.

The 2-source FNN was trained on noiseless coherent and incoherent sets (Fig. 5.8). With few snapshots, the cross terms from source incoherence lead to DOA errors, despite noiseless data. The effect can be reduced by training the FNN on multiple (5) incoherent instances. Figure 5.8 shows that the coherently trained FNN (tested on coherent) achieves an accuracy of 1 for noiseless validation data because the input features for the training and validation set are identical. By increasing the number of incoherent training instances, the incoherent model improves from 0.94 to 0.99 accuracy on validation data, demonstrating its increased robustness to random source phases.

### 5.4.2 Hidden layers and nodes

The number of hidden layers, hidden nodes, and activation function for the 2-source and $K$-source FNN was selected using the accuracy on the validation sets described in Sec. 5.3.4. Each additional hidden layer introduces $(S+1)S$ model parameters (weights and biases) for $S$ hidden nodes, improving the representation capability but increasing the model complexity.

The results in Fig. 5.9(a)–(c) suggest that a minimum of 3 hidden layers improves the accuracy on validation data compared to shallower FNNs. Likewise, at least 512 hidden nodes were required for highest accuracy (Fig. 5.9(d), shown for 2-source FNN), while more than 512

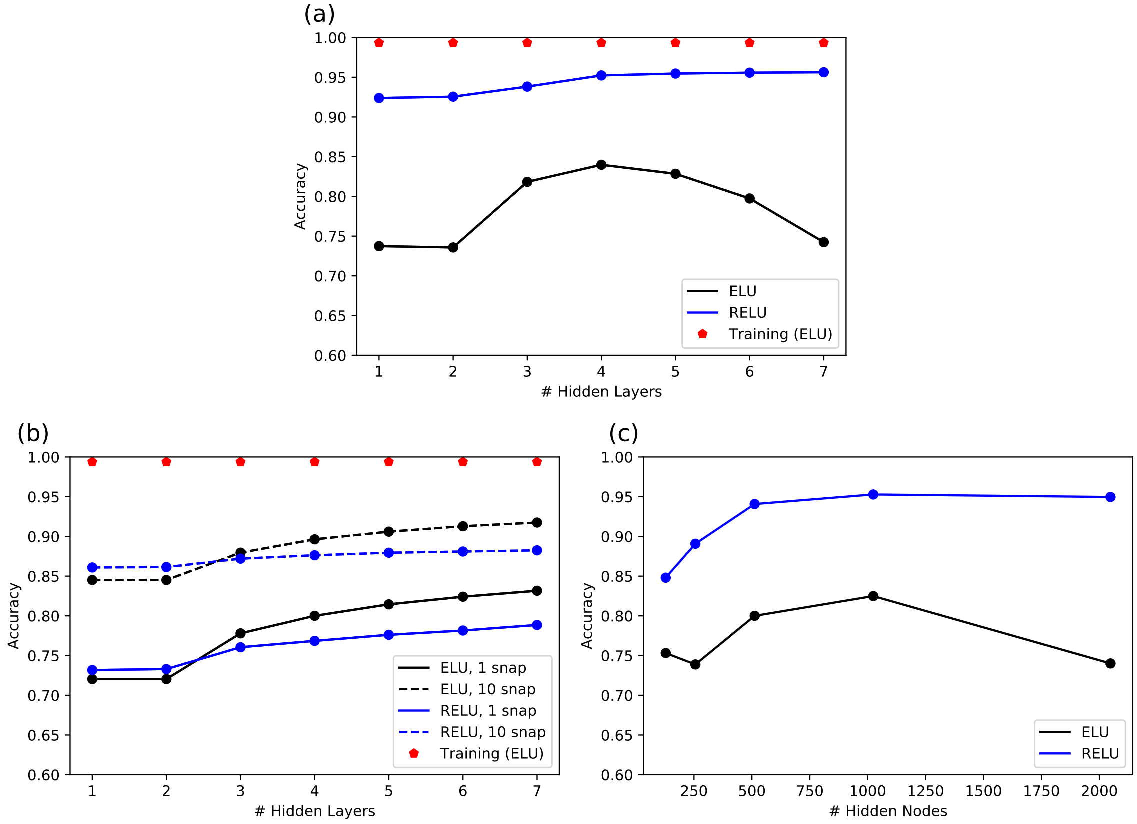

**Figure 5.9**: Validation accuracy vs *(a,b)* number of hidden layers and *(c)* number of hidden nodes for *(a,c)* 2-source FNN and *(b)* $K$-source FNN. All cases used $L = 1$ snapshots; $L = 10$ is also shown in *(b)*.

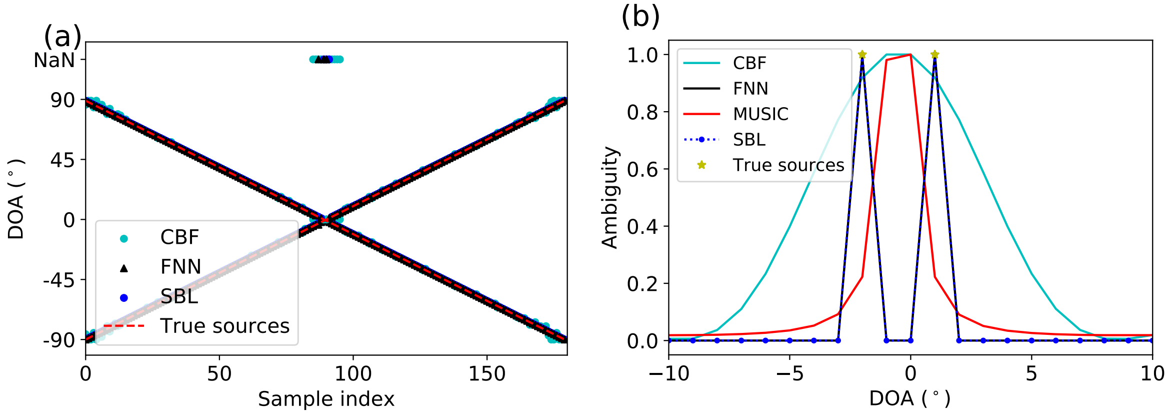

**Figure 5.10**: Test set DOA single-snapshot estimation of two incoherent sources traveling along an intersecting path. *(a)* The maximum peaks for all samples are found by taking the top two peaks of the ambiguity surfaces. Missing detections (NaNs) are shown at the top of the grid. *(b)* Sample ambiguity surface near the source crossing (see (5.35), (5.40)–(5.42)) at $\theta_1 = 1^\circ, \theta_2 = -2^\circ$.

hidden nodes did not improve the results.

The effect of the hidden layer activation function depended on the FNN model. For 2-source FNN, ELU greatly reduced the accuracy of the FNN, while ReLU was consistent and accurate across number of hidden layers and nodes. For $K$-source FNN, ELU and ReLU were more consistent, with ELU leading to a small increase in accuracy.

### 5.4.3 DOA estimation of incoherent sources

Figure 5.10 demonstrates the performance of the incoherently trained 2-source FNN as a high–resolution beamformer on a single-snapshot two source transit scenario. SBL and CBF results are shown for comparison. In Fig. 5.10a, both 2-source FNN and SBL fail close to the source crossing, while CBF is unable to separate sources within 6°. These failures are shown by NaNs.

Figure 5.11 shows a random sample from the $K$-source test data set with $K = 3$ sources. High resolution methods, including deep FNN and SBL, are used to localize the two close sources. MUSIC performance is unreliable for estimating DOA of incoherence sources with a single

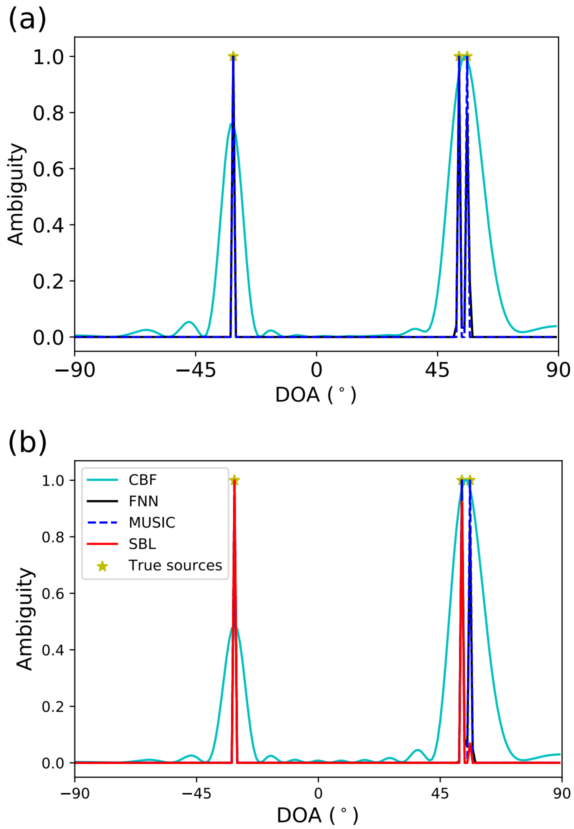

**Figure 5.11**: Ambiguity surfaces for CBF (5.35), MUSIC, SBL, and deep FNN (5.40)-(5.42) for a random sample with $K = 3$ sources, using *(a)* $L = 1$ snapshots and *(b)* $L = 10$ snapshots. The true sources are at $-31°$, $53°$, and $56°$.

snapshot.

A comparison of beamforming methods for the 2-source incoherent case is shown in Table 5.2. Both SBL and 2-source FNN are feasible single snapshot DOA estimators. At 10 snapshots, MUSIC is also highly accurate without noise. All high resolutions methods surpass CBF in their ability to resolve close sources. All methods showed improved performance at 10 snapshots.

**Table 5.2**: Mean error (5.33) and accuracy (5.34) for test data with $K = 2$ sources, $\theta_k \in [-90^\circ, 90^\circ)$.

| **1 snapshot** | | | **10 snapshots** | | |
|---|---|---|---|---|---|
| Method | Error (°) | Acc. (%) | Method | Error (°) | Acc. (%) |
| CBF | 1.09 | 77.5 | CBF | 0.52 | 86.7 |
| | | | MUSIC | 0.00 | 99.5 |
| SBL | 0.06 | 98.0 | SBL | 0.06 | 98.5 |
| FNN (7) | 0.16 | 95.9 | FNN (7) | 0.14 | 96.6 |

**Table 5.3**: Mean error (5.33) and accuracy (5.34) for test data with $K \in \mathcal{U}(1, 10)$ random sources, $\theta_k \in [-90^\circ, 90^\circ)$.

| **1 snapshot** | | | **10 snapshots** | | |
|---|---|---|---|---|---|
| Method | Error (°) | Acc. (%) | Method | Error (°) | Acc. (%) |
| CBF | 2.07 | 50.5 | CBF | 0.97 | 61.6 |
| | | | MUSIC | 0.00 | 97.1 |
| SBL | 0.64 | 82.1 | SBL | 0.10 | 93.9 |
| FNN (7) | 0.92 | 83.1 | FNN (7) | 0.13 | 91.6 |

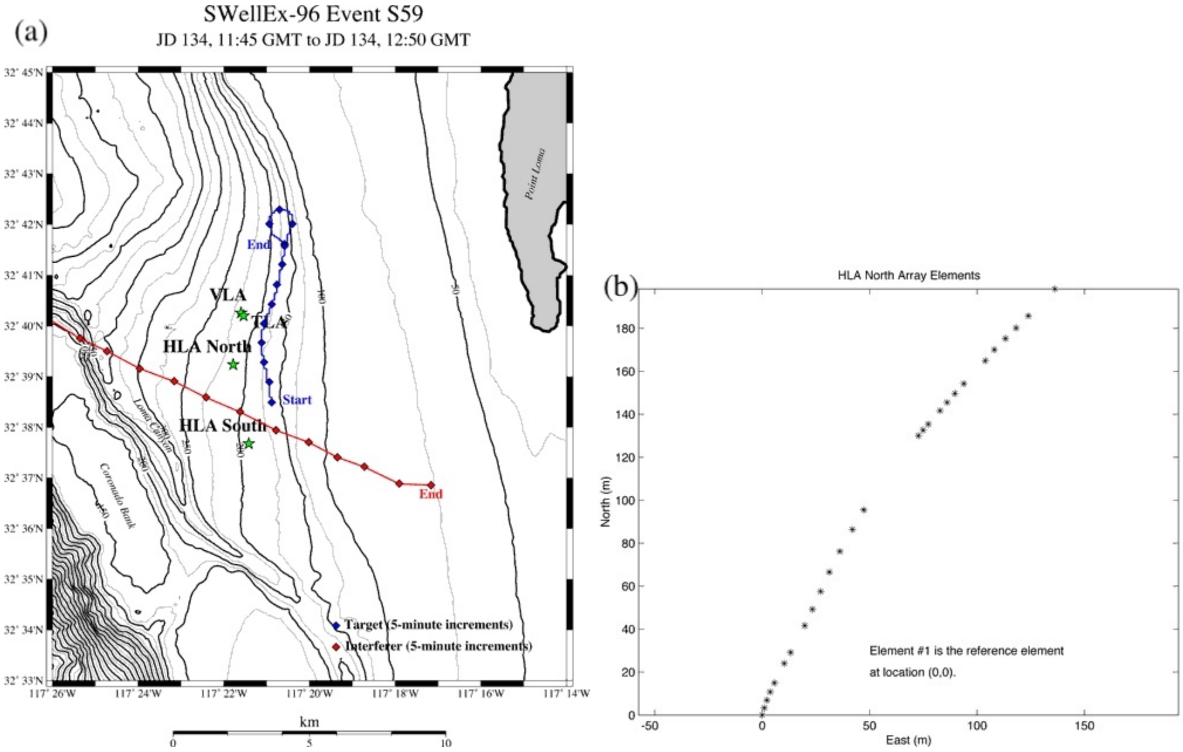

**Figure 5.12**: *(a)* Map of the Swellex-96 experimental setup for May 13, 1996. Shown are transits of the deep towed source and the loud interferer along with four hydrophone arrays. *(b)* Element orientation for HLA North, used in this study.

## 5.5 Swellex96

As an experimental test, the azimuth (North = 0°, East = 90°) of the Swellex-96 S95 deep source tow event and a loud interferer to the North horizontal line array (North HLA) was estimated. [30–32] The Swellex-96 experiment was conducted in a shallow water waveguide with downward refracting sound speed and two sediment layers (Fig. 5.12). [33] Azimuth estimation has been shown to be less sensitive to the sloping bottom for this experiment. [34] A range-independent environment with plane wave propagation is here assumed for the source tow and interferer tracks.

Training data for the FNN was generated with plane waves according to (5.2) using the measured positions of the HLA North array elements. The models were trained on data described in Sec. 5.3.4.

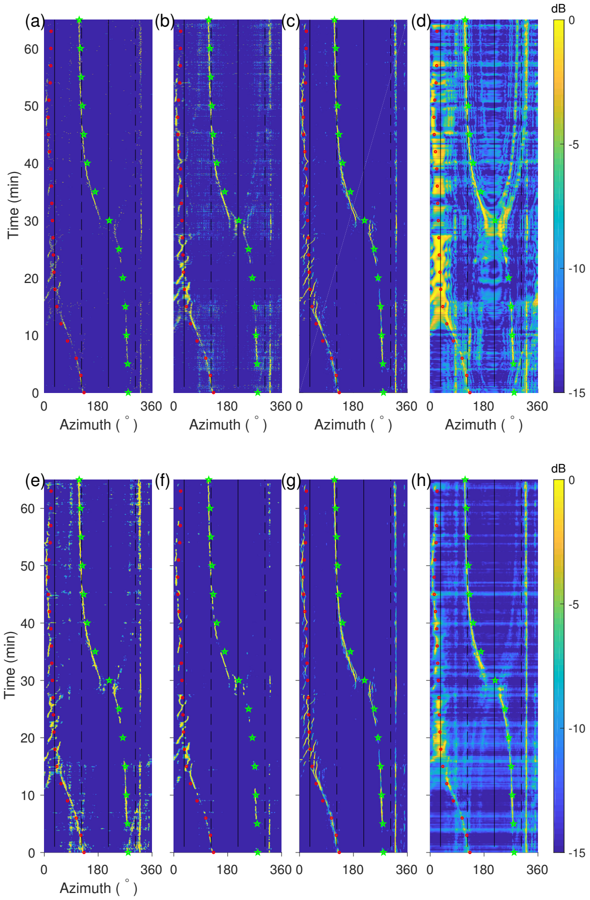

**Figure 5.13**: *(a-d)* Single-snapshot and *(e-h)* 10 snapshot. *(a,e)* $K$-source FNN, *(b,f)* 2-source FNN, *(c,g)* SBL, *(d)* CBF, and *(h)* MUSIC ambiguities, normalized to max at each time, for the Swellex-96 S95 source tow event (circles) and a loud interferer (stars). Array endfire (solid) and broadside (dashed) directions are shown.

The experimental data (test data) was recorded from 11:45–12:50 GMT, May 13, 1996 on HLA North with a sampling rate of 3267.8 Hz. The data was processed using an FFT length of 4096 samples, or 1.25 s (bin width 0.8 Hz). The covariance matrix was constructed at 79 Hz according to (5.5) and contains a tonal from the towed source and strong ship noise, with $L = 1$ (Fig. 5.13*(a-c)* and $L = 10$ (Fig. 5.13)*(d-f)* snapshots. There were 3120 total test samples (1.25 s chunks over 65 min.).

Figure 5.13 shows the ambiguity surfaces with $L = 1$ and $L = 10$ snapshots across azimuth and time for FNN, SBL, and CBF. All ambiguities are normalized to their maximum for each time. The ambiguities for SBL and FNN were convolved with a 3x3 unit filter to widen the peak for improved visualization.

The results demonstrate that FNN can estimate DOA for an unknown number of sources in single-snapshot experimental data. At high SNR, the performance of the FNN is similar to SBL in simulated and experimental results. [35] In experiment, CBF exhibits higher ambiguity sidelobes, while MUSIC exhibits lower SNR without additional regularization. The interferer at 340° is a secondary source, FLIP.

A benefit of neural network estimation is that the heavy computation is conducted offline. Using a TITAN XP graphics processing unit (GPU), the 2-source FNN training for Sec. 5.5 takes 20 minutes and the $K$-source FNN training takes 6 hours. DOA estimation CPU time took 0.06 s for CBF, 20 s for SBL, and 0.5 s for both FNNs on a Macbook Pro with Intel Core i7 processor, for a single sample of Sec. 5.5.

## 5.6 Convolutional Neural Network

Convolutional networks [28] (CNNs) are deep neural networks that leverage spatial relationships in 2D data. Here, the SCM was used as an input image, $\mathbf{X}_t = [\text{Re}\{\mathbf{P}_t\}, \text{Im}\{\mathbf{P}_t\}] \in \mathbb{R}^{N \times N \times 2}$, with dimensions $N = 20$ and two channels representing the real and imaginary components. 2D filters $\mathbf{K}^j \in \mathbb{R}^{N_c \times N_c}$, $j = 1, ..., S$, were convolved with the input image, where $N_c = 3$ (typically, $N_c < 10$), [28] and translated with a stride of 1 across the image. The output dimension of a single hidden layer was thus $(N - N_c + 1) \times (N - N_c + 1) \times 2$, a decrease by 2 in each dimension when $N_c = 3$.

Three CNN models were tested, see Fig. 5.14. The first model in Fig. 5.14a was a 9-layer convolutional model with output of size $(2 \times 2)$. In the second model, zero-padding was used to increase the output size at each layer to $20 \times 20$ (not shown). The third model in Fig. 5.14b included zero-padding at each layer and three MaxPool layers, which downsampled the image by

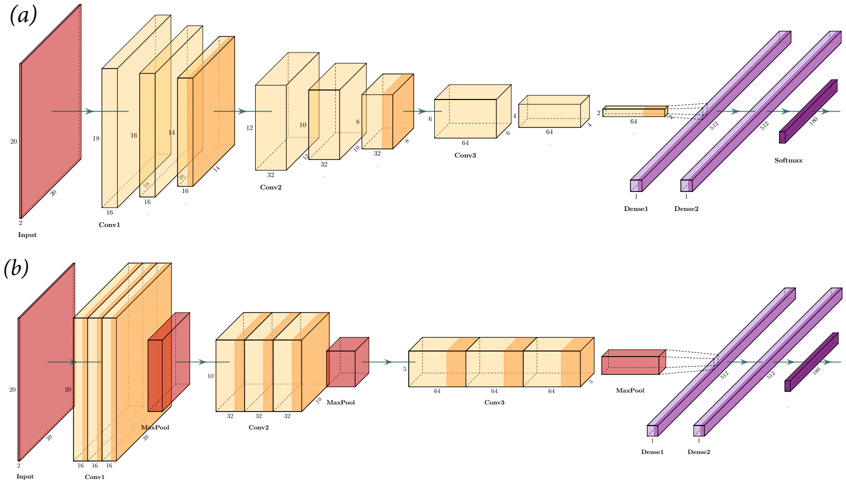

**Figure 5.14**: CNN models for DOA estimation with 9 convolutional layers and 2 dense layers. *(a)* 9 convolutional layers and *(b)* with zero–padding and MaxPool layers. A model with zero–padding only was also tested.

taking the max value within $2 \times 2$ patches, resulting in an output layer of size $(2 \times 2)$.

All CNN models included two fully connected layers followed by an output softmax layer. The cross-entropy error [36] in was used for categorical classification training, see Appendix 5.8.2 for details.

The results of the three CNN models are shown in Table 5.4. The addition of zero-padding increases the error. The addition of MaxPool layers may marginally improve the error caused by zero-padding.

## 5.7 Conclusion

In this paper, CBF was formulated as linear in the weight covariance matrix and the data SCM instead of quadratic in the beam weights. With a set of measured data and labels, the linear formulation was used to directly learn the CBF weights in a perturbed array scenario.

The linearized SCM was used as an input to a deep FNN model for estimating DOA. First, the deep FNN was trained exhaustively for the noiseless two source scenario. Then, the deep

**Table 5.4**: Mean absolute error and accuracy for FNN and three CNN models.

| Method | Error<br>(°) | Accuracy<br>(%) |
|---|---|---|
| **1 snapshot** | | |
| FNN (5) | 1.10 | 81.0 |
| CNN | 1.05 | 74.1 |
| CNN, Zero-Padding | 3.21 | 59.8 |
| CNN, Zero-Padding, MaxPool | 1.79 | 68.3 |
| **10 snapshots** | | |
| FNN (5) | 0.13 | 91.3 |
| CNN | 0.29 | 82.8 |
| CNN9, Zero-Padding | 0.68 | 82.3 |
| CNN9, Zero-Padding, MaxPool | 0.35 | 83.0 |

FNN was trained using a dataset with $K$ randomly generated number of sources DOAs, with $K \in (1,10)$ and $\theta_k \in [-90°, 90°)$. On a separate test set, both 2-source and $K$-source deep FNN as well as SBL estimated DOA accurately using a single snapshot.

The FNN models were selected by measuring DOA estimation accuracy on a noiseless validation set. The number of hidden layers in each model were varied for ReLU and ELU hidden activation functions. Generally, deeper models with increased hidden nodes per layer had improved accuracy.

The real world applicability of the 2-source and $K$-source FNN were demonstrated on the S95 source tow and loud interferer in the Swellex96 experiment. For single-snapshot estimation, FNN performs comparable to SBL, and both methods have increased resolution over CBF. Similar results are seen for 10 snapshots. The prediction step of FNN is faster than SBL while giving comparable performance.

Finally, the potential of the convolutional neural network was demonstrated for future research. A major challenge of the convolutional neural network is the small dimensionality of the input SCM, which may be addressed with more sophisticated network structures.

# 5.8 Appendix

## 5.8.1 Sparse Bayesian Learning

Sparse Bayesian Learning (SBL) is a compressive sensing technique that uses a Bayesian framework to solve sparse parameter estimation problems. The algorithmic implementation is detailed in Nannuru et al. [20]

First, assume that the observation in (5.2) can be represented by a sparse dictionary of weights and the plane wave replicas,

$$\mathbf{Y} = [\mathbf{p}_1, ..., \mathbf{p}_l, ...\mathbf{p}_L] = \mathbf{A}\mathrm{X} + \mathbf{N}, \tag{5.49}$$

where

$$\begin{aligned} \mathbf{A} &= [\mathbf{a}(\theta_1), ..., \mathbf{a}(\theta_M)], \\ \mathrm{X} &= [\mathrm{x}_1, ..., \mathrm{x}_L], \\ \mathbf{N} &= [\mathbf{n}_1, ..., \mathbf{n}_L], \quad \mathbf{n}_{\mathbf{l}} \in \mathcal{CN}^{N \times 1} \end{aligned}$$

with $\mathbf{a}(\theta_m)$ from (5.1). $\mathrm{x}_l \in \mathbb{C}^N$ is a sparse vector with $K \ll M$ nonzero entries in (5.2). The nonzero indices, or *active sources*, must remain active across all snapshots.

Assume $p(\mathbf{p}_l|\mathbf{x}_t)$ are i.i.d for $l = 1, ..., L$, with the likelihood function [20, 37]

$$p(\mathbf{Y}|\mathrm{X}) = \prod_{l=1}^{L} p(\mathbf{p}_l|\mathrm{x}_l) = \prod_{l=1}^{L} \mathcal{CN}(\mathbf{p}_l; \mathbf{A}\mathrm{x}_l, \sigma^2\mathbf{I}). \tag{5.50}$$

The prior over $\mathtt{X}$ is zero-mean complex Gaussian and the evidence is given by Bayes' rule:

$$p(\mathtt{X}) = \prod_{l=1}^{L} p(\mathtt{x}_l) = \prod_{l=1}^{L} \mathcal{CN}(\mathtt{x}_l; \mathbf{0}, \mathbf{\Gamma}), \tag{5.51}$$

$$p(\mathbf{Y}) = \int p(\mathtt{X}) p(\mathbf{Y}|\mathtt{X}) d\mathtt{X} = \prod_{l=1}^{L} \mathcal{CN}(\mathbf{p}_l; \mathbf{0}, \mathbf{\Sigma}) \tag{5.52}$$

where $\mathbf{\Gamma} = \mathrm{diag}(\gamma_1, ..., \gamma_M) = \mathrm{diag}(\gamma)$ is a diagonal covariance and the model covariance is given by $\mathbf{\Sigma} = \sigma^2 \mathbf{I} + \mathbf{A}\mathbf{\Gamma}\mathbf{A}^H$.

SBL solves for $\Gamma$ by maximizing (5.52)

$$\begin{aligned}(\hat{\gamma}_1, ..., \hat{\gamma}_M) &= \arg\max_{\gamma} \ \log p(\mathbf{Y}) \\ &= \arg\max_{\gamma} \left\{ -\frac{1}{L} \sum_{l=1}^{L} (\mathbf{p}_l^H \mathbf{\Sigma}^{-1} \mathbf{p}_l + \log |\mathbf{\Sigma}|) \right\}\end{aligned} \tag{5.53}$$

A fixed-point update rule [38] is derived by differentiating (5.53)

$$\gamma_m^{\mathrm{new}} = \gamma_m^{\mathrm{old}} \frac{1}{L} \frac{||\mathbf{Y}^H \mathbf{\Sigma}_{\mathbf{p}}^{-1} \mathbf{a}(\theta_m)||^2}{\mathbf{a}(\theta_m)^H \mathbf{\Sigma}_{\mathbf{p}}^{-1} \mathbf{a}(\theta_m)} \tag{5.54}$$

Thus SBL estimates the signal power, $\gamma_m$, for each DOA. The source DOAs and number of sources correspond to the estimates $\gamma_m \neq 0$.

### 5.8.2 FNN Training Loss

The neural network output depends on its application and the problem set-up. The model may be trained with different combinations of output and cost function, which measures the estimation error at the output compared to the training labels. For multiclass classification, used in this paper, the most common output function is the *softmax function* [36]

$$\hat{y}_{\mathrm{t}}^{m} = \frac{e^{z^{m}}}{\sum_{j=1}^{M} e^{z^{j}}}, \quad z^{m} = \sum_{s=1}^{S} w^{s,m(2)} a^{s} + w_{0}^{m(2)}, \tag{5.55}$$

where $a^s$ is the activation at the $s$th hidden node, $w^{s,m(2)}$ and $w_0^{m(2)}$ are the weights and bias, and $\hat{y}_t^m$ is the output prediction for a single class, normalized between 0 and 1. The softmax is a likelihood-like function. As in probability estimates, all classes sum to 1.

The corresponding cost function is the *categorical cross-entropy*, which measures the similarity between the predicted and true class across $T$ samples and $M$ classes,

$$J(\hat{y}_t^m) = -\sum_{t=1}^{T} \sum_{m=1}^{M} y_t^m \ln[\hat{y}_t^m (a^s)]. \tag{5.56}$$

For a regression problem, the target value is predicted directly instead of its likelihood. The regression output function is linear output,

$$\hat{y}_t = \sum_{s} w^{s(2)} a^s + w_0^{(2)}. \tag{5.57}$$

The *mean squared error* [39] can be used to measure the closeness of the estimated to the target value,

$$J(\hat{y}_t) = \sum_{t=1}^{T} |y_t - \hat{y}_t|^2. \tag{5.58}$$

The *cosine proximity* is an alternative regression cost function that measures the similarity between the target and estimate, [14]

$$J(\hat{y}^t) = -\sum_{t=1}^{T} y_t \cdot \hat{y}_t. \tag{5.59}$$

## 5.9 Acknowledgments

Thank you to the Office of Naval Research for funding this work under award number N00014-18-1-2065. Haiqiang Niu thanks the Youth Innovation Promotion Association, Chinese Academy of Sciences.

The text of Chapter Five is in full a reprint of the material as it appears in Emma Ozanich, Peter Gerstoft, and Haiqiang Niu, "Feedforward neural network for direction-of-arrival estimation," *Journal of the Acoustical Society of America*, 147(3):2035–2048, 2020. The dissertation author was the primary researcher and author in Chapter Four. The coauthors listed in this publication directed and supervised the research.

## Bibliography

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**, 436–444 (2015).

[2] M. J. Bianco, P. Gerstoft, J. Traer, E. Ozanich, M. A. Roch, S. Gannot, and C.-A. Deledalle, "Machine learning in acoustics: Theory and applications," J.Acoust. Soc. Am. **146**, 3590–3628 (2019).

[3] H. Niu, E. Reeves, and P. Gerstoft, "Source localization in an ocean waveguide using supervised machine learning," Journal of the Acoustical Society of America **142(3)**, 1176–1188 (2017).

[4] H. Niu, E. Ozanich, and P. Gerstoft, "Ship localization in Santa Barbara Channel using machine learning classifiers," J. Acoust. Soc. Am. **142**, EL455–EL460 (2017).

[5] Y. Wang and H. Peng, "Underwater acoustic source localization using generalized regression neural network," J. Acoust. Soc. Am. **143**, 2321–2331 (2018).

[6] Z. Huang, J. Xu, Z. Gong, H. Wang, and Y. Yan, "Source localization using deep neural networks in a shallow water environment," J. Acoust. Soc. Am. **143**, 2922–2932 (2018).

[7] R. Lefort, G. Real, and A. Drémeau, "Direct regressions for underwater acoustic source localization in fluctuating oceans," App. Acoustics **116**, 303–310 (2017).

[8] H. Niu, Z. Gong, E. Ozanich, P. Gerstoft, H. Wang, and Z. Li, "Deep-learning source localization using multi-frequency magnitude-only data," J.Acoust. Soc. Am. **146**, 211–222 (2019).

[9] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, “A learning-based approach to direction of arrival estimation in noisy and reverberant environments,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015), pp. 2814–2818.

[10] S. Chakrabarty and E. A. P. Habets, “Multi-speaker DOA estimation using deep convolutional networks trained with noise signals,” IEEE Journal of Selected Topics in Signal Processing **13**(1), 8–21 (2019).

[11] G. Izacard, B. Bernstein, and C. Fernandez-Granda, “A learning-based framework for line-spectra super-resolution,” CoRR **abs/1811.05844** (2018) `http://arxiv.org/abs/1811.05844`.

[12] G. Izacard, B. Bernstein, and C. Fernandez-Granda, “A learning-based framework for line-spectra super-resolution,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), pp. 3632–3636.

[13] W. Wang, H. Ni, L. Su, T. Hu, Q. Ren, P. Gerstoft, and L. Ma, “Deep transfer learning for source ranging: Deep-sea experiment results,” J. Acoust. Soc. Am. **146**(4), EL317–EL322 (2019).

[14] F. Chollet *et al.*, “Keras,” https://keras.io (2015).

[15] N. C. Raj, P. V. Aswathy, and K. V. Sagar, “Determination of angle of arrival using nonlinear support vector machine regressors,” in *2007 International Conference on Signal Processing, Communications and Networking* (2007), pp. 512–515, doi: `10.1109/ICSCN.2007.350652`.

[16] D. Salvati, C. Drioli, and G. L. Foresti, “A weighted MVDR beamformer based on SVM learning for sound source localization,” Pattern Recognition Letters **84**, 15–21 (2016).

[17] G. Lin, Y. Li, and B. Jin, “Research on support vector machines framework for uniform arrays beamforming,” in *2010 International Conference on Intelligent Computation Technology and Automation* (2010), Vol. 3, pp. 124–127, doi: `10.1109/ICICTA.2010.215`.

[18] M. M. Ramon, N. Xu, and C. G. Christodoulou, “Beamforming using support vector machines,” IEEE Antennas Wirel. Propag. Lett. **4**, 439–442 (2005) doi: `10.1109/LAWP.2005.860196`.

[19] M. Martinez-Ramon, J. L. Rojo-Alvarez, G. Camps-Valls, and C. G. Christodoulou, “Kernel antenna array processing,” IEEE Trans. Antennas Propag. **55**(3), 642–650 (2007) doi: `10.1109/TAP.2007.891550`.

[20] S. Nannuru, A. Koochakzadeh, K. L. Gemba, P. Pal, and P. Gerstoft, “Sparse Bayesian learning for beamforming using sparse linear arrays,” J. Acoust. Soc. Am. **144**(5), 2719–2729 (2018).

[21] D. DeFatta, J. Lucas, and W. S. Hodgkiss, *Digital Signal Processing: A System Design Approach*, Vol. 1 (John Wiley & Sons, New York, 1988), pp. 628–649, Appendix 11.A.

[22] T. Hastie, R. Tibshriani, and J. Friedman, *The Elements of Statistical Learning*, $2^{nd}$ ed. (Springer Series in Statistics Springer New York Inc., New York, 2009), Chap. 11, p. 395.

[23] T. Hastie, R. Tibshriani, and J. Friedman, *The Elements of Statistical Learning*, $2^{nd}$edition ed. (Springer Series in Statistics Springer New York Inc., New York, 2009), Chap. 12, p. 418–420.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," J. Mach. Learn. Res **12**, 2825–2830 (2011).

[25] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006), Chap. 5.1, pp. 227–229.

[26] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *ICLR (2016)* (2016).

[27] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Proc. of the 3rd ICLR (2014).

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning* (MIT press, 2016), pp. 168–170.

[29] H. L. VanTrees, *Optimal Array Processing* (Wiley, New York, 2002), pp. 599–620, Ch. 6.12: Beamforming for Correlated Signal and Interference.

[30] P. Hursky, W. S. Hodgkiss, and W. A. Kuperman, "Matched field processing with data-derived modes," J. Acoust. Soc. Am. **109**(4), 1355–1366 (2001).

[31] K. L. Gemba, W. S. Hodgkiss, and P. Gerstoft, "Adaptive and compressive matched field processing," J. Acoust. Soc. Am. **141**(1), 92–103 (2017).

[32] C. Yardim, Z. Michalopoulou, and P. Gerstoft, "An overview of sequential Bayesian filtering in ocean acoustics," IEEE J. Ocean Eng. **36**(1), 71–89 (2011).

[33] N. O. Booth, A. T. Abawi, P. W. Schey, and W. S. Hodgkiss, "Detectability of low–level broadband signals using adaptive matched-field processing with vertical aperture arrays," IEEE J. Oceanic Eng. **25**, 296–313 (2000).

[34] G. L. D'Spain, J. J. Murray, W. S. Hodgkiss, N. O. Booth, and P. W. Schey, "Mirages in shallow water matched field processing," J. Acoust. Soc. Am. **105**(6), 3245–3265 (1999).

[35] K. L. Gemba, S. Nannuru, and P. Gerstoft, "Robust Ocean Acoustic Localization With Sparse Bayesian Learning," IEEE J. Sel. Top. Sig. Proc. **13**(1), 49–60 (2019).

[36] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006), Chap. 5.2, pp. 235–236.

[37] P. Gerstoft, C. F. Mecklenbräuker, A. Xenaki, and S. Nannuru, “Multisnapshot sparse Bayesian learning for doa,” IEEE Signal Proc. Let. **23**(10), 1469–1473 (2016).

[38] D. Wipf and B. D. Rao, “An empirical Bayesian strategy for solving the simultaneous sparse approximation problem,” IEEE Trans. Signal Proc. **55**(7), 3704–3716 (2007).

[39] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, New York, 2006), Chap. 5.2, p. 233.

# Chapter 6

# Conclusion

In this dissertation, supervised and unsupervised machine learning methods were used to solve problems within two areas of passive underwater acoustics: characterizing and classifying unlabeled passive acoustic data, and localizing seagoing vessels in passive acoustic recordings. The methodologies incorporated intuition from decades of research development in statistical processing for various underwater soundscapes [1, 2] as well as localization techniques using array processing and physical models of ocean sound propagation. [3–12] This work also leveraged recent advances in machine learning software and algorithm development. [13–15]

Statistical spectral processing methods were discussed for analyzing ambient noise in the Eastern Arctic on a drifting vertical hydrophone array between April and September 2013. Noise sources were observed manually, including ice noise, bowhead whale calling, airgun survey pulses, and earthquake $T$–phases. The data were processed into three and four day median spectral estimates that demonstrate the variation in the occurrence and received level. The median spectral ambient noise estimate for May 2013 was lower than a nearby estimate from April 1982 [16] but similar to an estimate from an ice–covered region in the Beaufort Sea (Western Arctic) in April 1975, [17] indicating that local ice source effects may be as significant as regional effects in determining ambient noise levels in the Arctic. This study demonstrated the importance of environmental factors in data-driven soundscape analysis.

Next, unsupervised machine learning was used to cluster whale song and coral reef fish calling, two major contributors observed in a Hawaiian coral reef soundscape in February 2020. Two clustering approaches were proposed. First, handpicked features known to be associated with coral reef fish species [2, 18, 19] and other relevant acoustic metrics [20, 21] were extracted and clustered using common unsupervised clustering methods. In simulations, the handpicked features overlapped for the signals, reducing unsupervised clustering accuracy. Then, deep embedded clustering (DEC), a deep convolutional network approach, was used to jointly learn features and cluster labels directly from the spectrograms. In simulations, DEC demonstrated higher accuracy and recall than the handpicked features and was more likely to correctly classify existing whale song events. When applied to automatically detected events on the Hawaiian coral reef, DEC achieved separation of whale song events from fish calling but suffered from reduced accuracy due to class imbalance, while clustering of handpicked features did not have good agreement with the manual labels.

In the final chapters, underwater source localization was considered within a machine learning framework. First, source ranging was posed as a supervised learning problem and solved separately using feed-forward neural networks (FNN), support vector machines (SVM) and random forests (RF). Normalized sample covariance matrices (SCM) were used as input to the machine learning models. Simulations showed that FNN achieved good prediction performance for SNR above 0 dB. When applied to experimental data, it was discovered that multi-frequency inputs generated more accurate predictions than single frequency (based on FNN) and that classification methods performed better than regression and MFP methods.

Then, CBF was formulated as linear in the weight covariance matrix and the data SCM and used to directly learn the CBF weights in a perturbed array scenario. The linearized SCM was used as an input to a deep FNN model for estimating DOA. The deep FNN was trained for both a noiseless two source scenario and for $K$ randomly generated number of sources DOAs, with $K \in (1,10)$ and $\theta_k \in [-90°, 90°)$. On a separate test set, both 2-source and $K$-source deep FNN as well as sparse Bayesian learning (SBL) [22] estimated DOA accurately using a single snapshot. Generally, deeper FNN models with increased hidden nodes per layer had improved accuracy. In experimental data with two transiting seagoing vessels, FNN performed comparably to SBL, and both methods had increased resolution over CBF.

Each study in this dissertation demonstrated the potential of using data-driven and machine learning methods for passive underwater acoustics problems. Future work is needed to further develop these methods and broadly apply them to a set of problems. Initially, collection of more good quality acoustic data from regions of interest, including the Arctic and coral reefs, will better enable the advantage in using machine learning and big data processing. The issue of training set diversity for localization problems should be considered more deeply, either by relying on sophisticated propagation models that reflect complexity in the ocean [23, 24] or, preferably, by including larger experimental training sets that represent ocean processes in acoustic channels. Alternate input formats could be used to incorporate relevant physical variation. Last, the algorithms used for each passive acoustics problem should be designed to overcome potential concerns with the environmental and data properties, such as class imbalance among types of acoustic signals or temporal overlap. In some cases, development of new algorithms may be necessary. Overall, the application of recent machine learning methods to passive underwater acoustics is relatively new, but its underlying aim echoes decades of advances driven by the underwater acoustic signal processing community.

# Bibliography

[1] G. B. Kinda, Y. Simard, C. Gervaise, J. I. Mars, and L. Fortier. Arctic underwater noise transients from sea ice deformation: Characteristics, annual time series, and forcing in beaufort sea. *J. Acoust. Soc. Am*, 138:2034–2045, 2015.

[2] T. C. Tricas and K. S. Boyle. Acoustic behaviors in Hawaiian coral reef fish communities. *Mar Ecol Prog Ser*, 511:1–16, September 2014.

[3] B. Z. Steinberg, M. J. Beran, S. H. Chin, and J. H. Howard. A neural network approach to source localization. *J. Acoust. Soc. Am*, 90:2081–2090, 1991.

[4] J. M. Ozard, P. Zakarauskas, and P. Ko. An artificial neural network for range and depth discrimination in matched field processing. *J. Acoust. Soc. Am*, 90:2658–2663, 1991.

[5] A. Caiti and T. Parisini. Mapping ocean sediments by rbf networks. *IEEE J. Ocean. Eng.*, 19:577–582, 1994.

[6] R. Lefort, G. Real, and A. Drémeau. Direct regressions for underwater acoustic source localization in fluctuating oceans. *Appl. Acoust.*, 116:303–310, 2017.

[7] A. B. Baggeroer, W. A. Kuperman, and P. N. Mikhalevsky. An overview of matched field methods in ocean acoustics. *IEEE J. Ocean. Eng.*, 18:401–424, 1993.

[8] C. Debever and W. A. Kuperman. Robust matched-field processing using a coherent broadband white noise constraint processor. *J. Acoust. Soc. Am*, 122:1979–1986, 2007.

[9] L. T. Fialkowski, M. D. Collins, W. A. Kuperman, J. S. Perkins, L. J. Kelly, A. Larsson, J. A. Fawcett, and L. H. Hall. Matched-field processing using measured replica fields. *J. Acoust. Soc. Am*, 107:739–746, 2000.

[10] H. C. Song and Chomgun Cho. Array invariant-based source localization in shallow water using a sparse vertical array. *J. Acoust. Soc. Am*, 141:183–188, 2017.

[11] D. F. Gingras and P. Gerstoft. Inversion for geometric and geoacoustic parameters in shallow water: Experimental results. *J. Acoust. Soc. Am*, 97:3589–3598, 1995.

[12] S. E. Dosso and M. J. Wilmut. Bayesian multiple source localization in an uncertain environment. *J. Acoust. Soc. Am*, 129:3577–3589, 2011.

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, G. S. Corrado C. Citro, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, M. Isard G. Irving, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *Software available from tensorflow.org*, 2015.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res*, 12:2825–2830, 2011.

[15] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[16] N. C. Makris and I. Dyer. Environmental correlates of pack ice noise. *J. Acoust. Soc. Am*, 79:1434–1440, 1986.

[17] B. M. Buck and J. H. Wilson. Nearfield noise measurements from an arctic pressure ridge. *J. Acoust. Soc. Am*, 80:256–264, 1986.

[18] D. A. Mann and P. S. Lobel. Propagation of damselfish (*pomacentridae*) courtship sounds. *J. Acoust. Soc. Am.*, 101(6):3783–3791, February 1997.

[19] K. P. Maruska, K. S. Boyle, L. R. Dewan, and T. C. Tricas. Sound production and spectral hearing sensitivity in the Hawaiian sergeant damselfish, *abudefduf abdominalis*. *J. Exp. Biol.*, 210:3990–4004, 2007.

[20] S. B. Martin, K. Lucke, and D. R. Barclay. Techniques for distinguishing between impulsive and non-impulsive sound in the context of regulating sound exposure for marine mammals. *J. Acoust. Soc. Am.*, 147(4):2159–2176, April 2020.

[21] M. Malfante, J. I. Mars, M. D. Mura, and C. Gervaise. Automatic fish sounds classification. *J. Acoust. Soc. Am.*, 143(5):2834–2846, May 2018.

[22] K. L. Gemba, S. Nannuru, and P. Gerstoft. Robust Ocean Acoustic Localization With Sparse Bayesian Learning. *IEEE J. Sel. Top. Sig. Proc.*, 13(1):49–60, 2019.

[23] David Van Komen, Tracianne B. Neilsen, David P. Knobles, and Mohsen Badiey. A convolutional neural network for source range and ocean seabed classification using pressure time-series. *Proceedings of Meetings on Acoustics*, 36(1):070004, 2019.

[24] H. Niu, Z. Gong, E. Ozanich, P. Gerstoft, H. Wang, and Z. Li. Deep-learning source localization using multi-frequency magnitude-only data. *J.Acoust. Soc. Am.*, 146:211–222, July 2019.