

Seabed type and source parameters predictions using ship spectrograms in convolutional neural networks

David F. Van Komen, Tracianne B. Neilsen, Daniel B. Mortenson, Mason C. Acree, David P. Knobles, Mohsen Badiey, and William S. Hodgkiss

Citation: *The Journal of the Acoustical Society of America* **149**, 1198 (2021); doi: 10.1121/10.0003502

View online: <https://doi.org/10.1121/10.0003502>

View Table of Contents: <https://asa.scitation.org/toc/jas/149/2>

Published by the [Acoustical Society of America](#)

ARTICLES YOU MAY BE INTERESTED IN

[Learning location and seabed type from a moving mid-frequency source](#)

The Journal of the Acoustical Society of America **149**, 692 (2021); <https://doi.org/10.1121/10.0003361>

[Acoustics Apps: Interactive simulations for digital teaching and learning of acoustics](#)

The Journal of the Acoustical Society of America **149**, 1175 (2021); <https://doi.org/10.1121/10.0003438>

[A computational method whose time had come](#)

The Journal of the Acoustical Society of America **148**, R7 (2020); <https://doi.org/10.1121/10.0002055>

[Generative adversarial networks for the design of acoustic metamaterials](#)

The Journal of the Acoustical Society of America **149**, 1162 (2021); <https://doi.org/10.1121/10.0003501>

[Spectral contrast reduction in Australian English //final rimes](#)

The Journal of the Acoustical Society of America **149**, 1183 (2021); <https://doi.org/10.1121/10.0003499>

[Automated two-dimensional localization of underwater acoustic transient impulses using vector sensor image processing \(vector sensor localization\)](#)

The Journal of the Acoustical Society of America **149**, 770 (2021); <https://doi.org/10.1121/10.0003382>



**Advance your science and career
as a member of the**

ACOUSTICAL SOCIETY OF AMERICA

LEARN MORE



Seabed type and source parameters predictions using ship spectrograms in convolutional neural networks^{a)}

David F. Van Komen,^{1,b)} Tracianne B. Neilsen,^{1,c)} Daniel B. Mortenson,¹ Mason C. Acree,¹ David P. Knobles,² Mohsen Badiey,³ and William S. Hodgkiss⁴

¹Physics and Astronomy, Brigham Young University, Provo, Utah, 84604, USA

²Knobles Scientific and Analysis, Austin, Texas, 78731, USA

³Department of Electrical Engineering, University of Delaware, Newark, Delaware 19716, USA

⁴Marine Physical Laboratory, Scripps Institution of Oceanography, University of California, San Diego, La Jolla, California 92093, USA

ABSTRACT:

Broadband spectrograms from surface ships are employed in convolutional neural networks (CNNs) to predict the seabed type, ship speed, and closest point of approach (CPA) range. Three CNN architectures of differing size and depth are trained on different representations of the spectrograms. Multitask learning is employed; the seabed type prediction comes from classification, and the ship speed and CPA range are estimated via regression. Due to the lack of labeled field data, the CNNs are trained on synthetic data generated using measured sound speed profiles, four seabed types, and a random distribution of source parameters. Additional synthetic datasets are used to evaluate the ability of the trained CNNs to interpolate and extrapolate source parameters. The trained models are then applied to a measured data sample from the 2017 Seabed Characterization Experiment (SBCEX 2017). While the largest network provides slightly more accurate predictions on tests with synthetic data, the smallest network generalized better to the measured data sample. With regard to the input data type, complex pressure spectral values gave the most accurate and consistent results for the ship speed and CPA predictions with the smallest network, whereas using absolute values of the pressure provided more accurate results compared to the expected seabed types.

© 2021 Acoustical Society of America. <https://doi.org/10.1121/10.0003502>

(Received 14 October 2020; revised 18 January 2021; accepted 22 January 2021; published online 18 February 2021)

[Editor: Zoi-Heleni Michalopoulou]

Pages: 1198–1210

I. INTRODUCTION

Large container and oil tanker merchant ships produce broadband noise as they move through the ocean. The emitted noise propagates through the ocean waveguide and, thus, contains information about the seabed for bottom limited environments. These ships of opportunity (SOO) tend to travel along straight tracks at a constant velocity (along shipping lanes) and the automatic identification system (AIS)¹ provides sufficient information to estimate their range and speed. The broadband noise from SOO has been studied, and a model for the broadband spectral shape was developed by Wales and Heitmeyer,² which represents the noise from the highly nonlinear interaction between turbulence from the ship hull and propeller motion. The propagation of the broadband SOO noise through the ocean environment means that received SOO spectrograms contain significant information about the seafloor. This information can be used to infer the seabed properties as well as the closest point of approach (CPA) range and ship speed.

SOO noise has been used for seabed characterization in optimization algorithms. Battle *et al.*³ applied near-

matched-field processing for geoacoustic inversion to ship noise received on a towed horizontal line array (HLA) at 356 Hz and noted that even quiet ships radiate enough noise for basic geoacoustic inversion. Nicholas *et al.*⁴ used three low-frequency tones received on an *L*-shaped array to perform environmental inversion and matched-field tracking and noted that the vertical aperture of the array provides environmental information and ship range and depth information, whereas the horizontal aperture provides ship bearing information. Heaney⁵ noted the challenges of predicting the full ocean response due to environmental mismatch and variability with range and instead moved to prediction of average field levels and time spreads of the acoustic field in the ocean. Heaney then used SOO noise (200–500 Hz) to perform geoacoustic inversion of the ship radiated noise from a single hydrophone. Park *et al.*⁶ used the SOO signals on a towed HLA in a time-reversal technique to estimate the geoacoustic parameters. Koch and Knobles⁷ used 15 min of SOO noise (at 120–200 Hz) on subapertures of a HLA to obtain geoacoustic parameter values. Tollefsen and Dosso⁸ quantified the information content of five low-frequency components received on a moored HLA and concluded that problems with low signal-to-noise ratio (SNR) can be overcome by combining multiple data segments, and Stotts *et al.*⁹ used a 10–250 Hz spectrogram to examine the impact of interfering noise signals on the parameter estimates.

^{a)}This paper is part of a special issue on Machine Learning in Acoustics.

^{b)}Electronic mail: david.vankomen@gmail.com, ORCID: 0000-0003-0610-0806.

^{c)}ORCID: 0000-0002-9729-373X.

Other studies have expanded the geoacoustic inversion problem through waveguide invariant extraction and other statistical estimation techniques. Gervaise *et al.*¹⁰ used inversion techniques on SOO noise (50–500 Hz) from a single hydrophone to extract the waveguide invariant for geoacoustic inversion, although they concluded that attenuation cannot be estimated accurately with their inversion scheme and density and sound speed estimation required injection of *a priori* information about the bottom. The same study notes [see Eq. (2)] that if the ship range varies slowly enough, then effects from the Doppler shift can be ignored. Crocker *et al.*¹¹ measured SOO noise with an autonomous underwater vehicle and used Bayesian methods (with one-third octave band levels 125–8000 Hz) to obtain posterior probability distributions for the properties of a lower half-space. Byun *et al.*¹² applied ray blind deconvolution to the vertical line array (VLA) data at 200–900 Hz to estimate the Green's function for the ocean environment and inform matched-field processing. Gemba *et al.*¹³ calculated channel impulse responses from 2.5 s of SOO noise measured on a VLA by estimating ray paths with sparse Bayesian learning on three VLAs. Muzi *et al.*¹⁴ used a VLA and time-averaged SOO noise (5–500 Hz) to estimate passive bottom loss and noted that incorrect bottom loss estimation is a major source of error in sonar performance prediction. Xu *et al.*¹⁵ used joint time-frequency inversion for seabed parameters with ship noise recorded on a VLA. Tollefsen and Dosso⁸ used Bayesian techniques with ship noise on a towed HLA to obtain estimates of geoacoustic parameters and uncertainties and showed how ship range and the bearing relative to the HLA are related to these uncertainties.

SOO noise has also been used for source parameter estimation such as in the source range. For examples, Koch and Knobles⁷ and Stotts *et al.*⁹ simultaneously obtain geoacoustic and source track parameter values. Tollefsen and Dosso have applied a nonlinear Bayesian marginalization technique to track a surface ship in three dimensions.^{16,17} As extensions of this study, research was done into the impact of multiple sources¹⁸ and multiple fixed arrays.¹⁹

Deep learning efforts have also used SOO noise for source localization. Huang *et al.*²⁰ trained networks on extracted features (500–1000 Hz) from simulated surface ships to predict the range at individual time steps on data from a 16 element VLA. Niu *et al.*^{21,22} used deep learning techniques for SOO localization by calculating normalized complex sample covariance matrices (300–900 Hz on a VLA) from simulated ship noise in varied environments. More recently, Niu *et al.*²³ used complex pressure values (100–200 Hz) on a single hydrophone in a multistep approach with an initial residual network identifying large source range classes and then used a secondary set of residual networks for smaller classes for the source range (based on predictions from the first network) and depth. Ozanich *et al.* used both a deep network²⁴ and a feed-forward neural network approach²⁵ to estimate the direction of the arrival of ship sources. Ferguson *et al.*²⁶ used convolutional neural networks (CNNs) on 0.1 s waveguide invariant-based

cepstrums up to 10 kHz calculated from a broadband source from a single sensor recording of a passing motorboat. Liu *et al.*²⁷ used the magnitude of spectrum values (550–1050 Hz) on a VLA with CNNs to localize towed transmitter sources through time with simulated and measured samples.

This paper moves beyond these prior deep learning approaches in that broadband SOO spectrograms are now employed to yield information about both seabed properties and the source track. Specifically, SOO spectrograms (300–1500 Hz over a 15 min time interval) on a single sensor are used in CNNs to classify a seabed type and estimate the ship's CPA and speed. In this work, only four seabed classes are used, representing a sparse characterization of the seabed. Three different multitask CNN architectures are employed to investigate the impact of the CNN depth and number of network parameters on generalizability, i.e., the ability of the CNN to make predictions on data drawn from a different statistical distribution than the training set. A comparison based on input data type is also presented to decide if complex values, squared magnitudes, or levels work best. Synthetic SOO spectrograms are used during training due to a lack of labeled field data. The trained networks are applied to additional synthetic datasets to evaluate their ability to interpolate and extrapolate. Finally, the trained networks are applied to SOO noise measured during the 2017 Seabed Characterization Experiment (SBCEX 2017). These tests show the potential for CNNs to distinguish a seabed type, ship speed, and CPA range from broadband spectrograms and highlight the importance of avoiding overfitting with very deep networks.

II. BACKGROUND

The CNNs are trained on synthetic data because one measured SOO spectrogram was available for an initial test of the networks. This measured spectrogram is from the Kalamata and was recorded during the SBCEX 2017 (Ref. 28) in the New England Mud-Patch. Though many sensors were present during the experiment, the data used in this study were collected at a VLA deployed by the Marine Physical Laboratory of the Scripps Institution of Oceanography (MPL), which was located in the southeast (SE) region of the experimental area of the study at [40.442°N, 70.527°W]. Denoted VLA 2, this 16 hydrophone array had an element spacing of 3.75 m with the lowest sensor at 5 m above the ocean floor. For this preliminary study, only the recording from channel 8 (33 m from the ocean floor in approximately 75 m of water) was used.

VLA 2 recorded the passage of the Kalamata. According to the AIS database, the Kalamata container ship had a CPA to the VLA of 3.29 km and was traveling at 19.9 kn. The draft of the hull is 8 m. The measured Kalamata spectrogram from channel 8 of VLA 2 is shown in Fig. 1(a). The spectrogram is calculated via a fast Fourier transform on the pressure time series, sampled at 25 000 Hz, using 50% overlap and blocks of 2¹² samples. Due to low-frequency background noise, the 300–1500 Hz band is

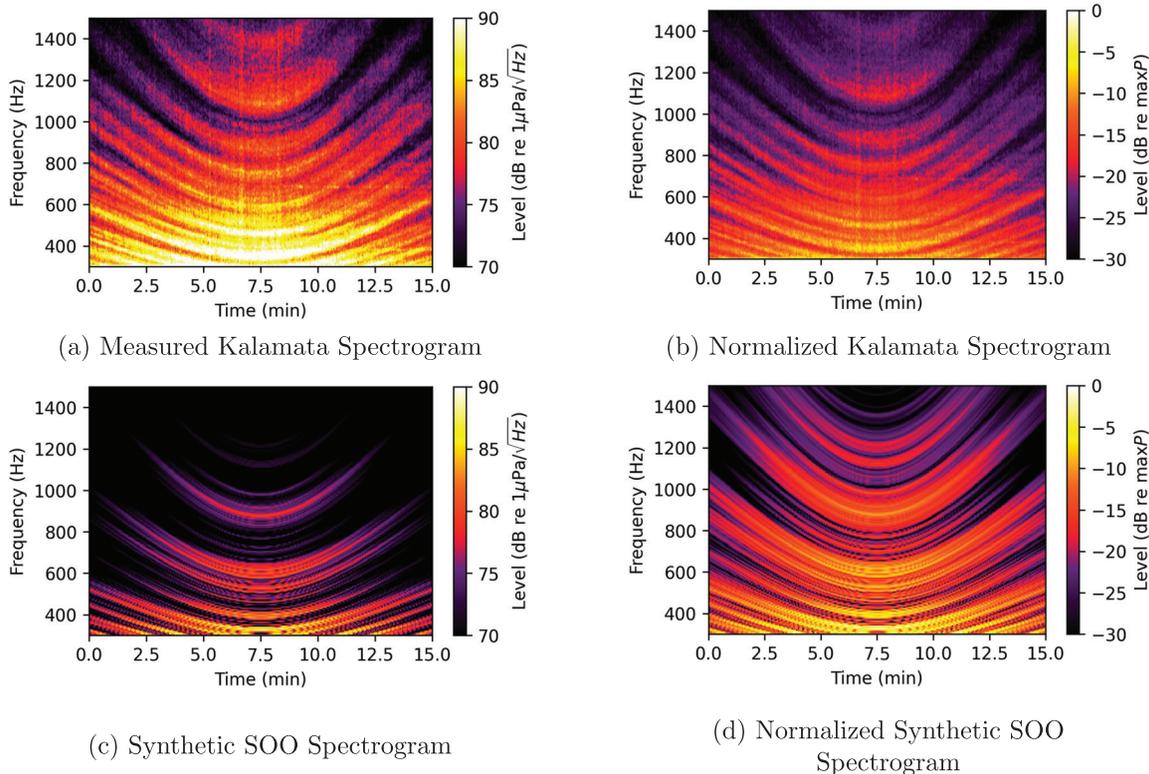


FIG. 1. (Color online) Example SOO spectrograms. (a) Absolute and (b) normalized spectrograms of the Kalamata measured on channel 8 of the Marine Physical Laboratory of the Scripps Institution of Oceanography (MPL) VLA 2 during SBCEX 2017. (c) Absolute and (d) normalized synthetic spectrograms for a ship traveling 20.0 kn with a CPA of 3.3 km—similar to the Kalamata—using the mud-over-sand seabed type. The “normalized” spectrograms use the maximum pressure of the spectrogram as the reference instead of 1 μPa .

selected for this work. The resulting 15 min spectrogram has the CPA at 7.5 min, which centers the characteristic “U” shape of ship spectrograms. Although the ship noise is visible over a longer period of time, 15 minutes was selected to balance computation time and storage needs of the synthetic datasets (Sec. III A).

The authors realize that one data sample is not sufficient to fully test deep learning models. To address this concern, two synthetic datasets are generated for validation purposes, and multiple network architectures and types of input data are explored. Ten training instances are executed for each network-input data combination and applied to two validation datasets and the measured data sample. The use of ten training instances allows for a statistical look at performance, taking into account the random initializations of the weights in the networks. These validation datasets combined with the multiple networks provide an in-depth look at how using CNNs on SOO spectrograms could perform and consists of the bulk of the discussion in this work.

III. METHODS

Supervised deep learning models require training datasets of many labeled samples. Due to the scarcity and expense of labeled field data, synthetic data are used during the training and validation steps. Nearly 50 000 SOO spectrograms are simulated using four different seabed types with water column sound speed profiles (SSPs)

representative of the SBCEX 2017 experiment for numerous ship speeds and CPAs to provide datasets for training and validation of the CNNs. These trained CNNs are then applied to measured data as preliminary evidence of how SOO spectrograms can be used to obtain estimates of the seabed type and source parameters. Details regarding these steps are provided in Secs. III A–III C.

A. Synthetic data

Synthetic SOO spectrograms are generated using a modeled source spectrum and a range-independent, normal mode model. The source spectrum of the radiated ship noise is approximated by the ensemble source model described by the unlabeled equation in Sec. 3 of Wales and Heitmeyer.² The source spectral density level in dB as a function of frequency, f , is calculated as

$$S(f) = S_0 - 10 \log(f^{3.594}) + 10 \log((1 + (f/340)^2)^{0.917}), \quad (1)$$

where S_0 might be described as the “y-intercept” of the source spectrum. The levels from Eq. (1) are converted to pressures (in Pa) and a random phase is assigned for each frequency. The source pressure spectrum is then multiplied by the Green’s function of the propagation to simulate the received pressure spectrum.

The Green’s function is modeled with ORCA,²⁹ a range-independent, normal mode model for acousto-elastic ocean environments. The simulated ocean environments include different combinations of SSPs and four seabed types. The four representative seabed types selected for this study are the same as those used in Van Komen *et al.*:³⁰ (1) deep mud,³¹ (2) mud-over-sand,³² (3) sandy-silt,³³ and (4) sand.³⁴ The ocean SSPs are taken from measurements made during the SBCEX 2017, shown in Fig. 2, with the ocean depths of the profiles selected between 73 and 78 m to approximate changes in the bathymetry across the area spanned by the experiment. A fixed ocean depth was randomly selected for each SSP-seabed combination for use in the range-independent calculations of the ORCA model. The seabed at the SBCEX 2017 is known to contain mud,²⁸ but the approach of this paper could be applied if seabed properties were unknown and an approximate seabed classification were desired from the SOO data.

The SOO source is modeled with a quasi-static assumption corresponding to a point source at discrete locations throughout time. The SOO are assumed to travel in a straight line at a constant velocity. No Doppler shift was included due to the effects being small at the near-CPA range for the frequencies used.¹⁰ CPA range—the minimum range between the ship and the hydrophone—is assumed to occur at the center of the time vector, t . Range positions over the entire interval, r , are calculated based on time and speed, and the corresponding Green’s functions are obtained.

The frequency-dependent Green’s functions are multiplied by the source spectrum to calculate the received complex spectrum, $\tilde{P}(f)$, at each time position. By iterating through each range/time, the spectra are then combined to form a complex matrix of the time-evolving complex pressure, \mathbf{P} , with $\mathbf{P}_{ij} = \tilde{P}(f_i, t_j)$. This process is repeated for all selected combinations of seabed type, SSP, ship speed, CPA range, and effective ship source depth to generate the synthetic dataset.

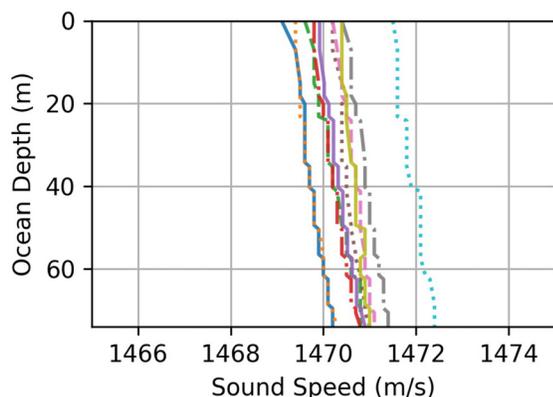


FIG. 2. (Color online) Ten SSPs measured during the SBCEX 2017. Random water depths were used with each SSP to create the 50 training and 30 validation SSPs. These profiles are slightly upward refracting, but the range of sound speeds in the x axis exaggerates such effects.

The frequency and time dimensions of the synthetic \mathbf{P} matrices must match those of the final data being tested. In this study, decisions were guided by considering a prominent SOO event recorded by MPL VLA 2: the passage of the Kalamata (details in Sec. II). The measured Kalamata spectrogram (the time-evolving power spectrum) is shown in Fig. 1(a) for the 300–1501.7 Hz band (with a Δf spacing of 6.1 Hz) over 15 min (with time step intervals of 3 s). This frequency band and time span were chosen as a result of the clarity of the striations in the measured spectrogram. Thus, each synthetic data sample, \mathbf{P} , contains 301 time steps and 198 frequencies.

An example spectrogram generated for the mud-over-sand bottom at the same CPA and with the same speed as that of the measured ship is shown in Fig. 1(c) for comparison with the measured spectrogram from the Kalamata ship [Fig. 1(a)]. The difference between the measured and synthetic power spectral levels occurs because the exact source level of the Kalamata is unknown, corresponding to an unknown S_0 in Eq. (1). To circumvent this limitation, each \mathbf{P} is normalized by its absolute maximum. The normalized spectral levels, \mathbf{L}_P , are shown in Figs. 1(d) and 1(b). This normalization discards information about the range that is contained in both the absolute levels and the relative levels between individual samples of differing source and receiver parameters, thus, presenting an additional challenge for deep learning.

The normalization used in this study differs from recent work by Niu *et al.*²³ in which different frequency bands (over the 100–200 Hz interval) were normalized separately to account for the frequency-dependent source spectrum. This frequency-band normalization is not needed in the current work because (1) the 300–1500 Hz band is above any apparent tones, and (2) the Wales-Heitmeyer² model is used to synthesize the SOO spectral shape leaving only a single parameter (S_0) that is unknown.

The synthetic training dataset needs to contain sufficient variability to account for the real-world conditions that exist in the measured data as discussed in Van Komen *et al.*³⁰ To provide variability, the training dataset was generated with the following parameters. To represent the ocean, 50 different SSPs were used in combination with the four designated seabeds. Ten SSPs were drawn from measurements made during the SBCEX 2017 shown in Fig. 2. For each SSP, five water depths were randomly assigned over $75 \text{ m} \pm 0\text{--}3 \text{ m}$ (while preserving the SSP gradient). The distribution of the randomly assigned ocean depths across the 50 SSPs is shown in Fig. 3(d). The 50 SSPs combined with each of the four seabeds creates 200 ocean environments used for simulation.

For each of these environments, nine ship speeds, nine CPA ranges, and two source depths were selected. The combinations of these source parameters and the 200 environments produced a dataset containing 32 400 samples for training. Previous internal studies showed that using randomly drawn source parameters improved the generalizability of the network as long as a sufficient number of samples

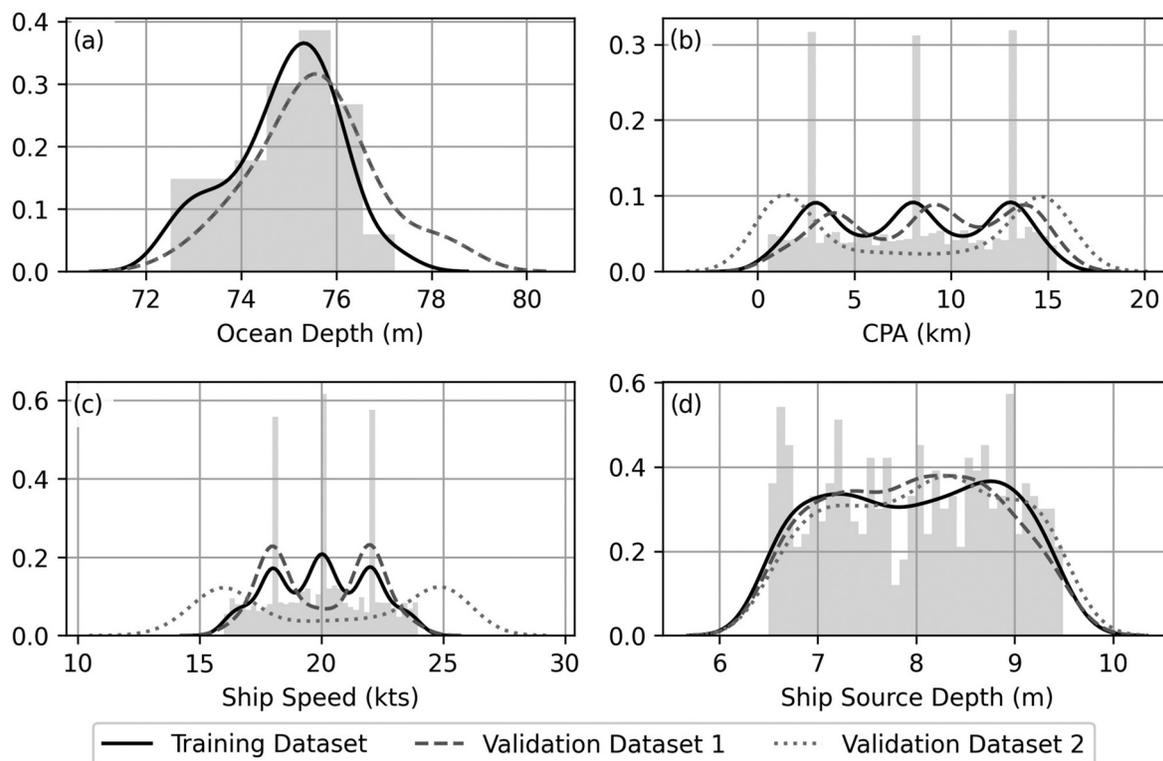


FIG. 3. Normalized histograms (shaded gray areas) and kernel density estimates (solid lines) of the random parameters selected for the training dataset. (a) Ocean depths over the 50 SSPs, (b) CPA ranges, (c) ship speeds, and (d) source depths over the 200 environments. The large peaks correspond to values that were selected for each environment. The dashed and dotted lines represent the kernel density estimates of the various parameters for the two validation sets.

were selected over the entire span of the labels of interest. Maintaining fixed values in the dataset has the potential to reinforce common values that are present in the real-world data. To follow these guidelines, both fixed and random sets of the source parameters were used for each of the 200 environments. The nine CPA values for each environment included three fixed values (3, 8, and 13 km) and six additional ranges randomly drawn between 0.5 and 15.5 km. The ship speed values included three fixed values (18, 20, and 22 kn) and six random values drawn between 16 and 24 kn. The two ship depths were randomly drawn to be between 6.5 and 9.5 m for each environment. Histograms of the source parameters over the 200 environments used in the training are shown as gray boxes in Fig. 3. The peaks in the distributions for the CPA and ship speed correspond to the fixed values, selected to ensure that each of the 200 environments contained several of the same source parameters as a means of consistency. Whereas the simulations could be improved further by employing additional parameter randomness at each of the selected time windows in the spectrograms, this step was omitted for consistency and processing constraints while generating the data. The histogram is converted to a smooth line using kernel density estimates (solid lines) generated via the “distplot” function from the Seaborn Python library.

To test the ability of the networks to interpolate between labels used in the training data samples and extrapolate beyond the edges of the training labels, two validation

datasets were generated using the same procedure as was used for the training dataset. To test the ability of the networks to interpolate, validation dataset 1 (5400 samples) contains source parameters similar to those encountered in the training dataset but with slight adjustments. Thirty SSPs were used (from the same ten measured SSPs but with only three random water depths each). Four ship speeds were selected (18 and 22 kn for every environment and two randomly drawn between 16 and 24 kn), and nine CPAs were drawn (4, 9, and 14 km for every environment along with six random CPAs between 0.5 and 16 km). The remaining parameters were the same. The kernel density estimates of the water depths and source parameters used in validation dataset 1 are shown as dashed lines in Fig. 3. Validation dataset 1 measures the ability of the trained networks to interpolate.

The ability of the networks to perform at the edges of the training domain and extrapolate beyond are tested with validation dataset 2. For this second validation dataset, the same 30 SSPs were used as for validation dataset 1, but the CPAs (1, 2, 14, and 15 km for every environment with eight random CPAs between 0.5 and 16 km) and ship speeds (16 and 25 kn for every environment with two random CPAs between 14 and 26 kn) were chosen to be clustered closer to the edges of the values used for the training dataset. Validation dataset 2 contains 8640 samples. The kernel density estimates of the water depths and source parameters used in validation dataset 2 are shown as dotted lines in

Fig. 3. Results of the training and validation are discussed in Sec. IV A.

1. Data and label preparation for training

Because spectra can be represented as complex values or magnitudes (in Pascals) or levels (in decibels), the type of input that should be used with CNNs is studied in this paper. This work evaluates if CNNs learn better on the complex values \mathbf{P} , the squared absolute value of those values $|\mathbf{P}|^2$, or the levels \mathbf{L}_P . For \mathbf{P} , the real and imaginary parts are split into two channels in which the real part is channel 0 and the imaginary part is channel 1, analogous to the red, blue, and green channels in digital images. $|\mathbf{P}|^2$ and \mathbf{L}_P are input as single channels, similar to the single black and white channel. As mentioned in Sec. III A, all samples are also individually normalized before being passed to the network. For the i th data sample, $|\mathbf{P}|^2$ is normalized by $|\mathbf{P}|^2_i/|\mathbf{P}|^2_{i,max}$ and $|\mathbf{P}|$ is normalized by $|\mathbf{P}|_i/|\mathbf{P}|_{i,max}$. In the case of the levels input type, the normalization is applied as $\mathbf{L}_{P,i} = 20 \log (|\mathbf{P}|_i/|\mathbf{P}|_{i,max})$.

Another preparatory step was the normalization of the CPA and ship speed labels (a common practice with neural networks³⁵). Instead of the networks learning raw values in meters and knots, the networks learn scaled labels. Since the CPA in the training set varies between 0.5 and 15 km and the ship speed varies between 16 and 24 kn, this scaling forces the values to be on the same scale, which has the potential to expedite learning and mitigate potential biases due to the different scales. This scaling corresponds to taking the distributions of each label (Fig. 3) and scaling and shifting each distribution to lie between 0 and 1. While different types of scaling may be employed, in this work, each set of labels, y , are scaled via

$$\hat{y}_i = (y_i - \min y)/(max y - \min y), \tag{2}$$

where \hat{y} are the scaled labels, y_i corresponds to the raw label of data sample i , and the bold font indicates the vector of labels for all of the data samples.

B. CNN topology

For this study, three different neural networks were selected to make simultaneous predictions on the seabed type, CPA, and ship speed. The first two CNNs were hand-designed and incorporate different numbers of convolutional layers with the full network architectures shown in Fig. 4. The third CNN is an implementation of “AlexNet,” presented in Krizhevsky *et al.*,³⁶ originally designed to classify between 1000 different classes of images in the ImageNet³⁷ database.

Although various opinions on CNN design exist in the community, the two hand-designed CNNs were created with simplicity in mind and to compare the effect of depth and number of parameters on predictions. The first CNN, named Selkie3, has three convolutional layers, each followed by maximum pooling layers. These convolutional layers are followed by two hidden layers before reaching the output

layer. The second CNN (Selkie5) differs by having five convolutional layers with no pooling. In both networks, the convolutional layers use Rectified Linear Unit (ReLU) activations and are followed by batch normalization. Selkie3 was designed to include pooling to reduce the number of features in each layer while using small filters (kernels) to reduce the number of learnable parameters. Selkie5 uses more convolutional layers with more channels to give the model more learnable parameters. The stride parameters of the convolutions reduce the features within each layer instead of allowing that reduction to come directly from the pooling layers (as in Selkie 3). Selkie5 has larger hidden layers than Selkie3 has because more parameters in the hidden layers can allow more complex linear transformations. Selkie3 has 4.3×10^6 parameters, whereas Selkie5 has 14.2×10^6 parameters. Both networks used kernel sizes and strides in the convolutional layers to allow an overlap across the input and intermediate spaces in an effort to learn patterns across differing bands of frequencies and time steps. No formal hyperparameter tuning study was done to find the most optimal set for these particular networks; such a study is beyond the scope of the current work.

The implementation of AlexNet used in this paper comes from using half of the network size depicted in Fig. 2 of Krizhevsky *et al.*³⁶ and is referred to as “HalfAlexNet” throughout the current study. Krizhevsky *et al.*³⁶ mention dividing the network for simultaneous training across two graphics processing units (GPUs), and one of those halves was implemented for this study. HalfAlexNet has over 15.6×10^6 learnable parameters.

C. Training CNNs

The implementation of the neural networks mentioned in Sec. III B was done in Python 3 using the open-source PyTorch framework³⁸ (version 1.5.0). PyTorch was written using Python-native syntax with a focus on speed and usability and provides the tools required for loading data, building models, and training on a GPU using algorithms standard to the machine learning community. In particular, this study uses the PyTorch implementation of the Adam optimizer³⁹ for stochastic optimization during training. As with the procedure detailed in Van Komen *et al.*,⁴⁰ a cosine annealing learning rate was used to limit the variable learning rate used by the Adam optimizer. (The method of warm restarts in the original paper by Loschilov *et al.*⁴¹ was not used.) The annealing learning rate was implemented to expedite learning and prevent overfitting.

Because of the random initialization of the weights and splits of the dataset into batches, ten instances of each network type were trained for each type of input. Training multiple networks allows a statistical approach to evaluating the model’s performance as every new instance of a model is initialized with random values for all parameters. Each network was trained for 50 epochs to provide a baseline for comparisons by giving each network trained the same

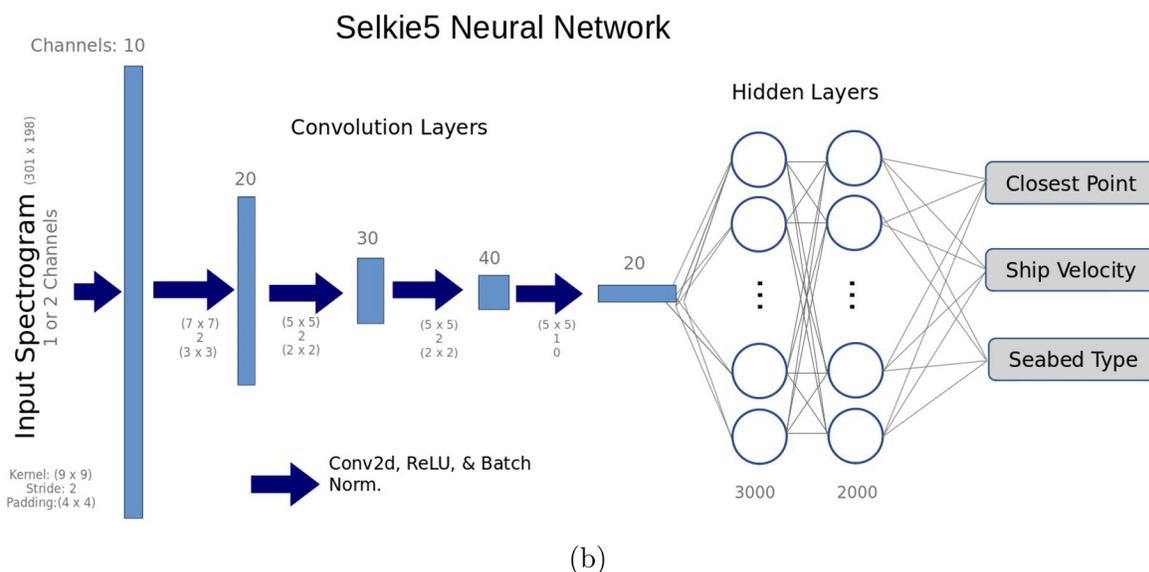
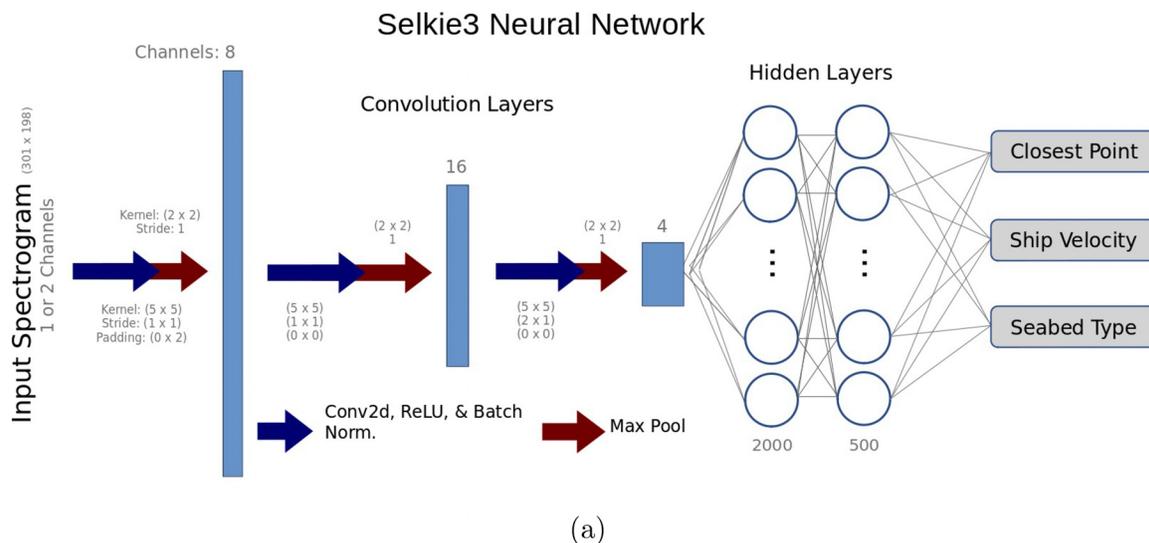


FIG. 4. (Color online) Network topology for the two hand-designed CNN networks in this study. (a) Selkie3 features three convolutional layers, each followed by a maxpooling step. (b) Selkie5 features five deeper convolutional layers with no pooling steps. Each network also has two hidden layers before the output layer.

number of iterations during training, as well as to reduce the amount of training time and avoid common overfitting problems when networks are trained for too long. Results in Sec. IV illustrate that 50 epochs was sufficient, although future work seeks to improve these outcomes. Early stopping was not implemented because of the large number of networks being trained with an annealing learning rate and the self-scaling loss function (detailed in Sec. III C 1). The networks were trained on a (NVIDIA Tesla T4 GPU, Santa Clara, CA) to accelerate learning.

1. Multitask learning

The final decision for CNN implementation is how to assign outputs for the final layer. Networks learn to make predictions through a loss function that determines how predictions are compared to assigned labels during training.

Through the selection of a loss function, these outputs can be trained to make different types of predictions such as regression or classification. Prior studies using a one-second pressure time series^{30,40} and towed tonals⁴² used a regression approach to learn a number representing both the seabed type and source labels. Although some success was seen using regression to predict a seabed type, regression loss functions generally rely on a definition of distance between the truth and prediction. Whereas differences in range, speed, or depth have a physical connection to distance, the choice to represent the seabed type as discrete numbers does not have a physical connection to distance. The seabed types (used in Refs. 30 and 40, and the current study) were ordered from highest to lowest bottom loss to provide a sequential progression, but the “distance” between seabeds 1 and 2 is not the same as the distance between seabeds 2 and 3. Thus, in this work, the seabed type is found through classification

while the ship speed and CPA range are found via regression.

To accommodate both types of predictions, the multitask method proposed in Kendall *et al.*,⁴³ which weighs losses for individual tasks by considering the homoscedastic uncertainty through learning the weighting, was implemented. This multitask method works by calculating an original loss value for each of the outputs of the network (mean squared error for regression and cross-entropy for classification) and then scaling each loss by a learnable parameter. When building the network, the final output layer needs to include enough neurons to accomplish each of the tasks. For this particular study, out of the six output neurons used, the first four were reserved for classification on the seabed type and the last two were reserved for the CPA and ship speed. Although more details on this implementation of scaling loss can be found in Ref. 43, we also implemented a variable transformation⁴⁴ of the learnable parameters σ_i via $\eta_i = 2 \log \sigma_i$. This transformation changes Eq. (7) in Ref. 43 to

$$\sum_{n=1}^i \frac{1}{2\sigma_i} \mathcal{L}_i + \log \prod_{n=1}^i \sigma_i = \frac{1}{2} \sum_{n=1}^i (e^{-\eta_i} \mathcal{L}_i + \eta_i), \quad (3)$$

where \mathcal{L}_i is the loss obtained for each task i . This transformation avoids numerical instability as the original equation limits $\sigma \in \mathbf{R}_{>0}$ and allows $\eta \in \mathbf{R}$.

Thus, during training, the network learns how to properly scale the loss values for each of the tasks. In this particular study, the initial η_i were selected to be 1.0, 0.5, and 0.8 for the seabed type, CPA, and ship speed, respectively. No exhaustive testing was performed on what the initial η_i should be. However, because these values are learnable parameters, the network adjusts them to minimize overall loss along with the other network parameters, therefore, these initial η_i were sufficient.

IV. RESULTS AND DISCUSSION

The results of the predictions from the trained neural networks are presented in this section. In particular, model validation and model generalization are presented. As explained in Neilsen *et al.*,⁴² the difference between validation and generalization of the CNN models is important, especially when using both synthetic and measured data. For the validation results, we present CNN predictions on synthetic SOO spectrograms with similar environments as the training data but different source parameters and slightly varied water depths. For the generalization result, we present CNN predictions on a measured SOO spectrogram. To provide a statistical representation of the CNN performance, the metrics used to quantify predictive ability for the regression tasks are root mean squared error (RMSE) and standard deviation of the RMSE over the ten training instances for a combination of CNNs and input data type. For the classification task, the percentage of times the correct class has the highest output is defined as the accuracy.

A. Validation

Two methods of validation are presented in this section. The first is k -fold cross-validation,⁴⁵ in which the training dataset is divided into k random splits. The network trains on $k - 1$ of those splits and tests on the remaining splits. The network is then reinitialized and training is performed on the next combination until all k blocks have been used for testing.

To validate that all three networks are learning relationships between the features of the training dataset and the six outputs, k -fold cross-validation was done with $k = 5$. As an example, the fivefold validation results on the complex \mathbf{P} inputs are shown in Table I. For seabed type predictions, HalfAlexNet has the highest accuracy of 98.98%, although Selkie3 and Selkie5 trail by less than a percentage point. For the CPA predictions, HalfAlexNet has the lowest RMSE of 0.28 km, whereas the RMSE for Selkie3 and Selkie5 are approximately 0.05 km greater. For ship speed, Selkie3 has the lowest RMSE at 0.61 kn, similar to Selkie5, whereas HalfAlexNet has a speed RMSE of 1.04 kn. These results show that overall the networks perform similarly and are learning features that allow for estimations of the seabed type, CPA, and ship speed, although more testing is required.

The models trained through k -fold cross-validation are not used for further testing in this study. Instead, ten instances of each CNN/input data type combination are trained on 100% of the training dataset (32 400 samples) as an alternative to a random subset as in the k -fold cross-validation. These fully trained models are then applied to the synthetic validation datasets (Sec. III A) and measured data sample.

The second type of validation involves testing the networks on different separate datasets simulated in the same manner as for the training dataset but with different values for water depth, CPA range, and speed as shown in Fig. 3. The results of the ten training instances of each network applied to the 5400 samples in validation dataset 1 are displayed in Fig. 5. The metrics used are the average inaccuracy of the seabed classification (100 - accuracy) and the RMSE of the CPA and ship speed. The results from all three networks (colors) and all three input data types (horizontal axis) are shown in Fig. 5. The numbers in boxes indicate the exact height of the bars. The error bars indicate the standard deviation over the ten trained instances for each CNN.

Validation dataset 1 was designed to investigate the ability of the trained networks to interpolate between source parameters in the presence of a slight ocean depth mismatch

TABLE I. Results from fivefold cross-validation tests using \mathbf{P} as input. The bold numbers indicate the highest performing network for that metric, although results for all three networks are similar.

Network	Seabed accuracy	CPA RMSE	Speed RMSE
Selkie3	99.55%	0.33 km	0.61 kn
Selkie5	98.98%	0.34 km	0.63 kn
HalfAlexNet	99.84%	0.28 km	1.04 kn

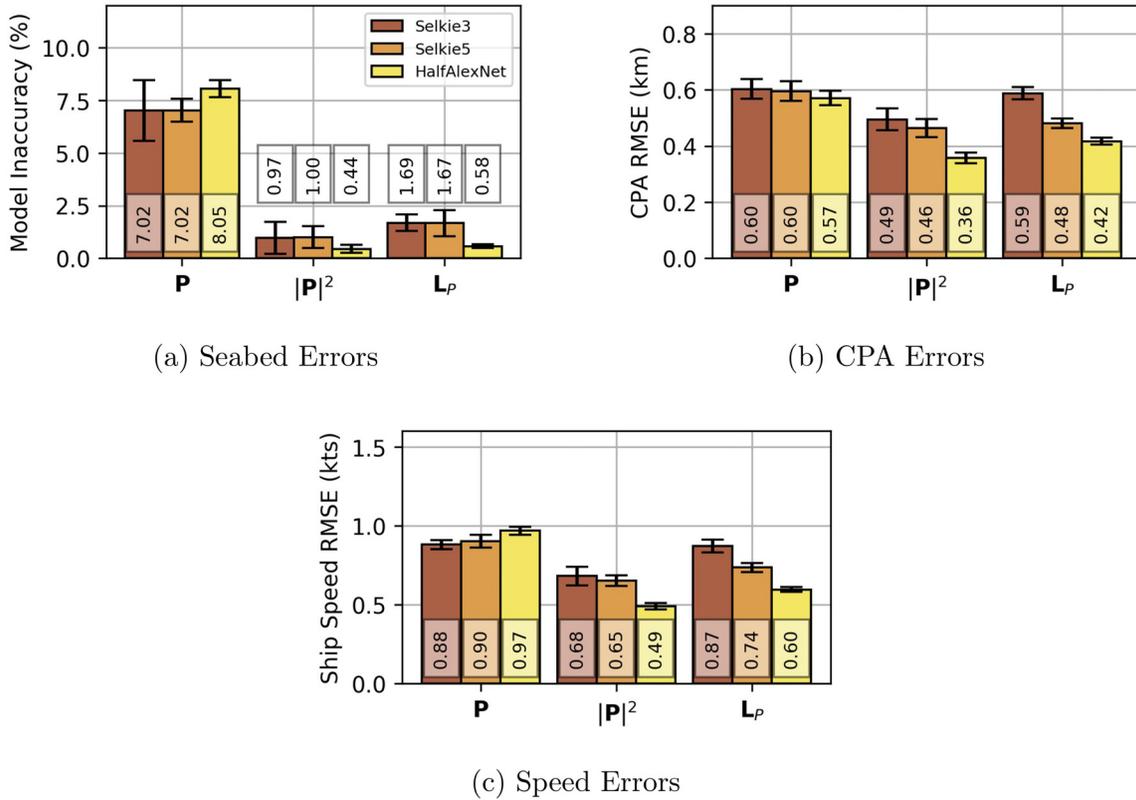


FIG. 5. (Color online) Results from ten training instances of each network on validation dataset 1 containing 5400 samples and designed to test the ability of the networks to interpolate (see Fig. 3). Results for (a) the seabed prediction inaccuracy, (b) the RMSE for the CPA range, and (c) the RMSE for the ship speed are grouped by input type on the horizontal axis with color indicating the network type. The height of the bars and numbers in boxes indicate the mean values, and the error bars indicate the standard deviation over the ten training instances.

(see Fig. 3). For validation dataset 1, all seabed predictions have an accuracy above 90%, which indicates that all of the networks are learning features required for seabed classification. Analysis of individual results shows—across all networks—that misclassifications generally involved confusion between the sandy-silt seabed and the sandy seabed. This confusion is understandable because under visual inspection, synthetic spectrograms for the same ship speed and CPA using the sandy-silt and sandy seabeds appear nearly identical to the naked eye as a result of the similarities in propagation over the distances involved and frequencies included. The network that performed best on seabed predictions was HalfAlexNet, using $|P|^2$ as input, although the L_p input performs less than 0.2% poorer. Selkie3 and Selkie5 give seabed predictions similar to those of HalfAlexNet. This similar performance is significant because Selkie3 has around 10×10^6 fewer parameters than Selkie5 and HalfAlexNet have.

For the CPA range and speed, HalfAlexNet performs best on $|P|^2$, but the input data type impacts the predictions more than the network architecture does. As shown in Fig. 5, HalfAlexNet has the lowest RMSE on the CPA range and speed, but the RMSE values in the other networks differ by less than 0.15 km for the CPA and less than 0.2 kn for the ship speed. The mean absolute percent errors (MAPE) for all three networks were calculated, although they vary by less than 1%. A larger difference is seen when based on the

input data type. For this first validation set, $|P|^2$ gives lower errors for both the CPA and ship speed.

Validation dataset 2 was designed to test the ability of the networks to extrapolate. Validation dataset 2 has the same ocean depth mismatch as for validation dataset 1, but the distribution of the CPAs and speeds are concentrated on the edge of and outside the values used in the training as shown in Fig. 3. The average performance on the ten network instances on the 8640 samples of validation dataset 2 is summarized in Fig. 6. The performance on validation dataset 2 is generally worse than that on validation dataset 1, as expected, because of the extreme nature of the speed and CPA selection used, the networks were never trained on speeds outside 16–24 kn, and few CPAs were included outside the 1–15 km range. For these edge cases, the networks are still near or above 90% accuracy on seabed classification. This time, HalfAlexNet on L_p performs the best on seabed classification with an accuracy of 97.83%, although Selkie5 and Selkie3 still have accuracies greater than 97% for these source parameter edge cases.

The biggest increase in error between the two validation datasets is in the CPA and speed predictions as expected because many of them extend beyond the ranges used during training. When looking at individual network predictions, the speed and CPA ranges tend to be underpredicted at the largest ranges and speeds, whereas they are generally overpredicted at the smallest ranges and speeds. In part, these

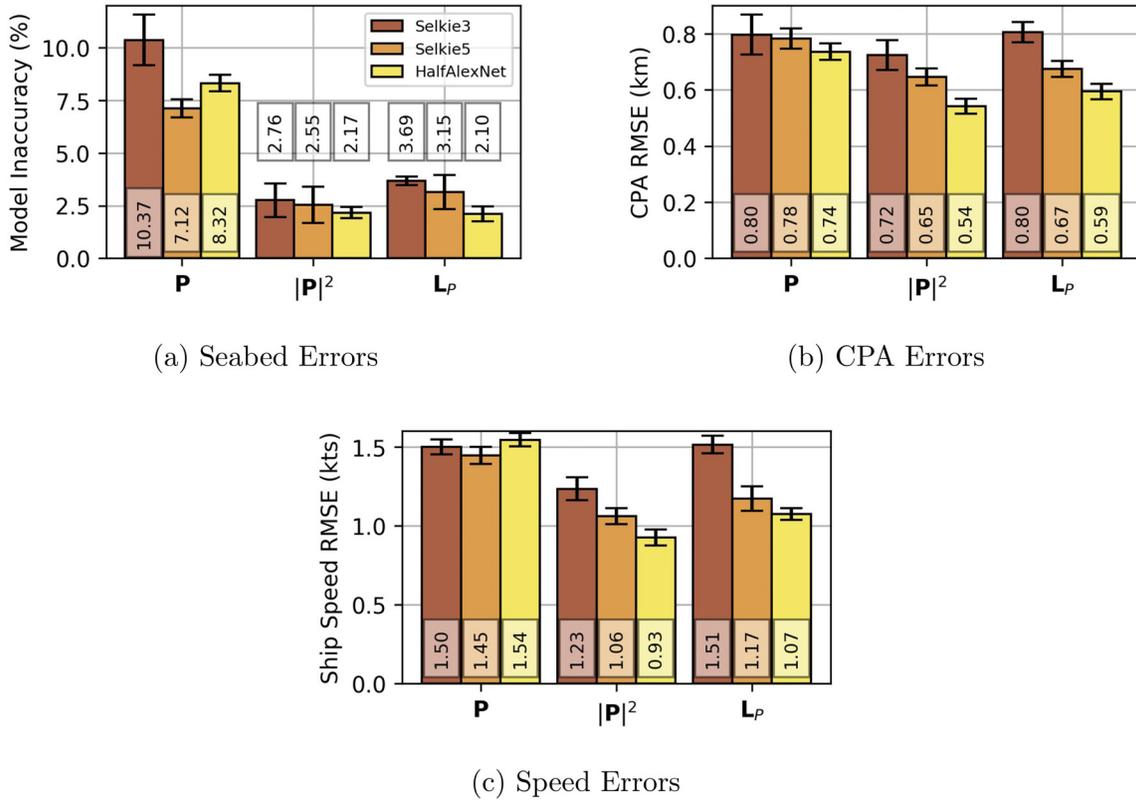


FIG. 6. (Color online) Results from ten instances of each network on validation dataset 2 containing 8640 samples, which had more speeds and CPAs near and beyond the boundary of those values in the training dataset (see Fig. 3). Results for (a) the seabed prediction inaccuracy, (b) the RMSE for the CPA range, and (c) the RMSE similar to Fig. 5.

trends can be attributed to the label normalization process mentioned in Sec. III A 1. Since the smallest speed in the training dataset, for example, was 16 kn, that value was normalized to zero while the largest speed of 24 kn was normalized to one. Any true values less than 16 would then have their normalized value as less than 0, which the network never needed to predict during training. The same holds for values above the maximum. This label normalization is beneficial for training and testing on datasets within the bounds of the training set but potentially becomes a detriment for datasets that contain samples beyond those bounds. For the purposes of this study, the label normalization suffices as the measured data, discussed next, falls within the bounds of the training dataset.

An interesting observation from both validation datasets is how the different data input methods produce different results. In general, the differences in results based on input type are larger than differences due to the network type. When using the complex \mathbf{P} matrix, the models have the lowest seabed accuracy across all networks. The prediction results when using $|\mathbf{P}|^2$ or \mathbf{L}_P are close for all network types, but $|\mathbf{P}|^2$ does slightly better than \mathbf{L}_P does on all of the predictions. This difference could be a result of the squaring of the $|\mathbf{P}|$ causing larger differences between features that help the networks learn. These validation tests emphasize the importance of the input data type and have shown the performance of the networks in the presence of slight ocean

depth mismatch and for source labels near and beyond the edges included in the training data.

B. CNN generalization on measured data

The generalizability of the network is tested by applying the trained networks to the measured spectrogram from the Kalamata (described in Sec. II). AIS data for the Kalamata during this time indicates an approximate CPA range of 3.3 km and speed of 19.9 kn. The expected seabed prediction is the mud-over-sand seabed as it was obtained from a maximum entropy optimization for the SBCEX 2017 area³² or the deep mud, which has similar surficial properties, including an angle of intromission (at which there is nearly complete transmission of acoustic energy from the water into the sediment). For the $|\mathbf{P}|^2$ and \mathbf{L}_P input types, the Kalamata spectrograms were obtained using time-averaging, whereas those for \mathbf{P} were not. The predictions of all the networks on the Kalamata sample are shown in Fig. 7.

The individual prediction distributions of each network for the CPA range and speed show how the smallest network has the most reliable predictions using \mathbf{P} . To represent the statistical distribution of results, Figs. 7(b) and 7(a) show violin plots (a combination of normalized distributions and box-and-whisker plots) of the CPA and speed predictions, respectively. The combination of network and input data

that most consistently predicts a CPA range and speed close to the AIS data (dashed lines) is Selkie3 trained on the \mathbf{P} input data; the tighter distribution for the speed and CPA than for the other networks indicates the robustness of the predictions to the random initializations of the network. Even though some of the medians of other networks are closer to the truth (the dashed horizontal line), the tighter distribution shows more confidence and repeatability from Selkie3.

That the best predictions come from the Selkie3 model on \mathbf{P} is surprising because of the slightly worse performance on the validation datasets (Sec. IV A). This result could be an example of overfitting due to the 14+ million parameters present in Selkie5 and HalfAlexNet. Although the larger networks perform the best in the validation results (Sec. IV A) drawn from the same statistical distribution as the training data, these networks do not generalize as well as Selkie3 does on measured data. To investigate this further, the learned weights of the linear layers were examined by determining if their absolute values were less than a target threshold. As the number of weights increased, not only did the number of these small weights increase, the percentage of small weights to total number of weights did as well. These findings serve as a caution against using the deepest networks presented in the literature as they might be too large for a specific problem. However, as the complexity of the problem increases (i.e., including further seabed types or

adding an additional source parameter to learn), those larger network sizes may provide better results.

The seabed type classification results are depicted differently in Fig. 7(c). Stacked barcharts of the predictions are used for the seabed class. The color and shading represent the different seabed classes. The height of each bar indicates the percentage of times (out of the ten trained networks) that each seabed had the highest classification output. The input data type is again listed on the horizontal axis, but the network type is now indicated with a number: (1) Selkie3, (2) Selkie5, and (3) HalfAlexNet.

The different results for seabed classification relate to the physics of ocean acoustics. First, the key propagation feature for the frequencies included (300–1500 Hz) is the angle of intromission (where there is nearly complete transmission of energy into the sediment) due to the speed of sound at the top of the seabed being less than the speed of sound of the water at the bottom of the ocean. With this in mind, the $|\mathbf{P}|^2$ and $|\mathbf{L}_P|$ predictions of a combination of deep mud and mud-over-sand are reasonable. In contrast, the complex \mathbf{P} inputs increase the difficulty in getting an accurate seabed prediction. The time-averaged $|\mathbf{P}|$ and $|\mathbf{P}|^2$ inputs reduce the uncorrelated noise in the spectrograms and, thus, allow the networks to identify a seabed with an angle of intromission 100% of the time for Selkie3 and HalfAlexNet and 80% of the time for Selkie5. Thus, for seabed type predictions on the Kalamata measured data, \mathbf{P} is

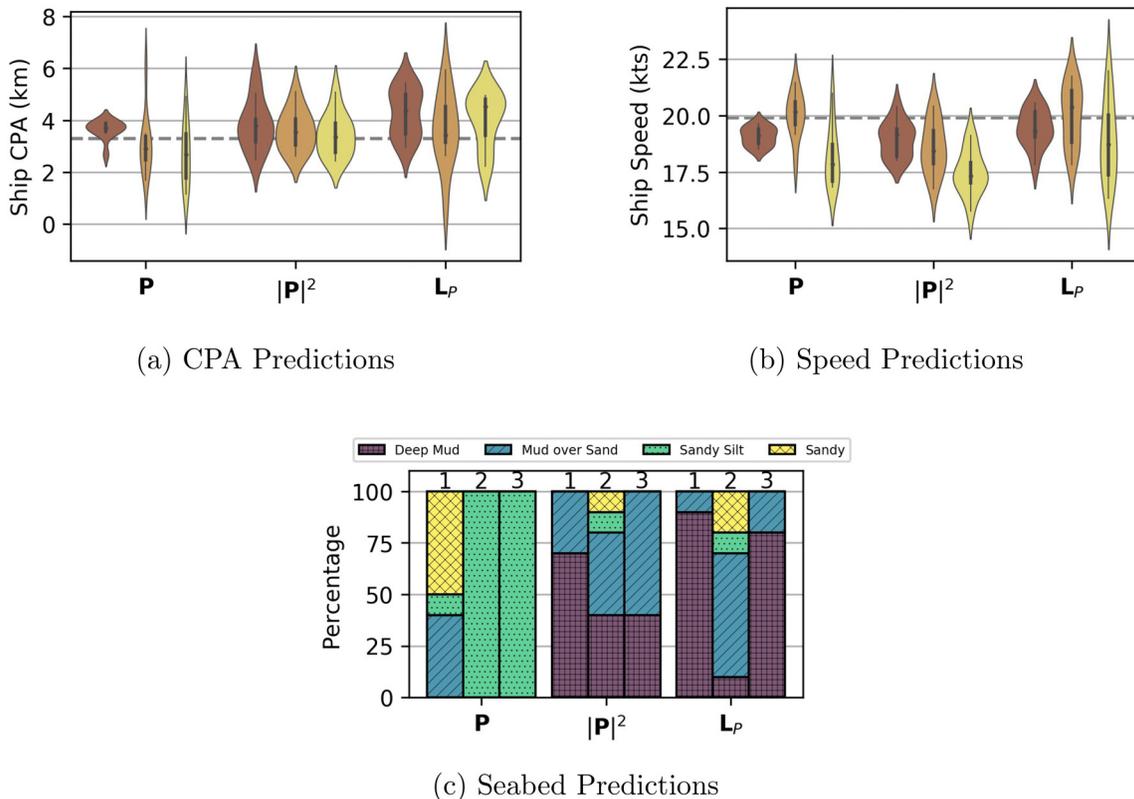


FIG. 7. (Color online) Results from ten instances of each network on the measured Kalamata spectrogram. Violin plots (a normalized probability distribution kernel with the median and quartile ranges over the ten training instances) of (a) the CPA range and (b) the speed predictions. (c) Stacked barchart showing the percentage of predictions for each seabed type. The input data type is listed on the horizontal axis. The three networks are distinguished by color in (a) and (b), similar to Figs. 5 and 6, and by numbers on top of the bars in (c) with 1 = Selkie3, 2 = Selkie5, and 3 = HalfAlexNet.

not a reasonable input type. Whereas further testing on more measured data is required, the improved performance with $|\mathbf{P}|^2$ and $|\mathbf{L}_P|$ make sense as the seabed type is often tied to bottom loss.

Because the true source level of the Kalamata was unknown (see Fig. 1), the simulated levels do not match the measured levels. To account for this difference, the data were normalized, which discarded information about range. One idea for future implementation is to include a source level parameter as a label for the CNN to learn. In addition, the measured spectrogram contains significantly more noise than the synthetic spectrograms, which increases the uncertainty in the predictions. The addition of random noise during training on the synthetic data could potentially teach the network to ignore such noise.

V. CONCLUSION

This study provides evidence that CNNs trained on synthetic SOO spectrograms have the potential to make predictions on the seabed type, ship speed, and CPA range. Specifically, 15 min SOO spectrograms spanning 300–1500 Hz were simulated with four seabed types, ten measured SSPs with different ocean depths, and a variety of source speeds and CPAs on a single hydrophone. The synthetic training dataset was used to train CNNs to make these predictions. The trained CNNs were then applied to additional synthetic testing datasets and a spectrogram from the ship Kalamata measured during the SBCEX 2017.

The results from the validation studies indicate that SOO spectrograms contain learnable features and patterns that CNNs can identify to make predictions. Predictions from three network architectures were each compared with three input types. To better quantify the network performance, ten training instances of each network are applied to the testing sets, which leads to a statistical representation of the results that indicates repeatability. On the validation cases, the seabed type accuracy was greater than 97% when the squared magnitudes of the complex pressure, $|\mathbf{P}|^2$, or levels, \mathbf{L}_P , were used. Inaccuracies in seabed classification for the validation tests were related to the similarities between a deep mud and mud-over-sand seabed or a sandy-silt and sandy seabed due to the frequency range used.

Although the spectrograms were individually normalized, the CPA range and speed prediction errors were small for the validation sets. When the validation data were tested on samples with slight water depth mismatch and CPA range and speed within the bounds included in the training data, the RMSE was less than 0.6 km and 1 kn, respectively. When the validation data were generated with source parameters at the edges and outside the training data, these errors increased to 0.8 km and 1.5 kn, showing that even the networks performed well on synthetic data near the edges of the training distribution regardless of the data input method.

The ability of the networks to generalize was investigated using a measured data sample. Although only data from a single ship, the Kalamata, were used, results from

ten instances of each network for each dataset type indicated that a smaller network generalizes better to the noisier measured data. The smallest network was able to provide good CPAs and speed predictions while also providing reasonable seabed predictions. However, the input data type impacted the predictions. The CPA range and speed predictions were most consistently near the expected values using the real and imaginary parts of the complex pressure, \mathbf{P} . However, using \mathbf{P} for seabed classification did not work well as the networks chose more reflective environments no matter the size of the network. When the input was time-averaged $|\mathbf{P}^2|$ or spectral levels in decibels, the networks consistently chose the environments with an angle of intromission.

The use of ship spectrograms in neural networks is a promising tool for ocean acoustics. This preliminary study needs to be followed by future work to expand the findings. First, introducing random noise representative of ambient or wind noise into the training data could help the network learn to ignore noise in the input data. Additional testing on the generalizability to measured data is needed. This study only used a single receiver at 33 m; hence, additional receivers on the array could also be used. The source level could be added as a learned label so the normalization would not be necessary. A proper way to systematically increase the variety of seabed types is necessary as well.

ACKNOWLEDGMENTS

This research was supported by the Office of Naval Research, Small Business Innovation Research (SBIR) Grant No. N68335-18-C-0806 and Contract No. N00014-19-C-2001. The SBCEX 2017 was funded by the U.S. Navy, Office of Naval Research. The authors also acknowledge and thank the captains and crew of the research vessel (RV) Endeavor.

¹S. Mao, E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, and G.-B. Huang, “An automatic identification system (AIS) database for maritime trajectory prediction and data mining,” in *Proceedings of ELM-2016* (Springer, New York, 2018), pp. 241–257.

²S. C. Wales and R. M. Heitmeyer, “An ensemble source spectra model for merchant ship-radiated noise,” *J. Acoust. Soc. Am.* **111**(3), 1211–1231 (2002).

³D. J. Battle, P. Gerstoft, W. A. Kuperman, W. S. Hodgkiss, and M. Siderius, “Geoacoustic inversion of tow-ship noise via near-field-matched-field processing,” *IEEE J. Oceanic Eng.* **28**(3), 454–467 (2003).

⁴M. Nicholas, J. S. Perkins, G. J. Orris, L. T. Fialkowski, and G. J. Heard, “Environmental inversion and matched-field tracking with a surface ship and an l-shaped receiver array,” *J. Acoust. Soc. Am.* **116**(5), 2891–2901 (2004).

⁵K. D. Heaney, “Rapid geoacoustic characterization using a surface ship of opportunity,” *IEEE J. Oceanic Eng.* **29**(1), 88–99 (2004).

⁶C. Park, W. Seong, and P. Gerstoft, “Geoacoustic inversion in time domain using ship of opportunity noise recorded on a horizontal towed array,” *J. Acoust. Soc. Am.* **117**(4), 1933–1941 (2005).

⁷R. A. Koch and D. P. Knobles, “Geoacoustic inversion with ships as sources,” *J. Acoust. Soc. Am.* **117**(2), 626–637 (2005).

⁸D. Tollefsen and S. E. Dosso, “Bayesian geoacoustic inversion of ship noise on a horizontal array,” *J. Acoust. Soc. Am.* **124**(2), 788–795 (2008).

⁹S. A. Stotts, R. A. Koch, S. M. Joshi, V. T. Nguyen, V. W. Ferreri, and D. P. Knobles, “Geoacoustic inversions of horizontal and vertical line array acoustic data from a surface ship source of opportunity,” *IEEE J. Oceanic Eng.* **35**(1), 79–102 (2010).

- ¹⁰C. Gervaise, B. G. Kinda, J. Bonnel, Y. Stéphan, and S. Vallez, "Passive geoacoustic inversion with a single hydrophone using broadband ship noise," *J. Acoust. Soc. Am.* **131**(3), 1999–2010 (2012).
- ¹¹S. E. Crocker, P. L. Nielsen, J. H. Miller, and M. Siderius, "Geoacoustic inversion of ship radiated noise in shallow water using data from a single hydrophone," *J. Acoust. Soc. Am.* **136**(5), EL362–EL368 (2014).
- ¹²S.-H. Byun, C. M. Verlinden, and K. G. Sabra, "Blind deconvolution of shipping sources in an ocean waveguide," *J. Acoust. Soc. Am.* **141**(2), 797–807 (2017).
- ¹³K. L. Gemba, J. Sarkar, B. Cornuelle, W. S. Hodgkiss, and W. Kuperman, "Estimating relative channel impulse responses from ships of opportunity in a shallow water environment," *J. Acoust. Soc. Am.* **144**(3), 1231–1244 (2018).
- ¹⁴L. Muzi, M. Siderius, and C. M. Verlinden, "Passive bottom reflection-loss estimation using ship noise and a vertical line array," *J. Acoust. Soc. Am.* **141**(6), 4372–4379 (2017).
- ¹⁵L. Xu, K. Yang, and Q. Yang, "Joint time-frequency inversion for seabed properties of ship noise on a vertical line array in south china sea," *IEEE Access* **6**, 62856–62864 (2018).
- ¹⁶D. Tollefsen and S. E. Dosso, "Three-dimensional source tracking in an uncertain environment," *J. Acoust. Soc. Am.* **125**(5), 2909–2917 (2009).
- ¹⁷D. Tollefsen, "Bayesian geoacoustic inversion and source tracking for horizontal line array data," *J. Acoust. Soc. Am.* **128**(1), 506 (2010).
- ¹⁸D. Tollefsen and S. E. Dosso, "Three-dimensional localization of multiple sources in an uncertain ocean environment," *Proc. Mtgs. Acoust.* **19**(1), 070071 (2013).
- ¹⁹D. Tollefsen, P. Gerstoft, and W. S. Hodgkiss, "Multiple-array passive acoustic source localization in shallow water," *J. Acoust. Soc. Am.* **141**(3), 1501–1513 (2017).
- ²⁰Z. 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**(5), 2922–2932 (2018).
- ²¹H. Niu, E. Ozanich, and P. Gerstoft, "Ship localization in santa barbara channel using machine learning classifiers," *J. Acoust. Soc. Am.* **142**(5), EL455–EL460 (2017).
- ²²H. Niu, E. Reeves, and P. Gerstoft, "Source localization in an ocean waveguide using supervised machine learning," *J. Acoust. Soc. Am.* **142**(3), 1176–1188 (2017).
- ²³H. Niu, Z. Gong, E. Ozanich, P. Gerstoft, H. Wang, and Z. Li, "Deep learning for ocean acoustic source localization using one sensor," *arXiv:1903.12319* (2019).
- ²⁴E. Ozanich, P. Gerstoft, and H. Niu, "A deep network for single-snapshot direction of arrival estimation," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)* (IEEE, New York, 2019), pp. 1–6.
- ²⁵E. Ozanich, P. Gerstoft, and H. Niu, "A feedforward neural network for direction-of-arrival estimation," *J. Acoust. Soc. Am.* **147**(3), 2035–2048 (2020).
- ²⁶E. L. Ferguson, R. Ramakrishnan, S. B. Williams, and C. T. Jin, "Convolutional neural networks for passive monitoring of a shallow water environment using a single sensor," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, New York, 2017), pp. 2657–2661.
- ²⁷W. Liu, Y. Yang, M. Xu, L. Lü, Z. Liu, and Y. Shi, "Source localization in the deep ocean using a convolutional neural network," *J. Acoust. Soc. Am.* **147**(4), EL314–EL319 (2020).
- ²⁸P. S. Wilson, D. P. Knobles, and T. B. Neilsen, "Guest editorial an overview of the seabed characterization experiment," *IEEE J. Oceanic Eng.* **45**(1), 1–13 (2020).
- ²⁹E. K. Westwood, C. T. Tindle, and N. R. Chapman, "A normal mode model for acousto-elastic ocean environments," *J. Acoust. Soc. Am.* **100**(6), 3631–3645 (1996).
- ³⁰D. F. Van Komen, T. B. Neilsen, K. Howarth, D. P. Knobles, and P. H. Dahl, "Seabed and range estimation of impulsive time series using a convolutional neural network," *J. Acoust. Soc. Am.* **147**(5), EL403–EL408 (2020).
- ³¹D. P. Knobles, R. A. Koch, L. A. Thompson, K. C. Focke, and P. E. Eisman, "Broadband sound propagation in shallow water and geoacoustic inversion," *J. Acoust. Soc. Am.* **113**(1), 205–222 (2003).
- ³²D. P. Knobles, P. S. Wilson, J. A. Goff, L. Wan, M. J. Buckingham, J. D. Chaytor, and M. Badiy, "Maximum entropy derived statistics of sound-speed structure in a fine-grained sediment inferred from sparse broadband acoustic measurements on the new england continental shelf," *IEEE J. Oceanic Eng.* **45**, 161–173 (2020).
- ³³G. R. Potty, J. H. Miller, and J. F. Lynch, "Inversion for sediment geoacoustic properties at the new england bight," *J. Acoust. Soc. Am.* **114**(4), 1874–1887 (2003).
- ³⁴J.-X. Zhou, X.-Z. Zhang, and D. P. Knobles, "Low-frequency geoacoustic model for the effective properties of sandy seabottoms," *J. Acoust. Soc. Am.* **125**(5), 2847–2866 (2009).
- ³⁵C. M. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, Oxford, UK, 1995).
- ³⁶A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (2012), pp. 1097–1105.
- ³⁷J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, New York, 2009), pp. 248–255.
- ³⁸A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems* (2019), pp. 8024–8035.
- ³⁹D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980* (2014).
- ⁴⁰D. F. Van Komen, T. B. Neilsen, D. P. Knobles, and M. Badiy, "A convolutional neural network for source range and ocean seabed classification using pressure time-series," *Proc. Mtgs. Acoust.* **36**(1), 070004 (2019).
- ⁴¹I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv:1608.03983* (2016).
- ⁴²T. B. Neilsen, C. D. Escobar, M. C. Acree, W. S. Hodgkiss, D. F. Van Komen, D. P. Knobles, M. Badiy, and J. Castro, "Learning source location and seabed type from towed mid-frequency tonals on a vertical line array," *J. Acoust. Soc. Am.* **149**(1), 692–705 (2021).
- ⁴³A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 7482–7491.
- ⁴⁴This suggestion was given by Tony-Y on the <https://discuss.pytorch.org/t/how-to-learn-the-weights-between-two-losses/39681/12> PyTorch online forums (Last viewed May 19, 2020).
- ⁴⁵R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *IJCAI*, Montreal, Canada (1995), Vol. 14, pp. 1137–1145.