Model-based convolutional neural network approach to underwater source-range estimation

Citation: . Chen, H. Schmidt; Model-based convolutional neural network approach to underwater source-range estimation. J. Acoust. Soc. Am. 1 January 2021; 149 (1): 405–420.

As Published: 10.1121/10.0003329

Publisher: Acoustical Society of America

Persistent URL: https://hdl.handle.net/1721.1/154268

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

JANUARY 15 2021

Model-based convolutional neural network approach to underwater source-range estimation

Special Collection: Machine Learning in Acoustics

R. Chen ; H. Schmidt

J. Acoust. Soc. Am. 149, 405–420 (2021)
https://doi.org/10.1121/10.0003329

JASA ARTICLE

Model-based convolutional neural network approach to underwater source-range estimation

R. Chen[a] and H. Schmidt
Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

ABSTRACT:
This paper is part of a special issue on machine learning in acoustics. A model-based convolutional neural network (CNN) approach is presented to test the viability of this method as an alternative to conventional matched-field processing (MFP) for underwater source-range estimation. The networks are trained with simulated data generated under a particular model of the environment. When tested with data simulated in environments that deviate slightly from the training environment, this approach shows improved prediction accuracy and lower mean-absolute-error (MAE) compared to MFP. The performance of this model-based approach also transfers to real data, as demonstrated separately with field data collected in the Beaufort Sea and off the coast of Southern California. For the former, the CNN predictions are consistent with expected source range while for the latter, the CNN estimates have lower MAE compared to MFP. Examination of the trained CNNs' intermediate outputs suggests that the approach is more constrained than MFP from outputting very inaccurate predictions when there is a slight environmental mismatch. This improvement appears to be at the expense of decreased certainty in the correct source range prediction when the environment is precisely modeled.
https://doi.org/10.1121/10.0003329

(Received 10 August 2020; revised 12 December 2020; accepted 15 December 2020; published online 15 January 2021)
[Editor: Zoi-Heleni Michalopoulou] Pages: 405–420

I. INTRODUCTION

Acoustic source localization is pertinent to many fields such as ship tracking, marine mammal monitoring, and underwater vehicle operations. Approaches to this task can be broadly divided into two categories: model-based and data-driven. Model-based methods largely make use of matched-field processing (MFP) and its variants,[1–13] which employ acoustical modeling to simulate the expected propagation environment. Consequently, the performance of such methods can be sensitive to model mismatch. Alternatively, growth in computational and data management capabilities have promoted interest in data-driven approaches; particularly, machine learning (ML) methods that learn propagation features directly from collected data without the need for any environmental modeling. Studies have demonstrated the capability of data-driven ML methods to perform on par or better than MFP when given adequate training data for source localization under a variety of environments.[14–17] However, drawbacks exist as well; data-driven techniques are often limited by the impracticality of collecting enough acoustic data over a sampled space of source locations in order to build their required training dataset, making them potentially costly to implement due to increased ship time and experiment logistics. Taking advantage of the performance of ML methods and the ease of the model-based approach to simulate training data, other works have examined model-based ML as another alternative.[18–23] This mixed approach shows promise as the ML methods demonstrate comparable or improved performance to conventional methods. Thus, the model-based ML approach may offer a good compromise between performance and ease of data generation. However, some research questions still being investigated regarding this approach include:

(1) How does the performance of model-based ML methods compare to MFP when tested on simulated environments outside the bounds of the originally modeled parameters? Do they suffer the same environmental robustness issue as MFP?
(2) Does the performance of model-based ML methods transfer to real data collected in the field? How does their performance compare to MFP in this case?
(3) If model-based ML methods show improvement over MFP for questions 1 and 2, how may they be achieving their better performance?

This paper takes an initial step to address these questions. We propose a model-based convolutional neural network (CNN) approach to source-range estimation and test its performance against MFP in two separate environments with different types of mismatch. In Sec. II, the two propagation environments are introduced. The first is an Arctic case modeled after the Beaufort Sea in which robustness to the water column sound speed profile (SSP) is tested. The second is a shallow waveguide modeled after a region off the coast of Point Loma, CA, in which robustness to sea bottom depth is examined. These separate test cases are chosen because of the availability of field data collected there during past experiments, which are used to verify the accuracy and utility of our CNN approach. We also describe our

[a] Electronic mail: aruic@mit.edu, ORCID: 0000-0001-7790-2553.

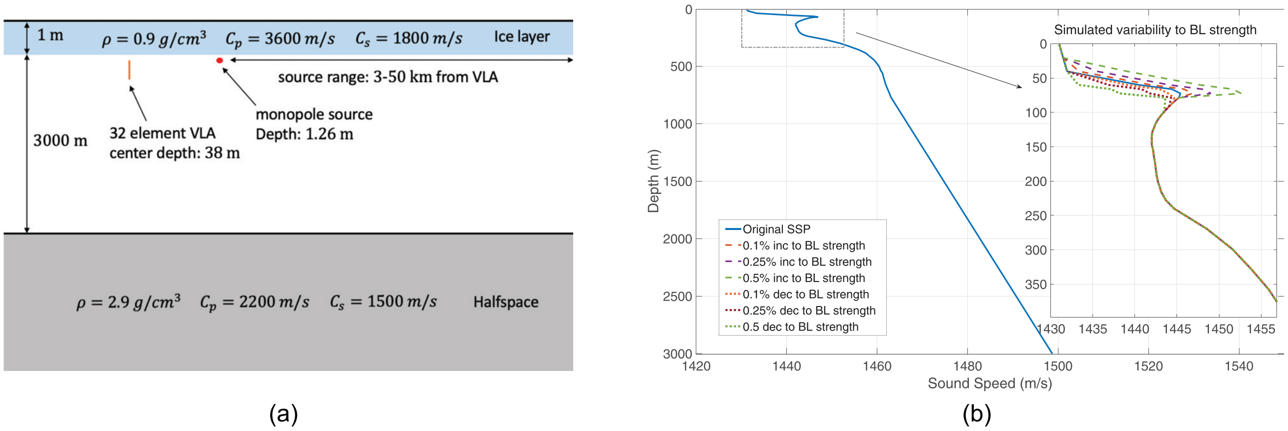


Fig. 1. (Color online) (a) Simulation setup for generating training and testing datasets in the Beaufort Sea environment. (b) Original, measured ICEX16 SSP used to generate training dataset (solid line); SSPs with deviations to the BL strength used to generate testing datasets (dashed and dotted lines).

methods for generating training and testing datasets in simulation in this section. In Sec. III, we provide an overview of our proposed CNN architecture and training process. We also show performance comparisons between CNN and MFP on simulated data. For both environments, the performances of the model-based CNN and MFP on real data collected in the regions are compared in Sec. IV to see if they agree with results from the simulated data tests. Then, in Sec. V, we further investigate how our CNN approach may be achieving its performance by taking a closer look at the networks' intermediate outputs.

II. PROPAGATION ENVIRONMENTS

A. Beaufort Sea

The first propagation environment we present in this study is based off of the Beaufort Sea region of the Arctic Ocean during the U.S. Navy's 2016 ICE Exercise (ICEX16). Specifically, the model consists of a 1 m surface ice layer, a 3000 m water column, and a solid bottom halfspace [Fig. 1(a)]. The parameters for these layers are summarized in Table I. All environmental simulations in this study are done with OASES.[24]

The SSP simulated in the water column is measured during ICEX16 [Fig. 1(b), solid-blue] and contains a local maximum at ∼70 m depth caused by the Beaufort Lens (BL)—a layer of warm Pacific water neutrally buoyant at that depth.[25] This feature creates a double duct propagation environment and its strength, as defined by the difference between the local SSP maximum and the local minimum below, is a dominant factor affecting underwater acoustics in the region.[26–29] Furthermore, the Beaufort Sea SSP is continuously varying, with most of the variability focused near the local SSP maximum.[28] As a result, any viable model-based ML approach to source-range estimation in this region must show robustness to some SSP mismatch. With this in mind, we generate a training dataset for our CNN approach using the originally measured SSP during ICEX16. We then generate several testing datasets with deviations to the original SSP BL strength [Fig. 1(b), inset] to measure our method's robustness to SSP mismatch. To generate the training dataset, an 850 Hz monopole source is placed 0.26 m below the ice cover and moved from 3 to 50 km from a recording vertical line array (VLA) at 10 m increments. A discrete, near-surface source is deployed because our goal is to use our approach to estimate the range of ice cover generated ambient noise, which has been shown

TABLE I. Parameters for simulated ICEX16 and SWellEx-96 environments.

	ICEX16	SWellEx-96
Ice Layer	$C_p = 3600$ m/s, $C_s = 1800$ m/s, $\rho = 0.9$ g/cm^3, thickness = 1 m, Root-mean-square roughness = 0.2 m Roughness correlation length = 20 m	N/A
Water Column	Thickness = 3000 m ICEX16 SSP with varying BL strength	Thickness = 213.5 − 219.5 m SWellEx-96 SSP
Bottom Layer 1	$C_p = 2200$ m/s, $C_s = 1500$ m/s, $\rho = 2.9$ g/cm^3, halfspace	$C_{top} = 1572.3$ m/s, $C_{bot} = 1593$ m/s $\rho = 1.76$ g/cm^3, thickness = 23.5 m
Bottom Layer 2	N/A	$C_{top} = 1881$ m/s, $C_{bot} = 3245$ m/s $\rho = 2.06$ g/cm^3, thickness = 800 m
Bottom Layer 3	N/A	$C_p = 5200$ m/s, $\rho = 2.66$ g/cm^3 Halfspace

to be well modeled by a discrete source in the recent Arctic environment with younger and thinner ice cover.[29] The source frequency and range interval are chosen based on those of real noise data collected during ICEX16. The simulated VLA consists of 32 elements with nested spacing (1.5 m for the outer ten elements, 0.75 m for the inner 22 elements). The normalized sample covariance matrix (SCM) of the acoustics signal recorded on the array at each source range increment is measured and makes up the training dataset. These matrices are of size 32 × 32 × 2, with the third dimension containing the real and imaginary parts of the SCM. Normalization of the matrices[14,15] is performed to reduce the effect of acoustic amplitude so that our approach may be used regardless of the simulated source amplitude. The testing datasets are generated in a similar manner, but each with a different amount of BL strength deviation to the original SSP. One thousand SCM samples are generated for each testing dataset by placing the source at random ranges within the training interval of 3–50 km.

B. Point Loma, CA

A second propagation environment included in this study is modeled after the site of the SWellEx-96 experiment[30] off the coast of Point Loma, CA. The environment consists of a 216.5 m water column atop of three solid bottom layers with increasing density [Fig. 2(a), Table I]. The SSP used in simulated environment is the average of profiles collected during the experiment. The simulated array matches the VLA deployed during SWellEx-96 and contains 21 elements between 94.125 and 212.25 m depth [Fig. 2(b)].

At this site, the main environmental variability results from the depth of the ocean bottom as the bathymetry around the VLA varies from ∼150 to 270 m. Thus, we use this environment to test the robustness of our CNN approach to ocean bottom depth mismatch. To generate the training dataset, the ocean bottom is set at 216.5 m depth. A 109 Hz monopole source is placed 9 m below the ocean surface and moved from 0 to 10 km away from the VLA at 10 m increments [Fig. 2(a)]. The SCM recorded on the VLA at each source location, size 21 × 21 × 2 in this case, makes up the dataset. As with the ICEX16 environment, the source frequency, depth, and range interval are chosen based on the experimental setup and real data collected during the field experiment. Four testing datasets are generated with ocean bottom depth set to 213.5, 215.5, 217.5, and 219.5 m, respectively. For each, the source is placed at 500 random ranges within the training interval and the SCM recorded on the VLA is simulated.

III. MFP AND CNN APPROACHES

A. MFP approach

With MFP, localization prediction is made by comparing the input data SCM against a template of replica vectors modeled with the source at various locations, **a**. At a particular frequency, the optimal estimate is calculated as

$$\underset{\mathbf{a}}{\operatorname{argmax}}\left[B(\mathbf{a}) = \mathbf{w}^H(\mathbf{a})\mathbf{P}\mathbf{w}(\mathbf{a})\right], \qquad (1)$$

where **P** is the SCM of the input data and **w**(**a**) is the replica vector for location **a**. The MFP replica vector templates are generated for the two environments presented in Sec. II in the same manner the training SCM datasets are generated for the CNN approach. In other words, the information content of the MFP template vectors is equivalent to the training datasets for the CNN approach.

B. CNN approach

Source range estimation is performed using a CNN approach in which spatial filters extract features from input SCMs and learn a relationship between those features and their corresponding labeled source ranges. In this sense, this approach models a mapping between the training data and the source range outputs and uses that model to make

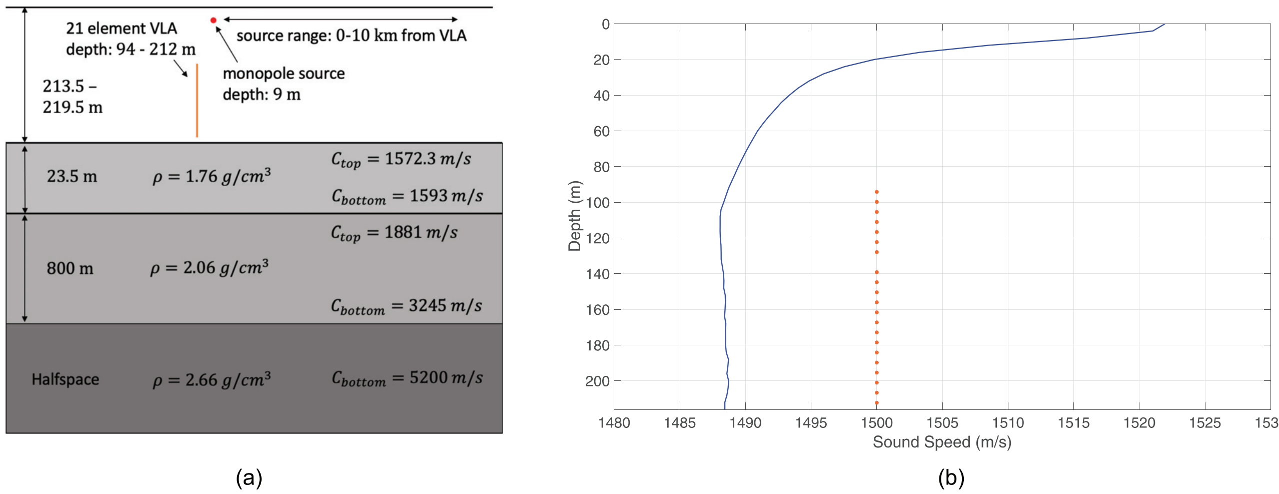


Fig. 2. (Color online) (a) Simulation setup for generating training and testing datasets in the SWellEx-96 environment. To generate the training dataset, ocean bottom depth is set at 216.5 m. To generate the testing datasets, ocean bottom depth is varied between 213.5 and 219.5 m at 2 m increments. (b) SSP used to generate training and testing datasets and VLA element locations.

predictions for new, testing data. This differs from MFP, where testing data is directly compared with the training data (replica vectors). The complexity of the CNN model is governed by its number of parameters, decided, in part, by the number of layers and filters. The larger the number of parameters, the more exact the mapping between the training data and the source-range outputs may be. However, an exact mapping to the outputs for the training dataset does not guarantee good performance on unseen testing data. Thus, we must regularize the CNN approach so that over-fitting on the training data does not occur and the network generalizes well to new data. We decided to use a CNN approach because the inputs to our problem are matrices and CNNs specialize in ML problems where the input is not a one-dimensional (1-D) vector, such as image classification. A detailed overview of CNNs is presented in a review by Bianco *et al*.[31] Two separate CNNs are trained for each of the two environments presented in this study. One takes a classification approach (CNN-c), while the other takes a regression approach (CNN-r). For the ICEX16 environment, the categorical training labels for CNN-c are created by rounding the source ranges in the training interval to the nearest 0.5 km, thus creating 95 classes between 3 and 50 km. For the SWellEx-96 environment, source ranges are rounded to the nearest 0.1 km due to the smaller training interval to create 101 classes between 0 and 10 km. For both environments, the training labels for CNN-r are kept as the training increment values, as the regression approach outputs predictions in continuous space.

C. Network architecture and training

The CNN network architectures designed for both environments and both the classification and regression approaches are largely similar. All networks contain three convolutional layers with 16, 128, and 256 scaled exponential linear unit (SELU) activated filters in each layer, respectively (a description of all activation functions used in this study can be found in the Appendix). These are followed by a fully connected layer of 256 sigmoid activated nodes, and then an output layer. The number of layers and number of filters in each layer are selected after some initial testing on the performance of various CNN architectures with more and less layers and filters. For CNN-c, the activation in the output layer is the softmax function, while for CNN-r, the activation in the output layer is the linear function. Batch normalization regularization is performed after each convolutional layer while dropout regularization is performed after each convolutional (drop rate $= 0.5$) and fully connected layer (drop rate $= 0.25$). These regularization layers help to prevent the CNNs from over-fitting during training.[32,33] The major difference between the CNN architectures of the two environments is the size of the filters used in the convolutional layers. For ICEX16, the sizes are 3×3, 5×5, and 7×7, respectively, for the three layers. All filters in each layer are applied with a stride size of 2 to condense information from one layer to the next. For the SWellEx-96 environment, the same stride size of 2 is applied in each layer. However, because of the smaller SCM input dimensions compared to the ICEX16 case, the filter sizes for the layers are set to 3×3, 5×5, and 5×5, respectively, for the three convolutional layers. We decided to increase filter size with network depth because our initial testing showed that smaller filters in the first layer and larger filters in the deeper layers performed better than the reverse. We suspect that this is because the smaller filters capture more detail in the SCMs that gets passed on to the later layers while the larger filters in the later layers help more with condensing information to pass onto the fully connected layer. A schematic of the CNN architectures is shown in Fig. 3 (right).

The CNNs are implemented and trained using the Keras and Tensorflow libraries.[34] The training dataset is randomly segmented into an 80/20 split, where 80% of the data is used for training and 20% is used for validation. The categorical cross-entropy cost function is used for classification and mean-squared-error (MSE) is used for regression. The Adam optimizer[35] is used with a batch size of 128 and an initial learning rate of $\gamma = 0.0001$. γ subsequently decreases by 90% if the validation cost does not decrease for 75 epochs. Training stops if the validation cost does not decrease for 125 epochs to help prevent over-training.

To further optimize the CNNs' architectures, network pruning[36] is done to strip away under-activated filters as determined by the filters' $L1$ norms. This process is as follows:

(1) The original, full network is trained until stoppage.
(2) The $L1$ norms of the filter weights in each convolutional layer of the trained network are plotted (example shown in Fig. 3, left); all filters whose $L1$ norm is much smaller compared to the largest $L1$ norm value are deleted from the network.
(3) Training is continued on the updated, smaller network to re-adjust the weights of the kept filters, until stoppage again. The initial training rate is set as the same as when training last stopped.
(4) Steps 2 and 3 are repeated until the validation accuracy of the reduced network decreases from that of the original, full size network.

These pruning steps reduce the trained networks' complexity by decreasing their number of parameters, making the final models more lightweight. For both environments, the CNN-c networks are reduced much more than the CNN-r networks. This may be attributed to regression being a more difficult task than classification. For the ICEX16 environment, the final number of filters in each convolutional layer of the CNN-c model is 12, 24, and 46, respectively, and that for the CNN-r model is 12, 108, and 206, respectively. For the SWellEx-96 environment, the final number of filters in each convolutional layer of the CNN-c model is 6, 38, and 40, respectively, and that for the CNN-r model remained the same as the full network, as any reduction caused a decrease in performance. The details of the pruned networks are shown in Table II.

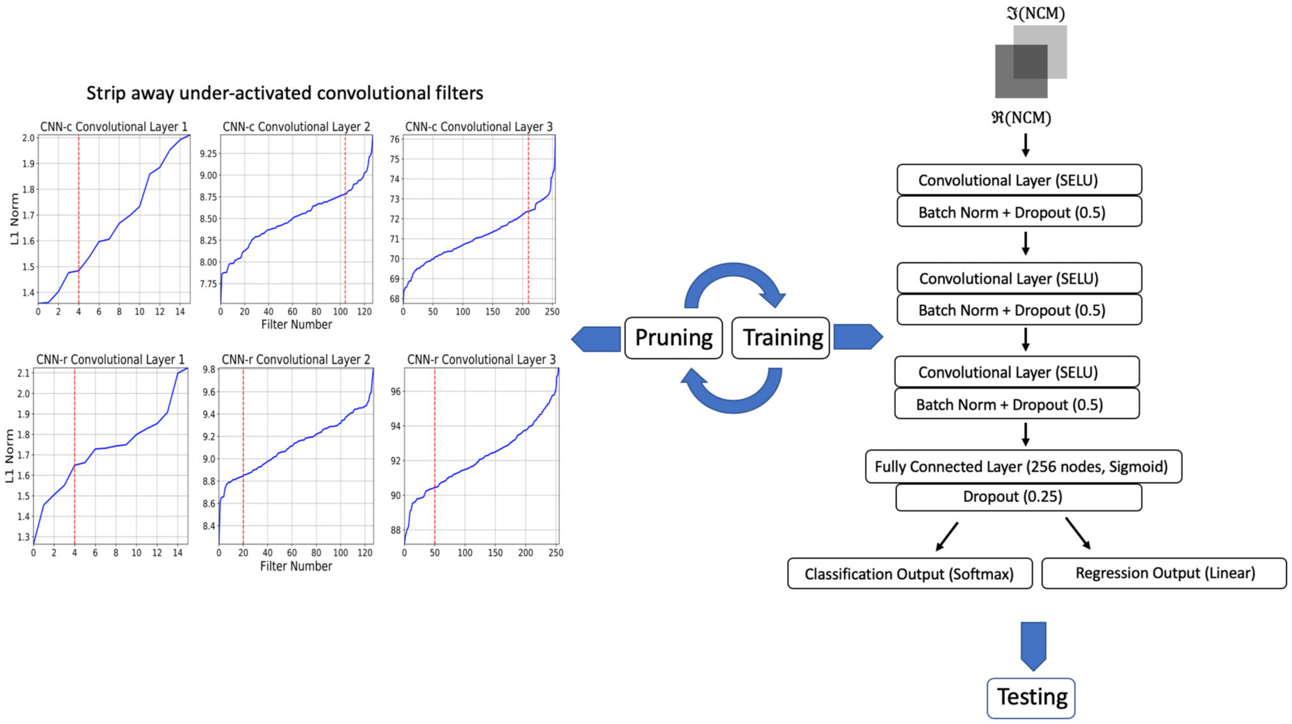

Strip away under-activated convolutional filters
CNN-c Convolutional Layer 1
CNN-c Convolutional Layer 2
CNN-c Convolutional Layer 3
L1 Norm
Filter Number
CNN-r Convolutional Layer 1
CNN-r Convolutional Layer 2
CNN-r Convolutional Layer 3
L1 Norm
Filter Number
Pruning
Training
ℑ(NCM)
ℜ(NCM)
Convolutional Layer (SELU)
Batch Norm + Dropout (0.5)
Convolutional Layer (SELU)
Batch Norm + Dropout (0.5)
Convolutional Layer (SELU)
Batch Norm + Dropout (0.5)
Fully Connected Layer (256 nodes, Sigmoid)
Dropout (0.25)
Classification Output (Softmax)
Regression Output (Linear)
Testing

Fig. 3. (Color online) (Right) Architecture of CNNs trained in this study. CNN-r and CNN-c differ only in the output layer. The number of filters in each convolutional layer depends on the pruning process. The size of convolutional filters depends on the simulated environment. (Left) $L1$ norms of CNN-c and CNN-r convolutional layer filters for the ICEX16 environment. Filters to the left of the vertical line on each plot are deleted in the final reduced networks after successive rounds of pruning.

D. Performance on simulated testing data

1. Beaufort Sea environment

As noted previously, the training dataset for the Beaufort Sea environment is generated by simulating a source at various ranges under the measured SSP during ICEX16, as shown in Fig. 1(b). To test the robustness of our CNN approaches compared to conventional MFP, we generate testing datasets—each with 1000 test samples with the surface source placed at a random range within the training interval—with varying deviation to the original ICEX16 SSP.

Two performance metrics are compared in this section. One is the percentage of testing predictions that are within 1 km of the actual source range. This metric reflects how accurate each individual prediction is to the corresponding correct value. The other metric is MAE and reflects the averaged error over all predictions for a testing dataset. This metric is formally defined as

$$MAE = \frac{1}{N}\sum_{i}^{N} |Prediction[i] - Actual[i]|. \qquad (2)$$

TABLE II. CNN architectures after iterative pruning.

CNN Type	ICEX16 CNN-c	ICEX16 CNN-r	SWellEx-96 CNN-c	SWellEx-96 CNN-r
# of Conv. Filters	12; 24; 46	12; 108; 206	6; 38; 40	16; 128; 256
# of Parameters	275 009	1 968 687	162 433	1 463 025

The performance of our trained CNNs on the testing datasets is compared to MFP in Figs. 4 and 5. As expected, as the magnitude of BL strength deviation (SSP mismatch) increases in the generation of the testing datasets, the performance of all three methods decreases by both metrics. However, the CNN-c and CNN-r approaches show improved performance over MFP with SSP mismatch while performing similarly to MFP with no mismatch (Fig. 5). The panels within Fig. 4 further reveal that there is high variability in the MFP predictions. This means that close-to-correct predictions can often be extremely accurate while for incorrect predictions, the margin of the mistakes can be quite large. In contrast, the CNN methods, particularly CNN-r, show lower variability in their predictions. As a result, although the accuracy of any individual CNN prediction may not be as high as the corresponding MFP prediction, the overall predictions are more consistent with ground truth. Thus, the CNN approaches appear to gain robustness to environmental mismatch over MFP by trading off individual data-point prediction accuracy for overall prediction consistency. Furthermore, the reason that CNN-r is the more prominent example of this trade-off over CNN-c is likely because of the difference in the cost function used during their training. For CNN-r, training focuses on minimizing the MSE loss, which inherently leads to more consistent predictions and lower overall error than predictions derived from categorical association, as is the case with CNN-c. Further discussion on how the CNN approaches achieve their robustness to environmental mismatch is presented in Sec. V.

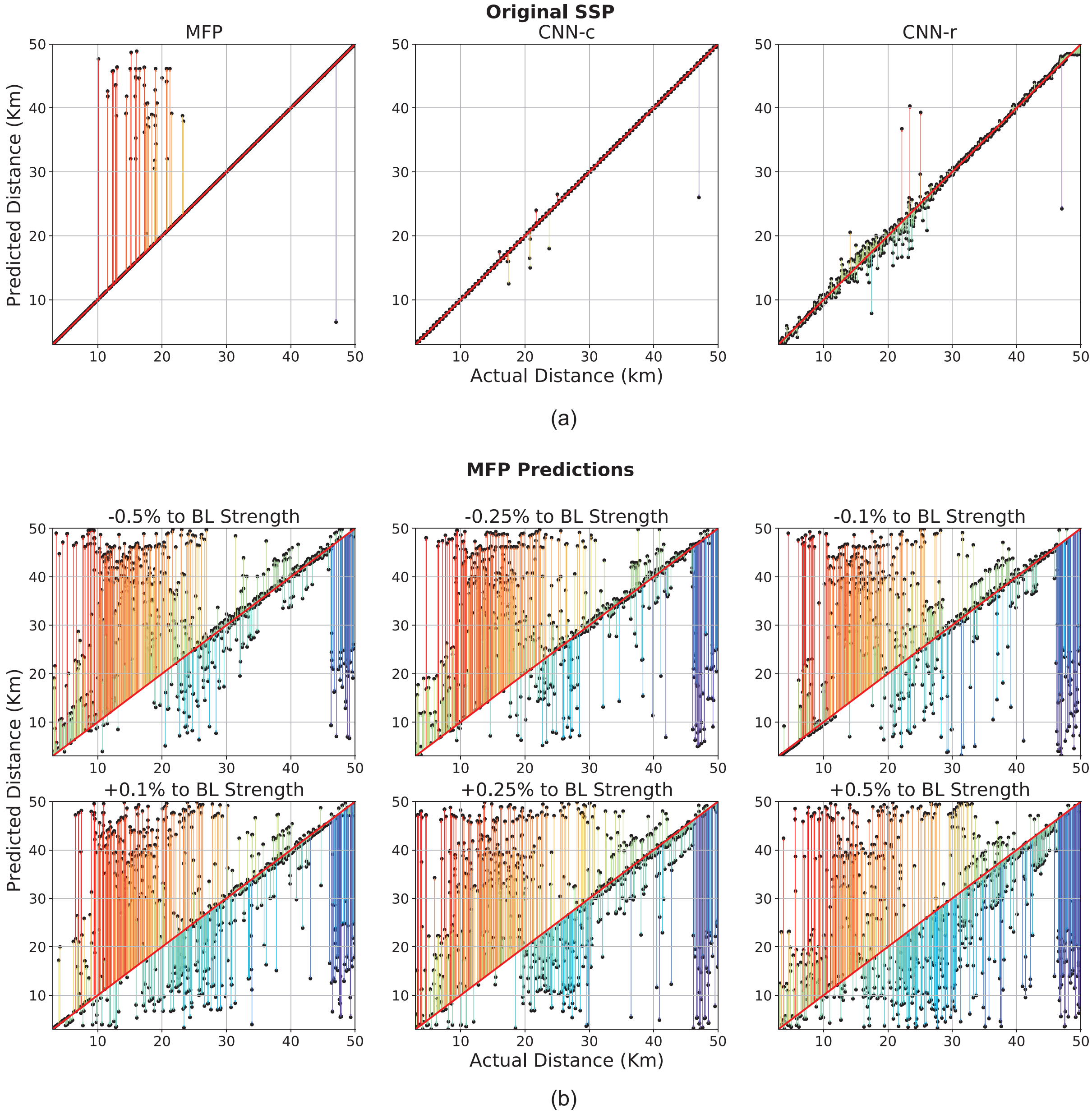


Fig. 4. (Color online) Diagonal solid line represents ground truth. Dots show prediction values. Quivers represent the difference between each prediction and the corresponding ground truth value. (a) Performance of MFP, CNN-c, and CNN-r on testing data generated using original ICEX16 SSP. (b)–(d) Performance of each approach on testing data generated with deviations to the original ICEX16 SSP.

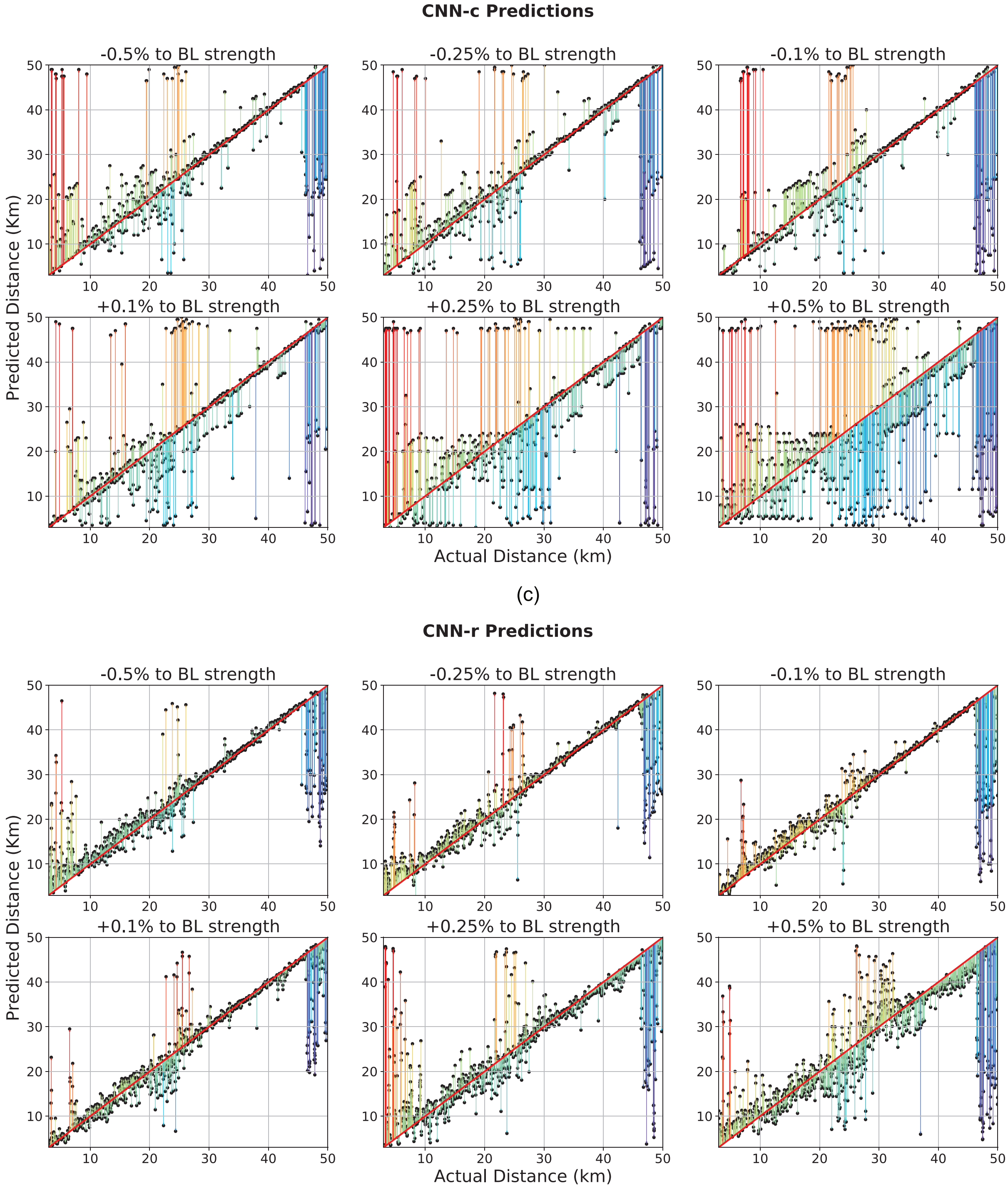
2. SWellEx-96 environment

With the SWellEx-96 environment, our CNN approaches' robustness to bottom depth mismatch is compared with conventional MFP. The environment with a bottom depth of 216.5 m is used to generate the training dataset for the CNN methods and the replica vector template for MFP. The testing datasets are created with bottom depths of 213.5, 215.5, 216.5, 217.5, and 219.5 m. The performance comparison, based on the two metrics introduced previously, is shown in Figs. 6 and 7. These plots show a similar trend in the methods' performance compared to the ICEX16 environment. However, in this case, the CNN-c approach shows a clear improvement in performance compared to the other two methods by both metrics for all bottom depths. Comparing CNN-r to MFP, the results again demonstrate that, similar to the ICEX16 simulated tests, CNN-r trades off accuracy of individual predictions for lower overall error. However, for this environment, this trade-off may be overdone. While the MAE of CNN-r is lower than MFP for all bottom depths, Fig. 6 shows that CNN-r typically has a larger error margin than MFP when comparing individual predictions, especially as the bottom depth mismatch increases from 216.5 m. Thus, for this environment, the CNN-r approach may not be a more robust alternative to MFP while CNN-c shows promise.

IV. PERFORMANCE ON REAL DATA

Field experiment data collected from the two environments presented in this study are used to validate whether

https://doi.org/10.1121/10.0003329



(c)

(d)

Fig. 4. (Color online) (*Continued*).

the performance improvements of our CNN approaches over MFP on simulated data transfer to real data. As noted, the goal of this section is to examine whether our model-based CNN approaches can still be a more robust alternative to conventional MFP when used on real data. Accordingly, the CNNs used to process the real data are exactly the ones trained with simulated data, as presented in Sec. III.

A. ICEX16

As described in Sec. II, ICEX16 was a U.S. Navy exercise and research expedition conducted in the Beaufort Sea in March, 2016. As part of this effort, ∼8 h of ambient noise data were collected on March 13th (UTC) to record the under-ice soundscape using a 32 element VLA with a sampling frequency of 12 000 Hz. The simulated environment used to generate the CNN training datasets is modeled after

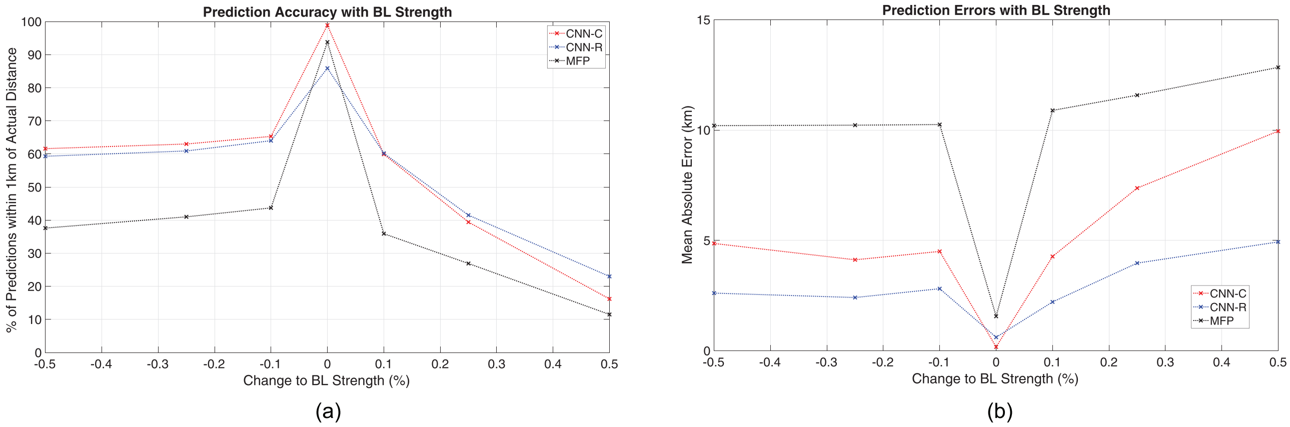


Fig. 5. (Color online) Performance metric comparison between MPF, CNN-c, and CNN-r with varying amounts of deviation to the original ICEX16 SSP. By both metrics, CNN approaches show similar performance to MFP without any SSP mismatch but improved performance with SSP mismatch.

this experimental setup [Fig. 1(a)]. To prepare the testing dataset, the collected data is segmented into ten 240-point Hanning windows with 50% overlap. The SCM averaged over 800–900 Hz is then calculated for each data snapshot window using the chirp-z transform with a fast Fourier transform (FFT) size of 512. Following this, the SCMs over every 32 snapshot window are averaged so that the resultant covariance matrices for MFP are not singular. Although not shown in the results, longer snapshot averages are tried as well and the prediction outputs for all approaches are consistent across all averaging lengths tested. The ICEX16 dataset was collected overnight under quiet camp noise conditions so the major contributor of ambient noise is from the ice cover. A previous analysis[29] of this data suggest that ambient noise recorded during this time is predominantly and consistently generated by a surface ice ridge ∼27–34 km from the VLA. Ice temperature satellite imagery from the National Snow and Ice Data Center[37] captured on the day of data recording further confirms this result. As shown in Fig. 8(a), an ice pressure ridge was present to the northeast of the ICEX16 camp site (VLA location) with the most prominent portion of the ridge ∼34 km away.

Figure 8(b) presents the source range estimations of MFP, CNN-c, and CNN-r on the ICEX16 ambient noise test dataset. The dots show the individual predictions for each snapshot window while the solid line represents the 10-min moving average of the predictions. From the top plot, MFP predictions deviate up to ∼20 km from the range of the ice ridge formation, where the ambient noise was likely generated. The prediction outputs of CNN-c and CNN-r are more consistent with the range of the ice ridge. This is especially true for CNN-r, whose predictions show very little variability and largely remain between ∼25 and 40 km.

B. SWellEx-96

Acoustic recording by the 21 element VLA (sampling frequency = 1500 Hz) from the S5 event of the SWellEx-96 experiment is used to test the performance of our trained CNNs on real data from this environment. Again, the CNNs are trained with simulated data generated under the environment shown in Fig. 2. As part of the S5 event, a 9 m deep source emitting at 109 Hz was towed by a ship along the blue track shown in Fig. 9(a). While the recording VLA was deployed at a location with ocean bottom depth of 216.5 m, the bathymetry along the source track varied mostly between ∼180 and 220 m depth. Thus, there is mismatch between the simulated training environment and the real testing environment. Similar to the ICEX16 dataset, the S5 event dataset is segmented into Hanning windows, in this case of size 512, with 50% overlap. The SCM averaged over 108.5–109.5 Hz is then calculated for each data snapshot window using the chirp-z transform with an FFT size of 12. Following this, the SCMs over every 25 snapshot window are averaged to prevent singular covariance matrices. The ground truth range of the towed source with time is calculated from recorded GPS coordinates of the VLA and the source during the experiment; this is shown as the solid line in Fig. 9(b).

Figure 9(b) also shows the range predictions by MFP, CNN-c, and CNN-r. From this plot, it appears that MFP and CNN-c have similar performances, while CNN-r does worse in comparison. Table III shows that based on the performance metrics introduced in Sec. III, CNN-c has the highest percentage of predictions within 1 km of ground truth with 70.7%, followed by MFP at 57.5%, and last, CNN-r with 37.8%. However, comparing their MAEs, CNN-r, and CNN-c have similar performances by this metric, with 1.4 and 1.41 km, respectively. Both out-perform MFP, which has a MAE of 1.73 km. These results match the observations from the simulated testing cases, where CNN-c has the best performance of the three methods and CNN-r had lower MAE than MFP but lacked accuracy in individual predictions. This similarity again demonstrates that the performance of our model-based CNN approaches does indeed transfer to real data. Here, as with the simulated testing cases, the CNN-r approach appears to over-compensate for lowering overall error at the expense of less accurate individual predictions. On the other hand, the CNN-c approach

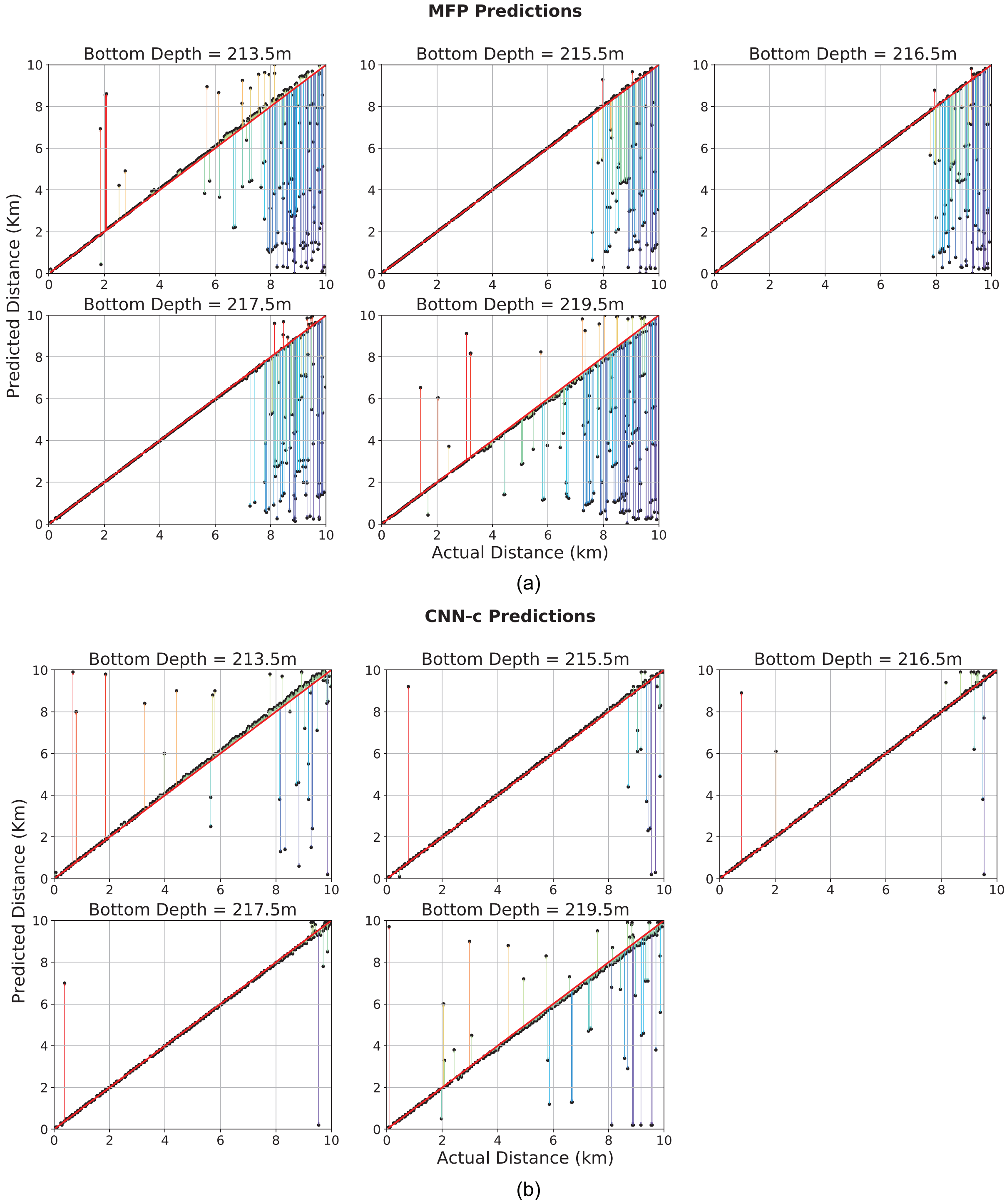
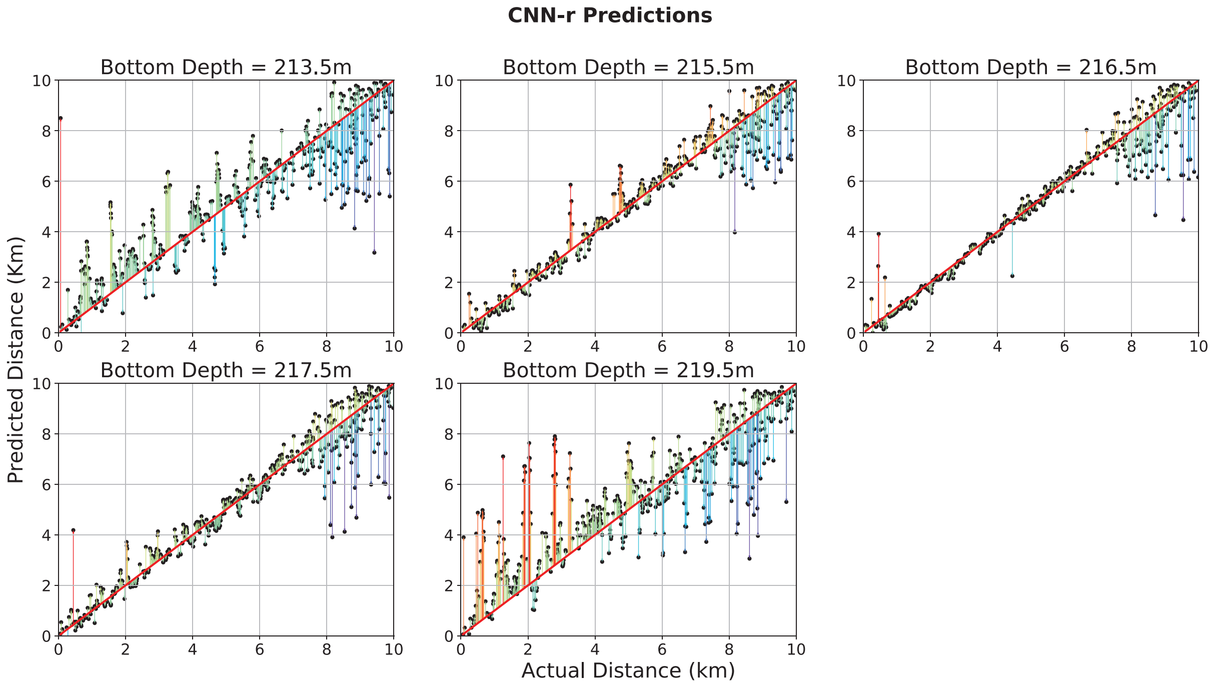


Fig. 6. (Color online) Diagonal solid line represents ground truth. Dots show prediction values. Quivers represent the difference between each prediction and the corresponding ground truth value. (a)–(c) Performance of each approach on testing data generated with different bottom depths.

shows better performance than MFP but is not immune to environmental mismatch, as demonstrated by its prediction errors at similar source ranges as MFP shown in Fig. 9(b). Also, seen on this plot is that there is a consistent overestimation of ∼1 km or more in the MFP and CNN-c predictions from ground truth after the ∼10 min mark along the source track. The CNN-r predictions also greatly overestimates the source range during this part of the track. This overestimation may be attributed to the mirage effect in shallow water[38]—as a source moves over shallower bathymetry than what is modeled, the mismatch leads to the appearance of the source at a greater range than ground truth. Figure 9(a) shows that, indeed, after around the 10 min mark, the bathymetry along the source track begins to become shallower than the modeled bottom depth of 216.5 m. Evidence of the mirage effect on the CNN



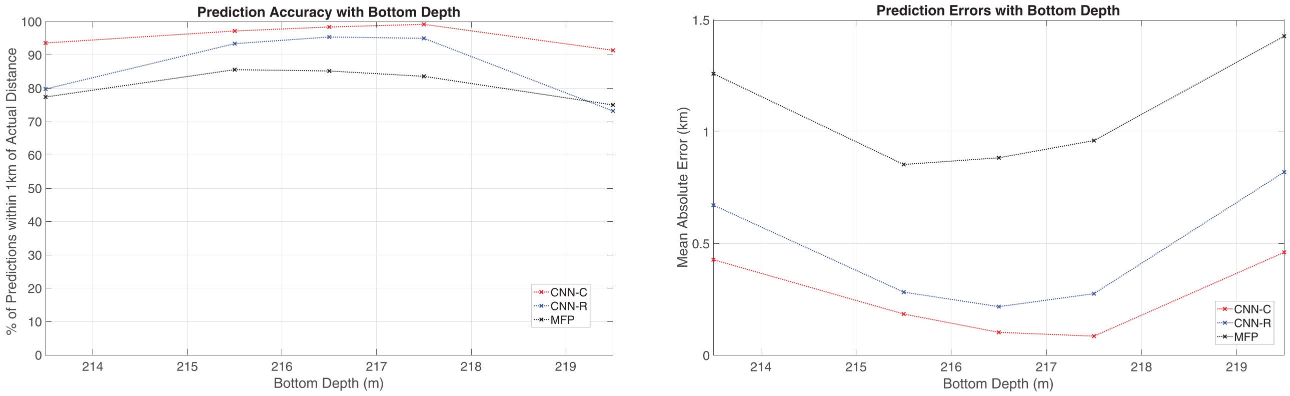
(c)

Fig. 6. (Color online) (*Continued*).

predictions again demonstrates that although this approach may achieve more robust performance than MFP (CNN-c in this case), it is nonetheless subject to the same mismatch challenges that exist for all model-based methods.

V. INSIGHT INTO HOW CNNS ACHIEVE THEIR PERFORMANCE

In this section, we explore in more detail how the CNN approaches achieve their more robust performance compared to MFP. To do this, it is helpful to examine an intermediate output of the networks. In MFP, source range predictions are made by essentially comparing the recorded data replica vector (in the form of data SCM) with the template of modeled replica vectors. Analogous to the MFP template replica vectors for the CNN approach would be the pre-prediction output vectors of the fully connected (FC) layer (last layer before prediction layer) in the trained CNN-c and CNN-r networks. The MFP replica vectors consist of 32, complex-valued entries while the CNN FC-layer template vectors are 256-element, real-valued vectors (the length of the vectors match the number of nodes in the FC-layer). Both sets of vectors are used in the last calculation step in their respective approach before a prediction is outputted. We use the ICEX16 environment with the source at 33 km as a specific example and demonstrate how the MFP,



(a)

(b)

Fig. 7. (Color online) Performance metric comparison between MFP, CNN-c, and CNN-r with various bottom depths. CNN-c shows improved performance to MFP in all cases by both metrics. CNN-r shows similar or improved performance over MFP in all cases by both metrics; however, it still may not be preferable to MFP based on results shown in Fig. 6.

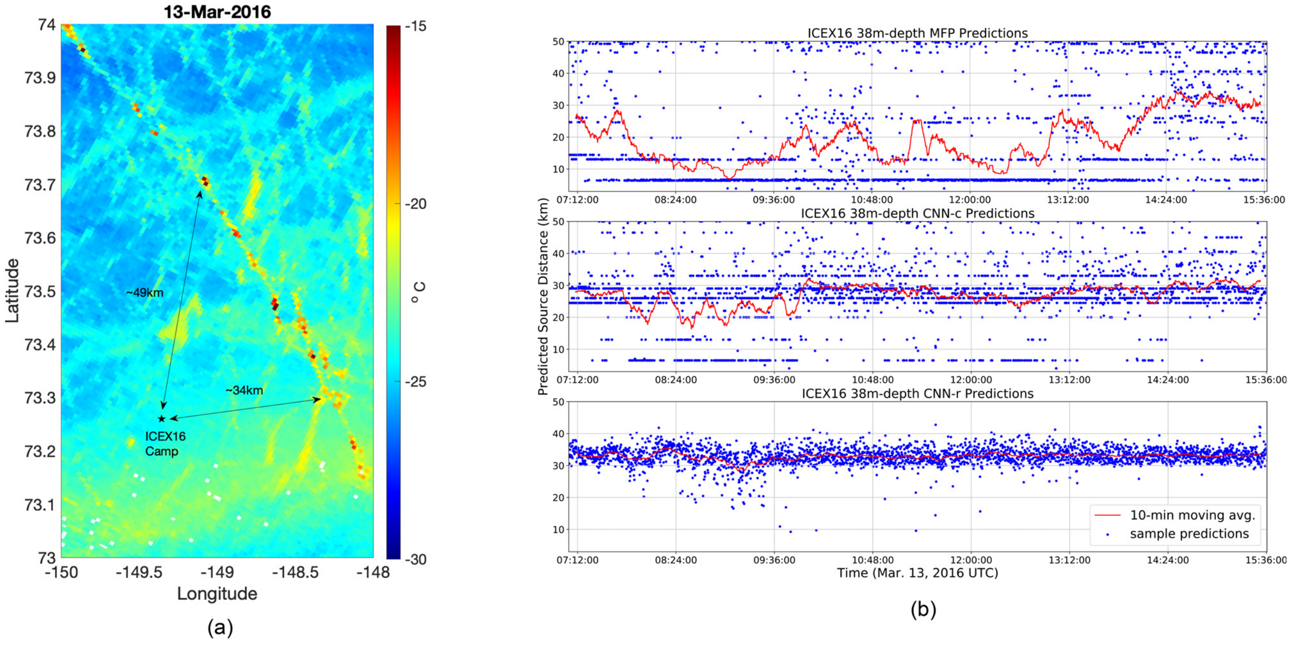


Fig. 8. (Color online) (a) Ice temperature satellite imagery on the day of field data collection shows a prominent ice ridge ~30–50 km away from the VLA (ICEX16 camp) location, as reflected by the warmer ice temperature. (b) Prediction outputs from MFP, CNN-c, and CNN-r on collected ambient noise data. Dots represent individual predictions; solid lines denote 10-min moving averages.

CNN-c, and CNN-r vector sets are affected by different amounts of SSP mismatch. For this demonstration, we first need to define a quantitative measure to describe how different one vector is from another. We adopt the Euclidean distance between two vectors, which is defined as

$$Distance = \sqrt{\sum_{n=1}^{N}\left[\overline{(\vec{v}_i[n]-\vec{v}_j[n])} * (\vec{v}_i[n]-\vec{v}_j[n])\right]}, \quad (3)$$

where $\bar{x}$ denotes the complex conjugate of x and N is the length of the vectors.

Taking a look at MFP first, Fig. 10(a) shows the normalized Euclidean distance between the MFP template replica vectors with a data replica vector simulated with the source at 33 km under no SSP mismatch (0% change to BL strength). The normalization is accomplished by dividing by the maximum distance between the data vector and every vector in the template set. Unsurprisingly, because there is no SSP mismatch, the data replica vector is exactly the same as the MFP template replica vector with source at 33 km. Thus, the Euclidean distance between the two vectors is 0. Away from the correct source distance of 33 km, the Euclidean distance values increase very rapidly such that there is a sharp and narrow minimum at 33 km. Because of this steep minimum, it is obvious from this plot that MFP should output 33 km as the correct prediction, which it does, as shown by the dotted vertical line in Fig. 10(a). However, when SSP mismatch is introduced, the correct output becomes much less obvious in the Euclidean distance plot. Figure 11(a) shows that when the data replica vector is generated under SSP mismatch (again with source at 33 km), the normalized Euclidean distance between the data replica vector and the MFP template vector for source at 33 km grows to about the same value as that of any other vector in the template set—nearly all Euclidean distance values in Fig. 11(a) are between 0.75 and 1 and there is no longer a steep and obvious minimum. While MFP may still output a prediction close-to-correct answer in this case (as shown by plots for 0.1% inc., 0.25% inc., and 0.25% dec. to BL strength), it is also more likely than the no mismatch case that MFP will output a very inaccurate prediction (as is the case for 0.5% inc., 0.1% dec., and 0.5% dec. to BL strength). Thus, we can view the Euclidean distance metric as a proxy for predictive confidence. When there is no SSP mismatch, MFP has much higher confidence that the correct source range is 33 km than any other range value. However, when mismatch is introduced, the predictive confidence of MFP for the correct source range decreases significantly more compared to other range values. As a result, the MFP prediction may become very inaccurate.

Now we examine the Euclidean distance plots for CNN-c and CNN-r. Similar to the MFP case, we first plot the distance between the CNN FC-layer template vectors and the FC-layer output vector when the source is at 33 km under no SSP mismatch. Figure 10(b) shows that for both CNN-c and CNN-r, the minimum Euclidean distance occurs at the correct source range of 33 km. However, different from the MFP plot [Fig. 10(a)], away from the correct range, the increase in Euclidean distance is more gradual. This is especially true for the CNN-r plot. The more gradual increase away from the minimum means that, unlike MFP template replica vectors, CNN FC-layer template vectors for

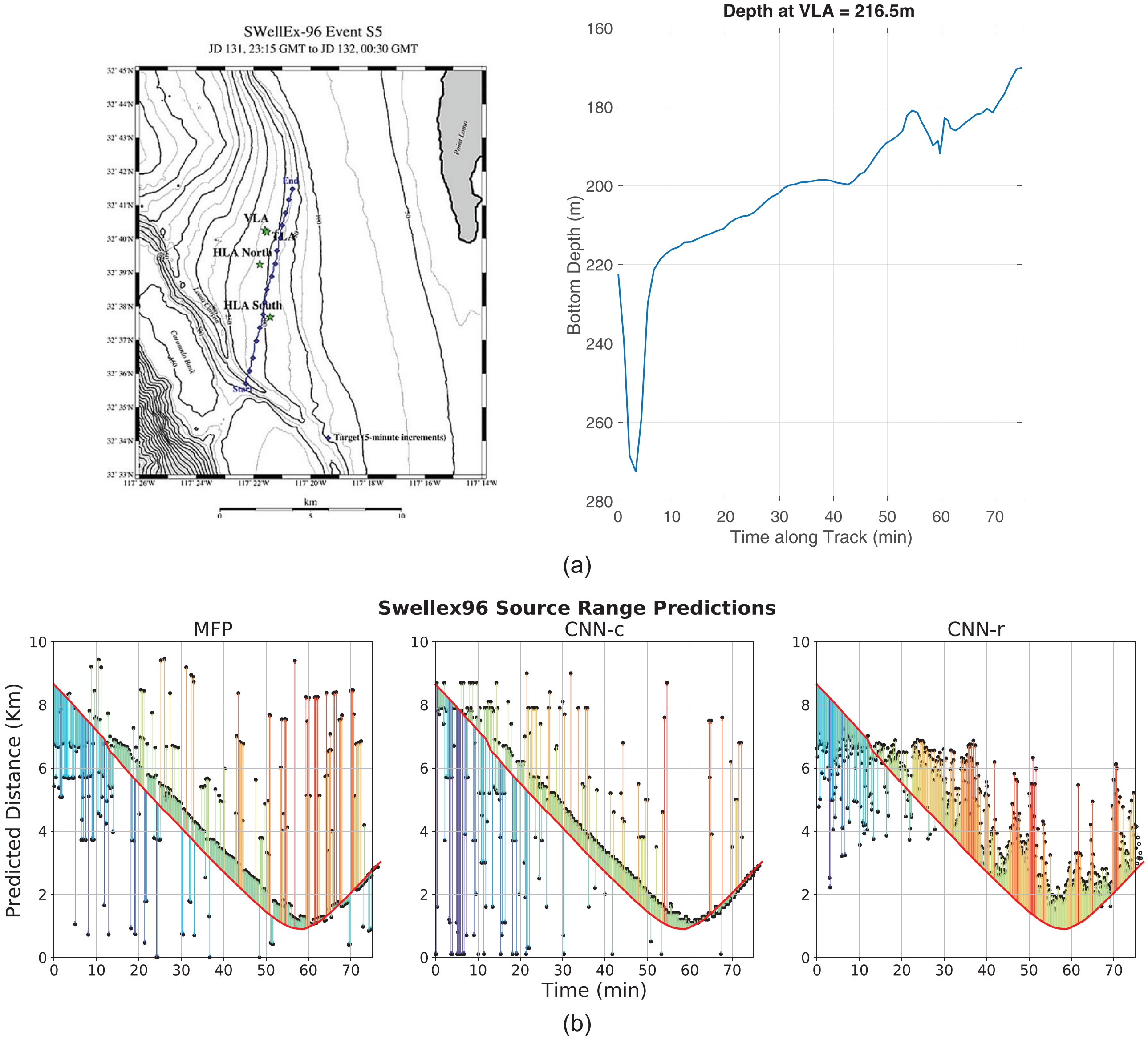
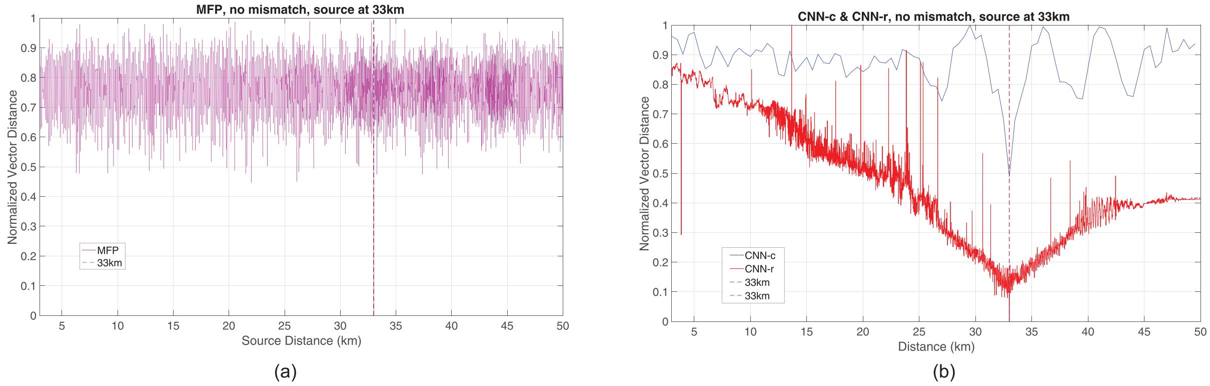


Fig. 9. (Color online) (a) Left plot shows source track (solid line with dots) and the location of the VLA during event S5 of the SWellEx-96 experiment. Figure from SWellEx-96 data website (Ref. 30). Right plot shows the bottom depth along the source track; the bottom depth at the VLA is 216.5 m. (b) Prediction outputs from MFP, CNN-c, and CNN-r on collected S5 data. Solid lines represent ground truth of source range to VLA with time. Dots show individual predictions. Quivers show difference between individual predictions and corresponding ground truth values.

neighboring source range inputs are also closer to each other in Euclidean space. For example, the CNN FC-layer vector for source at 33 km is closer to the vector for source at 32.5 km in Euclidean space than it is to vectors for source at 10 km. This neighboring property of the CNN FC-layer template vectors may be what increases the robustness of the CNN approaches compared to MFP. Given the same amount of environmental mismatch which causes a slight change in the data input, the CNN approaches are more constrained from outputting a prediction drastically different from the true value than MFP because their FC-layer template vectors closest to correct vector also represent source ranges near the correct range value. Of course, if the mismatch causes a large enough change to the data input, the CNN approaches are not immune from making a very inaccurate prediction (as is the case for CNN-c under 0.5% inc. to BL strength). The CNN approaches' increased robustness can be seen in Fig. 11(b), which shows their Euclidean distance plots under varying degrees of SSP mismatch. These plots retain their minima near the correct range (33 km) much better than the MFP plots [Fig. 11(a)]. Although the CNN predictions in

TABLE III. Testing performance on SWellEx-96 S5 event dataset.

Method	MFP	CNN-c	CNN-r
% within 1 km of actual	57.5	70.7	37.8
MAE (km)	1.73	1.41	1.40



(a)

(b)

Fig. 10. (Color online) Plots are generated from simulated data in the ICEX16 environment with no SSP mismatch. (a) Normalized Euclidean distances between replica vectors in the MFP template set and data replica vector with source at 33 km. By definition, the normalized distance of the vector with itself is 0 (at 33 km); away from 33 km, the distance values quickly increase from 0, forming a sharp and narrow minimum. Dotted line shows MFP prediction for this case. (b) Normalized Euclidean distances between CNN-c (blue) and CNN-r (red) FC-layer template vectors with their respective FC-layer output vector with source at 33 km. Note, the normalized distance of the vector with itself (at 33 km) is not 0 for CNN-c because there are multiple samples in each output class; the distance shown in the figures represents the averaged distance between an input with every sample in each output class. The CNN plots show more gradual increase away from the Euclidean distance minima than the plot for MPF. Dotted lines show CNN-c and CNN-r predictions for this case.

these cases (dotted lines) are not exactly equal to 33 km, they are very close to the correct value and remain more consistent between different amounts of mismatch than MFP. However, the more gradual increase from the Euclidean distance minima also suggests that the CNN approaches are less certain of the exact correct source range compared to MFP when the environment is precisely modeled (MFP has a very sharp and narrow minimum in this case). Thus, this appears to be the trade-off for the CNNs' improved performance when mismatch is present.

VI. CONCLUSIONS

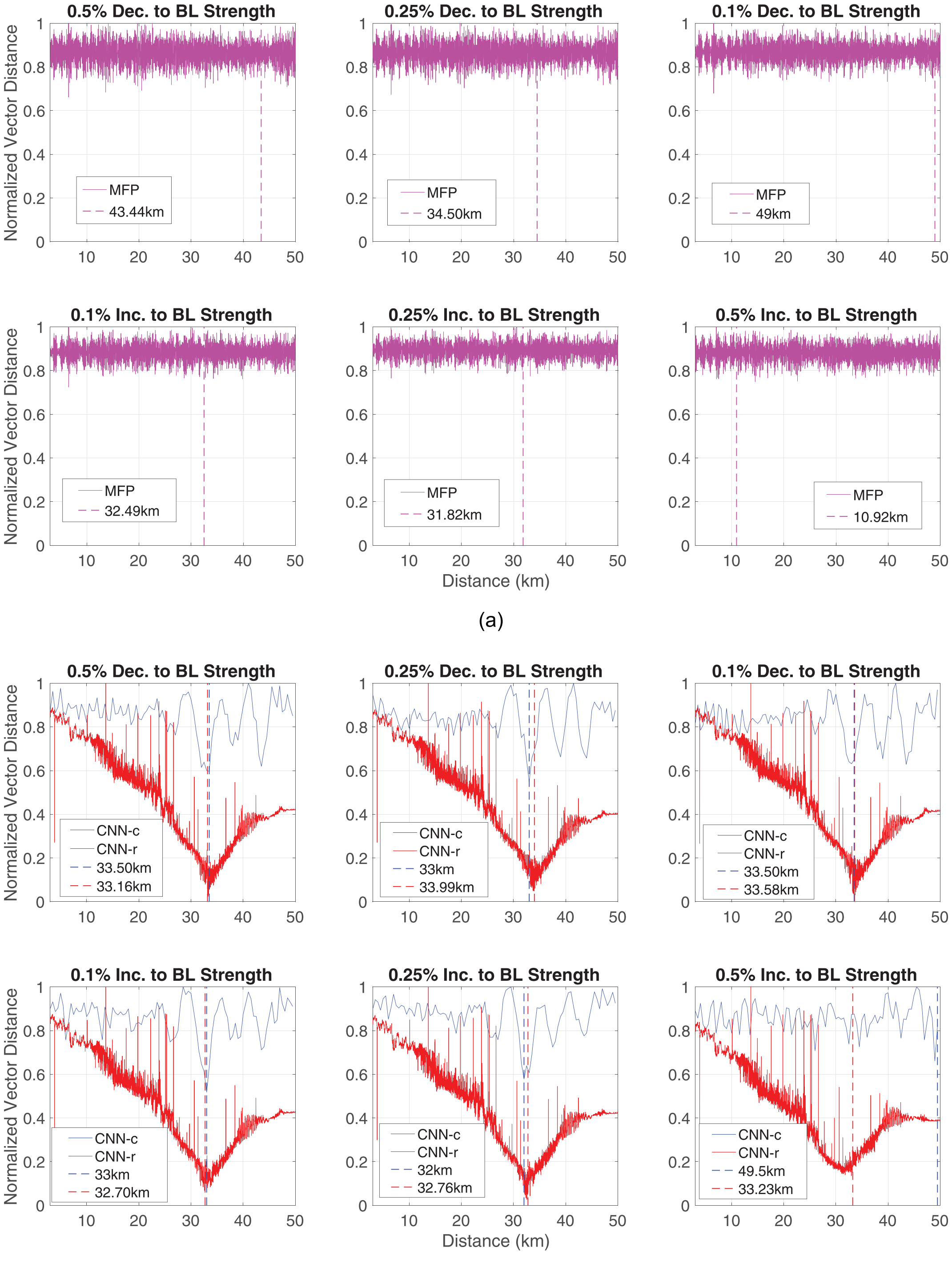
In this study, we set out to answer three questions. The first is whether a model-based ML approach would show more robustness to environmental mismatch than MFP on simulated testing data. To this end, we proposed two CNN approaches (CNN-c and CNN-r) and compared their performance to MFP under SSP and ocean bottom mismatch, respectively, in two separate simulated environments. CNN-c shows improved performance over MFP in both cases. On the other hand, CNN-r performs better than MFP against SSP mismatch but less clearly so against ocean bottom mismatch. The reason for CNN-r's inconsistent performance is likely due to its goal of lowering the overall MSE cost during training. This specification causes CNN-r to have less variability in its predictions, which lowers the overall error of the estimates but increases error on individual predictions compared to MFP.

Second, we used field data collected in the two environments to test whether the performances of our model-based CNN-c and CNN-r transfer to real data. Our results show that, yes, this is indeed the case. For the ICEX16 dataset, both of our CNN approaches return predictions consistent with the expected source range. For the SWellEx-96 dataset, CNN-c outperforms MFP by both of our metrics while CNN-r shows better overall MAE than MFP but is less accurate on individual predictions. These results are consistent with our simulated data test results.

Last, we explored how our model-based CNNs may be achieving their performance by examining their intermediate outputs. Using the ICEX16 environment as an example, we compared the Euclidean distance plots of MFP's template replica vectors to those of the CNNs' FC-layer (pre-prediction) output vectors (Figs. 10 and 11). For MFP, these plots show a sharp and narrow minimum when this is no SSP mismatch; the steep minimum disappears when mismatch is introduced. This result demonstrates that MFP can make very accurate predictions under no mismatch but can become very inaccurate with mismatch. In comparison, Euclidean distance plots for CNN-c and CNN-r show more gradual increases away from their respective minimum. This result means that CNN FC-layer output vectors for neighboring source ranges are also near each other in Euclidean space. Thus, any slight change to the FC-layer vectors as a result of environmental mismatch is less likely to cause the CNN methods to output a prediction that is drastically different from the correct output, as would be the case with MFP. However, the broad minima of the CNN plots also means that these networks are less certain of the correct range prediction than MFP when there is no environmental mismatch.

ACKNOWLEDGMENTS

This research is supported by the National Science Foundation Graduate Research Fellowship under Grant No. 2389237 and by the Office of Naval Research Code 322OA under Grant No. N00014-17-1-2474. The authors would also like to thank Jim Murray and Dave Ensberg for their maintenance of the SWellEx-96 data website. We also thank the attentive reviewers whose thoughtful suggestions further improved this work.



(a)

(b)

Fig. 11. (Color online) Plots are generated from simulated data in the ICEX16 environment with varying amounts of SSP mismatch. (a) Normalized Euclidean distances between replica vectors in the MFP template set and data replica vector with source at 33 km. With SSP mismatch, the expected minimum at 33 km increases to around the same value as at any other range. As a result, the MFP predictions (dotted lines) may become very inaccurate. (b) Normalized Euclidean distances between CNN-c (blue) and CNN-r (red) FC-layer template vectors with their respective FC-layer output vector with source at 33 km. The CNN plots retain their minima near the correct range (33 km) much better than the MFP plots under mismatch. As a result, their predictions remain consistent and closer to the correct range value.

APPENDIX: DESCRIPTION OF ACTIVATION FUNCTIONS

More information is provided here on the four activation functions mentioned in this study. These functions are SELU, sigmoid, linear, and softmax (Fig. 12).

The SELU function is defined as

$$f(x) = \begin{cases} \lambda x, & \text{if } x > 0 \\ \lambda\alpha(\exp(x) - 1), & \text{if } x \leq 0. \end{cases} \quad \text{(A1)}$$

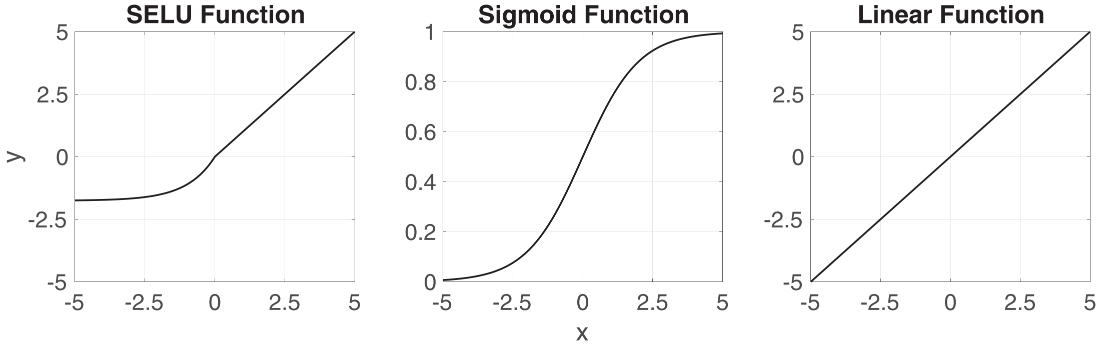


Fig. 12. Plots of activation functions mentioned in this study for $x = [-5\ 5]$. The softmax function has a vector input and is not shown here.

The parameters α and λ are pre-defined constants and not hyperparameters of the CNNs. Their values are $\alpha = 1.67326324$ and $\lambda = 1.05070098$ and are chosen to help standardized the input between consecutive neural network layers.[39] This in turn decreases the chance of vanishing or exploding gradients during backpropagation which are detrimental to network training.

The sigmoid function is another popular activation function, it is a logistic function whose output ranges from 0 to 1

$$y(x) = \frac{\exp(x)}{\exp(x) + 1}. \tag{A2}$$

The linear function is commonly used in the output layer of regression neural networks, it is defined as

$$y(x) = x. \tag{A3}$$

The softmax function is typically used in the output layer of classification neural networks. It takes in a vector input $\mathbf{x}$ of length K and outputs a probability distribution based on the exponential of each entry in the vector

$$y_k(\mathbf{x}) = \frac{\exp(x_k)}{\sum_{j=1}^{K} \exp(x_j)}, \quad \text{where} \quad k = 1, \ldots, K. \tag{A4}$$

[1] H. P. Bucker, "Use of calculated sound fields and matched field detection to locate sound sources in shallow water," J. Acoust. Soc. Am. **59**, 368–373 (1976).

[2] H. Schmidt, A. B. Baggeroer, W. A. Kuperman, and E. K. Scheer, "Environmentally tolerant beamforming for high- resolution matched field processing: Deterministic mismatch," J. Acoust. Soc. Am. **88**, 1851–1862 (1990).

[3] E. K. Westwood, "Broadband matched-field source localization," J. Acoust. Soc. Am. **91**, 2777–2789 (1992).

[4] A. B. Baggeroer, W. A. Kuperman, and H. Schmidt, "Matched field processing: Source localization in correlated noise as an optimum parameter estimation problem," J. Acoust. Soc. Am. **83**, 571–587 (1988).

[5] 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).

[6] Z.-H. Michalopoulou and M. B. Porter, "Matched-field processing for broadband source localization," IEEE J. Ocean. Eng. **21**, 384–392 (1996).

[7] S. P. Czenszak and J. L. Krolik, "Robust wideband matched-field processing with a short vertical array," J. Acoust. Soc. Am. **101**, 749–759 (1997).

[8] 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).

[9] C. Soares and S. M. Jesus, "Broadband matched-field processing: Coherent and incoherent approaches," J. Acoust. Soc. Am. **113**, 2587–2598 (2003).

[10] K. L. Gemba, W. S. Hodgkiss, and P. Gerstoft, "Adaptive and compressive matched field processing," J. Acoust. Soc. Am. **141**, 92–103 (2017).

[11] D. J. Geroski and D. R. Dowling, "Long-range frequency-difference source localization in the philippine sea," J. Acoust. Soc. Am. **146**, 4727–4739 (2019).

[12] Z. H. Michalopoulou, A. Pole, and A. Abdi, "Bayesian coherent and incoherent matched-field localization and detection in the ocean," J. Acoust. Soc. Am. **146**, 4812–4820 (2019).

[13] G. Byun, F. Hunter Akins, K. L. Gembab, H. C. Song, and W. A. Kuperman, "Multiple constraint matched field processing tolerant to array tilt mismatch," J. Acoust. Soc. Am. **147**, 1231–1238 (2020).

[14] H. Niu, E. Reeves, and P. Gerstoft, "Source localization in an ocean waveguide using supervised machine learning," J. Acoust. Soc. Am. **142**, 1176–1188 (2017).

[15] 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).

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

[17] J. Yangzhou and Z. Ma, "A deep neural network approach to acoustic source localization in a shallow water tank experiment," J. Acoust. Soc. Am. **146**, 4802–4811 (2019).

[18] 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).

[19] J. Chi, X. Li, H. Wang, D. Gao, and P. Gerstoft, "Sound source ranging using a feed-forward neural network trained with fitting-based early stopping," J. Acoust. Soc. Am. **146**, EL258–EL264 (2019).

[20] 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).

[21] 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**, EL317–EL322 (2019).

[22] W. Liu, Y. Yang, M. Xu, L. Lü, and Y. Shi, "Source localization in the deep ocean using a convolutional neural network," J. Acoust. Soc. Am. **147**, EL314–EL319 (2020).

[23] E. Ozanich, P. Gerstoft, and H. Niu, "A feedforward neural network for direction-of-arrival estimation," J. Acoust. Soc. Am. **147**, 2035–2048 (2020).

[24] H. Schmidt, "OASES: Ocean acoustic and seismic exploration synthesis," http://lamss.mit.edu/lamss/pmwiki/pmwiki.php?n=Site.Oases (Last viewed July 24 2020).

[25] J. M. Toole, M. L. Timmermans, D. K. Perovich, R. A. Krishfield, A. Proshutinsky, and J. A. Richter-Menge, "Influences of the ocean surface mixed layer and thermohaline stratification on arctic sea ice in the central

canada basin," J. Geophys. Res. **115**, C10018, https://doi.org/10.1029/2009JC005660 (2010).

[26] T. Howe, "Modal analysis of acoustic propagation in the changing arctic environment," Master's thesis, MIT, Cambridge, MA (2015).

[27] H. Schmidt and T. Schneider, "Acoustic communication and navigation in the new arctic: A model case for environmental adaptation," in *Proceedings of the 2016 IEEE Third Underwater Communications and Networking Conference (UComms)*, Lerici, Italy (August 30–September 1, 2016). pp. 1–4.

[28] S. Carper, "Low frequency active sonar performance in the arctic Beaufort lens," Master's thesis, MIT, Cambridge, MA (2017).

[29] R. Chen, A. Poulsen, and H. Schmidt, "Spectral, spatial, and temporal characteristics of underwater ambient noise in the beaufort sea in 1994 and 2016," J. Acoust. Soc. Am. **145**, 605–614 (2019).

[30] J. Murray and D. Ensberg, "The SWellEx-96 experiment," http://swellex96.ucsd.edu/ (Last viewed December 8, 2020).

[31] 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).

[32] N. Srivastava, G. 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), http://jmlr.org/papers/v15/srivastava14a.html.

[33] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France (July 6–11, 2015), pp. 448–456.

[34] 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," https://www.tensorflow.org/ (Last viewed January 8, 2021).

[35] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 (2014).

[36] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural networks," arXiv:1506.02626 (2015).

[37] M. Tschudi, G. Riggs, D. K. Hall, and M. O. Román, "VIIRS/NPP ice surface temperature 6-min l2 swath 750m, version 1," https://doi.org/10.5067/VIIRS/VNP30.001 (Last viewed July 24, 2020).

[38] 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).

[39] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," arXiv:1706.02515 (2017).