2022

ON-ICE DETECTION, CLASSIFICATION, LOCALIZATION AND TRACKING OF ANTHROPOGENIC ACOUSTIC SOURCES WITH MACHINE LEARNING

Steven J. Whitaker
Michigan Technological University, sjwhitak@mtu.edu

Recommended Citation
Whitaker, Steven J., "ON-ICE DETECTION, CLASSIFICATION, LOCALIZATION AND TRACKING OF ANTHROPOGENIC ACOUSTIC SOURCES WITH MACHINE LEARNING", Open Access Dissertation, Michigan Technological University, 2022.
https://doi.org/10.37099/mtu.dc.etdr/1431

ON-ICE DETECTION, CLASSIFICATION, LOCALIZATION AND TRACKING OF ANTHROPOGENIC ACOUSTIC SOURCES WITH MACHINE LEARNING

By

Steven J. Whitaker

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Electrical Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2022

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Electrical Engineering.

Electrical and Computer Engineering

Dissertation Co-advisor:	*Dr. Timothy C. Havens*
Dissertation Co-advisor:	*Dr. Andrew Barnard*
Committee Member:	*Dr. Tony Pinar*
Committee Member:	*Dr. Lan Zhang*
Committee Member:	*Dr. Miles Penhale*
Committee Member:	*Dr. Laura Brown*
Department Chair:	*Dr. Jin W. Choi*

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank all of my friends who reminded me that one should always strive to take it easy. Thank you Cooper, Ryan, Tyler, Sam, Mike, George and Avril. Additionally, I would like to thank all of the SENSE enterprise team that helped conduct my experiments. Thank you Hannah, Carter, Alden, Jacob, Harvey, Daniel, Sophia, and Elizabeth. Finally, I would like to thank my colleagues and my two co-advisors for helping step through the technical side of this Ph.D. Thank you Dr. Havens, Dr. Barnard, Cooper (again), Evan, Yilin, Nik, Siva, Sunit, Steve, Troy, Walker, Carter (again), and Abby.

List of Abbreviations

ADC	Analog-to-digital converter
AoA	Angle of arrival
AVS	Acoustic vector sensor
CNN	Convolutional neural network
cRIO	Compact real-time Input-Output
DAQ	Data acquisition
dB	Decibels
DOA	Direction of arrival
DNN	Deep neural network
FFT	Fast-Fourier Transform
GPS	Global positional system
IJCNN	International joint conference on neural networks
JASA	Journal of the Acoustical Society of America
LSTM	Long short-term memory
MHA	Multi-headed attention
MIMO	Multi-input, multi-output
MISO	Multi-input, single-output
ML	Machine learning
MLP	Multi-layer perceptron

MSE	Mean-squared error
NI	National Instruments
NLP	Natural language processing
pa-type AVS	Pressure-acceleration acoustic vector sensor
ReLU	Rectified linear unit
RMSE	Root mean-squared error
RNN	Recurrent neural network
SGD	Stochastic gradient descent
SNR	Signal-to-noise ratio
SONAR	Sound navigation and ranging
STFT	Short-time Fourier Transform
STD	Standard deviation
TBPTT	Truncated back-propagation through time
TDOA	Time-difference of arrival
ViT	Vision Transformer

Abstract

Arctic acoustics have been of concern in recent years for the US navy. First-year ice is now the prevalent factor in ice coverage in the Arctic, which changes the previously understood acoustic properties. Due to the ice melting each year, anthropogenic sources in the Arctic region are more common: military exercises, shipping, and tourism. For the navy, it is of interest to detect, classify, localize, and track these sources to have situational awareness of these surroundings. Because the sources are on-water or on-ice, acoustic radiation propagates at a longer distance and so acoustics are the method by which the sources are detected, classified, localized, and tracked. These methods are all part of *sound navigation and ranging* (SONAR).

This dissertation describes algorithms which will better SONAR results without modification of the sensors or the environment and the process by which to arrive to this point. The focus is to use supervised machine learning algorithms to facilitate such technological enhancements. Specifically, neural networks analyze labeled experimental data from a first-year, shore-fast, shallow and narrow water environment. The experiments were conducted over the span of three years from 2019 to 2022, mostly during the months from January to March where ice formed over the Keweenaw Waterway at the Michigan Technological University. All experiments were conducted to analyze a passive acoustic source; that is, the source was non-cooperative and did not

send any localizing pings for active SONAR. The experiments were recorded using an underwater *pa*-type *acoustic vector sensor* (AVS). The data and analysis were done intermittently to update any upcoming experiments with discrepancies found in the analysis to create a more generalized algorithm.

The work in this dissertation focuses on two topics for passive SONAR: localization and classification. Because of the “black box” nature in machine learning, tracking the target source is an extension of localization and thought of as the same goal within machine learning. To introduce and verify the complexity of the testing environment, an underwater acoustic simulation is shown with Ray tracing and bathymetry data to compare with the experimental results used in machine learning. The focus of the algorithms is to produce the best results for the experiments and compare the results with traditional methods, such as a simulation or a linear Gaussian localization with a Kalman filter. Experiments studying neural network types have shown that the *Vision Transformer* (ViT) produces excellent results. The ViT is capable of analyzing acoustic intensity azimuthal spectrogram (azigram) data and localizing a moving target at high accuracy, and the ViT is capable of classifying multiple acoustic sources with the acoustic intensity magnitude spectrogram at high accuracy as well.

Chapter 1

Introduction

1.1 Motivation

It is well known that the global climate change is affecting the Arctic ice layers [1, 2, 3, 4]. In general, the ice layer formations are much different than those which were studied in early acoustic experiments. The majority of multi-year pack ice, which has been extensively studied, is now melting between seasons giving rise to an increase of annually formed first-year ice [1, 2]. The shore-fast ice sheet has previously been composed of multi-year ice that travels to shore on currents and gets trapped in the first-year ice. Due to the overwhelming loss of multi-year ice in the Arctic as a whole, the near-shore environment is now composed of predominantly first year ice.

First-year, shore-fast ice is thinner, more saline, and of different density and strength than multi-year ice [5, 6] and is deserving of specific study into its acoustic properties.

In addition, this changing Arctic environment warrants new investigation into the acoustic detection, identification, and tracking of anthropogenic sources. Because there is less ice in the Arctic environment for longer time periods during the year, there is expected to be increased near-shore anthropogenic activity [7, 8, 9, 10, 11]. This activity may come in the form of Arctic shipping through the Northwest Passage, natural resource exploration, tourism, and both foreign and domestic military activity. It is of interest to determine the location and type of these anthropogenic sources for situational awareness in the ocean battlespace. Sensing of sources in the first-year shore-fast ice environment is non-trivial due to ice ridging and ever-changing ice movements. Furthermore, first-year, near-shore ice is not well understood in terms of acoustic properties. Therefore, new data are required to understand the acoustic transmission paths in the first-year, near-shore ice environment and to validate algorithms for detection, identification, and tracking of anthropogenic sources in shallow water (less than 50 meters) with thin, irregular ice sheets.

New data analysis techniques are explored to adaptively respond to the chaotic under-ice environment in the Arctic near-shore zones. The ice bottom-profiles are constantly changing making a fixed localization algorithm sensitive to errors due to the changing acoustic scattering field. Modern deep learning approaches, enabled by major advances in computational hardware and software architectures, have shown to be very effective for localization, tracking, and classification problems, and are the winning approaches in nearly every major pattern recognition challenge. Deep learning approaches require large databases of existing labeled data: data for which the answer is known—in this case, the location and class of the anthropogenic source. Deep learning algorithms essentially learn implicit models from these labeled data to produce predictions from future measured data.

1.2 Objectives

This dissertation will focus on practical applications to machine learning in the on-ice and underwater acoustic environment. The applications fall under any portion of SONAR, be it detection, classification, and localization. The objectives are as follows:

- † Prepare a test set up for on-ice and under-ice experiments to generate labeled anthropogenic acoustic data for use with deep neural networks.

- † Determine optimal pre-processing methods for acoustic vector sensor data before being analyzed by deep neural networks.

- † Compare the effectiveness of various neural network architecture types for on-ice and underwater acoustic source localization and tracking.

† Compare the effectiveness of various neural network architecture types for on-ice and underwater acoustic source classification.

† Determine an understanding as to why certain neural network architectures may perform better than others for on-ice and underwater sensing applications.

1.3 Layout of Chapters

Chapter 2 is a reprint of a *Journal of the Acoustical Society of America* (JASA) article, titled *Recurrent networks for direction-of-arrival identification of an acoustic source in a shallow water channel using a vector sensor* [12]. The article describes the incremental progress found when studying a deep neural network utilizing an LSTM. The data was conducted over June 2020 using a boat from the Great Lakes Research Center at Michigan Technological University and a handheld GPS receiver to track its position while simultaneously recording the boat's acoustic signature with one *Meggit VS-209 acoustic vector sensor* (AVS). With a single AVS receiver, only *direction of arrival* could be determined, and as such, an additional AVS set up was required for true localization, which leads to the next chapter.

Chapter 3 is a reprint of a *Multidisciplinary Digital Publishing Institute* (MDPI)

Sensors journal article, titled *Through-ice Acoustic Source Tracking Using Vision Transformers with Ordinal Classification* [13]. The *Sensors* article is a continuation of a *International joint conference on neural networks* (IJCNN) conference proceeding, titled *Uncertain Inference Using Ordinal Classification in Deep Networks for Acoustic Localization* [14]. The article describes a novel approach to localization, using an ordinal classification approach. The initial study for ordinal classification attempted to find a measure of uncertainty in the neural network prediction, but two other benefits to ordinal classification were found. The two benefits were a higher accuracy for certain networks and a simpler method to constrain prediction results. The article pushes a newer type of network, a *Vision Transformer* (ViT) that performed exceptionally well. With localization analyzed with a ViT, the next chapter discusses classification using the ViT.

Chapter 4 is a reprint of a *Proceedings of Meetings in Acoustics* (POMA), titled *Using Vision Transformers for classification of through-ice acoustic sources* [15]. With the success of the ViT found in localization, we extended these results to classifying multiple acoustic targets with great success too. The discrepancy between classes limits the extend of this paper with certain classes having a small number of samples while other classes have a high number of samples. This chapter shows the feasibility of classifying acoustic sources in a first-year ice environment using machine learning.

Chapter 5 is the concluding remarks about the dissertation and what can be used with this analysis, as well as what further studies can be continued after analyzing this data.

Chapter 2

Recurrent networks for DOA identification of an acoustic source in a shallow water channel using a vector sensor

This chapter is a reprint of a JASA article on DOA estimation [12]. The permission for reprint has been given in Appendix B.1. Reproduced from “Recurrent networks for direction-of-arrival identification of an acoustic source in a shallow water channel using a vector sensor.” *The Journal of the Acoustical Society of America*, 150(1):111–119, July 2021, with the permission of AIP Publishing [12]. Copyright 2021, Acoustic

Society of America.

2.1 Introduction

Source DOA estimation in shallow water has seen strong advancements for applied water acoustics in the past decade with success specifically in machine learning [16, 17, 18]. It is of interest to determine the location of anthropogenic sources for many applications: naval operations, merchant shipping, and environmental studies, to name a few. Using neural networks to estimate the DOA of an underwater acoustic source is of recent interest, including the use of *multi-layer perceptron* (MLP) networks [19, 20, 21], *convolutional neural networks* (CNNs) [22, 23], and *recurrent neural networks* (RNNs) [24, 25].

This paper discusses conventional and machine learning methods of improving surface-water angle-finding utilizing a single underwater *acoustic vector sensor* (AVS). Generally, multiple sensors working together are required to find the angle-of-arrival of a signal source [19, 26, 27]. A pressure-particle acceleration (*pa*) AVS is capable of determining the angle-of-arrival with a triaxial piezoelectric accelerometer in a neutrally buoyant body. The triaxial accelerometer in the AVS generates a vector quantity of the DOA of the acoustic wave[28, 29, 30]. There are different types of AVSs: pressure-particle velocity (*pu*), pressure-particle acceleration (*pa*), pressure-pressure (*pp*), and particle velocity-particle velocity (uu); all have their advantages and disadvantages. This paper solely discusses angle-finding utilizing a *Meggitt VS-209* underwater pa AVS for its broader frequency response, though the methods described here would generalize to any AVS.

We will investigate a shallow RNN architecture and a deep RNN architecture as the machine learning algorithms in the paper. The parameters, such as the inner node lengths and depth of the network, were tested and compared for accuracy. The best models we found with our data are shown in Section 2.4.

2.2 Materials and methods

2.2.1 Acoustic vector sensor

The *Meggitt VS-209* AVS consists of a hydrophone and a triaxial accelerometer oriented with its $-x$, $-y$, and $-z$ orientations—as shown in Fig. 2.1—with respect to the physical sensor's orientation. The underwater pa-type AVS records the particle acceleration in three orthogonal axes together with a scalar underwater sound pressure measurement. The particle acceleration and sound pressure are combine to produce a sound intensity vector, where the intensity vector contains the strength and angle-of-arrival of all the incident wavefronts.

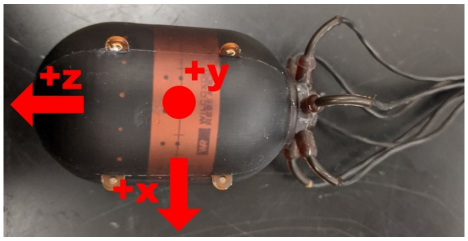


Figure 2.1: Underwater *acoustic vector sensor* (AVS) accelerometer orientation

2.2.2 Acoustic post-processing

The estimation techniques in this paper require some post-processing of the AVS data. Let $\mathbf{a}_x(t)$, $\mathbf{a}_y(t)$, and $\mathbf{a}_z(t)$ be the three components of the time-domain accelerometer data, and $\mathbf{p}(t)$ be the pressure time-series data from the underwater *pa*-AVS. To account for sensor bandwidth and noise, the sensor measurements are first projected into the frequency domain, where $\mathbf{a}_x(\omega) = \mathcal{F}(\mathbf{a}_x(t))$ is the Fourier transform of $\mathbf{a}_x(t)$, and respectively for each component of the sensor data. Since we are concerned with a moving acoustic source, a *short-time Fourier transform* (STFT) facilitates its time-dependence. Using the STFT, we compute $A_x, A_y, A_z, P \in \mathbb{C}^{N\times T}$ for the respective three time-domain accelerometer data and hydrophone data where N is the block-size of the STFT and T is the number of time-series samples divided by the block-size, rounded down. Eqs. (2.1) and (2.2) are computed along each axis with only the x-axis shown for brevity. The measurements are composed into the crosspower spectra, via

$$G_{A_xP} = A_x^*P, \tag{2.1}$$

where A_x^* is the complex conjugate of the frequency domain accelerometer data in the x-axis direction and P is the pressure vector. With the crosspower spectra, $G_{A_xP} \in \mathbb{C}^{N \times T}$, the acoustic intensity is computed as

$$I_x = \mathbb{R}\left\{\frac{G_{A_xP}}{j\omega}\right\}, \tag{2.2}$$

where $I_x \in \mathbb{R}^{N \times T}$ are the active intensity levels in the x-axis direction. The intensities are computed for all three axes, i.e., the x-, y-, and z-directions corresponding to the 3-axis accelerometer. With the three AVS-relative intensity orientations, an intensity vector, $\mathbf{I}_r = (I_x, I_y, I_z)^T \in \mathbb{R}^{3 \times N \times T}$, can be composed. The intensity vector is relative to the orientation of the AVS, as shown in Fig. 2.1.

The *Meggitt VS-209* AVS has a magnetic heading sensor and a gravitational sensor to remove any relative orientation in data collection. The pitch, roll, and heading are the respective rotations along the x-, y-, and z-axes in Fig. 2.1. A rotation matrix, Q_{fixed}, is calculated from the magnetic and gravitational sensors [31], such that

$$\mathbf{I}_g = \begin{bmatrix} I_{west} \\ I_{north} \\ I_{up} \end{bmatrix} = Q_{fixed}^T \mathbf{I}_r = Q_{fixed}^T \begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix}. \tag{2.3}$$

After the rotation, the intensity vector $\mathbf{I}_g$ is no longer oriented with respect to the sensor's orientation; instead, it is oriented relative to magnetic North and the gravity vector. We call this a global coordinate system, and global angle measurements are now considered for localization.

The re-oriented intensity vector, $\mathbf{I}_g = (I_{west}, I_{north}, I_{up})^T$, is then converted to a spherical coordinate system with

$$|I| = \sqrt{I_{north}^2 + I_{west}^2 + I_{up}^2}, \tag{2.4a}$$

$$\Theta = \arctan \frac{I_{west}}{I_{north}}, \tag{2.4b}$$

$$\Phi = \arctan \frac{I_{up}}{\sqrt{I_{north}^2 + I_{west}^2}}, \tag{2.4c}$$

where $|I|$, Θ, and Φ are the magnitude of the acoustic intensity vector, azimuth angle, and elevation angle of the received signal, respectively. Notice that each of these is a function of frequency and time. The magnitude of the intensity vector shows the signal strength at each frequency at a specific time. $|I|$ is an indicator of the *signal-to-noise ratio* (SNR) in the system. The two angles show the DOA of the incident sound wave at each frequency at a specific time. If a particular magnitude of the signal, $|I_{\omega_i,t_i}|$, is at the noise floor, then the associated angle of arrivals, θ_{ω_i,t_i} and ϕ_{ω_i,t_i}, correspond to a DOA of noise; therefore, the measurement at that frequency is not a useful measurement. A noise gate is used to remove these angles at the noise floor in post-processing. Table 2.1 shows the post-processing parameters used in this paper.

Table 2.1
Post-processing Parameters

Parameter	Value
Sample Rate	17067 Hz
STFT Block-size	1706 Samples
STFT Zero Padding	1024 Samples
Noise Gate Threshold	-40 dB (re 1 pW/m^2)
Frequency Range	100 – 8000 Hz

In the experiments in this paper, all signal sources are assumed to be on the surface of the water; hence, we only need to estimate the azimuth angle Θ from the AVS signals. Also note that this paper focuses on DOA estimation; so, range is not of interest. To determine the estimated azimuth angle, $\boldsymbol{\theta}^*$, of the signal source in our experiment, Θ must be processed along its frequency axis into a single angle prediction at each time step, such that

$$\theta_t^* = f(\theta_{f,t}). \tag{2.5}$$

To process Θ in a machine learning approach, a linear regression—i.e., *single-layer perceptron* (SLP) network—can be trained to output $\boldsymbol{\theta}^*$ using the input Θ. Comparatively, a conventional approach can average Θ along its frequency axis to generate a $\boldsymbol{\theta}^*$ angle prediction.

After processing Θ to estimate $\boldsymbol{\theta}^*$, time-series filtering can be performed to smooth out the effect of noise and outliers to generate more realistic results. Considering machine learning, our hypothesis is that an RNN architecture can be trained to output a better estimate of $\boldsymbol{\theta}^*$ than conventional averaging, enhancing the localization performance of the AVS.

2.2.3 Weighted average

We use a weighted average with our experimental data to demonstrate a conventional approach for combining the predicted DOA of an acoustic signal from an AVS. For each frequency component in the AVS signal, there is an angle measurement Θ and intensity measurement $|I|$. The intensity measurement is directly proportional to the SNR; hence, the intensity is used as a weight for the angle measurement. The sampled-based average of the weighted angles are the estimated $\boldsymbol{\theta}^*$. It follows that

$$\boldsymbol{\theta}^*_{avg} = \frac{\sum_{i=1}^{N} |I_{f_i}| \boldsymbol{\theta}_{f_i}}{\sum_{i=1}^{N} |I_{f_i}|}, \tag{2.6}$$

with the intensities, I, in *decibel* (dB) scale normalized on the interval $[0, 1]$, and each f_i term corresponds to a frequency bin from $i = 1, 2, ..., N$. This estimate gives more weight to an angle that has a stronger corresponding intensity, with the assumption that this signal is emanating from the direct path of the source to be localized. This approach works well with high SNR measurements [29], though the results deteriorate appreciably with band-limited, low SNR responses, as demonstrated in Section 2.5. When the acoustic source generates a strong signal, the acoustic intensity, I, at that point dominates the weighted average, while a weak signal will vary greatly depending upon the noise. To address this degraded performance with low SNR measurements, we next explore use of an SLP as an alternative approach to estimate DOA.

2.2.4 SLP network

While the weighted average is a reasonable approach for processing the AVS measurements into a predicted DOA, there are numerous sources of error which are not taken into account. The source may be a band-limited signal and thus only be present in certain frequencies; there may be signal outside these bands that emanates from other sources, say marine mammals, other underwater activity, or noise. Hence, in order to implicitly learn the best relationship between the AVS measurements, $|I|$ and θ, we will employ machine learning, specifically a neural network. For this experiment, we use an SLP network regression to process the frequency domain of the signal. The SLP network processes the frequency domain angle measurements by

$$\theta_t^* = \sum_{f=1}^{N} w_f \theta_{f,t} + b, \forall t, \tag{2.7}$$

where w_f is a vector of weights for each frequency bin in $\boldsymbol{\theta}_t$ and b is a scalar bias. In essence, if $w_f = 1/N$, $\forall f$, where N is the number of frequency bins, and $b = 0$, then the neural network would estimate a non-weighted average of the angle measurements across the frequency axis. To create a weighted average, the neural network learns w_f

and b such that it minimizes E, with respect to the *root-mean squared error* (RMSE),

$$E = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \|\theta_t^* - \theta_t^{true}\|^2}, \tag{2.8}$$

where θ_t^{true} is the true angle measurement (or label) and the neural network predicts θ_t^* at each time step, t.

Since the AVS is the source of the angle measurements, the neural network must minimize a modified RMSE that considers the AVS's polar nature. The angle measurements for the noise source are wrapped around a $-180°$ and $180°$ range, so a circular RMSE where the error is the difference between two angles is necessary. This is important because a prediction that is at $-179°$ with a true angle at $181°$ should have an angle difference of $2°$. A standard RMSE would have an angle difference of $358°$, overly penalizing this small error. The circular mean squared error that the neural network incorporates is

$$E = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \left|\arctan\frac{\sin d_t}{\cos d_t}\right|^2} \tag{2.9}$$

where $d_t = ||\theta_t^* - \theta_t^{true}||_1$ is the absolute difference of predicted angle and truth angle at each time step, t. The SLP processes the AVS measurements in a linear fashion—see Eq. (2.7)—hence, this algorithm may be unable to capture non-linearity present in the system. Thus, we next describe a neural network architecture that can better model non-linearities.

2.2.5 Recurrent neural network

The SLP network is useful in determining the frequencies at which a band-limited signal is present; the learned weights w_f in Eq. (2.7) show how the SLP weights the measurements at each frequency. On the other hand, the SLP architecture does not handle time-dependent parameters or non-linearity in the environment. Following Eq. (2.7), the SLP estimates at each time step, t, calculated independently of one another. However, an RNN considers the current and previous samples [32]. Thus, an RNN is better able to handle temporal aspects of the signal, creating a time-dependency in its predictions from looking at previous samples. We use a conventional form of an RNN, a fully recurrent neural network with no gates as a basis for the simplest neural network model. A fully RNN predicts with n previous samples and its current sample,

$$\mathbf{h}_{t-n} = \mathbf{w}^T \boldsymbol{\theta}_{t-n} + \mathbf{h}_{t-n-1}^T \boldsymbol{\theta}_{t-n-1}, \tag{2.10}$$

where $\mathbf{w}$ and $\mathbf{h}$ are trainable parameters. Eq. (2.10) is repeated n times, for each $\boldsymbol{\theta}_t$ until

$$\mathbf{h}_t = \mathbf{w}^T \boldsymbol{\theta}_t + \mathbf{h}_{t-1}^T \boldsymbol{\theta}_{t-1} \tag{2.11a}$$

$$\boldsymbol{\theta}^*_{RNN} = \mathbf{h}_t^T \boldsymbol{\theta}_t + \mathbf{b}, \tag{2.11b}$$

where $\mathbf{b}$ is also a trainable bias parameter. There is an inherent issue with fully RNN architectures where $\mathbf{w}$ is back-propagated n times during training. The issue arises with values significantly greater than 1 or significantly less than 1 cause very large gradient or close to zero gradients respectively [33]. For example, with $n = 20$ and $w = 1.4$, the gradient would increase to $1.4^{20} = 836$. An SLP is used to reduce the dimensionality of the RNN backbone and a small n value is used to prevent forms of the gradient descent failing due to this issue. The weights in the RNN—$\mathbf{w}$, $\mathbf{h}$, and $\mathbf{b}$—are learned using the *truncated back-propagation through time* (TBPTT) algorithm [34] to minimize E in Eq. (2.9).

The output of an RNN is either *multi-input, multi-output* (MIMO) or *multi-input, single-output* (MISO), shown in Fig. 2.2. In this paper, the MIMO-type RNN is used for internal layers. With the output of the MIMO-type RNN having the same vector length as the input, the internal layers can be connected multiple times, permitting use of a *deep neural network* (DNN) architecture. The MISO-type RNN is used for the final prediction layer so that a single prediction is made, $\boldsymbol{\theta}^*$. The MISO-type

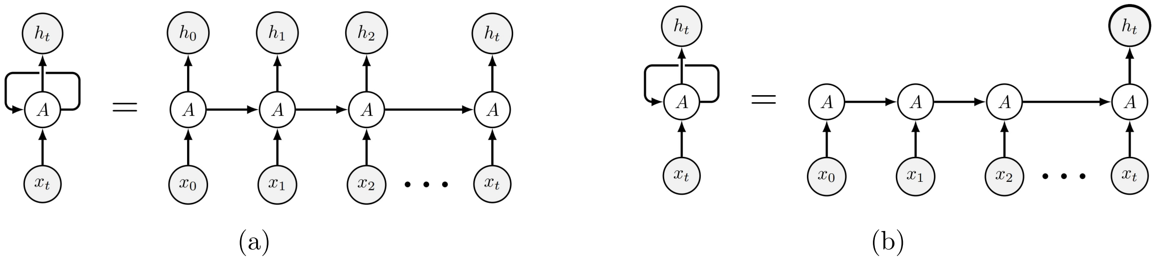


Figure 2.2: (a) Multi-input, multi-output RNN and (b) multi-input, single-output RNN

RNN is useful for predicting a single angle measurement based off of the previous n samples. A combination of both the SLP network and the RNN network can be combined such that the output of one network is the input of another. Now that we have described the basis of the three main algorithms we will use for predicting DOA, we turn to our experiments.

2.3 Experiments

To record angle data, we staged collections from three events on the Keweenaw Portage Waterway in Houghton, Michigan, on July 14, July 27, and August 18, 2020. Fig. 2.3 shows the location of the Keweenaw Portage Waterway in Michigan. The events consisted of driving a boat near the AVS while recording the boat's GPS position at a 1 Hz sample rate. The three experiments total roughly 79 minutes of GPS and acoustic data. A bathymetric cross section and measured sound speed profile is shown in the Section 2.6. The sensor data were recorded using a *data acquisi-*

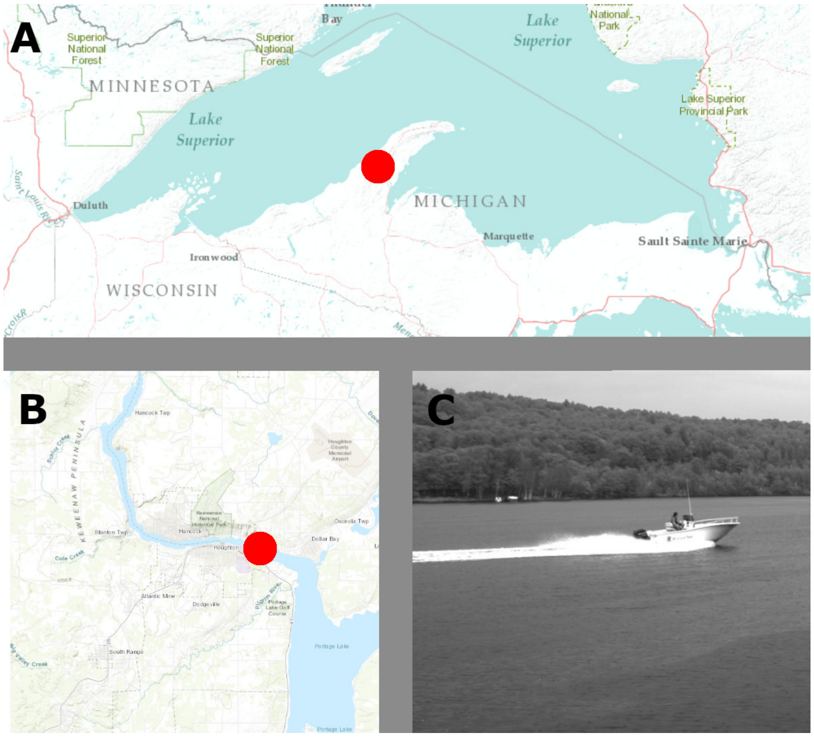


Figure 2.3: Experiment location at Keweenaw Portage Waterway, (A) location in upper peninsula of Michigan, (B) location in Keweenaw peninsula, and (C) on-site location of experiment.

tion (DAQ) unit, the National-Instruments (NI) cRIO-9035, which has 8 slots for NI C-series modules. The C-series modules used in this setup were two NI-9234 *analog-to-digital converters* (ADC) for reading the acoustic data, one NI-9467 GPS receiver for timing and location, and one NI-9344 switch module for system-related control. The NI-9234 ADC has 24-bit precision and stored each data point as a 32-bit, single floating point number. The acoustic data collected on the cRIO-9035 were sampled at 17.067 kHz and chunked into 4-minute intervals. These intervals are continuous, meaning that there is no missing data between each 4-minute interval. The 17.067 kHz sample rate was used since this rate is the closest discrete range that the NI-9234 module has above the Meggitt VS-209 *pa*-AVS 3-dB frequency cutoff above 7 kHz.

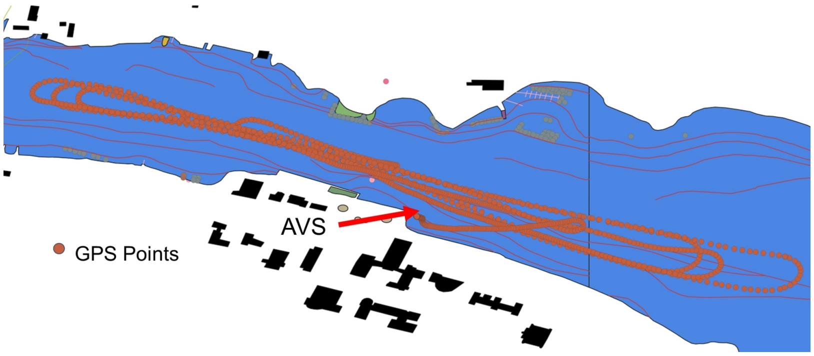
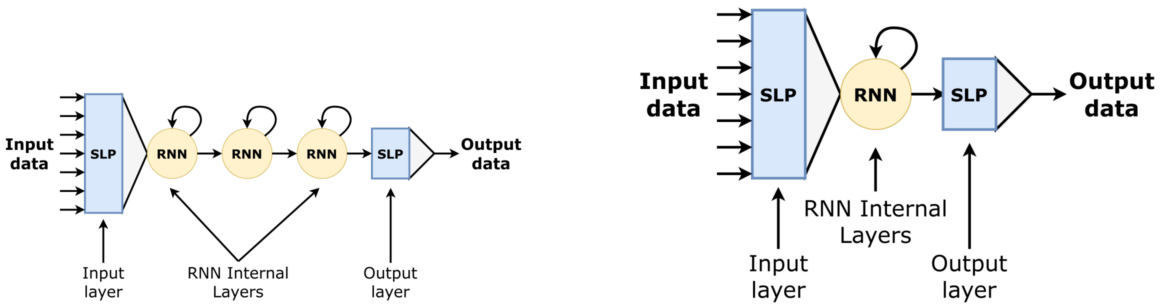


Figure 2.4: First experiment's GPS data

The post-processing of these data, described in Table 2.1, converts the 17.067 kHz sampled data into $1,023$ frequency bins at a block-size of 0.1 seconds using the STFT. The four AVS channels are used to generate Θ in Eq. (2.4). Since the GPS data were recorded at 1 Hz, we linearly interpolated between GPS measurements to match the time interval at which the AVS data were post-processed. Fig. 2.4 shows the 1 Hz rate at which the GPS locations were mapped onto the Keweenaw Portage Waterway.

2.4 Architectures

Table 2.2 shows the parameters used within the two compared RNN architectures and Table 2.3 shows the layer structures, which are illustrated visually in Fig. 2.5. The optimizer used is *stochastic gradient descent* (SGD) with a learning rate of 0.01. No activation function is used on the output layer of the neural network to prevent any skewing of the angle measurement data. The experimental data is split between training and testing for the machine learning algorithm 20 times, so that 20 different



(a) (b)

Figure 2.5: Deep RNN (a) and shallow RNN (b) architectures

Table 2.2
Experimental Parameters

Parameter	Value
SLP Activation	None
RNN Activation	tanh
RNN Lookback	5 Steps
Epochs	20
Train/Validation/Test split	90%/5%/5%
Optimizer	SGD
Learning rate	0.01

models are generated per neural network architecture to test on every portion of the data set in a cross-fold validation setup. Within a single data split, 5% of the training data is used as validation data to determine lowest error in the training set. Then, the neural network predicts the test data using the lowest validation error along each fold of the data split. To generate the network architectures, we use the Keras open-source library for its simple modularity and ease of use. Since Keras is written in Python, the AVS post-processing in Section 2.2.2 is also written in Python.

Table 2.3
RNN architecture shape

Layer Type	Deep RNN dimensions	Shallow RNN dimensions
SLP	$\mathbb{R}^{1023\times 1}$	$\mathbb{R}^{1023\times 1}$
RNN	$\mathbb{R}^{1\times 32}$	$\mathbb{R}^{1\times 1}$
RNN	$\mathbb{R}^{32\times 32}$	-
RNN	$\mathbb{R}^{32\times 32}$	-
SLP	$\mathbb{R}^{32\times 1}$	$\mathbb{R}^{1\times 1}$

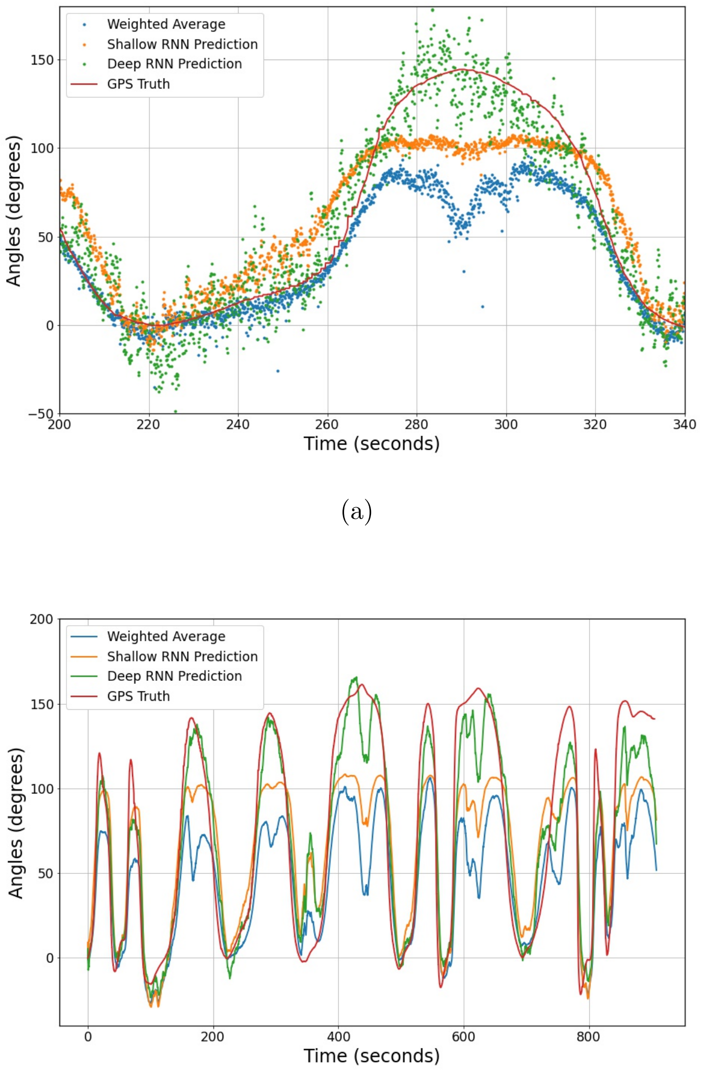
Table 2.4
Root mean squared error results of experiments

	Weighted Average	Shallow RNN	Deep RNN
RMSE	39.4°	33.5°	24.8°
STD	45.3°	22.4°	13.8°

2.5 Results

All results in this section only use the test data defined per model described in Section 2.4. Once the networks have been trained on the experiment training data, the networks are compared with one another. The RMSE of the test data follow Eq. (2.9) and are shown in Table 2.4.

Each neural network has its test data folded 20 times and averaged to yield a RMSE and standard deviation (STD). The time-series predictions of the different algorithms are compared to the total testing truth data in Fig. 2.6 with Fig. 2.6(b) using a Kalman filter added to the output of each algorithm. The covariance of the process noise ($Q = 10^{-6}$) and covariance of the observation noise ($R = 0.025$) are chosen



(a)

(b)

Figure 2.6: (a) Subset of algorithm predictions and (b) full test data predictions with Kalman filter

empirically to show the differences between each algorithm along a larger portion of the data set. It should be noted that no results other than Fig. 2.6(b) use this filtered data; every other figure, table, result and discussion use the original algorithm data.

The results show that the trained deep RNN has the lowest total error throughout the data set, but a single RMSE does not fully convey the deep RNN's results. Another representation is the average angle error with respect to the SNR of the signal. The SNR is calculated by subtracting the ambient acoustic intensity off the acoustic source intensity. A time average of 4 minutes before the acoustic experiment was conducted is used as the acoustic ambient signature. Fig. 2.7 shows the comparison of the acoustic source signal at different boat distances with a time average of 30 seconds each.

Fig. 2.8 shows the error with respect to SNR. These data are presented by averaging the RMSEs according to the respective $0.5-$dB SNR bins, then comparing the results of the three different estimation techniques. For example, at the discrete SNR range of 10 to 10.5 dB, there contains 121 error points inside this range, and the mean of these errors for a deep RNN is 13.47 degrees. The shallow RNN and weighted average at this range have an error of 30.26 degrees and 44.22 degrees. To prevent any discrepancies, if an SNR average contains less than 5 samples within the SNR bin, the SNR average is removed. The data with high SNR correspond to a small portion of very fast crossings of the boat driving by the sensor. Due to the high vessel speed, the experimental timing errors become noticeable at these data.

What is of particular note is in the range from 0 dB to 20 dB SNR. Both RNN architectures perform significantly better than the weighted average. The shallow

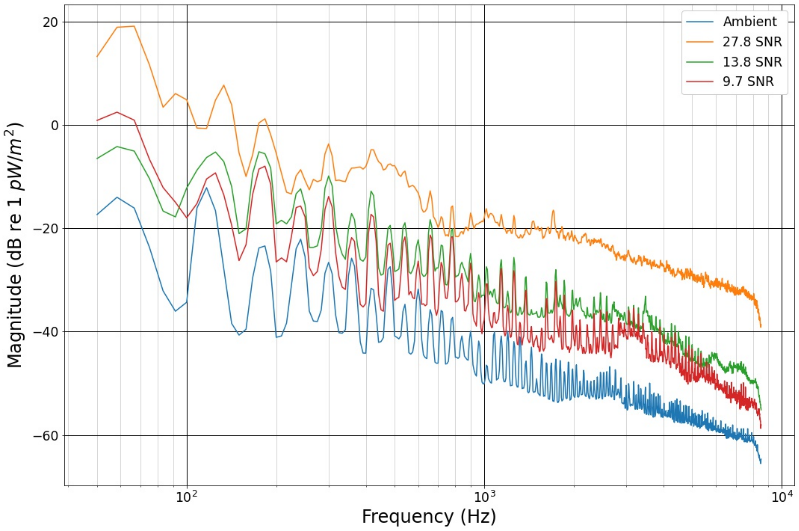


Figure 2.7: Comparison of ambient background and acoustic signal source at varying SNRs

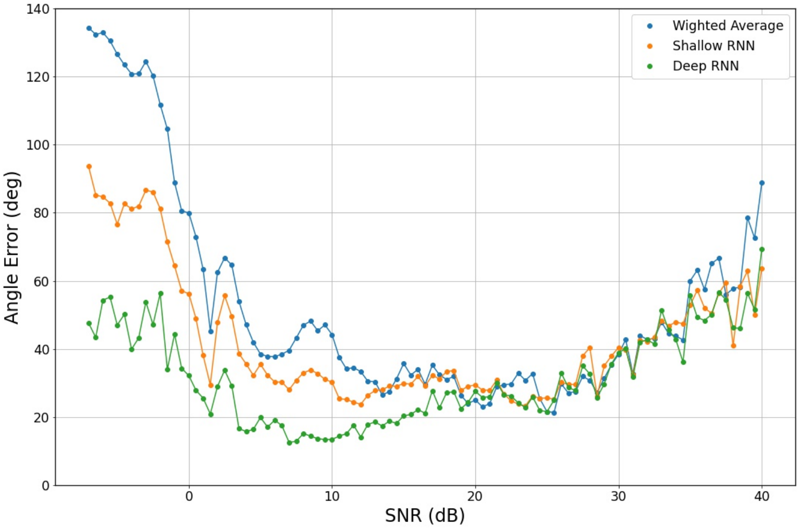


Figure 2.8: Algorithm mean errors with respect to SNR

RNN produces results slightly better than a weighted average of the angles and the deep RNN produces results significantly better than the shallow RNN and a weighted average of the angles inside this range. The shallow RNN architecture gives more non-linearity in the algorithm, but the amount of training data permits the usage of a deeper RNN without overfitting. The large amount of training data prevents the deep RNN from overfitting the data while training.

Each model converges in quality at an SNR of 20 dB. We see that the weighted average algorithm performs equally well to the neural network architectures at this SNR. An SNR of 20 dB is high enough for the weighted average, a linear model, to perform as well as the neural networks, a non-linear model. Our data finds the neural networks unnecessary for signals above 20 dB SNR in our acoustic environment.

In some points in these data, the acoustic source's distance from the AVS is too large, and/or there is no direct acoustic path to the AVS. Using solely the weighted frequency intensity analysis, the results are poor at high angle values, above 100^o, shown in Fig. 2.9. The high angles map to the boat to the west of the sensor, Fig. 2.4 with no direct acoustic path present and is far away from the sensor itself. These data are kept in the analysis, as the purpose of the machine learning algorithms are to work with these highly noisy signals and still map the DOA with higher accuracy than the weighted average. The results in Table 2.4 show this is the case.

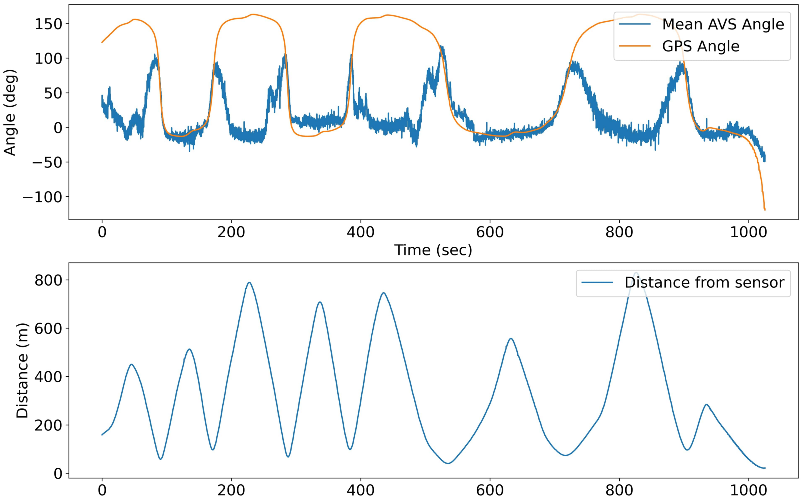
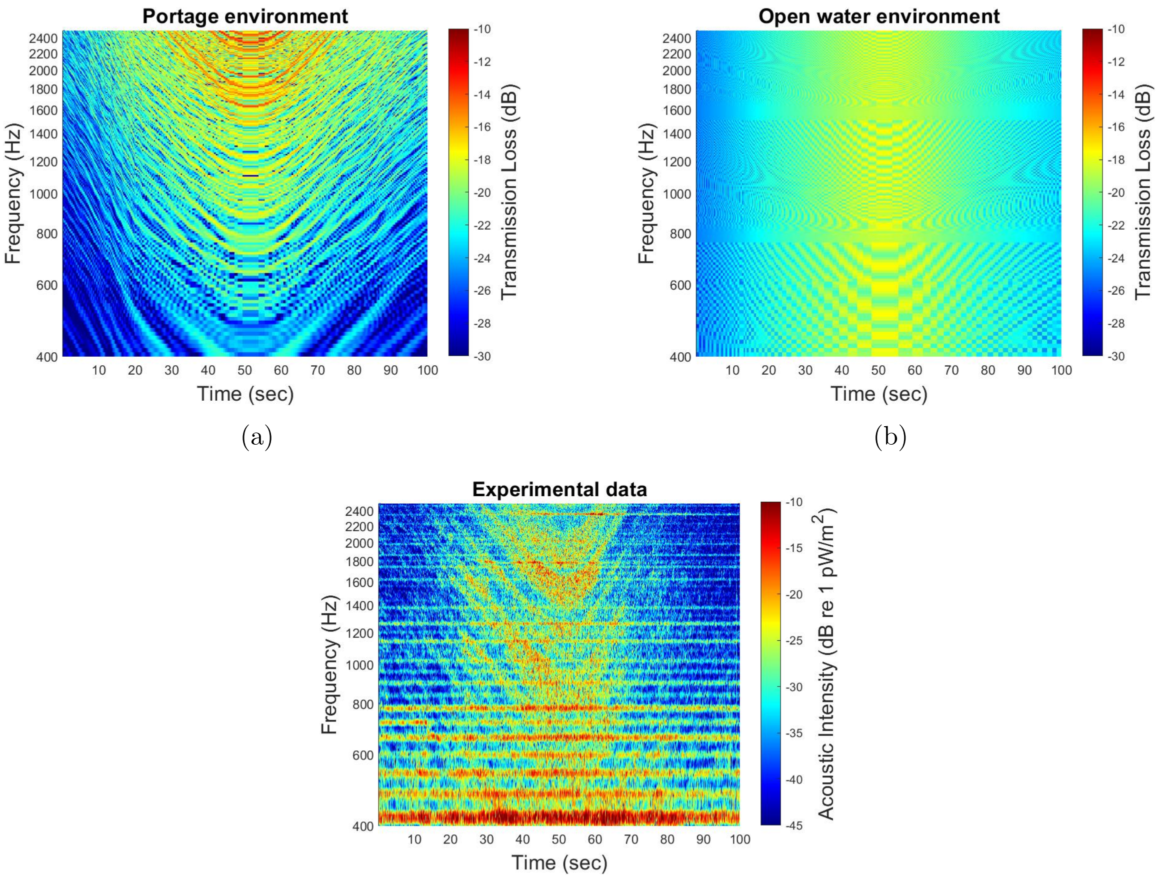


Figure 2.9: First experiment's data with (a) weighted average analysis and (b) the distance from the source.

2.6 Experimental Validation

The experimental data contain multi-path interferences. To validate this claim, two simulations were created to compare the Portage Waterway acoustic channel and an open field. Fig. 2.10 shows a comparison of two RAMGeo [35] simulations (one with multi-path and one without) and the corresponding experimental data. The distance is used equally among all subfigures in Fig. 2.10 using the experimental GPS distances from a single pass in Fig. 2.4, and each simulation time step is computed independently. The Portage Waterway simulation parameters are shown in Fig. 2.11 from recorded bathymetry and water velocity on the Portage Waterway. Note that the sound speed varies by less than 0.05 m/s at 1471.5 m/s.

The open water simulation has the same sound speed velocity with an infinite depth.



(a)

(b)

(c)

Figure 2.10: Moving source past sensor of a single pass for (a) Portage Waterway simulation (b) open water simulation and (c) Portage Waterway experimental data

The swept frequency patterns are a common result of acoustic interference patterns from a moving source in a channel, while the open water simulation contains very little of this pattern. Multi-path constructive and destructive interference is present in the shallow waveguide both in the Portage Waterway simulation and in the Portage Waterway experimental data. The experimental data also show electrical power noise present at harmonics of 60 Hz, common for working with AC power in a marine environment.

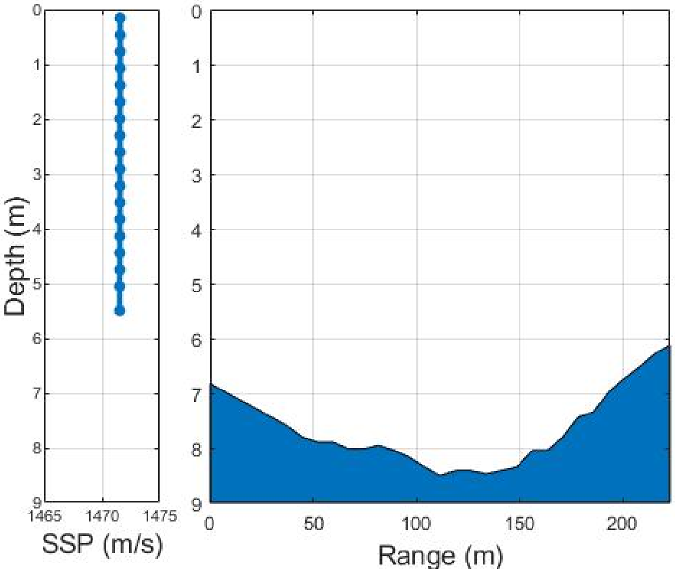


Figure 2.11: Portage Waterway environment simulation input from historical measured data

2.7 Conclusion and future work

In this paper, we compared two types of recurrent neural networks and a weighted acoustic intensity average to predict the direction-of-arrival from acoustic vector sensor data. The recurrent neural networks helped in predicting the temporal aspect of a moving acoustic source. The weighted acoustic intensity average was a good indicator to determine the benefits of using deep learning. Our real-world experiment results suggest that deep neural networks are a strong candidate for use for direction-of-arrival estimation in high-noise scenarios. Conversely, if the signal has a relatively high SNR—our data shows in our environment the threshold is around 25 dB SNR—linear methods, such as weighted averaging or single-layer perceptrons, suffice.

These results encourage further study of the use of machine learning for localization with multiple acoustic vector sensors in difficult-to-model acoustic environments. There is also opportunity to analyze detection and estimation tasks in near-shore ice in Houghton's surrogate Arctic environment [31, 36] with the neural network models. Near-shore ice has been shown to be a difficult acoustics environment [31, 36] and we anticipate that machine learning will show to be a good candidate for increased performance in detection and estimation tasks in this scenario. We are currently carrying out experiments to test this hypothesis. Future work will also examine advanced machine learning methods, such as other deep network architectures—long-short term memory networks [37], transformers [38], etc.—which will be enabled by ongoing data collects.

Chapter 3

Through-ice Acoustic Source Tracking Using Vision Transformers with Ordinal Classification

This chapter is a reprint of the Sensors article entitled “Through-ice Acoustic Source Tracking Using Vision Transformers with Ordinal Classification” [13]. This article is a continuation of the *international joint conference on neural networks*(IJCNN) conference paper [12]. The permission for reprinting the Sensors article is in Appendix B.3.

3.1 Introduction

Acoustic source localization is important in underwater acoustics. In underwater environments, acoustic frequencies propagate long distances, which permits acoustic analysis to be ideal for localization. Localizing a source is beneficial in numerous applications: search and rescue for the coast guard, tracking ships for merchant shipping, and situational awareness for military purposes, to name a few. In a deep water environment, such as the ocean, varying sound speed profiles present challenges in properly simulating the environment [39, 40, 41]. In ice environments, even more challenges arise: multi-path, scattering fields, interference patterns with a reflective ice surface, non-linear propagation through the ice, and a temporally changing field [31, 36]. Additionally, shallow-depth, narrow, ice-covered waveguide environments (e.g., a frozen river or a canal) generate more multipath reflections on the bottom and edges of the environment. These narrow ice environments are important for tracking snowmobiles or other anthropogenic sources on or under the ice. Therefore, Machine Learning (ML) is a promising method to investigate for such a highly complicated environment that can incorporate all the complex water environment and the complex ice environment.

ML has been used previously in acoustic localization approaches with great results [12, 14, 41, 42, 43, 44]. Long Short-Term Memory (LSTM) neural networks have been shown to analyze time-series acoustic data with success [12, 14, 37, 45]. LSTM is designed to analyze data with time dependence [37], but its computational complexity causes difficulty in training large networks, which is shown in Section 3.3. A newer concept is to utilize Vision Transformer (ViT) architectures [46, 47]. The ViT is a modified version of a Transformer neural network [48, 49] where the ViT is specialized for data with a large number of dimensions, e.g., acoustic spectrum data. The ViT has been used extensively in computer vision and image analysis [46, 50, 51], but to date, there has been no paper published on ViT-based localization for through-ice or underwater acoustic localization.

To combine multiple state-of-the-art concepts, our previous work showed that localization framed as a classification problem outperforms regression [14]. With a constrained area of interest, the regression values can be transformed to be classes that represent a grid of positions, and then, the neural network estimates these classes. This classification is an alternative to localizing the source with regression. With respect to our prior research [14], we tested the claims proposed in the classification method with new data and show that the proposed classification method has more nuanced results. We show that networks suited for classification problems show better localization performance with the proposed method, while networks suited for regression problems better localize the source with a regression loss. We validated this claim with newly conducted experiments on ice, a larger training dataset, and new, state-of-the-art neural network architectures.

We show the results of these algorithms with newly recorded data for localizing and tracking on-ice snowmobiles on the Keweenaw Waterway in Houghton, Michigan, by comparing the four described neural network architectures—Convolutional Neural Networks (CNNs), LSTMs, Transformers, and ViTs—with three loss functions: regression, categorical classification, and ordinal classification. We first provide an understanding for how our data are recorded to explain which properties our ML algorithms will exploit.

3.2 Materials and Methods

To record our acoustic source, we used an Acoustic Vector Sensor (AVS), which is capable of recording acoustic pressure and acoustic particle velocity (or acceleration) within a single sensor module [52]. Our experiments used two Meggitt *VS–209* underwater pressure and particle acceleration (*pa*-type) AVSs [53], which record acoustic pressure and acoustic acceleration simultaneously. A *pa*-type AVS consists of a hydrophone and a triaxial accelerometer in the same module and is a good choice for the experiments in this paper because the accelerometers' bandwidth reaches higher frequencies than a *pu*-type (pressure and particle velocity) AVS [53]. A snowmobile's response is a relatively broadband signal; hence, we can record more of the signal source's frequency domain signature. The Meggitt *VS–209* has a bandwidth up to 8000 Hz, and the snowmobile's broadband signal goes up to 10,000 Hz [54, 55], which is also seen in the raw data in Section 3.2.7.1.

3.2.1 Acoustic Post-Processing

A single *pa*-type AVSs generates four time-series data streams. Using a single sensor, we can produce an angle measurement by post-processing these time-series streams. This angle measurement, the Direction Of Arrival (DOA), tells us from which direction the sound arrives, no matter if the sound is from the acoustic source we are trying to track or if the sound is from other sources, e.g., waves crashing, biometrics, or anthropogenic sources that are not our target, to name a few. Each AVS produces its own acoustic intensity, I, with post-processing [52]:

$$I_{x,y,z}(f,t) = \frac{P(f,t)A^*_{x,y,z}(f,t)}{j2\pi f}, \tag{3.1}$$

where $P(f,t)$ is the acoustic pressure in the frequency domain at time t (i.e., P is the Short-Time Fourier Transform (STFT) of the pressure time-series $p(t)$), $A_{x,y,z}(f,t)$ is the three-dimensional acoustic accelerations in the three axial directions, x, y, z, from the AVS accelerometer in the frequency domain at time t, f is frequency, * is the complex conjugate, and $j = \sqrt{-1}$. The *VS-209* contains a coordinate transform to transform the $I_{x,y,z}$ positions into a "global" coordinate system that is aligned from

Earth's magnetic field and Earth's gravitational field:

$$I_{\text{west,north,up}}(f,t) = Q^T I_{x,y,z}(f,t), \tag{3.2}$$

where Q^T is the coordinate transform defined in the *VS-209* system manual. Acoustic intensity is then used for azimuth calculation via

$$\theta(f,t) = \arctan \frac{I_{west}(f,t)}{I_{north}(f,t)}, \tag{3.3}$$

where θ is the azimuth DOA of the acoustic source; east is 0 degrees, and north is 90 degrees. When using an STFT, θ is a spectrum of angles, called an *azigram* [56]. From this point on, we will consider azigrams as a two-dimensional image, where $\theta_{f,t} = \theta(f,t)$, which matches well with the computer vision background of deep networks. The vector θ_t denotes the column of the matrix θ at time t.

Thus far, our post-processing has yet to deal with any aspect of multi-path, scattering fields, interference patterns, or reflections prevalent in this signal, i.e., interferences are still incorporated in θ. Suppose a target were not generating a signal at some time, e.g., the target has moved out of range of the sensors or the target powered off its noise source. In this scenario, angle measurements would come from the ambient background, which often presents a localized noise or "noise coming from certain angles." Because θ is a noisy signal, we need to further process this signal. We will use ML to handle the noise, which is an excellent algorithm for working with high-dimensional and noisy data. Specifically, we discuss four different neural network approaches. The four neural networks we investigated are: (i) a CNN, (ii) an LSTM neural network, (iii) a Transformer neural network [48], and (iv) a ViT [46]. Let us now describe each of these networks in detail and adapt these different networks to our localization problem.

3.2.2 Convolutional Neural Network

The CNN performs convolution operations on the input signal, and in this regard, we perform a 2-dimensional convolution along both the frequency and time:

$$Y = W \star \theta, \tag{3.4}$$

where $\star$ is the convolution operation, W are the trainable parameters in the CNN, and Y is the output of a single CNN operation. The convolution operation, $W \star \theta$:

$$Y_{f,t} = \sum_{i=0}^{F} \sum_{j=0}^{T} W_{i,j} \theta_{f-i,t-j}, \tag{3.5}$$

elucidates local relations spanning across the time domain, $t \in [0, T]$, and the frequency domain, $f \in [0, F]$. The kernel size—i.e., the dimensionality of W—is a parameter that can be adjusted to allow larger relations across time and frequency.

With an activation function, such as tanh or ReLU:

$$\mathrm{ReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

surrounding Equation (3.4), the CNN is now a non-linear transform. CNN layers are extremely powerful in a Deep Neural Network (DNN) [57], but there are some pitfalls. The CNN handles spatially localized features, but the CNN lacks any temporal aspect, i.e., any long-term or temporal relations are not represented or handled. With a CNN, each input is independent of the next. Our data are not independent of each other, since our data are time-series and the position of an acoustic source traveling by the sensors is dependent on its previous position; that is, real-world sources have temporal correlation in their acoustic signal. We incorporated this temporal information with an LSTM.

3.2.3 Long Short-Term Memory Neural Network

LSTMs address some of the weaknesses of CNNs for time-series data. They look into the temporal and long-term relations with the short-term hidden state, $\mathbf{h}_{t-1}$, and long-term candidate state, $\mathbf{c}_{t-1}$, in each LSTM cell, seen in Figure 3.1 [37].

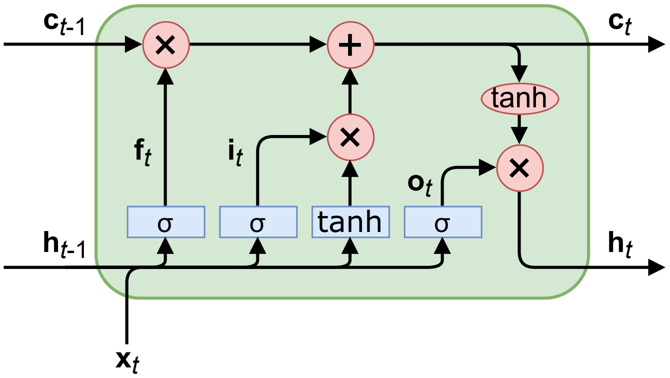


Figure 3.1: A long short-term memory cell, where blue rectangles indicate trainable parameters and red ovals indicate a math operation (non-trainable).

The equations derived from Figure 3.1 consist of "gating" the logical flow. For example, the "forget" gate, $\mathbf{f}_t$, limits how much the long-term candidate state, $\mathbf{c}_{t-1}$, is incorporated into the output, $\mathbf{h}_t$. The other two gates operate similarly; the "input" gate, $\mathbf{i}_t$, limits the effect of input data, $\mathbf{h}_{t-1}$ and $\mathbf{x}_t$, and the "output" gate; $\mathbf{o}_t$, limits the effect of total data on the output, $\mathbf{h}_t$. This is reminiscent of a Kalman filter's capability to adjust the estimate based on its prior knowledge; however, an LSTM can also adjust the output of its prior knowledge in addition to the new measurements. The equations for these gates are

$$\mathbf{i}_t = \sigma(W_i \left[\mathbf{h}_{t-1}^T \; \mathbf{x}_t^T\right]^T + \mathbf{b}_i) \tag{3.7}$$

$$\mathbf{f}_t = \sigma(W_f \left[\mathbf{h}_{t-1}^T \; \mathbf{x}_t^T\right]^T + \mathbf{b}_f) \tag{3.8}$$

$$\mathbf{o}_t = \sigma(W_o \left[\mathbf{h}_{t-1}^T \; \mathbf{x}_t^T\right]^T + \mathbf{b}_o), \tag{3.9}$$

where matrices W and vectors $\mathbf{b}$ correspond to the trainable gate parameters (input, forget, output), σ (shown in Figure 3.1) is the sigmoid activation function, $1/(1+e^{-x})$, and $[\cdot]$ is a concatenation of the vectors. The equations for the LSTM outputs are

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tanh\left(\left[\mathbf{h}_{t-1}^T \, \mathbf{x}_t^T\right]^T\right) \tag{3.10}$$

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t), \tag{3.11}$$

where $\circ$ is an elementwise multiplication.

LSTMs are "chained together" successively using the LSTM cells in Figure 3.1; that is, the output, Equation (3.10), of the previous LSTM cell is the input to the next LSTM cell. This chaining can be used for long-term memory in the system. The vectors, $\mathbf{c}$ and $\mathbf{h}$, are stateful values of the LSTM, i.e., they are dependent on the input data to and internal weights of the LSTM cell (and subsequently, all previous LSTM cells). The LSTM is dependent on its previous state because the outputs of the previous LSTM cell is the input of the next LSTM cell (along with $\mathbf{x}_t$), and so, the mathematical operations are sequential for each LSTM cell. This means the LSTM operations cannot be computed in parallel. Because of this limitation, LSTMs inherently train slower because other neural network architectures can utilize GPU parallel processing more. The training speeds are shown in Section 3.3. The Transformer architecture attempts to avoid the LSTM's sequential computational processing while keeping temporal relations with attention-based networks, which we explain now.

3.2.4 Transformers

Since Transformers have seen promising results in natural language processing [48] and image classification [46], we anticipate Transformers and Transformer variants will perform well in spectrum analysis. Transformers utilize self-attention [58], where self-attention is defined as the normalized dot product:

$$\text{attention}(\theta) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \tag{3.12}$$

where Q, K, and V are the projected query, key, and value tensors: $Q = W_Q\theta$, $K = W_K\theta$, $V = W_V\theta$, where W are trainable parameters [48]. θ are the input data, i.e., the azigram image. The scaling parameter, $\sqrt{d}$, is found to better normalize the data, suggested in [48]. For our data, $d = 512$, the number of frequency bins in the azigram. The softmax function:

$$\text{softmax}(\mathbf{x}) = \frac{e^{\mathbf{x}}}{\sum_{k=1}^{K} e^{x_k}}, \tag{3.13}$$

normalizes the data such that $\sum \text{softmax}(\mathbf{x}) = 1$. Multi-Head Attention (MHA) calculates Equation (3.12) multiple times to permit different attention interconnections with the same data. MHA allows for multiple relations to be found within the same layer in the Transformer.

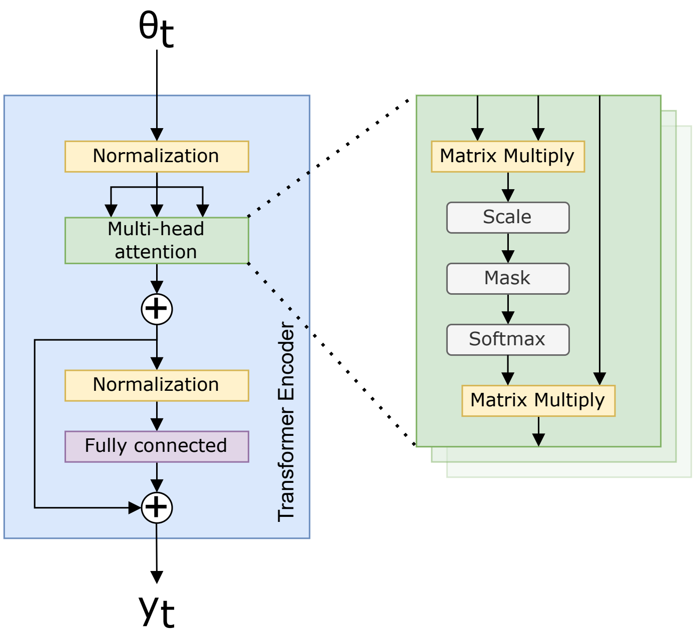


Figure 3.2: Transformer neural network encoder.

The Transformer then projects the results of Equation (3.12) by

$$y = \phi(\theta + \text{attention}(\theta)) + \theta + \text{attention}(\theta), \tag{3.14}$$

where ϕ is a projection operator; in our case, ϕ is a fully connected neural network. Figure 3.2 illustrates Equation (3.14), along with the additional normalizing used within the Transformer architecture. The normalization ensures invariance to scale differences in the feature space, as suggested in [48].

The benefit to self-attention is any abstract relation can be represented within a sample along the temporal and frequency dimensions of our azigram data [48]. This abstraction results in a more broadly applicable CNN. Additionally, a Transformer outperforms the LSTM in training speed with its capability to train in parallel, rather than sequentially, since all the operations in Equation (3.12) are independent of one another. With a spectrum, the Transformer finds attention across all possible azimuth, θ, values, which generates a massive matrix of attended values. If there are a large number of dimensions to which the Transformer attends, there is a large scope to search. The vanilla Transformer struggles to analyze such high-dimensional data. The Vision Transformer better handles this issue using positional embedding.

3.2.5 Vision Transformers

A ViT is a modified Transformer that encodes a highly dimensional image (in our case, an azigram) into smaller patches within its position embedded into the Transformer. A positional embedding is added; Figure 3.3 shows a setup where the spectrum data are chunked into the Transformer with the positions embedded [46].

For example, with 16 positional embeddings and a 512×512 image, the ViT can embed 16 images of size 128×128 in a 4×4 grid pattern, enclosing the 512×512 azimuth input. The positional embedding is a trainable parameter, so this example is not used in the network itself, but rather as a simple representation of the positional embeddings being adjusted by the ViT. Generalizing this example, we change from N^2 parameters with the Transformer to N^2/M attention values with the ViT when

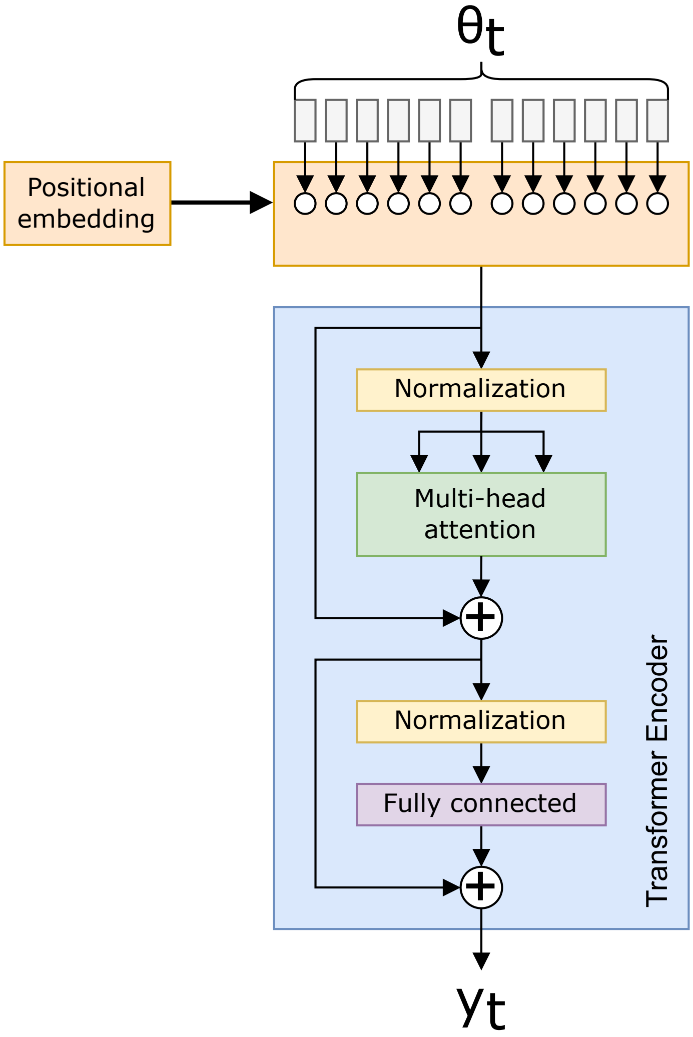


Figure 3.3: Example *Vision Transformer* (ViT) where our input data are positionally embedded prior to passing into the Transformer, being Equations (3.12)–(3.14).

each of M embeddings are the same size [46, 50]. The reduced attention relations are beneficial for data with large numbers of dimensions, the benefits of which are shown in Section 3.3.

3.2.6 Loss Functions

With each of the networks described, we now turn to defining our separate loss functions for localizing our target, the first of which is the "standard," or most common loss function for localization: regression.

3.2.6.1 Regression

A regression loss function is typically an l^p-norm equation, commonly the *Mean-Squared Error* (MSE) or *Root-Mean-Squared Error* (RMSE). For example, the RMSE is

$$L = \left\| \mathbf{p}^* - \mathbf{p}^{true} \right\|_2 , \tag{3.15}$$

where $\mathbf{p}^*$ and $\mathbf{p}^{true}$ are the predicted target position and true target position, respectively.

Fundamental faults of a typical regression loss function are the lack of predicted certainty of the results and the inability to constrain predictions in a nuanced manner. It is of importance in some applications to know how confident the localization is, e.g., tracking the signal while it travels out of the sensors' effective *Signal-to-Noise Ratio* (SNR) ranges. Additionally, a more constrained field of predicted values can benefit performance if one is predicting in a predetermined area (such as the bend of a river) [14]. As such, we will now propose a classification approach whereby we predict locations on a predetermined grid and then aggregate to predict a location. This method also provides the confidence or uncertainty of the prediction.

3.2.6.2 Categorical Classification

A regression loss function provides no measure of confidence and, thus, simply provides a localization estimate even when the network is presented with pure noise. This is not adequate for a generalized solution for localization. In contrast, categorical classification was initially investigated as a method to not only provide a location prediction, but also the confidence in this prediction [12]. Another benefit of a classification approach to localization is that the localization region can be predefined, i.e., a neural network with a classification output can be designed to only predict at specific regions (e.g., water, and not beach). Neural networks with a regression output predict *any* output, and this may not be viable in a real-world scenario, such as a water vessel being constrained to within the banks of a river.

Our categorical classification method manipulates a grid mapping of locations, then predicts the classes in a manner where one can determine the certainty of the network prediction. We use a soft label classification equation:

$$y_k = \frac{1}{\Delta} \prod_{d=1}^{D} \begin{cases} \Delta - |p_d - (\mathbf{c}_k)_d| & \text{if } |y_d - (\mathbf{c}_k)_d| \leq \Delta \\ 0 & \text{otherwise,} \end{cases} \tag{3.16}$$

where $\mathbf{p}$ is the true target position, $\mathbf{c}_k$ is the vector location of the kth classification grid position, Δ is a distance threshold, and y_k is the soft-labeled true target corresponding to the classification grid positions, $\mathbf{c}_k$. To generate $\mathbf{c}_k$, a D-dimensional grid of positions is created that correspond to positions in the real world.

Our data are 2D in nature with variations in only latitude and longitude; thus, $D = 2$. To simplify calculations, the distance between adjacent classes—i.e., grid positions $\mathbf{c}_k$—is normalized to be 1. To ensure that only adjacent classes in $\mathbf{c}_k$ to any given ground truth location $\mathbf{p}$ are non-zero-valued, we chose $\Delta < 1$. For example, in Figure 3.4, the green circles would be the only elements of $\mathbf{y}$ that are non-zero-valued.

Figure 3.4 shows a position, $\mathbf{p}$, among the 4 closest grid points, $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$, and $\mathbf{c}_4$. The associated soft label y_k for each of these grid locations is inversely proportional to the distance from class location $\mathbf{c}_k$ and $\mathbf{p}$, described in Equation (3.16). As such, the upper-right truth label y_2 of the 4 classes in Figure 3.4 has the smallest soft label, and the lower-left truth label y_3 has the highest value.

If a position, $\mathbf{p}$, is equidistant to all surrounding classes, the non-zero values of $\mathbf{y}$ are

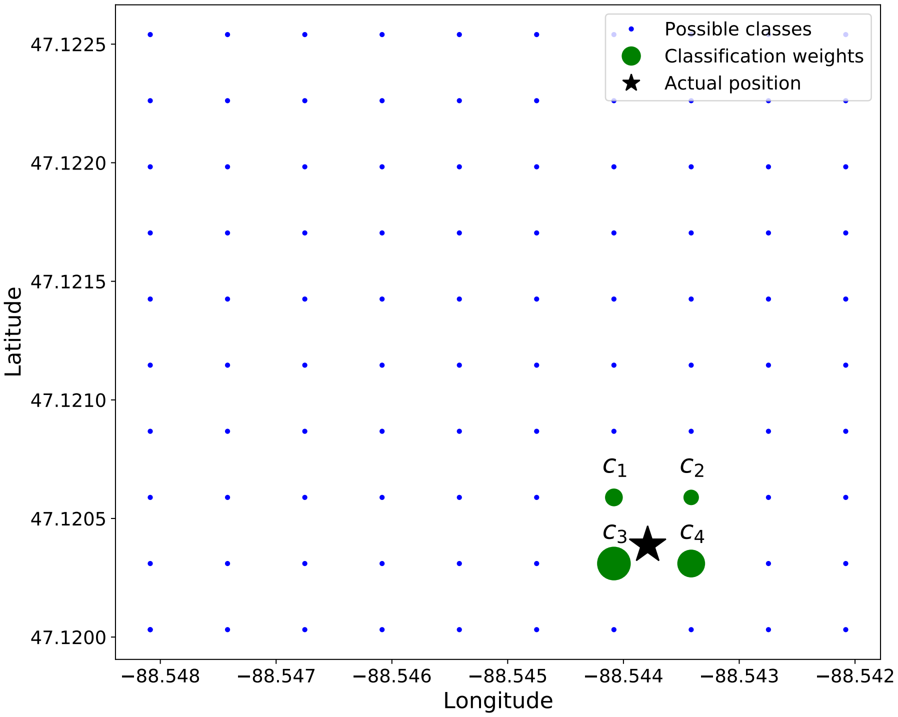


Figure 3.4: Soft classification of linear position where $\Delta = 1$. The star is the original position, and the circle size corresponds to the weight of each value.

all equal. Additionally, suppose the ground truth position, $\mathbf{p}$, is positioned directly on a class, $\mathbf{c}_k$, then

$$y_k = \begin{cases} 1 & \mathbf{p} = \mathbf{c}_k \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

When converting back to a continuous location space, each classification grid is defined on specific coordinates; thus, we can yield the original position,

$$\mathbf{p} = \sum_{k=1}^{N} y_k \mathbf{g}_k \tag{3.18}$$

where **g** corresponds to the "real-world" grid mapping to the classification locations, **y**. For example, **g** can be a grid of GPS coordinates or a grid of pixel positions in an image.

Soft classification is also useful when the truth data are uncertain (e.g., a distribution) as opposed to classifying a single class for the truth data. For the purposes of this paper, the errors in the truth data and their distribution are not considered because the uncertainties of our truth data (within 2.5 m [59]) are smaller than the the distances between each class (28 m), i.e., there are no benefits to adding uncertainty when localizing our target.

When calculating Equation (3.16) for our target positions, we may find that the locations are constrained to smaller regions of the full rectangular grid; thus, the grid can be adjusted such that only certain locations are used. The dimensionality of the prediction can be reduced by removing classes—i.e., grid locations. For example, these removed locations can materialize if there are physical obstructions at those locations. Additionally, we observed that background noise often will manifest as position estimates that are outside the region of interest (i.e., the water body). In the future, we will look at how we can specifically design our algorithms to identify background noise when no source to track is present, but for this study, we simply constrained the classification grid to within the banks of the region of study (a canal), where Figure 3.5 shows the regions outside the banks. Because our experiments are

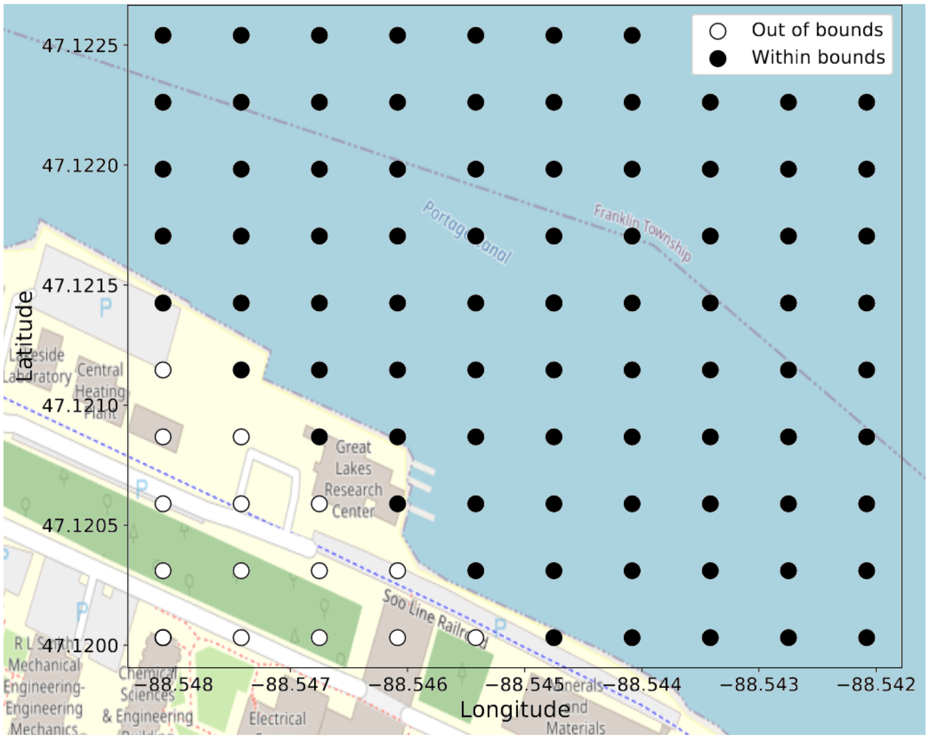


Figure 3.5: Eighty-five classes, out of the possible 100, for a 10 × 10 grid where the training and test data are not present in any of the "out of bounds" labels.

simulating environments of a ship in the water or a snowmobile traveling across the ice, we can constrain the classification grids to regions where the acoustic sources can only reach physically. These constrains are a benefit to the classification approach to localization, but further constraints could bias our results to the data.

An example of the grid location classes for a 10 × 10 grid is shown in Figure 3.5. The "out of bounds" labels on the bottom-left corner in Figure 3.5 correspond to outside the banks of the Keweenaw Waterway, and no data are present on these grid location classes.

When the classification labels are represented as soft-labeled grid locations, we can use an MSE loss between each dimension,

$$L = \frac{1}{K} \sum_{k=1}^{K} \left(y_k - \hat{y}_k \right)^2 , \tag{3.19}$$

where K corresponds to the number of classes, $\mathbf{y}$ is the true (soft) classification label vector, and $\hat{\mathbf{y}}$ is the predicted (soft) classification label vector.

The weakness of classifying with grid representation and the categorical loss in Equation (3.19) is their ordinal (spatial) nature is not fully considered. If the network were to predict an incorrect location physically close to the true location, this should not be equally penalized to predicting a location far away from the true location. Categorical classification fails to represent this; hence, we describe how to extend this idea to ordinal classification for localization.

3.2.6.3 Ordinal Classification

To give an example of the impetus for the ordinal (spatial) property of the classification grid, consider a prediction at position $(0, 1)$ when the correct class is at position $(0, 0)$; clearly, this incorrect prediction is not as poor as predicting at the position $(99, 99)$. Categorical loss would consider these two incorrect predictions to be equally poor, but our proposed ordinal loss properly represents the relative error of each of these predictions. Extending ordinality to the classification problem introduces complexity, as the loss function becomes more advanced, but this added complexity better represents our localization problem [60]. Our proposed ordinal loss function gives lower weight to closer predictions to the truth [12],

$$L = \frac{1}{K}\mathbf{y}^T W(\mathbf{y} - \hat{\mathbf{y}})^2, \tag{3.20}$$

where W is a weighting matrix and $(\cdot)^2$ indicates an elementwise square operation in this equation. The weighting matrix, W, $W_{i,j} = \|\mathbf{c}_i - \mathbf{c}_j\|_2$, is a $K \times K$ matrix of the pairwise l_2-norm distances between each grid position $\mathbf{c}$. One can think of the product $\mathbf{y}^T W$ as the weighted mean distance of each grid location to the predicted location represented by $\mathbf{y}$. This is then multiplied by the vector that represents the squared differences between the predicted location $\mathbf{y}$ and the truth $\hat{\mathbf{y}}$. Consider the following example.

Consider a 2×2 grid of locations, where $\mathbf{c}_k$, $k = 1, 2, 3, 4$, represent the grid positions $[(0, 0), (0, 1), (1, 0), (1, 1)]$. In this scenario, the weight matrix is

$$W = \begin{bmatrix} 0 & 1 & 1 & \sqrt{2} \\ 1 & 0 & \sqrt{2} & 1 \\ 1 & \sqrt{2} & 0 & 1 \\ \sqrt{2} & 1 & 1 & 0 \end{bmatrix}.$$

Suppose $\mathbf{y} = [0,1,0,0]^T$ (representing a prediction at position $(0,1)$) and $\hat{\mathbf{y}} = [0,0,1,0]^T$ (representing the ground truth position $(1,0)$). The product $\mathbf{y}^T W = [1,0,\sqrt{2},1]$, and the loss is

$$
\begin{aligned}
L &= \frac{1}{4}[1,0,\sqrt{2},1]([0,1,-1,0]^T)^2 \\
&= \frac{1}{4}[1,0,\sqrt{2},1][0,1,1,0]^T \\
&= \frac{\sqrt{2}}{4} \approx 0.35.
\end{aligned}
$$

Now, suppose $\mathbf{y} = \left[\frac{1}{2},\frac{1}{2},0,0\right]^T$ (representing a prediction at position $\left(0,\frac{1}{2}\right)$) and $\hat{\mathbf{y}} = \left[\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}\right]^T$ (representing the ground truth position $\left(\frac{1}{2},\frac{1}{2}\right)$). Clearly, the prediction in this example is better than the previous example. The product $\mathbf{y}^T W = \left[\frac{1}{2},\frac{1}{2},\frac{(1+\sqrt{2})}{2},\frac{(1+\sqrt{2})}{2}\right]$, and the loss is

$$
\begin{aligned}
L &= \frac{1}{4}\left[\frac{1}{2},\frac{1}{2},\frac{(1+\sqrt{2})}{2},\frac{(1+\sqrt{2})}{2}\right]\left(\left[\frac{1}{4},\frac{1}{4},-\frac{1}{4},-\frac{1}{4}\right]^T\right)^2 \\
&= \frac{1}{4}\left[\frac{1}{2},\frac{1}{2},\frac{(1+\sqrt{2})}{2},\frac{(1+\sqrt{2})}{2}\right]\left[\frac{1}{16},\frac{1}{16},\frac{1}{16},\frac{1}{16}\right]^T \\
&= \frac{2+\sqrt{2}}{64} \approx 0.05.
\end{aligned}
$$

As expected, the loss in the second example is less than that of the first example.

3.2.7 Experiments

Eight experiments were conducted between February 17 and 20, 2021, on the Keweenaw Waterway near Michigan Technological University. Figure 3.6 shows the experimental setup. The Keweenaw Waterway is a narrow and shallow channel of water (a canal), which causes many multipath reflections and scattering. The ice was between 0.4 and 0.5 m thick, and the water was between 6 and 8 m deep. The first three experiments (one on February 17 and two on 18) had snow above the ice, insulating the ice, which caused an uneven, thin layer of slush. By February 19th, high winds had removed the snow, and the surface ice hardened again, so the remaining five experiments were conducted in a hard ice environment.

A snowmobile drove back and forth in front of our sensors to represent a moving acoustic source. A handheld GPS on the snowmobile kept track of the position of the snowmobiles. The two AVSs passively recorded the noise from the snowmobile, which included engine intake and exhaust, as well as track–ice structural–acoustic interaction, for the purpose of localization.

After the data were synchronized, trimmed, and labeled, a total of roughly 3.2 h—$11,526$ s—of snowmobile acoustic data were recorded on the two AVSs. The position of these AVSs were kept constant, 30 m apart, on either end of the dock next to the

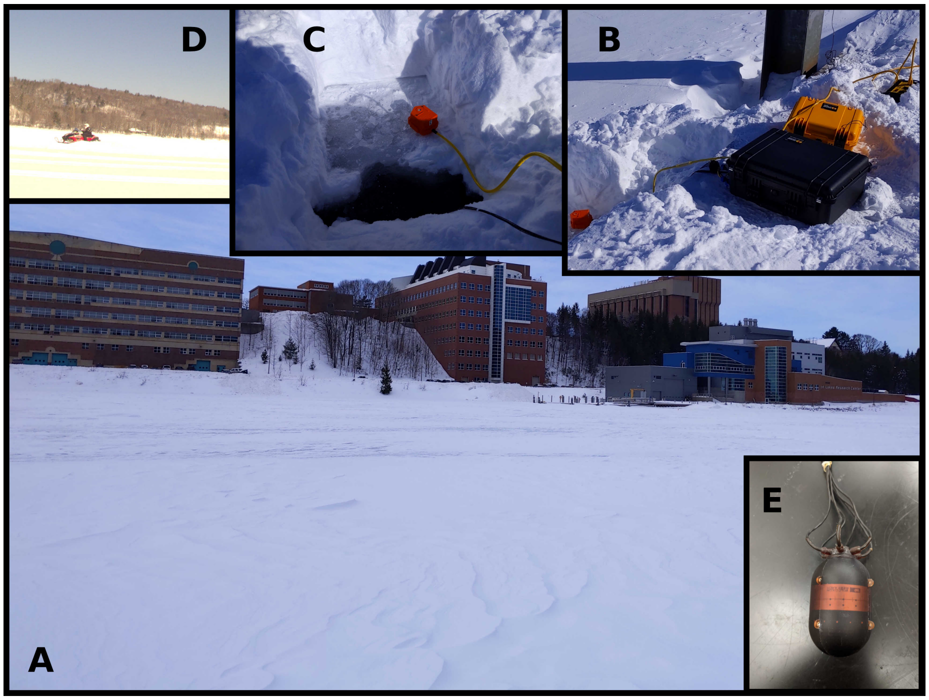


Figure 3.6: Conditions under which the experiments were conducted: (**A**) shows the Keweenaw Waterway frozen over, looking SSW at Michigan Technological University; (**B**) shows the sensors and data acquisition system on a dock near the Great Lakes Research Center; (**C**) shows a close-up of where the sensors are deployed in the water; (**D**) shows a snowmobile driving in one of the experiments; (**E**) is a close up of the AVS.

Great Lakes Research Center.

3.2.7.1 Data Explanation

The acoustic data were recorded in time-series at a sample rate of 17, 067 Hz using a National Instruments cRIO-9035 with NI-9234 data acquisition cards. The sample rate was set to 17, 067 Hz since the sensor's 3 dB cutoff frequency was at 8000 Hz; thus, frequencies above 8000 Hz were not used in post-processing. The data were transformed into an azigram using Equations (3.1)–(3.3). The STFT used a Hanning

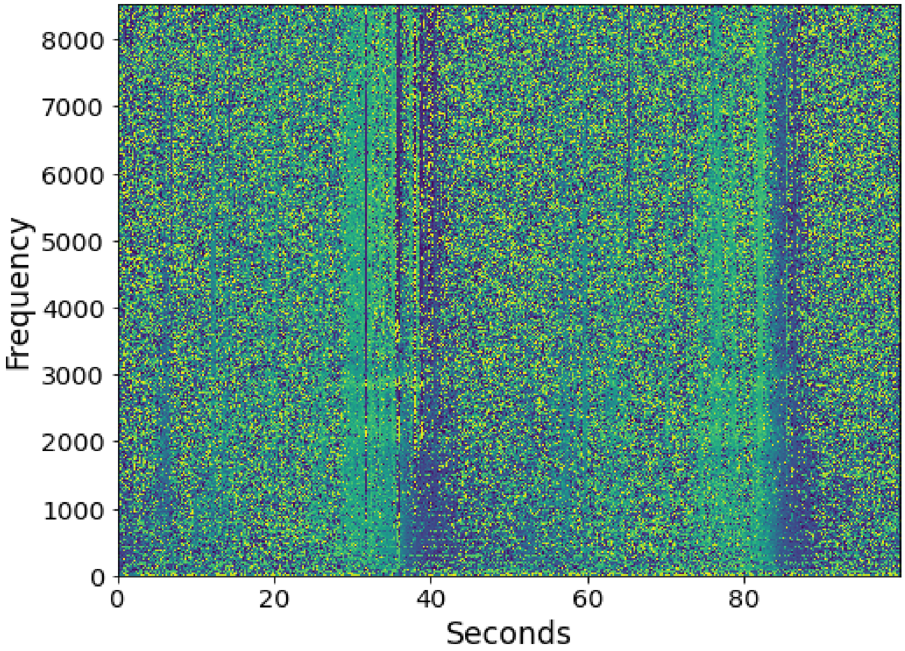


Figure 3.7: Azigram response from a single AVS of a snowmobile driving past the AVS at roughly 40 and 85 s.

window, 50% overlap, and a segment size of 1706 samples to yield a time step of 0.05 s. Figure 3.7 shows an example of the azigram of the first 100 s of data. Note there are two snowmobile passes in the azigram, around the 40 and 85 s marks. The snowmobile drives by the sensor around the 40 s mark (heading eastwardly), turns around, then drives by the sensor again near the 85-second mark (heading eastwardly again).

The truth data, being GPS data, were recorded at 1 Hz using a handheld GPS receiver. Figure 3.8 shows the GPS data through all the experiments. The GPS data were then linearly interpolated, resulting in an upsampling of 20 times, to match the sample rate of the azigram data.

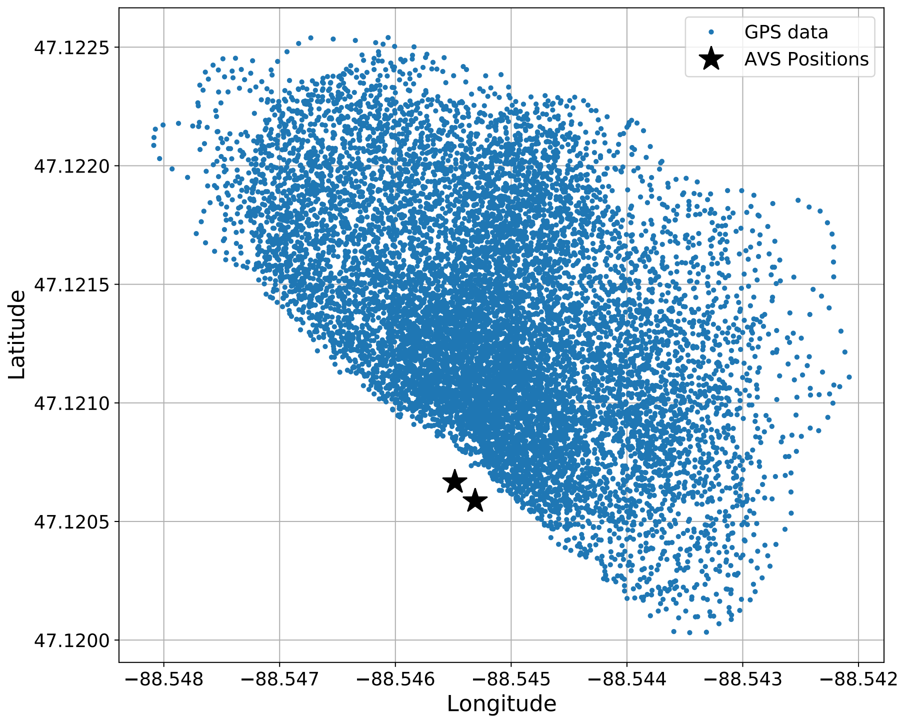


Figure 3.8: Bird's eye view of the total amount of GPS data in all datasets before 20-times interpolated. GPS data are accumulated from 8 experiments. The two AVS positions are shown for a reference.

To prepare the data for input to the neural network, the azigram was linearly normalized from its $[-\pi, \pi]$ range to $[0, 1]$ and the GPS data were linearly normalized with the total maximum and minimum latitude and longitudes set to the interval $[0, 1]$: latitude was normalized from $[47.1200°, 47.1225°]$ to $[0, 1]$ and longitude from $[-88.548°, -88.542°]$ to $[0, 1]$. For classification networks, the GPS data were processed with Equation (3.16) with $k = 100$ to represent a 10×10 grid of latitude and longitude position.

3.2.8 Network Explanations

The neural networks process the same data, i.e., the data are pre-processed in the same manner for every neural network. The azigram data are shaped to use the prior 512 time steps for a single prediction. Each AVS's azigram frequencies are downsampled to contain 256 frequency bins. The two AVS's frequency data are concatenated along the frequencies; hence, the input data are a 512×512 azigram in the neural network. Each network predicts a single output value, $\mathbf{y}$, at the final time step of the 512×512 sample. In other words, the networks' input data are a sliding window of 512 samples, and each network predicts the new location at the end of the 512 window, then the window is moved forward by 1 sample from a time window of $[n, n + 512]$ to $[n + 1, n + 513]$.

We compared four large neural networks and four small neural networks. The small neural networks are demonstrated as a simpler method in localizing an acoustic source; less training time, less training data collection, and less calculation time are required for "small" networks. Because our dataset is very large, we also explored large neural networks, though this may not be practical for situations where data collection is difficult or impossible to achieve due to budget limitations, lack of available data, or time limitations in labeling, or the environment is not complex enough to require such a large network.

Table 3.1
Depths of the backbone for each type of network shown in Figures 3.9 and 3.10.

	CNN	**LSTM**	**Transformer**	**ViT**
Large	16	5	8	12
Small	4	1	1	8

Table 3.2
Total trainable parameters for each neural network architecture.

	CNN	**LSTM**	**Transformer**	**ViT**
Large	$23,849$ k	$13,825$ k	$16,911$ k	$85,846$ k
Small	892 k	905 k	843 k	928 k

The four large networks are the following: a ResNet50 [57] CNN-based network, an LSTM-based network, a Transformer-based network [48], and a ViT-based network [46]. The four small networks have an arbitrary requirement to contain less than 1 million parameters to give a fair comparison. Figures 3.9 and 3.10 show a comparison of each of the networks. The difference between a “small” and “large” network is adjusted in the architectures by the $\times N$ value in both Figures 3.9 and 3.10, i.e., N is smaller in small networks. Table 3.1 shows the number of layers N for each of the neural networks, and Table 3.2 contains the number of trainable parameters for each neural network.

The categorical classification neural networks predict a probability of each grid location class. This classification network predicts its results in a softmax activation function—Equation (3.13)—to assert a probability output. The benefit of the categorical classification neural network is its opportunity to add uncertainty to its prediction.

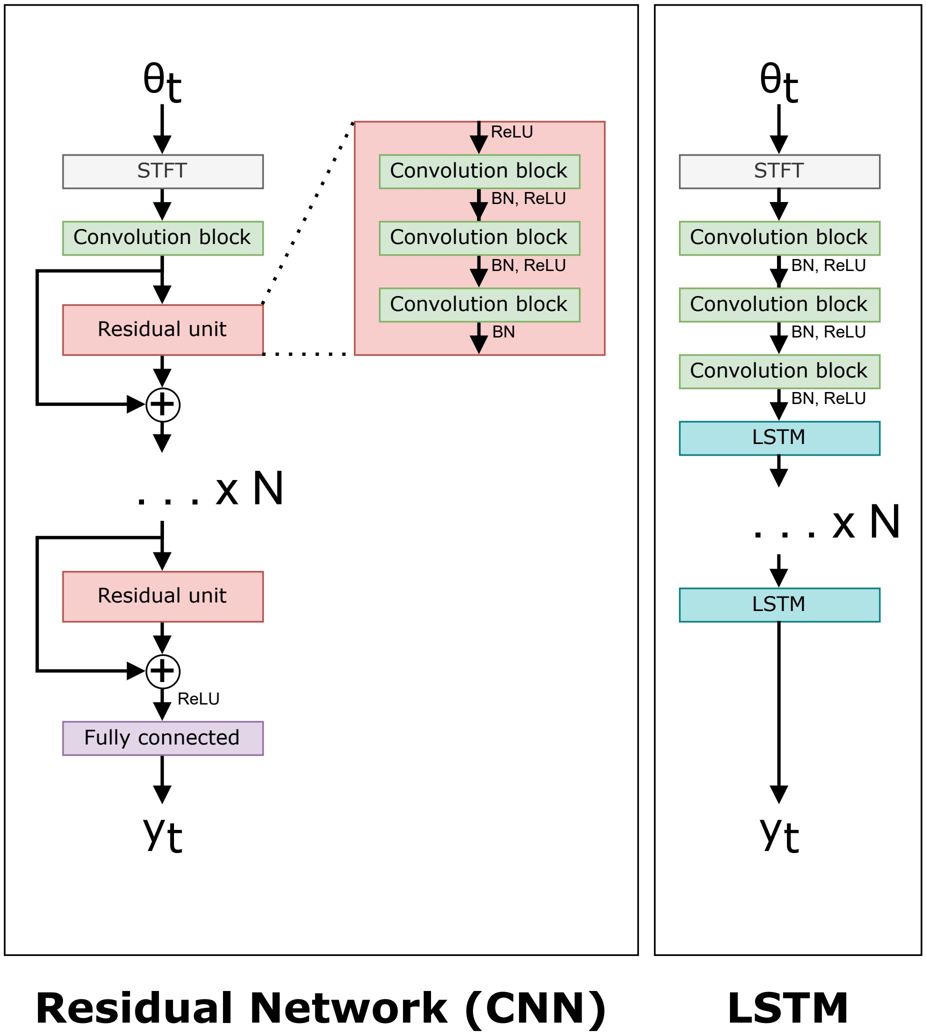


Figure 3.9: Network architectures for the CNN (left) and LSTM (right).

The ordinal classification neural network predicts exactly the same type of output as the categorical classification network, but rather than using the mean-squared error loss function in Equation (3.19), the network uses the ordinal loss function in Equation (3.20).

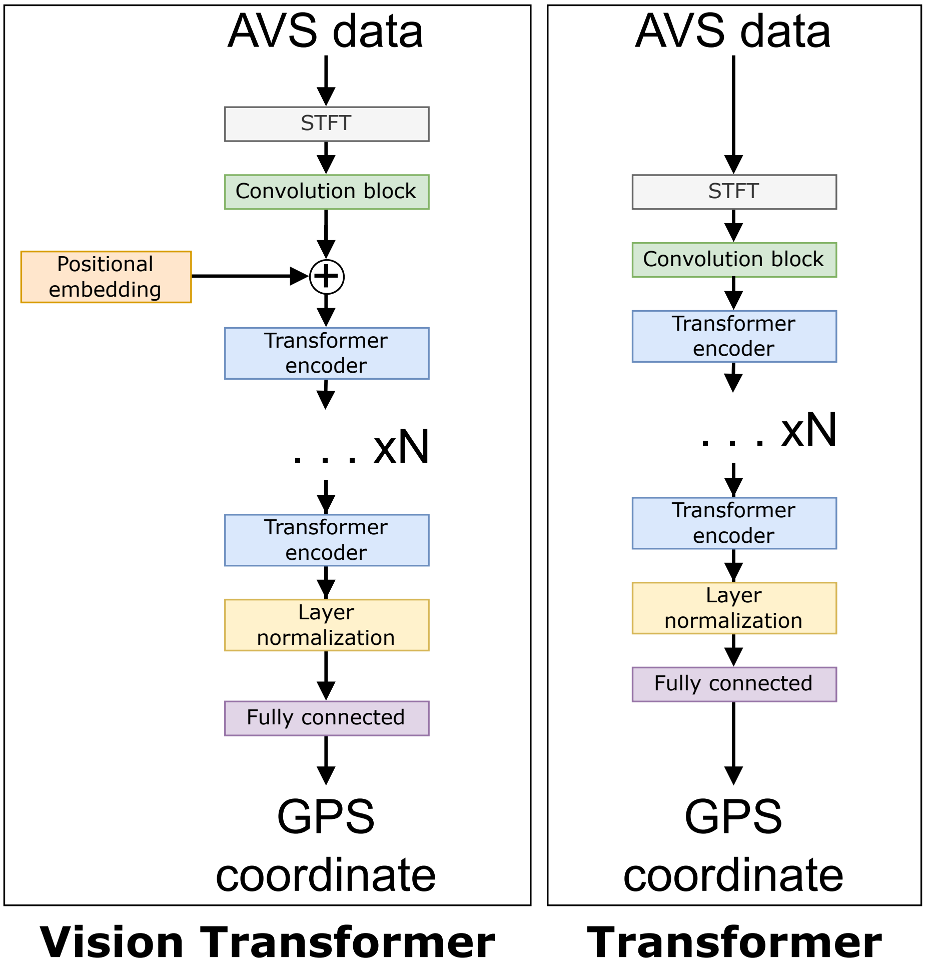


Figure 3.10: Network architectures for the ViT (left) and Transformer (right).

3.2.9 Training and Hyperparameters

Each network used the Adam optimizer [61] with a learning rate of 0.0001 and parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We batched 32 samples of size 512 × 512 in a single backwards propagation step. Each batch had its data randomized except for the LSTM, where batches were sequential to support the LSTM's long-term memory.

Seven of the 8 experiments were used as training data, consisting of a total of 189.868 samples, i.e., roughly 2.6 h of data. Ten percent of the training data were used to validate the model, i.e., 18.987 samples. The model weights with the lowest loss using this validation set were then tested on the test data, which we can now show.

3.3 Results

The data on which we tested our algorithms consisted of an experiment where a single snowmobile moved by the sensors back and forth on February 17. There were 39.628 samples, i.e., 1.981 seconds, and no neural network was trained on any data from this day to isolate the training and test data.

The neural networks were programmed in Python using the Tensorflow backend and Keras frontend to create these models [62, 63]. The networks were trained using an NVIDIA GeForce RTX 3090.

The accuracies of each neural network and their respective loss functions are shown in Table 3.3. Notice the ViT has almost over a 10-fold increase in accuracy. When comparing the two sizes of networks, the training times for each are telling, tabulated in Table 3.4. The timing differences between each model are significantly different, except for the large and small CNN models.

Table 3.3
Neural network results on test data from February 17. The results indicate the mean distance in meters between the predicted results by the neural network and the recorded results by the GPS. A $\pm 1\sigma$ deviation is shown.

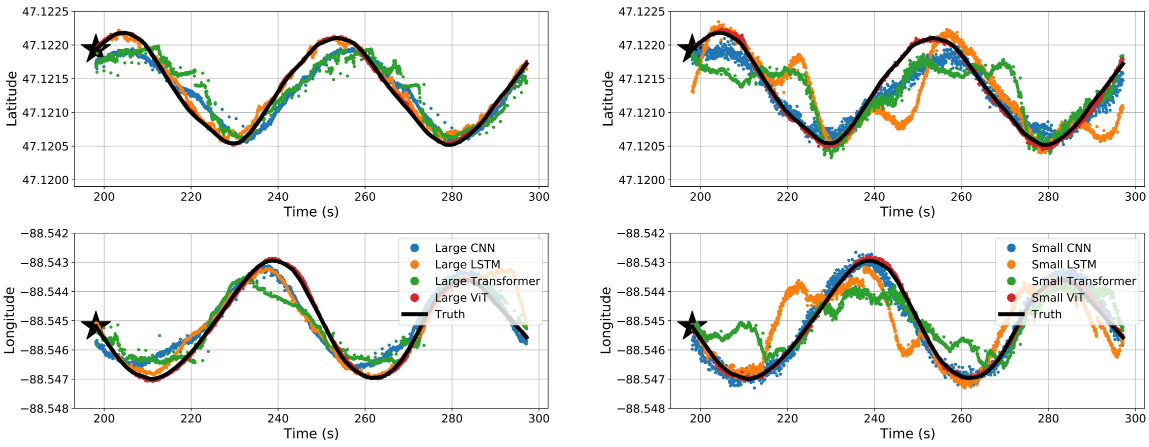
	CNN		LSTM	
	Large	**Small**	**Large**	**Small**
Regression	39.3± 29.1	27.1± 21.7	44.2± 53.9	58.7± 62.7
Categorical	26.7± 57.3	28.4± 27.0	49.4± 47.8	41.9± 40.0
Ordinal	21.4± 31.2	33.1± 35.2	64.6± 49.2	68.2± 54.3
	Transformer		**ViT**	
	Large	**Small**	**Large**	**Small**
Regression	42.1± 30.3	65.7± 48.8	4.9± 3.7	5.9± 4.8
Categorical	49.8± 39.3	44.5± 45.1	3.1± 2.5	3.7± 3.0
Ordinal	53.9± 45.0	44.5± 45.1	2.9± 2.5	6.7± 6.1

Table 3.4
Neural network mean training times per epoch.

	CNN		LSTM	
	Large	**Small**	**Large**	**Small**
Regression	671 s	654 s	2,358 s	1,931 s
Categorical	620 s	657 s	2,170 s	1,933 s
Ordinal	675 s	656 s	2,150 s	1,930 s
	Transformer		**ViT**	
	Large	**Small**	**Large**	**Small**
Regression	1,358 s	639 s	1,700 s	654 s
Categorical	1,188 s	648 s	1,785 s	658 s
Ordinal	1,070 s	647 s	1,752 s	660 s

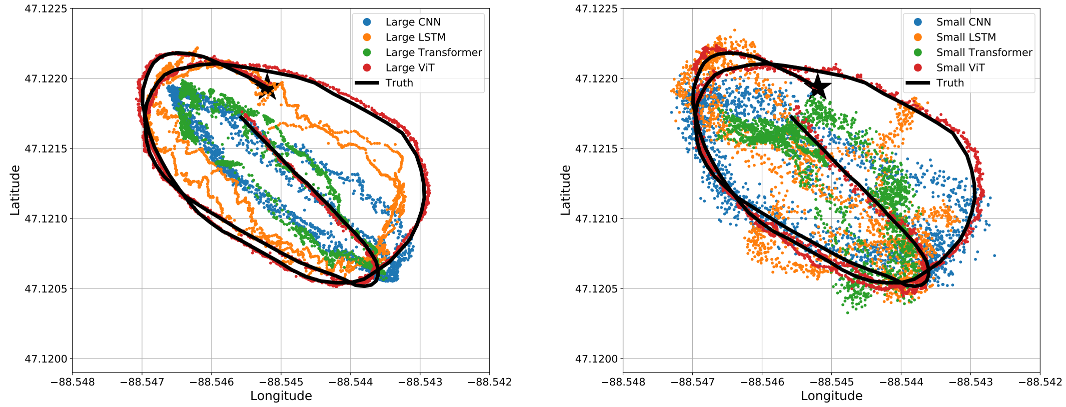
The results may be misunderstood simply reading Tables 3.3 and 3.4. For a visual representation of our data, we will start off with the predicted coordinates for what the MSE actually represents. Figure 3.11 shows a section of a time-series representation of the data using both the latitude and longitude positions of the snowmobile and each algorithms' predicted positions of the snowmobile. Figure 3.11 can be misleading where one may see the CNN and Transformer networks seem to be on-par with, or close to, the results of the ViT. Mapping these results to an $-x, -y$ plane, Figure 3.12 shows a more critical view for what these small amounts of errors indicate. Even with a top-down view of the experiment, the ViT tracks the snowmobile at high accuracy in comparison to the other models. For the ViT, it should be noted that its mean accuracy is 2.9 m. This is very close to the accuracy of our GPS receiver: the reported 95th-percentile mean error is 2.545 m in Minneapolis, Minnesota, which is relatively near Houghton, Michigan, from January 1st to March 31st, 2021 [59], and the GPS was recorded in a relatively open area. Therefore, the ViT appears to have reached the maximum achievable accuracy of our experimental truth data. That is, our truth data are not accurate enough to verify errors significantly better than 2.9 m. These significant results are further discussed in Section 3.4. Almost all of the test data are similar to Figures 3.11 and 3.12, shown in Appendix A Figures A.1—A.7.

Although most test data are similar, there exists a section of the test data where a snowmobile idles (does not move) for 25 s, and the networks perform relatively poorly with these data. Figure 3.13 shows the predicted locations from each network at the time where the snowmobile is idling (not moving) in a bird's eye view. The Transformer, CNN, and LSTM networks all struggle to notice when the snowmobile is idle. Those three neural networks were not able to notice the stationary source



(**a**) (**b**)

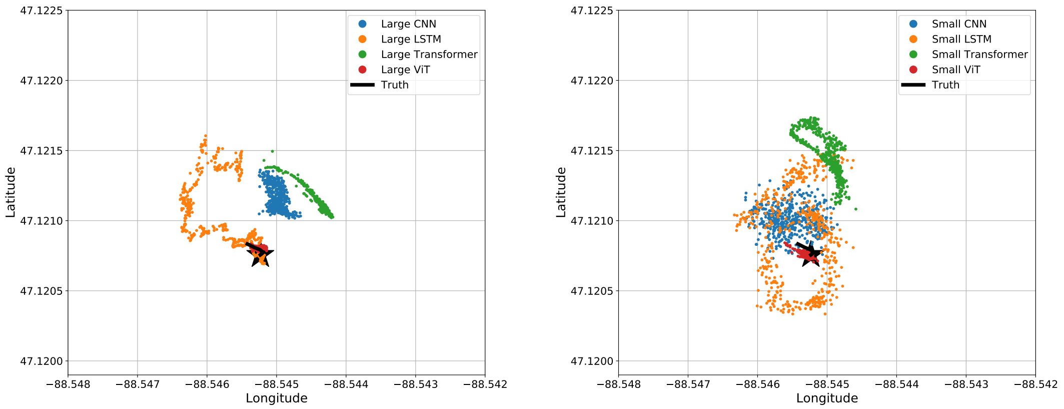
Figure 3.11: Time-series split predicted results for the four different (**a**) large regression algorithms and (**b**) small regression algorithms.



(**a**) (**b**)

Figure 3.12: Bird's eye view of results for the four different (**a**) large regression algorithms and (**b**) small regression algorithms. The same data and predictions from Figure 3.11 are shown.

and continued to predict movement. Note that the LSTM seems to follow a circular pattern, which indicates the network is anticipating the snowmobile to drive in this pattern.



(a) (b)

Figure 3.13: Bird's eye view of results for the four different (**a**) large regression algorithms and (**b**) small regression algorithms when the acoustic source is stationary for 25 s.

3.4 Discussion

The core structure behind the network architectures described in Section 3.2.2–3.2.5 is indicative of the results shown in Sections 3.3. Similar to our results, ViTs have shown excellent results in image classification [46]. What may be surprising or not intuitive is the magnitude by which the ViT performance surpassed all other models, most surprisingly the similarly structured Transformer. Each neural network tracks the general trend of the snowmobile position, while the ViT tracks the positions almost perfectly. To explain this, the Transformer determines attention for each input sample individually, and the ViT attends to subsections of the input data. The input data have 512 samples of dimension 512, and the Transformer attends each time step to all the other time steps. This produces a significant amount of attention solely within the

Transformer block. On the other hand, the ViT positionally embeds the 512 samples so that attention is made in a temporal and frequency connection. The embeddings also reduce the attention matrix with the size of patches per Transformer network. These reduced attention matrices allow for a “deeper” model with the ViT network.

The original Transformer well describes *Natural Language Processing* (NLP) [48] with its connection between word embeddings, but this does not transfer well to spectrum analysis. The Transformer network does not allow for attention along the time and frequency domain, which is addressed in modified Transformer papers [64]. Specifically, the ViT is a type of modified Transformer, which, in our example, embeds the 512 attention parameters into 64 smaller regions of interest. These areas are trainable, and the embedded positions allow the ViT to attend to time and frequency patterns rather than solely time. Additionally, the embedded positions yield a smaller number of positions to which the ViT attends. This embedding helps scale the ViT to attend to higher-dimensional data with the lower amount of attention values. Even with this explanation, the significant increase in accuracy exhibited by the ViT is remarkable. Similar behavior in results is summarized in surveys of ViTs [50, 65]; notably, the results in [66] show major improvements in image classification accuracy using the ViT.

The ordinal classification approach for localization is not a panacea for every localization problem. The ordinal classification approach shows improved results in our experiments and gives way for soft-labeled truth data when the truth data are not absolutely accurate. The network is capable of predicting with low confidence, although our training data do not facilitate the networks utilizing this yet. An important question arises: Why are some networks better than others with different loss functions? We believe that the LSTM and Transformer networks are most suited for regression, as LSTMs were constructed for time-series data [37] and Transformers were developed for NLP [48]. The opposite is true for the CNNs and ViTs. The CNN and ViT architectures are suited for classifying images; hence, it is understandable why the CNN and ViT performs better for the classification approaches we propose for localization.

The results for the large neural networks are impressive, but it may be unacceptable or impractical to use such large networks in real-world scenarios. For example, in an remote embedded system, using an 85 million parameter network such as the large ViT would be wasteful with power consumption to calculate all the operations. Additionally, the number of data points required to train such large neural networks costs an extraordinary amount of time and effort to produce, as well as using an expensive GPU to train the network. In contrast, the small neural network results are a more practical view for the number of parameters to be used in a real-world scenario. Therefore, it is important to look at the loss in accuracy as a friendlier, real-world use case with smaller networks.

3.5 Conclusions

In this paper, we developed different neural networks for processing data from a pair of underwater AVSs. The sensors recorded a moving anthropogenic acoustic source, and the data were analyzed using different neural networks to estimate the location of the target. Each network—the CNN, LSTM, Transformer, and ViT—all tracked the position relatively well, but when comparing the networks, we found that the ViT predicted source location with excellent accuracy, an order of magnitude more accuracy. The ViT was able to analyze our highly dimensional data and track the acoustic source well. Additionally, the networks were reduced to have a smaller number of parameters in order to compare the loss in accuracy.

Finally, we studied three approaches to localizing a moving target. A regression loss function was the baseline method to compare with our non-conventional methods: a categorical classification and ordinal classification approach for localization. We showed that the ordinal classification approaches performed better for networks better suited for classification, being the CNN and ViT. The regression loss function performed better for the networks better suited for time-series data, being the LSTM and Transformer.

Chapter 4

Using Vision Transformers for classification of through-ice acoustic sources

This chapter is a reprint of the *proceedings of meetings in acoustics* (POMA). The permission for reprint has been given in Appendix B.2. Reproduced from "Using Vision Transformers for classification of through-ice acoustic sources." *Proceedings of Meetings on Acoustics*, Denver, CO, June 2022, with the permission of AIP Publishing [15]. Copyright 2022, Acoustic Society of America.

4.1 Introduction

Classifying an acoustic source is important in Arctic underwater environments for situational awareness. Whether classifying an on-ice anthropogenic source, such as a snowmobile or an ice pick, or an under-ice biologic source, such as a seal or a whale, the algorithms are useful in academic, civilian, and military applications. In water and ice environments, acoustic signatures propagate at high distances and, thus, acoustics is the main method for detecting and classifying foreign sources [67]. Passive sonar is a method where no active ping is transmitted and so background noise is a very concerning matter when handling source classification [68]. Recently, machine learning has been employed to classify sources in high noise environments, described further in a survey by Domingos et al. [69] Analysis has been previously been done with underwater acoustics classification and machine learning [14, 68, 70, 71, 72]. Choi et al. and Cinelli et al. analyzed different machine learning techniques, two of which used a *fully connected neural network* (FCNN) architecture and a *convolutional neural network* (CNN) architecture to classify different vessels [70, 71]. Our research extends these analyses to a new architecture, the *Vision Transformer* (ViT) [46]. The ViT claims to outperform the LSTM and CNN in speed and performance, and in this paper, we validate this claim in this new field of through-ice acoustics.

Our research looks into analyzing different experiments on the ice of the Keweenaw

Waterway in Houghton, Michigan. These experiments generated a multitude of anthropogenic sources and we classified these real-world acoustic signals using machine learning. The experiments were conducted with varying types of classes: snowmobile noise, ice augers, ice saws, hammers, and an underwater speaker playing a chirp signal. The acoustic signatures of each class were passively recorded with a *pa*-type (pressure-acceleration) *acoustic vector sensor* (AVS). The *pa*-type AVS recorded both acoustic pressure and acceleration simultaneously with a piezoelectric transducer and a triaxial accelerometer respectively [53]. Using both of these sensors inside a single AVS, we could estimate acoustic intensity with post-processing [52]. We have previously found success in post-processing the AVS data to acoustic intensity before analyzing the data with *machine learning* (ML) [12, 14].

After labeling our data for a classification data set, we adopted supervised ML to map a transform from AVS spectrogram data to a predicted class. The supervised ML method used was a neural network, and the core of the neural network architecture used was a ViT [46], a modification of the Transformer architecture [48]. To our knowledge, ViTs have not been used in underwater acoustic classification, let alone through-ice acoustic classification. In this paper, we compare different neural network architectures in through-ice acoustic classification, namely the CNN architecture and the *long short-term memory* (LSTM) network architecture. Our analysis in Section 4.4 has shown that the ViT architecture outperforms the CNN architecture and the LSTM network architecture. Before analyzing the results, we will describe the analysis and experiments conducted.

4.2 Algorithmic Methods

There are two processes implemented within this analysis. The first step is to process the acoustic data into a spectrogram. A spectrogram initially extracts frequency-specific features, which are present in our data [73]. The second step is to input the processed acoustic data into the neural network architecture, producing a prediction of the source location.

4.2.1 Acoustic processing

The data from our experiments were recorded using an underwater *pa*-type AVS—the Meggitt VS-209. The AVS records four time-series data streams: $x-$, $y-$, $z-$axis particle acceleration data streams, denoted as $a_x(t), a_y(t), a_z(t)$; and a particle pressure data stream, $p(t)$. The data are transformed into the frequency domain using a *short-time Fourier transform* (STFT);

$$A_x(f) = \mathcal{F}\left(a_x(t)\right), \tag{4.1a}$$

$$A_y(f) = \mathcal{F}\left(a_y(t)\right), \tag{4.1b}$$

$$A_z(f) = \mathcal{F}\left(a_z(t)\right), \tag{4.1c}$$

$$P(f) = \mathcal{F}\left(p(t)\right), \tag{4.1d}$$

where $\mathcal{F}(\cdot)$ is the Fourier operator. Acoustic intensity is calculated using the combination of both the particle acceleration and pressure, where

$$I_x(f) = P(f)A_x^*(f)/(2\pi f), \tag{4.2a}$$

$$I_y(f) = P(f)A_y^*(f)/(2\pi f), \tag{4.2b}$$

$$I_z(f) = P(f)A_z^*(f)/(2\pi f), \tag{4.2c}$$

and $(\cdot)^*$ is the complex conjugate. The acoustic intensity is then transformed into polar coordinates by

$$|I(f)| = \sqrt{I_x^2(f) + I_y^2(f) + I_z^2(f)}, \tag{4.3a}$$

$$\Theta(f) = \arctan\frac{I_x(f)}{I_y(f)}, \tag{4.3b}$$

$$\Phi(f) = \arctan\frac{I_x(f)}{\sqrt{I_y^2(f) + I_z^2(f)}}, \tag{4.3c}$$

where $|I(f)|$, $\Theta(f)$, and $\Phi(f)$ are the magnitude, azimuth, and elevation, respectively, of the acoustic intensity. For classification, $|I(f)|$ is processed by the neural networks. Because we use $|I(f)|$ as the input to our neural network, we denote $|I(f)| = X$ for the remainder of this paper.

4.2.2 Neural network processing

A neural network requires a consistently sized input, X, and many of our events do not occur at the same time duration. For example, the hammer strike class is a transient that ends in under a second, and the snowmobile class is a signal that lasts for 30 minutes. The difference between these scales can cause issues when directly passing the data into the neural networks. In our case, we set X to be a constant size of N time steps, where N is 200 time steps—1 second—for the experiments presented in Section 4.4. If the event occurs in fewer than N time steps, then the signal is zero-padded. If the event occurs in more than N time steps, we first trim the signal to equal N time steps, then we use the remaining signal as another event, repeating for very long signals. To make the neural network more robust, when zero-padding is used, the zero-padding is added randomly both before and after the signal clip, placing the short clip somewhere in the middle of the N time steps. The randomized offset prevents the neural network from failing to predict only when the class only occurs at the beginning of X. This allows the neural network to generalize any position of the transient in X.

Suppose there exists an event $C \in \mathbb{R}^{F \times T}$ where F is the number of frequency bins, T is the number of time steps, and $T < N$, then

$$X = \begin{bmatrix} \mathbf{0}^{F \times N_l} & C & \mathbf{0}^{F \times N_r} \end{bmatrix}, \tag{4.4}$$

where $X \in \mathbb{R}^{F \times N}$, $\mathbf{0}^{F \times M}$ is a matrix of zeros of size $F \times M$, and N_l and N_r are randomly drawn such that $N = T + N_l + N_r$.

Now, suppose there exists an event $C \in \mathbb{R}^{F \times T}$ with $T > N$, then

$$X_1 = \begin{bmatrix} C_1^{F \times N} \end{bmatrix} \tag{4.5a}$$

$$X_2 = \begin{bmatrix} C_2^{F \times N} \end{bmatrix} \tag{4.5b}$$

$$\vdots$$

$$X_M = \begin{bmatrix} \mathbf{0}^{F \times N_l} & C_M^{F \times K} & \mathbf{0}^{F \times N_r} \end{bmatrix} \tag{4.5c}$$

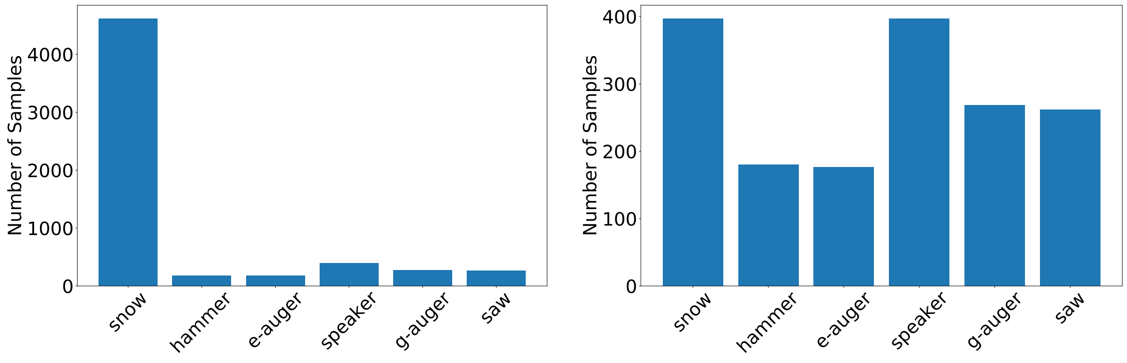
where C_i spans the columns of C from $(i-1)N$ to iN; $X_1, X_2, ...X_M$ are the “trimmed” events; $K = T - N(M-1)$; $M = T/N$ is rounded up to the nearest integer; and N_l and N_r are randomly drawn such that $N = K + N_l + N_r$. If the final trimmed event does not perfectly match the size N—i.e., T/N is not an integer—then the final event is also zero-padded, as described in Equation (4.5c).

Finally, suppose there exists an event $C \in \mathbb{R}^{F \times T}$ with $T = N$, then $X = C$.

We describe X to be a full input when there is no zero-padding and we describe X to be a padded input when there is zero-padding added to X.

4.2.3 Class Processing

After this processing we were presented with a significantly unbalanced class distribution within our experiments, which has shown to degrade performance in machine learning [74, 75]; this will also be shown with our data in Section 4.4. In our experiments, the hammer strike class contained 180 seconds of data and the snowmobile contained 4,618 seconds. If these were the only two classes, the network could determine that snowmobiles were the only class present, label everything as snowmobile, and it would be 96.2% accurate. As such, we reduce the disparity between classes by removing much of the snowmobile data, as seen in Figure 4.1. This reasoning is supported in Section 4.4, where the algorithm overpredicts snowmobile events when no class normalization is done. With the data mapped into a usable form for the neural network architecture, we now describe the different architectures we will be comparing.



(a) (b)

Figure 4.1: Normalization of class distribution to remove excess sample training with snowmobile data. View (a) shows the class distributions without reduction and (b) shows the class distributions after reduction.

4.2.4 Convolutional neural network

The CNN is a network that computes the convolution operation on input data,

$$y_{f,t} = \sum_{i=0}^{F} \sum_{j=0}^{N} w_{i,j} x_{f-i,n-j} + b_{i,j} \tag{4.6}$$

where w is a trainable weight that is convolved along the input data, x; b is a bias term at each kernel position; and y is the convolved output. The CNN has its power in finding spatially close relations in the data [57]. It is also computationally fast, but it is challenging for the CNN to find long-term relations since Eq. (4.6) convolves only spatially close positions within x. To handle long-term relations, we now look to the LSTM.

4.2.5 Long short-term memory network

The LSTM network handles long-term relations within the data using recurrence [37]. Recurrence has been known to have a problem with its vanishing gradient in during back propagation using an unconstrained recurrent neural network [76]. Because of this, the LSTM constrains the data with "gates" to handle the input data,

$$\mathbf{i}_t = \sigma(W_i \left[\mathbf{h}_{t-1}^T \; \mathbf{x}_t^T\right]^T + \mathbf{b}_i), \tag{4.7a}$$

$$\mathbf{f}_t = \sigma(W_f \left[\mathbf{h}_{t-1}^T \; \mathbf{x}_t^T\right]^T + \mathbf{b}_f), \tag{4.7b}$$

$$\mathbf{o}_t = \sigma(W_o \left[\mathbf{h}_{t-1}^T \; \mathbf{x}_t^T\right]^T + \mathbf{b}_o), \tag{4.7c}$$

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tanh\left(\left[\mathbf{h}_{t-1}^T \; \mathbf{x}_t^T\right]^T\right), \tag{4.7d}$$

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t), \tag{4.7e}$$

where W and $\mathbf{b}$ are trainable parameters for the three gates, the "input" gate, the "forget" gate, and the "output" gate; $\mathbf{h}$ is the output of the LSTM; $\mathbf{c}$ is the state of the LSTM that is adjusted with new input data; σ is the sigmoid activation function, $1/(1 + e^{-x})$; $[\cdot]$ concatenates the two values within the brackets; and $\circ$ is the Hadamard product, or element-wise multiplication of the two values. An LSTM cell is all operations in Eq. 4.7.

The gates within the LSTM attempt to reduce the effects of the vanishing gradient by constraining the gradients through backpropagation [34]. The LSTM handles long-term relations with the states **c** and **h**, but these two states also depend upon the previous state, which means that there is little parallelization that can be computed within the LSTM. Because of this, a GPU cannot be used at its fullest potential to parallelize computations. We now look to the ViT, which allows parallel computation while still maintaining long-term relations.

4.2.6 Vision Transformer

The ViT is a modified version of the originally proposed Transformer [46, 48]. In a ViT, the input data are positionally embedded before the data are transformed by the Transformer encoder. This means that the samples of the input data—i.e., the spectrogram—are split into rectangular chunks along both time and frequency.

The ViT calculates attention by

$$\text{attention}(X) = \text{softmax}\left(\frac{(W_Q X)(W_K X)^T}{\sqrt{d}}\right) W_V X, \tag{4.8}$$

where the input is X after being positionally embedded and the trainable weight matrices are W_Q, W_K, and W_V. The scaling parameter $\sqrt{d}$ used is as proposed by

Vaswani et al. [48], and softmax(·) is defined as

$$\text{softmax}(\mathbf{x}) = \frac{e^{\mathbf{x}}}{\sum_{k=1}^{K} e^{x_k}}, \tag{4.9}$$

which is a normalizing factor on the product $W_V X$.

Attention is used to determine the importance of various elements of the data. Since the data in Equation (4.8) contain the same X data for the three weight matrices, this equation is defined as self-attention and features are determined within the data itself using attention.

Multi-headed attention (MHA) is a method in which Equation (4.8) is computed multiple times for more trainable parameters [48]. MHA allows multiple attention connections to be computed for the same data which gives an opportunity for the ML algorithm to learn multiple attentions. We compare the number of MHAs as a hyperparameter within our ViT architecture in an effort to yield better performance on the test data.

The Transformer encoder is computed as

$$Z = W(X + \text{attention}(X)) + X + \text{attention}(X), \tag{4.10}$$

where W is an FCNN, and attention$(\cdot)$ is defined at Equation (4.8). With the mathematics of the networks explained, we now look into the architectures themselves.

4.2.7 Neural network architectures

The CNN architecture consists of a series of alternating CNN layers and *batch normalization* (BN) layers[77]. After the BN layer, we use the ReLU activation function,

$$\mathrm{ReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{4.11}$$

to add more non-linearities to the network. Each CNN had a kernel size of $(3, 5)$, a convolution step size of $(2, 2)$, and there are 64 filters at each CNN layer. The last part of the CNN architecture has an FCNN with a softmax function for predicting each acoustic class. The number of CNN layers was varied in hyperparameter tuning, described in Section 4.3.

The LSTM architecture consists of a series of LSTM layers with 64 cells per layer. Each LSTM layer predicts at each timestep—i.e., the output of each LSTM layer contains 64 samples. The end of the LSTM architecture contains an FCNN with a softmax function for predicting each acoustic class. The number of LSTM layers was varied in hyperparameter tuning, described in Section 4.3.

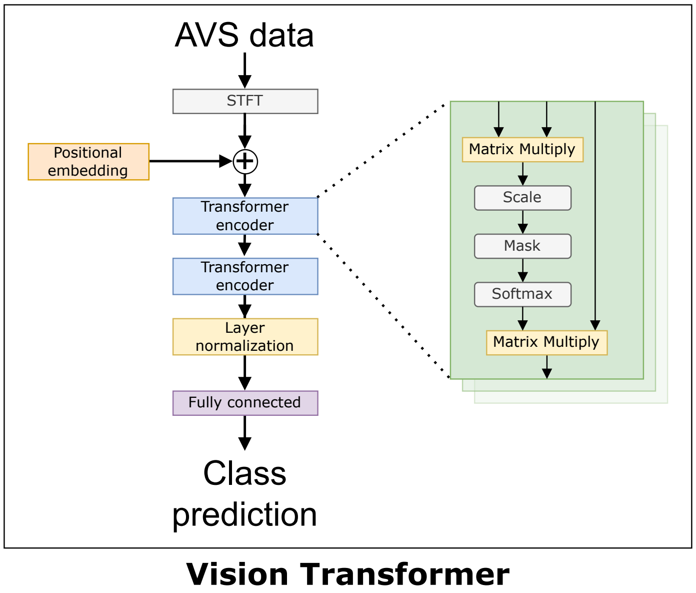


Figure 4.2: Vision Transformer neural network architecture for classifier acoustic vector sensor data

The ViT architecture consists of a series of Transformer encoders—i.e., a series of Equation (4.10)—followed by an FCNN with a softmax activation function for classification. Figure 4.2 illustrates the full neural network model. The depth of the neural network is not very deep in comparison to those in the original Transformer papers [46, 48], but the data in this experiment is not as extensive as in those papers; therefore, such a deep network would go beyond the limited data and overfit quickly, essentially learning the noise in the data.

4.3 Experimental Methods

The experiments were conducted at the Keweenaw waterway next to Michigan Technological University [12, 13, 14]. The Keweenaw waterway is a narrow and shallow water channel where first-year ice forms typically from January to March. Our experiments were conducted from January to March, 2021.

A single Meggitt VS-209 AVS passively recorded the signature of each acoustic source in an uncooperative manner; i.e., there is no active sonar. The AVS was recorded at a sample rate of 17,076 Hz. This sample rate was chosen because the Meggitt VS-209 3-dB frequency bandwidth is around 8,000 Hz and the closest discrete sampling rate for the National Instruments NI-9234 *analog-digital converter* (ADC) is 17,067 Hz, which was used for the experiment.

Each class was recorded individually such that no class would overlap one another. There exists classes where ice cracks and background noise—sources of sound not corresponding to one of our classes—occurred, generating transients and disturbances, but these are a realistic view of the ice environment and were thus not removed. The data were labeled and classified at the individual sample-level within a tenth of a second. The tally of classes and their sample count are shown in Table 4.1 along with the normalized class count, i.e., where some snowmobile events were removed.

Class name	Class count	Reduced class count
Snowmobile	4618	397
Hammer	180	180
Electric auger	176	176
Underwater speaker	397	397
Gas auger	268	268
Ice Saw	262	262

Table 4.1
A comparison of data counts in all classes lasting 1 second for each count

The data were post-processed using an STFT hanning window, 50% overlap, 0.01 second bin length (171 samples), and zero-padded to 256 samples. The small bin length is necessary to facilitate the small hammer strike duration, else the duration of each hammer transient would be contained within a single bin length. A single class count is considered to be a spectrogram consisting of 200 time steps, or 1 second.

The spectrum data, $|I(f)| \in \mathbb{R}^{128\times 200}$, is globally normalized from 0 to 1, where

$$X_{dB} = 10\log_{10}|I(f)|, \tag{4.12a}$$

$$X = \frac{X_{dB} - \min(X_{dB})}{\max(X_{dB}) - \min(X_{dB})} \tag{4.12b}$$

describes the normalized input data, X, which is then analyzed by the neural network architecture.

Once the neural network parameters were trained on the training data, predictions were made on the test data. The neural network was trained using the Adam

		Patches								
		8			16			32		
		Heads			Heads			Heads		
		2	4	8	2	4	8	2	4	8
learning rate	0.005	0.51	0.42	0.45	0.89	0.48	0.45	0.45	0.46	0.49
	0.002	0.43	0.45	0.51	0.45	0.48	0.41	0.56	0.5	0.56
	0.001	0.4	0.44	0.48	0.47	**0.39**	0.44	0.7	0.53	0.95

Table 4.2
Validation loss with varying hyperparameters for the ViT architecture

optimizer[61] with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and the learning rate adjusted via hyperparameter tuning for each network. The test data set was split to contain 20% of the data, while the training data set contained 80% of the data. The training data were further split into two sets: 90% used for training and 10% used for validation. The validation data were used to tune the hyperparameters of the architecture, meaning that the validation data were used as a surrogate test data set by which parameters that cannot be learned by ML were adjusted. Hyperparameters were determined with a grid search.

For the ViT architecture, the hyperparameters we studied were the learning rate, the number of heads in the MHA in the ViT, and the size of the positional embedding patches in the ViT. Table 4.2 shows the validation losses for sets of hyperparameter combinations for the ViT architecture.

With the results in Table 4.2, we have found the network hyperparameters that resulted in the best validation loss were a learning rate of 0.001, 16 ViT patches, and 4 Transformer heads in MHA.

		CNN+BN+ReLU count				LSTM layer count			
		2	3	4	5	2	3	4	5
learning rate	0.005	0.38	0.55	0.42	0.39	0.43	0.47	0.45	0.46
	0.002	0.49	0.61	0.46	**0.32**	0.44	0.45	0.50	0.48
	0.001	0.65	0.45	0.36	0.42	0.43	**0.39**	0.58	0.78

Table 4.3
Validation loss with varying hyperparameters for the CNN architecture and LSTM architecture

For the CNN and LSTM network architectures, the number of layers for the network was determined using hyperparameter tuning. The best hyperparameters found for the CNN was a learning rate of 0.002 for the Adam optimizer and a network containing a CNN layer and a BN layer followed by a ReLU activation 5 times. The best hyperparameters found for the LSTM was a learning rate of 0.001 and 3 LSTM layers. These analyses are shown in Table 4.3.

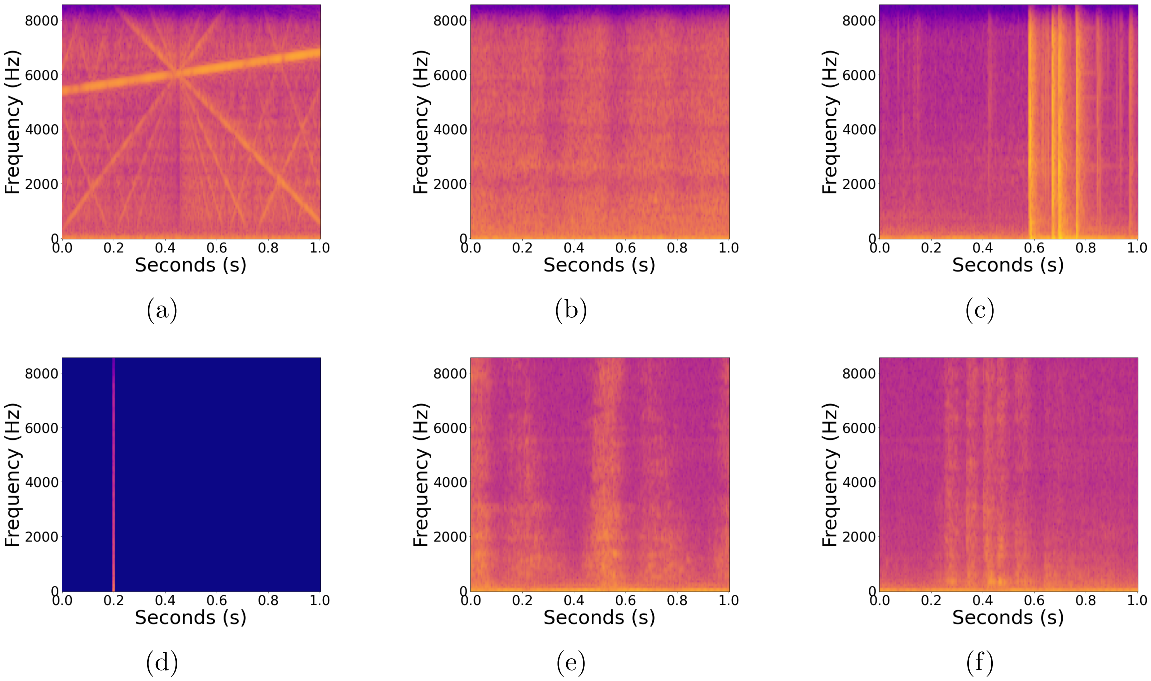
The hyperparameters that resulted in the lowest loss in the validation data, along with the trainable parameters of the network, were used to test each neural network. With the neural network trained and hyperparameters determined, we can now test the network with the held-out test data and analyze the results.

4.4 Experimental Results

The experimental data contains 5,901 events of 1 second in length in the non-normalized data set. The normalized data set, with removed snowmobile events, contains 1,680 events of 1 second in length. The distribution of total events is shown in Figure 4.1. The test data contained 1180 events in the non-normalized data set and 336 events when a portion of the snowmobile events were removed. One should note that the test data were processed in the same manner of the training data from Eqns. 4.4—4.5c, which zero-padded the test data. This removes the exact scenario of the real world environment; i.e., we are not testing on a stream of input data, but it helps describe the results by having direct sample sizes along each event.

Sample spectrogram data are shown in Figure 4.3 as a visual representation of all the classes on which the neural network was trained. These spectrogram data are examples of the exact inputs to the neural network. Note in Figure 4.3(c) that the transients are the ice cracking, which is typical of heavy, on-ice moving sources in a first-year ice environment.

When the number of snowmobile events were not reduced, the classification performance of the neural network was degraded. When the class distribution was more balanced, each neural network was able to classify without bias towards the snowmobile. This bias is shown in Figure 4.4 with the ViT architecture. One can see the overconfidence of the snowmobile class in Figure 4.4(a), where the snowmobile class was incorrectly predicted more often on other data. One should also note that the accuracies for the snowmobile and underwater speaker are exceptional, but those two classes were very distinct, as seen in Figures 4.3(a) and 4.3(c). The underwater



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.3: Acoustic intensity spectrogram plots for (a) an underwater speaker event, (b) a gas auger event, (c) a snowmobile event, (d) a hammer strike event, (e) an ice saw event, and (f) an electric auger event.

speaker had a very unnatural and distinct pattern, and the snowmobile source was constantly moving; therefore, the frequency response was constantly changing due to the destructive and constructive interference patterns in the shallow ice environment [12].

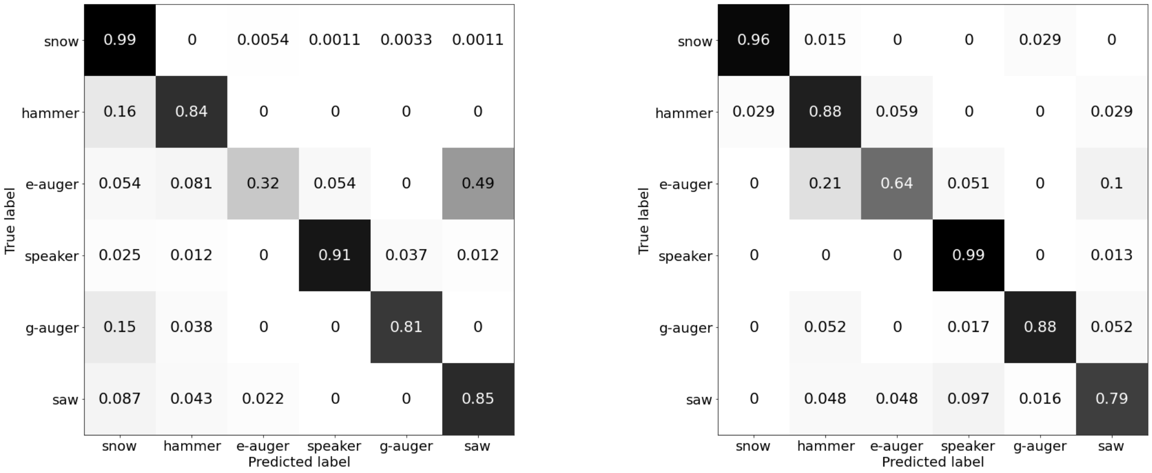
To compare a single value between each architecture, we look to the F1-score. The F1-scores was calculated individually for each class,

$$F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{4.13}$$

and then the mean of each F1-score along each class was found to determine a single,

	Reduced	Non-reduced
ViT	0.853	0.794
LSTM	0.809	0.808
CNN	0.735	0.773

Table 4.4
F1-scores for the reduced-snowmobile and non-reduced data sets for each neural network



(a) (b)

Figure 4.4: Classification accuracy for the ViT architecture for (a) non-reduced (5,901 events) and (b) reduced-snowmobile class distributions (1,680 events).

composite F1-score. The purpose for a balanced class F1-score was to evenly compare the neural network classifier for each class. A comparison of the F1-scores in both types of data sets is shown in Table 4.4.

One behavior we observed was that the neural network performed poorly for the data produced by Equation (4.5c), i.e., the data at the end of a long event. When analyzing the results, we found a significant decrease in accuracy on the zero-padded events. Of all 37 non-hammer zero-padded events in the normalized data set, 30 were incorrectly classified with the ViT architecture, an accuracy of 19%. The training data only contained 103 events—7.6% of the training data—with non-hammer zero-padded events. We hypothesize that the neural network learned the patterns within a full event, which was prevalent in the long signals, but failed to discover the patterns zero-padded events because there were not enough events for the neural network to consider properly.

4.5 Conclusion

In this paper, we have demonstrated a method to classify on-ice acoustic sources with an AVS using machine learning. Once the acoustic data were transformed into a spectrogram, the ViT was capable of classifying acoustic sources with a relatively high F1-score for ice acoustic classification. The ViT showed excellent performance in benchmark data [46]; we demonstrated that the ViT also has good promise for acoustic classification in real-world data in a complex scattering environment.

These results show the strength of the ViT for this application. Further fine-tuning and optimization with deep learning techniques can be employed, but the accuracies presented here are encouraging for the use of the ViT in practical scenarios. We anticipate further improvements within the intersection of acoustics and machine learning.

Chapter 5

Conclusion and future work

This dissertation researched two main topics: how to analyze underwater and on-ice acoustic data using a neural network to calculate multiple SONAR-related results, and how to conduct experiments to facilitate the training data required for such networks.

We showed in Chapter 2 that with a single AVS, a network with an LSTM-backbone could track the DOA of a moving source with moderate accuracy, increasing the accuracy of DOA estimation by 14.6^o when comparing a weighted average of the acoustic intensity azimuthal DOA at each frequency bin.

This accuracy was then greatly improved upon in Chapter 3 using two AVSs and a ViT neural network architecture in a more complex environment: localizing an on-ice source, rather than estimating the DOA of a surface-water source.Using the ViT architeture, we localized the source with 3 meter accuracy from the source GPS coordinates. We also proposed a novel approach to localizing a target in a constrained environment and described the results in the complex environment. This localization approach processes the data to predict a grid of locations as a classification problem, where individual classes are removed to constrain the locations the source could be localized.

Finally, we showed in Chapter 4 that the ViT was also capable of classifying six on-ice acoustic classes with relatively good accuracy with an F1-score of 0.85.

Initially, experiments were conducted in the summer of 2020 at the Keweenaw Waterway to track a boat using a single AVS. Using a single AVS, only DOA estimation was possible, but the analysis showed promising accuracy using RNNs.

Many experiments were conducted in the following winter of 2020-2021 at the Keweenaw Waterway, where two AVSs were separated by 30 m. The snowmobile experiments had its acoustic signatures localized using both an LSTM network and ViT network with the ViT performing better accuracy. Other experiments were conducted to generate various on-ice acoustic transients: an electric auger, a gas auger, an ice saw, and a hammer.

In the next winter of 2021-2022, the remaining experiments focused on reducing noise and expanding the capability of the experiments. These experiments introduced future work and analysis. Experiments have been conducted to generalize the relative position of the two sensors, which was possible once the sensor suite became portable. From these experiments, a large data set has been generated from three years of ice experiments and passive recordings, totalling beyond 2 TB of raw data.

This research has shown that there is promise with utilizing machine learning in the field of ice acoustics. Machine learning is relatively new in ice acoustics, so this research hopes to encourage other researchers to continue studying different machine learning techniques in this field.

5.1 Future Work

With this extensive amount of data, first-year ice characteristics can be further studied. Going forward, many new types of experiments can be conducted with research to conduct networks for a more generalized approach to the analyses described in this dissertation.

5.1.1 Generalized sensor positions

Every experiment in this dissertation analyzed results with the one or two sensors in the same position, but the machine learning algorithm can perform very poorly if the positions of these sensors are changed. Experiments can be conducted with varying relative position to find a more generalized algorithm. It is a hope that the relative position between each sensor can be parameterized and adjusted for within the neural network architecture. With the positions parameterized, an underwater or on-ice network of sensors can work together to localize targets at far distances.

5.1.2 Generalized environment

Every experiment in this dissertation was conducted at the Keweenaw Waterway, which may cause our specific networks to not generalize to other areas. Experiments can be conducted at different locations: shallow and wide (a small lake), deep and narrow (large river), or deep and wide (a large lake), or in saltwater environments (the ocean). These different environments will hopefully find a manner in which to parameterize the acoustic environment. With the combination of both sensor positions and the water environment being parameterized, a highly generalized neural network may possibly be found.

5.1.3 Sources with varying elevation

Every experiment in this dissertation studied moving or non-moving sources at the surface of the water or the top of the ice. The elevation angles did not change much and because of this, the dissertation disregarded elevation angle because they were very similar. Experiments can be conducted to have varying elevation angles, such as submerged sources like an underwater ROV or a scuba diver.

5.1.4 Multiple target source localization

Every experiment in this dissertation studied a single moving or non-moving source. If there were two events that occur at the same time in experiments conducted during this dissertation, these data were removed to reduce the complexity of an already complex environment. Experiments can be conducted with multiple sources at the same time: two moving broadband sources, one broadband and another with transients, or three or more sources at once. If multiple sources at once are properly analyzed, this can transform the analyses into a largely real-world environment.

5.1.5 On-ice acoustic data sets

There is still a lack of labeled data in the underwater acoustics and on-ice acoustics field. This is a tedious and time-consuming task that has not yet been done for a large data set. Further efforts can be done to contribute to an on-ice acoustic data set to be used in machine learning experimentation.

References

[1] Maslanik, J.; Stroeve, J.; Fowler, C.; Emery, W. Distribution and Trends in Arctic Sea Ice Age Through Spring 2011. *Geophysical Research Letters* **2011**, *38*.

[2] Stroeve, J.C.; Markus, T.; Boisvert, L.; Miller, J.; Barrett, A. Changes in Arctic Melt Season and Implications for Sea Ice Loss. *Geophysical Research Letters* **2014**, *41*, 1216–1225.

[3] Rodrigues, J. The Rapid Decline of the Sea Ice in the Russian Arctic. *Cold Regions in Sci. and Technology* **2008**, *54*, 124–142.

[4] Lei, R.; Xie, H.; Wang, J.; Lepparanta, M.; Jonsdottir, I.; Zhang, Z. Changes in Sea Ice Conditions along the Arctic Northeast Passage from 1979-2012. *Cold Regions in Sci. and Technology* **2015**, *119*, 132–144.

[5] Timco, G.W.; Weeks, W.F. A Review of the Engineering Properties of Sea Ice. *Cold Regions in Sci. and Technology* **2010**, *60.*

[6] Timco, G.W.; Frederking, R.M.W. A Review of Sea Ice Density. *Cold Regions in Sci. and Technology* **1995**, *24*.

[7] Smith, L.C.; Stephenson, S.R. New Trans-Arctic Shipping Routes Navigable by Midcentury. *Proc. of the Nat. Academy of Sci. of the United States of America* **2013**, *110, no. 13*.

[8] Stephenson, S.R.; Brigham, L.W.; Smith, L.C. Marine Accessibility Along Russia's Northern Sea Route. *Polar Geography* **2014**, *37, no. 2*.

[9] Somanathan, S.; Flynn, P.C.; Szymanski, J. The Northwest Passage: A Simulation. *Winter Simulation Conference* **2006**.

[10] Lasserre, F.; Pelletier, S. Polar Super Seways? Maritime Transport in the Arctic: an Analysis of Shipowners' Intentions. *J. of Transport Geography* **2001**, *19*.

[11] Roston, M. The Northwest Passage's Emergence as an International Highway. *Southwestern J. of International Law* **2009**, *15*.

[12] Whitaker, S.; Dekraker, Z.; Barnard, A.; Havens, T.C.; Anderson, G.D. Uncertain Inference Using Ordinal Classification in Deep Networks for Acoustic Localization. 2021 International Joint Conference on Neural Networks (IJCNN). IEEE, 2021. doi:10.1109/ijcnn52387.2021.9533605.

[13] Whitaker, S.; Barnard, A.; Anderson, G.D.; Havens, T.C. Through-Ice Acoustic Source Tracking Using Vision Transformers with Ordinal Classification. *Sensors* **2022**, *22*. doi:10.3390/s22134703.

[14] Whitaker, S.; Barnard, A.; Anderson, G.D.; Havens, T.C. Recurrent networks for direction-of-arrival identification of an acoustic source in a shallow water channel using a vector sensor. *The Journal of the Acoustical Society of America* **2021**, *150*, 111–119. doi:10.1121/10.0005536.

[15] Whitaker, S.; Barnard, A.; Anderson, G.D.; Havens, T.C. Submitted to POMA, Using Vision Transformers for classification of through-ice acoustic sources. Proceedings of Meetings on Acoustics. ASA, 2022.

[16] Niu, H.; Ozanich, E.; Gerstoft, P. Ship localization in Santa Barbara Channel using machine learning classifiers. *The Journal of the Acoustical Society of America* **2017**, *142*, EL455–EL460. doi:10.1121/1.5010064.

[17] Niu, H.; Reeves, E.; Gerstoft, P. Source localization in an ocean waveguide using supervised machine learning. *The Journal of the Acoustical Society of America* **2017**, *142*, 1176–1188. doi:10.1121/1.5000165.

[18] Wang, Y.; Peng, H. Underwater acoustic source localization using generalized regression neural network. *The Journal of the Acoustical Society of America* **2018**, *143*, 2321–2331. doi:10.1121/1.5032311.

[19] Yangzhou, J.; Ma, Z.; Huang, X. A deep neural network approach to acoustic source localization in a shallow water tank experiment. *The Journal of the Acoustical Society of America* **2019**, *146*, 4802–4811.

[20] Ozanich, E.; Gerstoft, P.; Niu, H. A feedforward neural network for direction-of-arrival estimation. *The Journal of the Acoustical Society of America* **2020**, *147*, 2035–2048. doi:10.1121/10.0000944.

[21] Zou, Y.; Gu, R.; Wang, D.; Jiang, A.; Ritz, C.H. Learning a robust DOA estimation model with acoustic vector sensor cues. 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIP ASC), 2017, pp. 1688–1691.

[22] Ferguson, E.L.; Williams, S.B.; Jin, C.T. Convolutional neural network for single-sensor acoustic localization of a transiting broadband source in very shallow water. *The Journal of the Acoustical Society of America* **2019**, *146*, 4687–4698.

[23] Cao, H.; Wang, W.; Su, L.; Ni, H.; Gerstoft, P.; Ren, Q.; Ma, L. Deep transfer learning for underwater direction of arrival using one vector sensor. *The Journal of the Acoustical Society of America* **2021**, *149*, 1699–1711.

[24] Qin, D.; Tang, J.; Yan, Z. Underwater Acoustic Source Localization Using LSTM Neural Network. 2020 39th Chinese Control Conference (CCC), 2020, pp. 7452–7457.

[25] Huang, Z.; Xu, J.; Gong, Z.; Wang, H.; Yan, Y. Multiple Source Localization in a Shallow Water Waveguide Exploiting Subarray Beamforming and Deep Neural Networks. *Sensors* **2019**, *19*, 4768. doi:10.3390/s19214768.

[26] Huang, Z.; Xu, J.; Gong, Z.; Wang, H.; Yan, Y. Source localization using deep neural networks in a shallow water environment. *The Journal of the Acoustical Society of America* **2018**, *143*, 2922–2932.

[27] Trees, H.L.V. *Optimum Array Processing*; 2002.

[28] Kang, K.; Gabrielson, T.B.; Lauchle, G.C. Development of an accelerometer-based underwater acoustic intensity sensor. *The Journal of the Acoustical Society of America* **2004**, *116*, 3384–3392.

[29] Bereketli, A.; Guldogan, M.B.; Kolcak, T.; Gudu, T.; Avsar, A.L. Experimental Results for Direction of Arrival Estimation with a Single Acoustic Vector Sensor in Shallow Water. *Journal of Sensors* **2015**, *2015*.

[30] Fahy, F. *Sound Intensity, Second Edition*; 1995.

[31] Penhale, M.B. Acoustic localization techniques for application in near-shore arctic environments. PhD thesis, Michigan Technological University, 2019.

[32] Connor, J.; Atlas, L. Recurrent neural networks and time series prediction. IJCNN-91-Seattle International Joint Conference on Neural Networks, 1991, Vol. i, pp. 301–306 vol.1. doi:10.1109/IJCNN.1991.155194.

[33] Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* **1994**, *5*, 157–166. doi:10.1109/72.279181.

[34] Werbos, P.J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* **1990**, *78*, 1550–1560.

[35] Collins, M.D. A split-step Padé solution for the parabolic equation method. *The Journal of the Acoustical Society of America* **1993**, *93*, 1736–1742.

[36] Penhale, M.B.; Barnard, A.R.; Shuchman, R. Multi-modal and short-range transmission loss in thin, ice-covered, near-shore Arctic waters. *The Journal of the Acoustical Society of America* **2018**, *143*, 3126–3137. doi:10.1121/1.5038569.

[37] Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **1997**, *9*, 1735–1780.

[38] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.u.; Polosukhin, I. Attention is All you Need. Advances in Neural Information Processing Systems; Guyon, I.; Luxburg, U.V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; Garnett, R., Eds. Curran Associates, Inc., 2017, Vol. 30.

[39] Erol-Kantarci, M.; Mouftah, H.T.; Oktug, S. A Survey of Architectures and Localization Techniques for Underwater Acoustic Sensor Networks. *IEEE Communications Surveys & Tutorials* **2011**, *13*, 487–502.

[40] Anand, A.; Mukul, M.K. Comparative analysis of different direction of arrival estimation techniques. 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). IEEE, 2016.

[41] Pinheiro, B.C.; Moreno, U.F.; de Sousa, J.T.B.; Rodriguez, O.C. Kernel-Function-Based Models for Acoustic Localization of Underwater Vehicles. *IEEE Journal of Oceanic Engineering* **2017**, *42*, 603–618.

[42] Huang, Z.; Xu, J.; Gong, Z.; Wang, H.; Yan, Y. Source localization using deep neural networks in a shallow water environment. *The Journal of the Acoustical Society of America* **2018**, *143*, 2922–2932. doi:10.1121/1.5036725.

[43] Ullah, I.; Chen, J.; Su, X.; Esposito, C.; Choi, C. Localization and Detection of Targets in Underwater Wireless Sensor Using Distance and Angle Based Algorithms. *IEEE Access* **2019**, *7*, 45693–45704.

[44] Huang, Z.; Xu, J.; Li, C.; Gong, Z.; Pan, J.; Yan, Y. Deep Neural Network for Source Localization Using Underwater Horizontal Circular Array. 2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO). IEEE, 2018.

[45] Qin, D.; Tang, J.; Yan, Z. Underwater Acoustic Source Localization Using LSTM Neural Network. 2020 39th Chinese Control Conference (CCC). IEEE, 2020. doi:10.23919/ccc50068.2020.9189504.

[46] Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2021, [arXiv:cs.CV/2010.11929].

[47] Gong, Y.; Chung, Y.A.; Glass, J. AST: Audio Spectrogram Transformer, 2021, [arXiv:cs.SD/2104.01778].

[48] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need, 2017, [arXiv:cs.CL/1706.03762].

[49] Sudarsanam, P.; Politis, A.; Drossos, K. Assessment of Self-Attention on Learned Features For Sound Event Localization and Detection, 2021, [arXiv:cs.SD/2107.09388].

[50] Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A Survey on Vision Transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2022**, pp. 1–1. doi:10.1109/tpami.2022.3152247.

[51] Zhai, X.; Kolesnikov, A.; Houlsby, N.; Beyer, L. Scaling Vision Transformers. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 12104–12113.

[52] Fahy, F. *Sound Intensity, Second Edition*; 1995.

[53] Kim, K.; Gabrielson, T.B.; Lauchle, G.C. Development of an accelerometer-based underwater acoustic intensity sensor. *The Journal of the Acoustical Society of America* **2004**, *116*, 3384–3392.

[54] Liikonen, L.; Alanko, M.; Jokinen, S.; Niskanen, I.; Virrankoski, L. Snowmobile noise **2007**.

[55] Mullet, T.C.; Morton, J.M.; Gage, S.H.; Huettmann, F. Acoustic footprint of snowmobile noise and natural quiet refugia in an Alaskan wilderness. *Natural Areas Journal* **2017**, *37*, 332–349.

[56] Thode, A.M.; Sakai, T.; Michalec, J.; Rankin, S.; Soldevilla, M.S.; Martin, B.; Kim, K.H. Displaying bioacoustic directional information from sonobuoys using "azigrams". *The Journal of the Acoustical Society of America* **2019**, *146*, 95–102. doi:10.1121/1.5114810.

[57] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition, 2015, [arXiv:cs.CV/1512.03385].

[58] Kim, Y.; Denton, C.; Hoang, L.; Rush, A.M. Structured Attention Networks, 2017, [arXiv:cs.CL/1702.00887].

[59] ANG-E66. *Global Positioning System Standard Positioning Service Performance Analysis Report*; FAA William J. Hughes Technical Center, 2021, Q1; pp. 20–21.

[60] Frank, E.; Hall, M. A Simple Approach to Ordinal Classification. In *Machine Learning: ECML 2001*; Springer Berlin Heidelberg, 2001; pp. 145–156.

[61] Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization, 2017, [arXiv:cs.LG/1412.6980].

[62] Chollet, F.; et al. Keras, 2015.

[63] Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Software available from tensorflow.org.

[64] Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *arXiv:2004.05150* **2020**.

[65] Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in Vision: A Survey. *ACM Comput. Surv.* **2021**. doi:10.1145/3505244.

[66] Jiang, Z.H.; Hou, Q.; Yuan, L.; Zhou, D.; Shi, Y.; Jin, X.; Wang, A.; Feng, J. All Tokens Matter: Token Labeling for Training Better Vision Transformers. Advances in Neural Information Processing Systems; Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; Vaughan, J.W., Eds. Curran Associates, Inc., 2021, Vol. 34, pp. 18590–18602.

[67] Boashash, B.; O'shea, P. A methodology for detection and classification of some underwater acoustic signals using time-frequency analysis techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **1990**, *38*, 1829–1841.

[68] Vahidpour, V.; Rastegarnia, A.; Khalili, A. An automated approach to passive sonar classification using binary image features. *Journal of Marine Science and Application* **2015**, *14*, 327–333. doi:10.1007/s11804-015-1312-z.

[69] Domingos, L.C.; Santos, P.E.; Skelton, P.S.; Brinkworth, R.S.; Sammut, K. A survey of underwater acoustic data classification methods using deep learning for shoreline surveillance. *Sensors* **2022**, *22*, 2181.

[70] Choi, J.; Choo, Y.; Lee, K. Acoustic Classification of Surface and Underwater Vessels in the Ocean Using Supervised Machine Learning. *Sensors* **2019**, *19*, 3492. doi:10.3390/s19163492.

[71] Cinelli, L.; Chaves, G.; Lima, M. Vessel Classification through Convolutional Neural Networks using Passive Sonar Spectrogram Images. Anais de XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais. Sociedade Brasileira de Telecomunicações, 2018. doi:10.14209/sbrt.2018.340.

[72] Chen, C.H.; Lee, J.D.; Lin, M.C. Classification of Underwater Signals Using Neural Networks. *Journal of Applied Science and Engineering* **2000**, *3*, 31–48. doi:10.6180/jase.2000.3.1.04.

[73] Bonet-Solà, D.; Alsina-Pagès, R.M. A Comparative Survey of Feature Extraction and Machine Learning Methods in Diverse Acoustic Environments. *Sensors* **2021**, *21*.

[74] Kaur, H.; Pannu, H.S.; Malhi, A.K. A Systematic Review on Imbalanced Data Challenges in Machine Learning: Applications and Solutions. *ACM Comput. Surv.* **2019**, *52*. doi:10.1145/3343440.

[75] Mohammed, R.; Rawashdeh, J.; Abdullah, M. Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. 2020 11th International Conference on Information and Communication Systems (ICICS), 2020, pp. 243–248. doi:10.1109/ICICS49469.2020.239556.

[76] Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **1998**, *6*, 107–116.

[77] Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. International conference on machine learning. PMLR, 2015, pp. 448–456.

[78] Agarap, A.F. Deep Learning using Rectified Linear Units (ReLU). *CoRR* **2018**, *abs/1803.08375*, [1803.08375].

[79] Cardoso, J.S.; Sousa, R. Measuring the Performance of Ordinal Classification.

International Journal of Pattern Recognition and Artificial Intelligence **2011**, *25*, 1173–1195.

[80] Zou, Y.; Gu, R.; Wang, D.; Jiang, A.; Ritz, C.H. Learning a robust DOA estimation model with acoustic vector sensor cues. 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017, pp. 1688–1691. doi:10.1109/APSIPA.2017.8282304.

[81] Zheng, W.Q.; Zou, Y.X.; Ritz, C. Spectral mask estimation using deep neural networks for inter-sensor data ratio model based robust DOA estimation. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 325–329. doi:10.1109/ICASSP.2015.7177984.

[82] Hoang, M.T.; Yuen, B.; Dong, X.; Lu, T.; Westendorp, R.; Reddy, K. Recurrent Neural Networks for Accurate RSSI Indoor Localization. *IEEE Internet of Things Journal* **2019**, *6*, 10639–10651.

[83] Sun, S.; Zhang, X.; Zheng, C.; Fu, J.; Zhao, C. Underwater Acoustical Localization of the Black Box Utilizing Single Autonomous Underwater Vehicle Based on the Second-Order Time Difference of Arrival. *IEEE Journal of Oceanic Engineering* **2020**, *45*, 1268–1279.

[84] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge, 2015, [arXiv:cs.CV/1409.0575].

[85] Kitaev, N.; Kaiser, L.; Levskaya, A. Reformer: The Efficient Transformer. International Conference on Learning Representations, 2020.

[86] Almeida, F.; Xexéo, G. Word Embeddings: A Survey, 2019. doi:10.48550/ARXIV.1901.09069.

Appendix A

Figures

Figures A.1–A.7 show a single snowmobile experiment conducted on February 17th, 2021. These figures show the totality of the truth data in Chapter 3 [13]. The purpose of these figures is to show that the data in Section 3.3 are not hand-picked, but rather they are typical of all the other test data.

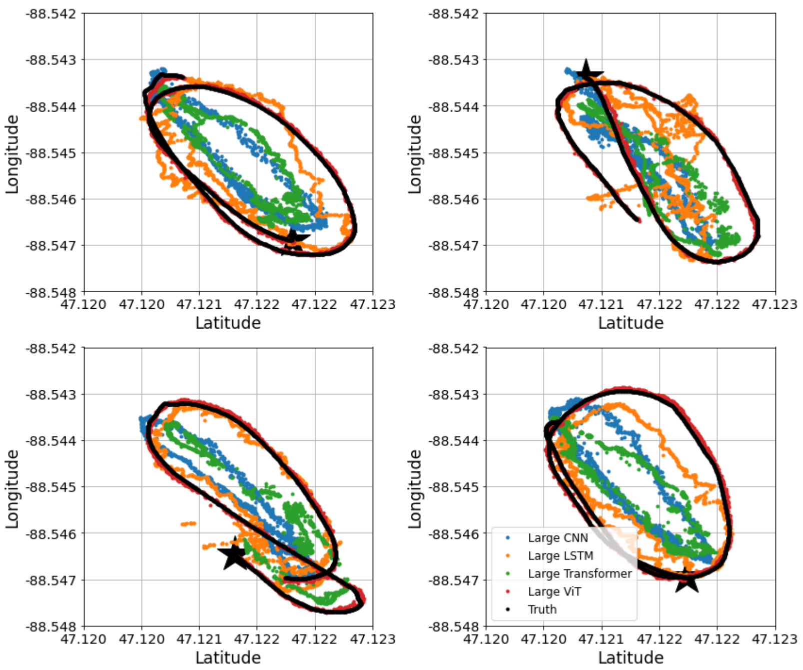


Figure A.1: Seconds 0 to 283 of test data using large regression networks where the star indicates the start of the each time.

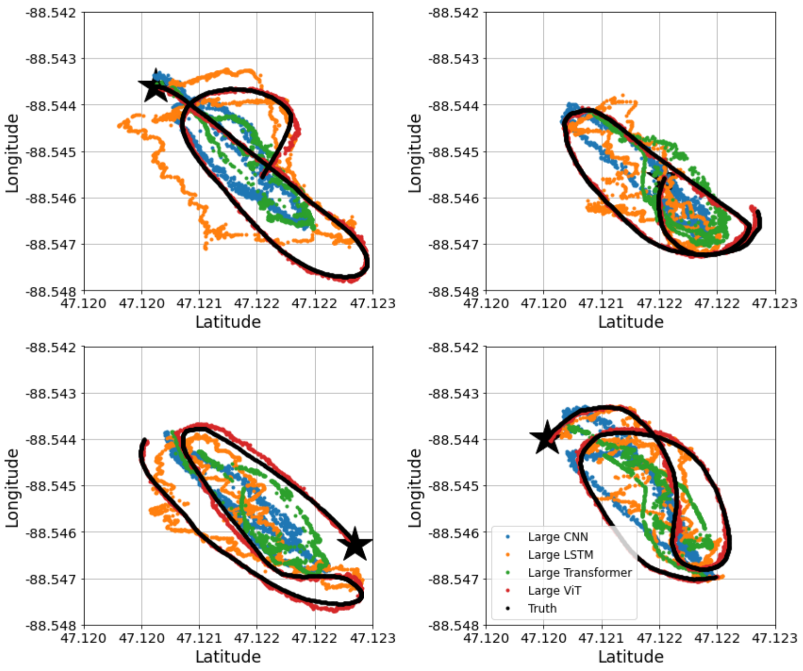


Figure A.2: Seconds 283 to 566 of test data using large regression networks where the star indicates the start of the each time.

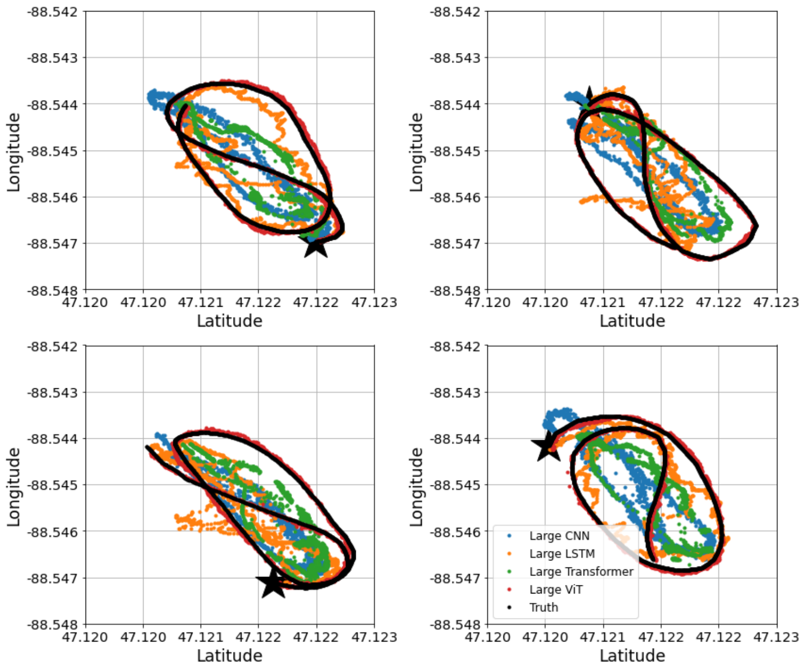


Figure A.3: Seconds 566 to 849 of test data using large regression networks where the star indicates the start of the each time.

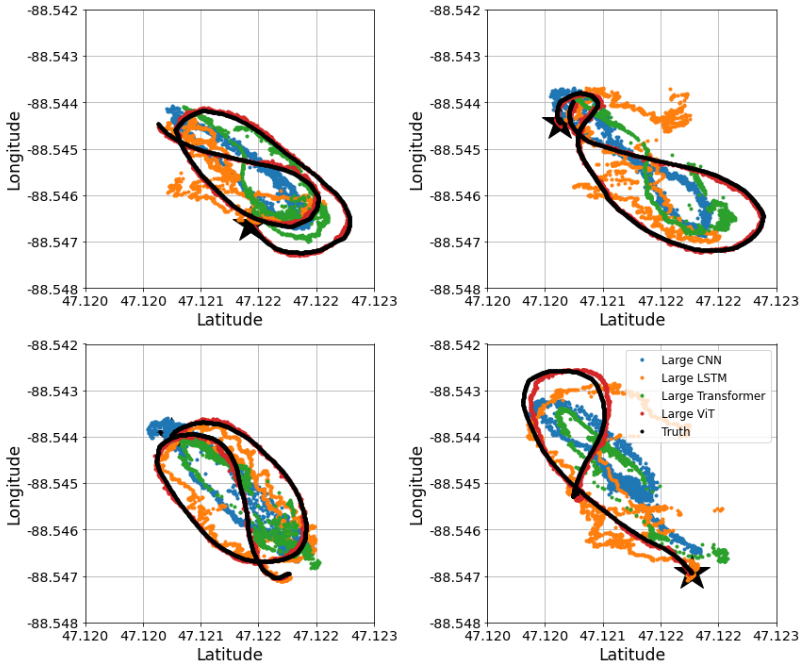


Figure A.4: Seconds 849 to 1132 of test data using large regression networks where the star indicates the start of the each time.

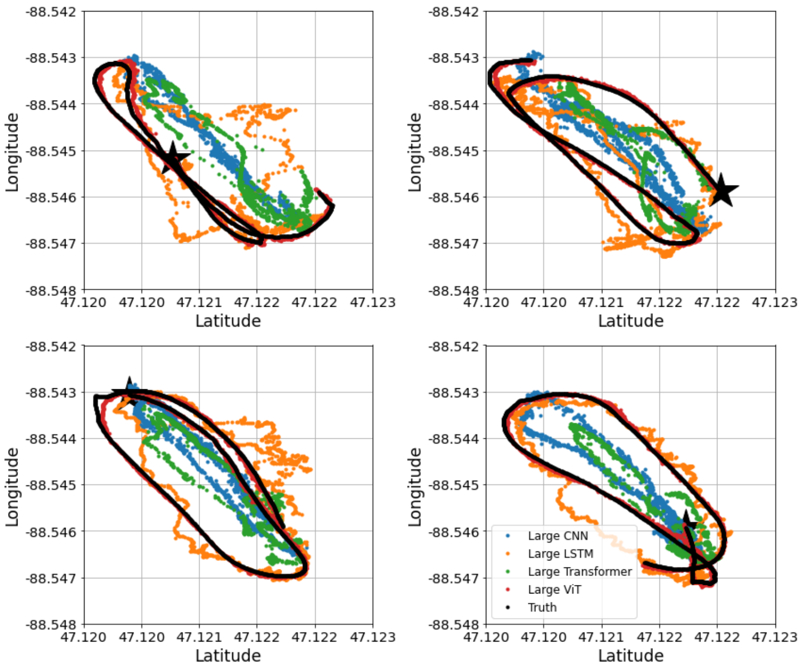


Figure A.5: Seconds 1132 to 1415 of test data using large regression networks where the star indicates the start of the each time.

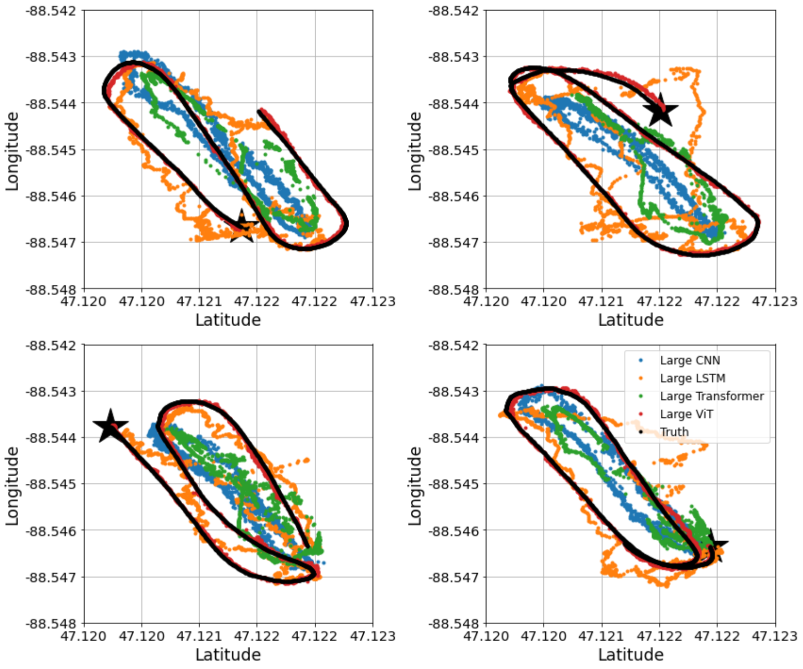


Figure A.6: Seconds 1415 to 1698 of test data using large regression networks where the star indicates the start of the each time.

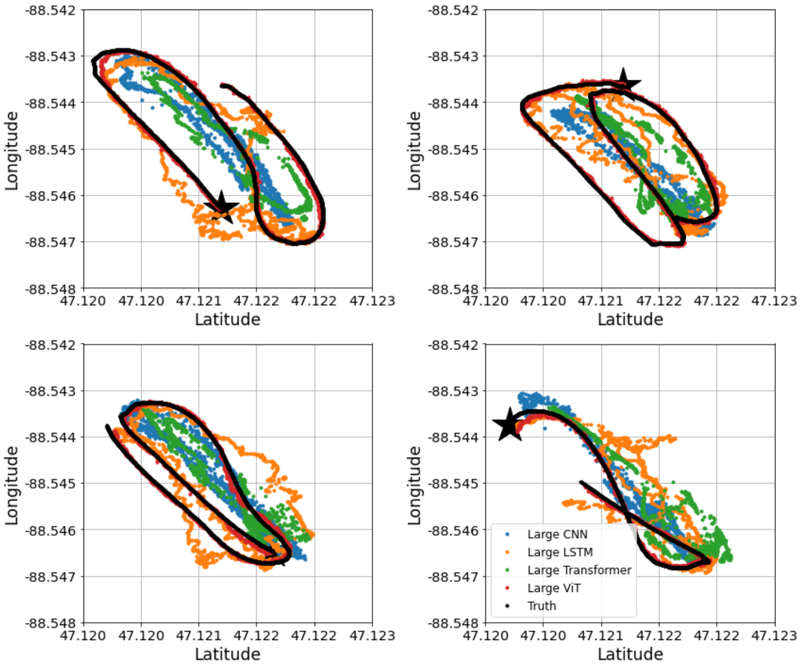


Figure A.7: Seconds 1698 to 1981 of test data using large regression networks where the star indicates the start of the each time.

Appendix B

Permissions for reuse

This appendix contains permissions granted by journal articles and conference proceedings to reuse each written material.

B.1 JASA

Dear Dr. Steven Whitaker:

You are permitted to include all or part of your published article in your thesis/dissertation, provided you also include a credit line referencing the original publication.

Our preferred format is (please fill in the citation information):

“Reproduced from [FULL CITATION], with the permission of AIP Publishing.”

If the thesis will be available electronically, please include a link to the version of record on AIP Publishing’s site.

Please let us know if you have any questions.

Sincerely,

B.2 POMA

Dear Dr. Steven Whitaker:

You are permitted to include all or part of your published article in your thesis/dissertation, provided you also include a credit line referencing the original publication.

Our preferred format is (please fill in the citation information):

“Reproduced from [FULL CITATION], with the permission of AIP Publishing.”

If the thesis will be available electronically, please include a link to the version of record on AIP Publishing’s site.

Please let us know if you have any questions.

Sincerely,

Barbara Rupp

B.3 MDPI

For all articles published in MDPI journals, copyright is retained by the authors. Articles are licensed under an open access Creative Commons CC BY 4.0 license, meaning that anyone may download and read the paper for free. In addition, the article may be reused and quoted provided that the original published version is cited. These conditions allow for maximum use and exposure of the work, while ensuring that the authors receive proper credit. Please visit https://www.mdpi.com/authors/rights for more information.