MACHINE LEARNING APPROACHES FOR CHARACTERIZING ELECTROMAGNETIC DUCTING WITHIN THE MARINE ATMOSPHERIC BOUNDARY LAYER

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Hilarie Sit

May 2021

© 2021 Hilarie Sit

ALL RIGHTS RESERVED

MACHINE LEARNING APPROACHES FOR CHARACTERIZING ELECTROMAGNETIC DUCTING WITHIN THE MARINE ATMOSPHERIC BOUNDARY LAYER

Hilarie Sit, Ph.D.

Cornell University 2021

This dissertation explores machine learning approaches for estimating the refractivity within the marine atmospheric boundary layer (MABL) under various electromagnetic ducting conditions. We use simulated radar propagation data that is representative of data that can be sparsely measured in practice. In conjunction with the sparse data collection scheme, a trained artificial neural network can be used to effectively characterize evaporation duct height (EDH) from the data, in real-time. We further show that Gaussian process regression (GPR) can accomplish this task, and produce uncertainty quantification on the EDH predictions, also in real-time. Finally, we show that a two-step deep learning model can classify and characterize different types of ducting conditions.

BIOGRAPHICAL SKETCH

Hilarie Sit grew up in Tracy, California. She completed her B.S. in Biological Engineering from Cornell University, where she was the recipient of the John McMullen Dean's Scholarship and the Hunter R. Rawlings III Presidential Research Scholarship. She continued at Cornell for her graduate studies, earning her M.S. and pursuing her Ph.D. in Remote Sensing under the guidance of Professor Christopher Earls. At Cornell, she was involved in various organizations and programs, and served as President of the Civil and Environmental Engineering Graduate Student Association, Civil and Environmental Engineering GPSA Field Representative, Engineering Peer Advisor, and Cornell Alumni Admissions Ambassador.

For my parents and my sisters, Attie and Echo

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Christopher Earls, for his guidance, patience, and unwavering confidence in me. I am so incredibly grateful for the opportunity to learn from him and for the profound impact that he has made in my life. I am also grateful to my committee members, Professor Austin Benson and Professor Derek Warner, for their support and guidance. I thank Professor Ben Cosgrove and Ms. Erin McKay for their key roles in supporting my academic and research endeavors prior to graduate school.

This research was funded though Office of Naval Research Grant N00014-19-1-2095, and I gratefully acknowledge ONR Division 331 and Dr. Steve Russell for the financial support of this work.

I would like to acknowledge Earls Group members and thank them for the memorable experiences and time shared inside and outside of Hollister 411. I especially thank John for the many enjoyable and insightful late-night conversations, Wensi for the adventures and encouragement to make bold decisions, Christophe for the teamwork in classes, and Harsh for the entertaining moments during my last semester. I am also grateful for the support from Civil and Environmental Engineering staff as well as the friendships and connections on the Civil and Environmental Engineering Graduate Student Association board.

I thank Rachel and Martin, who have been alongside me in Ithaca since the beginning of graduate school. Similarly, I thank Aleksa, Alice, Arjan, Feng, Florian, Inyoung, Sam, SiQing, Trey, and Zeki for their friendship and the unforgettable moments at Cornell. Finally, I would not be here without the support of Sarah, who believed in me when I needed it the most.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Performance of marine radar systems and other transmission systems that operate at microwave electromagnetic (EM) wavelengths within the marine atmospheric boundary layer (MABL) can be adversely impacted by the fluxes of heat, moisture, and momentum that occur in the atmospheric region above the ocean surface [45]. This dynamic activity can result in the formation of sharp refractive index gradients within the atmosphere that cause EM signal propagation to deviate from expected behavior. For example, a sharp decrease in the refractive index with an increase in altitude can lead to the formation of a "trapping layer", within which some EM signals are refracted backwards toward the ocean surface [37]. This phenomena, called atmospheric ducting, is problematic for radar systems because trapped signals are extended, leaving other regions without coverage (*i.e.,* radar holes) [48]. Understanding how the refractive index changes with altitude (described by the refractive index profile) within the MABL, in real-time, can assist radar operators in making adjustments to account for anomalous EM propagation behavior.

This dissertation explores and highlights the use of machine learning approaches as effective real-time methods for approximating parameters instantiating the refractive index profile, given sparsely sampled, noisy radar propagation data. Each chapter in this dissertation is either a reformatted published paper or manuscript under review, consisting of its own abstract, introduction, and conclusion. All references are collected in the bibliography section at the end.

Evaporation ducting is the most common type of EM ducting that occurs

within the MABL. The refractivity profile of an evaporation duct can be approximated as a log-linear function, within which evaporation duct height (EDH) is an important parameter defined as the height at which a horizontally propagating signal refracts in such a way as to follow the curvature of Earth [37]. In Chapter 2, we show that artificial neural networks can learn an approximate function mapping propagation data to EDH, thus providing an effective model capable of accurately predicting EDH in real-time. In Chapter 3, we show that Gaussian process regression can also solve the duct characterization problem, while also providing uncertainty quantification on EDH predictions. We additionally introduce Monte-Carlo importance sampling-like approaches to approximate the uncertainty of the predictions on noise-contaminated test observations. In addition to evaporation ducts, surface-based ducts and elevated ducts can also occur within the MABL. The latter two of these ducts can be modeled with a trilinear refractivity profile [59]. Since elevated ducts do not greatly impact radar propagation due to their height in the atmosphere, in Chapter 4, we expand the scope to include surface-based ducts and tackle the classification and characterization problem by using a two-step deep learning architecture.

CHAPTER 2

CHARACTERIZING EVAPORATION DUCTS WITHIN THE MARINE ATMOSPHERIC BOUNDARY LAYER USING ARTIFICIAL NEURAL NETWORKS

2.1 Abstract

We apply a multilayer perceptron machine learning (ML) regression approach to infer electromagnetic (EM) duct heights within the marine atmospheric boundary layer (MABL) using sparsely sampled EM propagation data obtained within a bistatic context. This paper explains the rationale behind the selection of the ML network architecture, along with other model hyperparameters, in an effort to demystify the process of arriving at a useful ML model. The resulting speed of our ML predictions of EM duct heights, using sparse data measurements within MABL, indicates the suitability of the proposed method for real-time applications.

2.2 Introduction

The marine atmospheric boundary layer (MABL) is a dynamic region in the lower troposphere that directly interacts with the ocean via turbulent transport of heat, moisture, and momentum [45]. Atmospheric conditions within the MABL, including its high temperature, pressure, and humidity gradients, influence the refractive index thus, affecting electromagnetic (EM) wave propagation characteristics. Rapid decrease in the refractive index with altitude can

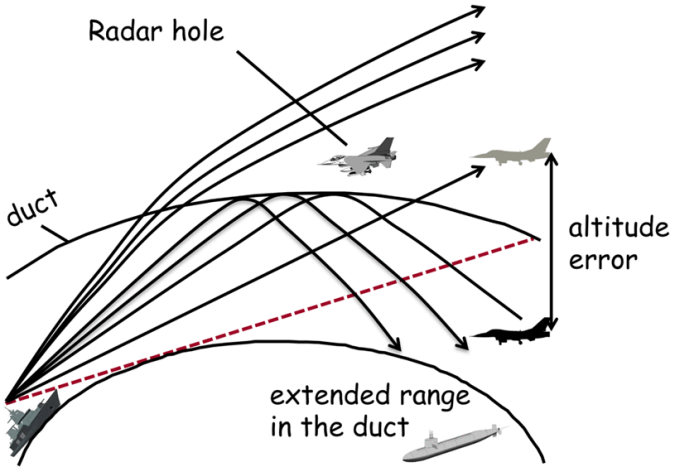


Figure 2.1: Effect of atmospheric ducting on radar operations [19]

cause EM waves to bend backwards toward the ocean surface [48]. This type of behavior is called atmospheric ducting and results in the trapping of EM waves near the ocean-free surface. Atmospheric ducting can influence radar performance by creating unexpected holes in the radar coverage, inaccurate altitude predictions, and false targets from clutter returns [48]. Thus, real-time identification and characterization of EM ducting within the MABL is important when assessing radar performance as shown in Figure 2.1.

The refractive index can be approximated by directly measuring atmospheric conditions within the MABL: [3] describe a relationship between the refractive index profile and atmospheric temperature, pressure, and humidity. Sensitivity of the various atmospheric and refractivity profile parameters in modeling EM propagation is detailed in [31]. Direct measurement of atmospheric parameters can be carried out at discrete altitudes by using radiosondes or rocketsondes, but such measurements are impractical when estimating MABL refractivity because data collection is sparse and implementation can be expensive [3].

Recently, literature on the MABL EM duct characterization problem has focused on identifying and improving optimization within the context of the inverse problem estimating refractivity from clutter (RFC) returns. RFC uses measured propagation loss from radar clutter returns to estimate the refractivity profile, which is then used to characterize ducts. [40] how that duct heights can be inferred from a slope fitted to backscattered power from sea surface clutter, by performing a nonlinear least squares inversion. One common approach to solving the RFC inverse problem employs multiple calls of a suitable forward solver to predict EM propagation, but this can be computationally expensive. Efforts to make RFC inversion more efficient can also be found in the literature, including matched-field processing approaches [20], Markov-chain Monte Carlo [60], and Markov state-space models [53]. A detailed review of RFC is available in [27].

In a different approach, [19] develop a simplified physics model by approximating a blurring operator (so as to approximate the effects of the MABL) with modes from proper orthogonal decomposition (POD) from field observations that are organized within an offline library. Manifold interpolation is performed on the sparse set of POD modes to construct a real time, online estimate for POD modes, which are needed within an optimization algorithm to infer duct parameters. Within the context of this method, the coverage is either sampled along a sinusoidal flight path or a linear path (consistent with a rocketsonde-receiver sampling system, RRSS). RRSS consists of a receiver attached to a solid rocket flown at a constant altitude, thus effecting coverage sampling in a reduced amount of time (as compared to a UAV flying the sinusoidal path). In this paper, as part of our machine learning (ML) MABL EM duct prediction, we will use two bistatic (i.e., separate transmitter and receiver locations) EM

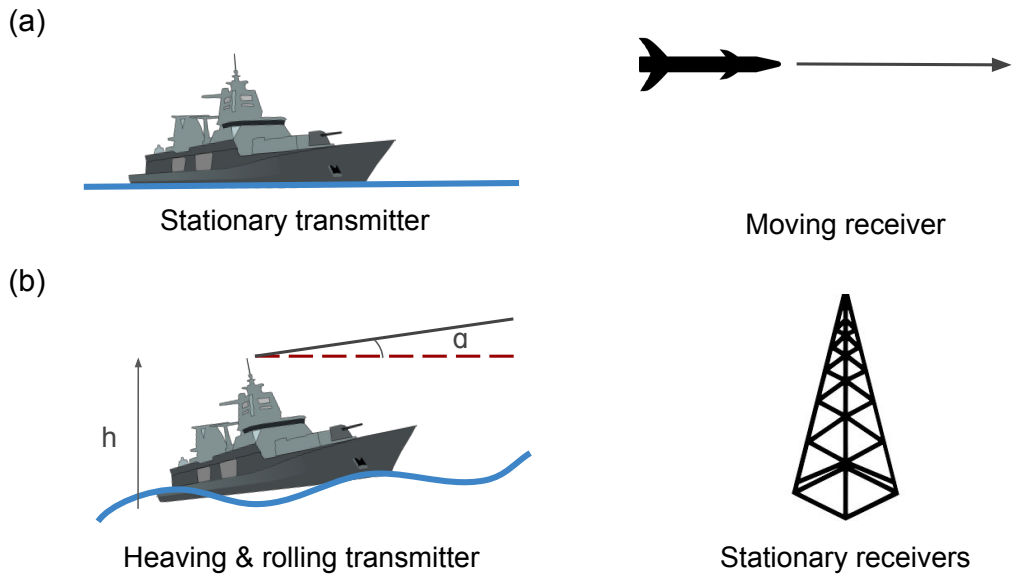


Figure 2.2: Two bistatic radar sampling schemes: (a) moving receiver and static transmitter height and angle, and (b) stationary receivers on tower and varying transmitter heights and angles

sampling schemes corresponding to: 1) a single moving receiver exploiting the practicality of RRSS, and 2) an array of stationary receivers mounted on a fixed tower (Figure 2.2).

ML algorithms have previously been employed within RFC inversion literature. [15] showed that least squares support vector machines (LS-SVM) can speed up inversion, although sacrificing some accuracy, by utilizing a pre-generated database of propagation losses and respective parameter values. Within the RFC literature, variants of artificial neural networks (ANNs), primarily radial basis function neural networks, were considered but discounted in favor of the convex property of LS-SVM. A comparison of both approaches can be found in [57]. With the recent surge in popularity of machine learning

algorithms and deep learning models, enhancements to ANN architecture and optimization has led to increased model flexibility, stability, and efficiency. Recent literature has seen a revival in the use of ANNs for the duct characterization problem within an RFC context. [51] introduce a hybrid model using parameter estimations from neural networks as initializers in genetic algorithms to refine the inversion. [24] use deep neural networks to estimate refractivity parameters from calculated sea clutter power.

In the present paper, we propose using supervised machine learning with ANNs to directly estimate the function that maps propagation factors into duct height. ANNs are universal approximators with the ability to represent complex nonlinear functions using a few parameters [44]. Trained models can evaluate novel inputs and provide fast predictions, thus making ANNs potential candidates for real-time duct height predictions. We train and evaluate separate ANNs to estimate duct height using the aforementioned two bistatic sampling schemes (not RFC). Propagation data needed for training the ANNs are obtained using a fast split-step parabolic equation (SSPE) solver, as described in the sequel.

This paper is outlined as follows. We begin by describing the physical problem and its domain. We subsequently introduce two bistatic sampling schemes along with the SSPE forward solver used in the numerical simulations needed to generate training and test data. We then provide a background on supervised machine learning with multilayer perceptrons and describe our specific machine learning regressive model. Finally, we report our results and provide a discussion of limitations in the model selection process.

2.3 Forward Model

2.3.1 Evaporation Ducts

Evaporation ducts are the most prevalent duct type within the MABL [2]. These shallow ducts form as a result of trapping gradients that change the refractivity profile over the ocean surface [37]. The refractive index, n, is the ratio of EM propagation velocity in free space to that within a given medium. To better reflect the small changes in the refractive index that characterize the MABL, we employ a quantity known as refractivity, $N = (n - 1)10^6$ [48]. The modified refractivity profile for evaporation ducts can be modeled as a log-linear function that is assumed to be homogenous in the range direction:

$$M(z) = M_0 + c\left(z - z_d ln\left(\frac{z + z_0}{z_0}\right)\right) \tag{2.1}$$

where M_0 is base refractivity, c is critical potential refractivity, z_0 is aerodynamic surface roughness of the ocean, z_d is the duct height, and z is the altitude [19]. Evaporation duct heights are commonly observed to be between 0m and 40m in altitude [37]. With the following parameters as constants: M_0 = 428.89M-units, c = 0.13M-units/m, and z_0 = 0.00015m, we are interested in the inversion for evaporation duct height, z_d.

2.3.2 Problem Domain and Description

In the present work, a horizontally polarized Gaussian antenna pattern with radar signal frequency of 9.3 GHz, corresponding to the X-band, is transmitted through a rectangular 2D problem domain with a maximum altitude of 113m and range of 50km. The associated SSPE problem domain is discretized using grid spacings of 0.1m and 40m, in altitude and range, respectively. Propagation factors within the problem domain, collectively called the coverage diagram, are the observables, or feature vector, within the context of this work. In radar calculations, *propagation factors* account for environmental effects and surface roughness and can be obtained by scaling the electric field magnitude with that observed in free space [41].

Rather than using the entire coverage diagram to train our ANNs (as it is impractical to measure such large areas), propagation factors are sampled so as to be consistent with two bistatic radar sampling schemes mentioned earlier (Figure 2.2). A bistatic radar system uses different antennas for transmitting and receiving signals [48]. We alternatively consider a moving receiver, as well as an array of stationary receivers mounted on a tower, to examine how variations in data collection may potentially influence ANN duct height estimates. For the moving receiver, the transmitter antenna is stationary with an antenna angle of $\alpha = 0^\circ$ (with respect to the horizontal) and antenna height of $h = 10$m. Propagation factors are sampled at 250 points along a horizontal path over the range, 5-15 km, positioned at 21m above the ocean free surface.

In the case of the array of stationary receivers, we consider two different configurations for the transmitter antenna, deterministic rocking and stochastic rocking, to enable the investigation of any influence from patterning within the

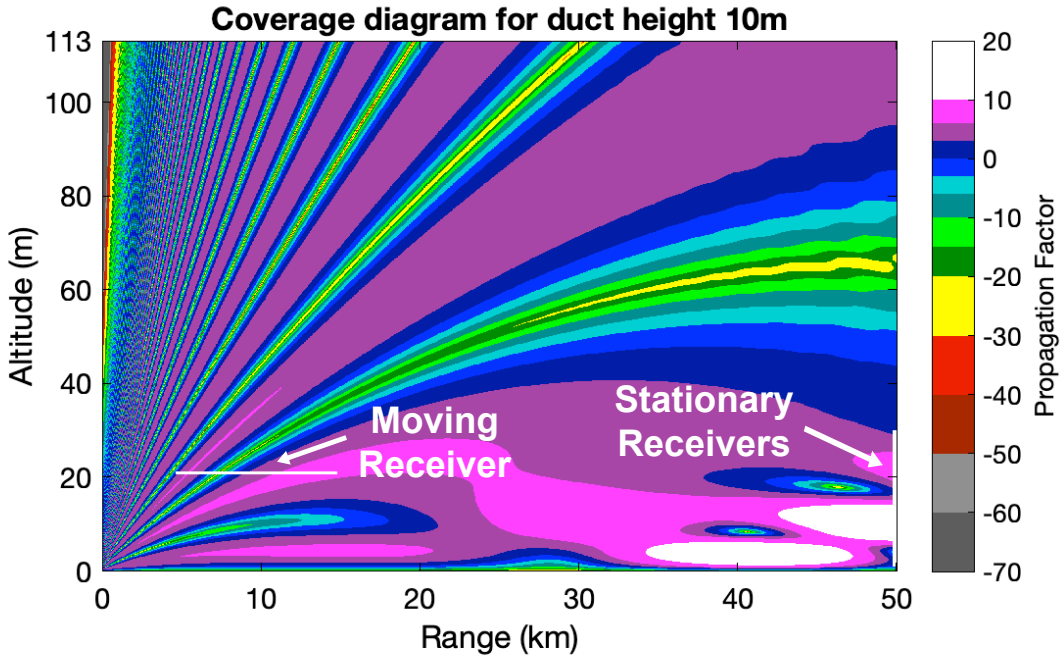


Figure 2.3: Sampling paths on representative coverage diagram

resulting data sets. For deterministic rocking, six observations are collected using deterministic combinations of transmitter antenna angles $\alpha = -0.5^\circ,\ 0^\circ,\ 0.5^\circ$ and heights $h = 20\text{m},\ 30\text{m}$ to simulate a transmitter antenna on a rocking ship. In this arbitrary setup, the expected heave of the ship is 10m with rolls of $\pm 0.5^\circ$. By altering the antenna height and angle in steady increments, this configuration leads to a defined pattern in the dataset, which is in contrast to the stochastic rocking configuration. Stochastic rocking is considered by collecting five observations from uniformly distributed antenna angles $\alpha \sim U[-0.5^\circ, 0.5^\circ]$ and heights $h \sim U[20\text{m}, 30\text{m}]$. Propagation factors are sampled using 30 evenly spaced receiver antennae distributed along on a vertical mast, 1-30m in the altitude direction, positioned 50km down range from the rocking transmitter. Observations are concatenated to form observation arrays of lengths 180 and 150,

Table 2.1: Description of the Six Sampling Cases

Case 1:	Stationary transmitter, moving receiver (RRSS)
Case 2:	Deterministically rocking transmitter, stationary receivers
Case 3:	Stochastically rocking transmitter, stationary receivers
Case 4:	Stationary transmitter, moving receiver with noise (RRSS)
Case 5:	Deterministically rocking transmitter, stationary receivers with noise
Case 6:	Stochastically rocking transmitter, stationary receivers with noise

for the deterministic and stochastic rocking, respectively. Please refer to Figure 2.3 for a depiction of the two sampling paths. In order to form surrogate experimental observations, the modeled datasets are also contaminated with Gaussian noise of $N(0, \sigma)$; $\sigma = 0.1\|x\|_\infty$, where standard deviation is 0.1 times the absolute value of the largest propagation factor in an observation array, to simulate severe electronic sensor noise [19]. Overall, we assess six cases described in Table 2.1.

2.3.3 SSPE Solution

EM propagation within the MABL can be modeled using the parabolic equation approximation to the Helmholtz wave equation. [36] developed PETOOL, a MATLAB®-based software, for analyzing EM propagation over variable terrain using the split step parabolic equation (SSPE) method. SSPE solves the wide angle parabolic equation as an initial value problem. The initial condition is specified by injecting EM fields into the domain at the transmitter's location, in order that a series of Fourier transformations can be performed to propagate the fields down range. PETOOL approximates the Sommerfeld ra-

diation condition at the top of the problem domain by extending the domain's altitude and applying a Hanning window to remove the non-physical reflections that would ordinarily result from truncating our domain altitude at 113m. The lower domain boundary is the ocean free surface, which is assumed to be smooth. To enforce continuity of the tangential components of the electric and magnetic fields at this boundary, the ocean surface is assumed to be a finite conductor with a homogeneous dielectric constant that can be calculated by using the semi-empirical Debye expression [41]:

$$\epsilon(\omega) = \epsilon_{ir} + \frac{\epsilon_0 - \epsilon_{ir}}{1 - i\omega\tau} + \frac{i\sigma}{\omega\epsilon_0} \tag{2.2}$$

where the far-infrared dielectric constant of water, ϵ_{ir}, is 4.9, and the relaxation time, τ, ionic conductivity, σ, and static dielectric constant of sea water, ϵ_0, are obtained using thermodynamic data consist with the South China Sea - 100% humidity at ocean surface, surface temperature of 29.7 $^{\circ}C$, and ocean salinity of 35 ppt. We use code adapted from PETOOL that specifies this Leontovich surface impedance condition at the lower boundary [21].

2.4 Machine Learning Regressive Model

2.4.1 Background

The goal of supervised ML is to learn an hypothesis $h : X \rightarrow Y$ that maps inputs to outputs by only sampling points from some underlying distribution $(x_i, y_i) \in X \times Y$ [44]. The ML task of performing MABL duct height inversion is a

regression problem, because duct height is a continuous variable. The parameters of the hypothesis (*i.e.,* our ML regression network weights and biases) can be learned by minimizing a loss function on training pairs (*i.e.,* known sets of propagation samples at given duct heights). Mean Squared Error (MSE), which measures the average squared difference between the predicted value and true value, is typically minimized when training ML regression networks [26]:

$$J = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{2.3}$$

where $\hat{y}$ is the output from the model (*i.e.,* the predicted duct height from the ANN) and y is the label (*i.e.,* the true duct height). We are interested in employing artificial neural networks, a class of machine learning algorithms, for this task because of their universal approximation property and quick evaluation speed.

Multilayer perceptrons (MLP) are feedforward ANNs, consisting of an input layer, hidden layer(s), and an output layer. The input layer accepts a d-dimensional feature vector $x \in \mathbb{R}^d$ (in our case, samples of propagation factors) where each dimension, d, corresponds to a neuron at a discrete spatial point (corresponding to a propagation factor measurement), such that each input neuron is connected to every neuron in the subsequent hidden layer by weighted edges within the network graph (Figure 2.4). The collection of network weights $w \in \mathbb{R}^{k\times d}$ are summed together along with an offset term $b \in \mathbb{R}^k$; thus the value at a given hidden neuron can be written as:

$$h_k = \sum_{d} w_{k,d}x_d + b_k \tag{2.4}$$

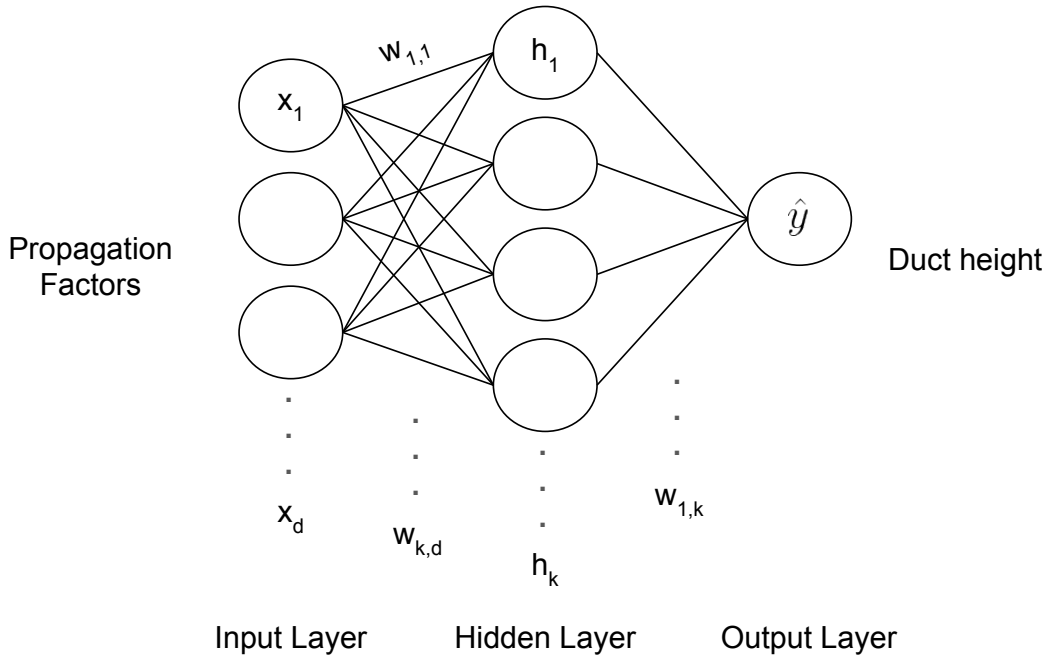


Figure 2.4: MLP architecture

where d is the input dimensionality and k is the number of hidden neurons. The collection of weights and biases is referred to as the networks *parameters*. Non-linearity is added to the model by applying activation functions independently to the value at every hidden neuron. Common activation functions include sigmoid, hyperbolic tangent and rectified nonlinear unit (ReLU), with ReLU preferred due to its computational efficiency and reduction of the vanishing gradient problem [34]:

$$ReLU(h) = max(0, h) \tag{2.5}$$

The next hidden layer is connected to the previous hidden layer in the same manner as that of the input layer (*i.e.,* it is fully connected). The number of

hidden layers and neurons in each layer are referred to as *hyperparameters* (not to be confused with the network parameters in equation 2.4). Hyperparameters are values that cannot be learned by the model and thus are carefully selected by the analyst prior to training, to achieve maximum predictive performance. In order to allow the MLP to predict a single continuous variable, the output layer must consist of a single neuron with no activation function: $\hat{y} \in \mathbb{R}$.

For MLPs, the MSE loss function is commonly minimized using a variant of gradient descent, an iterative, first-order gradient method. Higher-order optimization methods and quasi-Newton methods are not typically employed, as these methods can be slow to implement. In gradient descent, network parameters, w, are randomly initialized at the start of training and subsequently adjusted to improve the network's predictive power by taking an updating step in the opposite direction of the loss function's gradient with respect to the parameters, $\nabla J(w_t)$ [44]:

$$w_{t+1} := w_t - \alpha \nabla J(w_t) \tag{2.6}$$

where learning rate, α, governs the step sizes of each iteration, and thus embodies a trade-off between convergence and speed. If α is too small, then the model takes too long to converge and conversely, if α is too large, the model can diverge. Gradients in neural networks can be efficiently calculated using the two-part *backpropagation* algorithm. n the forward pass, inputs from the training set are fed into the neural network, and network activations are calculated and cached. These values are then retrieved during the backward pass to calculate the gradients with respect to every parameter using the chain rule from calculus, which describe the interrelationship between parameters from the back of

the network to the front [30]. Gradients are obtained for each example in the training set, and subsequently averaged to get $\nabla J(w_t)$ for the update in equation 2.6.

The loss function landscape of neural networks is typically non-convex, and thus, a stochastic variant of gradient descent can help the model escape local minima and saddle points as well as prevent overfitting. In stochastic gradient descent (SGD), rather than calculating the gradient of the loss function using the entire training set, the direction at every iteration is determined by a randomly selected training example [44]. SGD is expected to behave similarly to gradient descent, but introduces noise, which could potentially limit convergence rate [7]. A solution to this problem is using the averaged gradient from a small number of training examples, referred to as a mini-batch, to update the parameters in order to reduce the variance of the gradient approximations [8]. In a version of mini-batch gradient descent, the training set is randomly shuffled and mini-batches of training examples are sampled sequentially, without replacement, for the update. The training set is then reshuffled after every epoch, which is defined as the occurrence of every training example having been passed through, and subsequently updated, the model.

[28] introduced Adam optimization, a stochastic optimization scheme that incorporates "momentum" with individual adaptive learning rates for parameters by using estimates of the gradient's first and second moments. The bias-corrected exponential moving average of the gradient, with decay parameter $\rho_1 = 0.9$, is used as the first moment estimate, and the bias-corrected exponential moving average of the gradient squared, with decay parameter $\rho_2 = 0.999$, is used as the second moment estimate. Updates are performed until conver-

gence. We trained the MLPs with mini-batch Adam optimizer as the learning approach in the current work.

In supervised learning, the model is trained using a subset of the complete, labeled dataset, in order to measure model generalization when the network is applied to the excluded set. A statistical technique used to obtain a principled metric of model performance on unseen data is *k-fold cross-validation*, where the dataset is split into k folds, and each fold is used as the testing set for k independent training sessions [44]. Common k values for k-fold cross-validation are $k = 5$ and $k = 10$, as these are shown to appropriately balance the bias-variance tradeoff and reduce computational time as compared to $k = n$, in the leave-one-out cross validation technique, where n is the cardinality of the dataset [26]. The losses from each fold are then averaged to achieve an aggregated cross-validation score, which can be used to compare different hyperparameter instances to define an appropriate MLP architecture.

2.4.2 MLP Model Selection

The specific MLP model design and architecture used in each of the six cases from Table 2.1 is influenced by the dimensionality of the dataset and involves the selection of appropriate hyperparameters, which include the number of hidden layers, number of neurons, number of epochs employed during training, initial learning rate, and batch size. Hyperparameter selection is a complex problem that unfortunately can require prior experience, as mentioned in [57]. Barring prior experience, the analyst typically chooses hyperparameters via trial and error, so as to achieve a sufficient test loss or to satisfy some threshold cross

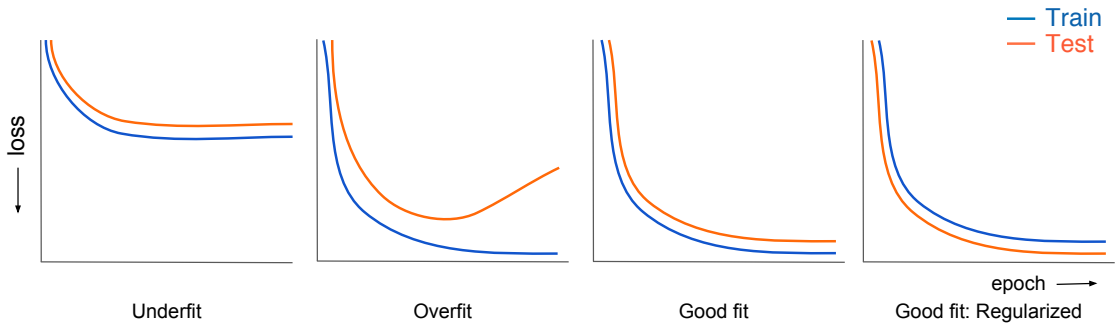


Figure 2.5: Performance indicators for diagnosis of train and test loss behavior in training artificial neural networks

validation score. We attempt to demystify this process by discussing important concepts, summarized in Figure 2.5, and the intuition behind our model selection process.

A description of the four canonical network training behaviors from Figure 2.5 can be summarized as follows.

Underfitting. Underfitting occurs when the model fails to sufficiently learn the underlying function approximation. When both training and test losses remain large, it is possible that the model has converged to a non-optimal local minimum or saddle point within the loss function output space. Underfitting can also be a result of having too few parameters to represent the pattern in the dataset. Increasing model complexity by increasing number of neurons as well as decreasing learning rate can help resolve underfitting. The model can also be underfitted if the selected number of epochs is too small, and the model has not yet converged since it has not seen the training examples enough times.

Overfitting. Overfitting occurs when model performance on the training set far exceeds the performance on the test set [44]. This phenomenon is evi-

dent when test loss plateaus or increases as training loss decreases. For a small dataset, overfitting is a primary concern, because the number of parameters in the model may be too numerous for effective generalization to other, previously unseen, data. Methods to mitigate overfitting include stopping the training at an earlier epoch, decreasing the number of parameters within the model (*e.g.*, by either decreasing the neuronal cardinality within a layer and/or reducing the number of layers), and adding regularization.

Regularized. Regularization adds restrictions to the model, to address the overfitting problem, by forcing the network to learn a less complex representation by penalizing it for learning unwanted behaviors. A model with lower test loss than train loss is possible when using regularization techniques. Adding noise to the input is related to adding regularization to the weights of the ANN [5].

Spiky convergence. Spiky convergence is common when using stochastic methods, such as using a mini-batch to update the parameters. Decreasing the learning rate or increasing the batch size can give smoother convergence. However, some stochasticity can be beneficial for escaping saddle points or local minima; thus, care should be exercised when reducing its influence.

Trial and error methods based on intuition alone do not produce the best hyperparameters when designing a regressive MLP. Rather, we use a more systematic approach by performing grid search in a restricted hyperparameter space. In this grid search, the number of hidden layers is set to one in all six cases and the number of epochs is chosen to be sufficiently large to allow for convergence for each case, as gauged by the loss function, equation 2.3. For each of the noise-less cases (1-3), we explored 45 different model settings with the follow-

Table 2.2: Selected Models from Grid Search

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Input Dimensionality	250	180	120	250	180	120
Hidden Neurons	200	180	195	100	70	95
Epochs	1000	2500	2500	400	500	500
Learning Rate	1e-3	1e-3	1e-3	1e-4	1e-3	5e-4
Batch Size	16	4	4	32	32	32
Loss mean / CV score (m^2)	1.34e-2	2.41e-1	2.75e-1	3.09e-1	2.78e-1	3.52e-1
Loss s.t.d. (m^2)	8.42e-2	1.69e-1	1.47e-1	7.74e-2	9.89e-2	2.58e-1
Accuracy mean (%)	98.7	74.3	80.7	70.1	69.2	74.1
Accuracy s.t.d. (%)	2.67	19.1	5.23	3.46	15.0	5.56

ing hyperparameter combinations: hidden neuron cardinality within the single layer [180, 185, 190, 195, 200], learning rate [1e-4, 5e-4, 1e-3], and batch size [4, 8, 16]. For each of the noise-contaminated cases (4-6), we explored 60 different hyperparameter combinations: hidden neuron cardinality within the single layer [70, 75, 80, 85, 90, 95, 100, 105, 110, 115], learning rate [1e-4, 5e-4, 1e-3], and batch size [32, 64]. The model with the lowest 5-fold cross validation (CV) score from the training procedure detailed below is subsequently selected for each case as shown in Table 2.2. Other methods for hyperparameter optimization, besides grid search, include random search and bayesian optimization [4].

2.4.3 MLP Network Training

Datasets are constructed by sampling propagation factors corresponding to evaporation duct heights ranging from 2m to 40m, in half meter increments.

The dataset is randomly split: 80% for training and 20% for testing. For cases 4-6, the inclusion of noise into the smaller dataset increases the difficulty of the problem. Overfitting is a common problem when training neural networks on a small dataset, as is the case in our problem. To address overfitting to the noise in cases 4-6, the training set is augmented by applying the Gaussian noise, $N(0, \sigma);\ \sigma = 0.1\|x\|_\infty$, to each training example instance (*i.e.,* at each considered duct height) 100 times. While the same noise treatment is applied to the observations in the test set, the test set is not augmented. Our multilayer perceptions were implemented using Tensorflow [1]. The model weights are randomly initialized from a truncated gaussian distribution ($\mu = 0,\ \sigma = 0.01$) and biases are initialized as zeros.

MSE (equation 2.3) and accuracy are used as evaluation metrics when comparing MLP performance. Accuracy is normally a classification metric, but it can be informative to quantify instances of good predictions. In this paper, accuracy is defined as

$$A = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(|\mathrm{y} - \hat{\mathrm{y}}| < \delta) \tag{2.7}$$

where $\mathbb{1}(\cdot)$ is the indicator function and $\delta = 0.5$m. This metric conservatively scores a difference of 0.5m, or more, between the prediction and true duct height as incorrect.

During training, test loss is calculated after every five epochs and model parameters are saved for the best test loss. Both training and evaluation are performed using 12-cores of Intel Xeon E5 microprocessor having a clock speed of 2.7 GHz. Timing is calculated using the Python time.clock() method, which

measures processor system time in seconds.

2.4.4 Evaluation

To determine the performance of the chosen model design and architecture, the entire dataset is once again split randomly into a training and test set, and the best models for each case from the 5-fold cross-validation hyperparameter exploration, is trained on the training set from scratch, and subsequently evaluated on the remaining test data. For each of the six cases, we report the MSE test loss, test accuracy, and absolute error of the worst prediction by the model. Training time is calculated by summing all forward pass and backpropagation calls with the training set, for the specified number of epochs in Table 2.2, and evaluation time is calculated as the average time to evaluate the entire test set.

Training and test curves for both loss and accuracy during model evaluation are shown in Figures 2.6 and 2.7. A noticeable characteristic of the curves are the spikes that result from the stochasticity with using small batch sizes (bolded lines show a smoothed curve), which can help the model achieve better performance. Further inspection of the curves in relation to Figure 2.5 show that case 3 is overfitted: the validation loss is substantially higher than the training loss and the test accuracy is lower than the training accuracy. Table 2.3 also shows that the highest difference of the worst prediction and its true label occurs in case 6: the absolute error is 2m, with the model predicting 38m for the true duct height of 40m. The inclusion of dataset endpoint(s), 2m and 40m, in the test set means that the ANN has to extrapolate beyond the range of the given training data, which can yield worse predictions. Variability in model selection and

Table 2.3: Evaluation of Models

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Loss (m^2)	1.24e-2	4.81e-2	6.16e-1	3.22e-1	1.75e-1	4.57e-1
Accuracy (%)	100	93.8	62.5	68.8	75.0	62.5
Absolute Error (m)	0.305	0.568	1.82	1.26	0.829	2.00
Training time (s)	8.42	67.9	68.2	143	169	173
Evaluation time (s)	8.99e-4	8.83e-4	8.98e-4	9.01e-4	8.88e-4	8.93e-4

evaluation should be expected due to variability in parameter initializations and random dataset shuffling, splitting, and batching.

2.4.5 Discussion

A single hidden layer MLP, with its ability to provide quick and reasonable approximations of duct heights given propagation factors, is sufficient for the MABL duct height estimation task. We found that deep neural networks, which include two or more hidden layers, do not improve performance and can even hinder performance by contributing to the overfitting problem. Regularization techniques, to address the overfitting problem in neural networks exist, but these are not suitable in our problem. Very deep models can benefit from dropout [49], which involves randomly dropping out neurons to force the network to learn a more general representation with fewer parameters. Another technique is batch normalization [25], which involves normalizing inputs into a layer with the mean and standard deviation of batches, but this is not recommended for small batch sizes. It is pointed out that the limitations of small datasets within the current study, imposed on the models, are representative of

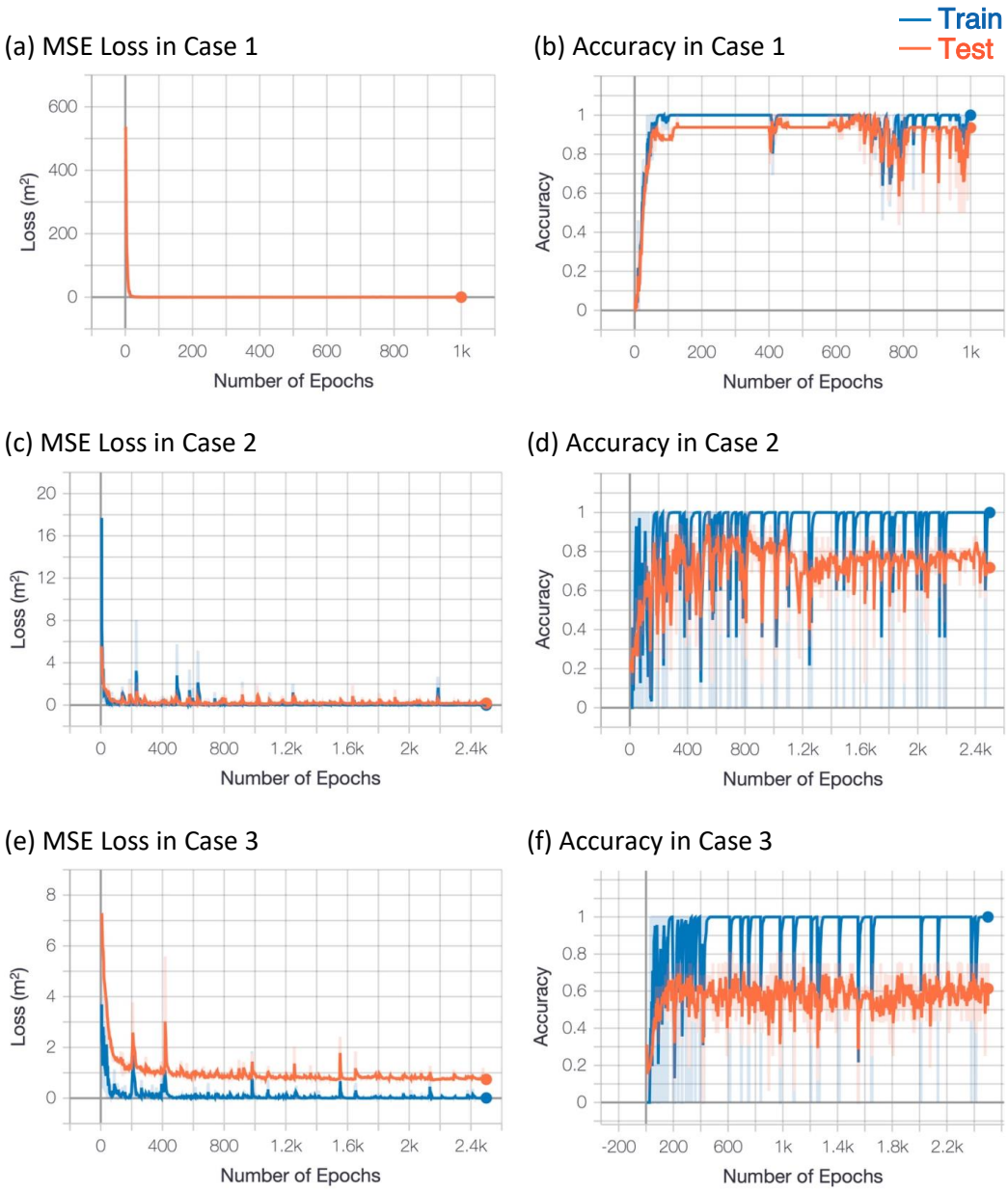


Figure 2.6: MSE loss and accuracy for both the training set (blue) and test set (orange) during evaluation of the noise-free cases 1-3

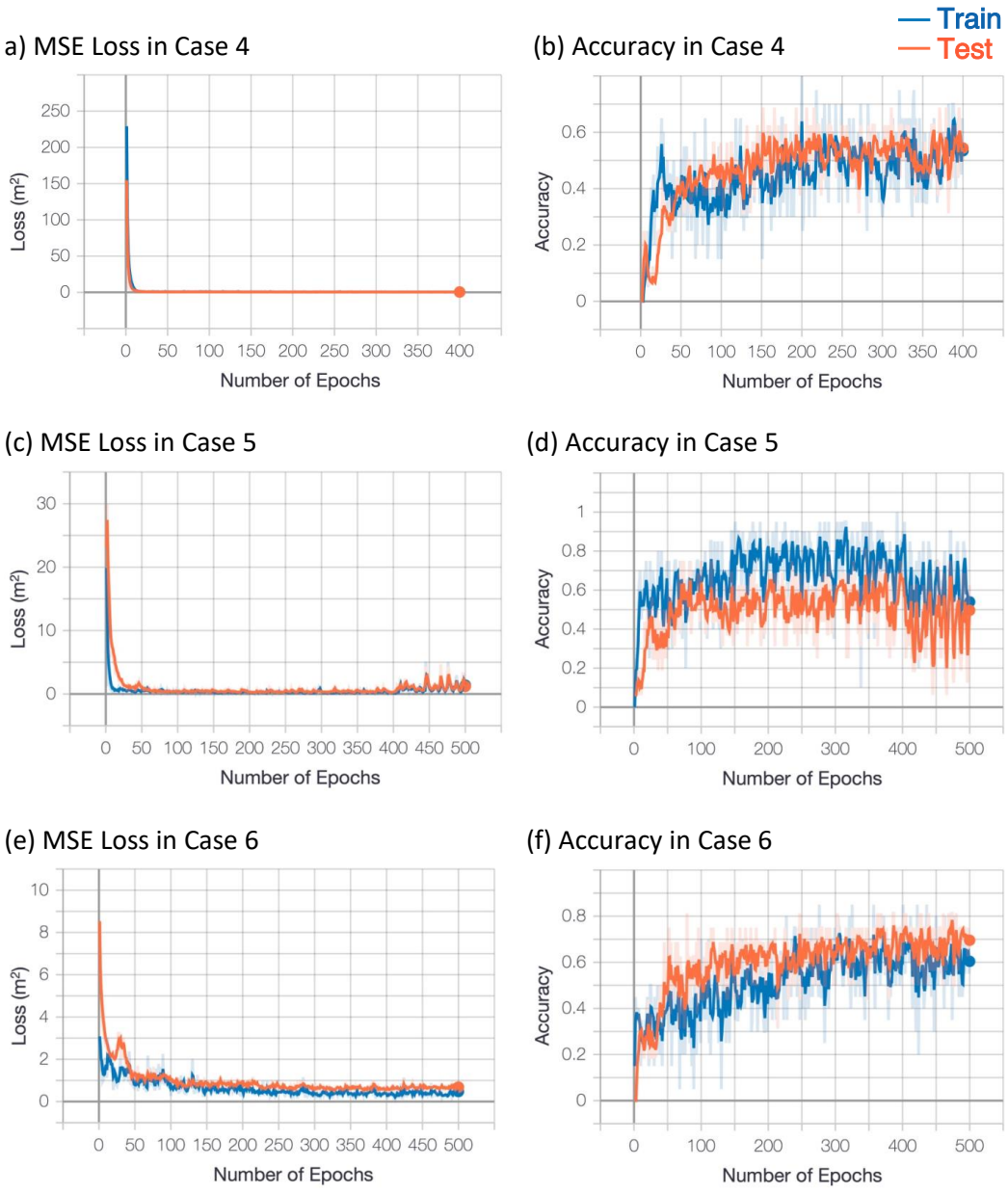


Figure 2.7: MSE loss and accuracy for both the training set (blue) and test set (orange) during evaluation of the noise-contaminated cases 4-6

problems in the physical world, where data are frequently difficult to obtain. This situation represents a challenge for many common machine learning techniques, when applied to problems of physics.

It is noted that the more formal nested cross-validation procedure is not performed within the current study due to the small dataset size. Instead, we evaluate model performance by re-splitting the dataset and re-training the model from scratch, but in general, re-training the model on the entire dataset and evaluating performance on a separate unseen set is better. Nested cross-validation addresses the potential problem of leaking validation set information into the model during the model selection process [10]. However, to do this, the dataset needs to be split into three sets for training, validating and testing, which our modeling context cannot afford (*i.e.,* we have far too few training pairs to accommodate this).

2.5 Conclusion

Artificial neural networks show promising results in characterizing evaporation ducts within the MABL. We have shown that single hidden layer MLPs can sufficiently estimate duct heights from propagation factors under different bistatic sampling schemes, indicating the robustness of MLPs on these different data collection methods. It is noted that the results here were based on the use of surrogate data and involved certain idealizations (*e.g.,* smooth ocean surface, constant refractivity profile in range, etc.). However, it is expected that the proposed methods will generalize to even more realistic cases, but the training data must reflect these less idealized conditions.

A highlight of the present work is that MLPs are extremely efficient during evaluation, using much less than a millisecond of CPU time to predict duct height from data. As a result of the observed accuracy and precision along with evaluation speed, we conclude that ANNs are excellent candidates for real-time prediction of ducting conditions in the MABL.

CHAPTER 3

GAUSSIAN PROCESS REGRESSION FOR ESTIMATING EM DUCTING WITHIN THE MARINE ATMOSPHERIC BOUNDARY LAYER

3.1 Abstract

We show that Gaussian process regression (GPR) can be used to infer the electromagnetic (EM) duct height within the marine atmospheric boundary layer (MABL) from sparsely sampled propagation factors within the context of bistatic radars. We use GPR to calculate the posterior predictive distribution on the labels (*i.e.,* duct height) from both noise-free and noise-contaminated array of propagation factors. For duct height inference from noise-contaminated propagation factors, we compare a naïve approach, utilizing one random sample from the input distribution (*i.e.,* disregarding the input noise), with an inverse-variance weighted approach, utilizing a few random samples to estimate the true predictive distribution. The resulting posterior predictive distributions from these two approaches are compared to a "ground truth" distribution, which is approximated using a large number of Monte-Carlo samples. The ability of GPR to yield accurate and fast duct height predictions using a few training examples indicates the suitability of the proposed method for real-time applications.

3.2 Introduction

Turbulent transport of momentum, moisture, and heat between the ocean and the atmosphere characterizes the marine atmospheric boundary layer (MABL), which is the region of the lower troposphere that directly contacts the ocean surface [45]. Spatiotemporal inhomogeneity in temperature, pressure, and humidity within the MABL can result in anomalous propagation of electromagnetic (EM) waves by altering the refractive index, which is the speed of light in medium relative to that in vacuum, within this region. Rapid decrease in the refractive index with increasing altitude can create a "trapping" layer where EM waves are refracted back toward the ocean surface. This phenomenon, called *atmospheric ducting*, can impact performance of radar and communication systems by causing unexpected holes in coverage, inaccurate measurement of elevation angle, and extension of radar horizon [48]. Thus, it is of great interest to devise real-time methods that can accurately identify and characterize these EM ducts, so that, for instance, radar operators can be informed about their system's expected performance in real-time.

A method for estimating duct characteristics is by calculating the refractive index using direct measurements of atmospheric conditions within the MABL. The refractivity profile, which is related to the refractive index, can be written as an empirical equation parameterized by atmospheric temperature, pressure, and humidity that can be measured via radiosondes or rocketsondes [3]. However, direct measurements of atmospheric conditions are necessarily sparse and implementation can be expensive, rendering this method impractical for real-time estimation of MABL refractivity. Other methods for estimating the refractive index include using GPS [32] and LIDAR [56] measurements. However,

these methods are not practical in the context of the duct characterization problem due to its dependence on favorable satellite alignment over the horizon and favorable weather conditions.

More recently, refractivity from clutter (RFC) methods have gained traction in the literature. RFC methods use radars to measure clutter (*i.e.*, backscattered power from the rough ocean surface); thus employing backscattered returns to estimate the refractivity profile. In RFC, the forward solver is called multiple times to predict clutter under certain ducting conditions, and these solutions are compared with new clutter observations to infer refractivity profiles. Such methods frequently use the forward solver and so a full, 3D solution of Maxwell's equations, can be computationally expensive. The Fourier split-step parabolic equation (SSPE) is commonly used as a fast and efficient forward model, however, even SSPE solutions may prove unwieldy for RFC inversions. As a result, efforts have focused on improving efficiency in optimization within the context of the RFC inversion: approaches in the literature include nonlinear least squares [40], matched-field processing approaches [20], Markov-chain Monte Carlo [60], Markov state-space models [53], particle swarm optimization [54], opposition-based learning [58], and dynamic cuckoo search [61]. A review of RFC methods is available in [27].

Apart from RFC methods, approaches for constructing simplified forward models using observations obtained by sparsely sampling EM power within the MABL have emerged. [19] utilize blurring operators to approximate effects of the MABL, so that "just in time" estimates, for inferring duct parameters, may be obtained by manifold interpolation within a library of sparse proper orthogonal decomposition (POD) modes calculated offline from field observations.

These field observations are sparsely sampled along a sinusoidal UAV flight path or a linear rocketsonde flight path. [21] bypass the full forward model by decomposing the governing partial differential equation into few propagating trapped normal modes and solving an easier optimization problem posed on normal mode subspaces. In the context of this method, coverage is sparsely sampled, so as to be consistent with a bistatic case, in which a receiver is located downrange of a hypothetical transmitter.

In prior work, the current authors show that artificial neural networks (ANNs) can accurately and efficiently predict duct height from sparsely measured EM propagation factors [46]. Similar to the above methods, the authors simulate coverage diagrams for duct heights of interest and utilize a series of sparse sampling schemes, that are consistent with practical deployment within bistatic contexts, to construct the dataset needed for training and testing. Parameters within the neural networks are then learned by minimizing the empirical risk on the training examples, and model generalizability is evaluated by assessing performance on test examples. While success was achieved in the foregoing work, in general, artificial neural networks are limited by small datasets in the duct characterization problem, as such models may be prone to overfit on the necessarily small training sets.

In the present paper, we propose using Gaussian process regression (GPR) to perform a similar task of predicting duct height from sparsely sampled propagation factors. GPR is a nonparametric, fully-Bayesian method that offers several advantages over other machine learning techniques, such as model interpretability and uncertainty quantification for predictions [39]. Using GPR, we tune the prior distribution over functions and subsequently calculate a predic-

tive distribution to estimate duct height from propagation factors obtained via sparse bistatic sampling schemes.

This paper is outlined as follows. We begin by describing the forward model and EM sampling schemes that correspond to different bistatic radar configurations. We then explain GPR and its application to this specific duct height prediction task as well as approaches for tackling severely noise contaminated observations. Finally, we show and discuss the results and limitations of our study.

3.3 Forward Model

EM wave propagation between transmitter and receiver antennas can be modeled as an initial-value problem, using the Fourier SSPE algorithm as an approximation to the time-independent Helmholtz wave equation [36]:

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial z^2} + k^2 n(z)^2 \varphi = 0 \tag{3.1}$$

where φ is the horizontal polarization of the electric field, k is the free-space wavenumber, and n is the refractive index. The refractive index, assumed to be height dependent and constant in range, is the ratio of EM propagation velocity in free space to that in the medium. We often define another quantity called *refractivity*, $N = (n-1)10^6$, to better study the small changes in the refractive index that characterize the MABL and *modified refractivity* to additionally account for the curvature of Earth's surface [48]. In the presence of evaporation ducts under neutral stability conditions, the modified refractivity profile within the MABL

can be modeled as a log-linear curve using the Paulus-Jeske model [27]:

$$M(z) = M_0 + c\left(z - z_d ln\left(\frac{z + z_0}{z_0}\right)\right) \tag{3.2}$$

where M_0 is surface refractivity, $c = 0.13$M-units/m is critical potential refractivity gradient for neutral evaporation ducts, $z_0 = 0.00015$m is aerodynamic surface roughness of the ocean, and z_d is the duct height. To be consistent with the South China Sea Monsoon Experiments in 1998 [43], we chose $M_0 = 428.89$M-units and solve the EM wave equation using SSPE for all considered duct heights, z_d, at every half meter between 2-40m in altitude, which encompasses practical evaporation duct height instances [19].

3.3.1 SSPE Solution

The present work considers a 2D rectangular problem domain with a maximum range (x) of 50km and altitude (z) of 113m, with grid spacings of 40m and 0.1m, respectively. The transmitter antenna is located at $x = 0$km and generates a horizontally-polarized Gaussian antenna pattern with a radar signal frequency of 9.3 GHz. The SSPE solution is obtained by specifying the initial field at the left boundary and then propagating the field down range by a series of fast Fourier transformations while satisfying the lower and upper boundary conditions.

We use an adaptation of PETOOL, a SSPE solver developed by [36], that specifies a Leontovich surface impedance condition at the lower boundary [21]. The continuity of the tangential components of the electric and magnetic fields in this boundary condition is satisfied by assuming that the ocean free surface

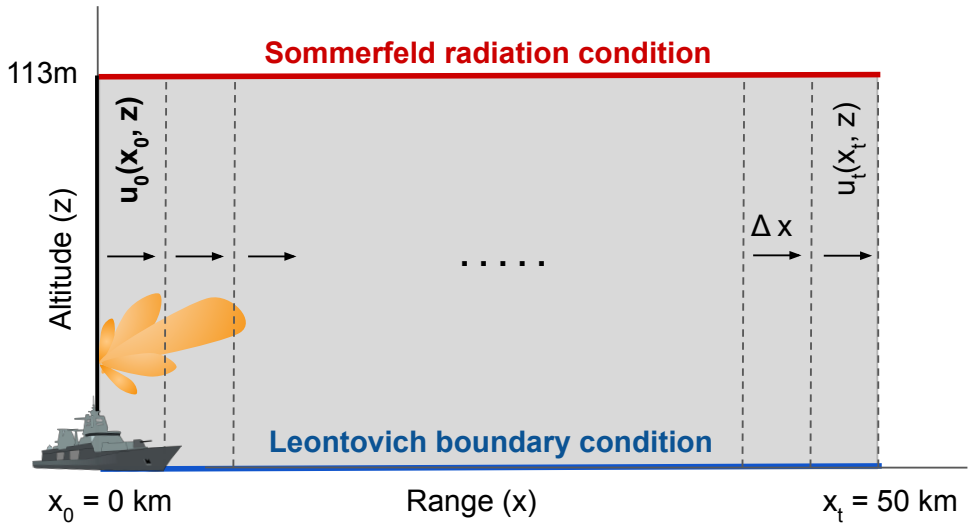


Figure 3.1: **Problem domain.** Domain of interest and associated boundary and initial conditions.

at $z = 0$m is a flat, finite conductor with a homogeneous dielectric constant. This constant is consistent with the thermodynamic conditions of the South China Sea in [43] (*i.e.*, 100% humidity at ocean surface, surface temperature of 29.7 $^{\circ}C$, and ocean salinity of 35 ppt.) and can be calculated using the semi-empirical Debye expression [41]:

$$\epsilon(\omega) = \epsilon_{ir} + \frac{\epsilon_0 - \epsilon_{ir}}{1 - i\omega\tau} + \frac{i\sigma}{\omega\epsilon_0} \tag{3.3}$$

where $\epsilon_{ir} = 4.9$ is the far-infrared dielectric constant of water, τ is the relaxation time, σ is the ionic conductivity, and ϵ_0 is the static dielectric constant of sea water. At infinity, the electric field is decayed to zero in the Sommerfeld radiation condition. Due to the electric field truncation at $z = 113$m, PETOOL approximates this condition at the upper boundary by extending the domain altitude and applying a Hanning window to remove non-physical reflections that result from this truncation. Please refer to Figure 3.1 for the setup of SSPE.

Equipped with the EM wave solution, *propagation factors* within the domain, collectively called the *coverage diagram*, can be calculated [36]:

$$PF = 20 \log |u| + 10 \log x + 10 \log \lambda \tag{3.4}$$

where $u = \exp(-ikx)\varphi(x, z)$ is the reduced amplitude function for the parabolic equation and λ is the free-space wavelength. Propagation factors (*i.e.,* the features of interest in our problem) are defined as the electric field magnitudes scaled with that observed in free space and they account for environmental effects and surface roughness in radar calculations [41].

3.3.2 EM Sampling Schemes

Propagation factors are sparsely sampled from within the coverage diagram to provide a practical context for collecting bistatic radar measurements. We use three EM sampling schemes to construct three separate datasets, each consisting of an array of propagation factors (observation/input) for every considered duct height (label):

Case 1: Stationary transmitter and moving receiver. We consider a stationary transmitter with antenna height of $h = 10$m and angle of $\alpha = 0°$, with respect to the horizontal, and a moving receiver attached to a *rocketsonde* (*i.e.,* solid rocket propelled sensing system). This rocketsonde-receiver sampling system (RRSS) is flown at a constant altitude of $y = 21$m, and samples 250 evenly spaced propagation factors along a horizontal line between 5-15km in range. Please refer to Figure 3.2 for a depiction of this sampling path. This flight tra-

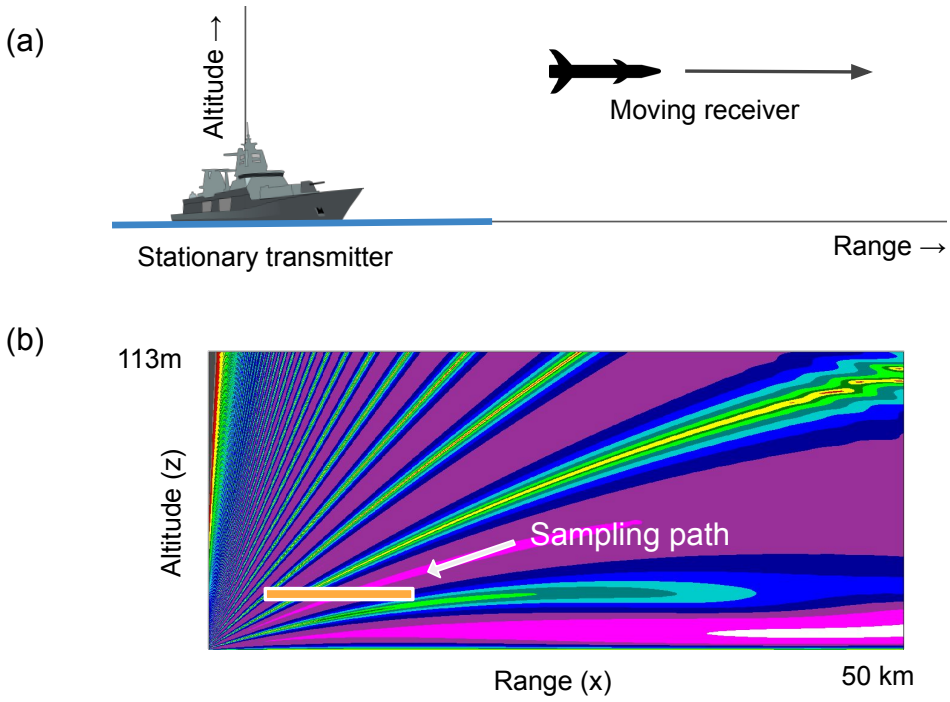
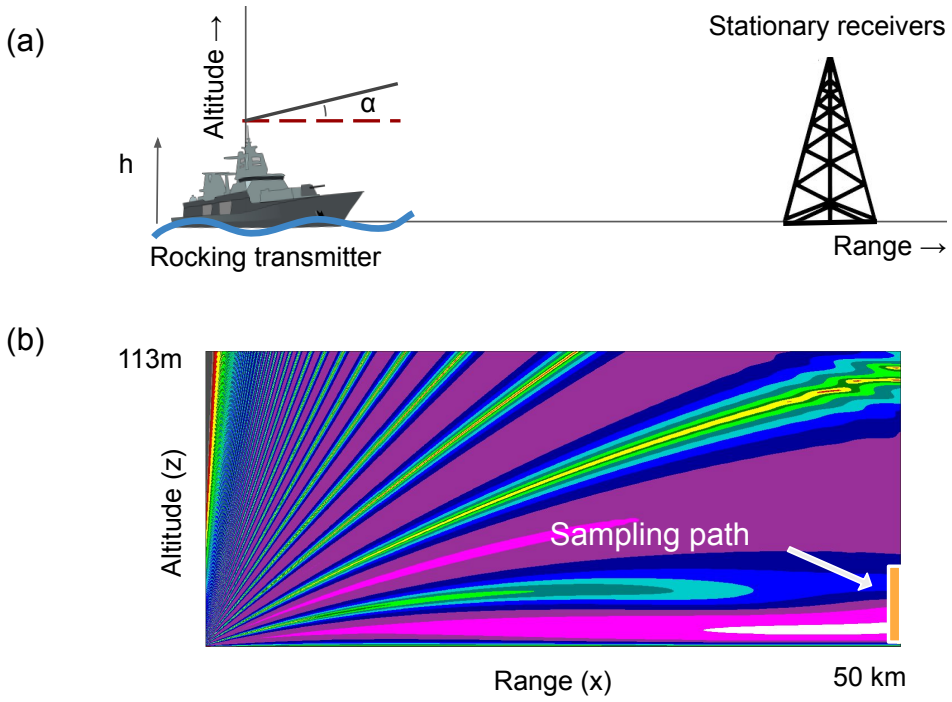


Figure 3.2: **EM Sampling Scheme 1.** Path is consistent with **a.** a stationary transmitter and moving receiver (RRSS), and is **b.** shown on representative coverage diagram.

jectory was arrived at as a result of an exhaustive sensitivity study aimed at uncovering short, flat sampling paths that lead to successful evaporation duct height inversions.

Case 2: Deterministically rocking transmitter and stationary receivers. We consider a rocking transmitter and stationary receivers mounted onto a vertical tower located at $x = 50$km downrange. These receivers sample 30 equally spaced propagation factors between 0-30m in altitude. Please refer to Figure 3.3 for a depiction of this sampling scheme. The rocking is meant to be more closely aligned with the notion of a floating transmitter. To mimic arbitrary rolling of $\pm 0.5^\circ$ and heaving of 10m of the transmitter antenna, six observations are collected using discrete combinations of transmitter heights $h = 20\text{m}, 30\text{m}$



Sampling Scheme 2. Path is consistent with **a.** a rolling and heaving transmitter and stationary receivers mounted on a tower, and is **b.** shown on representative coverage diagram.

EM

and antenna angles $\alpha = -0.5°, 0°, 0.5°$, concatenated within an array of propagation factor measurements.

Case 3: Stochastically rocking transmitter and stationary receivers. Using the same sampling scheme as Case 2, to mimic stochastic rolling and heaving of the transmitter antenna, five observations are collected by sampling uniformly for transmitter antenna heights $h \sim U[20\text{m}, 30\text{m}]$ and antenna angles $\alpha \sim U[-0.5°, 0.5°]$, so as to form another array of measured propagation factors. This configuration leads to less of a defined pattern in the dataset compared to the deterministic configuration.

3.3.3 Severe Sensor Noise

To create surrogate experimental results, we contaminate our propagation factor measurements with "severe" electronic sensor noise using either additive Gaussian white noise or pink noise. Noise is generated using MATLAB's `dsp.colorednoise` function, which generates colored noise with power spectral density of $1/|f|^{\beta}$, where β corresponds to the inverse frequency power (*e.g.*, $\beta = 0$ for Gaussian white noise, $\beta = 1$ for pink noise), and then applies a color filter if $\beta \neq 0$. The noise is scaled using $\sigma = 0.1\|x\|_{\infty}$, where sigma is 0.1 times the absolute value of the largest propagation factor from a given measurement array for white noise and $\sigma/1.6$ for pink noise. The factor 1.6 has been arrived at through a statistical analysis of pink and white noise contaminated signals, so as to result in second moment properties (variance) that are the same between the two different noise models.

3.4 Gaussian Process Regression

Gaussian process regression (GPR) is a nonparametric fully-Bayesian method that performs inference in the space of assumed (prior) functions to ultimately arrive at the posterior predictive distributions at points where we would like predictions. The joint prior distribution is specified as a Gaussian process, which is a collection of an uncountably infinite set of random variables such that any finite subset has a multivariate Gaussian distribution that can be specified using only the mean function, $m(x)$, and covariance function, $k(x, x')$ [39]:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{3.5}$$

If observation labels (on training points) are contaminated with independent and identically distributed (*i.i.d*) additive Gaussian white noise $\epsilon_y \sim \mathcal{N}(0, \sigma_n^2)$, its prior joint distribution can still be a Gaussian process, because the sum of two independent random variables from Gaussian distributions forms another Gaussian PDF that has mean and variance that are the sums of the two distributions. Thus, for noisy labels y, we can assume a Gaussian process with the following mean and covariance functions [39]:

$$y = f(x) + \epsilon_y \tag{3.6}$$

$$y \sim \mathcal{GP}(m(x), k(x, x') + \delta_{ij}\sigma_n^2) \tag{3.7}$$

where the noise variance, σ_n^2, is set at 0.1, to correspond to the grid spacing in the altitude direction of our PETOOL model (as a way to encode ambiguity in the vertical resolution). The functions evaluated on training and test sets are, thus, jointly Gaussian:

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu \\ \mu_* \end{bmatrix}, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix} \right) \tag{3.8}$$

where y is the vector containing the noisy training labels, f_* is the vector of test predictions, μ is the vector of training label means, μ_* is the vector of test label means, X are the training observations, X_* are the test observations, and K is the covariance kernel that can be obtained by evaluating the covariance function, k,

at input pairs. Covariance functions are chosen by the analyst and their associated "hyperparameters" are tuned by maximizing the log marginal likelihood on the training observations. The posterior predictive distributions for new observations can then be calculated by conditioning the joint prior distribution on the data.

3.4.1 Covariance Kernel function

The covariance kernel function specifies the point-wise similarity within the GP and acts as a constraint on the distribution of functions in GPR. A popular covariance kernel function is the squared-exponential (SE) kernel function [39]:

$$k_{SE}(x, x'; \theta) = \exp\left(-\frac{1}{2\ell^2}(x - x')^2 \right) \tag{3.9}$$

SE is stationary and infinitely differentiable [17]. Within the family of kernels are tunable "hyperparameters", θ, that further define the distribution of possible functions. The characteristic lengthscale, ℓ, in SE adjusts the smoothness of the functions (*i.e.*, the longer the ℓ, the more smooth the functions, since a large ℓ indicates long range, point-wise similarity in the GP prior) [39]. Other common covariance kernel functions include constant, periodic, linear, and Matérn kernel functions. More expressive functions can also be constructed by summing and multiplying these kernel functions. In the present work, we use the product of the constant kernel function and the SE kernel function, as our covariance kernel function, so that the constant kernel function scales the magnitude of the SE kernel function, to encode the signal variance:

$$k(x, x'; \theta) = k_C(x, x'; \theta) * k_{SE}(x, x'; \theta) = \sigma_f^2 \ \exp\left(-\frac{1}{2\ell^2}(x - x')^2 \right) \tag{3.10}$$

where x are the training observations, x' are the test observations, and σ_f^2 is the tunable signal variance. The initial value of the signal variance is specified as 1.0 with bounds of (1e-1, 1e3) for case 1 and bounds of (1e-1, 5e1) for cases 2 and 3. The initial lengthscale of the SE kernel is specified as 10.0 with bounds of (1e-3, 1e3). The values indicated were initial values that serve as a point of departure for a numerical optimization algorithm to arrive at appropriate values (vis-á-vis training data) via maximization of the log likelihood function.

3.4.2 Model Selection

Model selection for GPR, typically referred to as model "training", is the process of selecting the family of covariance functions, and setting the free hyperparameters (*e.g.,* characteristic lengthscale ℓ in squared exponential functions). A common approach is to use an optimization algorithm to find hyperparameters that maximizes the log marginal likelihood. Marginal likelihood is the probability of the data, given model parameters (*i.e.,* marginalize out the model parameters); thus, it inherently encodes a tradeoff between model complexity and fit [39]:

$$p(y|X) = \int P(y|f, X)p(f|X)df \tag{3.11}$$

$$\log p(y|X) = -\frac{1}{2}y^T(K + \sigma_n^2 I)^{-1}y - \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{n}{2}\log 2\pi \tag{3.12}$$

where y are the labels, X are the observations, and K is the covariance matrix emanating from the kernel function. When maximizing equation 3.12, gradient-based optimization is preferred, because the inversion of the covariance matrix is performed only once and partial gradients of the log marginal likelihood with

respect to the hyperparameters can easily be calculated [39]. Typically, inversion of this matrix within the calculation of the marginal likelihood is performed using Cholesky factorization for numerical stability. We use a computer memory efficient, quasi-Newton iterative method for bound-constrained optimization called L-BFGS-B within scikit-learn [38]. The log likelihood is typically non-convex and as result, optimization is allowed to be restarted $n = 10$ times with different initializations, so as to identify reasonable parameters for the Gaussian process.

3.4.3 Prediction and Evaluation

The joint probability distribution on the training points and the test points is a multivariate Gaussian and thus, when conditioned on the observations, the resulting distribution, called the posterior predictive distribution, is also a multivariate Gaussian [39]:

$$f_*|X, y, X_* \sim \mathcal{N}(\bar{f}_*, cov(f_*)) \tag{3.13}$$

$$\bar{f}_* = \mu_* + K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1}(y - \mu) \tag{3.14}$$

$$cov(f_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1}K(X, X_*) \tag{3.15}$$

As a result, GPR is able to provide a prediction, $\bar{f}_*$, which is the expected value from all possible functions, and its variance, which can be taken from the diagonal of the covariance matrix, $cov(f_*)$, for every desired test point. Calculation of the prediction and variance requires the inversion of the resulting covariance matrix, and again, for numerical stability, Cholesky factorization is

often used.

Within the present work, GPR model predictions (*i.e.,* duct heights) are evaluated using the mean squared error (MSE) metric, which measures the average squared difference between the predictions and labels.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\bar{f}_{*_i} - y_i)^2 \tag{3.16}$$

where n is the number of test points (where predictions are made), $\bar{f}_*$ are the model predictions (mean predicted duct heights), and y are the true labels (*i.e.,* actual duct heights).

3.4.4 Observation Noise

Unlike noise on the labels, which can be easily incorporated into the Gaussian process, noise in the observations (*i.e.,* inputs) is problematic. To obtain a prediction for a noise-contaminated input, we need to marginalize out the input distribution from the posterior predictive distribution, which results in a non-Gaussian distribution [22]:

$$x = u + \epsilon_x \tag{3.17}$$

$$p(y|X, f, u, \Sigma_x) = \int p(y|X, f, x)p(x|u, \Sigma_x)dx \tag{3.18}$$

where u are the ground truth observations and Σ_x is a diagonal matrix containing the variance of the *i.i.d.* Gaussian white noise, ϵ_x, on the observation. Several approaches can be used to numerically or analytically approximate the integral

(equation 3.18). A numerical approximation can be obtained using Monte-Carlo sampling, which involves sampling the input distribution and mixing their Gaussian posterior predictive distributions to obtain a numerical estimation of the actual posterior predictive distribution [22]:

$$p(y|X, f, u, \Sigma_x) \simeq \sum_{t=1}^{T} w_t p(y|X, f, x_t) = \sum_{t=1}^{T} w_t \, \mathcal{N}(\mu(x_t), cov(x_t)) \tag{3.19}$$

where T is the number of samples, w_t is the mixing proportion assigned the Gaussian (note: $\sum_{t=1}^{T} w_t = 1$), and x_t is a sampled input. For a given x_t, we compute the mean, $\mu(x_t)$, and covariance, $cov(x_t)$, of the Gaussian posterior predictive distribution. The resulting distribution from the mixture of Gaussians might not be Gaussian, but the mean and variance of the mixture distribution can still be calculated. The mean is the weighted mean of the means of the Gaussians, and the variance is the weighted variances of the Gaussians with additional terms that account for the weighted dispersion of the means of the Gaussians [52]:

$$\mu_{mixture} = \sum_{t=1}^{T} w_t \mu(x_t) \tag{3.20}$$

$$\sigma^2_{mixture} = \sum_{t=1}^{T} w_t(\sigma^2(x_t) + \mu(x_t)^2) - \mu^2_{mixture} \tag{3.21}$$

The prediction can be approximated using the mean, and its 95% confidence interval can be generated using the variance (equation 3.21). The desired true distribution can be approached with this approximate distribution by increasing the number of samples; thus we choose large $T = 1000$ with equal mixing proportions $w_t = \frac{1}{T}$ to estimate a "ground truth" distribution.

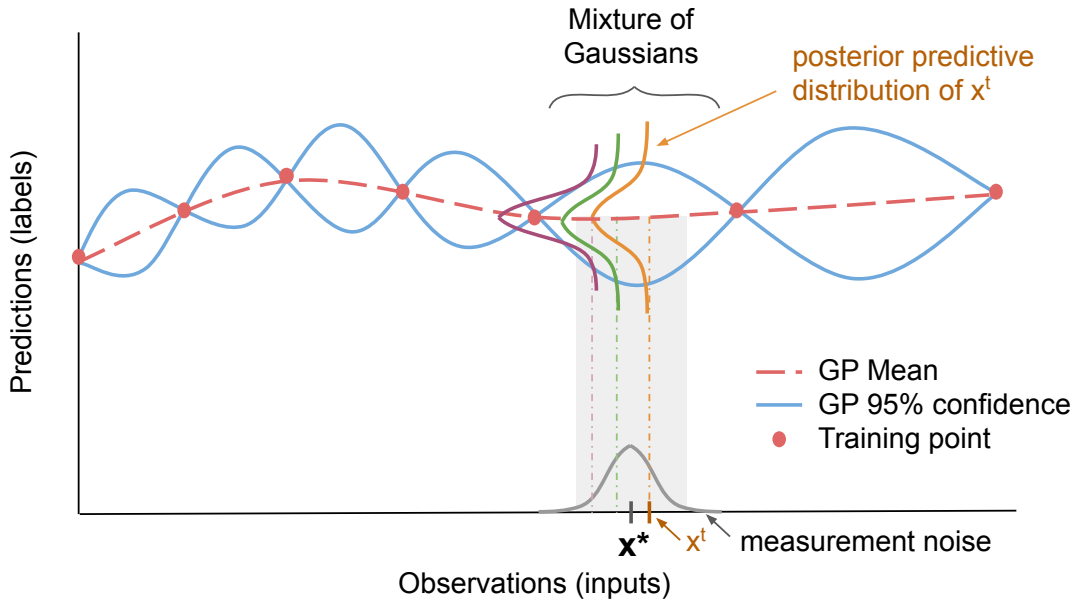


Figure 3.3: **Monte Carlo sampling of noisy observations.** A numerical estimation method that samples from the test point distribution and mixes the Gaussian posterior predictive distributions (on the labels) obtained for each sample, x^t, leading to an approximate posterior predictive distribution for x^*. (x^* is shown for a one-dimensional case for simplicity.)

While other methods exist for arriving at the mixing proportions, we propose an inverse-variance weighting scheme (equation 3.22) for use when the number of observations is small. In this way, the Gaussian with the smallest variance can be given the highest mixing proportion, w_t:

$$w_t = \frac{1/\sigma_t^2}{\sum_{t=1}^{T} 1/\sigma_t^2} \tag{3.22}$$

Any of the techniques discussed above can then be applied broadly when we want predictions from an observation contaminated with Gaussian noise. For application, we can assume that a precise sensor is used for obtaining train-

ing observations, and the extreme electronic sensor noise is only present on the testing observations.

3.5 Results and Discussion

We use Gaussian process regression (GPR) to predict the duct height from both noise-free and noise-contaminated propagation factors. The mean squared error, compared to the true label, is calculated on all predictions as a metric of model performance. Table 3.1 shows this metric calculated on (a) noise-free predictions as well as (b) white-noise and (c) pink noise-contaminated predictions obtained from different approaches discussed in Section 3.4. For reference, root mean square error (RMSE) of 0.5m, which roughly corresponds to an MSE of 2.50e-1 represents a reasonable practical ambiguity. Some variability can be expected due to random train/test splits and in the optimization of hyperparameters.

Table 3.1: MSE for Gaussian process regression predictions compared to true label

(a) Noise-free observations

Train/Test Ratio	Case 1	Case 2	Case 3
80/20	4.92E-04	5.24E-02	1.08E-01
70/30	4.66E-04	3.78E-01	2.62E-01
60/40	4.05E-04	5.75E-01	4.38E-01
50/50	9.06E-03	2.03E+00	2.19E+00

(b) White Noise-contaminated observations

	Train/Test Ratio	MC GT[a]	Naïve	IVW5[b]	IVW10[c]	% Imp[b]	% Imp[c]
Case 1	80/20	7.81E-03	1.28E+00	2.15E-01	8.97E-02	83.2	93.0
	70/30	1.13E-02	1.31E+00	2.41E-01	8.29E-02	81.5	93.7
	60/40	1.81E-02	1.24E+00	2.68E-01	8.53E-02	78.4	93.1
	50/50	5.63E-02	1.12E+00	2.69E-01	1.09E-01	75.9	90.2
Case 2	80/20	1.48E-01	5.37E-01	1.84E-01	1.29E-01	65.8	76.0
	70/30	7.07E-01	1.18E+00	7.59E-01	6.53E-01	35.5	44.6
	60/40	9.16E-01	1.47E+00	1.01E+00	9.06E-01	31.4	38.4
	50/50	2.70E+00	2.95E+00	2.83E+00	2.63E+00	4.18	11.0
Case 3	80/20	1.52E-01	4.52E-01	1.72E-01	1.89E-01	61.9	58.3
	70/30	4.07E-01	8.64E-01	5.11E-01	4.93E-01	40.8	43.0
	60/40	9.39E-01	1.12E+00	9.82E-01	1.07E+00	12.6	5.20
	50/50	3.31E+00	3.32E+00	3.43E+00	3.43E+00	-3.18	-3.11

(c) Pink Noise-contaminated observations

	Train/Test Ratio	MC GT[a]	Naïve	IVW5[b]	IVW10[c]	% Imp[b]	% Imp[c]
Case 1	80/20	1.22E-02	3.28E+00	6.83E-01	2.06E-01	79.2	93.7
	70/30	1.72E-02	2.82E+00	5.70E-01	2.14E-01	79.8	92.4
	60/40	2.52E-02	2.73E+00	6.16E-01	2.82E-01	77.5	89.7
	50/50	6.77E-02	2.61E+00	5.55E-01	2.59E-01	78.7	90.1
Case 2	80/20	2.05E-01	3.27E+00	1.91E+00	3.91E-01	41.4	88.0
	70/30	8.07E-01	5.91E+00	2.26E+00	1.69E+00	61.8	71.5
	60/40	1.05E+00	6.95E+00	2.67E+00	2.20E+00	61.7	68.4
	50/50	2.87E+00	8.16E+00	4.64E+00	4.47E+00	43.1	45.3
Case 3	80/20	2.32E-01	5.60E+00	1.22E+00	1.36E+00	78.2	75.8
	70/30	5.49E-01	6.89E+00	2.44E+00	2.12E+00	64.5	69.2
	60/40	1.27E+00	6.28E+00	3.61E+00	2.35E+00	42.5	62.6
	50/50	3.87E+00	8.16E+00	5.02E+00	4.12E+00	38.5	49.5

[a]MC GT - Monte Carlo Ground Truth

[b]IVW5 - Inverse-variance weighting approach for 5 samples

[c]IVW10 - Inverse-variance weighting approach for 10 samples

Table 3.2: MSE for Gaussian process regression predictions (from naïve approach and inverse-variance weighting, IVW, approach) compared to mean of "ground truth" distribution

(a) White Noise-contaminated observations

	Train/Test Ratio	Naïve	IVW5[a]	IVW10[b]	% Imp[a]	% Imp[b]
Case 1	80/20	1.21E+00	1.71E-01	6.29E-02	85.9	94.8
	70/30	1.28E+00	2.21E-01	6.36E-02	82.7	95.0
	60/40	1.18E+00	2.21E-01	6.00E-02	81.3	94.9
	50/50	1.04E+00	1.94E-01	6.14E-02	81.4	94.1
Case 2	80/20	2.94E-01	4.66E-02	4.15E-02	84.1	85.9
	70/30	3.89E-01	3.34E-02	3.13E-02	91.4	92.0
	60/40	3.62E-01	4.15E-02	2.74E-02	88.5	92.4
	50/50	2.47E-01	3.81E-02	2.28E-02	84.6	90.8
Case 3	80/20	4.07E-01	7.26E-02	2.92E-02	82.2	92.8
	70/30	3.59E-01	6.92E-02	3.77E-02	80.7	89.5
	60/40	3.88E-01	6.98E-02	3.24E-02	82.0	91.6
	50/50	3.44E-01	6.91E-02	2.80E-02	79.9	91.8

(b) Pink Noise-contaminated observations

	Train/Test Ratio	Naïve	IVW5[a]	IVW10[b]	% Imp[a]	% Imp[b]
Case 1	80/20	3.01E+00	5.96E-01	1.69E-01	80.2	94.4
	70/30	2.57E+00	5.32E-01	2.03E-01	79.3	92.1
	60/40	2.54E+00	5.61E-01	2.63E-01	77.9	89.6
	50/50	2.55E+00	4.62E-01	2.35E-01	81.9	90.8
Case 2	80/20	2.88E+00	1.88E+00	3.41E-01	34.7	88.1
	70/30	4.29E+00	1.39E+00	4.95E-01	67.6	88.5
	60/40	4.95E+00	1.30E+00	6.11E-01	73.8	87.7
	50/50	4.82E+00	1.42E+00	6.33E-01	70.5	86.9
Case 3	80/20	6.22E+00	1.12E+00	1.01E+00	82.0	83.7
	70/30	6.46E+00	2.09E+00	1.38E+00	67.7	78.6
	60/40	6.16E+00	2.98E+00	1.38E+00	51.7	77.6
	50/50	6.01E+00	2.75E+00	1.31E+00	54.2	78.2

[a]IVW5 - Inverse-variance weighting approach for 5 samples

[b]IVW10 - Inverse-variance weighting approach for 10 samples

3.5.1 Effect of Train to Test Ratio

To observe GPR performance on even smaller training datasets, we vary the ratio of training points to testing points, from 80/20 down to 50/50. The end points of the dataset, 2m and 40m, are included in the training set, so that the model does not need to extrapolate at those points during inference. As can be seen in Table 3.1a, model performance is proportional to the train to test ratio. This observation is in line with the fact that a higher number of training points allows the GP to better constrain the prior distribution based on the training data before inference. When the ratio decreases, performance of GPR on cases 2 and 3 degrades below the ideal threshold. Thus, it is recommended to col-

lect data that corresponds to the sampling scheme in case 1, which maintains favorable performance even at a ratio of 50/50.

3.5.2 Effect of Observation Noise

In the present paper, the model is trained solely on noise-free observations and used to estimate duct heights from observations with significant colored noise contamination, which is unknown to the model. We consider both additive Gaussian white noise and non-Gaussian pink noise to observe the generalizability of GPR techniques on different forms of noise contamination.

For these noise-contaminated propagation factors, we estimate the true posterior predictive distribution by calculating a numerical approximation using 1000 Monte-Carlo samples (approach shown in Figure 3.3). From this, we obtain a "ground-truth" distribution, whose mean is the prediction from the noise-free measurement and variance is inflated to account for the input noise, for comparison with the *naïve approach* and the *inverse-variance weighting approach*. In the naïve approach, a single noise-contaminated measurement is assumed to be deterministic, and the posterior predictive distribution is calculated as if the measurement were a noise-free measurement. In the inverse-variance weighted approach, detailed in Section 3.4, five and ten mixing components are used with inverse-variance weighted mixing proportions (equation 3.22). MSE on the predictions from the naïve approach and inverse-variance approach are compared to the "ground truth" duct height, shown in Table 3.2.

The inverse-variance weighted approach with five samples offers a large improvement in comparison to the naïve approach by decreasing the MSE to be-

tween 79.9% to 91.4% for white noise and 34.7% to 82.0% for pink noise, as can be seen in Table 3.2. Predictions on the pink noise-contaminated observations are slightly worse than those on the white noise-contaminated observations and thus benefit from utilizing more samples in this method. Further improvements can be seen when ten samples are used as the MSE decreases to between 85.9% to 95.0% for white noise and 77.6% to 94.4% for pink noise. However, it is important to note that the performance of this method relies on the ability of the model to get accurate predictions on the noise-free observations, and subsequently, for the "ground truth". Performance degrades when this is not true, as can be seen in the last two columns of Table 3.1 where the improvement in MSE, calculated on the labels, of the inverse-variance weighting approach decreases as the train to test ratio decreases.

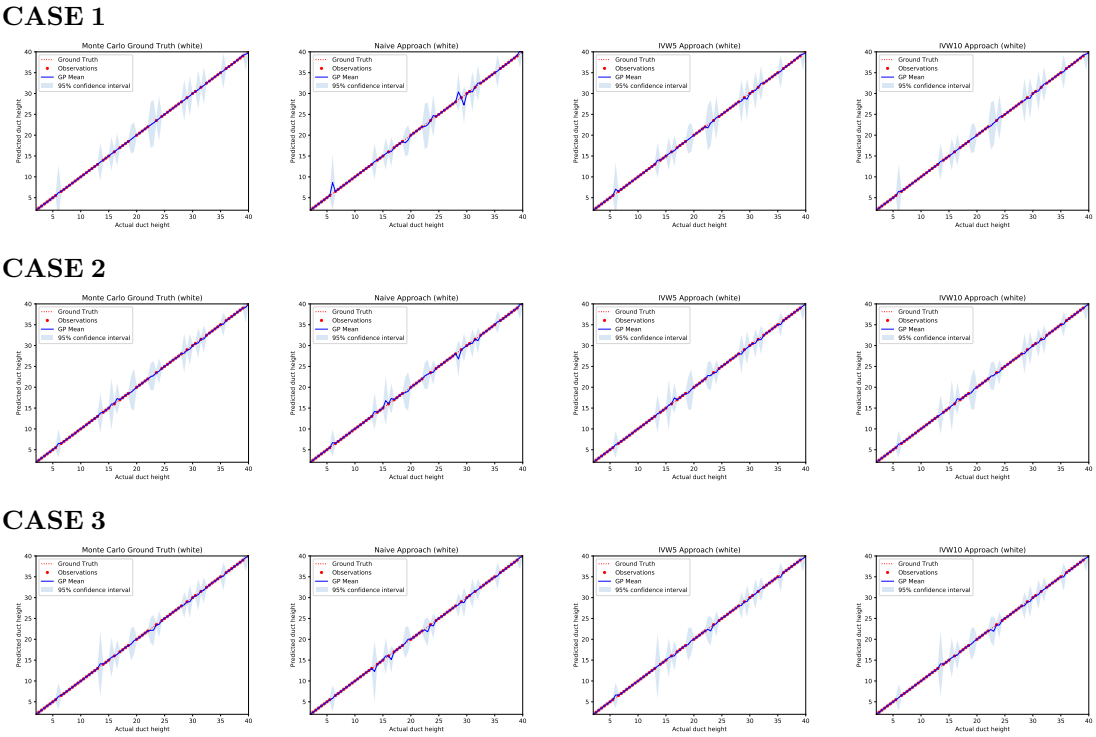
Unexpected variance inflation in the presence of noise can be attributed to the high dimensionality of the array of propagation factors (*i.e* adding noise to every dimension can knock the test point further away from the training points in space). As can be seen in Table 3.3, the average Euclidean distance between the test inputs and their closest training input for the noise-contaminated test inputs is between 1.22 and 2.73 times further away. As can be seen in Figure 3.6, the increase in variance from using the inverse-variance weighted method made labels be within the 95% confidence interval of the prediction. The inverse-variance weighted method thus can offer an improvement without the computational cost of MC sampling with 1000 samples.

Table 3.3: Average Euclidean distance between test point and its closest training point.

	Train/Test Ratio	Noise-free	White Noise	Pink Noise
Case 1	80/20	12.2	32.5 (2.66×)	30.6 (2.50×)
	70/30	11.9	32.5 (2.73×)	30.4 (2.55×)
	60/40	12.4	33.9 (2.73×)	31.1 (2.50×)
	50/50	14.2	35.1 (2.47×)	33.0 (2.32×)
Case 2	80/20	26.6	34.5 (1.30×)	34.6 (1.30×)
	70/30	23.0	37.5 (1.25×)	37.9 (1.26×)
	60/40	29.8	37.1 (1.25×)	38.1 (1.28×)
	50/50	32.0	39.2 (1.23×)	40.0 (1.25×)
Case 3	80/20	27.0	33.6 (1.25×)	33.3 (1.24×)
	70/30	28.3	34.7 (1.23×)	34.6 (1.22×)
	60/40	28.3	35.7 (1.26×)	35.5 (1.25×)
	50/50	30.1	37.2 (1.23×)	36.8 (1.22×)

3.5.3 Timing

Both training and inference are performed using 4-cores of Intel i5 microprocessor having a clock speed of 2.7 GHz, and timing is calculated using the Python time.clock() method, which measures processor system time in seconds. Table 3.4 shows the training time using scikit-learn's Gaussian process regression `fit` function and inference time using its `predict` function [38]. Inference time for obtaining the "ground-truth" distribution includes the inference time of the 1000 samples per test point, as well as extra computation for calculating the mean and variance of the posterior predictive distribution for each test point. Similarly, for the inverse-variance weighted approach, inference time includes the inference time of the five samples per test point as well as the extra com-

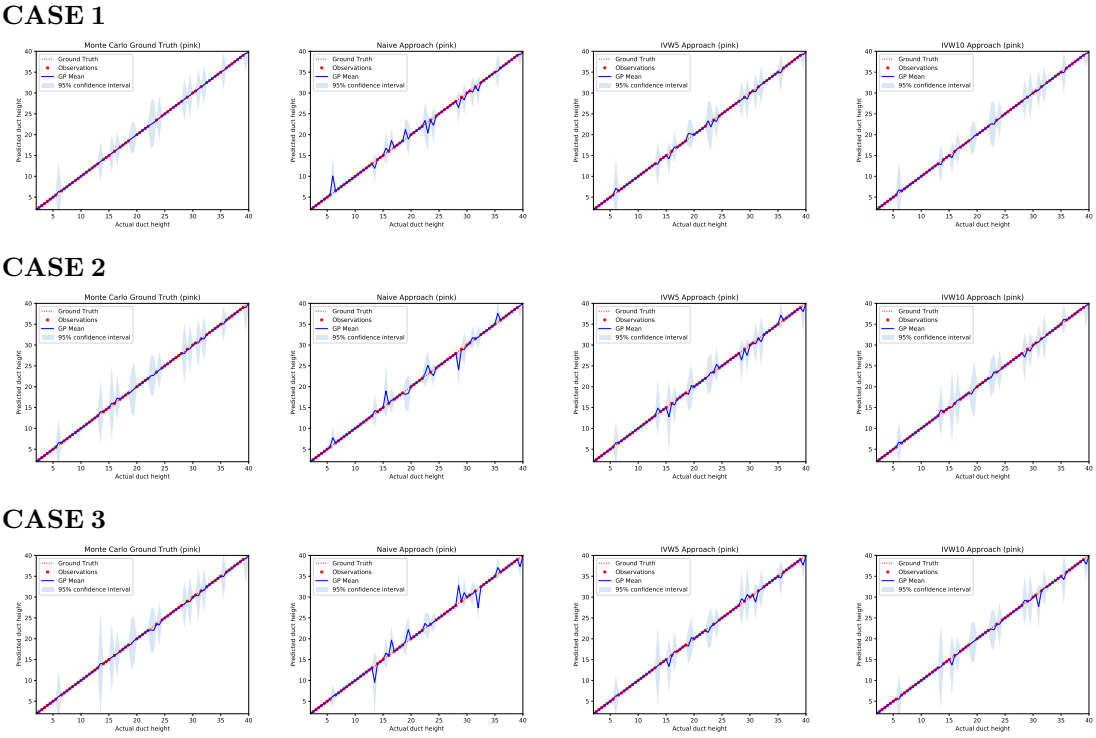


(a) Ground truth (b) Naïve (c) IVW5 (d) IVW10

Figure 3.4: GPR results for white noise contamination with 80/20 train to test ratio

putation. Inference times for noise-contaminated observations are averaged for white and pink noise.

Training time (offline) in GPR is relatively modest, requiring fewer than two seconds of processor system time in all cases considered. It is interesting to note that inference time depends on the number of training points, m, since calculating the mean of the posterior predictive distribution requires inversion of the $K(X,X)+\sigma_n^2 I$ matrix, which dominates the calculation with $O(\mathrm{m}^3)$ computational complexity. Online inference takes fewer than two milliseconds for one random sample, and 18 milliseconds for five random samples and 22 milliseconds



(a) Ground truth (b) Naïve (c) IVW5 (d) IVW10

Figure 3.5: GPR results for pink noise contamination with 80/20 train to test ratio

for ten random samples using the inverse-variance weighting approach. The additional time needed for the inverse-variance weighting method is modest, compared to the time required for ground truth estimation using MC sampling, which is on the order of hundreds of milliseconds.

3.6 Conclusion

Gaussian process regression can be a powerful tool for problems limited by small datasets, such as the duct characterization problem. Unlike many super-

Table 3.4: Fitting and Inference Times (seconds)

	Train/Test Ratio	No.[a]	Training	MC	Naïve	IVW5[b]	IVW10[c]
Case 1	80/20	62	1.10E+00	3.14E-01	6.56E-04	2.13E-03	7.94E-03
	70/30	54	8.46E-01	4.14E-01	1.00E-03	4.83E-03	5.02E-03
	60/40	47	9.09E-01	4.58E-01	1.06E-03	4.18E-03	2.12E-02
	50/50	39	9.87E-01	4.93E-01	1.32E-03	3.23E-03	1.52E-02
Case 2	80/20	62	6.30E-01	2.34E-01	8.89E-04	1.72E-02	2.77E-03
	70/30	54	3.35E-01	2.93E-01	7.49E-04	8.17E-03	1.08E-02
	60/40	47	6.12E-01	3.59E-01	8.62E-04	2.54E-03	9.14E-03
	50/50	39	5.35E-01	3.66E-01	7.12E-04	2.60E-03	9.88E-03
Case 3	80/20	62	5.79E-01	2.06E-01	7.20E-04	2.19E-03	6.20E-03
	70/30	54	3.30E-01	2.67E-01	5.91E-04	1.99E-03	7.01E-03
	60/40	47	3.37E-01	3.33E-01	6.87E-04	2.55E-03	6.59E-03
	50/50	39	3.14E-01	3.18E-01	1.34E-03	2.49E-03	1.30E-02

[a]No. of training points

[b]IVW5 - Inverse-variance weighting approach for 5 samples

[c]IVW10 - Inverse-variance weighting approach for 10 samples

vised machine learning algorithms, including least squares and artificial neural networks, GPR is not prone to overfitting and can provide uncertainty quantification on those predictions.

We show that GPR is able to accurately predict duct height from an array of propagation factors that is sparsely sampled within a context consistent with bistatic radar systems. For noise-contaminated propagation factors, we show that the inverse-variance weighing approach generally performs better than the naïve approach by decreasing the MSE and increasing the variance. Based on favorable model performance in the presence of severe colored noise contami-

nation with a low train to test ratio, the first sampling case is recommended for evaporation duct height prediction. The results in this study are obtained from surrogate data with certain environmental idealizations, such as a smooth ocean surface, constant refractivity profile in range, etc. However, GPR is expected to generalize to more realistic cases under the condition that the training data reflects these less idealized conditions. From these results and recorded inference times, Gaussian process regression provides suitable, "real-time" approach for estimating duct height.

CHAPTER 4

DEEP LEARNING FOR CLASSIFYING AND CHARACTERIZING ATMOSPHERIC DUCTING WITHIN THE MARITIME SETTING

4.1 Abstract

Real-time characterization of refractivity within the marine atmospheric boundary layer can provide valuable information that can potentially be used to mitigate the effects of atmospheric ducting on radar performance. Many duct characterization models are successful at predicting parameters from a specific refractivity profile associated with a given type of duct; however, the ability to classify, and then subsequently characterize, various duct types is an important step towards a more comprehensive prediction model. We introduce a two-step approach using deep learning to differentiate sparsely sampled propagation factor measurements collected under evaporation ducting conditions with those collected under surface-based duct conditions in order to subsequently estimate the appropriate refractivity parameters based on that differentiation. We show that this approach is not only accurate, but also efficient; thus providing a suitable method for real-time applications.

4.2 Introduction

Propagation of electromagnetic (EM) signals through the marine atmospheric boundary layer (MABL) (*i.e.,* the lowest region of the troposphere above the ocean) can be heavily impacted by turbulent interactions at the air-sea inter-

face. The MABL is characterized by turbulent fluxes of heat, moisture, and momentum between the ocean surface and the atmosphere that cause significant inhomogeneity in meteorological conditions within this region [45]. Spatial distributions of temperature, pressure, and humidity factor into the refractive index profile of the MABL; thus strongly influencing the behavior of EM waves propagating through this medium. For example, anomalous EM propagation can occur from a sharp decrease in the refractive index with respect to altitude above the ocean free surface [48]. This specific type of anomalous propagation is called *atmospheric ducting*, and depending on its effect on the refractive index profile, a duct can be categorized as an *evaporation duct*, *surface-based duct*, *elevated duct*; combinations of these are also possible [27]. In ducting, the high refractive index gradient may create a *trapping layer*, within which EM waves may be refracted back toward the ocean surface at a higher curvature than that of Earth's surface [48]. Such behavior can negatively impact communication systems and navigational radars.

The specific case of radar performance motivates the current work: the goal of which is to arrive at a means for rapidly identifying and characterizing ducts within the MABL to allow radar operators on maritime vessels to gauge radar performance and to compensate for the effects of ducting in real-time. Such a capability requires a real-time estimation of the refractive conditions within the MABL. One approach for achieving this is to exploit the relationship between the refractive index and measurable meteorological conditions in order to directly calculate the refractive index [3]. A variety of methods exist for collecting atmospheric data: thermodynamic sensors on both sides of a ship [2], radiosondes [13], aircraft and buoy [9], *etc*. Unfortunately, for the purposes of real-time duct characterization, direct measurements of meteorological conditions, such

as in the foregoing, can be time-consuming, expensive, and data-sparse (*i.e.,* sampling is carried out at too low of a spatio-temporal resolution). More direct measurements using lidar [56] and target-of-opportunity GPS signals [32] can also be used to estimate refractivity, but these methods have their own limitations for this particular problem (*e.g.,* the need for precise satellite alignment over the horizon and ideal weather conditions, respectively). Several alternative approaches to direct measurement involve the use of the radar transmitter, along with various measurements related to EM propagation.

Refractivity from clutter (RFC) methods use radar clutter (*i.e.,* radio signals backscattered from the rough ocean free surface) to estimate the refractive index profile [27]. A common implementation of RFC involves a forward model to predict clutter returns under certain environmental conditions, followed by an inversion to recover parameters of the refractivity profile assumed to be governing the ducting in the MABL. A comprehensive history and review of RFC methods can be found in [27]. More recent methods for optimizing the RFC inversion include particle swarm optimization [54], proper orthogonal decomposition [18], opposition-based learning [58], and dynamic cuckoo search [61]. Additionally, deep learning has become a popular technique due to recent improvements in neural network architecture and training. [51] use neural networks as part of a hybrid model to refine inversion predictions from genetic algorithms. [24] show that deep neural networks can efficiently predict the refractive index parameters associated with evaporation ducts and surface-based ducts, outperforming certain specific genetic algorithms and radial basis function neural networks. [50] similarly demonstrate the success of deep learning strategies for predicting surface-based duct refractive index parameters as compared with a certain instance of particle swarm optimization. Many RFC inver-

sion approaches have focused on either evaporation duct or surface-based duct profiles; however inversion for combined evaporation and surface-based duct profiles has also been introduced in [16] and [55].

Other EM-based techniques that involve the direct transmission of EM energy into the MABL use sparsely measured EM propagation factors (PFs) to estimate the refractive index profile. In one method, collections of PFs are expressed as proper orthogonal modes and organized into a library that can be used for efficient real-time interpolation [19]. Similarly, a low order basis of normal modes can be constructed from the sampled PFs to recast the multimodal inversion problem into a simplified optimization problem [21]. Other work utilizes the PFs directly, to obtain duct parameter predictions using artificial neural networks [46] and Gaussian process regression [47].

While inversion for duct parameters associated with a specific refractivity profile has achieved some success, there is a need for a real-time model that can differentiate between multiple types of ducting phenomena, so that subsequent duct characterization might be carried out. In the present paper, we introduce a two-step deep learning model to: 1) classify the duct; and 2) subsequently estimate the refractive index parameters corresponding to the identified class. We leverage the recent successes in the literature in applying deep learning models to predict evaporation duct and surface duct refractive index parameters and precede such analyses with a classification step.

The present paper is organized as follows: we begin, in Section 4.2, by describing our forward model and the creation of a pre-generated dataset that consists of sparsely sampled propagation factors; we then discuss model selection and deep neural network training in the context of the MABL duct classifica-

tion and regression problems in Section 4.3; finally, we offer an analysis of the two-step model's performance on unseen, noise-contaminated data in Section 4.4.

4.3 Forward Model

Electromagnetic wave propagation within the MABL can be modelled using the time-independent Helmholtz wave equation [21]:

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial z^2} + k^2 n(z)^2 \varphi = 0 \tag{4.1}$$

where φ is the horizontal polarization of the electric field, x is the range, z is the altitude, k is the free-space wave number, and n is the refractive index (*i.e.*, the ratio of EM propagation velocity in free space to that within the medium). To more efficiently estimate propagation for MABL duct characterization, the Helmholtz wave equation can be approximated using a parabolic equation [36]; this is further discussed in section 4.3.

4.3.1 Refractive Index

Meteorological conditions within the MABL impact the propagation of EM waves by affecting the refractive index term in equation 4.1. The refractive index profile can be related to atmospheric temperature, pressure and humidity in a relationship derived by [3]. Within the standard atmosphere, the refractive index steadily decreases with altitude as a result of decreasing temperature, pressure, and humidity [29]. Rays of EM energy are deflected as the refractive

index decreases; thus, EM waves under standard propagation conditions bend slightly downward with a curvature double that of Earth's curvature [29]. To better model the refractive conditions within the MABL, it is practical to define another quantity called the *modified refractivity*, $M(z) = (n - 1 + z/R_e) \cdot 10^6$, where R_e is Earth's radius, to account for Earth's curvature and to amplify the small changes in the refractive index value [42].

Sharp changes in temperature, pressure, and/or humidity gradients are not uncommon within the MABL, and can lead to anomalous propagation events such as *ducting*, which occurs as a result of a sharp decrease in refractivity. The three primary duct classes are evaporation ducts, surface-base ducts, and elevated ducts. To characterize their effects on the refractivity within the MABL, we commonly use simplified modified refractivity models for each of the duct classes, in order to reduce the number of parameters we must search over to characterize the duct [42]. We additionally assume invariance of the modified refractivity profile with range along the path of transmission. Figure 4.1 shows the simplified modified refractivity profiles of three common ducts as well as the parameters of interest in this study, shown in red.

Evaporation Duct

The most prevalent ducts in the MABL are evaporation ducts [59]. Evaporation ducts form in the presence of a high humidity gradient caused by the sharp decrease in water vapor content, away from the ocean free surface [12]. Models based on bulk measurements are commonly used to describe the refractivity profile of the MABL under evaporation ducting, because measuring refractivity close to the ocean surface can be difficult [12]. The most popular bulk model is

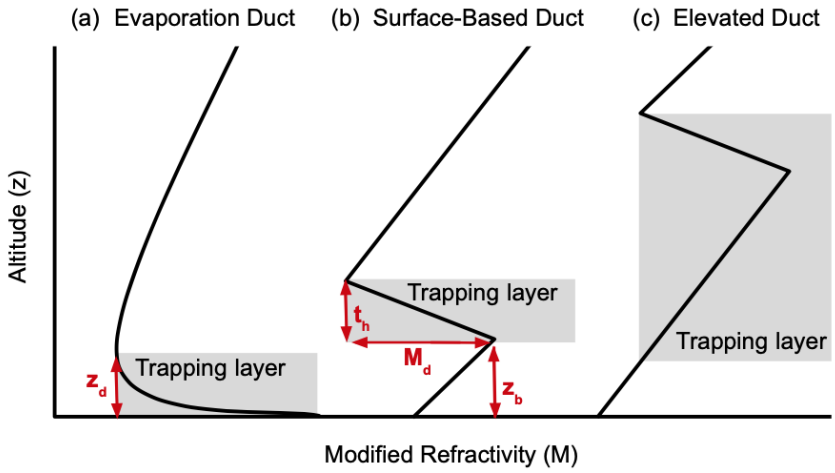


Figure 4.1: Modified refractivity profiles of three common duct types: (a) evaporation, (b) surface-based, and (c) elevated

the *Paulus-Jeske* (PJ) model and, for neutral conditions (*i.e.*, when the difference in temperature between the ocean surface and atmosphere is close to zero), the modified refractivity can be simplified as the following [27]:

$$M(z) = M_0 + c\left(z - z_d \, ln\left(\frac{z + z_0}{z_0}\right)\right) \tag{4.2}$$

where M_0 is the surface refractivity, $c = 0.13$M-units/m is the critical potential refractivity gradient, $z_0 = 0.00015m$ is the aerodynamic surface roughness of the ocean, and z_d is the duct height (Figure 4.1a). The value of surface refractivity has minimal effect on propagation and is fixed as $M_0 = 340$M-units for all ducts within this study [53]. The duct height, defined as the altitude at which the gradient of the modified refractivity profile is zero, is an important parameter for evaporation ducts because it corresponds to their *strength*: the greater the height, the stronger the duct and its ability to impact EM waves at lower frequencies [37]. The PJ model limits evaporation duct heights to be below 40m [2]; thus, in this study, we consider z_d between 2m and 40m.

Surface-Based Duct

Although not as common as evaporation ducts, surface-based ducts can have a greater effect on radar efficacy: causing spurious increases in radar ranging, as well as clutter returns around the transmitter [27]. Surface-based ducts typically form within the MABL as a result of humidity and temperature inversions caused by warm and dry air moving over the cooler ocean surface [48]. A trilinear refractivity model, employing four parameters, can be used to instantiate a surface-based duct [21]:

$$M(z) = \begin{cases} M_0 + s_1 z & z \leq z_b \\ M(z_b) - \frac{M_d}{t_h}(z - z_b) & z_b \leq z \leq z_b + t_h \\ M(z_b + t_h) - 0.118z & z_b + t_h \leq z \end{cases} \tag{4.3}$$

where s_1 is the initial slope of the refractivity, z_b is the base height, M_d is the M-deficit, and t_h is the duct thickness (Figure 4.1b). Following [21], we fix the initial slope at $s_1 = 0.118$M-units/m, and aligning closely to the ranges they considered, we consider z_b between 1m and 30m, t_h between 1m and 30m, and M_d between 4M-units and 50M-units.

Elevated Duct

Elevated ducts form under a similar mechanism as surface-based ducts, and their effect on the modified refractivity profile can also be modeled as a trilinear function (Figure 4.1c) [59]. Unlike surface-based ducts, elevated ducts do not come in contact with the ocean surface and are located at higher altitudes, typically between 600m to 3000m [14]. Due to their location in the atmosphere, elevated ducts do not influence radar systems in a way that is salient to the

present study [27]; thus we do not treat them in the present work. We limit our study to evaporation ducts and surface-based ducts, although their combinations can be explored in future work.

4.3.2 Problem Domain

EM propagation from a transmitter antenna extends from the sea surface to infinity. Due to practical considerations and for computational feasibility, we consider a finite, rectangular 2D domain and apply the necessary boundary conditions to mimic the semi-infinite physical domain. In our domain, the range spans from $x = 0$km (at the location of the transmitter antenna) to $x = 50$km with grid spacing of 40m, and the altitude spans from $z = 0$m (at the ocean surface) to 113m with grid spacing of 0.1m.

As the altitude approaches infinity, EM energy from the transmitter antenna decays to zero. A *Sommerfeld radiation condition* is typically used to achieve this behavior in the forward solver [41]. However, a truncated problem domain can cause artificial reflections in the solution, posing a challenge when implementing the upper boundary condition [35]. Our solution is to extend the upper boundary of the problem domain, and apply a Hanning window to smoothly decay the EM field; as suggested in the literature [36]. At the lower boundary, the *Leontovich surface impedance condition* is used to enforce continuity in the tangential components of the electric and magnetic fields; in order to do so, the ocean surface is modeled as a thin, flat, finite conductor [41]. In this study, the dielectric constant of the ocean surface aligns with thermodynamic conditions of the South China Sea during the SCSMEX experiments [43]: 100% humidity

at ocean surface, surface temperature of $29.7°C$, and ocean salinity of 35 ppt. The dielectric constant under these thermodynamic conditions can then be calculated using the semi-empirical *Debye expression* [41]:

$$\epsilon(\omega) = \epsilon_{ir} + \frac{\epsilon_0 - \epsilon_{ir}}{1 - i\omega\tau} + \frac{i\sigma}{\omega\epsilon_0} \tag{4.4}$$

where $\epsilon_{ir} = 4.9$ is the far-infrared dielectric constant of water, τ is the relaxation time, σ is the ionic conductivity, and ϵ_0 is the static dielectric constant of sea water.

4.3.3 SSPE Solution

In this study, surrogate data is generated using PETOOL [36], which utilizes the *Fourier split-step parabolic equation* (SSPE) algorithm to solve the parabolic equation approximation of the Helmholtz equation, that has been modified by [21]. In SSPE, the initial field is specified at the location of the transmitter antenna (*i.e.,* left boundary in the problem domain) by using parameters of its antenna pattern. In this work, a Gaussian antenna pattern is specified using antenna height of 10m, antenna angle of 0°, and radar signal frequency of 9.3 GHz. The initial signal is then propagated down range with fast Fourier transformations. To obtain the features of interest in this study, *propagation factors* (PFs) (*i.e.,* electric field strength scaled with that observed in free space) are calculated using the SSPE solution [36]:

$$PF = 20log|u| + 10logx + 10log\lambda \tag{4.5}$$

where $u = \exp(-ikx)\varphi(x, z)$ is the reduced amplitude function for the parabolic equation, and λ is the free-space wavelength. In radar calculations, these propagation factors are strongly influenced by environmental effects and surface

roughness [41]. It is pointed out that we purposely ignore surface roughness in the ocean free surface, as we are concerned with demonstrating the possibility of employing deep learning to classify, and subsequently characterize, EM ducts. More practical considerations could follow - in later research work.

4.3.4 Sampling Path

An unmanned aerial vehicle (UAV), outfitted with an RF receiver, could provide a practical way to obtain coverage measurements within the MABL. UAVs can be pre-programmed with a desired flight path above the ocean free surface. To conserve time to get these measurements for our prediction models, a linear flight path is considered in this study: inspired by [19] who proposed a rocketsonde-receiver sampling system (RRSS) consisting of a X-band receiver attached to a solid rocket that is horizontally launched at a constant altitude. Using this latter approach, we can collect propagation factors down range from the radar within an order of minutes. In the present work, we acquire 250 equally spaced PF values sampled between 20km and 50km along the range direction from the transmitter, at an altitude of 6m above the ocean free surface. These sampled PFs, along with the duct classification and associated duct parameters, constitute an instance within our labeled surrogate dataset.

4.4 Deep Learning

In the present work, we show that a two-step deep learning model can accurately classify and characterize atmospheric ducting phenomena from sparsely

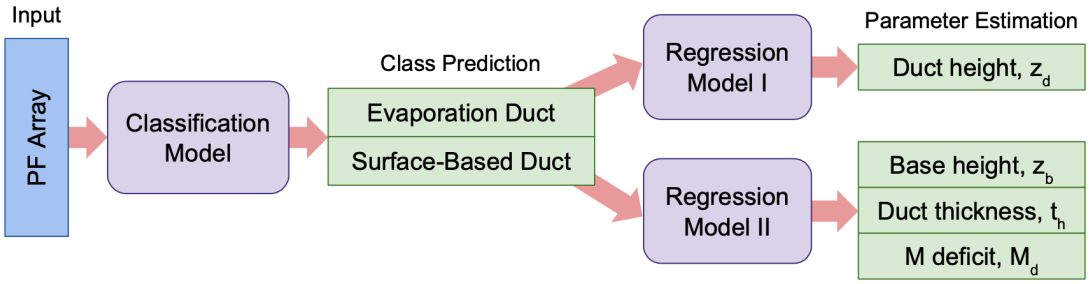


Figure 4.2: Proposed two-step deep learning model

sampled propagation factors. This two-step model consists of a classification model and associated regression models for evaporation ducts and surface-based ducts, as can be seen in Figure 4.2. PF arrays are passed, as inputs, to the classification model, and upon its subsequent prediction of duct type, the same PFs are then fed into the appropriate regression model to yield salient refractivity parameter estimates. We use deep neural networks for all our models and train them using supervised machine learning, to learn an approximate function that maps propagation factors (*i.e.*, inputs / feature vectors) to: 1) the duct type (*i.e.*, class) and 2) the associated collection of duct parameters (*i.e.*, regression labels).

4.4.1 Neural Network Architecture

A *deep learning* (DL) *artificial neural network* (ANN) consists of multiple hidden layers of stacked *perceptrons* that connect the input layer to the output layer, as shown in Figure 4.2. Within a single perceptron, commonly called a neuron within a neural network, inputs are multiplied by weighting factors, w, and summed with a scalar bias, b, such that a nonlinear activation function,

g, can then be applied. The result is called the *activation*: $a_i = g(x^T W + b_i)$. As universal function approximators, ANNs are able to approximate many nonlinear functions due to these nonlinear activation functions [44]. A common activation function for hidden layer neurons is the rectified linear unit (ReLU), $ReLU(h) = max(0, h)$, which is computationally efficient and minimizes the vanishing gradient problem (by having a gradient of 0 for inactive units and 1 for active units) [34]. In a neural network, the output of every neuron within the previous hidden layer becomes an input into the perceptrons within the subsequent layer.

The "depth" and "width" of the neural network are governed by the number of hidden layers and number of neurons per layer, respectively. These configurations are termed *hyperparameters* of the model (*i.e.*, these define the training scheme and network architecture whose *parameters*, the weights and biases, are *learned* during training) [44]. Following the hidden layers, at the end of the network, is the output layer, where the number of neurons and activation function depends on the task. For example, in a classification task, the number of output neurons correspond to the number of classes to be discriminated between, and a softmax function is commonly used to return probabilities over each of the classes. On the other hand, in a regression task, the number of neurons correspond to the number of continuous, scalar parameters that are desired as labels; thus no activation function is applied to the output layer.

Regularization

Overfitting is a common problem for supervised machine learning algorithms, particularly within many physical contexts where data are limited. An overfit-

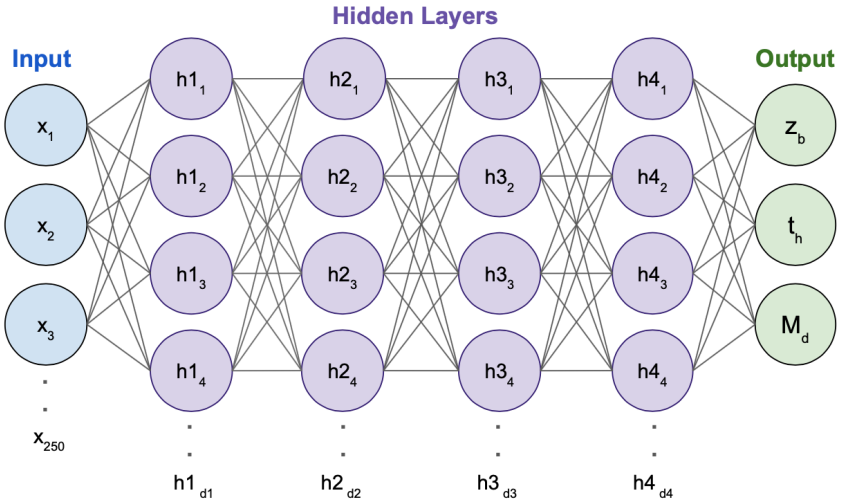


Figure 4.3: Deep neural network for regression of surface-based duct parameters from propagation factors

ted model fails to generalize to unseen data, achieving low errors on the training set but furnishing large errors on the validation/testing set [44]. This phenomenon occurs if the model has too many parameters in comparison to the available training data, as is common in the case with deep neural networks. Several changes to the network architecture can help improve generalization of deep neural networks.

Batch normalization (BN) is a popular technique used in deep neural networks to improve performance, accelerate training, and add regularization. In BN, inputs into the hidden layers of the network are standardized with their mean, μ, and standard deviation, σ [25]:

$$BN(u) = \gamma \cdot \frac{u - \mu}{\sigma} + \beta \tag{4.6}$$

where γ and β are learned parameters. Although the original paper applied BN before the activation function, in practice, it can also be applied after the activation function. The theory behind why BN is so effective is often disputed

[6]. It was originally suggested that BN reduces "internal covariate shift", which is the shift in the input distribution as parameters are tweaked [25]. Through experimentation, [6] deduced that BN prevents activations from blowing up, which allows for larger update steps, and in turn, speeds up convergence and improves regularization.

Dropout is another powerful technique for improving model generalization at a low computational cost. In dropout, neurons and their connections are randomly removed from the network, with probability of r (*i.e* dropout rate, which is a hyperparameter) in each layer, by multiplying its activation by zero [49]. By training the ANN with different activations zeroed out at every update, dropout forces many "thinned" sub-network models to learn the mapping function. The effect of dropout is similar to an ensemble of networks, which helps reduce variance by combining the predictions of multiple independent models [62]. When dropout is applied, model size is expanded by increasing network depth or width to help counter the decreased model capacity [23].

4.4.2 Dataset

Three mutually exclusive sets of data are commonly used to *train*, *validate*, and *evaluate* supervised machine learning models to ensure that performance appropriately reflects the models' ability to make predictions on unseen data [44]. We generate three surrogate datasets using the forward model, as described in Section 4.2. In these datasets, single point consists of three pieces of information: feature (*i.e.,* array of 250 propagation factors), classification (*i.e.,* evaporation duct or surface-based duct), and label (*i.e.,* array of duct refractivity parameters).

This could also be viewed as a class along with a vector of labels describing a given refractivity profile that is associated with points within a 250-dimensional *feature space* comprising the sampled PFs.

For the training set, *Latin Hypercube sampling* (LHS) is employed to efficiently sample the multidimensional parameter space of our problem [15]. In LHS, the 1D sample space for every refractivity parameter is partitioned using a disjoint union of N equally sized intervals. To instantiate a single point within the training dataset, for every refractivity parameter, an interval is randomly selected without replacement and a point is uniformly sampled from within that interval. This is repeated N times to complete the training dataset. Using LHS, $N = 500$ evaporation duct environments and $N = 50,000$ surface-based duct environments are collected within the duct parameter ranges specified within Sections 4.2.1.1. and 4.2.1.2.

For both the subsequent instantiation of validation and evaluation datasets, 50 evaporation duct environments and 5000 surface-based duct environments are uniformly sampled. To mimic experimental results with "severe" electronic sensor noise, the features in the test dataset are additionally contaminated with additive colored noise that has power spectral density of $1/|f|^{\beta}$, where β corresponds to the inverse frequency power. In this study, both Gaussian white noise, $\beta = 0$, and pink noise, $\beta = 1$, are considered. The generated white noise is scaled by $\sigma = 0.1||x||_{\infty}$ (*i.e.,* 0.1 times the absolute value of the largest propagation factor); that scaling factor is divided by 1.6 for the generated pink noise in order to maintain consistent variance between white noise and pink noise.

An imbalanced dataset can have negative consequences in classification, causing the trained model to favor the more represented class. A common so-

lution is to strengthen the role of the minority classes by either altering the loss function or balancing the class distribution (*i.e.,* up-sampling the smaller classes and down-sampling the larger classes) [33]. In this study, we randomly sample 50,000 points, with replacement, from the 500 point evaporation dataset to upsample the smaller class within the training set. We use the same method to correct the imbalance in the validation dataset.

4.4.3 Training

During training, parameters of the ANN (*i.e* w, b, γ, β) are randomly initialized and updated with an optimization algorithm, to improve a chosen loss function on the training dataset. Loss functions are selected based on the network application. For example, in binary classification, which involves a discrimination between two classes, a typical loss function employs the *binary cross-entropy* [11]:

$$L(y_i, \hat{y}_i) = -\frac{1}{N}[y_i \, log(\hat{y}_i) + (1 - y_i) \, log(1 - \hat{y}_i)] \tag{4.7}$$

where y_i is the label, $\hat{y}_i$ is the prediction, and N is the number of examples. Meanwhile, *mean squared error* (MSE) is popular for regression tasks [11]:

$$L(y_i, \hat{y}_i) = -\frac{1}{N}(y_i - \hat{y}_i)^2 \tag{4.8}$$

Variants of first-order gradient methods are commonly used to update parameters within a neural network. In first-order gradient methods, the gradient of the loss function is computed with respect to the network parameters in an approximate manner, such that parameters are adjusted along the directions of steepest descent in an attempt to reduce error. The magnitude of the parameter adjustments is dictated by the *learning rate*. A trade-off exists in the selection of

learning rate: a small learning rate can result in time-consuming and inefficient training phases, while a large learning rate can lead to divergence [44]. As a result of this trade off, it is common to use an adaptive learning rate that decreases throughout training. *Adam optimization* is a stochastic first order method that utilizes estimation of the first and second moment of the gradient to calculate an adaptive learning rate [28]. Mini-batch based methods, which involve updating parameters utilizing only a subset of the data, rather than the complete dataset, can decrease update time and help improve generalization [23]. This subset from within the training dataset is called a *mini-batch*; the size of which becomes yet another another hyperparameter of the network.

Adam optimization requires the calculation of partial derivatives of the loss function, taken with respect to the model parameters. For neural networks, an efficient way to calculate these gradients within the network is to use *back-propagation*, or auto-differentiation [30]. Backpropagation employs information stored during the forward pass of the training data, in which the input is fed into the network to calculate activations, such that a backward pass can be made, in which loss function computed errors are fed back through the network to make model parameter adjustments in proportion to partial derivatives of the loss function (using stored information from the forward pass). The parameters are updated until a specified *epoch*, which is defined as the stage at which the model has "seen" all the examples in the training dataset. We use Keras for this implementation [11].

4.4.4 Model Selection

Model selection via direct, random search was performed to find optimal neural network hyperparameters for each of the three tasks (*i.e.,* classification as either surface or evaporation ducting, followed by regression to determine refractivity parameters). During a random search, a reasonable search space, based on problem specific considerations, is specified for each of the tunable hyperparmeters (*e.g.,* number of hidden layers, number of neurons per layer, dropout rate, batch size, and learning rate), and values are randomly selected within the space of admissible values, to generate multiple random neural network architectures [4]. These network architectures are then trained and periodically evaluated using the validation dataset, and the high performing architectures are determined by the validation metric: the highest *accuracy*, defined as the number of correct predictions over all predictions, for the classification task, and the lowest *root mean squared error* (RMSE), defined as the square root of the average squared difference between the predicted value and true value, for the regression tasks.

Other hyperparameter optimization methods include *manual search*, *grid search*, and *Bayesian optimization*. While Bayesian optimization can find optimal hyperparameters utilizing fewer training examples, both grid search and random search are embarrassingly parallelizable (*i.e.,* model architectures can be trained and evaluated independently); thus leading to computational advantages due to scaling over cores. [4] showed that random search can achieve similar results to grid search while using less computation time. Thus, considering available computational resources (12-cores of Intel Xeon E5 microprocessor with a clock speed of 2.7 GHz), parallelized random search was employed in the current study.

Table 4.1: Hyperparameters corresponding to the top five classification models with the greatest validation accuracy

Hyperparameters	Model 1	Model 2	Model 3	Model 4	Model 5
hidden neurons	922,	935,	979,	868,	832,
	768,	666,	684,	641,	717,
	418,	597,	400,	500,	571,
	366,	232,	206,	351	379,
	237	305,	283	-	247,
	-	373	-	-	354
learning rate (e-4)	2.984	4.831	8.245	7.279	5.442
dropout	0.2703	0.2137	0.3563	0.2878	0.2007
batch size	134	112	73	55	74
training accuracy	0.9930	0.9996	0.9958	0.9948	0.9997
validation accuracy	**1**	**0.9999**	**0.9999**	**0.9999**	**0.9998**

In this study, the hyperparameter search space is defined using knowledge from previous literature, as well as prior experience. For each of the models, we consider between four and six hidden layers, with decreasing ranges for the number of hidden neurons in subsequent layers. We consider learning rates between 1e-3 and 1e-4, as well as appropriate dropout and batch size ranges for each task. For all our models, we use batch normalization and dropout to improve their ability to generalize to unseen data. Tables 4.1, 4.2, and 4.3 shows the five architecture and learning scheme combinations that achieved the best validation metric for the classification task, evaporation duct regression task, and surface-based duct regression task, respectively.

Table 4.2: Hyperparameters corresponding to the top five evaporation duct regression models with the least validation loss

Hyperparameters	Model 1	Model 2	Model 3	Model 4	Model 5
hidden neurons	829,	833,	935,	811,	983,
	603,	754,	719,	616,	713,
	474,	499,	426,	460,	539,
	284,	250,	248,	323,	395,
	265,	288,	399	303	322
	319	220	-	-	-
learning rate (e-4)	5.442	9.708	7.279	9.894	6.822
dropout	0.2007	0.3432	0.2878	0.2606	0.3325
batch size	50	58	84	93	85
training loss	0.07220	0.08977	0.06810	0.09150	0.08445
validation loss	0.04241	0.05730	0.05919	0.05978	0.05997
validation RMSE	**0.2059**	**0.2394**	**0.2433**	**0.2445**	**0.2449**

4.4.5 Evaluation

Architectures obtained from model selection in Tables 4.1, 4.2, and 4.3 are utilized in the *two-step model*. We evaluate this two-step model on a separate test dataset, to gauge its performance on previously unseen data. We additionally assess the performance of the model on severely contaminated data (*i.e.*, test dataset contaminated with severe white and pink noise, as discussed in Section 4.3.2).

Table 4.4 shows the accuracy of the classification models evaluated on the test dataset under different noise conditions. We also consider predictions from an *ensemble*, which is a model that incorporates predictions from multiple mod-

Table 4.3: Hyperparameters corresponding to the top five surface-based duct regression models with the least validation loss

Hyperparameters	Model 1	Model 2	Model 3	Model 4	Model 5
hidden neurons	910,	863,	913,	859,	901,
	745,	688,	740,	638,	672,
	401,	507,	550,	418,	494,
	377,	370,	392,	364,	350,
	284,	370,	254	333,	218,
	344	297	-	249	217
learning rate (e-4)	8.834	4.522	6.910	6.108	3.613
dropout	0.265	0.3081	0.3226	0.2382	0.3151
batch size	144	189	160	167	194
training loss	0.8504	0.9507	1.022	0.8433	1.066
validation loss	3.081	3.098	3.108	3.132	3.135
validation RMSE	**1.755**	**1.760**	**1.763**	**1.770**	**1.771**

els to reduce variance of the neural networks, potentially improving performance over a single model [62]. Many methods for ensembling models exist, such as averaging or weighted averaging the predictions, training another model to combine the predictions, averaging the weights of neural networks, *etc*. In this study, we simply average the class probabilities from the top five models and calculate the argmax to obtain predictions. Subtleties in performance between the ensemble model and top model are highlighted by presenting the difference in the number of correct predictions. The predictions from the ensemble are then used to sift the data and send the examples into the correct regression model.

Tables 4.5 and 4.6 shows the *RMSE* for the regression models as well as the

Table 4.4: Accuracy of classification models on test set during evaluation

Noise	Subset	Model 1	Model 2	Model 3	Model 4	Model 5	Ensemble	Diff.*
none	Evaporation	98%	100%	100%	98%	100%	100%	+1
	Surface-based	99.98%	99.90%	99.94%	99.88%	99.92%	99.96%	-1
white	Evaporation	96%	100%	98%	98%	100%	100%	+2
	Surface-based	99.98%	99.88%	99.92%	99.88%	99.92%	99.94%	-2
pink	Evaporation	88%	96%	96%	96%	92%	96%	+4
	Surface-based	99.86%	99.64%	99.84%	99.84%	99.82%	99.84%	-1

*Difference in number of correct predictions of ensemble from best model

Table 4.5: RMSE of evaporation duct regression model on test set during evaluation

Parameter	Noise	Model 1	Model 2	Model 3	Model 4	Model 5	Ensemble	% Impr.*
z_d (m)	none	0.2721	0.2708	0.2496	0.2849	0.2967	0.1761	35.28%
	white	0.2813	0.2822	0.2831	0.3610	0.3495	0.1999	28.94%
	pink	0.4517	0.5024	0.4135	0.9953	0.4746	0.4267	5.545%

*Percent improvement of ensemble to best model

ensemble model (averaging predictions from the top five models) corresponding to evaporation ducts and surface-based ducts, respectively. The percentage improvement of the ensembled predictions compared to the best model (*i.e.*, the negative percentage change: $-100 \times (\text{RMSE}_{ensemble} - \text{RMSE}_{best})/\text{RMSE}_{best}$) is also presented for reference.

The evaluation time for a single point was calculated for all three models by using python's `time_process` to obtain the time elapsed in seconds. The times shown in Table 4.7 are averaged across all noise conditions and 5050 test examples. Times for both a single model (*i.e.*, the best model) and the ensemble are reported.

Table 4.6: RMSE of surface-based duct regression model on test set during evaluation

Parameter	Noise	Model 1	Model 2	Model 3	Model 4	Model 5	Ensemble	% Impr.*
z_b (m)	none	0.481	0.4391	0.4405	0.4167	0.4667	0.3841	20.14%
	white	0.5282	0.5318	0.5623	0.513	0.5639	0.4759	9.904%
	pink	2.395	2.568	2.622	2.465	2.672	2.364	1.308%
t_h (m)	none	1.310	1.310	1.368	1.327	1.348	1.246	4.928%
	white	2.105	2.105	2.083	2.020	2.125	1.940	7.825%
	pink	3.602	3.522	3.545	3.503	3.606	3.278	8.997%
M_d (M-units)	none	2.648	2.669	2.594	2.701	2.684	2.548	3.761%
	white	3.74	3.659	3.729	3.666	3.739	3.502	6.364%
	pink	4.182	4.15	4.157	4.102	4.096	3.911	6.484%

*Percent improvement of ensemble to best model

Table 4.7: Average Evaluation Time (seconds)

	Classification	Regression	Total
Single	5.550e-4	6.753e-4	1.231e-3
Ensemble	3.572e-3	4.600e-3	8.173e-3

4.4.6 Discussion

Several observations can be made from the model selection results in Tables 4.1-4.3: validation losses are consistent across the top models within a task; models with four to six layers are represented, showing that a variety of model architectures can achieve comparable results; finally, models in Tables 4.1 and 4.2 achieved lower validation loss than training loss, suggesting good regularization.

From results in Table 4.4, it is seen that deep neural networks are extremely

efficient and accurate at classifying arrays of sparsely sampled propagation factors, as either measurements taken in the presence of evaporation ducting or surface-based ducting. Although trained solely on non-contaminated data, the classification models are robust to unseen data, even in the presence of severe white and pink noise. Under every noise condition, all five models achieved high accuracy, between 88% - 100% on evaporation duct PF measurements and 99.64% - 99.98% on surface-based duct PF measurements. This high performance can be difficult to improve; in fact, ensembling the models by averaging class probabilities generally yielded similar results to individual models. A notable +4 difference in the number of correct predictions can be seen for pink-noise contaminated evaporation duct measurements. For this set of data, the best model's performance of 88% is lower than that of other models, suggesting its weakness in generalization. Ensembling neural networks can decrease outlier effects by reducing variances in network performance, and in this case, ensembling achieved a performance of 96%.

For regression networks, ensembling the models improved RMSE to varying degrees, from 1.308% to 35.28%. As can be seen in Tables 4.5 and 4.6, the percentage decrease in RMSE from ensembling can be high in certain cases and insignificant in others. Thus, for real-time applications, it is important to consider its benefits with the trade-off in computation time. From Table 4.7, ensembling the networks increased the total computation time by a factor of approximately 6.64; despite this, the two-step ensembled model can produce a prediction in under nine milliseconds.

4.5 Conclusion

The two-step deep learning model, used in conjunction with the sparse data collection process (*i.e.*, collecting 250 propagation factor measurements using a radiosonde programmed on linear flight path), can be effectively and efficiently deployed to distinguish duct types and characterize their effect on refractivity within the MABL in real-time. In the deep learning model, classification of ducts from sparsely sampled propagation factors achieved high accuracy on both the noise-free and color-noise contaminated test set. Subsequently, the regression models for both duct types achieved reasonable success relative to other deep learning models in the literature. Furthermore, we show that ensembling the top five regression models from hyperparameter search slightly improves predictions, although with the caveat of increased computation time. It is noted that the surrogate data used to train these models assumes certain environmental idealizations: horizontally invariant refractivity profiles, smooth sea surface, simplified modified refractivity profiles, *etc*. Using more realistic training data, it is expected that the models would generalize well to more realistic cases. The observed accuracy and computational speed of the two-step deep learning model makes this a suitable option for real-time inference of ducting characteristics within the MABL.

BIBLIOGRAPHY

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

[2] S. M. Babin, G. S. Young, and J. A. Carton. A new model of the oceanic evaporation duct. *Journal of Applied Meteorology*, 36(3):193–204, 1997.

[3] B. R. Bean and E. J. Dutton. *Radio Meteorology*. Dover books on electronics and related areas. Dover, Washington D.C., 1968.

[4] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

[5] C. M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

[6] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger. Understanding batch normalization. *Advances in Neural Information Processing Systems*, 31:7694–7705, 2018.

[7] L. Bottou. Large-scale machine learning with stochastic gradient descent. *Proceedings of COMPSTAT*, 2010.

[8] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[9] P. Brilouet, P. Durand, and G. Canut. The marine atmospheric boundary layer under strong wind conditions: Organized turbulence structure and flux estimates by airborne measurements. *Journal of Geophysical Research: Atmospheres*, 122(4):2115–2130, 2017.

[10] G. C. Cawley and N. L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11:2079–2107, 2010.

[11] F. Chollet. Keras. https://github.com/fchollet/keras, 2015. GitHub Repository.

[12] J. Cook. A sensitivity study of weather data inaccuracies on evaporation duct height algorithms. *Radio science*, 26(03):731–746, 1991.

[13] WF Dabberdt, R Shellhorn, H Cole, A Paukkunen, J Hörhammer, and V Antikainen. *Radiosondes*. Elsevier, 2003.

[14] E. Dinc and O. B. Akan. Beyond-line-of-sight communications with ducting layer. *IEEE Communications Magazine*, 52(10):37–43, 2014.

[15] R. Douvenot, V. Fabbro, P. Gerstoft, C. Bourlier, and J. Saillard. A duct mapping method using least squares support vector machines. *Radio Science*, 43(06):1–12, 2008.

[16] R. Douvenot, V. Fabbro, and Y. Hurtaud. The detectable double atmospheric ducts for rfc. *IEEE Antennas and Propagation Society, AP-S International Symposium (Digest)*, 2014.

[17] D. Duvenaud. *Automatic Model Construction with Gaussian Processes*. Phd thesis, University of Cambridge, 2014.

[18] V. Fountoulakis and C. Earls. Duct heights inferred from radar sea clutter using proper orthogonal bases. *Radio Science*, 51(10):1614–1626, 2016.

[19] V. Fountoulakis and C. Earls. Inverting for maritime environments using proper orthogonal bases from sparsely sampled electromagnetic propagation data. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7166–7176, 2016.

[20] P. Gerstoft, D. F. Gingras, L. T. Rogers, and W. S. Hodgkiss. Estimation of radio refractivity structure using matched-field array processing. *IEEE Transactions on Antennas and Propagation*, 48(3):345–356, 2000.

[21] M. A. Gilles, C. J. Earls, and D. Bindel. A subspace pursuit method to infer refractivity in the marine atmospheric boundary layer. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–12, 2019.

[22] A. Girard and R. Murray-Smith. Gaussian processes: Prediction at a noisy input and application to iterative multiple-step ahead forecasting of time-

series. In *Switching and learning in feedback systems*, pages 158–184. Springer, 2005.

[23] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[24] X. Guo, J. Wu, J. Zhang, and J. Han. Deep learning for solving inversion problem of atmospheric refractivity estimation. *Sustainable Cities and Society*, 43:524 – 531, 2018.

[25] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 2015. Proceedings of Machine Learning Research.

[26] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer Texts in Statistics. Springer New York, 2014.

[27] A. Karimian, C. Yardim, P. Gerstoft, W. S. Hodgkiss, and A. E. Barrios. Refractivity estimation from sea clutter: An invited review. *Radio Science*, 46(6), 2011.

[28] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *Computing Research Repository*, abs/1412.6980, 2014.

[29] H. W. Ko, J. W. Sari, and J. P. Skura. Anomalous microwave propagation through atmospheric ducts. *JHATD*, 4:12–26, 1983.

[30] Y. LeCun. A theoretical framework for back-propagation. In D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*, pages 21–28, Pittsburgh, Pa, 1988.

[31] N. E. Lentini and E. E. Hackett. Global sensitivity of parabolic equation radar wave propagation simulation to sea state and atmospheric refractivity structure. *Radio Science*, 50(10):1027–1049, 2015.

[32] A. R. Lowry, C. Rocken, S. V. Sokolovskiy, and K. D. Anderson. Vertical profiling of atmospheric refractivity from ground-based gps. *Radio Science*, 37(3):13–1–13–19, 2002.

[33] G. Menardi and N. Torelli. Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, 28(1):92–122, 2014.

[34] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[35] F. Nataf. Absorbing boundary conditions and perfectly matched layers in wave propagation problems. *Radon Series on Computational and Applied Mathematics*, 11, 2013.

[36] O. Ozgun, G. Apaydin, M. Kuzuoglu, and L. Sevgi. Petool: Matlab-based one-way and two-way split-step parabolic equation tool for radiowave propagation over variable terrain. *Computer Physics Communications*, 182:2638–2654, 2011.

[37] R. A. Paulus. *Specification for Evaporation Duct Height Calculations, Technical Report 1596*. Naval Ocean Systems Center, San Diego, California, 1946.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12:2825–2830, 2011.

[39] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Massachusetts Institute of Technology, 2006.

[40] L. T. Rogers, C. P. Hattan, and J. K. Stapleton. Estimating evaporation duct heights from radar sea echo. *Radio Science*, 35(4):955–966, 2000.

[41] F. J. Ryan. *User's guide for the VTRPE (Variable Terrain Radio Parabolic Equation) computer model, Technical Report 1456*. Naval Ocean Systems Center, San Diego, California, 1991.

[42] J. T. Saeger, N. G. Grimes, H. E. Rickard, and E. E. Hackett. Evaluation of simplified evaporation duct refractivity models for inversion problems. *Radio Science*, 50(10):1110–1130, 2015.

[43] SCSMEX. Earth Observing Laboratory. https://www.eol.ucar.edu/field_projects/scsmex, 1998.

[44] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From*

Theory to Algorithms. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, New York, 2014.

[45] D. T Sikora and S Ufermann. Marine atmospheric boundary layer cellular convection and longitudinal roll vortices. In C. Jackson and J. Apel, editors, *Synthetic aperture radar marine user's manual*, pages 321–330. NOAA, Washington D.C., 2004.

[46] H. Sit and C. J. Earls. Characterizing evaporation ducts within the marine atmospheric boundary layer using artificial neural networks. *Radio Science*, 2019.

[47] H. Sit and C. J. Earls. Gaussian process regression for estimating em ducting within the marine atmospheric boundary layer. *Radio Science*, 55(6):e2019RS006890, 2020.

[48] M. I. Skolnik. *Introduction to Radar Systems*. Electrical Engineering Series. McGraw-Hill, 1980.

[49] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[50] W. Tang, H. Cha, M. Wei, B. Tian, and X. Ren. An atmospheric refractivity inversion method based on deep learning. *Results in Physics*, 12:582 – 584, 2019.

[51] C. Tepecik and I. Navruz. A novel hybrid model for inversion problem of atmospheric refractivity estimation. *AEU - International Journal of Electronics and Communications*, 84:258 – 264, 2018.

[52] L. Trailovic and L. Y. Pao. Variance estimation and ranking of gaussian mixture distributions in target tracking applications. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 2, pages 2195–2201 vol.2, 2002.

[53] S. Vasudevan, R. H. Anderson, S. Kraut, P. Gerstoft, T. Rogers, and J. L. Krolik. Recursive bayesian electromagnetic refractivity estimation from radar sea clutter. *Radio Science*, 42(2), 2007.

[54] B. Wang, Z. Wu, Z. Zhao, and H. Wang. Retrieving evaporation duct

heights from radar sea clutter using particle swarm optimization (pso) algorithm. *Progress In Electromagnetics Research*, 9:79–91, 2009.

[55] Q. Wang, R. J. Burkholder, C. Yardim, Q. Wang, R. Yamaguchi, K. Franklyn, D. Ortiz-Suslow, E. Creegan, and J. Fernando. Estimation of evaporation duct and surface-based duct parameters from a combined refractivity model. In *2018 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*, pages 879–880, 2018.

[56] A. Willitsford and C. Philbrick. Lidar description of the evaporative duct in ocean environments. *Proceedings of SPIE - The International Society for Optical Engineering*, 2005.

[57] C. Yang. A comparison of the machine learning algorithm for evaporation duct estimation. *Radioengineering*, 22:657–661, 2013.

[58] C. Yang, J. Zhang, and L. Guo. Investigation on the inversion of the atmospheric duct using the artificial bee colony algorithm based on opposition-based learning. *International Journal of Antennas and Propagation*, 2016.

[59] C. Yardim. Statistical estimation and tracking of refractivity from radar clutter. UC San Diego, 2007.

[60] C. Yardim, P. Gerstoft, and W.S. Hodgkiss. Estimation of radio refractivity from radar clutter using bayesian monte carlo analysis. *Antennas and Propagation, IEEE Transactions on*, 54:1318 – 1327, 2006.

[61] Z. Zhang, Z. Sheng, H. Shi, and Z. Fan. Inversion for refractivity parameters using a dynamic adaptive cuckoo search with crossover operator algorithm. *Computational intelligence and neuroscience*, 2016, 2016.

[62] Z. Zhou, J. Wu, and W. Tang. Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, 137:239–263, 2002.