



# **NAVAL POSTGRADUATE SCHOOL**

**MONTEREY, CALIFORNIA**

## **THESIS**

**SOURCE RANGE ESTIMATION USING BAYESIAN  
DEEP LEARNING FOR SINGLE ANTENNA  
SOFTWARE DEFINED RADIO RECEIVERS**

by

Tomas A. Carmona Richards

September 2023

Thesis Advisor:  
Second Reader:

Patrick McClure  
Frank E. Kragh

**Approved for public release. Distribution is unlimited.**

THIS PAGE INTENTIONALLY LEFT BLANK

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                 |                                                                |                                                         |                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|----------------------------------------------------------------|---------------------------------------------------------|------------------------------------------------------------|
| <b>REPORT DOCUMENTATION PAGE</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                 |                                                                | <i>Form Approved OMB<br/>No. 0704-0188</i>              |                                                            |
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                 |                                                                |                                                         |                                                            |
| <b>1. AGENCY USE ONLY</b><br>(Leave blank)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                 | <b>2. REPORT DATE</b><br>September 2023                        |                                                         | <b>3. REPORT TYPE AND DATES COVERED</b><br>Master's thesis |
| <b>4. TITLE AND SUBTITLE</b><br>SOURCE RANGE ESTIMATION USING BAYESIAN DEEP LEARNING FOR SINGLE ANTENNA SOFTWARE DEFINED RADIO RECEIVERS                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                 |                                                                | <b>5. FUNDING NUMBERS</b>                               |                                                            |
| <b>6. AUTHOR(S)</b> Tomas A. Carmona Richards                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                 |                                                                |                                                         |                                                            |
| <b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b><br>Naval Postgraduate School<br>Monterey, CA 93943-5000                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                 |                                                                | <b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>         |                                                            |
| <b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b><br>N/A                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                 |                                                                | <b>10. SPONSORING / MONITORING AGENCY REPORT NUMBER</b> |                                                            |
| <b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                 |                                                                |                                                         |                                                            |
| <b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b><br>Approved for public release. Distribution is unlimited.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                 |                                                                | <b>12b. DISTRIBUTION CODE</b><br>A                      |                                                            |
| <b>13. ABSTRACT (maximum 200 words)</b><br><br>The current physical model for radio frequency source-range estimation using single antenna receivers has substantial prediction error in real-world settings and does not provide uncertainty estimates for these predictions. The problem is amplified when the source is unknown and detailed information about its environment is not available. This research addresses this issue by developing deterministic and Bayesian deep neural network models that outperform the physical model on, and provide high-quality uncertainty estimates for, synthetic data, which were generated using a newly developed framework. Moreover, it presents the epistemic uncertainty as a rough indication of trust for the model's prediction, while acknowledging the challenges of establishing an absolute reliance threshold. Additionally, this thesis proposes using the physical model as a prior for the variational inference methods, resulting in models that outperform those using standard zero-mean Gaussian priors. This study paves the way for future development of new radio frequency source-localization techniques for simple single antenna Software Defined Radios, with applications in signals intelligence organizations, Maritime Search and Rescue Operations, and other domains where the use of complex receivers is not practical. |                                                                 |                                                                |                                                         |                                                            |
| <b>14. SUBJECT TERMS</b><br>BNN, DNN, SDR, range estimation, deep learning, uncertainty estimates, radio-frequency propagation, Bayesian methods                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                 |                                                                | <b>15. NUMBER OF PAGES</b><br>115                       |                                                            |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                 |                                                                | <b>16. PRICE CODE</b>                                   |                                                            |
| <b>17. SECURITY CLASSIFICATION OF REPORT</b><br>Unclassified                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | <b>18. SECURITY CLASSIFICATION OF THIS PAGE</b><br>Unclassified | <b>19. SECURITY CLASSIFICATION OF ABSTRACT</b><br>Unclassified | <b>20. LIMITATION OF ABSTRACT</b><br>UU                 |                                                            |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release. Distribution is unlimited.**

**SOURCE RANGE ESTIMATION USING BAYESIAN DEEP LEARNING  
FOR SINGLE ANTENNA SOFTWARE DEFINED RADIO RECEIVERS**

Tomas A. Carmona Richards  
Teniente Primero, Chilean Navy  
BSEE, Academia Politecnica Naval, 2015

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN COMPUTER SCIENCE**

from the

**NAVAL POSTGRADUATE SCHOOL  
September 2023**

Approved by: Patrick McClure  
Advisor

Frank E. Kragh  
Second Reader

Gurminder Singh  
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

## ABSTRACT

The current physical model for radio frequency source-range estimation using single antenna receivers has substantial prediction error in real-world settings and does not provide uncertainty estimates for these predictions. The problem is amplified when the source is unknown and detailed information about its environment is not available. This research addresses this issue by developing deterministic and Bayesian deep neural network models that outperform the physical model on, and provide high-quality uncertainty estimates for, synthetic data, which were generated using a newly developed framework. Moreover, it presents the epistemic uncertainty as a rough indication of trust for the model's prediction, while acknowledging the challenges of establishing an absolute reliance threshold. Additionally, this thesis proposes using the physical model as a prior for the variational inference methods, resulting in models that outperform those using standard zero-mean Gaussian priors. This study paves the way for future development of new radio frequency source-localization techniques for simple single antenna Software Defined Radios, with applications in signals intelligence organizations, Maritime Search and Rescue Operations, and other domains where the use of complex receivers is not practical.

THIS PAGE INTENTIONALLY LEFT BLANK

---

# Table of Contents

---

|          |                                                                          |           |
|----------|--------------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                                      | <b>1</b>  |
| 1.1      | Motivation . . . . .                                                     | 1         |
| 1.2      | Relevance for the DoD . . . . .                                          | 2         |
| 1.3      | Research Questions . . . . .                                             | 2         |
| 1.4      | Research Scope . . . . .                                                 | 2         |
| 1.5      | Thesis Organization . . . . .                                            | 4         |
| <br>     |                                                                          |           |
| <b>2</b> | <b>Methodology</b>                                                       | <b>7</b>  |
| 2.1      | General Problem Statement . . . . .                                      | 7         |
| 2.2      | Dataset Generation Framework . . . . .                                   | 9         |
| 2.3      | Waveform Simulator . . . . .                                             | 19        |
| 2.4      | Baseline Estimation Method . . . . .                                     | 30        |
| 2.5      | Machine Learning Methods . . . . .                                       | 31        |
| 2.6      | Evaluation Methods . . . . .                                             | 46        |
| <br>     |                                                                          |           |
| <b>3</b> | <b>Experimental Results</b>                                              | <b>51</b> |
| 3.1      | Organization . . . . .                                                   | 51        |
| 3.2      | Ideal Environment Results. . . . .                                       | 53        |
| 3.3      | Standard Environment Results . . . . .                                   | 63        |
| 3.4      | Harsh Environment Results . . . . .                                      | 72        |
| <br>     |                                                                          |           |
| <b>4</b> | <b>Conclusions and Discussion</b>                                        | <b>81</b> |
| 4.1      | Conclusions . . . . .                                                    | 81        |
| 4.2      | Discussion . . . . .                                                     | 82        |
| <br>     |                                                                          |           |
|          | <b>Appendix A Neural Network Architecture</b>                            | <b>85</b> |
| A.1      | Deterministic and Bayesian DNN Architectures . . . . .                   | 85        |
| A.2      | Data Preprocessing, Model Capacity, and Dynamic Range Related Issues . . | 87        |

|                                                            |           |
|------------------------------------------------------------|-----------|
| <b>Appendix B Parameters for Simulated RF Environments</b> | <b>89</b> |
| <b>List of References</b>                                  | <b>91</b> |
| <b>Initial Distribution List</b>                           | <b>97</b> |

---



---

## List of Figures

---

|             |                                                                  |    |
|-------------|------------------------------------------------------------------|----|
| Figure 2.1  | Dataset Generation Framework overview . . . . .                  | 10 |
| Figure 2.2  | PSD of the maximum allowable $R_s$ . . . . .                     | 13 |
| Figure 2.3  | Pitch and roll effects on antenna polarization angle . . . . .   | 17 |
| Figure 2.4  | Waveform Simulator . . . . .                                     | 19 |
| Figure 2.5  | Waveform IQ simulator baseband transmitter . . . . .             | 20 |
| Figure 2.6  | Channel effects on the transmitted signal . . . . .              | 23 |
| Figure 2.7  | Two-ray ground reflection model . . . . .                        | 25 |
| Figure 2.8  | Illustration of a heteroskedastic neural network . . . . .       | 32 |
| Figure 2.9  | Illustration of a deterministic NN v/s BNN . . . . .             | 39 |
| Figure 2.10 | Illustration of making predictions using BNN and MC-sampling .   | 42 |
| Figure 2.11 | Illustration of VI with physical model as prior . . . . .        | 45 |
| Figure 3.1  | Ideal environment representative example . . . . .               | 54 |
| Figure 3.2  | Ideal environment predictions with uncertainty metrics . . . . . | 56 |
| Figure 3.3  | Ideal environment aleatoric uncertainty . . . . .                | 58 |
| Figure 3.4  | Ideal environment epistemic uncertainty . . . . .                | 59 |
| Figure 3.5  | Ideal environment total uncertainty . . . . .                    | 60 |
| Figure 3.6  | Ideal environment epistemic uncertainty with unseen data . . . . | 62 |
| Figure 3.7  | Epistemic uncertainty with shifted domain . . . . .              | 63 |
| Figure 3.8  | Standard environment representative example . . . . .            | 64 |
| Figure 3.9  | Standard environment predictions with uncertainty metrics . . .  | 66 |
| Figure 3.10 | Standard environment aleatoric uncertainty . . . . .             | 67 |

|             |                                                                  |    |
|-------------|------------------------------------------------------------------|----|
| Figure 3.11 | Standard environment epistemic uncertainty . . . . .             | 68 |
| Figure 3.12 | Standard environment total uncertainty . . . . .                 | 69 |
| Figure 3.13 | Standard environment epistemic uncertainty with unseen data . .  | 70 |
| Figure 3.14 | Epistemic uncertainty with shifted domain . . . . .              | 71 |
| Figure 3.15 | Harsh environment representative example . . . . .               | 73 |
| Figure 3.16 | Harsh environment predictions with uncertainty metrics . . . . . | 74 |
| Figure 3.17 | Harsh environment aleatoric uncertainty . . . . .                | 76 |
| Figure 3.18 | Harsh environment epistemic uncertainty . . . . .                | 77 |
| Figure 3.19 | Harsh environment total uncertainty . . . . .                    | 78 |
| Figure 3.20 | Harsh environment epistemic uncertainty with unseen data . . . . | 79 |
| Figure 3.21 | Epistemic uncertainty with shifted domain . . . . .              | 80 |
| Figure A.1  | Feature engineering and architecture effects . . . . .           | 88 |

---

## List of Tables

---

|           |                                                                  |    |
|-----------|------------------------------------------------------------------|----|
| Table 2.1 | Beaufort wind scale . . . . .                                    | 15 |
| Table 3.1 | Environments characteristic summary . . . . .                    | 53 |
| Table 3.2 | Ideal environment overall performance . . . . .                  | 55 |
| Table 3.3 | Ideal environment domain shift and unseen data performance . . . | 61 |
| Table 3.4 | Standard environment overall performance . . . . .               | 65 |
| Table 3.5 | Standard environment domain shift and unseen data performance .  | 70 |
| Table 3.6 | Harsh environment overall performance . . . . .                  | 74 |
| Table 3.7 | Harsh environment domain shift and unseen data performance . .   | 78 |
| Table A.1 | Deterministic DNN architecture . . . . .                         | 85 |
| Table A.2 | BNN architecture . . . . .                                       | 86 |
| Table B.1 | Parameters for simulated environments . . . . .                  | 89 |

THIS PAGE INTENTIONALLY LEFT BLANK

---

## List of Acronyms and Abbreviations

---

|             |                                  |
|-------------|----------------------------------|
| <b>ADC</b>  | Analog to Digital Conversion     |
| <b>AGC</b>  | Automatic Gain Control           |
| <b>AOA</b>  | Angle of Arrival                 |
| <b>AWGN</b> | Additive White Gaussian Noise    |
| <b>BNN</b>  | Bayesian Neural Network          |
| <b>CNN</b>  | Convolutional Neural Network     |
| <b>DC</b>   | Direct Current                   |
| <b>DGF</b>  | Dataset Generation Framework     |
| <b>DL</b>   | Deep Learning                    |
| <b>DNN</b>  | Deep Neural Network              |
| <b>DoD</b>  | Department of Defense            |
| <b>GD</b>   | Gradient Descent                 |
| <b>GNLL</b> | Gaussian Negative Log-Likelihood |
| <b>KL</b>   | Kullback-Leibler Divergence      |
| <b>LOS</b>  | Line of Sight                    |
| <b>LRT</b>  | Local Reparameterization Trick   |
| <b>MAE</b>  | Mean Absolute Error              |
| <b>MAP</b>  | Maximum a Posterior              |
| <b>MC</b>   | Monte-Carlo                      |

|              |                                    |
|--------------|------------------------------------|
| <b>ML</b>    | Machine Learning                   |
| <b>MLE</b>   | Maximum Likelihood Estimation      |
| <b>M-PSK</b> | M-ary Phase Shift Keying           |
| <b>MSE</b>   | Mean Squared Error                 |
| <b>NF</b>    | Noise Figure                       |
| <b>NLL</b>   | Negative Log-Likelihood            |
| <b>NLOS</b>  | Non Line of Sight                  |
| <b>NN</b>    | Neural Network                     |
| <b>NPS</b>   | Naval Postgraduate School          |
| <b>PDF</b>   | Probability Density Function       |
| <b>PLF</b>   | Polarization Loss Factor           |
| <b>PSD</b>   | Power Spectrum Density             |
| <b>ReLU</b>  | Rectified Linear Activation Unit   |
| <b>RF</b>    | radio frequency                    |
| <b>RMS</b>   | Root Mean Square                   |
| <b>RSS</b>   | Received Signal Strength           |
| <b>RSSI</b>  | Received Signal Strength Indicator |
| <b>SDR</b>   | Software Defined Radio             |
| <b>SGD</b>   | Stochastic Gradient Descent        |
| <b>SPS</b>   | Samples per Symbol                 |
| <b>SRRC</b>  | Square Root-Raised Cosine          |
| <b>TDOA</b>  | Time Difference of Arrival         |

|            |                       |
|------------|-----------------------|
| <b>TOA</b> | Time of Arrival       |
| <b>UHF</b> | Ultra High Frequency  |
| <b>USN</b> | U.S. Navy             |
| <b>VHF</b> | Very High Frequency   |
| <b>VI</b>  | Variational Inference |

THIS PAGE INTENTIONALLY LEFT BLANK

---

# CHAPTER 1:

## Introduction

---

### 1.1 Motivation

The localization of radio frequency (RF) sources is an important military information asset for decision-making at all levels in the chain of command. Range estimation is the most basic form of radio-localization that can be achieved using a simple omnidirectional antenna with a single receiver. Although it is possible to use the mentioned setup to estimate the range to a source by means of Received Signal Strength Indicator (RSSI), this method has been shown to have a high prediction error [1]–[3]. Furthermore, this method does not provide uncertainty estimates for its predictions, which ideally should accompany the prediction to effectively reduce the doubts associated with the decision-making problem. Those issues led to the development of more complex and more specialized systems that usually require directional antennas, antenna arrays, and multiple receivers—an expensive setup that is not always feasible to carry and deploy.

On the other hand, the use of Software Defined Radios (SDRs) is ubiquitous in the military, and its software-based capabilities open a broad spectrum of signal analysis opportunities. These characteristics, in conjunction with the advent of Machine Learning (ML), in particular as Edge Deep Learning techniques, can provide new and exciting capabilities to simple single antenna radio systems. Research has shown the possibility to exploit such technologies in areas like receiver geo-localization through fingerprint maps, RF channel state estimation and even digital communication classification [4]–[6]. However, outdoor range estimation of a digital communication RF source in a simple single antenna-receiver setup, and its associated uncertainty, has not been addressed.

The methods covered in this work offer an approach that diverges from classical ones used to solve the outdoor range estimation problem. Deterministic Neural Networks (NNs) methods based on Maximum Likelihood Estimation (MLE) and Maximum a Posterior (MAP) estimation in conjunction with heteroskedastic modeling are used to obtain models that perform better in standard and harsh maritime conditions and provide aleatoric uncertainty estimates.

Epistemic uncertainty estimates are added to the predictions by means of Variational Inference (VI) techniques with a Gaussian variational distribution and the use of the analytical physical Received Signal Strength (RSS) model as prior.

## **1.2 Relevance for the DoD**

This thesis research is a contribution to the development of new RF source localization techniques based on NN methods and focused on simple SDRs using basic antennas. Organizations that could benefit from the study are those related to Signals Intelligence, Maritime Search and Rescue Operations, and others whose operational environment does not allow the deployment and use of multiple receivers and specialized antennas.

## **1.3 Research Questions**

The primary engineering objective of this study consists of the exploration of NN methods that outperform the physical models in estimating range to a radio-communication source from a single antenna receiver and provides high-quality uncertainty estimates for its predictions.

This primary objective is effectively addressed in this work by answering the following research questions:

- Can we develop a Deep Neural Network (DNN) model that predicts the range to an RF source using a single antenna receiver?
- Does such a DNN outperform the physical model?
- Can we get high-quality uncertainty estimates for the DNN predictions?
- Can the DNN generalize to environmental conditions not tested during training?

## **1.4 Research Scope**

The research examined the use of deterministic and Bayesian deep learning methods to predict source range from a received baseband radio frequency signal in IQ format. The models used consisted of exploratory architectures developed for this work that are available in Appendix A. No other families of ML techniques were compared during this work.

The explored scenario corresponds to a typical outdoor maritime communication environment in the Very High Frequency (VHF)/Ultra High Frequency (UHF) bands. The setup is composed of three basic elements: a transmitter, a channel, and a receiver. Moreover, every received signal is assumed to correspond to a single transmitter; thus, the problem of isolating multiple transmitters is outside the scope of this work. The maximum range of distances spans from the beginning of the transmitter's far-field approximation to the one defined by the radio horizon.

The transmitter's purpose is solely to adapt a symbol stream to the communication channel by the use of basic M-ary Phase Shift Keying (M-PSK) modulation schemes. Moreover, no communication protocol was implemented as part of the study. The output power used are those typically found in maritime communication equipment using a monopole antenna [7]. At the output of the transmitter, a standard Square Root-Raised Cosine (SRRC) filter was implemented; no other kind of pulse shaping filter was utilized.

The channel models the environmental factors that shape the received waveform that depends on many variables (e.g., receiver/transmitter imperfections, distance to the source, scatterers and humidity). Changes in these variables have a significant impact on the time and frequency representation of the received signal, which are the only features representation used for this study. From all the available channel variables, only the ones that produced the biggest effects on the waveform measurements were considered (see Section 2.2 for details).

The receiver purpose for this work is to save the received signals in their baseband form using the *SigMF* format [8], which is a popular format used in SDR communities. Also, no demodulation is performed and no other signal adaptation is expected. Only a fixed and known receiver amplification value is needed.

The construction of a synthetic dataset was mandatory given that the proposed topic has not been widely researched, and there are only a few compatible datasets, each having significant limitations. Some limitations include the use of a single modulation scheme, use of a fixed transmission power, use of communications over short distances only, and use of a small number different distances between the transmitter and the receiver.

The construction of a synthetic dataset provided the needed flexibility and could be used to establish the requirements to properly plan the construction of a real dataset in a future work.

Furthermore, this simulated approach gives some evidence about how feasible it is for a DNN to perform range estimation and how much data cleaning and pre-processing is required. Starting with synthetic data is reasonable, since models trained using synthetic datasets can demonstrate adequate real-world performance in radio frequency signal processing fields [9]. Furthermore, this study constitutes an exploratory analysis for the proposed problem, which is one of the common areas where the use of synthetic datasets is recommended [10].

Finally, the synthetic dataset was generated using the description of the transmitter, channel and receiver scopes described earlier in this chapter. To reduce the complexity of the dataset pre-processing, the training, validation and testing datasets were created in parallel using a uniform distribution of transmission ranges as explained in Section 2.2.5.

## 1.5 Thesis Organization

In Chapter 2, the research methodology is presented. This chapter is divided into six sections to address the most important methodological aspects of this work:

- Section 2.1, General Problem Statement, where the research problem of Chapter 1 is expressed mathematically.
- Section 2.2, Dataset Generation Framework (DGF), where the methods used to produce a synthetic dataset are explained.
- Section 2.3, Waveform Simulator, detailing how a single instance of a transmitted and received signal dataset pair is generated.
- Section 2.4, Baseline Estimation Method, presenting a classical physical method for solving the ranging problem. This method will serve as baseline benchmark for the ML methods in Chapter 3.
- Section 2.5, Machine Learning Methods, which explain the MLE, MAP and VI methods through the perspective of Section 2.1. Additionally, a summary of the details regarding training and optimization methods used are also presented. Finally, some important aspects of how distributions are implemented using NN are discussed.
- Section 2.6, Evaluation Methods, presenting the metrics used for performance evaluation and uncertainty estimation.

Chapter 3 summarizes the experimental results obtained from this thesis. The chapter is divided into various sections. For each simulated environment a section is presented providing evidence of advantages of ML methods over the baseline—in terms of prediction capabilities—and results of the uncertainty metrics.

This thesis report concludes with Chapter 4, which presents the conclusions obtained from the experimental results, as well as new findings. Furthermore, a small discussion section proposes future work and recommendations.

THIS PAGE INTENTIONALLY LEFT BLANK

---

## CHAPTER 2: Methodology

---

### 2.1 General Problem Statement

To address the ranging problem, one can describe the relationship between the actual range to a radio transmitter, denoted by  $y$ , and a received signal vector represented by  $\mathbf{x}$  as

$$y = h(\mathbf{x}), \quad (2.1)$$

where  $h$  corresponds to the underlying physical model that maps the received signal with the range from its source. Therefore, by finding  $h$ , the range for every possible received waveform could be obtained.

Nevertheless, the received signal from vector  $\mathbf{x}$  is noisy, given that it will be influenced by various random processes that occur between the transmitter and the receiver, as explained in Section 2.2. Consequently, the same received signal might have originated from transmitters at different ranges. This relationship can be better understood by considering the range  $Y$  and the signal vector  $\mathbf{X}$  as continuous random variables such that  $Y \sim P(y|\mathbf{x})$ .

Although the actual Probability Density Function (PDF)  $p(y|\mathbf{x})$  is unknown, sampling from its environment allows one to approximate it. The approximation can be thought of as  $p(y|\mathbf{x}, \mathcal{D})$ , where  $\mathcal{D}$  corresponds to the dataset containing the samples from the real environment.

Then, the guiding objective of this research is to find a proxy for  $p(y|\mathbf{x}, \mathcal{D})$  through a parameterized model using NN methods (i.e.,  $p(y|\mathbf{x}, \boldsymbol{\theta})$ ). In this context,  $\boldsymbol{\theta}$  denotes a vector that contains an instance of the parameter space, which represents the space of all possible models. However, it is necessary to consider that in practice, models are limited by their implementation and architecture  $\mathcal{M}$ , which comprises only a subset of the parameter space.

Using Bayesian Model Averaging and Monte-Carlo (MC) sampling techniques—needed to handle the typically intractable relationship between both distributions—an approximate

relationship between  $p(y|\mathbf{x}, \mathcal{D})$  and  $p(y|\mathbf{x}, \boldsymbol{\theta})$  is detailed in Equation (2.2) [11], where  $m$  corresponds to the number of samples taken from  $p(\boldsymbol{\theta}|\mathcal{D})$ .

$$p(y|\mathbf{x}, \mathcal{D}) \approx \frac{1}{m} \sum_{i=1}^m p(y|\mathbf{x}, \boldsymbol{\theta}^i) \quad \boldsymbol{\theta}^i \sim p(\boldsymbol{\theta}|\mathcal{D}), \quad (2.2)$$

From a Bayesian perspective,  $p(\boldsymbol{\theta}|\mathcal{D})$  is the *posterior* PDF of the parameters (i.e., the probability of the parameters given the data points contained in the dataset  $\mathcal{D}$ ) as shown in Equation (2.3).

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})} \quad (2.3)$$

Therefore, a reasonable prediction for the distance between a transmitter and a receiver would be the average distance between them based on their distribution. This value can be calculated using the following expression:

$$\hat{y}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \mathbb{E}_{p(y|\mathbf{x}, \boldsymbol{\theta}^i)} [y], \quad \boldsymbol{\theta}^i \sim p(\boldsymbol{\theta}|\mathcal{D}) \quad (2.4)$$

Similarly, a valid measure of uncertainty regarding the distribution can be calculated by predicting its variance using Equation (2.5).

$$\hat{\sigma}^2(\mathbf{x}) = \text{Var}_{\frac{1}{m} \sum_{i=1}^m p(y|\mathbf{x}, \boldsymbol{\theta}^i)} (y), \quad \boldsymbol{\theta}^i \sim p(\boldsymbol{\theta}|\mathcal{D}) \quad (2.5)$$

This entire research effort is built upon the aforementioned basic ideas in four different ways:

- First, by producing several synthetic datasets  $\mathcal{D}$  that simulate different maritime environments, as explained in Section 2.2 and 2.3.
- Second, by establishing a baseline benchmark using a classical analytical physical model as proposed in Section 2.4.
- Third, by training different types of deterministic and Bayesian NNs to obtain  $\hat{y}(\mathbf{x})$

and  $\hat{\sigma}^2(\mathbf{x})$  using the methods shown in Section 2.5.

- Finally, by evaluating those models and measuring their uncertainties, as explained in Section 2.6.

Implementation details of the NNs used are not included in this chapter, since the methods presented are general enough to accommodate a broad range of NN architectures. This decision aligns with the research’s primary objective, which focuses on the NN methods rather than fine-tuning to achieve optimal performance. Nonetheless, more comprehensive information regarding the NN architectures utilized in this work is presented in Appendix A<sup>1</sup>.

## 2.2 Dataset Generation Framework

The purpose of this section is to establish the methods used to generate a specific instance of the dataset  $\mathcal{D}$ . Note that the better the match  $p(\mathcal{D}_{\text{simulated-environment}}) \sim p(\mathcal{D}_{\text{real-environment}})$ , the better the predictions will be. This is the case because the predictions depend greatly on the dataset used as shown in Section 2.5.

This consideration is crucial since a synthetic dataset is being proposed in this thesis to approximate the real environment, as stated in Section 1.4. For this reason, a reduction of the dataset’s quality must be assumed, with the corresponding decrease of the ML model prediction performance in the real environment<sup>2</sup>.

The dataset  $\mathcal{D} = \{(y_1, \mathbf{x}_1), (y_2, \mathbf{x}_2), \dots, (y_i, \mathbf{x}_i), \dots, (y_k, \mathbf{x}_k)\}$  is comprised of pairs  $(y_i, \mathbf{x}_i)$ , with  $y_i$  representing the simulated distance between a transmitter and a receiver measured in meters and with  $\mathbf{x}_i = (z_{tx,i}[n], z_{rx,i}[n])$  representing the sampled complex envelope of the ideal reconstructed transmitted waveform  $z_{tx,i}[n]$  and the sampled complex envelope of the received signal  $z_{rx,i}[n]$ . The pairs are generated and associated together using the DGF.

The transmitted signal,  $z_{tx,i}[n]$ , is appended to the input vector  $\mathbf{x}_i$  in order to incorporate meaningful information about the transmitter, as required by the proposed NN architecture. This is necessary to enhance the capabilities of the developed architecture and enable it to make predictions for a wider range of transmitters, as detailed in Appendix A.

---

<sup>1</sup>The NN architecture source code is available at: <https://gitlab.nps.edu/thesis2/models>.

<sup>2</sup>The DGF source code is available at: <https://gitlab.nps.edu/thesis2/dataset>.

Figure 2.1 presents an overview of the DGF architecture. The bold labels on the dataflows indicate the parameters that are required to generate a single instance of the dataset, while the red labels indicate the locations within the architecture where the dataset pairs  $(y_i, \mathbf{x}_i)$  are generated.

With this framework, it is possible to generate arbitrarily large datasets, which simulate multiple transmitters in a common environment. The boundary parameters define the complexity of the random effects that shape the simulated received signal for a single receiver.

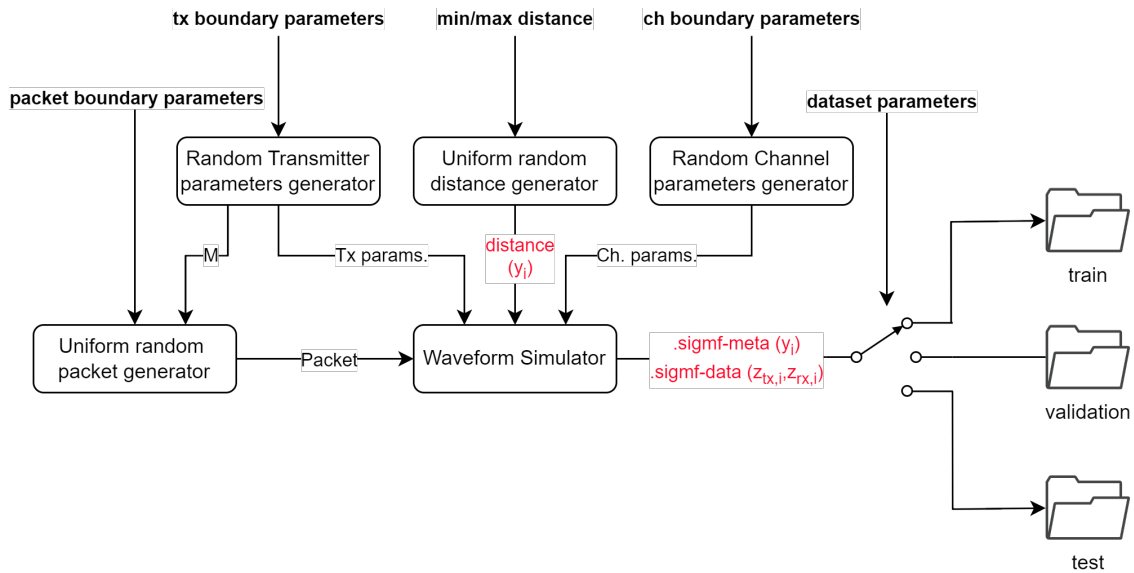


Figure 2.1. Dataset Generation Framework overview.

The primary objective of the DGF is to generate the parameters necessary for running multiple simulations and organizing the results into a single instance of the dataset. Meanwhile, the Waveform Simulator's purpose is to generate a single simulation based on the provided parameters. A comprehensive explanation of the Waveform Simulator is given in Section 2.3.

### 2.2.1 Uniform Random Packet Generator

The intention of a Digital Communication system is to transmit information over a channel. However, for the purposes of this research, the actual information conveyed is irrelevant. Despite this fact, the simulator needs a stream of random symbols to accurately represent a signal using a predefined modulation scheme.

The length of the input vector  $\mathbf{x}_i$  varies proportionally to the number of symbols transmitted. By defining a sequence of symbols as a packet, the vector length is determined by both the packet size  $L_i$  and the number of Samples per Symbol (SPS), as expressed in Equation (2.6).

$$\text{length}(\mathbf{x}_i) = 2L_i\text{SPS} \quad (2.6)$$

In this equation, the factor of two reflects that the input vector comprises both the transmitted and received waveforms. Additionally, the SPS variable is a randomly generated parameter based on Equation (2.10). This relationship is essential in determining the necessary resource requirement for model training in conjunction with the model architecture.

The packet lengths are sampled from a discrete uniform random distribution as shown in Equation (2.7), which is determined by the limiting parameters  $L_{min}$  and  $L_{max}$ . These parameters establish the minimum and maximum length, respectively, for a packet in the simulated environment.

$$L_i \sim \mathcal{U}\{L_{min}, L_{max}\} \quad (2.7)$$

This approach was chosen because, as mentioned in Section 1.4, there is no knowledge regarding the content or structure of the packets. Hence, it is necessary to assume that every packet length is equally likely to occur. This assumption is crucial, as the model's performance should not be affected by the packet size.

Similarly, the symbols within the packets are uniformly randomly distributed. This assumption is made due to the lack of knowledge about the content or structure of the packets, which applies to this scenario. This is especially true as the data is likely to be encrypted, scrambled using a pseudo-randomizer before transmission, or compressed via source en-

coding, as is common practice in Digital Communication systems [12]. In Equation (2.8), the packet  $S_i$  corresponds to the sequence:

$$S_i = [s_1, s_2, \dots, s_{L_i}], \quad s_j \sim \mathcal{U}\{1, M\} \quad \forall j \in \{1, \dots, L_i\} \quad (2.8)$$

The variable  $M$  represents the number of symbols in the  $M$ -ary modulation scheme used in the simulation instance.

### 2.2.2 Random Transmitter Parameters Generator

The proposed maritime environment allows for the use of multiple and distinct transmitters. This implies that the model should be trained in an environment where different signals are possible. The purpose of this block in the DGF is to randomly generate the transmitters, following the parameters defined in this Section.

#### Transmitter power parameter

Maritime radio equipment typically provides the operator with a fixed set of transmitting power options. A common approach is to set and maintain the power required for certain communication needs that are shared across multiple units. Therefore, the simulation environment is limited to a discrete number of power transmission choices (refer to Appendix B for a comprehensive list of selected parameters). Due to this constraint, the power selection process randomly chooses from the available power mode list using a uniform distribution.

#### Modulation scheme parameter

The  $M$ -ary modulation scheme used for this research is  $M$ -PSK. The number of symbols, denoted by  $M$ , is randomly selected from a set where each element is equally likely to be chosen. The set is passed as a parameter list to the DGF, while the values of the elements must be a power of two, such that  $M = 2^k$ , for all  $k \in \{1, 2, 3, \dots\}$ . This approach restricts the DGF from simulating a wide variety of transmitters. Nevertheless, the implementation code is adaptable, which allows for the inclusion of new schemes in future work. Note that  $M$  is also required for the uniform random packet generator discussed in Section 2.2.1. Refer to Appendix B for more information about the parameter  $M$  used in the simulations.

### Symbol rate parameter

The symbol rate  $R_s$ —an essential transmitter parameter—is measured in symbols per second or *bauds*. In most cases, the radio operator can only choose from a set of fixed rates. Even so, to achieve the model’s goal of being as versatile as possible, a discrete uniform random sampling method is used to obtain the data rate parameter instance. This distribution is governed by the complex sample rate  $R_{\text{sampling}}$  and the pulse shaping filter parameter  $\beta$ , which is explained in Section 2.3.1. Equation (2.9) shows the expression for obtaining the symbol rate:

$$R_s \sim \mathcal{U} \left\{ R_{\min}, \left\lfloor \frac{0.8 R_{\text{sampling}}}{1 + \beta} \right\rfloor \right\} \quad (2.9)$$

The predefined parameter  $R_{\text{sampling}}$  remains constant throughout the dataset instance. In reality, the SDR receiver determines  $R_{\text{sampling}}$ ; however, it is easier to define it as part of the transmitter for the purposes of simulation. Equation (2.9) is limited by the bandwidth of the signal  $B_s \propto (1 + \beta)R_s$  and the maximum bandwidth available for the complex sampling frequency, which is determined by the Nyquist frequency criteria for complex signals [13]. Specifically,  $R_{\text{sampling}}$  must satisfy  $R_{\text{sampling}} > B_{\max}$  to prevent aliasing. The scalar factor of 0.8 provides an additional 20% bandwidth gap to minimize the risk of aliasing. Figure 2.2 depicts an example of a Power Spectrum Density (PSD) function that reflects the maximum allowable  $R_s$  for a given  $\beta$  and  $R_{\text{sampling}}$ .

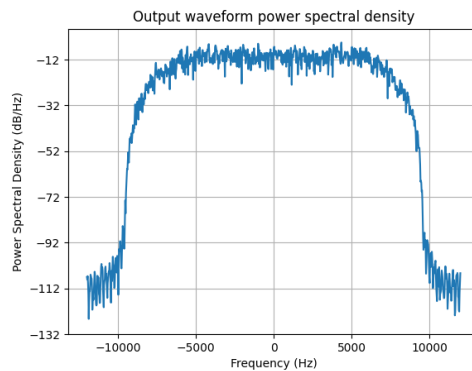


Figure 2.2. PSD of the maximum allowable  $R_s$  for  $R_{\text{sampling}} = 24,000$  and  $\beta = 0.35$ .

The number of SPS is a useful intermediary parameter that is utilized in various components of the simulator in Section 2.3. It is necessary in Equation (2.6), and its value can be obtained from Equation (2.10):

$$\text{SPS} = \frac{R_{\text{sampling}}}{R_s} \quad (2.10)$$

### 2.2.3 Uniform Random Distance Generator

The generator block for random distances depicted in Figure 2.1 has been segregated from the Random Channel Parameters Generator for highlighting its pivotal role. Being the source of the target variable  $y$  for the dataset, this block assumes added significance. It facilitates the generation of a continuous uniform distribution of distances  $Y \sim \mathcal{U}(r_{\min}, r_{\max})$ .

As a result, each distance in the dataset has an equal probability of being sampled. This approach ensures that the model is trained with a comparable amount of data for every distance, eliminating imbalanced target variables that can arise in regression problems. Such imbalances are notoriously difficult to handle and remain an open question, as noted by Krawczyk [14].

### 2.2.4 Random Channel Parameters Generator

The Random Channel Parameters Generator depicted in Figure 2.1 encompasses all the parameters that model the channel effects on the transmitted waveform. This block is dependent on the boundary parameters, defined beforehand, generating a particular instance of the dataset.

The channel parameters are defined based on the propagation model outlined in Section 2.3.2, as well as the multipath effects, and the noise effects. For further clarity on these effects, please refer to the aforementioned section.

### Effective antenna height fluctuations

An important constraint for the simulation environment is that it cannot assume that the transmitter and the receiver are static at sea. Waves will cause the antennas to move up and

down over the median sea level, known as the ship's heave, and change the polarization angle of the electric field by effects of the ship's pitch and roll. Since linearly polarized antennas are often used, both of these effects will impact the effective power received, which is a product of the path loss shown in Equation 2.22. Ship heave will produce both positive and negative non-linear effects on the power received, while pitch and roll will have always a negative linear impact on the received power. These considerations are significant because they reduce the ability of the baseline model explained later in Section 2.4 to provide accurate predictions. To effectively consider these phenomena, it is necessary to practically estimate the wave conditions.

Knowledge about the sea state can be estimated through local wind conditions. A common approach is to approximate the wave peak-to-peak amplitude using the Beaufort Wind Scale shown in Table 2.1 [15]. This scale correlates a quantifiable level of wind condition with an expected wave height.

Table 2.1. Beaufort wind scale. Adapted from [15, table 5].

| Level | Name            | Wave height [m] |
|-------|-----------------|-----------------|
| 0     | Calm            | 0.0             |
| 1     | Light air       | 0.1             |
| 2     | Light breeze    | 0.2 - 0.3       |
| 3     | Gentle breeze   | 0.6 - 1.0       |
| 4     | Moderate breeze | 1.0 - 1.5       |
| 5     | Fresh breeze    | 2.0 - 2.5       |
| 6     | Strong breeze   | 3.0 - 4.0       |
| 7     | Near gale       | 4.0 - 5.5       |
| 8     | Gale            | 5.5 - 7.5       |
| 9     | Strong gale     | 7.0 - 10.0      |
| 10    | Storm           | 9.0 - 12.5      |
| 11    | Violent Storm   | 11.5 - 16.0     |
| 12    | Hurricane       | 16.0+           |

Although one might think that this approximation would allow an observer on the receiver unit to produce a more accurate prediction of the transmitter distance through the technique explained in Section 2.4, it is impractical in nature. For example, the amount of time the unit's heave will remain in a fixed position will be small and hard to anticipate, the actual effects of the waves on the antenna height and angle will be difficult to measure, and no useful information about the transmitter antenna could be deduced. Nonetheless, this effect will still be present and contribute to the variance of the measured received signals. A simple approach consists of taking independent samples of height errors from a uniform distribution based on the Beaufort scale and adding those to the actual height of the transmitter and receiver, respectively.

Furthermore, in a Line of Sight (LOS) environment, one can assume that both the transmitter and the receiver will experience comparable sea conditions, considering that similar weather patterns tend to spread across long distances. This assumption implies that the same Beaufort level can be used for the transmitter and receiver.

From these reasons, the Random Channel Parameters Generator samples possible antenna heights using the nominal transmitter and receiver antenna height,  $h_{\text{nominal-tx}}$  and  $h_{\text{nominal-rx}}$ , and the peak-to-peak amplitude of the wave,  $h_{\text{wave}}$ , as expressed in Equation (2.11).

$$h_{tx/rx} = h_{\text{nominal-tx/rx}} + h_{\text{error}}, \quad h_{\text{error}} \sim \mathcal{U} \left[ -\frac{h_{\text{wave}}}{2}, \frac{h_{\text{wave}}}{2} \right] \quad (2.11)$$

$h_{\text{nominal-tx}}$ ,  $h_{\text{nominal-rx}}$ , and the Beaufort level that maps to  $h_{\text{wave}}$  are predefined and maintained fixed throughout the simulation instance.

### Antenna polarization mismatch

As previously explained, the pitch and roll of the ship will create a polarization mismatch between the transmitter and receiver antennas, resulting in a decrease in the power received at the distant end. This power loss is referred to as the Polarization Loss Factor (PLF) and is bounded between zero and one. The PLF ( $L_{\text{PLF}}$ ) will be one, when there is no angle difference between the transmitter and receiver, and zero, when the difference corresponds to a 90° angle between a pair of linear polarized antennas.

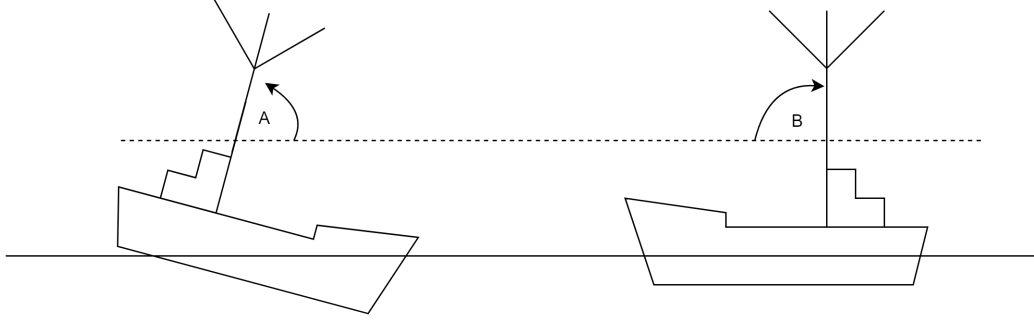


Figure 2.3. Pitch and roll and its effect on the antenna polarization angle.

The  $L_{PLF}$  is only dependent of the antenna angle difference  $\phi = A - B$  given by [16]:

$$L_{PLF} = \cos^2 \phi \quad (2.12)$$

Once more, the maximum angle difference is sampled uniformly from the highest permitted angle, which is determined based on the Beaufort Wind Scale as  $\phi \sim \mathcal{U}[0, \phi_{max}]$ . It should be noted that the specific maximum angle and its variability depend largely on the mechanical properties of the ship as well as its interaction with the ocean waves that are outside the scope of the present work. Thus, a basic scale ranging from 0 to 13 degrees was implemented to correspond with the 13 levels established by the Beaufort Scale.

### **Multipath random parameters**

The DGF was developed with the ability to simulate a variable multipath environment. In this scenario, reflected signals may arrive from different paths produced by the ocean waves, other vessels in proximity, or even the hull and superstructure of the units participating in the communication. The multipath sources will vary from transmission to transmission because the ocean conditions are constantly changing, and the ships are also non-stationary. The number of scatterers is sampled from a uniform distribution, with the maximum parameter pre-defined as a dataset instance parameter.

After sampling the number of scatterers, vectors of amplitudes, phases and delays are obtained using uniform distributions. All these parameters are necessary for the waveform simulator to generate the signal replicas and add them together to simulate the overall multipath effect.

It is worth mentioning that the parameters generated here do not count the main multipath reflection, which is typically accounted for as part of the propagation loss, as explained in the Path-Loss Subsection of Section 2.3.2.

### **Other channel random parameters**

The movement of the transmitter relative to the receiver results in a frequency shift on the received signal, referred to as the Doppler effect. Moreover, the local oscillator at the receiver might be tuned off to avoid the issues related to the Direct Current (DC) offset, which is a common problem found in SDR systems using Direct Conversion Receivers [17]. These factors contribute to an overall frequency offset to the down converted complex envelope signal, which can be modeled using a single random variable offset sampled from a uniform distribution  $f_{\text{offset}} \sim \mathcal{U}[0, f_{\text{max offset}}]$ . The maximum offset boundary is consistent throughout the dataset instance generation.

The Waveform Simulator directly simulates noise-like effects, such as noise temperature and quantization noise, since their random nature does not necessitate the generation of random parameters to model their effects across different signals on a dataset instance.

## **2.2.5 Dataset Splitting**

The *dataset parameters* aggregate the variables used to define the dataset size, the ratio of validation to training, the ratio of testing to training, and other miscellaneous parameters used to uniquely identify and generate the dataset, like names, file paths, and so on.

As illustrated, the framework directly generates and splits the dataset into training, validation and testing sets in accordance with established ML methods outlined in Section 2.5. These parameters must be defined before generating a new dataset instance and cannot be altered until the process is complete.

## 2.3 Waveform Simulator

The purpose of the *Waveform Simulator* is to generate a single simulated received waveform defined with the parameters generated in the DGF<sup>3</sup>. It is composed of a transmitter, a channel and a receiver as shown in Figure 2.4.

The transmitter is in charge of producing a random signal  $z_{tx,i}[n]$ . The channel, which heavily depends on the distance  $y$  between the source and the receiver, will transform the transmitted signal into the received signal  $z_{rx,i}[n]$ . And the receiver will save all the signal parameters along with the received and transmitter waveform to generate the dataset pairs.

The signals are presented as baseband sampled complex envelope signals from the standpoint of a homodyne receiver, as this is a common representation used in real systems for recording and analyzing signals using SDRs [18].

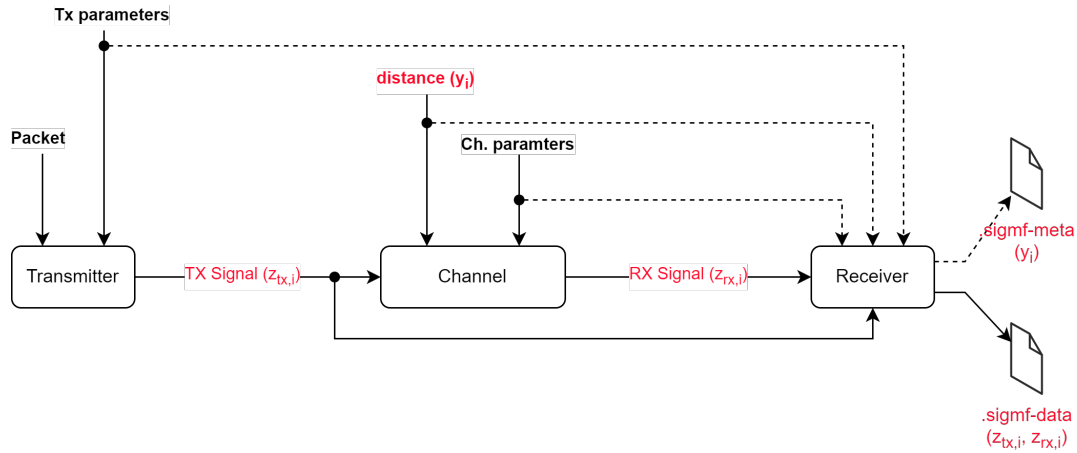


Figure 2.4. Waveform Simulator architecture.

Since the receiver in this simulator only transforms the parameter data into a metadata file and provides no further explanation of the transmitted and received waveforms, a detailed discussion is omitted. However, implementation details can be found in the NPS GitLab repository that accompanies this work.

<sup>3</sup>The DGF source code is available at: <https://gitlab.nps.edu/thesis2/dataset>.

### 2.3.1 The Transmitter

The objective of the transmitter is to generate a sampled version of the baseband representation of the signal. This is achieved by using the baseband M-PSK IQ transmitter shown in Figure 2.5 to produce a perfectly transmitted signal  $z_{tx}[n]$ .

In this setup, the transmitted baseband (i.e., complex envelope) signal  $z_{tx}(t)$  is related by the transmitted passband signal  $s(t)$  as shown by:

$$s_{tx}(t) = \text{Re}\{z_{tx}(t)\} \cos(2\pi f_c t) - \text{Im}\{z_{tx}(t)\} \sin(2\pi f_c t), \quad (2.13)$$

where  $f_c$  denotes the carrier frequency, and

$$z_{tx}(t) = I_{tx}(t) + jQ_{tx}(t), \quad (2.14)$$

where the In-phase component is denoted by  $I(t)$ , and the Quadrature component is denoted by  $Q(t)$ . Together, they form the IQ or baseband representation of the signal.

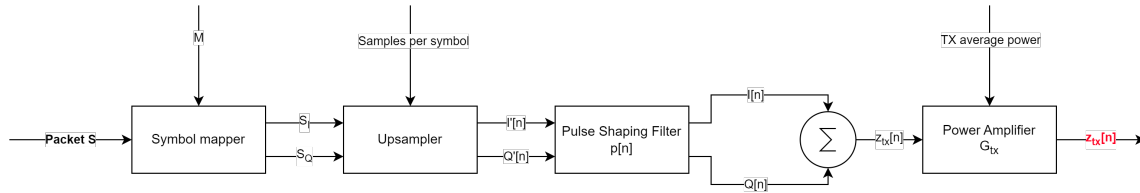


Figure 2.5. Waveform IQ simulator baseband transmitter.

Note that the mixers implementing the cosine and sine components of Equation (2.13) are not included in the proposed modulator. This omission is intended to preserve the baseband representation of the signal, which helps maintain independence of the carrier frequency and reduces the complex sampling rate ( $R_{\text{sampling}}$ ) requirements needed to comply with the Nyquist sampling rate criterion [13].

The conversion from the continuous time domain to the discrete regime is determined by the sampling rate  $R_{\text{sampling}}$  and is bounded by the signal length governed by Equation (2.7),

$$z[n] = z\left(\frac{n}{R_{\text{sampling}}}\right) \quad n \in \{1, \dots, \text{length}(\mathbf{x})\} \quad (2.15)$$

### Symbol mapper

The first step in generating the transmitted signal is to map the packet's symbols  $s_i \in S$  to the constellation diagram for M-PSK, where  $S$  is the packet as defined in Equation (2.8). This process involves assigning a phase to each symbol, which for a M-PSK signal is given by the relationship:

$$\phi_i = \frac{(2s_i - 1)\pi}{M} \quad (2.16)$$

where  $M$  is the number of symbols in the corresponding M-PSK modulation scheme.

Each symbol can later be represented in its IQ form using the following relationship:

$$\begin{aligned} s_{I,i} &= \cos \phi_i \\ s_{Q,i} &= \sin \phi_i \end{aligned} \quad (2.17)$$

As a result, the complete IQ symbol stream can be thought of as sequences  $S_I = [s_{I,1}, s_{I,2}, \dots, s_{I,L}]$  and  $S_Q = [s_{Q,1}, s_{Q,2}, \dots, s_{Q,L}]$ , where the sub-index  $L$  indicates the length of the packet, as defined in Section 2.2.1.

### Upsampler

As shown in Figure 2.5, the Upsampler takes an IQ symbol element  $S_{I/Q,i}$  and increases the number of samples by concatenating zeros to match the sample rate. The number of zeros appended is determined by the SPS parameter, which is calculated using Equation (2.10). Equation (2.18) is then used to obtain  $I'[n]$  and  $Q'[n]$ .

$$I'[n] = \begin{cases} S_{I,i}, & \text{if } n = \text{SPS}(i-1) + 1 \quad i \in 1, 2, \dots, L \\ 0, & \text{otherwise} \end{cases} \quad (2.18)$$

### Pulse Shaping Filter

The Pulse Shaping Filter transforms the upsampled symbol streams  $I'[n]$  and  $Q'[n]$  into actual sampled waveforms. In this simulator, a standard SRRC pulse with roll-off parameter  $\beta$  is used as the pulse shape  $p[n]$ . The  $I[n]$  and  $Q[n]$  waveforms are obtained by convolving the upsampled sequences with  $p[n]$ , as shown in Equation (2.19):

$$\begin{aligned} I[n] &= \sum_{k=0}^m I'[k]p[n-k] \\ Q[n] &= \sum_{k=0}^m Q'[k]p[n-k] \end{aligned} \quad (2.19)$$

### Power amplifier

The final step in the waveform generation process is to amplify the IQ waveforms to match the average transmitted output power parameter  $P_{tx}$ . To achieve this, the transmitter calculates the overall voltage gain as shown in Equation (2.20):

$$G_{tx} = \sqrt{\frac{P_{tx}}{\frac{1}{n} \sum_{i=1}^n I[i]^2 + Q[i]^2}} \quad (2.20)$$

The transmitted signal  $z_{tx}[n]$  is simply obtained by multiplying  $G_{tx}$  by the IQ waveforms at the output of the transmitter.

### 2.3.2 The Channel

The channel encapsulates all the waveform distortions that affect the signal from the perfectly generated signal at the transmitter to the recorded received signal. Figure 2.6 shows the concatenation of effects and how they are applied to the transmitted signal,  $z_{tx}[n]$ , to produce the received signal,  $z_{rx}[n]$ .

Since an IQ version of the signal is being used, an equivalent lowpass version of the bandpass system representing the channel must also be used. The equivalent lowpass system representation is almost identical to its bandpass version, with the only difference being the presence of a constant factor of  $1/2$  [13], which can be safely omitted for most simulation purposes.

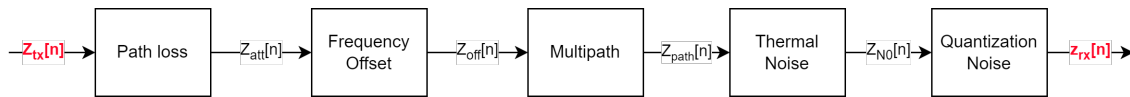


Figure 2.6. Channel effect on the transmitted signal.

Undoubtedly, there are other effects that can degrade the quality of predictions made using the baseline model of Section 2.4 in an SDR environment, such as DC offset, IQ imbalances, frequency jitter, transmission line loss, and weather conditions like rain and snow, to name a few. For the purpose of this work, the effects shown in Figure 2.6 are sufficient to demonstrate the benefits of using NN over the analytical baseline model. Nonetheless, the DGF is flexible enough to accommodate the inclusion of such effects in the simulator for future studies.

#### Path loss

In Section 1.4, the research scope established a typical maritime environment. This implies that a simple free-space path loss model is not the best option, as both the transmitter and the receiver are close to the surface of the Earth, where reflections with the surface are inevitable.

There are two general approaches to obtaining the path loss model for such an environment. First, if one has access to specific knowledge about the environment, using site-specific

models is recommended. Otherwise, when such information is not available, a model for average propagation conditions will yield better results [19].

Site-specific models are highly dependent on the current conditions of the dataset used; therefore, they should only be used if one is completely certain about the conditions, as significant errors (up to 30 dB) could arise [19].

Maritime environments are diverse, and in general, having a site-specific model is not possible. This is even more evident in the case of wars or conflicts in foreign waters. This is the primary reason for choosing an empirical model for average propagation. The Friis equation is an empirical model that represents the relationship between the received power of a signal  $P_{rx}$ , the average transmitted power  $P_{tx}$ , the transmitted and received antenna gains  $G_{rx}$  and  $G_{tx}$ , the path loss  $L_{path}$ , and the polarization factor  $L_{PLF}$ , which accounts for the polarization mismatch between the antennas [19]. Equation (2.21) presents the attenuation ratio  $P_{rx}/P_{tx}$ :

$$\frac{P_{rx}}{P_{tx}} = G_{tx} G_{rx} L_{path} L_{PLF} \quad (2.21)$$

As explained in Sections 2.2.2 and 2.2.4,  $P_{tx}$ ,  $G_{tx}$ , and  $G_{rx}$  are pre-defined parameters for the simulated instances and are known beforehand.  $L_{PLF}$  was defined in Equation (2.12). Therefore, only  $L_{path}$  needs to be defined to have a complete relationship between the transmitted and received signals, in terms of the power loss experienced due to the characteristics of the channel.

In this work, the Two-ray Ground Reflection model was employed due to its ability to provide more challenging path-loss predictions compared to the classical free-space model. Additionally, it offers greater accuracy for longer distances in unobstructed ship-to-ship LOS communications [19], [20]. Moreover, this model has demonstrated potential for the proposed maritime environment by producing reasonable predictions for communications near the sea surface [21], [22].

As depicted in Figure 2.7, the model is derived using a simple optical approach wherein a direct and reflected ray, over a smooth and flat surface, interfere with each other, resulting in a path-loss profile that exhibits distinctive interference zones [19], [20].

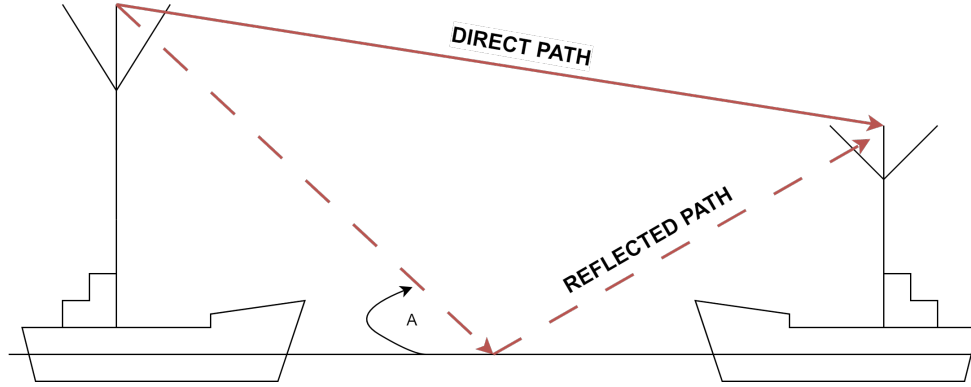


Figure 2.7. Two-ray ground reflection model.

For distances that are significantly larger than the square root of the product of the antenna heights (i.e.,  $y \gg \sqrt{h_{tx}h_{rx}}$ ), the Two-ray model reduces to Equation (2.22):

$$L_{\text{path}} = \frac{h_{tx}^2 h_{rx}^2}{y^4} \quad (2.22)$$

Since predicting distances over larger ranges is a key focus of this thesis, the reduced path-loss equation is being utilized. The inclusion of this model's complexity has demonstrated its ability to provide evidence regarding the advantages of the proposed machine learning (ML) methods (Section 2.5) compared to the traditional baseline (Section 2.4), as presented in Chapter 3.

However, it is important to exercise caution as deeper nulls have been observed under various conditions and at larger distances in the real-world. These nulls are suggested to arise from an additional ray path [20], [22].

It is worth noting the crucial role of the antenna heights for both the transmitter and receiver, denoted by  $h_{tx}$  and  $h_{rx}$ , respectively. The sea conditions, as explained in Section 2.2.4, may affect both antenna heights.

To calculate the attenuated transmitted signal  $z_{att}[n]$ , first obtain the overall voltage attenuation ratio using Equation (2.21) together with the path loss model given by Equation (2.22). Then multiply this ratio by the original transmitted signal to obtain the attenuated signal.

$$z_{att}[n] = z_{tx}[n] \sqrt{\frac{P_{rx}}{P_{tx}}} \quad (2.23)$$

### Frequency offset

While modern SDRs have stable local oscillators with precise frequency tuning, there is still the possibility of frequency offset  $f_{\text{offset}}$  caused by oscillator drift over time or inaccurate operator tuning. In order to account for this scenario, the received wave can be shifted using the modulation theorem, where  $R_{\text{sampling}}$  corresponds to the sampling rate:

$$z_{off}[n] = z_{att}[n] e^{j2\pi f_{\text{offset}} \frac{n}{R_{\text{sampling}}}} \quad (2.24)$$

Although the frequency offset typically does not affect the power of the received signal, leading to accurate range estimation using RSSI-based techniques, it can still impact the generalization of NN models that utilize the frequency representation of the signal.

### Multipath effects

Multipath refers to the reception of multiple copies of a signal at different times, with varying phases and amplitudes, as the transmitted signal reflects off objects (scatterers) in its path to the receiver. This effect can result in constructive or destructive interference, affecting the RSS. The overall effect is known as fading. As such, it has a significant impact on RSS-based ranging techniques and is often the primary source of error [2]. The multipath effect can be characterized by the channel impulse response  $h(t)$ , which is expressed in Equation (2.25) [2].

$$h[n] = \sum_{k=1}^N \alpha_k e^{j\phi_k} \delta[n - \tau_k] \quad (2.25)$$

Equation (2.25) shows that the channel impulse response  $h(t)$  comprises all  $N$  multipath components, with  $a_k$ ,  $\phi_k$ , and  $\tau_k$  representing the amplitude, phase, and propagation delay of the  $k$ -th reflected signal component, respectively. Convolution is used to obtain the combined received waveform  $z_{\text{path}}[n]$ , as shown in Equation (2.26).

$$z_{\text{path}}[n] = h[n] * z_{\text{off}}[n] \quad (2.26)$$

It is worth noting that the frequency offset effect is applied prior to the multipath effect, as it has the same impact on all multipath copies of the signal.

### Receiver noise

The receiver circuits constitute a significant source of signal noise, which is an unavoidable physical phenomenon that can result in signal distortion and diminish the receiver's capability to detect small signals. The received signal Root Mean Square (RMS) voltage noise is related to the RF power noise by Equation (2.27) [23]:

$$\frac{V_{\text{noise}}^2}{R} = KT_n B \quad (2.27)$$

In this equation,  $K$  represents Boltzmann's constant,  $T_n$  represents the receiver's noise temperature in Kelvin,  $R$  represents the system impedance in ohms, and  $B$  denotes the system's bandwidth measured in Hertz. The simulator's system bandwidth  $B$  is derived from the complex sampling frequency  $R_{\text{sampling}}$ , with the condition that the anti-aliasing filters remove any noise outside the bandpass range for the SDR.

A commonly used figure of merit to describe a receiver performance is the Noise Figure (NF). The noise temperature,  $T_n$ , is related to the NF and a reference temperature  $T$ , by Equation (2.28) [23]:

$$T_n = T \left( 10^{\frac{NF}{10}} - 1 \right) \quad (2.28)$$

Since the simulated environment involves a single receiver, both, the noise figure (NF) and the reference temperature  $T$ , are predetermined simulation parameters that remain fixed throughout the entire dataset instance. The reference temperature is commonly set to room temperature, which is equivalent to  $293^\circ K$ .

While it is impossible to predict the voltage noise at a particular moment, it can be estimated using a Gaussian probability distribution with a zero mean and a standard deviation of  $V_{\text{noise}}$  [23]:

$$p(v) = \frac{1}{\sqrt{2\pi V_{\text{noise}}^2}} e^{-\frac{v^2}{2V_{\text{noise}}^2}} \quad (2.29)$$

Equation (2.29) cannot be directly utilized to obtain noise samples for the I and Q channels, as the total variation of the noisy transmitted RF signal,  $z_{N0}[n]$ , should be distributed between the I and Q channels. Assuming that I and Q are independent and have equal variances ( $\text{Var}(I) = \text{Var}(Q)$ ),  $z_{N0}[n]$  is calculated using the following equation:

$$z_{N0}[n] = \sqrt{G_{\text{front-end}}} (z_{\text{path}}[n] + V_I[n] + jV_Q[n]) \quad (2.30)$$

$$V_I[n], V_Q[n] \sim \mathcal{N}\left(0, \frac{V_{\text{noise}}^2}{2}\right)$$

Here,  $G_{\text{front-end}}$  corresponds to the receiver front-end power gain, which is considered fixed, as explained in Section 2.3.2.

### Quantization noise

During the analog-to-digital conversion process, every continuous signal undergoes some amount of distortion and approximations. In the realm of SDRs, this problem can lead to a reduction in the system's ability to detect small signals.

The effects of Analog to Digital Conversion (ADC) resolution on the sampled signal can be modeled by estimating it in terms of noise. This method is known as Quantization Noise, which corresponds to a uniformly distributed noise sequence random variable for ADCs with a resolution of at least 8-bits [24]. Therefore, the quantization error for each channel,  $e_I[n]$  and  $e_Q[n]$ , are sampled from the mentioned distribution and added to obtain the received signal  $z_{rx}[n]$  as follows:

$$z_{rx}[n] = z_{N0}[n] + e_I[n] + je_Q[n] \quad (2.31)$$

$$e_I[n], e_Q[n] \sim \mathcal{U} \left[ \frac{-2^{-(R_{\text{ADC}}-1)}V_{\text{ADC}}}{2}, \frac{2^{-(R_{\text{ADC}}-1)}V_{\text{ADC}}}{2} \right],$$

where  $R_{\text{ADC}}$  corresponds to the number of bits per sample (i.e., the ADC resolution) and  $V_{\text{ADC}}$  is the full-scale value of the ADC measured in volts. This noise model assumes that the successive noise samples,  $e[n]$  and  $e[n+1]$ , are uncorrelated, as well as  $e[n]$  with  $z_{N0}[n]$  [24].

Although it is possible to overcome these effects by adjusting the receiver's amplification gain, doing so can reduce its dynamic range because stronger signals may saturate the subsequent stages at the front end.

It is important to note that in the proposed environment (see Section 1.4), there is no knowledge about the transmitter's location, only an approximation to the actual output power. Therefore, it is advisable to set the receiver's amplifier gain constant around some useful point, while turning off the Automatic Gain Control (AGC) to avoid disturbing the received power measurements, which are necessary to estimate the distance to the transmitter. For this reason, the quantization noise may have a negative impact on distance estimation, particularly for low-power and distant emissions.

At the end of this process, a dataset pair is formed consisting of the target variable  $y_i$  and the feature vector  $(z_{tx,i}[n], z_{rx,i}[n])$ .  $y_i$  represents the distance between the radios, while  $(z_{tx,i}[n], z_{rx,i}[n])$  corresponds to the transmitted and received waveform tuple.

## 2.4 Baseline Estimation Method

The physical baseline model is necessary for two major reasons. First, it serves as a point of comparison to evaluate the performance of the ML methods implemented in this work<sup>4</sup>. Second, it is used to obtain a better *prior* for the VI Bayesian Neural Networks (BNNs) methods.

As mentioned in Chapter 1, there are analytical models available to determine the range to a radio transmitter. However, given the restrictions proposed for this study, only methods that are useful for a single antenna receiver with no control over the transmitter can be considered.

Moreover, only techniques that use RSS are suitable for the simple scenario given. This is because the Time of Arrival (TOA) method requires synchronization with the transmitter, the Time Difference of Arrival (TDOA) technique requires at least two receivers, and the Angle of Arrival (AOA) method requires special antenna arrays or directional arrays with at least two receivers [2].

Methods based on RSS require knowledge of a reference received power  $P_1$ , which is typically defined at one meter from the transmitter, the receiver  $P_{rx}$ , and the path-loss exponent  $\gamma$  of the communication channel [2]. The MLE for the distance between the transmitter and the receiver is [2], [25]:

$$d_{MLE} = \sqrt[\gamma]{\frac{P_1}{P_{rx}}} 1m \quad (2.32)$$

It is worth noting that  $\gamma$  is assumed to be known beforehand, which is also the main source of error for RSS ranging methods [2].

Equations (2.21) and (2.22) can be used to obtain an expression for  $P_1$ , as shown below:

$$P_1 = P_{tx} G_{tx} G_{rx} h_{tx}^2 h_{rx}^2 \frac{1}{1m^4} \quad (2.33)$$

---

<sup>4</sup>The baseline estimation was implemented as part of the analysis script. Available online: <https://gitlab.nps.edu/thesis2/analysis/-/blob/master/src/baseline.py>.

It is clear that prior knowledge about the transmitter's output power, the overall gains from the receiver and transmitter, and their antenna heights will play an important role in how accurate the range estimation of the baseline model is compared to the ground truth.

From Equation (2.22) one can infer that the path-loss exponent will be close to  $\gamma \approx 4$ . This is only an approximation because the multipath effect will produce some variance from this central value depending on the simulated conditions for the dataset.

## 2.5 Machine Learning Methods

Recall from Section 2.1 that the objective is to be able to make range,  $\hat{y}(\mathbf{x})$ , and uncertainty,  $\hat{\sigma}^2(\mathbf{x})$ , predictions using Equations (2.4) and (2.5)<sup>5</sup>. Therefore, in order to use a dataset and ML to find  $p(y|\mathbf{x}, \theta)$ , a relationship between the dataset and the model must be established. Consider, for instance, that the data pairs from the dataset  $\mathcal{D} = \{(y_0, \mathbf{x}_0), \dots, (y_i, \mathbf{x}_i), \dots, (y_n, \mathbf{x}_n)\}$  come from a heteroskedastic parameterized model plus some additive Gaussian noise, such that:

$$y_i = f(\mathbf{x}_i, \theta) + \epsilon \quad \epsilon \sim \mathcal{N}(0, \sigma^2(\mathbf{x}_i, \theta)) \quad (2.34)$$

Heteroskedasticity refers to allowing the variance of the model to change depending on the input (i.e.,  $\sigma^2(\mathbf{x}_i, \theta)$ ) as opposed to homoskedasticity, where the variance is assumed to be constant for every input. Here, a heteroskedastic model is assumed, so that, later, the uncertainty estimates can be calculated, as explained in Section 2.6.2. In this setup, the variance of the noise becomes a learnable function that depends on the model parameters and the input vector.

The PDF  $p(y_i|\mathbf{x}_i, \theta)$ , which is derived from Equation (2.34), is:

$$p(y_i|\mathbf{x}_i, \theta) = \frac{1}{\sqrt{2\pi\sigma^2(\mathbf{x}_i, \theta)}} e^{-\frac{|y_i - f(\mathbf{x}_i, \theta)|^2}{2\sigma^2(\mathbf{x}_i, \theta)}} \quad (2.35)$$

---

<sup>5</sup>ML methods described in this section were implemented as part of the trainer set of scripts. Source code available at: <https://gitlab.nps.edu/thesis2/trainer>.

The use of additive Gaussian noise in Equation (2.34) stems from the assumption that the true distribution  $p(y|\mathbf{x})$  is also Gaussian. This is a commonly employed approach in ML to solve regression problems [26].

Since Equation (2.35) is Gaussian,  $f(\mathbf{x}, \boldsymbol{\theta})$  and  $\sigma^2(\mathbf{x}, \boldsymbol{\theta})$  are sufficient to represent the entire distribution. Thus, a suitable architecture for the NN, denoted as  $\mathcal{M}$ , should generate two outputs instead of the typical single output in a regression scenario. This general concept of NN architecture, incorporating heteroskedastic modeling, is illustrated in Figure 2.8.

The ML methods explained in this section are divided into two categories: deterministic methods, in which  $f(\mathbf{x}, \boldsymbol{\theta})$  and  $\sigma^2(\mathbf{x}, \boldsymbol{\theta})$  always produce the same output for a given input (discussed in Sections 2.5.1 and 2.5.3), and Bayesian methods, in which  $\boldsymbol{\theta}$  is sampled from a probabilistic distribution producing a distribution of possible outputs (explained in Sections 2.5.4 and 2.5.5).

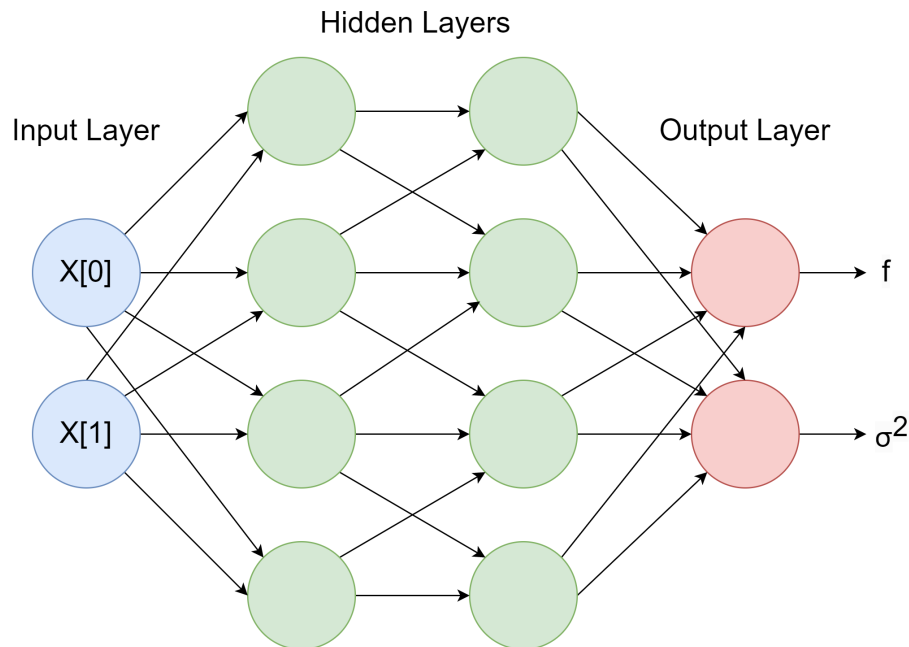


Figure 2.8. Illustration of a heteroskedastic neural network showing the output of the mean and the variance of its prediction.

### 2.5.1 MLE Method

The first technique used to obtain the best fit for a model is by searching the parameter space to find the subset that most likely generated the observed dataset. This process involves finding the closest match for Equation (2.34). The method is known as the MLE and is defined as follows [26]:

$$\theta_{\text{MLE}} = \arg \max_{\theta} p(\mathcal{D}|\theta) \quad (2.36)$$

In MLE, the objective is to find a single point estimate of the posterior  $p(\theta|\mathcal{D})$  as defined by Equation (2.3), assuming that every parameter value is equally likely to be chosen ( $p(\theta) = \text{constant}$ ). Since MLE generates a single point estimate, this means the posterior can then be approximated using a Dirac delta distribution as shown in Equation (2.37):

$$p(\theta|\mathcal{D}) \approx \delta(\theta - \theta_{\text{MLE}}) \quad (2.37)$$

In Equation (2.36),  $p(\mathcal{D}|\theta)$  corresponds to the PDF of the dataset  $\mathcal{D}$  given the parameters  $\theta$ . This assumes that every pair  $(y_i, \mathbf{x}_i)$  is identically independently distributed. For this reason, the objective function of MLE can be expressed as:

$$\arg \max_{\theta} p(\mathcal{D}|\theta) = \arg \max_{\theta} \prod_{i=1}^n p(\mathcal{D}_i|\theta) \quad (2.38)$$

By the properties of the arg max operator, the log base  $e$  function, and the joint probabilities, a more useful relation is found for Equation (2.38). This expression overcomes the numerical stability issues caused by a small likelihood  $p(\mathcal{D}|\theta)$ , which can arise when the dataset contains thousands of examples.

$$\theta_{\text{MLE}} = \arg \min_{\theta} \sum_{i=1}^n -\log p(y_i|\mathbf{x}_i, \theta) \quad (2.39)$$

The outcome of Equation (2.39) is commonly referred to as the Negative Log-Likelihood (NLL). In the context of a Gaussian regression problem, as defined in Equation (2.35), the NLL can be expressed as follows:

$$\theta_{\text{MLE}} = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{2} \log(2\pi\sigma^2(\mathbf{x}_i, \theta)) + \frac{|y_i - f(\mathbf{x}_i, \theta)|^2}{2\sigma^2(\mathbf{x}_i, \theta)} \right) \quad (2.40)$$

The scalar  $\frac{1}{n}$  included in Equation (2.40) does not impact the arg min operator, but is a valuable addition that aids in averaging the objective function. This averaging is particularly beneficial for numerical stability and also enables the use of a constant learning rate that does not change with more examples for the Gradient Descent (GD) techniques employed to find a feasible arg min.

The objective function of Equation (2.40) is commonly known as the Gaussian Negative Log-Likelihood (GNLL). Later, the same equation is used in conjunction with standard optimization approaches based on GD algorithms to find a suitable set of parameters  $\theta_{\text{MLE}}$  to make predictions.

Recall from Equation (2.35) that the assumed conditional PDF for the target variable corresponds to a Gaussian distribution. Therefore, to make a range prediction using the posterior distribution of the MLE given by Equation (2.37), the expression for  $\hat{y}(\mathbf{x})$  in Equation (2.4) is reduced to:

$$\hat{y}(\mathbf{x}) = f(\mathbf{x}, \theta_{\text{MLE}}) \quad (2.41)$$

Similarly, a prediction for the total uncertainty can be obtained using Equation (2.5), which corresponds to:

$$\hat{\sigma}^2(\mathbf{x}) = \sigma^2(\mathbf{x}, \theta_{\text{MLE}}) \quad (2.42)$$

## 2.5.2 Training and Optimization Methods

Training is an iterative process of searching the parameter space to find the subset that produces the minimum value for the objective functions defined in Sections 2.5.1, 2.5.3, 2.5.4, and 2.5.5. For this reason, the objective functions are typically known as the *Loss Function*  $L(\theta)$ . When  $L(\theta)$  is differentiable, as is the case for all the methods presented, minimization can be achieved using GD [26].

In general, the loss function over the parameters can be expressed in terms of the data term and the regularization term. As shown in this work, the data term corresponds to the sum of the NLL of individual dataset pairs  $L_i(\theta)$  over the entire dataset, while the regularization term  $R(\theta)$  specifically depends on the method used.

$$L(\theta) = \sum_{n=1}^N L_n(\theta) + R(\theta) \quad (2.43)$$

However, pure GD methods are computationally inefficient, and for this reason, a noisy approximation of the gradient is calculated using only a few pairs of data points. When the number of data points  $N$  is lower than the total number of pairs in the dataset, it is referred to as mini-batch Stochastic Gradient Descent (SGD).

The larger the mini-batch size, the better the estimate and the smaller the variance of the parameter's gradient, while the smaller the mini-batch, the fewer computational resources are needed to update the parameters, and the noisier the gradient approximation will be [26]. Importantly, the noise can help to escape from local minima, allowing the method to arrive at better models [26]. Although it is noisy, it is highly likely to converge to a local minimum [27].

Equation (2.44) provides a recursive method for updating the parameters  $\theta$  using mini-batch SGD [26, adapted from p. 232]. Here, the subscript  $i$  indicates the current state of the parameters.

$$\theta_{i+1} = \theta_i - \gamma_i \nabla \left( \sum_{n=1}^N L_n(\theta_i) + R(\theta_i) \right) \quad (2.44)$$

The hyperparameter  $\gamma_i$  is known as the *learning rate*, which is arguably the most important hyperparameter during the training process [28]. The symbol  $\nabla$  indicates the process of taking partial derivatives with respect to all the parameters of the network. The complete process is typically executed using auto-differentiation libraries (e.g., PyTorch, TensorFlow) based on the *backpropagation algorithm* [28].

A popular improved version of SGD with adaptive moment estimation, known as *Adam*, was introduced by Kingma et al. [29]. Advantages of *Adam* include invariance over gradient re-scaling and automatic adjustments of the learning rate  $\gamma$ . Furthermore, *Adam* has been empirically shown to be computationally efficient.

### **Initialization of Parameters**

The methods explained in Sections 2.5.1, 2.5.3, 2.5.4, and 2.5.5 are expressions for defining the loss functions suitable for solving the general problem statement of Section 2.1. Thus, they do not require an assumption about the initial state of the parameters. This poses a problem because an initial state is needed to apply GD methods, as reflected in the recursive Equation (2.44). Moreover, the initial state will determine the number of iterations needed to reach a local minimum.

In this work, the weight initialization of all the tested architectures was executed using the procedure proposed by He et al. [30]. The initialization greatly depends on the type of activation function used. For layers using Rectified Linear Activation Unit (ReLU) as the activation function, every internal weight was initialized by sampling from a normal distribution  $\mathcal{N}(0, 2/\text{fan-in})$ . Here, fan-in corresponds to the number of inputs of that particular neuron. For output layers, typically implemented using an unbounded linear layer in a regression problem, the initialization involves sampling from a normal distribution  $\mathcal{N}(0, 2/(\text{fan-in} + \text{fan-out}))$ , where fan-out corresponds to the number of outputs of the neuron.

From the previous paragraph, it should be noted that the weights are close to zero, resulting in predictions from the NN that are also close to zero by default. This behavior is reasonable for the initial state, considering that the input vector  $\mathbf{x}$  and target variable  $y$  are typically z-scored. Z-scoring the input vector and target variable is necessary to assist the optimizer in accelerating the training process and achieving convergence to a solution [30].

### 2.5.3 MAP Method

The second method used for obtaining a point estimate of the parameters is the MAP estimation, which is defined as [26]:

$$\boldsymbol{\theta}_{\text{MAP}} = \arg \max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathcal{D}) \quad (2.45)$$

This method aims to obtain directly the best parameters given the data. However, the true distribution over the parameters is unknown and therefore cannot be directly computed. The solution is to use Bayes' theorem (Equation (2.3)) to find an alternative expression through which an approximation to the true *posterior* can be computed.

From Equation (2.3), one can observe that MAP and MLE are related through Bayes' theorem. By employing similar reasoning as in Section 2.5.1, Equation (2.45) can be simplified to:

$$\boldsymbol{\theta}_{\text{MAP}} = \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^n -\log p(y_i|\mathbf{x}_i, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) \quad (2.46)$$

Equation (2.46) demonstrates that the objective function for  $\boldsymbol{\theta}_{\text{MAP}}$  is composed of the MLE objective function along with the negative log of the prior  $p(\boldsymbol{\theta})$ . The approximation arises from assumptions made regarding the MLE distribution and the fact that the optimal prior may not be known, but generally applicable priors can be used as substitutes.

The investigation of NN priors is an active research area. In fact, in Section 2.5.5, a method for obtaining a better prior is developed.

A classical approach is to consider the prior as a zero-mean Gaussian distribution, denoted as  $p(\boldsymbol{\theta}) = \mathcal{N}(0, \sigma_{\text{prior}}^2)$ . By setting  $\lambda = \frac{1}{2\sigma_{\text{prior}}^2}$ , which is referred to as the regularization strength, and utilizing the results from the MLE Equation (2.40), Equation (2.46) is equivalent to:

$$\boldsymbol{\theta}_{\text{MAP}} = \arg \min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{2} \log(2\pi\sigma^2(\mathbf{x}_i, \boldsymbol{\theta})) + \frac{|y_i - f(\mathbf{x}_i, \boldsymbol{\theta})|^2}{2\sigma^2(\mathbf{x}_i, \boldsymbol{\theta})} \right) + \lambda \sum_{i=1}^d |\theta_i|^2 \quad (2.47)$$

The regularization term that arises from a zero-mean Gaussian prior is commonly referred to as  $L_2$  regularization, the strength of which can be tuned by adjusting the hyperparameter  $\lambda$ , often using validation data. Similar to MLE, the objective function defined in Equation (2.47) is differentiable and can be utilized to find the MAP using the same GD optimization techniques. Moreover, the MAP represents the most likely single point estimate of the posterior, implying that the same Equations (2.41) and (2.42) can be applied to make predictions for the range and variance, respectively, but with the substitution of MAP.

### 2.5.4 VI with a Gaussian Variational Distribution

BNNs are interesting because they provide improved quality uncertainty estimates through the inclusion of epistemic uncertainty estimation for the model. Epistemic uncertainty corresponds to the uncertainty that results from the imperfections of the model and can be reduced by adding more examples to the training data [31]. In this work, two models were implemented to obtain epistemic uncertainties: VI using a Gaussian variational distribution and VI utilizing the physical baseline model discussed in Section 2.5.5 as a prior.

The primary concept behind Bayesian methods is to move away from obtaining a single point estimate of the posterior  $p(\theta|\mathcal{D})$ , as done in the MAP method discussed in Section 2.5.3. Instead, the objective is to obtain an approximate distribution for the entire parameter space  $p(\theta|\mathcal{D})$  [32]. Subsequently, this approximation can be utilized to estimate the epistemic uncertainty of the model, as explained in Section 2.6.2.

In VI, the approximate distribution over the parameter space is called the variational distribution  $q_\phi(\theta)$ , which is fitted to be as close as possible to the posterior distribution  $p(\theta|\mathcal{D})$ . This method requires the definition of a parameterized distribution, which, in this case, will be a Gaussian—with diagonal covariance—due to its mathematical properties. However, this also means that the variational distribution will only be a rough approximation to the posterior. Nonetheless, it will still provide better uncertainty estimates than other methods such as the Laplace approximation method [33] or MC Dropout [34].

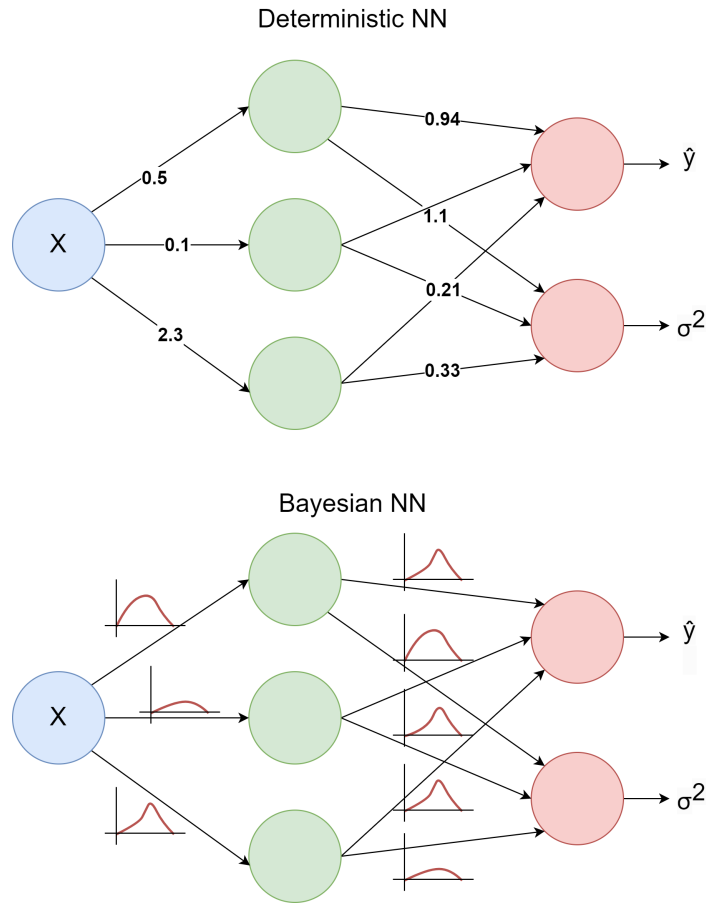


Figure 2.9. Illustration of a deterministic NN vs. BNN. Adapted from [35, figure 1].

Recall that for a NN model, the parameters  $\theta$  correspond to the weights of the neurons. However, in a VI setup, those weights are no longer a set using a point estimate, but rather a set using a probability distribution. Figure 2.9 depicts the difference between a deterministic NN and a BNN. In the case of a Gaussian distribution with diagonal covariance, the weights can be fully defined by their mean and standard deviation, which become the new parameter space for the BNN. Therefore, it is appropriate to define  $\phi$  as the set  $\{(\mu_1, \sigma_1), \dots, (\mu_i, \sigma_i), \dots, (\mu_k, \sigma_k)\}$ , where  $k$  corresponds to the total number of parameters in the BNN, and each tuple represents the parameters of the distribution for the  $i^{th}$  weight of the network. It is important to note that for the same architecture, this type of BNN will have twice the number of parameters compared to its deterministic counterpart.

This new setup raises the question: how close is the variational distribution to the posterior? Fortunately, there are methods to answer this question. The most common method for measuring this closeness comes from Information Theory and is called the Kullback-Leibler Divergence (KL) [32], [36]. The KL between the variational distribution and the posterior is defined as:

$$\text{KL} [q_\phi(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D})] = \mathbb{E}_{q_\phi(\boldsymbol{\theta})} \left[ \log \frac{q_\phi(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathcal{D})} \right] \quad (2.48)$$

The lower the value of Equation (2.48), the more similar  $q_\phi(\boldsymbol{\theta})$  and  $p(\boldsymbol{\theta}|\mathcal{D})$ . Consequently, one can apply the *arg min* operator to arrive at an expression that can serve as an objective function suitable for applying the same optimization methods used for MAP and MLE.

$$\begin{aligned} \arg \min_{\boldsymbol{\theta}} \text{KL} [q_\phi(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D})] = \\ \arg \min_{\boldsymbol{\theta}} \underbrace{\frac{1}{n} \int_{-\infty}^{\infty} q_\phi(\boldsymbol{\theta}) \sum_{i=1}^n -\log p(\mathcal{D}_i|\boldsymbol{\theta}) d\boldsymbol{\theta}}_{\text{data term}} + \underbrace{\frac{1}{n} \text{KL} [q_\phi(\boldsymbol{\theta})||p(\boldsymbol{\theta})]}_{\text{regularization term}} \end{aligned} \quad (2.49)$$

The integral of the NLL is referred to as the *data term*, while the KL is called the *regularization term*. Typically, the integral for the data term is intractable, but an approximation can be obtained using MC sampling [32], [37]:

$$\frac{1}{n} \int_{-\infty}^{\infty} q_\phi(\boldsymbol{\theta}) \sum_{i=1}^n -\log p(\mathcal{D}_i|\boldsymbol{\theta}) d\boldsymbol{\theta} \approx \frac{1}{n} \frac{1}{m} \sum_{i=1}^n \sum_{j=1}^m -\log p(\mathcal{D}_i|\boldsymbol{\theta}^j) \quad \boldsymbol{\theta}^j \sim q_\phi(\boldsymbol{\theta}) \quad (2.50)$$

Since the data term is the NLL when the parameters are sampled from the variational distribution, it is possible to compute this term using the same procedure as explained in Section 2.5.1 for MLE.

On the other hand, the regularization term corresponds to the KL between two Gaussians, since, in the used Gaussian VI methods, the variational distribution  $q_\phi(\boldsymbol{\theta})$  and the prior  $p(\boldsymbol{\theta})$ , are both Gaussian with diagonal covariance. The KL between two Gaussians is a well-known result [36]:

$$\text{KL} [q_\phi(\boldsymbol{\theta})||p(\boldsymbol{\theta})] = \sum_{i=1}^K \left[ \log \frac{\sigma_{p_i}}{\sigma_{q_i}} + \frac{\sigma_{q_i}^2 + (\mu_{q_i} - \mu_{p_i})^2}{2\sigma_{p_i}^2} - \frac{1}{2} \right], \quad (2.51)$$

where  $\sigma_{p_i}$  and  $\mu_{p_i}$  corresponds to the standard deviation and mean of the  $i$ -th prior parameter, respectively. Similarly,  $\sigma_{q_i}$  and  $\mu_{q_i}$  corresponds the standard deviation and mean of the  $i$ -th variational distribution parameter.

The resulting objective function for Equation (2.49) is differentiable [32], [38]. This implies that the same optimization techniques based on GD can be utilized to train the model.

Similarly, as explained for the MLE case in Section 2.5.1, a reasonable prediction using the VI method corresponds to the expected distance between the transmitter and the receiver, modeled by the distribution of Equation (2.35). In this case,  $\boldsymbol{\theta}$  is not a vector of fixed values but rather a vector of random variables from the distribution  $q_\phi(\boldsymbol{\theta})$ . Therefore, the overall distribution  $p(y|\mathbf{x}, \boldsymbol{\theta})$  becomes a *mixture distribution*, per Bayesian model averaging. Now, to make a range prediction, Equation (2.4) can be used again:

$$\hat{y}(\mathbf{x}) = \frac{1}{m} \sum_{j=1}^m f(\mathbf{x}, \boldsymbol{\theta}^j) \quad \boldsymbol{\theta}^j \sim q_\phi(\boldsymbol{\theta}) \quad (2.52)$$

Likewise, a prediction for the total uncertainty can be obtained by taking the variance of the mixture distribution using Equation (2.5):

$$\hat{\sigma}^2(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \left( \sigma^2(\mathbf{x}, \boldsymbol{\theta}^i) + (f(\mathbf{x}, \boldsymbol{\theta}^i))^2 \right) - \hat{y}(\mathbf{x})^2 \quad \boldsymbol{\theta}^i \sim q_\phi(\boldsymbol{\theta}) \quad (2.53)$$

Figure 2.10 depicts the operation of making predictions using BNNs and MC-Sampling.

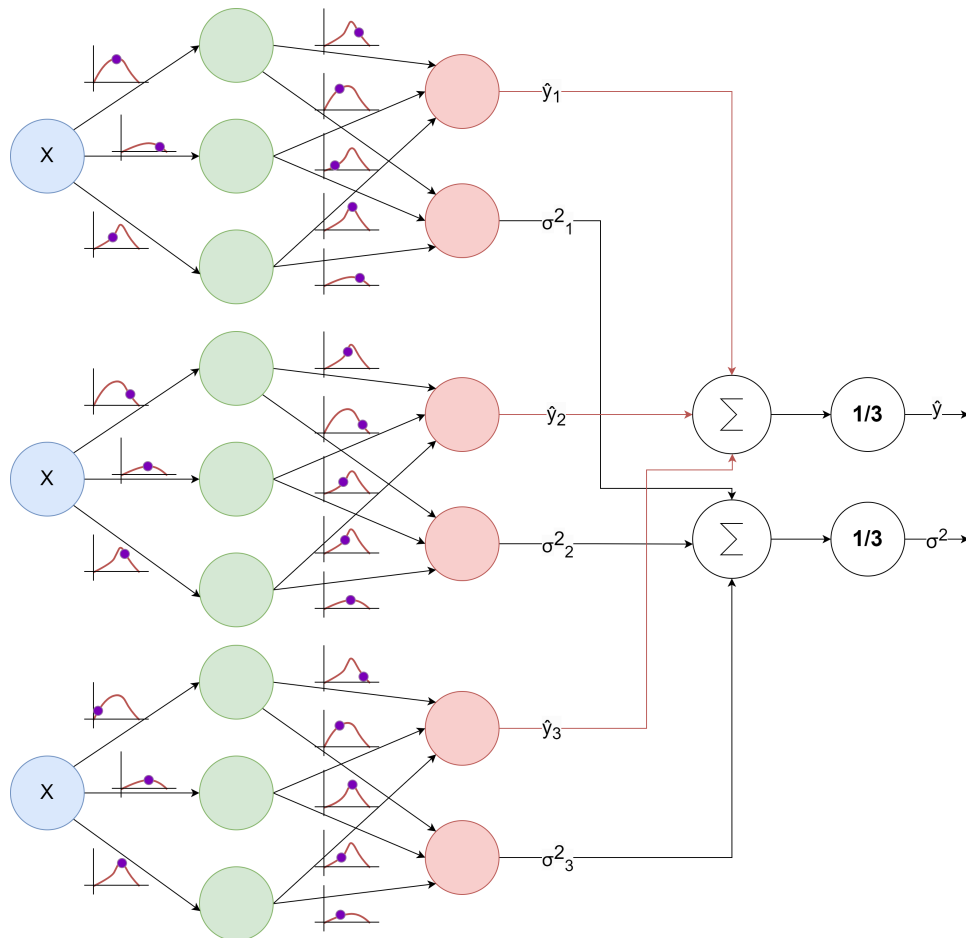


Figure 2.10. Illustration of making predictions using BNN and MC-sampling using 3 samples. Adapted from [35, figure 1].

### 2.5.5 VI with a Physical Model Prior

In Section 2.5.4, the assumption was made that the prior  $p(\theta)$  is a zero-mean Gaussian distribution. This assumed prior is reasonable because the weights of the NN are often initialized using a zero-mean Gaussian distribution [30] (refer to Section 2.5.2).

The strength of the regularization term in Equation (2.49) greatly depends on the selected prior  $p(\theta)$ . Therefore, a poor choice of prior can amplify the regularization effect and excessively restrict the model's flexibility or weaken its regularization capabilities, leading to overfitting.

An alternative approach that results in the utilization of a superior prior and initial state for the NN weights is to employ the physical baseline model described in Section 2.4 as the starting model and then retrain it using the observed dataset  $\mathcal{D}$ . To mitigate the risks of *catastrophic forgetting* [39], it is advisable to employ a Bayesian Continual Learning Framework [40].

To employ the same ML techniques, one must initially train the model using a dataset in which the target variables are calculated based on the physical model predictions outlined in Section 2.4, denoted as  $\mathcal{D}_{\text{physical}} = \{..., (y_{i,\text{physical}}, \mathbf{x}_i), ...\}$ . Consequently, the resulting posterior, obtained using Bayes' Theorem, is expressed as:

$$p(\boldsymbol{\theta}|\mathcal{D}_{\text{physical}}, \mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D}_{\text{physical}})}{p(\mathcal{D})} \quad (2.54)$$

In this equation,  $p(\boldsymbol{\theta}|\mathcal{D}_{\text{physical}})$  represents the posterior of the model when utilizing the physical dataset. It is noteworthy that this posterior contains all the required information for the physical model, including the weights' relative significance for making predictions [39]. Additionally, the distribution serves as the prior for the posterior distribution defined in Equation (2.54).

Applying the same methods for VI, as shown in Section 2.5.4 and Equation (2.54), the objective function, which is the KL between the objective variational distribution  $q_{\phi}(\boldsymbol{\theta})$  and the overall posterior, can now be expressed as:

$$\text{KL} [q_{\phi}(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D}_{\text{physical}}, \mathcal{D})] = \frac{1}{n} \int_{-\infty}^{\infty} q_{\phi}(\boldsymbol{\theta}) \sum_{i=1}^n -\log p(\mathcal{D}_i|\boldsymbol{\theta}) d\boldsymbol{\theta} + \frac{1}{n} \text{KL} [q_{\phi}(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D}_{\text{physical}})] \quad (2.55)$$

Once again, VI can be utilized with a standard Gaussian prior to obtain an approximate distribution for the physical model posterior, denoted as  $q_{\text{physical}}(\boldsymbol{\theta}) \approx p(\boldsymbol{\theta}|\mathcal{D}_{\text{physical}})$ . This resulting distribution  $q_{\text{physical}}(\boldsymbol{\theta})$  is also Gaussian, allowing the *regularization term* to be obtained using Equation (2.51), where the KL is:

$$\text{KL} [q_\phi(\boldsymbol{\theta}) || q_{\text{physical}}(\boldsymbol{\theta})] = \sum_{i=1}^K \left[ \log \frac{\sigma_{\text{physical}_i}}{\sigma_{q_i}} + \frac{\sigma_{q_i}^2 + (\mu_{q_i} - \mu_{\text{physical}_i})^2}{2\sigma_{\text{physical}_i}^2} - \frac{1}{2} \right] \quad (2.56)$$

The regularization is based on the physical model, meaning that the resulting distribution  $q_\phi(\boldsymbol{\theta})$  will force the BNN to make predictions close to the answers of the physical model by default, which is a desirable behavior. This condition will be particularly true when the number of training examples is insufficient or they are not well distributed over the entire range of values to pull the weights to reduce the data term instead of the regularization term.

As usual, the same optimization methods for GD hold to train the BNN using the physical model prior. But the training process will consist of two steps as defined by the Continual Learning framework. First, the model will be trained using the physical dataset  $\mathcal{D}_{\text{physical}}$  with a Gaussian prior  $\mathcal{N}(0, \sigma_{\text{prior}}^2 I)$ . Second, the resulting model and its posterior  $q_{\text{physical}}(\boldsymbol{\theta})$  are trained again, but this time using the regular dataset  $\mathcal{D}$  and the variational distribution  $q_{\text{physical}}(\boldsymbol{\theta})$  obtained from the first step as the prior. The overall concept is depicted in Figure 2.11. Finally, to make predictions, the same MC sampling method as described in Equations (2.52) and (2.53) can be used to obtain the predicted range  $\hat{y}(\mathbf{x})$  and variance  $\hat{\sigma}^2(\mathbf{x})$ , respectively.

## 2.5.6 Implementation Details for Modeling Distributions using NN

Although an in-depth discussion of architectural implementation details is unnecessary since this work focuses on general methods, a couple of noteworthy details regarding the modeling of distributions using NN that are applicable to most architectures are presented in this section.

### Heteroskedastic modeling using Neural Networks

As shown in Equation (2.35) and depicted in Figure 2.8, the NN effectively acts as a probabilistic model, specifically a heteroskedastic Gaussian distribution. This distribution is characterized by the predicted mean and variance of the NN, calculated respectively using Equations (2.41) and (2.42) for a deterministic NN, and Equations (2.52) and (2.53) for a BNN. Consequently, networks should produce two outputs.

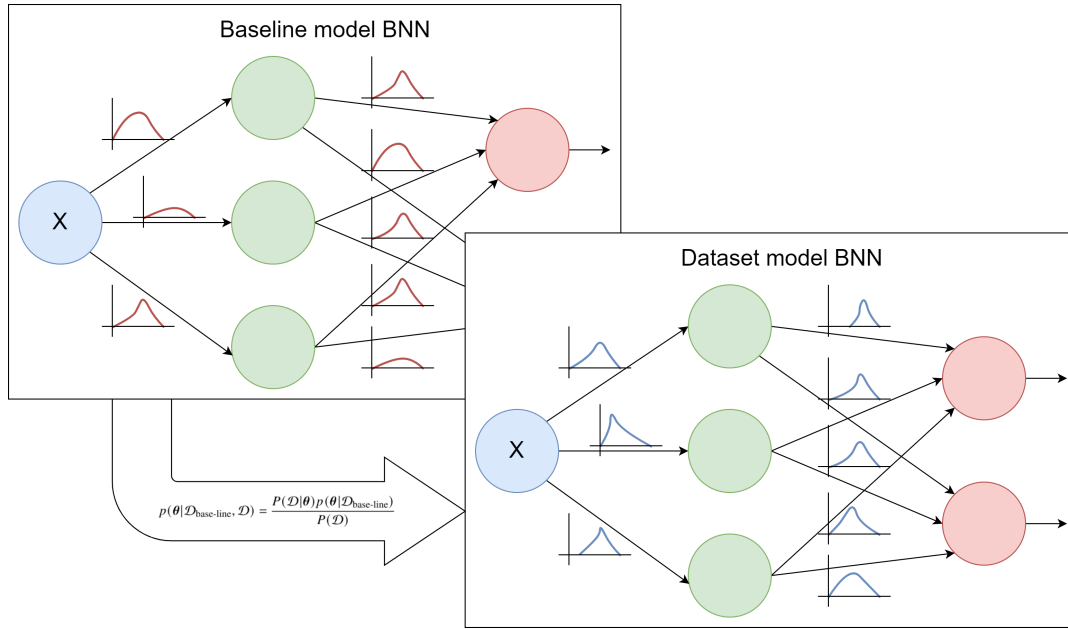


Figure 2.11. Illustration of VI with physical model as prior using the Continuous Learning framework.

The neuron  $\rho(\mathbf{x}', \theta_\rho)$ , used for predicting the variance in the heteroskedastic model of Equation (2.35) and depicted in Figure 2.8, is special due to the requirement that the variance must always be positive. As a result, the activation function  $\Phi$  must produce non-negative values, while allowing  $\rho$  to vary freely between negative and positive numbers. In other words,  $\Phi : \mathbb{R} \rightarrow \mathbb{R}_0^+$ . Furthermore, the activation function should be differentiable to facilitate the use of GD optimization techniques.

The activation function  $\Phi$  should also be a monotonically increasing function to ensure that a small value maps to a small variance and a high value maps to a high variance. This behavior cannot be achieved simply by taking the square or absolute value of  $\rho$ . Equation (2.57) proposes the use of the *Softplus* activation function, which is also utilized to model the variance of weights when employing BNN [32], as implemented by Krishnan in their BNN library [41].

$$\Phi(\rho) = \ln(1 + e^\rho) \quad (2.57)$$

This activation function ensures that the predicted variance is always positive, while reducing the risk of exploding gradients by employing the natural exponent function, and preventing numerical instability by guaranteeing that the argument of the logarithm is always equal to or greater than one. A reasonable initialization for  $\theta \in \rho$  should produce a variance much smaller than the available range of target variables  $y$  in the dataset. Note that the variance of the target variable is one because  $y$  is z-scored.

### **Sampling from the distribution over weights in a BNN neuron**

A popular implementation for sampling the parameter distributions in BNNs is the Local Reparameterization Trick (LRT) shown in Equation (2.58) [38]. The LRT consists of using the properties of Gaussian distributions to move the sampling process from directly sampling the weights from  $p(\theta|\mathcal{D})$  to sampling noise from a standard normal distribution and applying the sampled noise after the summation of the linear operation for each neuron. This method greatly improves the speed of sampling during training and inference using BNNs.

$$\theta_i = \mu_i + \sigma_i \epsilon, \quad \epsilon \sim \mathcal{N}(0, 1) \quad (2.58)$$

## **2.6 Evaluation Methods**

The methods listed here correspond to the metrics used to measure the level of accomplishment of the objectives defined in Section 1.3 for this thesis<sup>6</sup>. The methods are divided into two main categories: performance metrics and uncertainty metrics.

### **2.6.1 Performance Metrics**

Performance metrics are used to measure the overall performance of the NN methods in comparison with the classical physical baseline model, as shown in Section 2.4, and among the NN methods used.

---

<sup>6</sup>Evaluation methods were implemented as part of the analysis script. The source code is available at: <https://gitlab.nps.edu/thesis2/analysis>.

### Negative Log-Likelihood

The NLL corresponds to a relative measurement that is calculated over the *test* dataset using Equation (2.59). This measure is useful for comparing the performance of different NN architectures and methods in learning the dataset distribution. Moreover, the NLL evaluates the quality of the variance estimates, which cannot be accounted for using the other metrics detailed in this section. Additionally, it ensures repeatability of results.

$$\text{NLL}(\mathcal{D}_{\text{test}}, \hat{y}, \hat{\sigma}^2) = \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{2} \log(2\pi\hat{\sigma}^2(\mathbf{x}_i)) + \frac{|y_i - \hat{y}(\mathbf{x}_i)|^2}{2\hat{\sigma}^2(\mathbf{x}_i)} \right) \quad (2.59)$$

This equation shows the NLL for a Gaussian distribution as it was derived for the MLE method in Section 2.5.1, and uses the range estimate  $\hat{y}(\mathbf{x})$  and variance estimate  $\hat{\sigma}^2(\mathbf{x})$  from Equations (2.4) and (2.5) respectively, under the corresponding NN method, where  $(y_i, \mathbf{x}_i)$  corresponds to a dataset pair from the testing dataset  $\mathcal{D}_{\text{test}}$ .

### Mean Squared Error

The Mean Squared Error (MSE) is a classical measurement used in regression problems and is also used as an equivalent loss function to the GNLL for the MLE method when homoskedastic modeling with no trainable variance is used. Therefore, it is suitable for comparing the architectures of this work with other ML methods. Moreover, this metric is useful for comparing the performance of the NN models with respect to the physical baseline model. It is calculated as:

$$\text{MSE}(\mathcal{D}_{\text{test}}, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (\hat{y}(\mathbf{x}_i) - y_i)^2, \quad (2.60)$$

where  $\hat{y}(\mathbf{x}_i)$  is the predicted value obtained using Equation (2.4) that depends on the NN method used, and  $(y_i, \mathbf{x}_i)$  corresponds to a dataset pair from the testing dataset  $\mathcal{D}_{\text{test}}$ . It is worth mentioning that this metric overestimates the error and exacerbates the influence of outliers in the dataset.

### Mean Absolute Error

In particular, for this work, the Mean Absolute Error (MAE) serves as the absolute reference for performance because it better handles outliers, does not exacerbate errors, and its value can be directly interpreted as it contains the same units as the target variable. Similarly to the MSE metric, it is calculated using Equation (2.4) under the corresponding NN method and the dataset pairs  $(y_i, \mathbf{x}_i)$  from the testing dataset  $\mathcal{D}_{\text{test}}$ .

$$\text{MAE}(\mathcal{D}_{\text{test}}, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |\hat{y}(\mathbf{x}_i) - y_i| \quad (2.61)$$

### 2.6.2 Uncertainty Metrics

The NN methods presented in this work are limited probabilistic approximations to an underlying ground truth, achieved by using a noisy sample of reality. This raises the important question of how we can trust our models and measure the unknowns regarding the limitations of the models. Uncertainty measurements in this context are particularly crucial for the issues described in Chapter 1, which include safety-critical systems and decision-making scenarios.

Although there are various types of uncertainties, this work focuses particularly on the uncertainties that result from the imperfections of the model, known as *epistemic* uncertainty, as well as the intrinsic randomness that arises from the noisy environment from which samples of reality are taken, known as *aleatoric* uncertainty [31].

According to the work of Depeweg et al. [42], the total uncertainty predicted by a heteroskedastic model can be obtained from the Law of Total Variance, as follows:

$$\text{Var}_{p(y|\mathbf{x}, \mathcal{D})}(y) = \text{Var}_{p(\theta|\mathcal{D})}(\mathbb{E}_{p(y|\mathbf{x}, \theta)}[y]) + \mathbb{E}_{p(\theta|\mathcal{D})}[\text{Var}_{p(y|\mathbf{x}, \theta)}(y)] \quad (2.62)$$

From the ML methods discussed in Section 2.5,  $p(y|\mathbf{x}, \boldsymbol{\theta})$  is defined as a Gaussian distribution characterized by its mean  $f(\mathbf{x}, \boldsymbol{\theta})$  and variance  $\sigma^2(\mathbf{x}, \boldsymbol{\theta})$ . Therefore,

$$\text{Var}_{p(y|\mathbf{x}, \mathcal{D})}(y) = \underbrace{\text{Var}_{p(\boldsymbol{\theta}|\mathcal{D})}(f(\mathbf{x}, \boldsymbol{\theta}))}_{\text{epistemic uncertainty}} + \underbrace{\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\sigma^2(\mathbf{x}, \boldsymbol{\theta})]}_{\text{aleatoric uncertainty}} \quad (2.63)$$

In this context,  $\text{Var}_{p(\boldsymbol{\theta}|\mathcal{D})}(f(\mathbf{x}, \boldsymbol{\theta}))$  corresponds to a measure of epistemic uncertainty, since  $f(\mathbf{x}, \boldsymbol{\theta})$  is independent of the noise from the data and only depends on the parameters, as shown in Equation (2.34). The epistemic uncertainty is reduced by adding more training data, converging to zero in the limit of infinite data. On the other hand,  $\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\sigma^2(\mathbf{x}, \boldsymbol{\theta})]$  represents aleatoric uncertainty, as it models the noise in the data according to Equation (2.34). Furthermore, this expression represents a weighted average over all possible parameters, indicating that the effects of the parameters are effectively filtered out, leaving only the effects of the dataset. For this reason, adding more data does not reduce the aleatoric uncertainty [42].

The expression in Equation (2.63) is typically intractable, but it is possible to approximate it using MC-sampling, as done for the predicted total uncertainty in Equation (2.5):

$$\hat{\sigma}^2(\mathbf{x}) = \underbrace{\frac{1}{m} \sum_{i=1}^m \left( f(\mathbf{x}, \boldsymbol{\theta}^i) - \left( \frac{1}{m} \sum_{i=1}^m f(\mathbf{x}, \boldsymbol{\theta}^i) \right) \right)^2}_{\text{epistemic uncertainty}} + \underbrace{\frac{1}{m} \sum_{i=1}^m \sigma^2(\mathbf{x}, \boldsymbol{\theta}^i)}_{\text{aleatoric uncertainty}}, \quad \boldsymbol{\theta}^i \sim p(\boldsymbol{\theta}|\mathcal{D}) \quad (2.64)$$

THIS PAGE INTENTIONALLY LEFT BLANK

---

## CHAPTER 3:

# Experimental Results

---

### 3.1 Organization

The purpose of this section is to explain the performed experiments in order to provide evidence to support the claims of this thesis. All the experiments were performed using the methods presented in Chapter 2.

For this work, three different datasets were generated using the DGF<sup>7</sup>:

- **Ideal:** The purpose of this environment is to present a perfect scenario for evaluating the physical-model-based baseline estimation method presented in Section 2.4, as well as verifying the ability of the NN architecture to learn using the ML methods described in Section 2.5.
- **Standard:** The purpose of this dataset is to present a more complex environment that is more likely to be observed in a maritime scenario. The standard environment serves as the primary dataset for comparing the performance between the different ML methods against the baseline.
- **Harsh:** The purpose of this dataset is to present a challenging environment in order to further test the performance of the methods. Additionally, it is used to assess the capability of uncertainty metrics in detecting a domain shift when the BNNs are trained using the standard environment.

The primary parameter for simulating the environments using the DGF is the distance because it establishes the range of valid predictions for the NN. As mentioned in Section 1.4, the range of distances are determined from the beginning of the far-field approximation to the radio horizon.

In the case of monopole antennas, where the antenna size is smaller than the wavelength  $\lambda$ , the far-field approximation boundary is typically considered to be at  $5\lambda$  [43]. For this study, the limiting range was determined by the largest wavelength, which for VHF is

---

<sup>7</sup>The DGF dataset generation scripts can be found at: <https://gitlab.nps.edu/thesis2/dataset/scripts>.

approximately 10 m at 30 MHz. Therefore, the far-field approximation is set at 50 m. This value is comparable to the antenna heights found in maritime vessels, which conflicts with the distance requirement of the propagation path loss model as defined in Section 2.3. To resolve this conflict, the far-field approximation was increased by a factor of 10, resulting in a value of 500 m.

The radio horizon for LOS RF signals is approximately calculated using the antenna height in meters of the transmitter and receiver ( $h_{tx}$  and  $h_{rx}$ , respectively), the Earth's radius, and a commonly assumed refraction factor of 1.33. The refraction factor is defined as the ratio of the Earth bulge height, measured at half the distance between the transmitter and receiver, to the effective bulge due to the bending of the electromagnetic wave in the atmosphere. The Earth bulge model represents the effects of the Earth's curvature in a flat surface. This calculation is described by Equation (3.1) [44], [45].

$$y_{\max} = \left( 4.124\sqrt{h_1} + 4.124\sqrt{h_2} \right) 1\sqrt{m} \quad (3.1)$$

With an antenna height of 10 m for both the transmitter and receiver, the limit of the radio horizon is approximately 25 km.

Taking into consideration that one of the research questions of this thesis requires examining the behavior of the NN models in environments for which they were not trained, the models were trained using a reduced range of distances: 2-20 km. This approach allows the entire range (i.e., 0.5–25 km) to be used for testing the capabilities of the NN under *unseen* range conditions and to evaluate the detection capabilities of the epistemic uncertainty metric. The choice of a reduced training range is asymmetrical because more examples were needed at longer distances to properly capture the effects.

A summary of the environmental characteristics is presented in Table 3.1. For further information regarding the parameters utilized to generate the corresponding datasets, please refer to Appendix B.

In order to examine the model's performance under the same conditions, the analysis is divided into three different sections, one per environment. Deterministic NN and BNN models were trained for each environment; for more information refer to Appendix A. The

obtained models were tested according to the defined performance and uncertainty metrics of Section 2.6.1 and Section 2.6.2, respectively. The analysis also includes testing every model using unseen data to observe how the epistemic uncertainty is capable of identifying new inputs. Finally, every model was tested using a different environment to check how the uncertainty metrics performed in the presence of a domain shift.

Table 3.1. Environments characteristic summary.

| Environment | Beaufort | ADC     | Tx PWR [W] | Max Scatterers | Rx NF [dB] |
|-------------|----------|---------|------------|----------------|------------|
| Ideal       | 0        | 32-bits | 50         | 0              | 0          |
| Standard    | 4        | 12-bits | 10–25      | 4              | 0.9        |
| Harsh       | 7        | 12-bits | 10–25–50   | 40             | 2.0        |

## 3.2 Ideal Environment Results

As previously mentioned, the ideal environment serves as a verification dataset for the physical model baseline. Furthermore, it validates the correct implementation of the NN architectures and training procedures. In this environment, one should expect the MAE for the baseline to be close to zero and show the best attainable MAE for the NN methods.

Figure 3.1 displays a representative dataset pair  $(y_i, (z_{tx,i}[n], z_{rx,i}[n]))$  from the ideal environment. Here, the baseband time domain and PSD representation of the transmitted and received signals are presented for a target variable of 10 km. As anticipated, under these conditions, the received signal is nearly identical to the transmitted signal, with no noticeable distortion. The only discernible impact is the power loss attributed to the path-loss model.

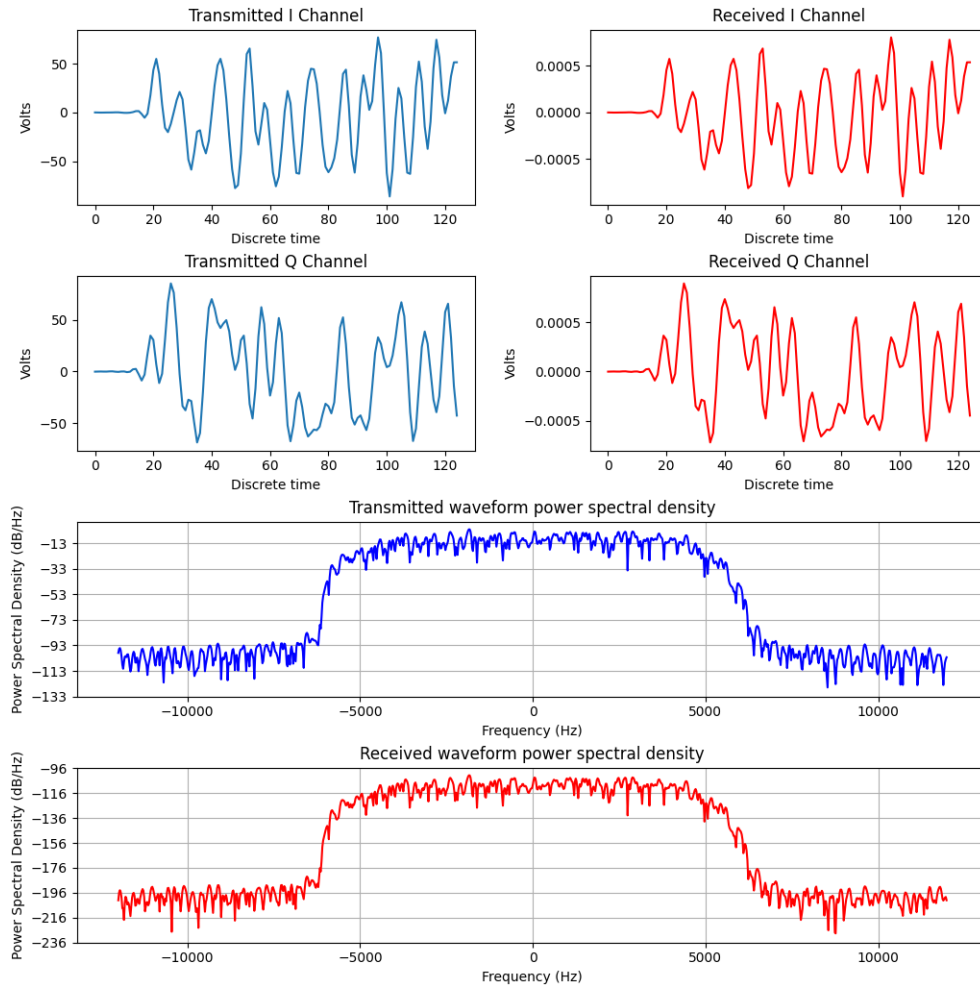


Figure 3.1. Ideal environment representative example showing a transmitted signal with its corresponding received signal at 10 km.

### 3.2.1 Performance Results

Table 3.2 presents the performance results of the physical model and the developed NN models in the ideal environment. The following explanations provide context for the different methods utilized:

- *Physical* refers to the application of the physical method, used as baseline metric as elucidated in Section 2.4.
- *MAP-S* indicates the utilization of the MAP method from Section 2.5.3, using the

- simulated RF source distance,  $y_i$ , as the target variable.
- *MAP-P* indicates the employment of the MAP method from Section 2.5.3, employing the RF source distance,  $y_{i,\text{physical}}$ , calculated using the physical model from Section 2.4 as the target variable.
  - *VI-S* indicates the utilization of the VI method from Section 2.5.4, using the simulated RF source distance,  $y_i$ , as the target variable.
  - *VI-P* indicates the employment of the VI method from Section 2.5.4, employing the RF source distance,  $y_{i,\text{physical}}$ , calculated using the physical model from Section 2.4 as the target variable.
  - *VI-PS* indicates the utilization of VI with the physical model as a prior according to Section 2.5.5, with the simulated RF source distance,  $y_i$ , as the target variable.

As anticipated, in this environment, the physical model outperformed the NN methods significantly. Additionally, the error for the baseline is almost zero. It is worth mentioning that the errors for the remaining methods were also within an acceptable range.

Table 3.2. Ideal environment overall performance.

| Method   | MAE [m] | MSE [ $m^2$ ]         | NLL  | Training Time [s] |
|----------|---------|-----------------------|------|-------------------|
| Physical | 0.004   | $7.08 \times 10^{-5}$ | –    | 0                 |
| MAP-S    | 261.62  | 108,385               | 6.66 | 5,752             |
| VI-S     | 217.93  | 94,498                | 6.57 | 19,088            |
| MAP-P    | 198.54  | 69,403                | 6.63 | 3,188             |
| VI-P     | 228.46  | 103,705               | 6.58 | 23,057            |
| VI-PS    | 171.54  | 55,075                | 6.49 | 308               |

The table reveals that most of the methods performed relatively similar. Given it is an ideal environment, it was expected that the models utilizing the physical model to generate the target variable (MAP-P/VI-P), would perform alike to those (MAP-S/VI-S) using the simulated target variable generated with the DGF. Furthermore, it is noteworthy how incorporating the physical model as a prior significantly reduced both MAE and MSE, in comparison to the other NN models.

These results also demonstrate the relative difference in training time required for training deterministic NN models compared to their Bayesian counterparts. However, it is important to note that this observation does not apply to the model trained using the physical model as a prior. In this case, the process involves training the VI-P model, initially. Nonetheless, this approach decreases the training time by approximately two orders of magnitude when the same prior is utilized to train multiple VI-PS models.

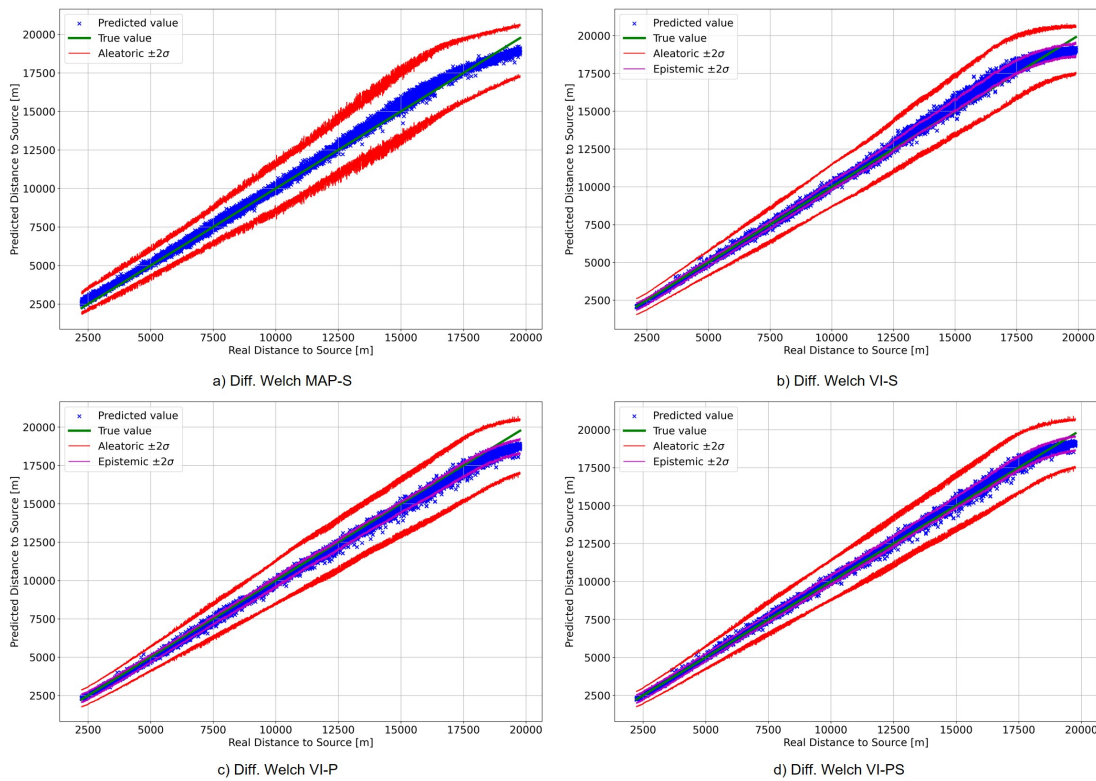


Figure 3.2. Ideal environment predictions with uncertainty metrics.

Figure 3.2 illustrates a plot showing the predicted distances against the ground truth using the test dataset split within a valid trained range. The plot also displays the aleatoric and epistemic uncertainties for these predictions. However, it is impossible to obtain the epistemic uncertainty using the MAP method, as noted in Section 2.6.2, the plot clearly demonstrates how the VI methods generated less noisy predictions for aleatoric uncertainty while also producing epistemic uncertainties.

All the models presented a difficulty in predicting values at larger distances, as depicted by the curvature at the far end. This effect is probably due to the data similarity. The received signals become harder to differentiate at longer distances, because of the use of a logarithmic scale. This scale was used to overcome the difficulties that the NNs presented to learn when the required dynamic range for the input features was too large (see Appendix A).

Nevertheless, it should be clear that VI-PS handled this difficulty better. This is likely to be the case because the VI-P model was used as a prior, thereby maintaining the ability to handle predictions at larger distances. This reflects the importance of prior selection and its regularization effect on the trained model.

### 3.2.2 Uncertainty Metrics

In this idealized environment, most of the methods demonstrated smooth and reasonable aleatoric uncertainty predictions. However, the MAP method produced noisier predictions compared to its VI counterparts, as depicted in Figure 3.3. It is worth mentioning that the aleatoric uncertainty for the MAP methods is equivalent to the total uncertainty of the model, given that the epistemic uncertainty is zero, as can be inferred from Equation (2.63).

The aleatoric uncertainty effectively captures the dataset variability as the distance increases, which is to be expected even in this idealized scenario. This is because small differences in the input signal will result in greater differences for the target variables as the distance increases. The ability of the architecture to handle larger distances is also captured by the aleatoric uncertainty plot. As seen, the MAP and VI-S methods start showing curvature before the VI-P and VI-PS methods. This indicates that the prior not only improves general performance metrics, but also enhances the model's capability to learn to predict larger ranges of values, at least for this basic environment.

Figure 3.4 presents a plot of epistemic uncertainty predictions using the VI methods. The MAP method is excluded, as explained earlier, since its epistemic uncertainty is always zero. In the ideal environment, all three methods performed similarly. However, the VI-P and VI-PS methods exhibited a more pronounced increase in uncertainty as the model predicted values near the limits of the training range, which is a desirable behavior.

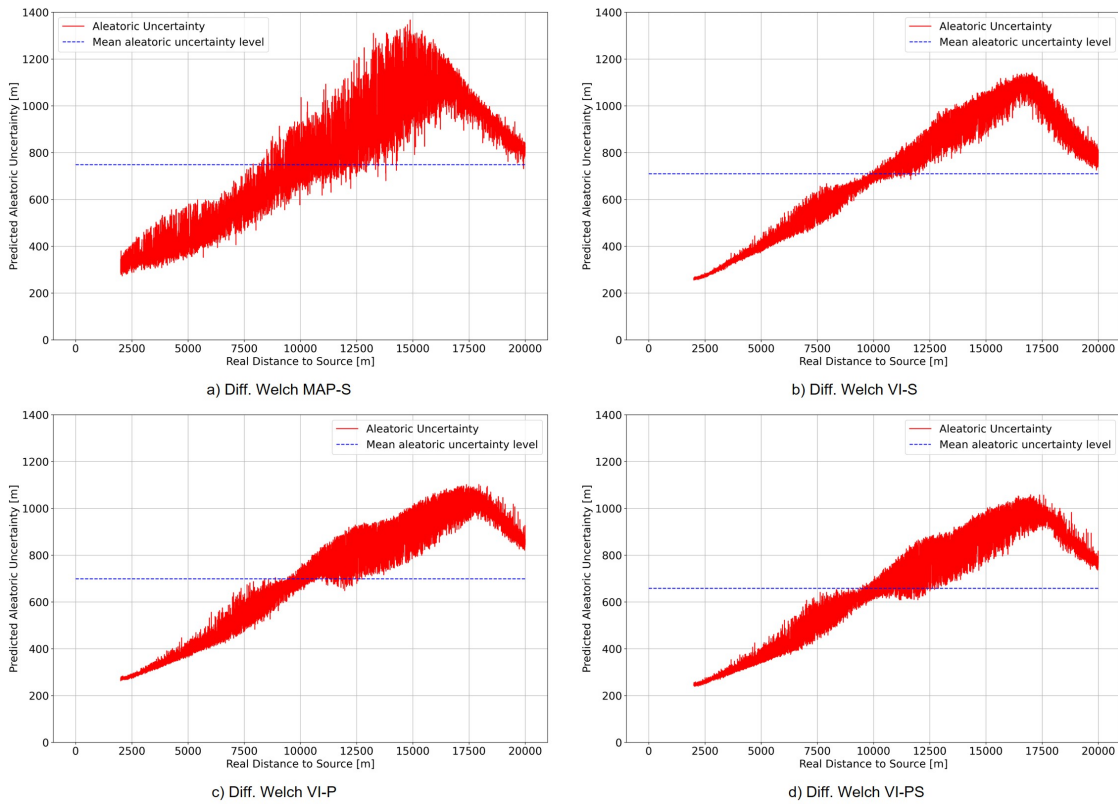


Figure 3.3. Ideal environment aleatoric uncertainty.

Although it may appear noisy, it is important to note that the epistemic uncertainty was relatively small when compared in scale with the predicted distances. This is better illustrated in Figure 3.2. The small MAE obtained for the ideal environment is consistent with the model’s confidence in its predictions. However, relying solely on this information makes it difficult to establish a firm threshold of trust for the model. Information about how it performs under unseen conditions should be obtained first, as discussed in Section 3.2.3.

Finally, it should be noted that the epistemic uncertainty was unable to capture the curvature in predictions. In this case, there is a different effect as the spread of predictions at those large distances tends to be maintained, in contrast to the aleatoric uncertainty, which started to decrease significantly. The implications of this behavior are not clear, and further analysis is required.

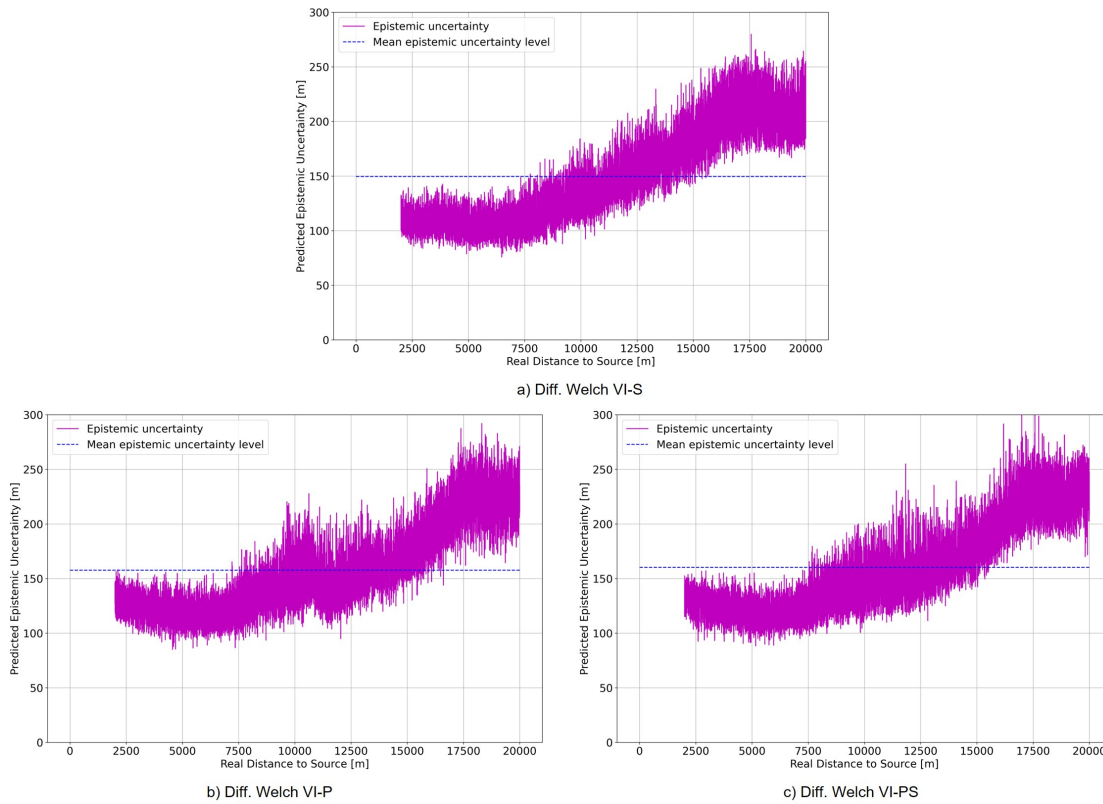


Figure 3.4. Ideal environment epistemic uncertainty.

Figure 3.5 provides a useful representation of the total uncertainty of the models. These graphs depict the probability of locating the RF source at a certain distance from the predicted value, which is an important capability that heteroskedastic deterministic NNs and BNNs can offer in support of the problem presented in Chapter 1. Since the prediction of NN models follow a Gaussian distribution,  $p(y_i|\mathbf{x}_i, \boldsymbol{\theta})$ , there is a 68% probability of finding the source within  $\pm 1\sigma$  of the predicted value, 95% within  $\pm 2\sigma$ , and 99.7% within  $\pm 3\sigma$ .

The total uncertainty, being the sum of the epistemic and aleatoric predictions, exhibits similar behavior to its corresponding components. Once again, the VI-PS method produced a smooth representation, extending the validity of its results further than the VI-S method as the curvature starts to be noticeable at longer distances.

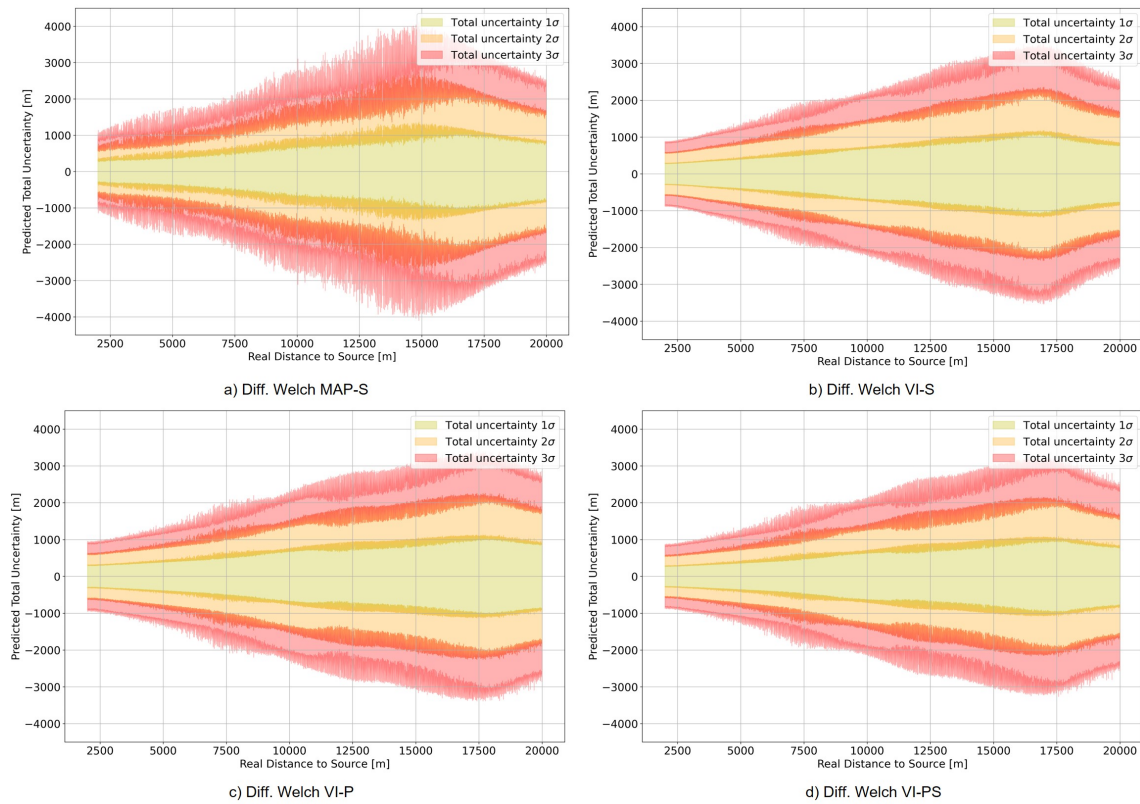


Figure 3.5. Ideal environment total uncertainty.

### 3.2.3 Performance under Unseen Data and Domain Shift Conditions

In order to fully comprehend the advantages of BNNs in comparison to traditional deterministic NNs, conducting a small study to showcase their capabilities is necessary. As mentioned in Section 2.6.2, BNNs enable models to generate epistemic predictions, which indicate the level of uncertainty the model has in its outputs, due to limited training data.

The characterization of BNN methods for unseen and domain shift data begins with Table 3.3, which provides a summary of the performance metrics. As observed, models trained in an ideal environment exhibit poor performance when faced with unseen data or a change in the domain where the model is deployed. Traditional methods would be unable to detect this error. In this example, the domain shift refers to testing the model with the testing dataset split from the standard environment.

Table 3.3. Ideal environment domain shift and unseen data performance.

| Method | MAE [m] | Unseen MAE [m] | Domain Shift MAE [m] |
|--------|---------|----------------|----------------------|
| MAP-S  | 261.62  | 661,08         | 5,409                |
| VI-S   | 217.93  | 774,08         | 5,573                |
| VI-PS  | 171.5   | 700,64         | 5,778                |

Epistemic uncertainty can be utilized to accurately identify new input patterns that were not encountered by the network during its training. Figure 3.6 displays the epistemic uncertainty of the VI models, using both, the original dataset test split from the ideal environment and an extended test dataset that encompasses examples from the entire LOS range within the same environment.

In this scenario, all the models were capable of effectively detecting an increase in epistemic uncertainty as the inputs moved further away from the range covered by the training dataset. As seen, both the VI-P and VI-PS models demonstrated a greater ability to capture changes in epistemic uncertainty, which is expected since VI-P is used as the prior for VI-PS, particularly in the presence of unseen data. The range of the tested data is asymmetrical due to the need for a larger range to observe an increase in epistemic uncertainty for longer distances. This observation aligns with the network's difficulty in accurately predicting ranges at greater distances, as the input features become more challenging to distinguish with small increments of distance. These findings provide strong evidence for the capabilities of epistemic uncertainty, an effect that cannot be captured by aleatoric uncertainty.

It should be noted that establishing an absolute threshold to inform an operator that the model's predictions cannot be blindly relied upon is a challenging task. For the VI-PS model, a suitable threshold at shorter distances appears to be approximately the mean value of the epistemic uncertainty. However, this threshold becomes invalid for distances greater than approximately 8 km. Establishing a threshold at the far end is even more difficult. In order to use this model to establish a hard limit of trust, a table presenting ranges

and expected epistemic uncertainty values could be provided. However, it is important to highlight that this table is specific to the model and needs to be characterized individually, as demonstrated by the differences observed between the VI-S and VI-PS models and the variations across different environments, as will be shown in the next sections.

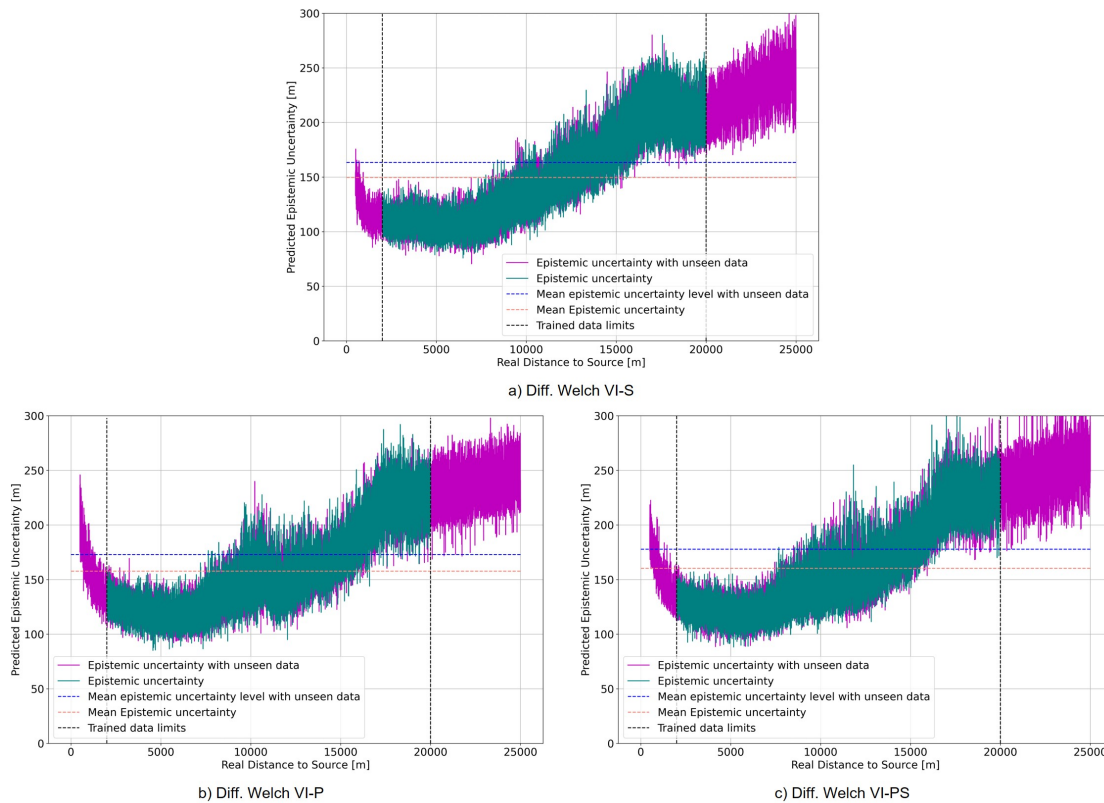


Figure 3.6. Ideal environment epistemic uncertainty with unseen data.

Lastly, Figure 3.7 depicts the behavior of epistemic uncertainty under domain shift conditions. For the models trained with the ideal dataset, all methods demonstrated the ability to easily detect a domain shift. However, caution should be exercised as this result does not hold when transitioning between the standard environment to the harsh environment, and vice versa. This will be discussed in detail, later.

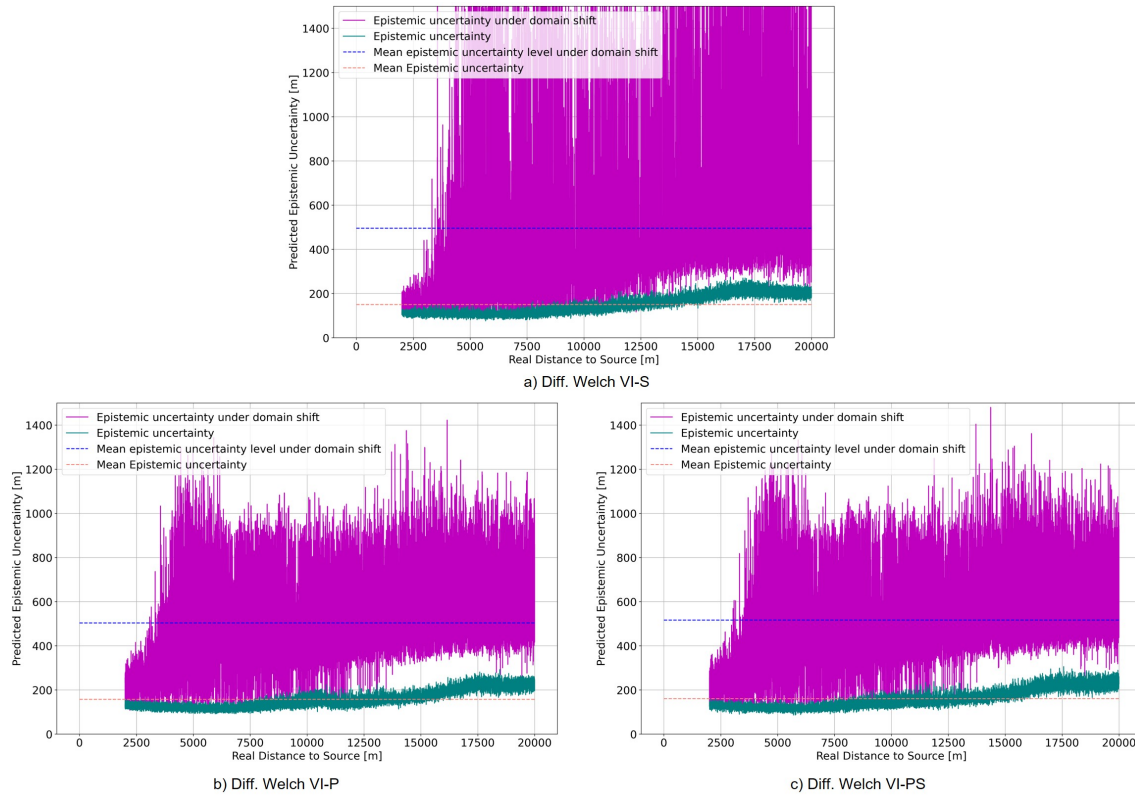


Figure 3.7. Epistemic uncertainty with shifted domain (Ideal  $\rightarrow$  Standard).

### 3.3 Standard Environment Results

The standard environment serves as the main point of comparison between the performance of the NN methods and the baseline. This environment is used to provide stronger evidence supporting the findings and conclusions in this thesis.

Figure 3.8 shows a graphical representation of the transmitted and received signal at 10 km for the standard environment, in the time and frequency domain. Unlike the ideal environment, the effects of noise, frequency offset, and other factors are noticeable, but the waveform remains easily recognizable.

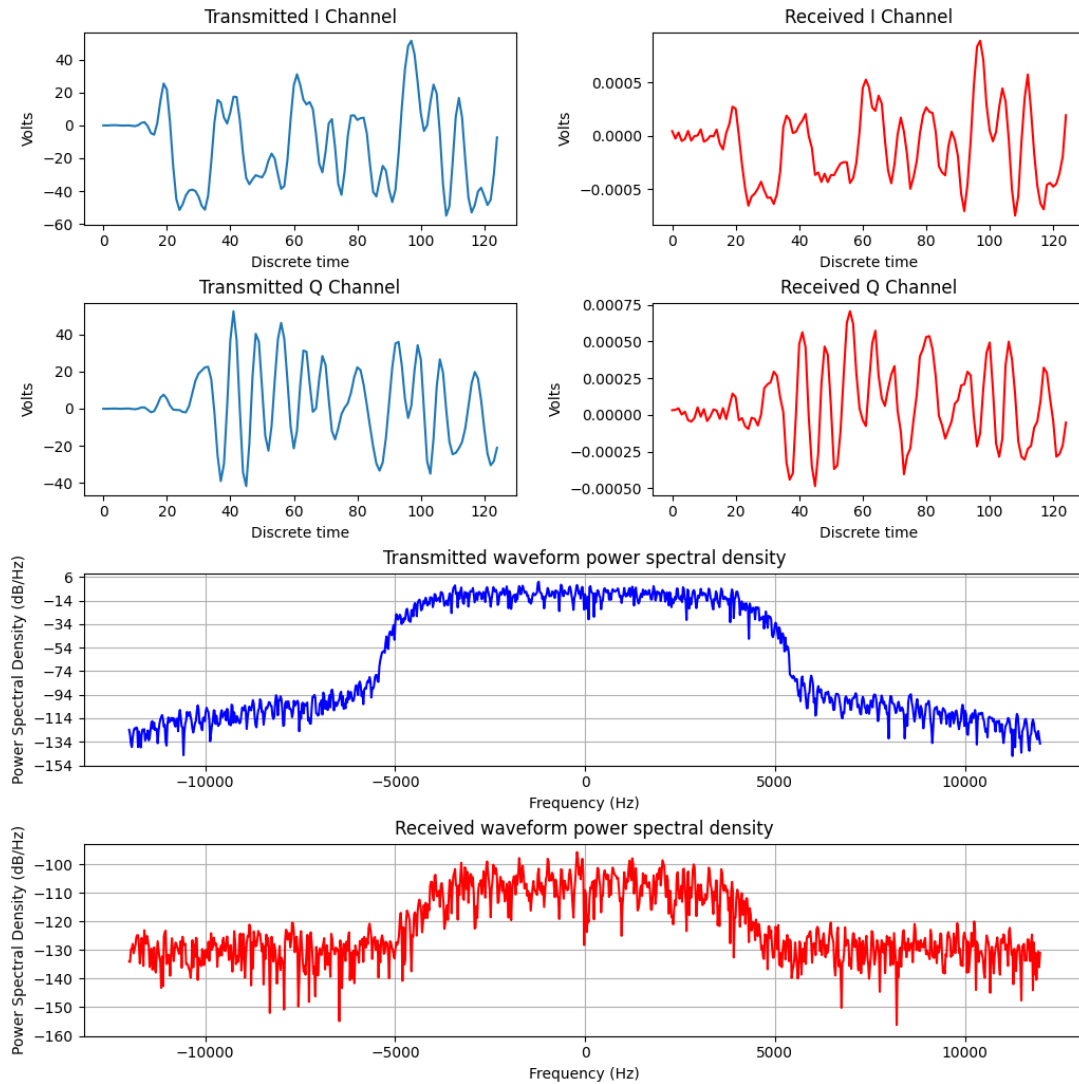


Figure 3.8. Standard environment representative example showing a transmitted signal with its corresponding received signal at 10 km.

### 3.3.1 Performance Results

Table 3.4 presents the performance results of the physical model and the developed architecture in the standard environment. In this environment, the two NN methods based on the simulated target variable outperformed the physical model baseline. MAP-P and VI-P were trained to predict the physical model; therefore, it is expected to observe a performance

close to the baseline. In particular, the VI-PS model reduced the MAE by approximately 45% and the MSE by 74% compared to the baseline. Additionally, it exhibited the best performance among all the measured metrics for all the methods.

Table 3.4. Standard environment overall performance.

| Method   | MAE [m] | MSE [ $m^2$ ] | NLL  | T. Time [s] |
|----------|---------|---------------|------|-------------|
| Physical | 827.36  | 1,516,480     | –    | 0           |
| MAP-S    | 523.0   | 519,816       | 7.10 | 1,763       |
| VI-S     | 536.1   | 490,497       | 7.18 | 9,501       |
| MAP-P    | 1,004   | 2,078,653     | 8.51 | 1,652       |
| VI-P     | 956.1   | 1,978,473     | 8.32 | 13,318      |
| VI-PS    | 450.0   | 387,205       | 6.94 | 305         |

The table reveals differences in the performance of the methods. For instance, VI-S had a higher MAE compared to MAP-S, but presented a lower MSE. As mentioned earlier, VI-PS performed better than both MAP-S and VI-S models, reducing the MAE by 14% and 16%, and the MSE by 25% and 21%, respectively. Similar to the ideal environment case, incorporating the physical model as a prior significantly improved the NN model's performance.

Once again, these results are consistent with the ideal environment, as they demonstrate the relative difference in time required for training deterministic NN models compared to their Bayesian counterparts, highlighting the advantage of the VI-PS method in this regard.

Figure 3.9 depicts a plot of the predicted distances against the ground truth using the test dataset split with data within the range seen during training. In this environment, the advantages of VI is even more pronounced, compared to the deterministic NNs. The aleatoric uncertainty is noticeably smoother, and the predictions are clustered closer to the true value, although there is no improvement in handling outliers.

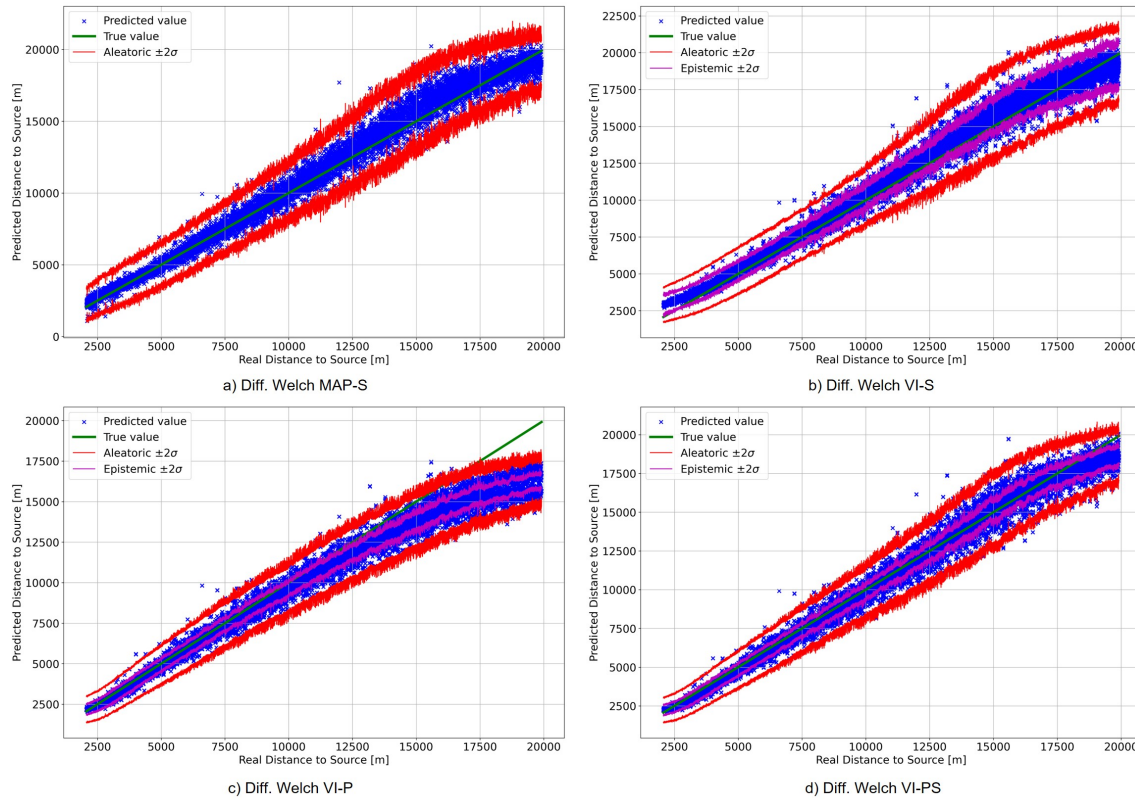


Figure 3.9. Standard environment predictions with uncertainty metrics.

It is interesting to observe the tendency of VI-S to overestimate predictions, while VI-P tends to underestimate them. In contrast, VI-PS arrives at predictions that lie in between and closer to the ground truth. This observation serves as an argument in favor of the VI-PS method, as the VI-P prior reduces the risk of overestimation that the classical VI-S model (using a zero-mean Gaussian prior) produces. These results highlight the importance of prior selection and its regularization effect.

Similar to the ideal environment case, all the models encountered difficulty in predicting values at larger distances, as shown by the curvature at the far end. There is no argument against the potential causes of this effect, as discussed in Section 3.2.1. As observed, VI-PS appears to handle this difficulty more effectively by producing a plot with a smaller curvature.

### 3.3.2 Uncertainty Metrics

In a more complex environment, the MAP method noticeably generated noisier predictions compared to its VI counterparts, as shown in Figure 3.10. The VI-PS aleatoric predictions not only exhibit a more linear pattern compared to VI-S, but also yield smaller uncertainty values within the same range.

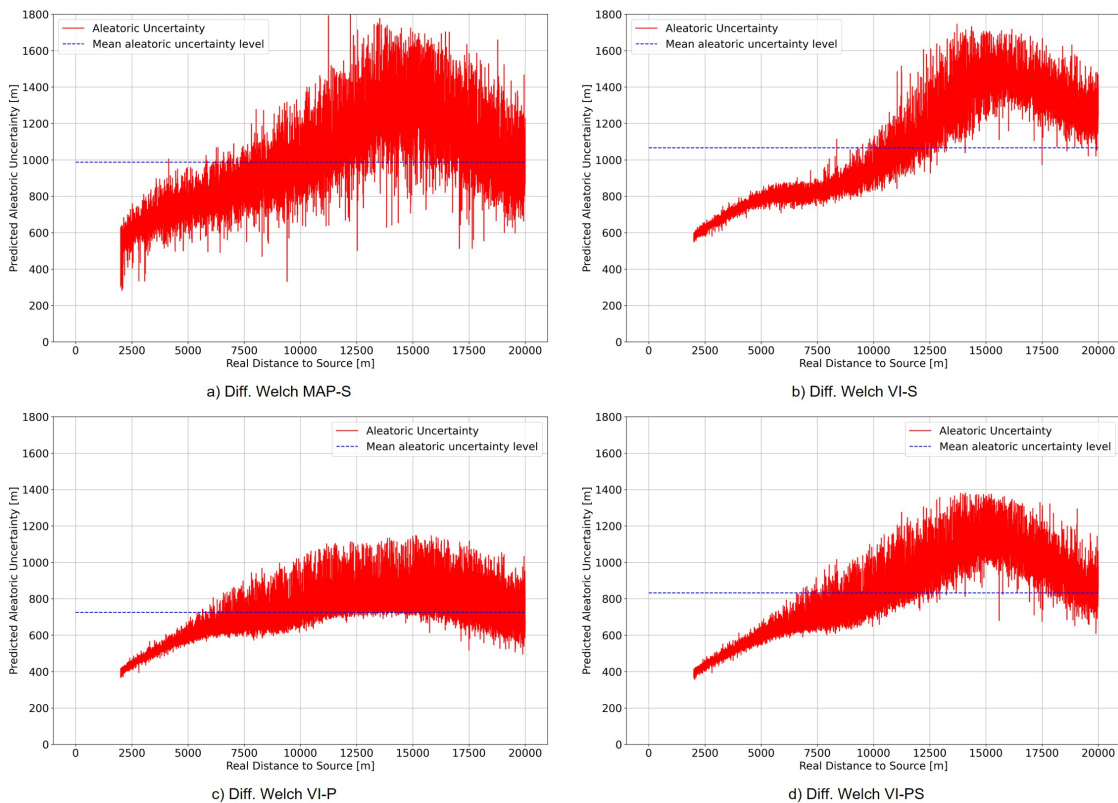


Figure 3.10. Standard environment aleatoric uncertainty.

In this environment, the aleatoric uncertainty captures the variability of the dataset, which increases as the distance increases, due to the same causes mentioned in Section 3.2.2. As observed, the VI-PS model demonstrates a slight advantage in reducing the effects noted in Section 3.2.2, particularly at the starting point of the curvature compared to the VI-S model, though it is still present.

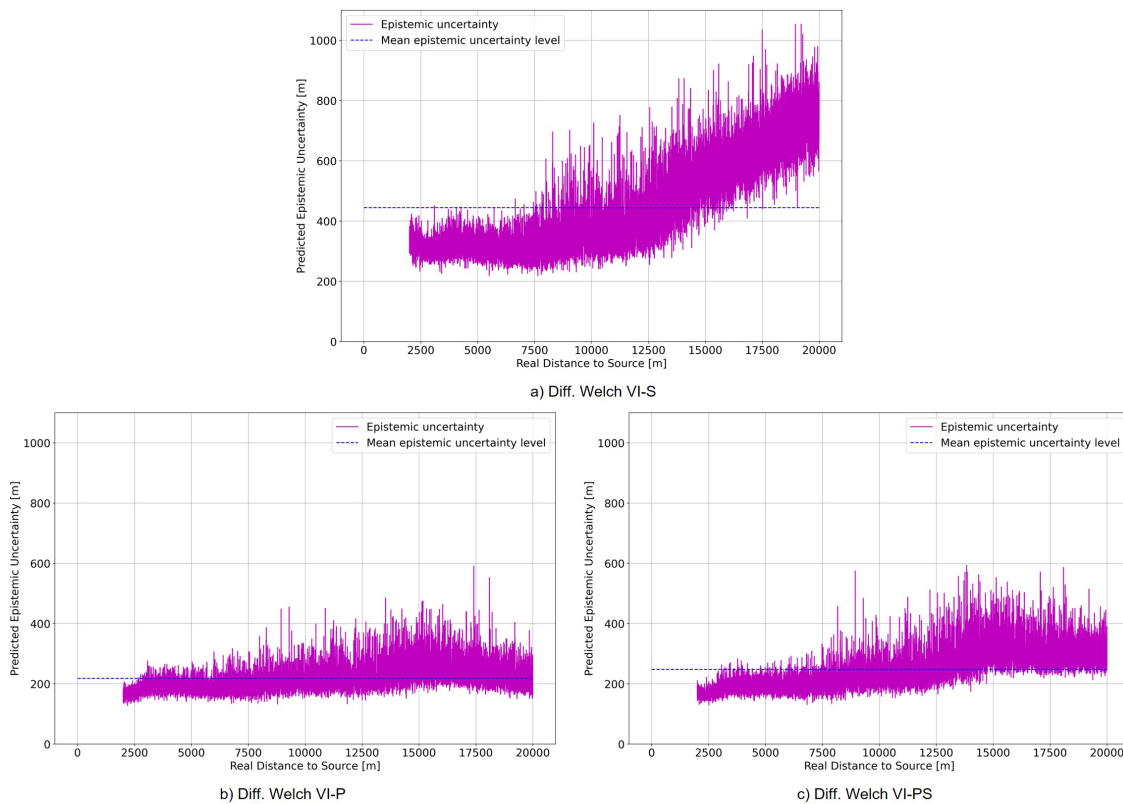


Figure 3.11. Standard environment epistemic uncertainty.

Figure 3.11 presents the plots of epistemic uncertainty predictions. Compared to the ideal environment, the effects of the epistemic uncertainty using the VI-P model as the prior are significant. First, note that the mean values for the epistemic predictions are much lower than the VI-S model, and they are also more stable across the prediction range. This could be interpreted as the model being more confident in its predictions, supporting the performance results in Section 3.3.1. The relative size of the epistemic uncertainty for the different models is better captured in Figure 3.9. The true utility of these results is better appreciated in Section 3.3.3.

Establishing an absolute threshold of trust for the models is still difficult. However, it is even more challenging for the VI-S model, as the range of predicted epistemic values increased significantly. Meanwhile, for the VI-PS method, the range remains under a more constrained

scope of possible values. In line with the ideal environment, the epistemic uncertainty was unable to capture the curvature in predictions.

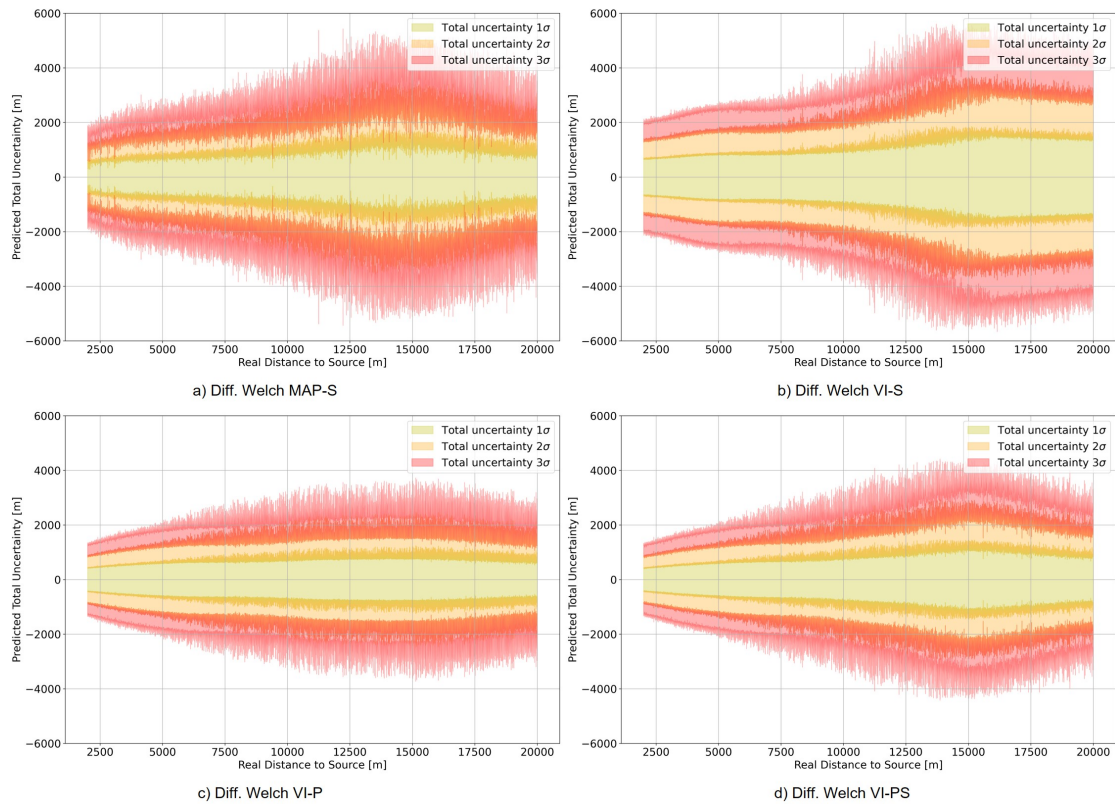


Figure 3.12. Standard environment total uncertainty.

Figure 3.12 depicts a useful representation of the total uncertainty of the models. As explained in a previous section, these graphs depict the probability of locating the RF source at a certain distance from the predicted value. Across the entire range of trained distances, the possible areas where the RF sources can be found are greater compared to the ideal environment. This is reasonable because a more complex environment will contain larger variances.

### 3.3.3 Performance under Unseen Data and Domain Shift Conditions

Table 3.5 presents a summary of the performance metrics when the models are exposed to unseen input features and a domain shift. As shown, models trained in a standard environment exhibit poor performance when faced with unseen data or a change in the domain from which the models were trained.

Table 3.5. Standard environment domain shift and unseen data performance.

| Method | MAE [m] | Unseen MAE [m] | Domain Shift MAE [m] |
|--------|---------|----------------|----------------------|
| MAP-S  | 523.0   | 933.13         | 2,474                |
| VI-S   | 536.1   | 885.29         | 2,287                |
| VI-PS  | 450.0   | 934.70         | 2,713                |

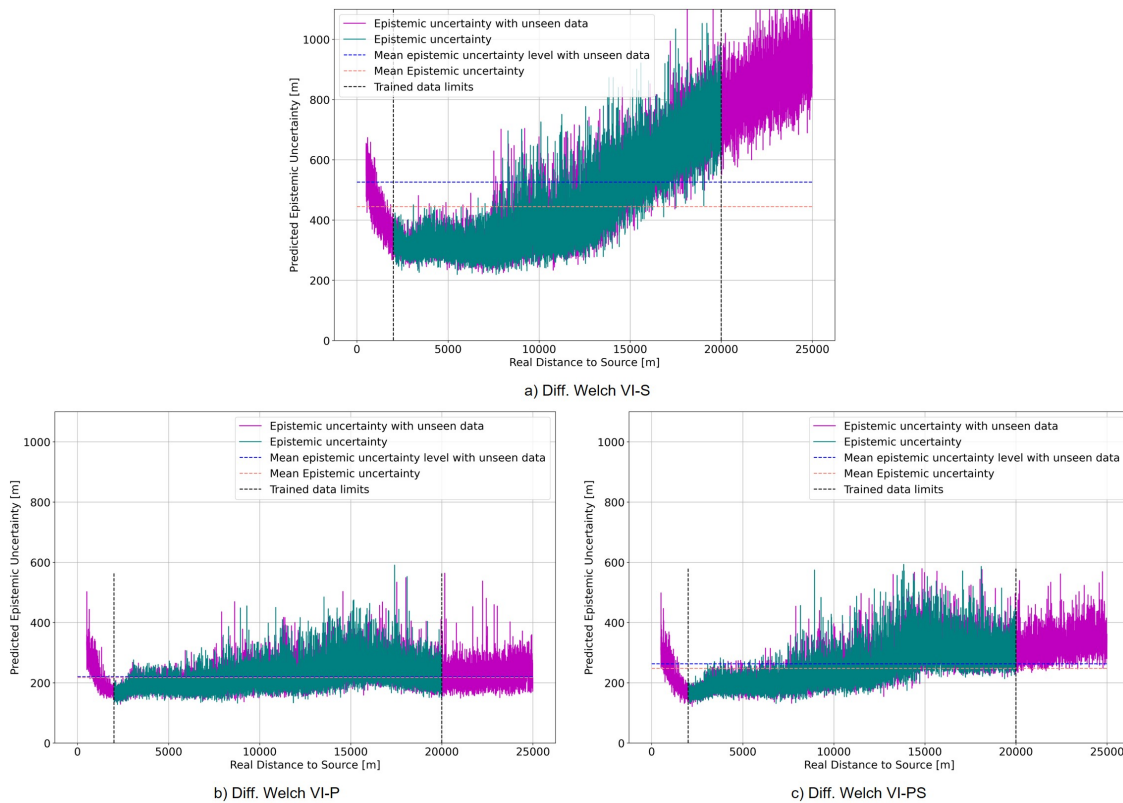


Figure 3.13. Standard environment epistemic uncertainty with unseen data.

The performance of the models on unseen data is comparable to the baseline predictions for the standard environment. Furthermore, they are even better than the physical model when tested using the harsh dataset of Section 3.4, as seen in the domain shift MAE results.

Figure 3.13 displays the epistemic uncertainty of the VI models using both the original dataset test split from the ideal environment and an extended test dataset that includes examples from the entire LOS range within the same environment.

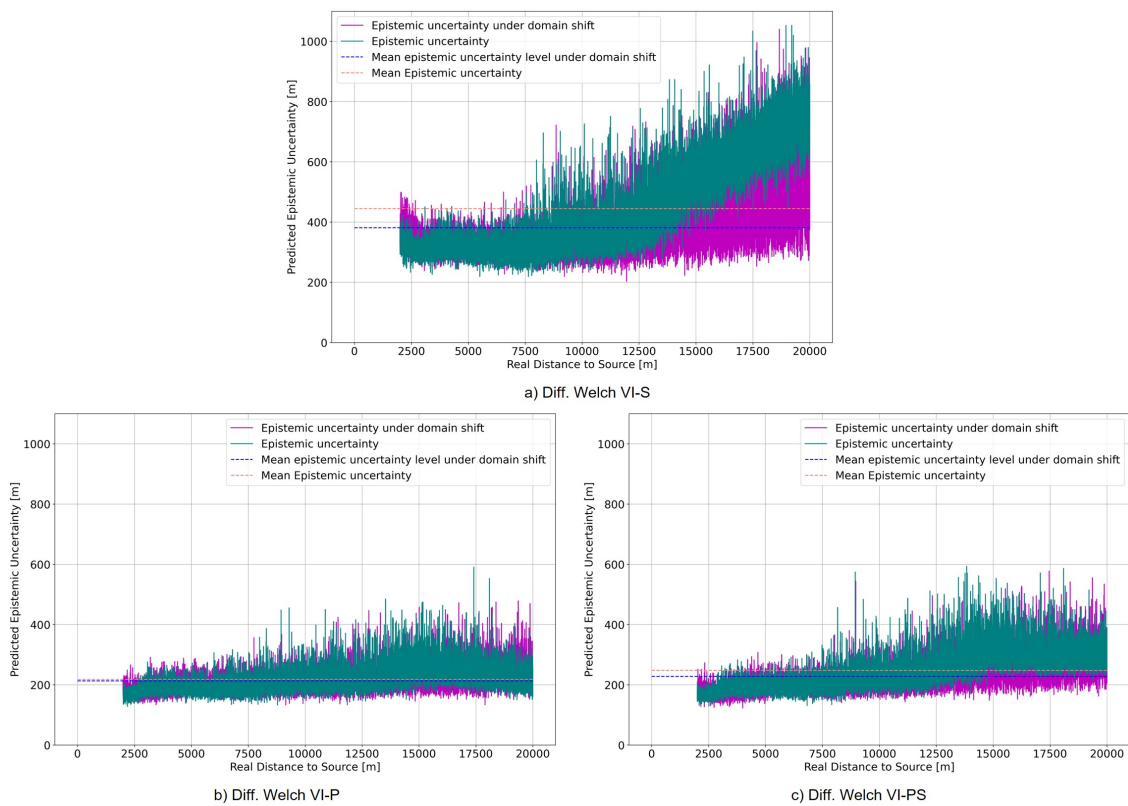


Figure 3.14. Epistemic uncertainty with shifted domain (Ideal → Standard).

Same as in the ideal case, in the standard scenario, all the VI-S and VI-PS models were capable of effectively detecting an increase in epistemic uncertainty as the inputs moved further away from the range covered by the training dataset. However, the difficulty of establishing an absolute threshold for longer distances remains. On the other hand, for

smaller distances, it seems to be easier for the models to establish a threshold, particularly for the VI-PS model, as its uncertainty prediction variability tends to be smaller at such ranges. This graph also supports the point made earlier in Section 3.2.3 that the limits of trust depend greatly on the environmental conditions captured by the dataset.

Contrary to the results in the ideal environment, Figure 3.14 shows that it is not possible to detect a domain shift from the standard to the harsh environment. For this reason, epistemic uncertainty cannot always be trusted for detecting errors in the predictions. This is the case because the NN may produce erroneous values while providing a low epistemic uncertainty, which could lead an operator to trust an erroneous prediction without a second thought.

### **3.4 Harsh Environment Results**

The harsh dataset is used to test the models in a more complex environment and to verify their consistency with the analysis made for the standard environment. It is also used to check the limits of the architecture, models and methods.

Figure 3.15 presents a graphical representation of the transmitted and received signal at a 10 km in the harsh environment, both in the time and frequency domains. Similar to the standard environment, the effects of noise, frequency offset, and other factors are apparent.

#### **3.4.1 Performance Results**

Table 3.6 presents the performance results of both the physical model and the developed architecture in the harsh environment.

In this environment, the two NN methods based on the simulated target variable exhibited better performance than the baseline. Similar to the standard case, the MAP-P and VI-P were trained to predict the physical model, resulting in their performance being close to the baseline. Upon closer examination, the VI-PS and VI-S methods performed almost identically and showed a reduction in MAE of approximately 50% and MSE of 70% compared to the baseline.

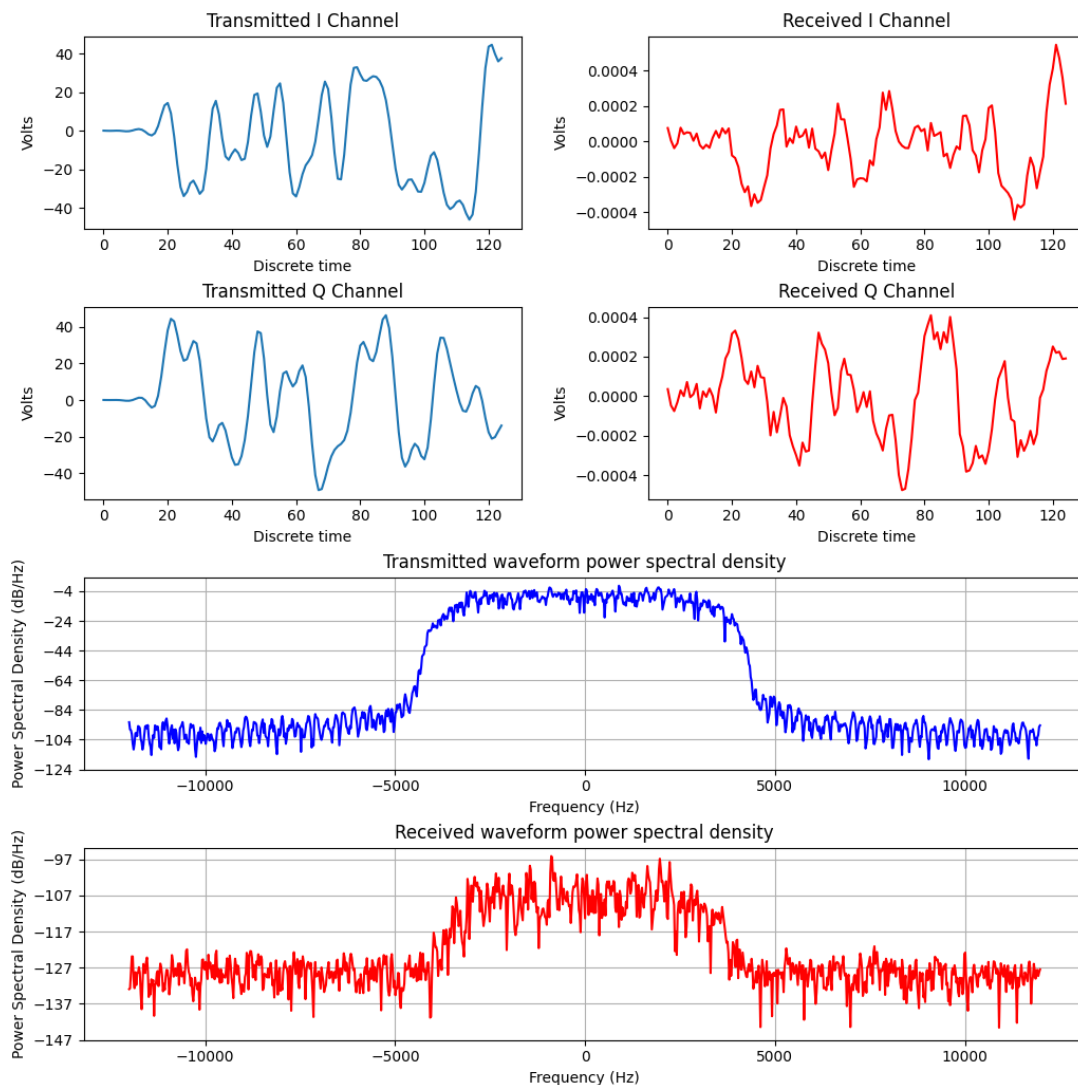


Figure 3.15. Harsh environment representative example showing a transmitted signal with its corresponding received signal at 10 km.

Interestingly, the MAP-S method performed the best in this scenario, surpassing both the VI-S and VI-PS in all metrics by a small margin. In such a challenging scenario, models were discovered that yielded significantly better results compared to the baseline. However, it is worth noting that these results may not be universally applicable and would depend on the specific application, considering that the error is approximately three times greater than in the standard environment.

Table 3.6. Harsh environment overall performance.

| Method   | MAE [m] | MSE [ $m^2$ ] | NLL   | T. Time [s] |
|----------|---------|---------------|-------|-------------|
| Physical | 2,907   | 12,316,210    | –     | 0           |
| MAP-S    | 1,381   | 3,240,905     | 7.857 | 2,899       |
| VI-S     | 1,463   | 3,758,782     | 7.975 | 15,881      |
| MAP-P    | 2,931   | 13,007,858    | 11.44 | 3,063       |
| VI-P     | 3,224   | 15,428,905    | 16.97 | 25,479      |
| VI-PS    | 1,488   | 3,787,183     | 7.925 | 267         |

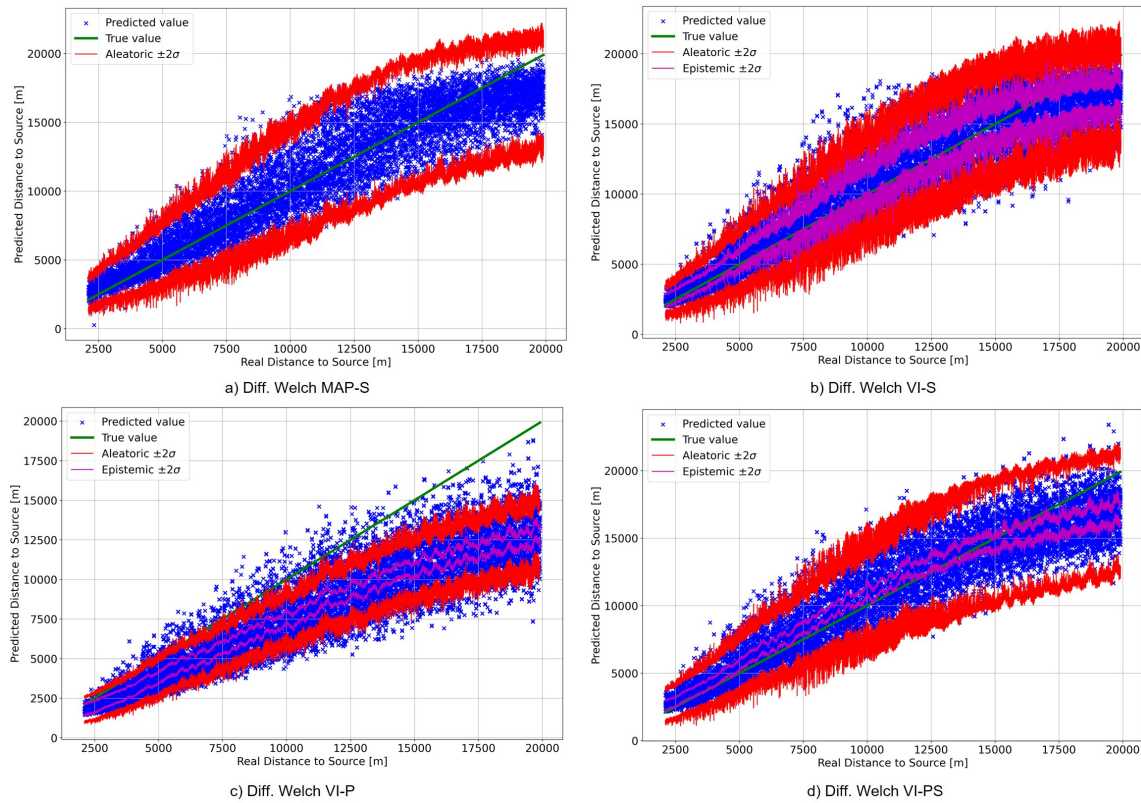


Figure 3.16. Harsh environment predictions with uncertainty metrics.

Figure 3.16 presents a plot comparing the predicted distances with the ground truth values using the test dataset within a valid trained range. In this environment, it is possible to better appreciate the challenges posed. As shown, the predictions are spread across much larger distances compared to the standard environment.

In this setup, the VI-S model still tends to overestimate the predictions, while the VI-P model tends to underestimate them. However, contrary to what was observed in the standard environment, the VI-PS model also overestimates the predictions, similar to the VI-S case. In this scenario, the pulling effects of the VI-P model are not noticeable, as its predictions are far from the ground truth used to train the VI-PS model. It is important to note that this does not apply to the uncertainty metrics, which will be analyzed next.

### 3.4.2 Uncertainty Metrics

For this extreme scenario, the MAP method generated less noisy aleatoric predictions compared to its VI counterparts, as demonstrated in Figure 3.17. The linearity of the VI-PS aleatoric predictions is still apparent, but not as pronounced as in simpler cases. In this environment as well, the aleatoric uncertainty captures the variability of the dataset, which increases as the distance increases for the same reasons mentioned in Section 3.2.2.

The three methods based on the simulated dataset reached an aleatoric ceiling of around 2,500 *m*. The interpretation of this phenomenon is not clear. However, a similar behavior was observed using other pre-processing methods that had difficulties handling a large input dynamic range, as explained in Appendix A. This may indicate the limitations of the developed architecture, but further analysis is needed to draw a valid conclusion.

As shown, the VI-PS model still generated a less noisy aleatoric uncertainty compared to its VI-S counterpart. Considering that both models performed similarly with regards to the error metrics, there is still a clear advantage to using the physical model as a prior, as mentioned in Section 2.5.5.

Figure 3.18 presents the plots of epistemic uncertainty predictions. In the harsh environment, the advantage of the VI-PS model are even more evident than in the standard scenario. First, note that the mean values for the epistemic predictions are much lower than the VI-S model, and they are also less variable across the prediction range. A different phenomena

is observed for the epistemic predictions of the VI-S model, as it starts to reduce in the far end. On the other hand, the VI-PS does not suffer as much from that tendency, showing a similar behavior as in the standard environment. The interpretation, though, is the same as the analysis of the epistemic uncertainty of Section 3.3.2, regarding that the VI-PS model is more confident in its predictions.

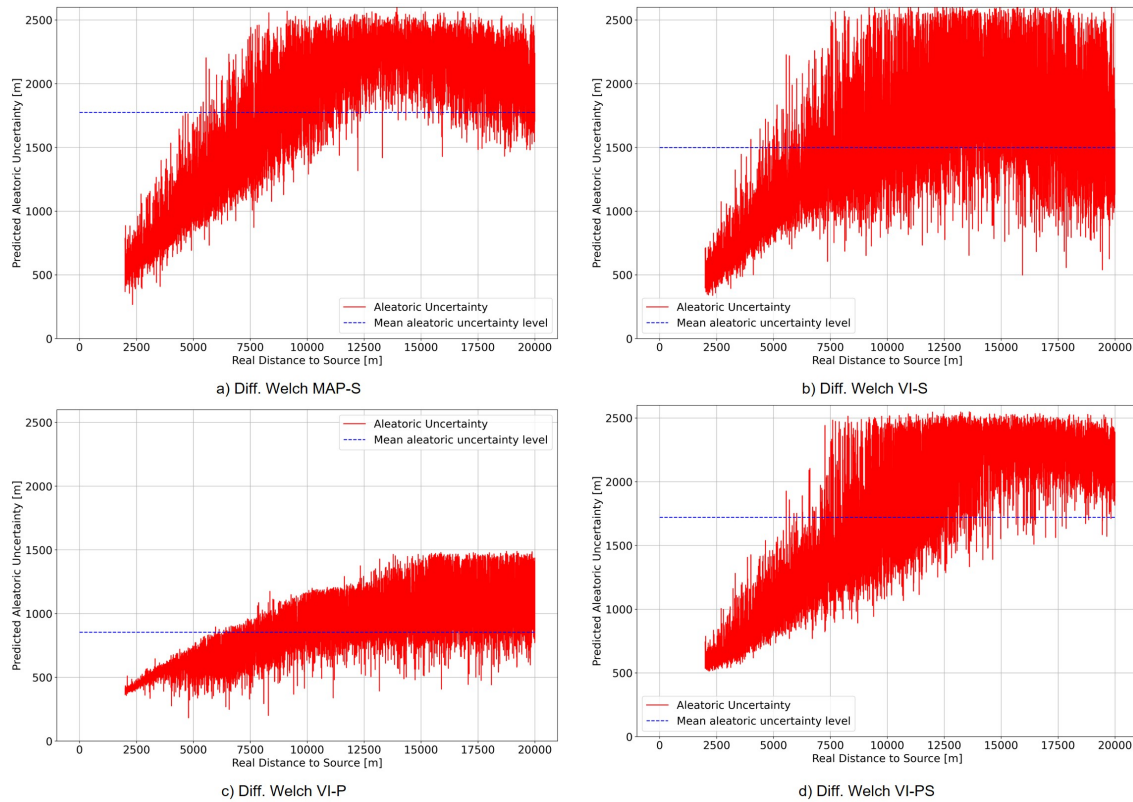


Figure 3.17. Harsh environment aleatoric uncertainty.

Figure 3.19 illustrates the total uncertainty of the models. As described in the ideal environment section, these graphs represent the probability of locating the RF source at a specific distance from the predicted value. No additional observation could be obtained for the harsh environment, as the total uncertainty is the combination of the aforementioned aleatoric and epistemic effects.

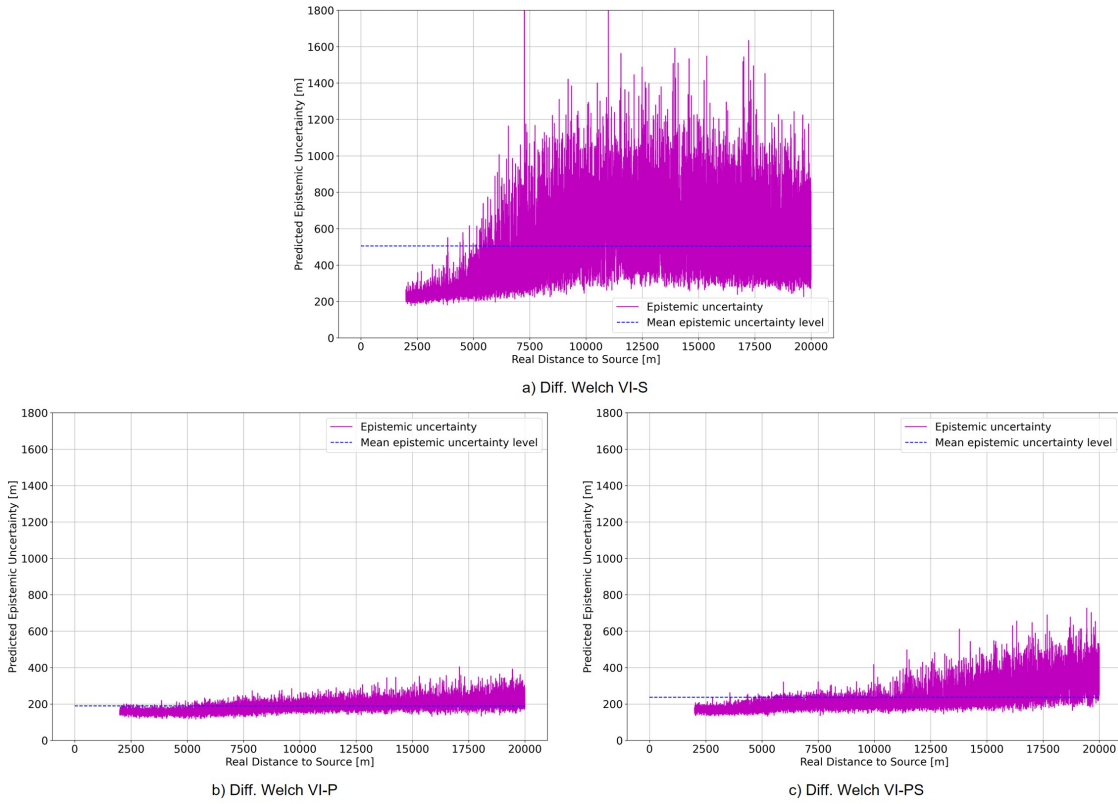


Figure 3.18. Harsh environment epistemic uncertainty.

### 3.4.3 Performance under Unseen Data and Domain Shift Conditions

Table 3.5 presents a summary of the performance metrics when the models are exposed to unseen patterns and a domain shift from the standard environment. As shown, models trained in the harsh environment exhibit comparable performance when faced with unseen data or a change in the domain from which the models were trained. In this sense, the models performed better than the ones obtained for the ideal and standard environment.

Here, for all the methods, the performance on unseen data is better compared to baseline predictions for the harsh environment. Conversely, they perform worse than the baseline, when tested using the standard dataset as a shifted domain with MAE.

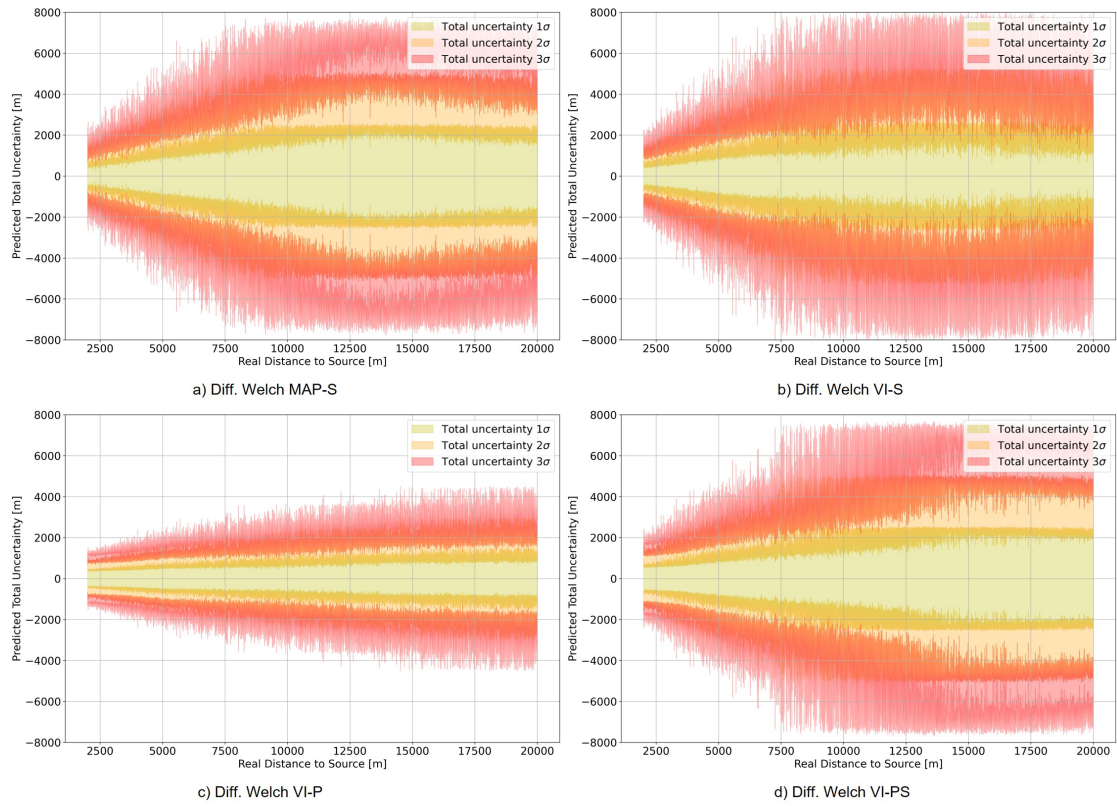


Figure 3.19. Harsh environment total uncertainty.

Table 3.7. Harsh environment domain shift and unseen data performance.

| Method | MAE [m] | Unseen MAE [m] | Domain Shift MAE [m] |
|--------|---------|----------------|----------------------|
| MAP-S  | 1,381   | 2,030          | 2,043                |
| VI-S   | 1,463   | 2,056          | 2,204                |
| VI-PS  | 1,488   | 2,107          | 2,161                |

Figure 3.20 displays the epistemic uncertainty of the VI models using both the original dataset test split from the harsh environment and an extended test dataset that includes examples from the entire LOS range within the same environment. Similar to what was observed in the standard scenario, the VI-PS was able to effectively detect unseen inputs at both ends of the input range. However, this is not the case for the VI-S, where the model struggles to effectively detect new inputs at the far end. As mentioned earlier for the other environments, defining thresholds of trust poses similar difficulties.

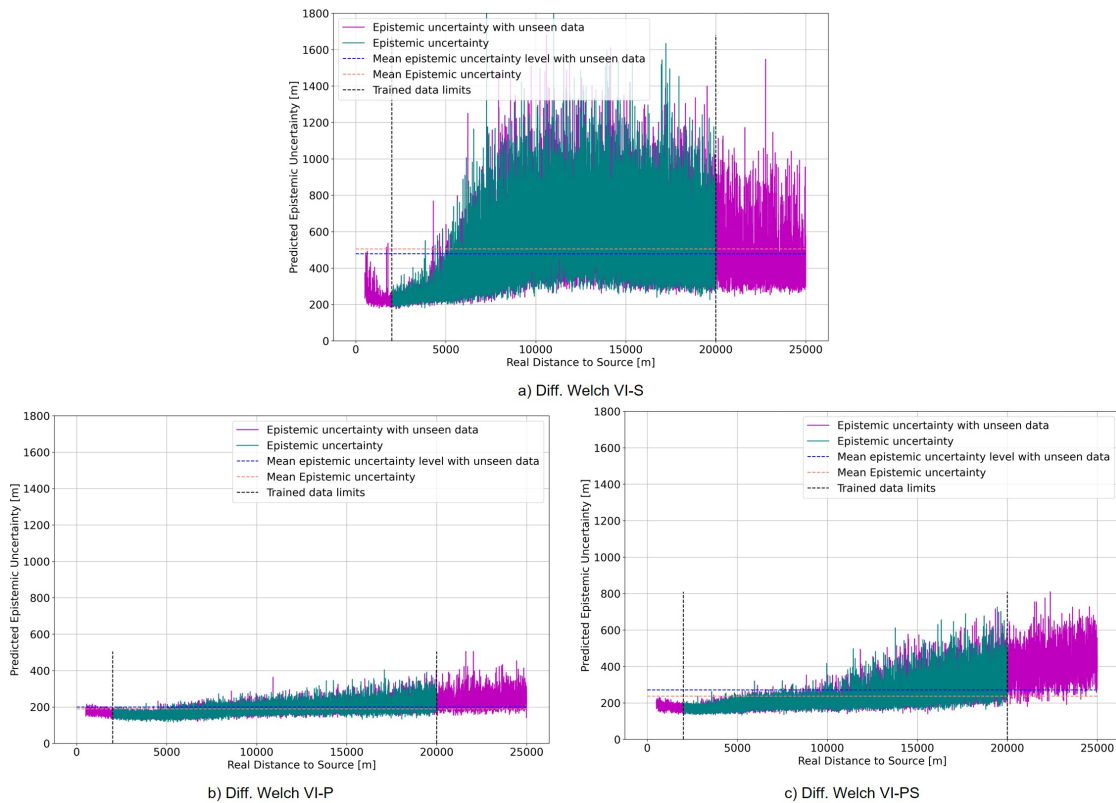


Figure 3.20. Harsh environment epistemic uncertainty with unseen data.

In line with the results of Section 3.3.3, the models trained in the harsh environment are unable to detect domain shift changes. However, in this case, this is actually the correct behavior, as the standard environment could contain examples that are also present in the

harsh environment. This is because the DGF generates random examples from the boundary parameters of the dataset, where the parameters of the standard environment are a subset of the harsh parameters.

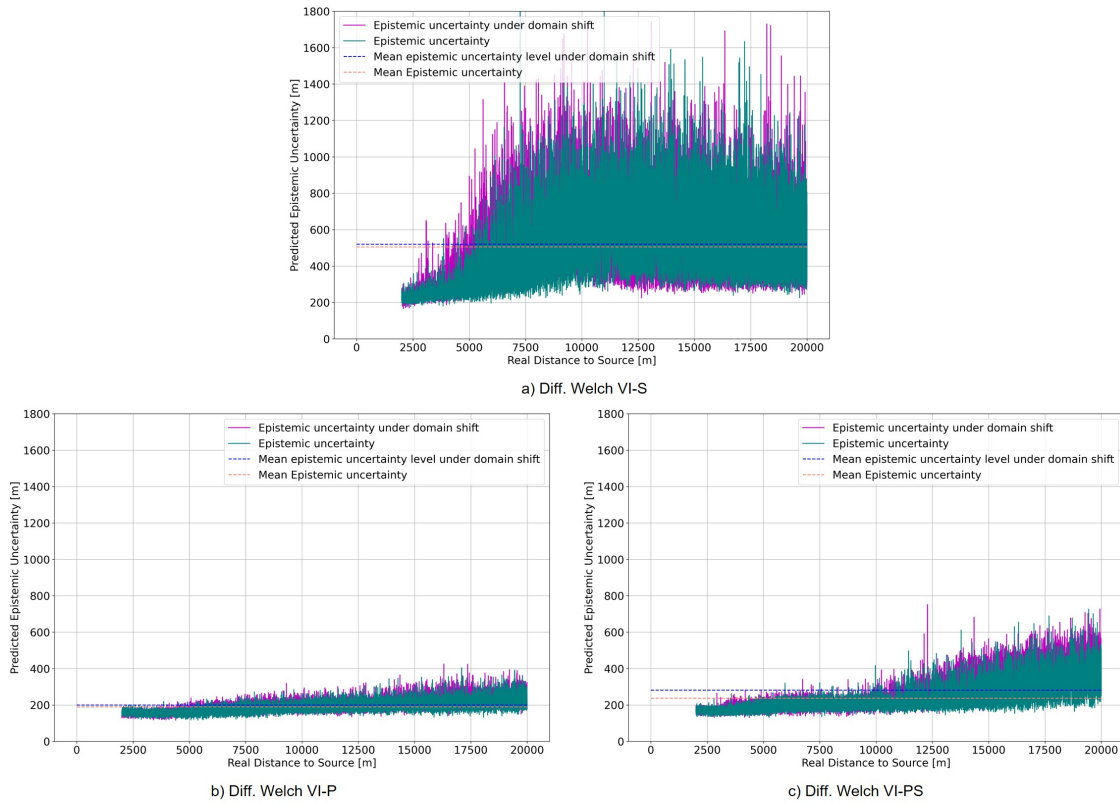


Figure 3.21. Epistemic uncertainty with shifted domain (Harsh  $\rightarrow$  Standard).

---

## CHAPTER 4:

### Conclusions and Discussion

---

#### 4.1 Conclusions

This work effectively demonstrates that it is possible to outperform the physical model in estimating the range to a radio-communication source using NN methods, while also providing aleatoric and epistemic uncertainty for predictions. However, caution must be exercised, as this result was obtained in a simulated environment and has not been tested in a real-world setting. Nevertheless, this result is particularly valuable in terms of paving the way for future implementations in radio-ranging using simple SDRs for embedded devices in military applications. Since the primary objective of this work was not directly focused on maximizing the performance of the architecture, there could be room for improvement.

While the total uncertainty is valuable in generating a range of possible transmitter locations, the epistemic uncertainty serves as an indicator of trust in the model's prediction when the model is applied to unseen data. This work demonstrates that epistemic uncertainty can serve as a rough, relative measure of confidence; nevertheless, utilizing this confidence based on a threshold depends on domain specific knowledge and extensive characterization of the model's behavior in the environment. No absolute threshold could be derived. Nevertheless, the trained BNNs were able to produce an easily detectable increase of epistemic uncertainty when presented with unseen data. This provides a clear indication to the operator that they should not blindly trust such predictions, even if they happen to closely align with the ground truth.

One of the biggest caveats is that the model may not perform well for all receiver, transmitter, and environmental conditions, as depicted by the domain shift performance experimentation. While this issue could be mitigated by expanding the dataset with more examples created using different conditions, the training data will still only represent a subset of all possible configurations. It may also be beneficial to consider incorporating additional input features that provide more comprehensive information about the receiver's status and the knowledge available regarding the transmitter and environment.

Another important consideration is that the model tends to fail silently in the event of a domain shift, meaning there are not always significant spikes of uncertainty that allow for detecting the domain shift. This can be problematic, especially for an operator who is unaware of the model's limitations. Even worse, if the model produces a low epistemic uncertainty value, it could falsely instill confidence regarding the prediction. As suggested in the previous paragraph, the addition of more features could potentially help mitigate this issue.

While VI methods can effectively provide an approximation to total model uncertainty by including aleatoric and epistemic uncertainty, their effectiveness is limited by the selected prior. In this case, having a physical model is useful since it can be used to obtain a better prior. This improved prior not only helps achieve a more performant model and tighter uncertainty estimates, but also significantly reduces the training time required for a new NN. Obtaining this new prior is not free, though; it requires training an entire VI BNN, which is as costly as VI with a standard prior. However, it has the advantage that it only needs to be trained once if later models use the same architecture.

## 4.2 Discussion

Typically, a large number of assumptions are made when modeling a system using BNNs. These assumptions include assuming various Gaussian distributions—chosen due to its mathematical properties and the applicability of the Central Limit Theorem—in different expressions throughout the methods:

- The PDF,  $p(y|\mathbf{x}, \boldsymbol{\theta})$ , used to model the regression problem.
- The prior,  $p(\boldsymbol{\theta})$ , that represents the parameter distribution.
- The variational distribution,  $q_{\phi}(\boldsymbol{\theta})$ , of the VI method used to get an approximation to the real posterior.
- The VI-P posterior,  $p(\boldsymbol{\theta}|\mathcal{D}_{\text{physical}})$ , that is used as prior for the VI-PS model.

However, even though Gaussian distributions have favorable properties, intractable expressions arise, necessitating the use of approximation techniques such as Monte Carlo Sampling. Having substantial knowledge of the environment should help in selecting the best assumptions for the problem, and using an environment simulator greatly aids in gaining this knowledge. As used in this thesis, teaming ML methods with simulators could

eventually help reduce the risks of implementation failures, thereby reducing costs and time. Additionally, using different assumptions can lead to different, or even misleading, posterior probability distributions, affecting the uncertainty estimates, accordingly. Future research could focus in obtaining better priors for the BNNs models by utilizing Functional Variational Inference [46] or implicit methods [47].

For this thesis, the simulated environments represent a limited set of digital communications scenarios that are restricted by the chosen effects, path-loss model, and assumptions implemented as part of the DGF. These limitations will reduce the performance of the obtained models in a real-world environment. Therefore, further research should be conducted to extend this work and account for a wider range of conditions. By doing so, more general models can be obtained, which could eventually be deployed.

The accuracy of the uncertainty prediction depends on the data capability of representing reality, capabilities of the NN architecture, the method used to estimate the posterior, the prior selection, and even the assumed underlying model distribution  $p(y|\mathbf{x}, \boldsymbol{\theta})$ . Further research aimed at developing an absolute metric, particularly for the epistemic uncertainty, could prove useful in enhancing the effectiveness of BNN methods. This could extend their use beyond serving as an relative uncertainty metric to also functioning as a more comprehensive metric of trustworthiness, particularly in safety-critical and military systems.

As shown, the methods utilized in this work effectively notify the user when a new qualify unseen input is being observed, while failing in detecting domain shifts. A more comprehensive study of how the model acts in the presence of a domain shift is needed to avoid the risk of overconfidence as the model may fail to detect it. This can mislead an operator into trusting the prediction in the presence of an error. Despite that possibility, BNNs represent a significant advancement by offering estimates of uncertainty that regular deterministic NN methods cannot provide. With the guidance of experienced individuals knowledgeable about the environment, these estimates can serve as a rough metric of trustworthiness. Although the usage of BNNs may require additional effort and resources, having some indication of the model's reliability is better than having none at all.

Finally, the proposed method of using physical models as priors was shown to be a useful technique for achieving more performant models. The general concept is logical, especially when there is insufficient data available, as predicting based on the behavior of the physical

model is a reasonable approach in this case. These results pave the way for future development of new radio frequency source localization techniques for simple single antenna Software Defined Radios, with applications in Signals Intelligence organizations, Maritime Search and Rescue Operations, and other domains where the use of complex receivers is not practical.

---

## APPENDIX A:

### Neural Network Architecture

---

#### A.1 Deterministic and Bayesian DNN Architectures

Although several different DNN architectures were developed and tested for this thesis, only the best performing architecture, based on validation data, is shown in Chapter 3. Table A.1 contains the structure of the deterministic NN architecture, while Table A.2 presents its Bayesian counterpart. The main difference between the two is the ability of the BNN to represent Gaussian distributions over the parameters. As expected, the Bayesian version has double the number of parameters compared to the deterministic version.

Table A.1. Deterministic DNN architecture.

| Name                                  | Type        | Activation | Input Features | Output Features | Num. Params |
|---------------------------------------|-------------|------------|----------------|-----------------|-------------|
| input                                 | BatchNorm1d | None       | 1024           | 1024            | 2,048       |
| layer_1                               | Linear      | ReLU       | 1024           | 512             | 524,800     |
| layer_2                               | Linear      | ReLU       | 512            | 512             | 262,656     |
| layer_3                               | Linear      | ReLU       | 512            | 512             | 262,656     |
| layer_4                               | Linear      | ReLU       | 512            | 512             | 262,656     |
| layer_5                               | Linear      | ReLU       | 512            | 512             | 262,656     |
| layer_6                               | Linear      | ReLU       | 512            | 512             | 262,656     |
| layer_7                               | Linear      | ReLU       | 512            | 512             | 262,656     |
| layer_8                               | Linear      | ReLU       | 512            | 512             | 262,656     |
| layer_9                               | Linear      | ReLU       | 512            | 512             | 262,656     |
| mean                                  | Linear      | None       | 512            | 1               | 513         |
| rho                                   | Linear      | Softplus   | 512            | 1               | 513         |
| Total number of parameters: 2,629,122 |             |            |                |                 |             |

Table A.2. BNN architecture.

| Name                                  | Type        | Activation | Input Features | Output Features | Num. Params |
|---------------------------------------|-------------|------------|----------------|-----------------|-------------|
| input                                 | BatchNorm1d | None       | 1024           | 1024            | 2,048       |
| layer_1                               | LinearLRT   | ReLU       | 1024           | 512             | 1,049,600   |
| layer_2                               | LinearLRT   | ReLU       | 512            | 512             | 525,312     |
| layer_3                               | LinearLRT   | ReLU       | 512            | 512             | 525,312     |
| layer_4                               | LinearLRT   | ReLU       | 512            | 512             | 525,312     |
| layer_5                               | LinearLRT   | ReLU       | 512            | 512             | 525,312     |
| layer_6                               | LinearLRT   | ReLU       | 512            | 512             | 525,312     |
| layer_7                               | LinearLRT   | ReLU       | 512            | 512             | 525,312     |
| layer_8                               | LinearLRT   | ReLU       | 512            | 512             | 525,312     |
| layer_9                               | LinearLRT   | ReLU       | 512            | 512             | 525,312     |
| mean                                  | LinearLRT   | None       | 512            | 1               | 1026        |
| rho                                   | LinearLRT   | Softplus   | 512            | 1               | 1026        |
| Total number of parameters: 5,256,196 |             |            |                |                 |             |

The complete architecture, including the data preprocessing pipelines, was named “Welch DNN” to distinguish it from other tested data preprocessing strategies. The name originates from the utilization of a Welch PSD representation of the signals as inputs for the DNN. More information about the specific data preprocessing strategies can be found in Appendix Section A.2.

One feasible approach is to incorporate information about the transmitter by appending a perfectly reconstructed version of the transmitted waveform, which is noiseless and undisturbed, to the received signal. This approach enables the inclusion of meaningful information that can be used by the NN models to differentiate between transmitters and environmental conditions. In this work, it is assumed that the transmitted signals are known. This assumption is reasonable since basic information about the transmitter is necessary for any RSS localization method. Additionally, an idealized version (not necessarily an exact representation) of the transmitted signal can be generated from the received demodulated signal.

It should be noted that while this approach requires demodulation of the received signal to make a prediction, the same architecture was successfully employed to predict distances for transmitters using a fixed output power, without reconstructing the transmitted signal. This approach yielded even better performance results than those presented in Chapter 3. However, demonstrating a more versatile model that can still outperform the physical model is of greater interest.

## **A.2 Data Preprocessing, Model Capacity, and Dynamic Range Related Issues**

During this thesis, multiple approaches were tested to represent the input data. Complex values, as well as real values in the time domain representation of the waveform, did not yield the expected results, as depicted in Figure A.1.a. Although using time domain data showed some learning ability in the model, several artifacts appeared across all distances.

A better model is obtained by utilizing the Welch PSD representation with a linear scale for the signals. However, the obtained model lacked the necessary dynamic range to effectively capture the significant amplitude differences for received signals at various ranges. This limitation becomes apparent when observing Figure A.1.b, where the predictions remain similar for longer distances. Additionally, similar artifacts to those seen in the time domain representation are present at shorter distances.

The observed issues are resolved by implementing a logarithmic scale. Figure A.1.c illustrates the impact of using a modified version of the Welch architecture with half the number of layers. In this case, the model exhibits a reasonable behavior, indicating its ability to learn. However, it still lacks the necessary capacity to generate accurate predictions.

Lastly, Figure A.1.d demonstrates how the architecture, which employs a logarithmic scale and preprocesses the data using the Welch PSD representation, resolves the majority of the identified issues. Nevertheless, there is a slight curvature at the end of the predictions for larger scales that could not be eliminated by increasing the network's capacity. This curvature is likely due to the similarity of the data at longer distances, which the logarithmic scale alone is insufficient to overcome, as mentioned in Section 3.2.

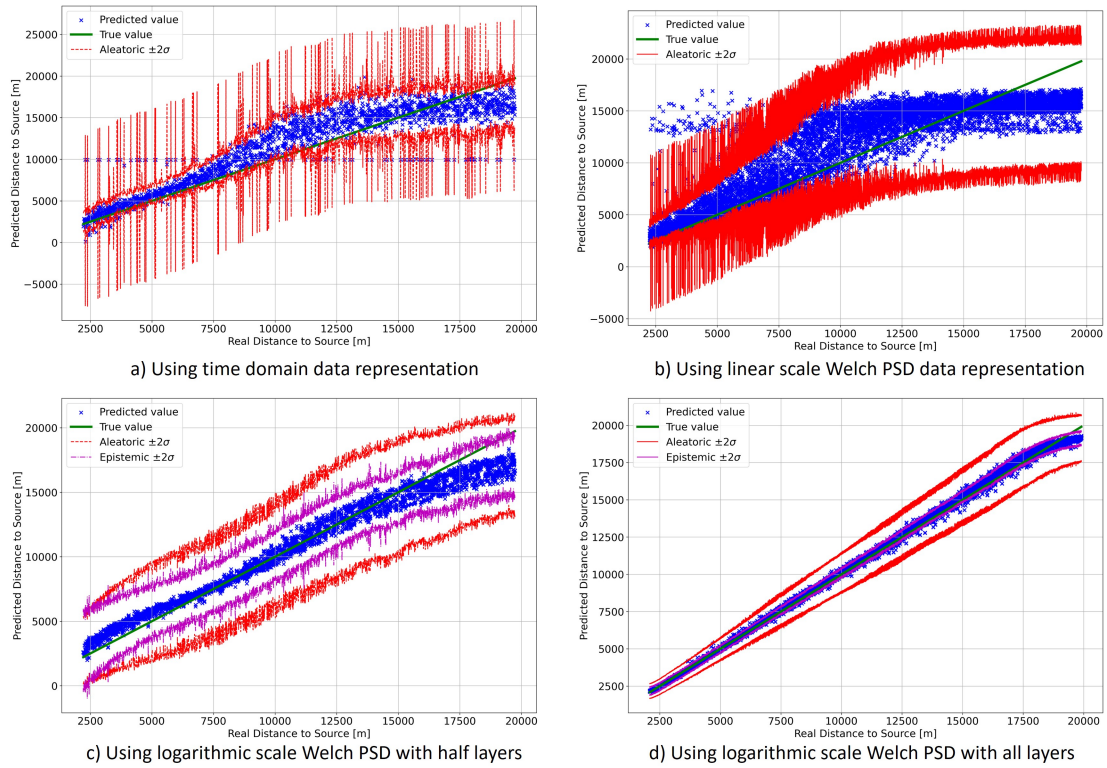


Figure A.1. Illustration of the effects of feature engineering and architecture.

---

## APPENDIX B:

### Parameters for Simulated RF Environments

---

Table B.1. Parameters used to generate the simulated environments.

| <i>Parameter</i>           | <i>Ideal</i> | <i>Environments</i> |              |
|----------------------------|--------------|---------------------|--------------|
|                            |              | <i>Standard</i>     | <i>Harsh</i> |
| Training examples          | 50,000       | 50,000              | 50,000       |
| Testing ratio              | 0.2          | 0.2                 | 0.2          |
| Validation ratio           | 0.2          | 0.2                 | 0.2          |
| Sample rate                | 24,000       | 24,000              | 24,000       |
| ADC Resolution             | 32–bits      | 12–bits             | 12–bits      |
| ADC Max Power              | 0.0 dBm      | 0.0 dBm             | 0.0 dBm      |
| Receiver Noise Figure      | 0.0 dB       | 0.9 dB              | 2.0 dB       |
| Receiver Gain              | 10 dB        | 10 dB               | 10 dB        |
| Receiver Antenna Height    | 10 m         | 10 m                | 10 m         |
| Receiver Antenna Gain      | 5.21 dBi     | 5.21 dBi            | 5.21 dBi     |
| Max. Frequency Offset      | 0 Hz         | 20 Hz               | 100 Hz       |
| Transmitter Power          | 50 W         | 10, 25 W            | 10, 25, 50 W |
| Transmitter Frequency      | 140 MHz      | 140 MHz             | 140 MHz      |
| Transmitter Antenna Height | 10 m         | 10 m                | 10 m         |
| Transmitter Antenna Gain   | 5.21 dBi     | 5.21 dBi            | 5.21 dBi     |
| Min. Packet Size           | 100 Symbols  | 100 Symbols         | 100 Symbols  |
| Max. Packet Size           | 300 Symbols  | 300 Symbols         | 300 Symbols  |
| Modulations                | M–PSK        | M–PSK               | M–PSK        |
| Number of Symbols (M)      | 4, 8, 16     | 4, 8, 16            | 4, 8, 16     |
| Beaufort Wind Scale        | 0            | 4                   | 7            |
| Number of Scatterers       | 0            | 4                   | 40           |
| Max. Multipath Amplitude   | –            | 0.6                 | 0.8          |
| Max. Multipath Delay       | –            | 100 Samples         | 500 Samples  |
| Min. Simulation Distance   | 2,000 m      | 2,000 m             | 2,000 m      |
| Max. Simulation Distance   | 20,000 m     | 20,000 m            | 20,000 m     |
| Min. Unseen Data Distance  | 500 m        | 500 m               | 500 m        |
| Max. Unseen Data Distance  | 25,000 m     | 25,000 m            | 25,000,      |
| Pathloss Model             | Two-ray      | Two-ray             | Two-ray      |

THIS PAGE INTENTIONALLY LEFT BLANK

---

## List of References

---

- [1] A. Parameswaran, M. Iftekhar Husain, and S. Upadhyaya, “Is RSSI a reliable parameter in sensor localization algorithms: An experimental study,” Jan. 2009. Available: [https://cse.buffalo.edu/srds2009/F2DA/f2da09\\_RSSI\\_Parameswaran.pdf](https://cse.buffalo.edu/srds2009/F2DA/f2da09_RSSI_Parameswaran.pdf)
- [2] C. Gentile, N. Alsindi, R. Raulefs, and C. Teolis, *Geolocation Techniques: Principles and Applications*. New York, NY, USA: Springer Science & Business Media, Nov. 2012, ch. Ranging and Localization in Harsh Multipath Environments, pp. 17–57.
- [3] X. Li, “RSS-based location estimation with unknown pathloss model,” *IEEE Transactions on Wireless Communications*, vol. 5, no. 12, pp. 3626–3633, Dec. 2006. Available: <https://doi.org/10.1109/TWC.2006.256985>
- [4] D. Dogan, Y. Dalveren, and A. Kara, “A mini-review on radio frequency fingerprinting localization in outdoor environments: Recent advances and challenges,” in *2022 14th International Conference on Communications (COMM)*, June 2022, pp. 1–5. Available: <https://doi.org/10.1109/COMM54429.2022.9817189>
- [5] S. R. Mattu, L. N. Theagarajan, and A. Chockalingam, “Deep channel prediction: A DNN framework for receiver design in time-varying fading channels,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 6439–6453, June 2022. Available: <https://doi.org/10.1109/TVT.2022.3162887>
- [6] N. Abbas, Y. Nasser, and K. E. Ahmad, “Recent advances on artificial intelligence and learning techniques in cognitive radio networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, p. 174, June 2015. Available: <https://doi.org/10.1186/s13638-015-0381-7>
- [7] *VHF Marine Transceiver IC-M200*, Icom. Available: <https://www.icomjapan.com/support/manual/1594/>
- [8] SigMF, *The Signal Metadata Format (SigMF)*, ver. 1.0.0, <https://github.com/sigmf/SigMF/>, 2023. Available: <https://github.com/sigmf/SigMF/blob/sigmf-v1.x/sigmf-spec.md>
- [9] T. J. O’Shea, T. Roy, and T. C. Clancy, “Over-the-air deep learning based radio signal classification,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, Feb. 2018. Available: <https://doi.org/10.1109/JSTSP.2018.2797022>

- [10] K. Emam and a. O. M. C. Safari, *Accelerating AI with Synthetic Data: Generating Data for AI Projects*. Sebastopol, CA, USA: O'Reilly Media, Incorporated, 2020.
- [11] M. Hinne, Q. F. Gronau, D. van den Bergh, and E.-J. Wagenmakers, "A conceptual introduction to bayesian model averaging," *Advances in Methods and Practices in Psychological Science*, vol. 3, no. 2, pp. 200–215, 2020. Available: <https://doi.org/10.1177/2515245919898657>
- [12] E. McCune, *Practical Digital Wireless Signals* (The Cambridge RF and Microwave Engineering Series). New York, NY, USA: Cambridge University Press, 2010, ch. Principles of Coding, pp. 270–298.
- [13] J. G. Proakis and M. Salehi, *Digital Communications*. New York, NY, USA: McGraw-Hill, Sep. 2008, ch. Deterministic and Random Signal Analysis, pp. 17–82.
- [14] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, Apr. 2016. Available: <http://dx.doi.org/10.1007/s13748-016-0094-0>
- [15] "Beaufort wind scale," NOAA's National Weather, accessed: Dec. 23, 2022. Available: <https://www.wpc.ncep.noaa.gov/html/beaufort.shtml>
- [16] "The polarization loss factor in antenna communication," Cadence System Analysis, accessed: Apr. 17, 2023. Available: <https://resources.system-analysis.cadence.com/blog/msa2021-the-polarization-loss-factor-in-antenna-communication>
- [17] R. Svitek and S. Raman, "Dc offsets in direct-conversion receivers: Characterization and implications," *IEEE Microwave Magazine*, vol. 6, no. 3, pp. 76–86, 2005. Available: <https://doi.org/10.1109/MMW.2005.1511916>
- [18] T. F. Collins, R. Getz, A. M. Wyglinski, and D. Pu, *Software-Defined Radio for Engineers*. Boston, MA, USA: Artech House, Apr. 2018, ch. Sampling Theory, pp. 23–37.
- [19] J. L. Volakis and J. T. Johnson, *Antenna Engineering Handbook*. New York, NY, USA: McGraw-Hill Education, 2007, ch. Propagation, pp. 11–19.
- [20] T. S. Rappaport, *Wireless Communications : Principles and Practice*, 2nd ed. (Prentice Hall communications engineering and emerging technologies series). Upper Saddle River, NJ, USA: Prentice Hall PTR, 2002, ch. Mobile Radio Propagation: Large-Scale Path Loss, pp. 69–138.

- [21] J. Wang, H. Zhou, Y. Li, Q. Sun, Y. Wu, S. Jin, T. Q. S. Quek, and C. Xu, “Wireless channel models for maritime communications,” *IEEE access*, vol. 6, pp. 68 070–68 088, 2018. Available: <https://doi.org/10.1109/ACCESS.2018.2879902>
- [22] Y. H. Lee, F. Dong, and Y. S. Meng, “Near sea-surface mobile radiowave propagation at 5 ghz: Measurements and modeling,” *Radioengineering*, vol. 23, no. 3, pp. 824–830, 2014.
- [23] P. Horowitz and W. Hill, *The Art of Electronics*. New York, NY, USA: Cambridge University Press, Mar. 2015, ch. Low-Noise Techniques, pp. 473 – 593.
- [24] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Upper Saddle River, NJ, USA: Pearson, Jan. 2010, ch. Sampling of Continuous-Time Signals, pp. 153–273.
- [25] N. Patwari, A. Hero, M. Perkins, N. Correal, and R. O’Dea, “Relative location estimation in wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003, conference Name: IEEE Transactions on Signal Processing. Available: <https://doi.org/10.1109/TSP.2003.814469>
- [26] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for Machine Learning*. New York, NY, USA: Cambridge University Press, Apr. 2020.
- [27] L. Bottou, “Online algorithms and stochastic approximations,” in *Online Learning and Neural Networks*, D. Saad, Ed. Cambridge, UK: Cambridge University Press, 1998, revised, Oct. 2012. Available: <http://leon.bottou.org/papers/bottou-98x>
- [28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, Nov. 2016.
- [29] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv, 2017. Available: <https://arxiv.org/abs/1412.6980>
- [30] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” arXiv, 2015. Available: <https://arxiv.org/abs/1502.01852>
- [31] A. D. Kiureghian and O. Ditlevsen, “Aleatory or epistemic? Does it matter?” *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009. Available: <https://doi.org/10.1016/j.strusafe.2008.06.020>
- [32] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight uncertainty in neural networks,” arXiv, 2015. Available: <https://doi.org/10.48550/arXiv.1505.05424>

- [33] E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig, “Laplace redux – effortless bayesian deep learning,” arXiv, 2022. Available: <https://doi.org/10.48550/arXiv.2106.14806>
- [34] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” arXiv, 2016. Available: <https://doi.org/10.48550/arXiv.1506.02142>
- [35] P. McClure, N. Rho, J. A. Lee, J. R. Kaczmarzyk, C. Y. Zheng, S. S. Ghosh, D. M. Nielson, A. G. Thomas, P. Bandettini, and F. Pereira, “Knowing what you know in brain segmentation using Bayesian deep neural networks,” *Frontiers in Neuroinformatics*, vol. 13, Oct. 2019. Available: <https://doi.org/10.3389/fninf.2019.00067>
- [36] K. P. Murphy, *Probabilistic Machine Learning: Advanced Topics*. Cambridge, MA, USA: MIT Press, Aug. 2023, vol. 1, ch. Information Theory, pp. 249–263.
- [37] A. Graves, “Practical variational inference for neural networks,” in *Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS’11)*. Red Hook, NY, USA: Curran Associates Inc., 2011, p. 2348–2356. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2011/file/7eb3c8be3d411e8ebfab08eba5f49632-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2011/file/7eb3c8be3d411e8ebfab08eba5f49632-Paper.pdf)
- [38] D. P. Kingma, T. Salimans, and M. Welling, “Variational dropout and the local reparameterization trick,” arXiv, 2015. Available: <https://arxiv.org/abs/1506.02557>
- [39] J. Kirkpatrick and et al., “Overcoming catastrophic forgetting in neural networks,” *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017. Available: <https://doi.org/10.1073/pnas.1611835114>
- [40] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, “Variational continual learning,” arXiv, 2018. Available: <https://doi.org/10.48550/arXiv.1710.10628>
- [41] R. Krishnan, P. Esposito, and M. Subedar, *Bayesian-Torch: Bayesian neural network layers for uncertainty estimation*, <https://github.com/IntelLabs/bayesian-torch>, Jan. 2022. Available: <https://doi.org/10.5281/zenodo.5908307>
- [42] S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft, “Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning,” arXiv, 2018. Available: <https://doi.org/10.48550/arXiv.1710.07283>
- [43] W. L. Stutzman, *Antenna Theory and Design*. Hoboken, NJ, USA: Wiley, 2012, ch. Antenna Fundamentals, pp. 23–69.

- [44] “Application note: The link budget and fade margin,” Campbell Scientific, Inc., Tech. Rep., 2016. Available: [https://s.campbellsci.com/documents/eu/technical-papers/link-budget\\_9-16.pdf](https://s.campbellsci.com/documents/eu/technical-papers/link-budget_9-16.pdf)
- [45] C. C. J. Haslett, *Essentials of Radio Wave Propagation* (The Cambridge wireless essentials series). New York, NY, USA: Cambridge University Press, 2008, ch. Atmospheric effects, pp. 116–149.
- [46] T. G. J. Rudner, Z. Chen, Y. W. Teh, and Y. Gal, “Tractable function-space variational inference in bayesian neural networks,” in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds. Curran Associates, Inc., 2022, vol. 35, pp. 22 686–22 698. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/8ea50bf458f6070548b11babbe0bf89b-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/8ea50bf458f6070548b11babbe0bf89b-Paper-Conference.pdf)
- [47] F. Huszár, “Variational inference using implicit distributions,” arXiv, 2017. Available: <https://doi.org/10.48550/arXiv.1702.08235>

THIS PAGE INTENTIONALLY LEFT BLANK

---

## Initial Distribution List

---

1. Defense Technical Information Center  
Ft. Belvoir, Virginia
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, California



## DUDLEY KNOX LIBRARY

NAVAL POSTGRADUATE SCHOOL

[WWW.NPS.EDU](http://WWW.NPS.EDU)

---

WHERE SCIENCE MEETS THE ART OF WARFARE