Deep Learning for Automatic Picking of Dispersion Curves

MASTER OF SCIENCE THESIS

for the degree of Master of Science in Applied Geophysics
by

Kilian Glatz

August 2, 2024

IDEA LEAGUE
JOINT MASTER'S IN APPLIED GEOPHYSICS

Delft University of Technology, The Netherlands
ETH Zürich, Switzerland
RWTH Aachen, Germany

Dated: *August 2, 2024*

Supervisors:

Assoc. Prof. Dr. Deyan S. Draganov

Dr. Florencia Balestrini

Committee Members:

Assoc. Prof. Dr. Deyan S. Draganov

Prof. Dr. Florian M. Wagner

Asst. Prof. Dr. Katrin Löer

Abstract

Multichannel Analysis of Surface Waves (MASW) is commonly used to determine the shallow subsurface velocity structure. The obtained velocities provide valuable information for foundation design and other geotechnical applications. The conventional method of picking surface-wave dispersion curves, which are subsequently inverted to obtain the desired velocity profiles, can be a labour-intensive task. In this thesis, we make use of recent advances in Deep Learning (DL), utilizing Convolutional Neural Networks (CNNs) to estimate dispersion curves directly from shot gathers or their corresponding dispersion spectra. For this, we first test the proposed approach on various synthetic data, successfully addressing challenges such as geological models with velocity inversion, different noise levels, and the estimation of the first higher mode. When applied to field data, our results indicate that training the CNN on dispersion spectra provides more accurate and reliable predictions compared to training on shot gathers directly. We improve the predictions using shot gathers as training data by modifying the original CNN architecture and applying transfer learning techniques. The enhanced CNN models demonstrate significant improvements, indicating that this approach could aid the analysis of MASW data. To fully realize the potential of this method, future efforts should focus on increasing the similarity between synthetic and field data, for example by incorporating realistic noise.

Acknowledgments

First of all, I want to thank my supervisors Florencia Balestrini and Deyan Draganov. Florencia for her continious and super fast help whenever I needed something or had questions throughout the project. Deyan for his constructive feedback and suggestions, and for providing me with the opportunity to attend the EAGE conference in Oslo to discuss how young talent should be nurtured in geosciences in light of climate change.
I also want to thank Danilo Chamorro Riascos for supplying me with the code of his work, building the foundation of the work presented in this thesis.
Special thanks to my parents, who always believe in me and without whom I would not be who and where I am today.
Lastly, I would like to thank Romy, who supported me throughout all highs and lows, endured me during every examination phase, and made the past two years fly by.
Thank you.

Kilian Glatz
Delft, August 2024

Contents

List of Figures

List of Tables

1 Introduction

With the continuous development of technology and the increasing amounts of data being gathered across various fields, traditional approaches for data analysis are struggling to keep up. Conventional statistical methods and manual data analysis are often insufficient for managing and interpreting the vast datasets generated. Consequently, new methods must be developed to manage and analyze this data efficiently. Fortunately, ongoing advances in computational resources, such as advances in Graphical Processing Units (GPUs), have enabled us to make use of these large amounts of data through efficient computational approaches.

In the current decade, Artificial Intelligence (AI) and its subfields, Machine Learning (ML) and Deep Learning (DL), are playing an ever-increasing role in society, as they leverage the advances in computational resources. Large Language Models (LLMs), like GPT and GitHub Copilot, are revolutionizing the way in which students gather information, write assignments, and code. People across different fields use these models for writing emails, drafting applications, brainstorming ideas, and translating texts into various languages. Moreover, images, almost indistinguishable from real art or photography, can now be generated within seconds using simple prompts, opening a new field of work: prompt engineering.

However, there are also downsides accompanying these new technologies. Political and regulatory frameworks are struggling to keep up, particularly in ensuring data protection and preventing the misuse of these technologies. The spread of fake news is becoming more prevalent, posing a significant threat to society if people are not properly educated on how to identify it. Furthermore, there is a risk that people might start using the outputs of LLMs without critically evaluating them, potentially leading to erroneous results and conclusions. This is particularly concerning in science, where comprehension and verification are crucial to maintaining integrity.

Altough the aforementioned political issues require attention, they fall outside the scope of this thesis and just serve as a reminder that every technological advance has its drawbacks. Instead, this thesis focuses on leveraging the use of DL for geophysical purposes. More precisely, we will apply DL for the automatic picking of dispersion curves. While DL is considered a black-box (Prieto et al., 2016; Choi et al., 2020; James et al., 2023) we will analyze different scenarios to get more insight on how our DL model is influenced, providing a deeper look at the processes involved and uncover possible limitations.

Dispersion curves are a phenomenon related to surface waves, which are seismic waves that propagate along the Earth's surface and surfaces in general. Surface waves are extensively used for subsurface characterization, and in particular for near-surface site characterization. The penetration depth of surface waves depends on their frequency, leading to different frequencies propagating at different velocities in a vertically heterogeneous medium. This property is known as geometric dispersion and enables the extraction of information about the subsurface structure from surface-wave recording. Multichannel Analysis of Surface Waves (MASW) (Park et al., 1999) is a commonly used technique to characterize the geometric dispersion of surface waves, intending to obtain the velocity profile of a certain area, crucial for applications like soil-structure interaction, earthquake site response, or nondestructive pavement testing (Foti, 2015). MASW makes use of multiple receivers that record the wavefield produced by a seismic source. Traditional processing of MASW data involves transforming the acquired shot gathers into domains where the frequency-dependent phase velocity of the surface waves, or in other words, their dispersion characteristics, can be determined either manually or semi-automatically.

The main objective of this thesis is to evaluate the capabilities of a DL approach to automate the extraction of dispersion curves directly from the shot gathers or the corresponding dispersion spectra. For this, we will adress the following questions:

- How does a DL approach compare to the conventional approach?
- What are the influences of different choices of data domains (namely, shot gathers and dispersion spectra) for training a DL model?
- How well does a DL approach trained on synthetic data generalize to real field data?
- What are the underlying assumptions and limitations of the DL approach?
- What can be improved about the current approach and what are our recommendations for future directions to automate the picking of dispersion curves?

To answer these questions, we first review the necessary theory of surface waves and DL in Chapter 2. In the following methodology section (Chapter 3), the reasoning behind our approach and the tools used to tackle the problem are presented in detail. In Chapter 4, we showcase the results of our experiments and subsequently discuss them in Chapter 5. Finally, the outcome of this thesis is concluded and recommendations on future research directions are given in Chapter 6.

Prior to this thesis, a comprehensive literature review was conducted to establish a solid foundation in MASW and to obtain an overview of existing ML methods for the automatic picking of dispersion curves. The outcome of this work is presented in Appendix A.4, which provides an overview of the fundamental theory and primary challenges associated with MASW, some of which will not be reiterated in this thesis. Consequently, readers unfamiliar with the basics of MASW are advised to refer to Appendix A.4 first for essential background information.

2 Theoretical Background

2.1 Surface Waves

The theory of surface waves has been extensively discussed by Aki and Richards (1980), Foti (2015), and Shearer (2019), to which the reader is referred to for detailed derivations. In this section, the most important concepts of surface-wave propagation and the surface-wave processing used in this thesis are summarized.

Details on dispersion-curve inversion are not provided in this section, as this thesis solely focuses on the extraction of dispersion curves, which can subsequently be used in different inversion algorithms. These inversion algorithms typically assume a laterally homogenous, layered subsurface. Thus, the following theory on surface waves will be limited to this special case.

Furthermore, while the term surface wave encompasses different types of waves, such as Rayleigh waves, Love waves, and Scholte waves, this thesis is concerned exclusively with Rayleigh waves. Therefore, the terms surface wave and Rayleigh wave are used interchangeably.

2.1.1 Surface-Wave Propagation

Surface waves are a type of seismic wave that propagates along the Earth's surface with a limited penetration depth, unlike body waves, which propagate through the Earth's interior. The phase velocities, group velocities, and attenuation characteristics of surface waves have proven useful in delineating the structure of the crust and upper mantle in various regions of the Earth (Socco et al., 2010). On a more localized scale, seismic surface waves and the analysis of their properties serve as a powerful tool for near-surface characterization in geophysical exploration.

Surface waves arise from the interaction of elastic waves with the free surface. They are composed of a linear combination of longitudinal and vertically polarized transverse particle motions, whose amplitude decays with depth within a medium (Socco and Strobbia, 2004; Shearer, 2019). In a three-dimensional medium, surface waves propagate cylindrically outward from the source location. Due to the restriction of energy in the vertical dimension, surface waves exhibit a geometric decrease in amplitude proportional to $\frac{1}{\sqrt{r}}$, where r is the distance from the source point. This amplitude decrease is significantly lower than the three-dimensional $\frac{1}{r}$ decay rate for body waves (Aki and Richards, 1980). As a result, Rayleigh-wave amplitudes tend to be much larger than body waves on seismograms (Lay and Wallace, 1995).

In addition to the amplitude decay as a function of source distance, the energy of the surface waves diminishes with depth. Mathematically, this can be expressed as:

$$A(z) = A_0 \cdot e^{-\alpha z}, \tag{2.1}$$

where $A(z)$ is the amplitude at depth z, A_0 is the amplitude at the surface, and α is an attenuation coefficient. This attenuation coefficient depends on the material properties of the subsurface. In softer materials, the dissipation of energy is typically more pronounced.

The propagation of the surface waves is, therefore, affected only by the properties of a limited section of the subsurface, with the thickness depending on the propagating wavelengths. Hence, in the same medium, waves of different wavelengths are affected by different depths. If the medium is vertically inhomogeneous they propagate with different velocities and attenuations. As a result, the propagation velocity can be strongly frequency-dependent, according to the geometric distribution of the soil properties. Low-frequency waves penetrate deeper and are influenced by the properties of the deeper layers, whereas high-frequency waves are more sensitive to the near-surface layers. This behavior is called geometric dispersion, as opposed to intrinsic dispersion, which depends on the inherent attenuation due to the subsurface material (Socco and Strobbia, 2004).

2.1.2 Dispersion-Curve Calculation

As previously described, surface waves show characteristic properties related to their propagation in the subsurface. Another property is the existence of multiple modes. In vertically inhomogeneous elastic media, the dispersion relation can implicitly be written as

$$\Phi_R[\lambda(z), \mu(z), \rho(z), k_j(\omega), \omega] = 0, \tag{2.2}$$

where λ and μ are the Lame parameters, ρ the density, k the wavenumber, and ω the angular frequency (Foti, 2015). The above equation, also known as the Rayleigh secular (or dispersion) equation, indicates that the propagation velocity is a multivalued function of angular frequency ω_j. In contrast to the fundamental mode, which exists for all frequencies, higher modes exist only above a certain cut-off frequency (Socco and Strobbia, 2004; Foti, 2015).

Multiple methods exist for solving Equation 2.2. In this thesis, we make use of the *Dunkin's matrix* algorithm (Dunkin, 1965) implemented via the Python library *disba* (Luu, 2021). This algorithm offers numerical improvements for high frequencies compared to the original transfer matrix method introduced by Thomson (1950) and Haskell (1953). The underlying assumption is a medium that consists of a stack of laterally homogenous, isotropic, elastic layers (Dunkin, 1965). A matrix is constructed for each layer based on the corresponding material properties, and continuity of stress and displacement fields are imposed at each layer interface (Foti, 2015). Through a sequence of matrix multiplications the dispersion relation (Equation 2.2) is constructed where the roots of the matrix determinant can be found, providing the modal dispersion curves (Dunkin, 1965).

2.1.3 Dispersion Analysis

The process of determining the dispersion curves from seismic data is known as dispersion analysis. As discussed in the previous subsection, higher modes can theoretically be computed, but their identification in seismic data depends on their energy distribution in the analyzed domain.

Various transformation techniques can be applied to space-time ($x - t$) seismic data, which allows the identification of dispersion curves in different domains. The basis for this analysis is the fact that surface waves exhibit dominant energy in seismic records. Common transformation techniques include the f-k-transformation (Nolet and Panza, 1976), the phase-shift (PS) method (Park et al., 1998), and frequency-domain beamforming (FDBF) (Zywicki and Rix, 2005). A comparative study on the advantages and disadvantages of different transformation techniques in various scenarios was carried out by Rahimi et al. (2021). They conclude that in most cases the FDBF performed best. In this thesis, however, the PS transform from Park et al. (1998) is implemented and the following derivations follow their work. The PS transform was chosen because of its simple implementation and because it was used in the previous processing of the field data later shown in this thesis.

For the PS method, the seismic data $u(x,t)$ is transformed to the frequency domain using the Fourier transform:

$$U(x,\omega) = \int u(x,t) \cdot e^{i\omega t} dt. \tag{2.3}$$

Equation 2.3 can be separated into amplitude and phase components:

$$U(x,\omega) = P(x,\omega) A(x,\omega), \tag{2.4}$$

where $P(x,\omega)$ and $A(x,\omega)$ are the phase and amplitude spectrum, respectively. The phase spectrum, noting that it contains the dispersion information, can be written as

$$P(x,\omega) = e^{-i\frac{\omega}{c_\omega}x} = e^{-ik_\omega x}, \tag{2.5}$$

where c_ω is the phase velocity and k_ω the wavenumber.

Combining Equations 2.5 and 2.4 and applying the operator $e^{i\omega\frac{x}{v_\omega}}$ to the integral $\int \frac{U(x,\omega}{|U(x,\omega)|}$ we obtain the dispersion image $V(\omega,v)$ (also referred to as $f - v$ spectra) as:

$$V(\omega,v) = \int e^{i\omega\frac{x}{v_\omega}} \frac{U(x,\omega)}{|U(x,\omega)|} dx = \int e^{i\omega(\frac{1}{v_\omega} - \frac{1}{c_\omega})x} \frac{A(x,\omega)}{|A(x,\omega)|} dx. \tag{2.6}$$

In the case that v_ω is equal to the phase velocity of the signal c_ω, the integral will have a maximum, allowing to identify the dispersion curve. Note that in Equation 2.6 the amplitude spectrum for each offset is normalized to compensate for attenuation.

In ideal conditions with noise-free seismic data and a dominant fundamental mode, the process of picking the dispersion curve by determining the maximum for each frequency in the dispersion image $V(\omega, v)$ is straightforward. However, in practice, noise and limited array length as well as higher modes with dominant energy leading to apparent dispersion curves, make the picking procedure cumbersome. Therefore, experienced operators are required to identify the correct modes to obtain reliable results.

2.2 Machine Learning

In this section, a short introduction is given to the ML techniques used in this thesis. While ML is a broad field, a rough division can be made into two branches: supervised and unsupervised ML techniques (Bishop, 2006). In supervised learning, we are given some dataset $\mathbf{D} = \{d_1, d_2, ..., d_n\}$ and a related response $\mathbf{y}$ (which can have any format, e.g., scalar, vector, or tensor), which is called the label. Supervised methods aim to learn the relationship between the data and the label to then make inferences on new datasets where the label might be unknown.

Unsupervised learning refers to techniques that identify patterns or relationships in the data $\mathbf{D}$, without the guidance of labels. One popular class of unsupervised methods is clustering algorithms, which divide the dataset into clusters based on a priori-defined types of similarity.

In this thesis, we will focus on supervised ML techniques, more specifically on convolutional neural networks (CNN), belonging to the subfield of DL. In the following subsections, the fundamentals behind neural networks (NN) and their extension to CNN will be explained to build the foundation for the methods utilized in this work.

NN form the cornerstone of DL, one of the most active areas of research at current times (James et al., 2023). The fundamental idea behind NN dates back to the attempt to create algorithms that are able to mimic the functioning of the human brain (Macukow, 2016). While a lot of progress has (already) been made during the 21st century, extending our understanding of the human brain (Lisman, 2015), we are still far away from mimicking it using algorithms. Nonetheless, NN has risen in popularity during the last decade and has proven to be useful in all different kinds of fields. In the following sections, we will take a look at the basic structure of a NN, explain the different components, and show how NNs are trained to solve different problems.

2.2.1 Basic Structure of Neural Networks

A simple NN can be broken down into the following components: neurons, layers, and activation functions. The basic unit of a neural network is the neuron. Multiple neurons are typically sorted into layers, which are concatenated together to form the NN (see Figure 2.1a). The first layer is referred to as the input layer as it receives the raw input data. It is followed by the hidden layers where the input is transformed and finally passed to the output layer. Each neuron has a bias and as many weights as connections to neurons of the previous layer. Given an input from the neurons of the previous layer, a neuron's output $\hat{y}$ can be described as the linear combination of weights w_i and inputs x_i and an additional bias b. Finally, a transformation in the form of the activation function f is applied to this sum resulting in the output which is passed as input to the neuron(s) in the next layer. For a single neuron, this can be expressed as

$$\hat{y} = f\left(b + \sum_{i=1}^{N} w_i \cdot x_i\right), \tag{2.7}$$

which can be rewritten for a layer, consisting of multiple neurons, in vectorial form as

$$\hat{\mathbf{y}} = f\left(\mathbf{W}^{\mathbf{T}}\mathbf{x} + \mathbf{b}\right), \tag{2.8}$$

and by defining $\mathbf{z} = \mathbf{W}^{\mathbf{T}}\mathbf{x} + \mathbf{b}$, where $\mathbf{W}$ is the weights matrix, $\mathbf{x}$ the input vector, and $\mathbf{b}$ the bias vector, as

$$\hat{\mathbf{y}} = f(\mathbf{z}). \tag{2.9}$$

Figure 2.1 b) schematically shows this basic unit of a neural network.

While the weights and biases are determined during the training process (see Section 2.2.2), different activation functions exist and must be chosen and combined according to the problem at hand. Activation functions play a crucial role as they introduce the necessary non-linearity to an NN (Rasamoelina et al., 2020; Dubey et al., 2022). There are several activation functions, but some examples of the most common activation functions include the sigmoid function, the linear function (equivalent to no activation function), and the Rectified Linear Unit (ReLU) function (and versions thereof), which are displayed in Figure 2.2. The sigmoid function

$$f(\mathbf{z})_{sigmoid} = \frac{1}{1 + e^{-z}}, \tag{2.10}$$

commonly used in logistic regression, transforms a linear function into values between zero and one (James et al., 2023). While popular in the early days of NN, the sigmoid activation function suffers from the vanishing gradient problem (Dubey et al., 2022). This problem is due to high or low input values leading to a similar output of the sigmoid

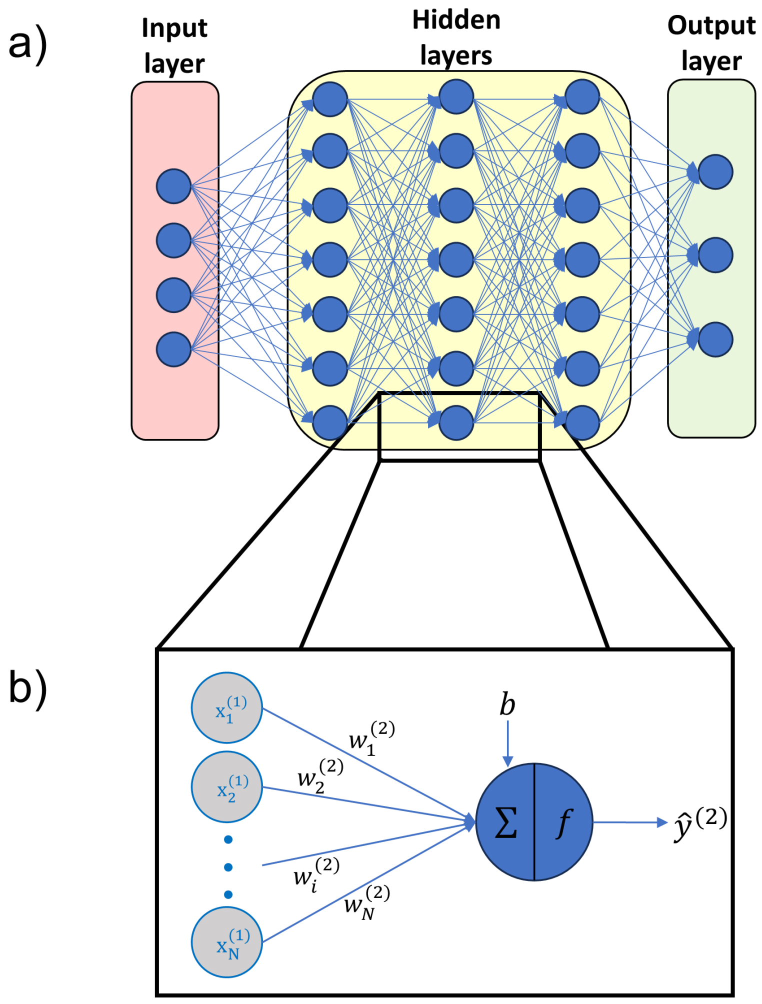

Figure 2.1: **a)** Overview of a simple neural network (NN) with the input, three hidden, and output layers. The layers are fully connected, meaning each neuron is connected to all neurons in the next layer. **b)** Zoom onto a single neuron. The inputs x_i are multiplied with their respective weights w_i and summed, to the result of which, a bias b is added. The result is transformed using an activation function f resulting in the neurons' output $\hat{y}$. The superscript of the weights and inputs refers to the layer number.

function, causing the gradient of the objective function to become close to zero. During training, this results in neglectible updates in the parameters, slowing down the process.

For this reason, ReLU and its modifications have become state-of-the-art. The ReLu activation function,

$$f(\mathbf{z})_{ReLU} = \begin{cases} 0, & \text{if } z < 0 \\ z, & \text{otherwise.} \end{cases} \tag{2.11}$$

returns zero for any negative z and z itself for any positive z. Accordingly, the gradient for positive inputs is one, and for negative inputs zero, which reduces the computational complexity and makes it very cheap to implement (Dubey et al., 2022; James et al., 2023). Another activation function, mostly implemented in the last layer, is the linear activation function which just returns the input itself.

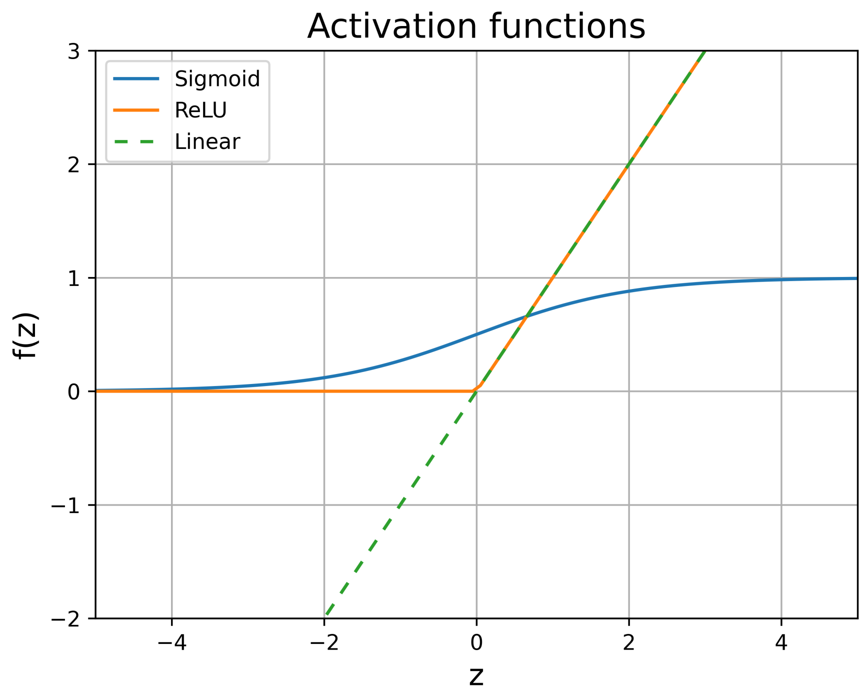

Figure 2.2: Visualization of the Sigmoid, Recitified Linear Unit (ReLU), and linear activation function. The Sigmoid function will output a value between 0 and 1 for any input value z, but has vanishing gradients for small or large values of z. The ReLU function returns 0 for any negative z and z itself otherwise. The gradient of the ReLU is either zero or one, making it cheap to compute. The linear activation is also referred to as no activation function, as it just outputs the input.

2.2.2 Training of Neural Networks

The key to making a NN useful for problem-solving is the training process. The training consists of updating the weights and biases of the neurons in an iterative fashion to improve their predictive performance on the training dataset. NN are trained in a supervised manner, meaning that the data is *labeled*, and therefore the correct output y for every input x is known.

After deciding on the structure of the network (how many layers and neurons, which activation function, etc.) all weights and biases are initialized. The updating of these weights and biases includes three steps:

- Forward propagation
- Calculating the loss
- Backward propagation

During the forward propagation, the data is passed to the input layer and the output $\hat{y}$ is calculated with the initialized weights and biases and defined activation functions (see Equation 2.8).

Using a loss function, the misfit between predicted output $\hat{y}$ and label y is calculated as a measurement of the NN performance. The choice of a loss function depends on the problem at hand. Two common choices with different purposes include the Mean Squared Error Loss Function (MSE) and the Cross-Entropy Loss Function (James et al., 2023). The MSE is calculated by

$$L_{MSE} = \frac{1}{2N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{2.12}$$

and is used in regression tasks, while the cross-entropy loss

$$L_{CrossEntropy} = -\sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i) \tag{2.13}$$

is used in multi-class classification tasks.

The former loss function calculates the average squared differences between predicted and observed values, giving a metric on how close these values match, i.e., how close the pixel values in an image are or how similar two curves are to each other. The latter loss function is intuitionally explained using the example of the classification problem of different types of animals. The label contains only zeros except for a single entry that corresponds to the correct animal, where it contains a one. Using, for example, the previously explained sigmoid activation function in the last hidden layer our prediction only contains values between zero and one, corresponding to the likeliness that an image belongs to a certain animal class. Because of the logarithm, the loss will grow exponentially with decreasing probabilities in the corresponding prediction entry of the correct class, making the model learn to assign higher probabilities to the correct animal.

Independent of which loss function is chosen, the target of the training process is to minimize the calculated loss. Recalling Equation 2.7, we observe that the weights w_i, the biases b_i, but also the input x_i (by changing the weights and biases of the previous layer) can be updated in order to minimize the losses of Equations 2.12 and 2.13. For now, we will consider the MSE loss function (Equation 2.12) and extend Equation 2.8 to a NN with multiple layers by introducing a superscript j which represents the layer number and replacing the input $\mathbf{x^j}$ by the previous layers output $\mathbf{\hat{y}^{j-1}}$:

$$\mathbf{\hat{y}^j} = f^j \left(\mathbf{W^j}^{\mathbf{T}} \mathbf{\hat{y}^{j-1}} + \mathbf{b^j} \right) = f^j(\mathbf{z^j}). \tag{2.14}$$

Combining Equations 2.12 and 2.14, we get

$$L_{MSE} = \frac{1}{2N} \sum_{i=1}^{N} (y_i - f^k(z_i^k))^2, \tag{2.15}$$

where the superscript k denotes the index of the last layer. Using gradient descent to update the weights and biases means that the partial derivatives of the loss function L with respect to all variable parameters, e.g., $\frac{\partial L}{\partial b_i^j}$ and $\frac{\partial L}{\partial w_i^j}$, have to be calculated. This is achieved using *backpropagation*, which starts in the last layer k and is then iteratively calculated for previous layers, hence the name.

Following the derivation of Nielsen (2015), we define the error δ_i^j of the i-th neuron in the j-th layer as

$$\delta_i^j = \frac{\partial L}{\partial z_i^j}, \tag{2.16}$$

which later will be related to our derivates of interests, e.g., $\frac{\partial L}{\partial b_i^j}$ and $\frac{\partial L}{\partial w_i^j}$. Applying the chain rule, the error of the output layer δ^k is given by

$$\delta_i^k = \frac{\partial L}{\partial \hat{y}_i^k} f'(z_i^k). \tag{2.17}$$

Equation 2.17 is easily computed as the first term on the right-hand side directly follows from Equation 2.12 as

$$\frac{\partial L}{\partial \hat{y}_i^k} = -(y_i - \hat{y}_i^k), \tag{2.18}$$

and the second term depends on our activation function, where the derivative in case of ReLU is either 1 or 0. Starting from the last layer, an equation for the error δ_i^j can be formulated in terms of the error in the next layer:

$$\delta = (\mathbf{W}^{j+1})^T \delta^{j+1} \odot f'(z^j), \tag{2.19}$$

where $\odot$ is the Hadamard product. Equation 2.19 allows us to compute the error backward through the network. It can be shown that the error is related to the derivatives of the bias and weight as

$$\frac{\partial L}{\partial b_i^j} = \delta_i^j \tag{2.20}$$

and

$$\frac{\partial L}{\partial w_{ik}^j} = \hat{y}_k^{j-1} \delta_i^j, \tag{2.21}$$

respectively. In the above equation, k denotes the neurons in the $(j-1)$-th layer

Having the partial derivatives for all weights and biases, the update can be calculated using gradient descent as

$$p_{t+1} = p_t - \alpha \frac{\partial L}{\partial p_t}, \tag{2.22}$$

where p now represents any parameter (weight or bias) of the NN and α is the step length. In fact, all the previous derivations only considered a single training example, and updating the weights and biases in this fashion for a dataset should be referred to as Stochastic Gradient Descent (SGD) (James et al., 2023).

A different and widely used optimization method for NN, also utilized in this thesis, is the Adaptive Moment Estimation (ADAM) optimizer introduced by Kingma and Ba (2017). The proposed optimizer updates the learning rates for each parameter individually based on a running average of the gradient of the loss m_t and the squared gradient of the loss v_t:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla_p L_t(p_{t-1}) \tag{2.23}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\nabla_p L_t(p_{t-1}) \odot L_t(p_{t-1}) \tag{2.24}$$

where β_1 and β_2 are hyperparameters controlling the exponential decay rates of the running averages. Hyperparameters are parameters whose values are set before the learning process begins and control the behavior of the learning algorithm. Unlike model parameters, which are learned during training, hyperparameters need to be specified in advance and can significantly affect the performance of the model. Kingma and Ba (2017) observed an initialization bias leading m_t and v_t towards zero during the initial timesteps. This is counteracted by applying a factor to the learning rate such that

$$\alpha_t = \alpha \frac{\sqrt{1 - \beta_2^t}}{1 - \beta_1^t}, \tag{2.25}$$

leading to the final updating rule of the parameters:

$$p_t = p_{t-1} - \alpha_t \frac{m_t}{\sqrt{v_t} + \hat{\epsilon}}. \tag{2.26}$$

The proposed default settings by Kingma and Ba (2017), based on an empirical evaluation, are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. This process is repeated iteratively until convergence of the NN to an acceptable loss.

2.2.3 Convolutional Neural Networks

Above we discussed the basic structure of NN and how the training process is carried out. Most of these concepts and equations are transferable to CNN and because it is out of the scope of this thesis, specifics and extension applying to CNNs will not be discussed. Instead, we will take a look at what makes a CNN special.

CNNs are one of the most popular types of NNs and are specifically suited for image-classification tasks (O'Shea and Nash, 2015; Albawi et al., 2017; Alzubaidi et al., 2021). The key concept behind convolutional layers is the use of learnable filters, so-called *kernels*. Given an input image to the convolutional layer, the filters are convolved with the image to produce *feature maps*. Convolving involves the calculation of the dot product of the kernel values with the corresponding image values which is mapped to a single entry of the feature map and repeated by shifting the kernel along the image. This process is illustrated in Figure 2.3 a).

Note that in this figure, a kernel size of 3×3, a stride of one (meaning that the filter position is always shifted by one pixel), and no padding is used. These hyperparameters can be adjusted for each convolutional layer to extract different features and adjust feature-map dimensions. Analogously to the process described in section 2.2.1, optimization aims at updating the learnable kernel weights to extract certain features that correspond to a certain class.

Another important layer type in CNN is the pooling layer (see Figure 2.3 b). While the preceding convolutional layer extracts feature maps with patterns from the input, the pooling layer sub-samples these feature maps (reducing their dimensionality). This allows to reduce the number of parameters, thus increasing computational efficiency while retaining the most important information (Alzubaidi et al., 2021). As for the convolutional kernels, the pool size, stride, and padding can be defined. Several pooling methods exist, with the most common ones being max and average pooling. As the names suggest, max pooling takes the maximum value within the pooling window and average pooling calculates the average within the pooling window to produce the downsampled feature map. Since pooling reduces the total information content large pooling windows (above size 3×3) will greatly decrease the models' performance (O'Shea and Nash, 2015).

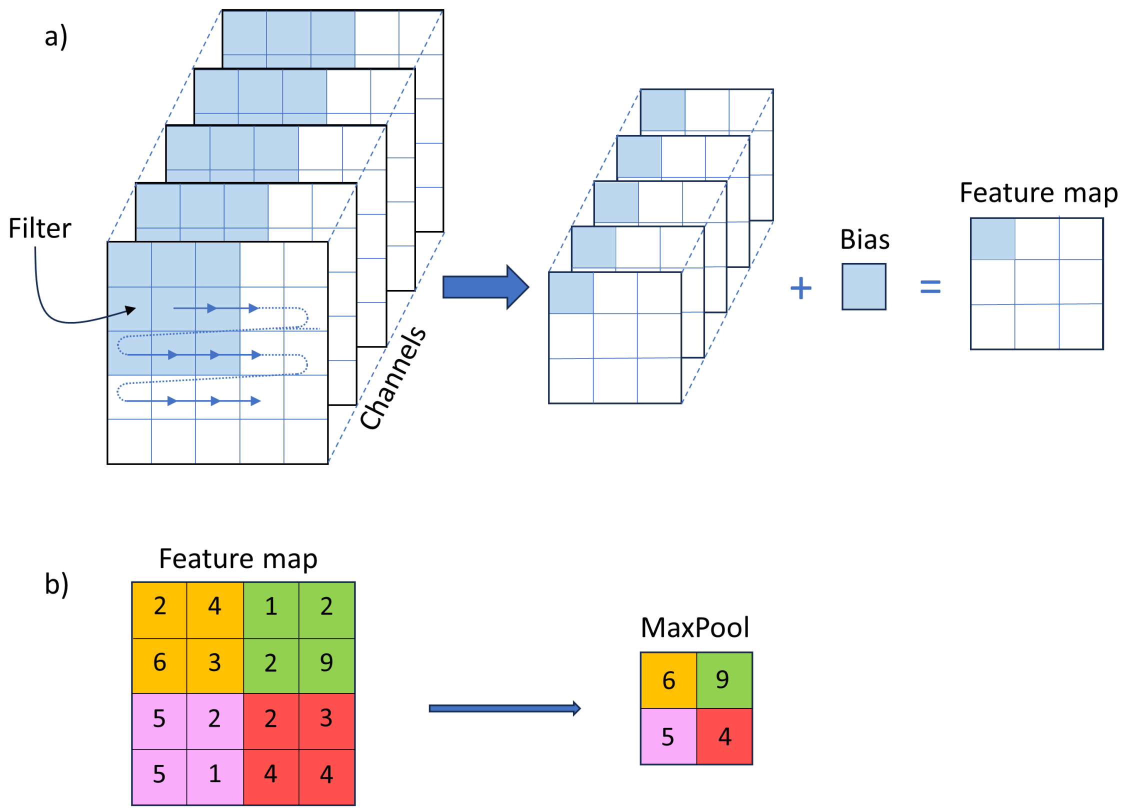

Figure 2.3: **a)** Visualization of the convolutional operation. The filter is shifted along the input image and each pixel is multiplied with the corresponding filter weight. Note that each convolutional filter is designed to have the same number of channels as the input image. Each channel represents one dimension of the input (e.g., three for a RGB image or the number of feature maps from the previous layer). During the convolution process, each filter channel independently interacts with its corresponding input channel. The feature map of the convolution operation is obtained by summing the contributions from all these interactions across the channels and adding a bias. **b)** Visualization of the pooling operation. In this case, MaxPooling is applied to the feature map with a size of 2×2 and stride of 2. The maximum value in each pool is taken to reduce the feature map dimension while keeping the most important information.

3 Methodology

3.1 Problem Definition and Proposed Methodology

The picking of dispersion curves from the phase-velocity spectrum, as described in section 2.1.3, is a labour-intensive task and highly susceptible to the professional carrying it out. Automatic picking methods exist, mainly using the maximum energy for each frequency in the dispersion spectra to pick the corresponding phase velocity. The drawbacks of these methods are the need for manual corrections in case of strong noise and apparent modes in the spectra.

Implementing a semi- or fully-automated, CNN-based process can circumvent these limitations. In this thesis, we aim to utilize the capabilities of CNNs which have been proven successful in different geophysical tasks. Araya-Polo et al. (2018) defined their CNN as the tomography operator, directly mapping semblance cubes to velocity models, Wu et al. (2020) applied CNN to invert seismic data for the impedance, and Li et al. (2022) trained a CNN to identify paleo-channels in 3-D seismic volumes.

Seismic data represents a recording of the wavefield sampled in time and space. The idea behind applying learnable kernels is the extraction of spatiotemporal features related to the dispersion of surface waves. These features can then be related to phase velocities for different frequencies leading to the extracted dispersion curves.

While deeper CNNs become less interpretable and more of a black box, some intuition behind this idea can be given when picturing edge-detection kernels. These edge detection kernels can be constructed to transform an image such that the output shows edges with specific orientations. In seismic data, the slope of maxima of refracted waves in neighbouring traces is translated directly into velocities. Thus, applying kernels that detect different slopes in the shot gathers results in strong activations in the feature maps if those slopes are present. In the succeeding layers, the kernels applied to the feature map volume calculated from the input shot gather might capture more complex relations, hopefully leading to the learning of the dispersion relation with offset.

Again, since the complexity of CNN increases with their depth, the previously described idea can only be taken as a general intuition, whereas the actual process of mapping shot gathers to dispersion curves will be more abstract.

Work on the extraction of dispersion curves using CNN has been done by Ren et al. (2020), Ren et al. (2021), Dai et al. (2021), and Chamorro et al. (2024). Ren et al. (2020) used a modified U-Net architecture to map from the f-k-spectrum directly to the corresponding dispersion curves. Dai et al. (2021) treated the problem as a segmentation task, where they trained a CNN to discriminate between background and dispersion energy in the f-v-spectrum. Their obtained output is subsequently used as a mask to fit the dispersion curves. Notably, Chamorro et al. (2024) was the first to apply the method described in this thesis and their code builds the framework for the work carried out in this thesis.

After reviewing Chamorro et al. (2024) and the accompanying codes, several questions emerged that we aim to address in this thesis. For their training process, only 1000 samples were generated, allowing for a very limited variation in the generated velocity profiles. In their case, the parameters for the model building were chosen after analyzing the field data. Combined with the limited variation in the training data, it is questionable if the CNN can identify features in the field shot gather related to the phase velocity, as any dispersion curve the model was trained on could be judged as acceptable. The model was also trained to predict phase velocities for frequencies that show no energy in the amplitude spectra of the field gathers. From a physical point of view, such predictions can not be made from the information content of the data. This also applies to their attempt to predict the first higher mode, as the picked first modes do not correlate with energy in the dispersion spectra (see Figure 10 in Chamorro et al., 2024) making their reliability hard to judge without additional measurements confirming the predictions. Furthermore, we compare the performance of models directly trained on shot gathers with those trained on the corresponding $f-v$ spectra.

The workflow of this thesis consists of the random generation of subsurface models which are then used to model synthetic shot gathers and calculate the corresponding $f-v$ spectra. The synthetic shot gathers and $f-v$ spectra are subsequently used to train the CNN for the inference on synthetic and field data. Figure 3.1 shows this workflow and the following subsections explain the steps in more detail.

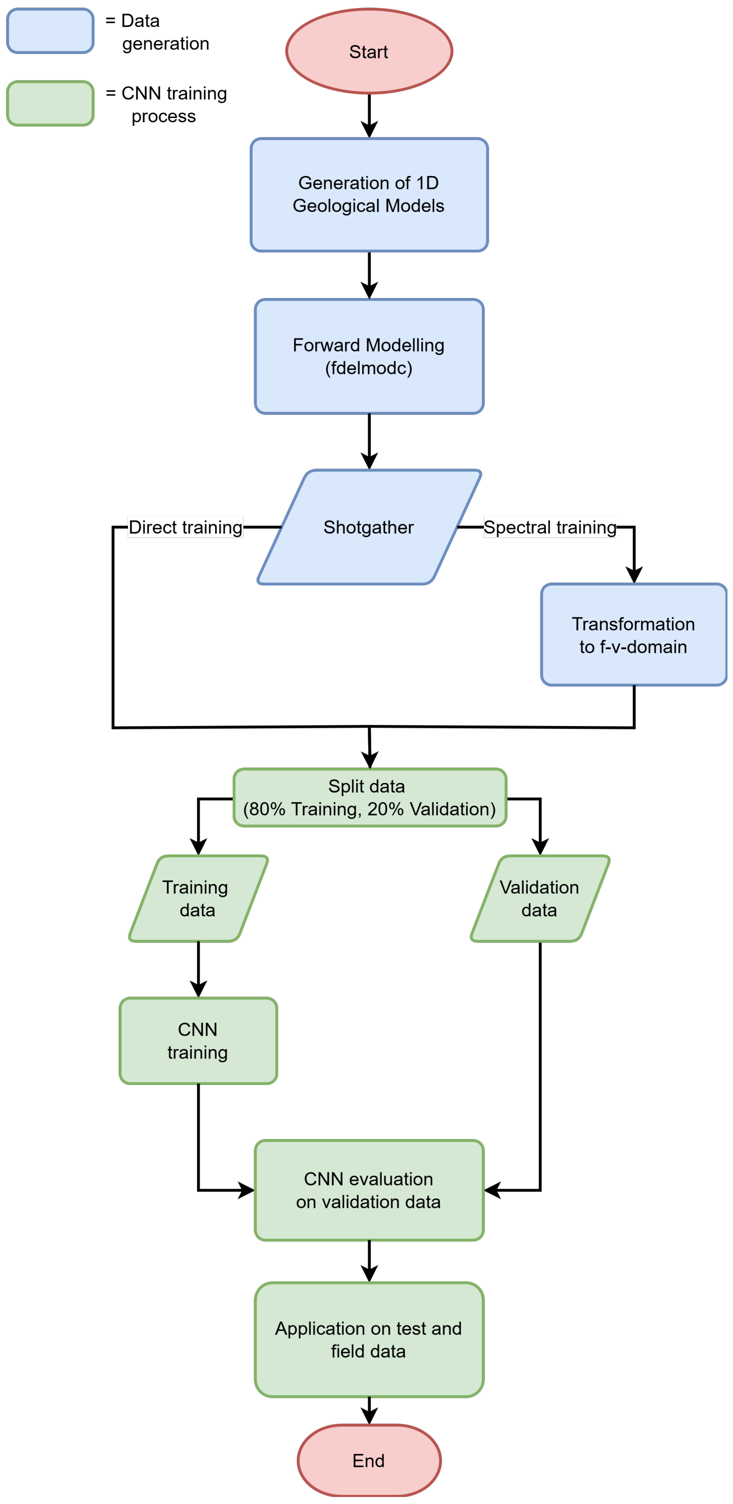
= Data generation
= CNN training process
Start
Generation of 1D Geological Models
Forward Modelling (fdelmodc)
Shotgather
Direct training
Spectral training
Transformation to f-v-domain
Split data (80% Training, 20% Validation)
Training data
Validation data
CNN training
CNN evaluation on validation data
Application on test and field data
End

Figure 3.1: Workflow for the experiments conducted in this thesis. First, random 1-D geological models are generated within a defined model space. For these geological models shot gathers are computed using the elastic finite difference (FD) implementation of *fdelmodc* (Thorbecke and Draganov, 2011). Depending if the CNN is trained on shot gathers or dispersion spectra, the CNN is either directly trained with the obtained shot gathers or the Phase-Shift transformation (Park et al., 1998) is applied to obtain the dispersion spectra. 80 % of the data is used for the training and the other 20 % for the validation. The model with the lowest loss on the validation data is used for the application on test and field data. The separate steps are detailed in the corresponding Sections.

3.2 Training Data

Training a CNN requires correctly labelled data, covering the settings it is intended to infer on (Choi et al., 2020). This implies that predictions inferred by the model are unreliable when applied to data outside the model space on which it was trained. In this thesis, we use synthetically generated data, as it allows us to cover a variety of different geological scenarios, whereas accessible field data is typically limited in amount and variety. The quality of predictions on field data using synthetic data for training is expected to perform worse, as less similarity exists with the simulated data. Therefore, it is advised to tailor the data-generation process to the field data. This applies to the used acquisition parameters, such as receiver number, receiver spacings, sampling rate, and source characteristics. Certain influences on the field data, such as noise, can not be determined beforehand and require special attention and additional techniques before inference is performed.

We compute the shot gathers for randomly generated geological models using *fdelmodc* (Thorbecke and Draganov, 2011), a software that implements a finite-difference (FD) scheme as described in section 3.2.2 and uses the SU data format as input and output. The corresponding dispersion curves for each geological model are calculated using the python package *disba* (Luu, 2021), which implements the Dunkin's matrix method (see Section 2.1.2).

Limitations for the synthetic data generation in this thesis arise due to limited computational resources. Because of the current state of our implementation, the number of training data that can be used is limited. Before training the CNN, pre-processing steps are applied to the data which require the data to be loaded into memory, limiting the number of data to a function of available memory and dataset size. We found that using 10000 models allowed us to run the training without running into memory issues and this number is therefore used for all experiments conducted in this thesis. It is important to note that this limitation in training data also limits the range of variability in the models used for training. Therefore, the random model generation parameters were carefully tuned to achieve good results without constraining the model space too much. Further details on the hardware and software specifications can be found in Appendix A.1.

3.2.1 Model Generation

The model generator we implement in this thesis iteratively builds 1-D velocity models from top to bottom, using the following parameters:

- $v_{s1,min}$, the minimum velocity of the first layer
- $v_{s1,max}$, the maximum velocity of the first layer
- g, largest expected velocity gradient

- n, the maximum number of modelled layers
- z_{max}, maximum depth to reach
- d_{min}, minimum thickness of layers
- d_{max}, maximum thickness of layers

First, a random thickness in the interval $[d_{min}, d_{max}]$ is initialized. Next, the layers' velocity is determined within the interval

$$v_s = [v_{s,min}, g \cdot z + v_{s,max}], \tag{3.1}$$

where z is the depth of the current layer midpoint. From the interval in equation 3.1, specified by the defined minimum $v_{s,min}$ and maximum $v_{s,max}$, defined as a linear function of depth using the maximum expected gradient, a random velocity is picked. The possible velocities are restricted to increments of 10. For the case where no velocity inversion is expected, the minimum velocity $v_{s,min}$ for the next layer is updated as the velocity of the current layer. In the case of creating models with velocity inversion, we implement additional rules that prevent too high-velocity differences of successive layers and limit the occurrence of the inversion layer to a defined depth range. The process is stopped after reaching the maximum depth z_{max} or the maximum number of layers n.

While the seismic shear wave velocity v_s is the desired property required for geotechnical purposes (Hussien and Karray, 2016), the compressional wave velocity v_p and the density ρ are needed for the modelling process. Because both v_p and ρ have limited influence on the propagation of Rayleigh waves, we use empirical relations to estimate them as a function of v_s. For the calculation of v_p, we assume a constant Poisson's ratio σ of $\frac{1}{3}$, which is the average for stiff to soft clays and on the upper end for silts and dense to loose sands (Kumar Thota et al., 2021). The Poisson's ratio is related to the $\frac{v_s}{v_p}$-ratio as

$$\frac{v_s}{v_p} = \sqrt{\frac{0.5 - \sigma}{1 - \sigma}}, \tag{3.2}$$

resulting in a $\frac{v_s}{v_p}$-ratio of 0.5. Therefore, we calculate the compressional-wave velocity as

$$v_p = 2 \cdot v_s \tag{3.3}$$

The density ρ can then be estimated using the empirical relationship from Gardner et al. (1974):

$$\rho = 0.31 \cdot v_p^{0.25} = 0.31 \cdot (2 \cdot v_s)^{0.25} \tag{3.4}$$

3.2.2 Forward Modelling

The forward modelling of the seismic wavefield is done for the 2D elastic wave equation. Since an FD scheme is used for approximating the wave propagation, certain criteria have to be obeyed depending on the velocities of our geological models and used wavelet. The FD uses a centralized 4th-order Crank-Nicolson scheme for the approximation of the first-order derivative in space and a 2nd-order central scheme for the first-order derivative in time. For further details on the implementation, the reader is referred to the *fdelmodc* manual (Thorbecke et al., 2020).

Thorbecke et al. (2020) give two equations to ensure stability and minimize numerical dispersion. The condition for the stability of the FD scheme is

$$\Delta t < \frac{0.606 \cdot \Delta h}{c_{max}}, \tag{3.5}$$

where Δh is the grid spacing (assuming that $\Delta h = \Delta x = \Delta z$) and c_{max} the largest velocity in the model. If not fulfilled, the wave will travel further than one grid spacing within one time-step, leading to errors in the solution. Numerical dispersion will increasingly occur if

$$\Delta h < \frac{c_{min}}{5 \cdot f_{max}} \tag{3.6}$$

is not fulfilled, where f_{max} is the highest modeled frequency and c_{min} the lowest velocity in the model. Numerical dispersion is, unlike geometric and visco-elastic dispersion, not a physical phenomenon of wave propagation but introduced by the numerical approximation of the FD scheme. Equation 3.6 is not a mathematically derived relation but a practical recommendation to keep the numerical dispersion neglectable.

In addition to the above stability and dispersion considerations, other parameters must be chosen prior to the modelling. Assuming the use of a sledgehammer (or drop-weight), the source is chosen to act in the vertical direction and the receivers to record the vertical component of the wavefield. Naturally, the upper boundary is chosen to be a free surface, while all other boundaries are tapered boundaries. Tapered boundaries extend the original model domain by a defined number of grid points at which the wave is increasingly damped. This is necessary to reduce unwanted side reflections but comes at the cost of increasing the computational cost depending on the number of taper points chosen. An alternative boundary to tapering is Perfectly Matched Layers (PML) which produces better results (Thorbecke et al., 2020) in suppressing side reflections but has not been implemented yet for the elastic case in *fdelmodc*.

As mentioned in Subsection 2.1.2, the forward calculation of the theoretical dispersion curves is implemented using the Python library *disba* (Luu, 2021). For the calculation of higher modes, we encountered the problem of root skipping (observable in Figure 4.11, middle panel). Root skipping occurs if the phase velocity increment for the root-finding is too large, therefore missing the root for the correct mode and instead using the root of a lower mode. One solution to this problem is decreasing the phase velocity increment for the root-finding, but we find that the increment needed to prevent root skipping is so small, that the computational cost increases to an undesirable level. Reviewing the code provided by Chamorro et al. (2024) we find that they encountered the same problem and we adopted their solution of a median filter, changing the filter length to 7, to reduce the amount of root skipping. While reducing the total number of outliers, root skipping still occurs using this approach, which has to be considered when training on the erroneous dispersion curves is performed.

3.2.3 Pre-Processing Steps

The shot gathers and corresponding dispersion spectra used for training, as well as the testing and field data utilized for prediction, are subject to minimal processing. We normalize the shot gathers trace-wise and the dispersion spectra frequency-wise. For the shot gathers we do this to enhance the structural continuity of the surface waves to the same range of values and for the dispersion spectra to normalize the maximum energy, corresponding to the desired phase velocity, for each frequency. The dispersion spectra are restricted to the frequency and phase velocity range we expect in our testing and field data. Furthermore, the time shift applied to our wavelets prior to the forward modelling is removed.

For testing purposes, we introduce Gaussian noise to selected shot gathers to evaluate its impact on the prediction accuracy of our CNN models, which were trained using noise-free data.

3.3 CNN Architecture

In this thesis, we use a modified version of the Residual Neural Network (ResNet) architecture introduced by He et al. (2015). The ResNet architecture is chosen because of its simple implementation using *PyTorch* and proven performance on image-classification tasks. Multiple variants of ResNet exist, differing in their amount of hidden layers. Because of the limited amount of used data, due to memory constraints, we implement the lightest version, ResNet-18.

ResNet gets its name from the utilization of residual connections (also kown as skip connections). In a traditional deep CNN, each layer feeds its output as input to the next layer in a sequential manner. However, as the network gets deeper, it becomes challenging to train due to problems like vanishing gradients, where gradients become very small and updates to weights during backpropagation become insignificant (He et al., 2015). Residual connections address this issue by allowing the input to a layer to be directly added to the output of that layer. Mathematically, if x is the input to a residual block, and $F(x)$ is the transformation applied by the block, the output y of the residual block is:

$$y = F(x) + x \tag{3.7}$$

The motivation behind adding the input to the residual block to its output is, that in case of the input already being nearly optimal, the solvers may simply drive the weights of the transformation $F(x)$ to zero (He et al., 2015).

ResNet-18 also makes use of batch normalization between each convolutional layer and activation function. Batch normalization was proposed by Ioffe and Szegedy (2015) to stabilize and accelerate the training. By normalizing the input of each layer to have a mean of zero and a variance of one, batch normalization addresses the internal covariate-shift issue discussed by Ioffe and Szegedy (2015). This normalization reduces the sensitivity to initial conditions and learning rates, thereby improving the generalization capability of the model.

Our primary modification to the traditional ResNet-18 architecture is its adaptation to output the dispersion curve directly from the input shot gather or dispersion spectra. This adaptation involved that we tailor the network's final layer to predict continuous-valued curves instead of discrete class labels, which the original architecture was designed for. Instead of the original 1000-dimensional fully-connected output layer, we adapt the dimension to the frequency content and frequency resolution of our data. For example, if the signal contains frequencies between 10 Hz and 80 Hz and the frequency resolution equals 1 Hz, the output layer will be 71-dimensional. We do this in an attempt to integrate the underlying physics into the CNN. The architecture of ResNet-18 is shown in Figure 3.2.

In some experiments, we also integrate dropout with rates of 0.15 or 0.3 after every ReLU activation function but the last one of the ResNet architecture, to prevent overfitting, as proposed by Hinton et al. (2012). Dropout is a regularization technique where a percentage of the neurons in each layer are randomly disabled during training. This means that in each training iteration, some neurons are ignored, forcing the network to learn redundant representations. By doing so, dropout helps in reducing overfitting and improves the model's ability to generalize better to unseen data.

We chose to use ADAM (see Subsection 2.2.2) for optimization and the Mean Squared Error (MSE) loss as the loss function. Originally, Chamorro et al. (2024) proposed to use the Huber loss and an additional regularization term to promote smoothness in the predictions. During our experiments, we found the Huber loss, particularly in the case of velocity inversions or high-velocity contrasts, to produce predictions that average the true dispersion curves, as these deviations are not as strongly penalized as for the MSE loss. We chose to not use the additional regularization term to produce smoother results, as we did not observe any case where this was beneficial.

In all our experiments, we enable fixed randomness in our code, allowing for reproducible results and better comparison after changing the experimental setup. This includes the seeding of the random number generation for *PyTorch* and *NumPy*.

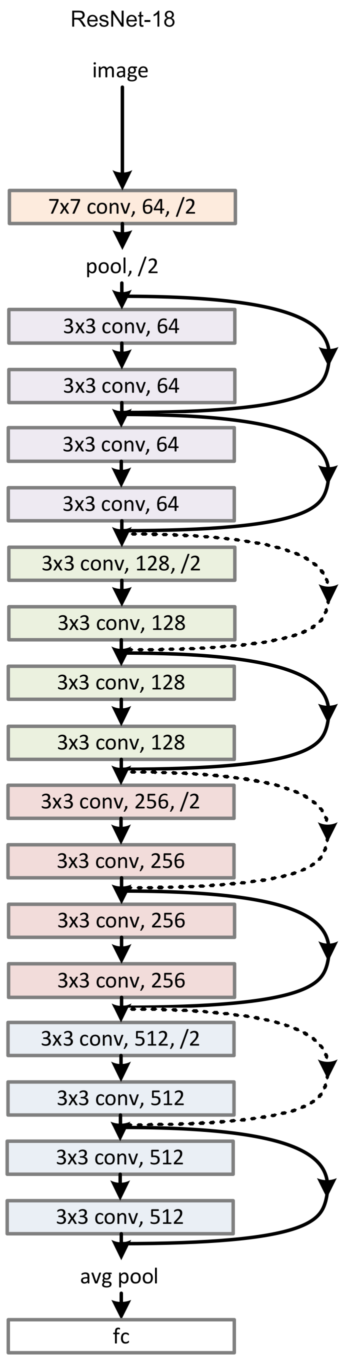

Figure 3.2: ResNet-18 architecture utilized in this thesis. The notation within each layer (colored box), e.g., [3x3 conv, 64], refers to the kernel size and the number of kernels, respectively. The notation /2 indicates the use of a stride of 2, halving the dimension of the inputs. Otherwise a stride and padding of 1 is used. The solid lines represent simple residual connections, whereas the dotted lines indicate an additional increase in the number of channels to account for the different dimensions between layer blocks (boxes with same color). Modified after He et al. (2015).

4 Results

In this chapter, we present the findings of this thesis. Initially, we show the results of the hyperparameter evaluation, which form the foundation for the subsequent analyses. Following this, we address the performance and limitations of the CNNs using different synthetic datasets. Finally, we apply the framework to field data and compare its performance to the conventional approach.

4.1 Hyperparameter Testing

One critical parameter for the training of CNN is the amount of training data used. As more data generally leads to better performance, we aim to maximize the number of training data while keeping the computational cost reasonable. As discussed in Section 3.2, we encountered memory limitations, therefore restricting the experiments to 10000 geological models per training.

Key hyperparameters such as the number of epochs, the batch size, the learning rate, and the dropout rate significantly influence the accuracy of the model. Proper tuning of these parameters is essential to prevent overfitting and underfitting the model to the training data. To this end, we carry out the following hyperparameter tuning procedure for the training on shot gathers. For the training on dispersion spectra, we find that the choice of hyperparameters has less influence on the outcome, and for this reason, those results are not detailed in this Subsection. We conduct a grid search for the following hyperparameters for training the CNN over 200 epochs:

- Dropout = [0.0, 0.3, 0.5]
- Batchsize = [4, 16, 64]
- Learningrate = [0.0001, 0.001, 0.01]

Given the computational expense of a grid search, we opt for a wide range of values for each parameter while using a smaller training set of 2000 models. Figures 4.1 and 4.2 show the training and validation loss, respectively, for all hyperparameter combinations. The observed fluctuations reveal the stochastic nature of the underlying training process. For a learning rate of 0.01, the fluctuations for the validation loss are judged to be exceedingly large, wherefore a smaller learning rate should be chosen for a smoother learning process. We avoided models with low validation errors but highly fluctuating validation losses, as these may result from sharp local minima, leading to poorer generalization. Comparing the results for learning rates of 0.001 and 0.0001, we observe that the set of curves with the latter has not converged after 200 epochs. This implies more epochs are required, thus increasing the computational cost to achieve similar results as with a learning rate of 0.001.

Analyzing the results for a learning rate of 0.001, we observe that a batch size of 64 is too large for our dataset of 2000 samples when used without dropout. This leads to higher training and validation losses when compared to the batch sizes of 4 and 16. This observation was also noted by Keskar et al. (2017), who attributed this phenomenon to the convergence to sharp minima when using large batch sizes. For batch sizes of 4 and 16, we observe similar results for the training and validation, indicating that both are suitable. On the other hand, if dropout is applied, the training loss increases with increasing dropout rate, while the validation loss remains similar for all hyperparameter combinations. This could indicate that without dropout the CNN is overfitted, as the training and validation dataset stem from the same population of models and are therefore "memorized", resulting in both low training and validation losses. The convergence to a similar, higher validation loss for different dropout rates and batch sizes could indicate that independent of both hyperparameters, the same features are learned from the training data, leading to a better generalization.

Across all combinations of learning rates and batch sizes, we observe that larger dropout rates result in larger training and validation errors. This might not necessarily be detrimental, as we will show in Subsections 4.2 and 4.3 when applying the trained CNN on testing data with different characteristics than our training data (e.g., synthetic data with noise and field data).

Table 1 presents the smallest validation loss achieved for each hyperparameter set and the corresponding computational time in seconds. The best result on the validation dataset is achieved with a learning rate of 0.01, a batch size of 4, and a dropout rate of 0. The second-lowest validation loss is achieved with a learning rate of 0.001, a batch size of 16, and a dropout rate of 0. The slight variations in these results may arise from inherent stochasticity. Conversely, with a learning rate of 0.001, the validation loss converges more smoothly to a desired optimum. Considering the duration of the training process and the use of only 2000 models instead of 10000, it becomes evident that larger batch sizes, leveraging GPU parallelization capabilities, are essential for handling extensive datasets.

Based on our hyperparameter evaluation, we adopt a learning rate of 0.001, which is also recommended by the authors of the ADAM optimizer (Kingma and Ba, 2017). To balance computational efficiency, we opt for a batch size of 16, as it is expected that the computational time will increase significantly for 10000 models, and the validation losses in Figure 4.2 are similar to the smaller batch size of 4. Since improvements are still noticeable beyond 200 epochs, training is performed for a total of 300 epochs. Additionally, we implement an early stopping mechanism, which stops the training if the validation loss does not decrease for 15 consecutive epochs. This adoption was based on experiments conducted in the following sections, which demonstrate a tendency for the model to overfit. This was evident through an increase in validation loss for a continuous decrease in training loss.

Although the lowest validation losses were achieved without dropout, we reserve the option to apply and compare results with a dropout probability of 0.15 and 0.3 when evaluating the CNN models on noisy synthetic and field data.

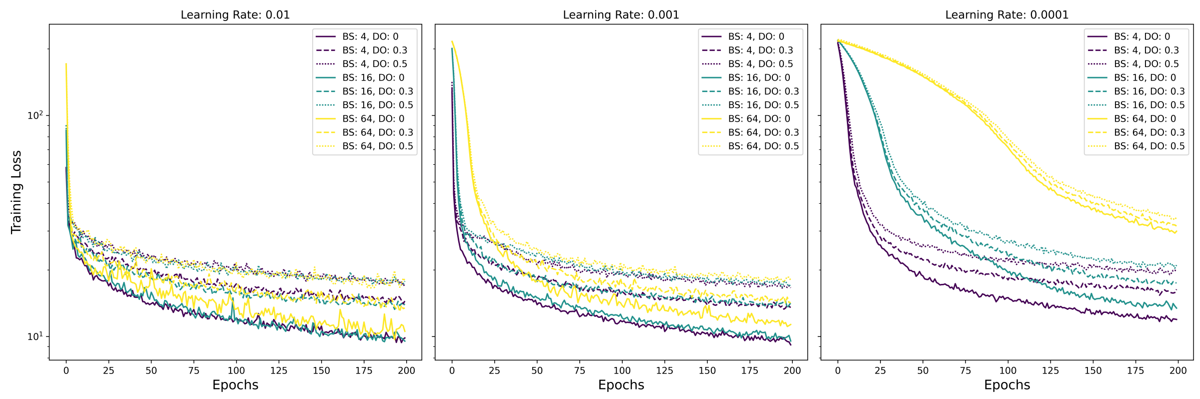

Figure 4.1: Training loss plot over training epochs for different combinations of hyperparameter. Each panel shows different combinations of batch size (BS) and dropout (DO) for a fixed learning rate. BS are differentiated by color, while different DO have a different linestyle (see legend).

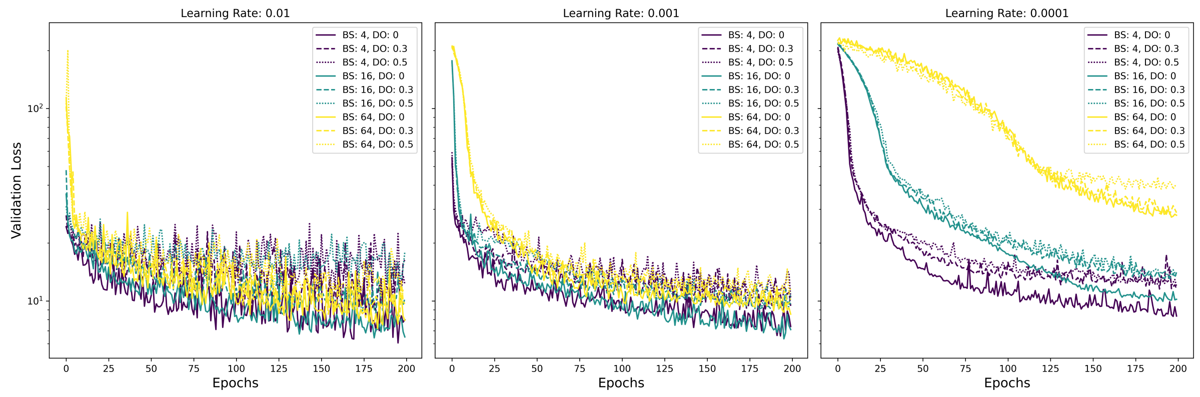

Figure 4.2: Validation loss plot over training epochs for different combinations of hyperparameter. Each panel shows different combinations of batch size (BS) and dropout (DO) for a fixed learning rate. BS are differentiated by color, while different DO have a different line style (see legend).

Table 1: Results of hyperparameter grid search. For each set of parameters (e.g., learning rate, batch size, and dropout) the CNN is trained on 2000 shot gathers for 200 epochs. The training time and minimum validation loss are saved and used to evaluate the hyperparameter combinations. Bright green indicates the best result, which corresponds to the smallest validation loss achieved across all combinations. Dark green indicates the second-best result.

Learning Rate	**Batch Size**	**Dropout**	**Minimum Value**	**Time** [s]
0.01	4	0	6.1	1028
0.01	4	0.3	8.8	1030
0.01	4	0.5	9.9	1034
0.01	16	0	6.4	893
0.01	16	0.3	8.5	894
0.01	16	0.5	12.4	894
0.01	64	0	7.2	858
0.01	64	0.3	8.2	857
0.01	64	0.5	10.1	858
0.001	4	0	6.6	1020
0.001	4	0.3	9.0	1028
0.001	4	0.5	10.0	1026
0.001	16	0	6.4	882
0.001	16	0.3	9.0	887
0.001	16	0.5	8.9	885
0.001	64	0	8.5	850
0.001	64	0.3	9.4	852
0.001	64	0.5	10.4	853
0.0001	4	0	8.3	1025
0.0001	4	0.3	11.5	1030
0.0001	4	0.5	11.9	1030
0.0001	16	0	10.0	886
0.0001	16	0.3	13.1	887
0.0001	16	0.5	13.3	887
0.0001	64	0	27.3	849
0.0001	64	0.3	27.8	851
0.0001	64	0.5	38.2	846

The results presented in the following Subsections are obtained from different training processes by changing the training data, the network architecture, and hyperparameters. The training and validation losses over epochs obtained for each result are shown in Appendix A.3.

4.2 Synthetic Data Results

For the generation of all the seismic synthetic data shown in this subsection, we use a flat-spectrum wavelet with frequencies of 5-10-60-70 Hz as the source signature. These frequency values correspond to the minimum frequency, the left attenuation point, the right attenuation point, and the maximum frequency of the wavelet, respectively. A time-shift of 0.25 s is applied to obtain a causal wavelet, which is accounted for during the pre-processing of the synthetic data. Figure 4.3 displays the wavelet and its corresponding amplitude spectrum. As dispersion can only occur for frequencies present in the wavelet, we limit the calculation and prediction of dispersion curves to the range between 10 and 60 Hz. We evaluate the performance and limitations of our approach using two different geological settings.

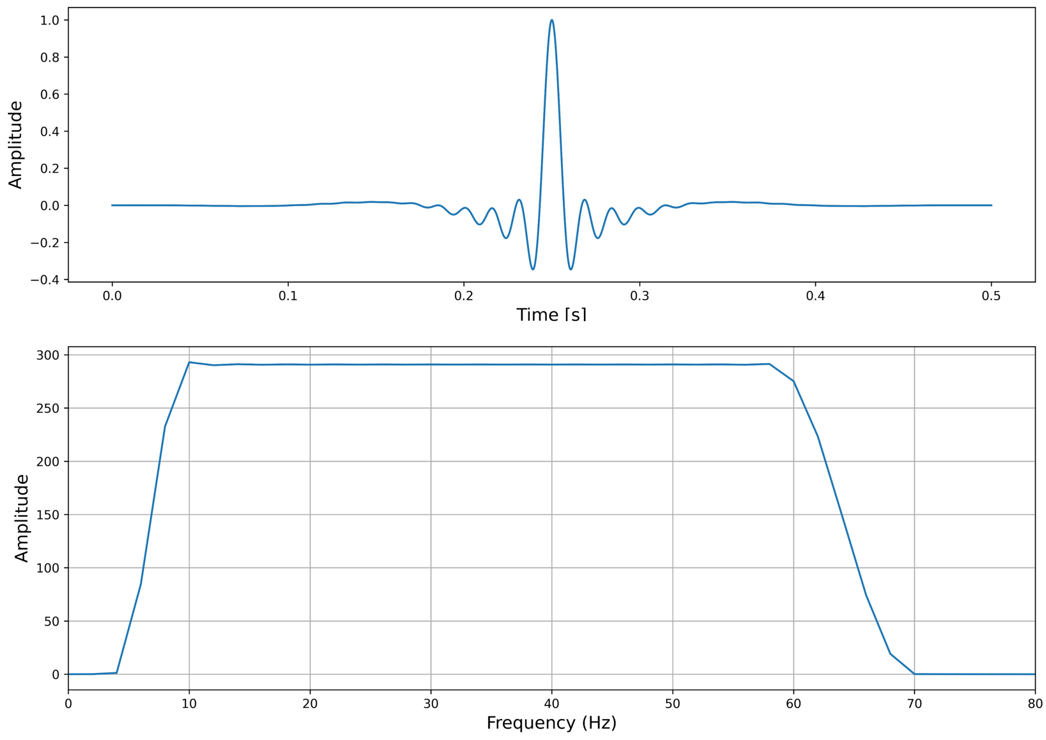

Figure 4.3: **Top**: Flat-spectrum wavelet utilized for the modelling of the synthetic data in this thesis. A time shift of 0.25 s is applied to make the wavelet causal. **Bottom**: Corresponding amplitude spectrum. The minimum and maximum frequencies are 5 Hz and 70 Hz, respectively. The left and right attenuation points are 10 Hz and 60 Hz, respectively.

4.2.1 Benchmark Model

Figure 4.4 shows the first v_s-model utilized to generate synthetic testing data. The left side of the model features a simple, layered structure without lateral variation. On the right side, the layers dip in the shallow part and a velocity inversion is introduced. We use this benchmark v_s-model to generate the synthetic seismic data to test our framework. The corresponding v_p and density models are obtained using Equations 3.3 and 3.4.

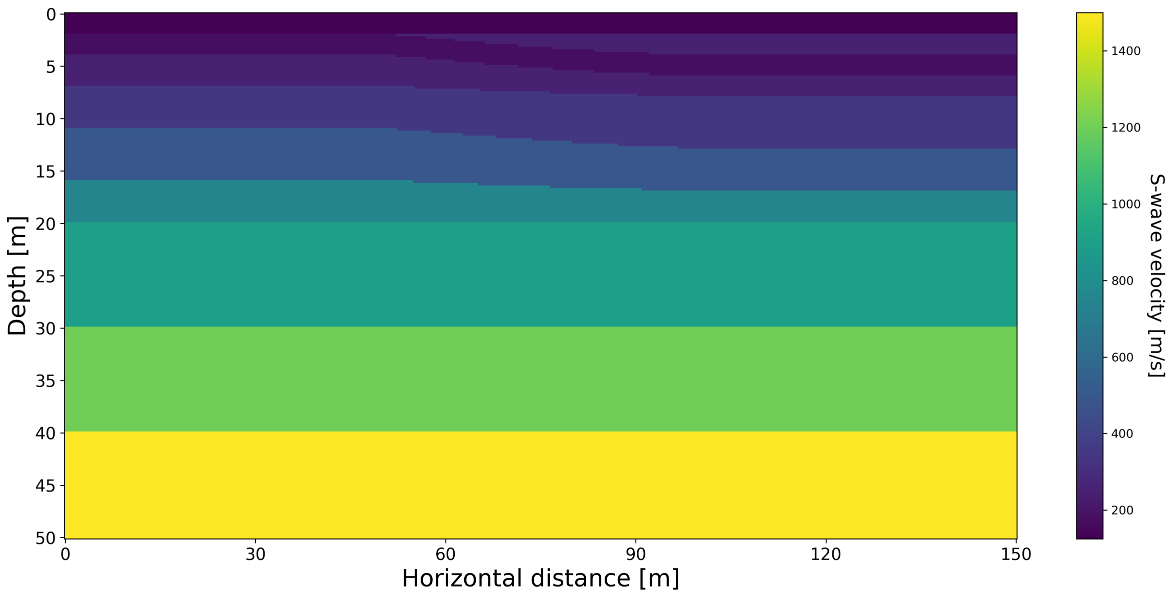

Figure 4.4: Simple shear-wave velocity model used for testing the proposed method. The left part is simply layered, showing increasing velocity with depth. In the right part, the upper layers start to dip and a velocity inversion is introduced as the third layer. The compressional-wave velocity and density models needed for the forward modelling are obtained using Equations 3.3 and 3.4.

Generating models with a single set of parameters for the random model generation, which accounts for both sides of the benchmark subsurface model, introduced too much variability in the generated models due to the velocity inversion. We find that the incorporation of a velocity inversion in the random model generating process needs further adjustment, as Equation 3.1 produces models with too large velocity jumps after the low-velocity layer, and thus generating a small number of models within the expected range for our benchmark model. Therefore, to address the variability introduced by the velocity inversion, we generate two separate training datasets, containing 5000 models for each side of the model. The models generated for the left side are created without allowing a velocity inversion, while the models for the right side are created with a velocity inversion in the third or fourth layer. Table 2 summarizes the settings for our random model generation. Figures 4.5 and 4.6 show the dispersion curves and v_s-profiles for the left side of the benchmark model (no velocity inversion) and right side of the benchmark model (with velocity inversion) training datasets, respectively.

Table 2: Modelling parameters used for the random model generation as described in Subsection 3.2.1. Two datasets are created to account for the velocity inversion on the right side of the model shown in Figure 4.4.

	Left Side	Right Side
$v_{s1,min}$ [m/s]	100	100
$v_{s1,max}$ [m/s]	350	300
g [m/s/m]	35	35
n	20	20
z_{max} [m]	50	50
d_{min} [m]	1	1
d_{max} [m]	8	10
Additional rules	-	VelInv in 3rd or 4th layer with thickness of 0.25 m to 3 m

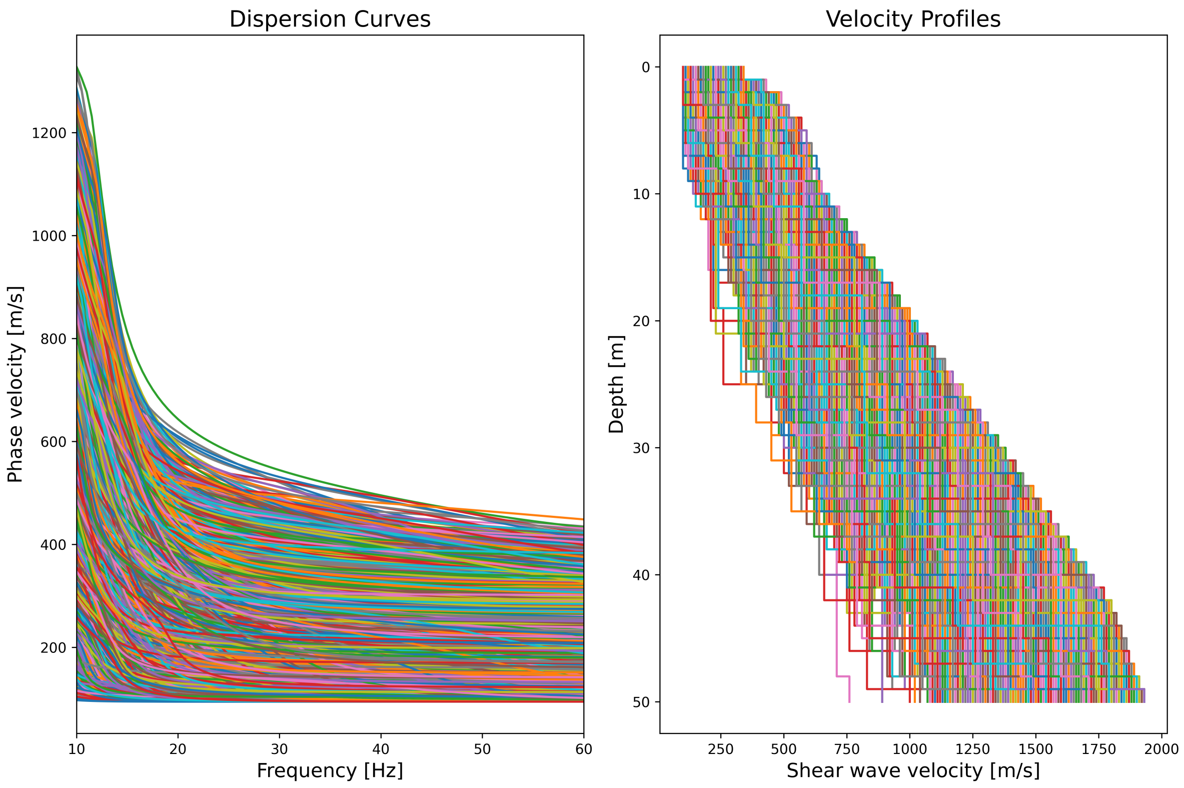

Figure 4.5: **Left**: Dispersion curves of the randomly generated 1-D geological models used as the training dataset for the left half of the model shown in Figure 4.4. **Right**: Their corresponding v_s profiles.

For the testing dataset, we model shots every 5 m along the benchmark model with a receiver array consisting of 24 geophones with 1 m spacing and a minimum source offset of 5 m. The total recording time is 0.75 s with a receiver time sampling of 3 ms. After accounting for the initial time shift of the wavelet, the final recording length is 0.5 s. For the midpoint of each shot gather, the 1D velocity profile at this midpoint location is

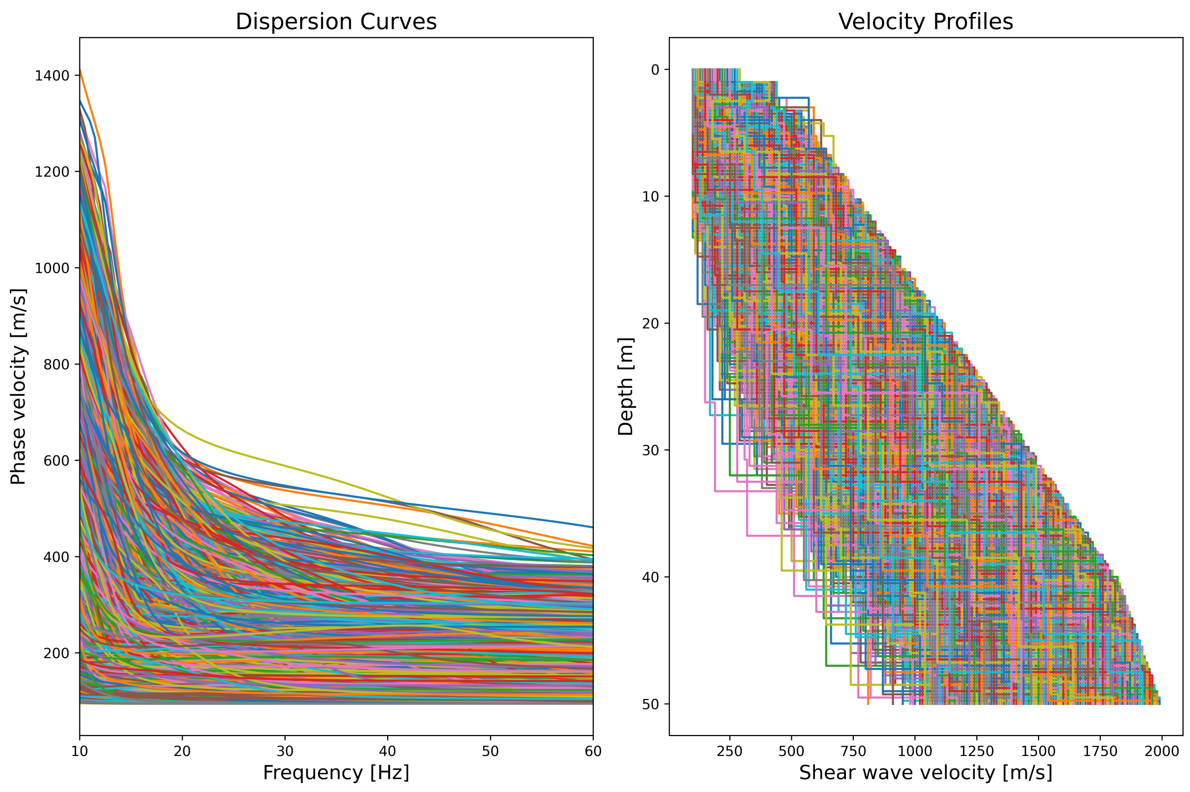

Figure 4.6: **Left**: Dispersion curves of the randomly generated 1-D geological models used as the training dataset for the right half of the model shown in Figure 4.4. **Right**: Their corresponding v_s velocity profiles.

extracted and the theoretical dispersion curve is calculated. Since our model is laterally heterogeneous, the calculated dispersion curves between 50 and 100 m horizontal distance do not represent the actual dispersion curves, as its calculation assumes a laterally homogenous subsurface (see Section 2.1.2). The lowest velocity in the generated models is 100 m/s and the highest frequency in our wavelet is 80 Hz. Using Equation 3.6 to avoid numerical dispersion, we choose a model grid spacing of 0.25 m, and using Equation 3.5 to ensure the numerical stability, we use a time-step of 0.3 ms for the FD modelling.

The CNN is first trained with each dataset separately and afterwards with the combined datasets. In total, six CNN models are trained: three on the shot gathers and three on the dispersion spectra. Note that no dropout is used for this experiment. Figure 4.7 shows the prediction results for the CNN trained on shot gathers for three shots with source positions at 10 m, 60 m, and 120 m.

The predictions for the first shot at 10 m (upper row in Figure 4.7) show a good match between theoretical and predicted dispersion curves, for both training on shot gathers and on dispersion spectra. For the CNN trained on shot gathers (top row, middle column), we observe that the best fit is achieved for the model trained on the dataset without velocity inversion, the predictions for the model trained on the dataset with velocity inversion show a larger deviation, and the predictions for the combined dataset shows intermediate deviation between the other two results. For the CNN trained on dispersion spectra (top row, right column), an almost perfect match between theoretical and predicted dispersion curves can be observed, independent of the training dataset used.

For the shot location at 60 m (middle row in Figure 4.7), which contains lateral velocity variations, the predictions are not able to accurately estimate the dispersion curve, but still show a good match with the underlying dispersion spectra for both, the CNN trained on shot gathers and dispersion spectra. For the models trained on shot gathers (middle row, middle column), the predictions produce dispersion curves without inflection points, not capturing the characteristics of the low velocity layer. Similar, the predictions for the models trained on the different dispersion spectra (middle row, right column) correctly estimate frequencies from 10 Hz to 15 Hz and from 40 Hz to 60 Hz, while underestimating the velocities in between, also producing a dispersion curve without inflection points. These results were anticipated, as the CNN model is neither trained on data with lateral velocity variations nor provided with accurate labels, given that the dispersion curve calculation assumes a laterally homogeneous subsurface.

The theoretical dispersion curve of the right side of the velocity model(solid black line in the bottom row of Figure 4.7) shows inflection points at around 20 Hz and 35 Hz. For the predictions made by the CNNs trained on shot gathers (bottom row, middle column), the inflection points, and the theoretical curve in general, are matched very well by the model trained on the dataset with velocity inversions. The model trained on the combined dataset also aligns closely with the shape of the theoretical dispersion curve, though it deviates slightly between 20 Hz and 30 Hz. In contrast, the predictions from the model trained on data without velocity inversion only vaguely follow the characteristic shape of the label. The predictions from the CNN trained on the dispersion spectra (bottom row, right column) are very similar to each other but less accurate in predicting the true dispersion curve compared to the CNN trained on shot gathers, as they tend to underestimate the velocities between the two inflection points.

The CNN models trained on the datasets with or without velocity inversion produce a more accurate prediction when the shot gather corresponds to a location in the benchmark model with or without velocity inversion, respectively. Predictions using the combined dataset demonstrate increased accuracy compared to the CNN models trained on a single dataset applied to the opposite side of the model they were trained on. Interestingly, regardless of the training dataset used, the predictions for the CNN models trained on dispersion spectra are all very similar and show only minor deviations.

Additionally, the predictions show good agreement with the theoretical dispersion curves considering the variety of dispersion curves they are trained on. Furthermore, we observe that the calculated dispersion spectra using the PS method show increasing deviations from the theoretical dispersion curves for frequencies above 45 Hz.

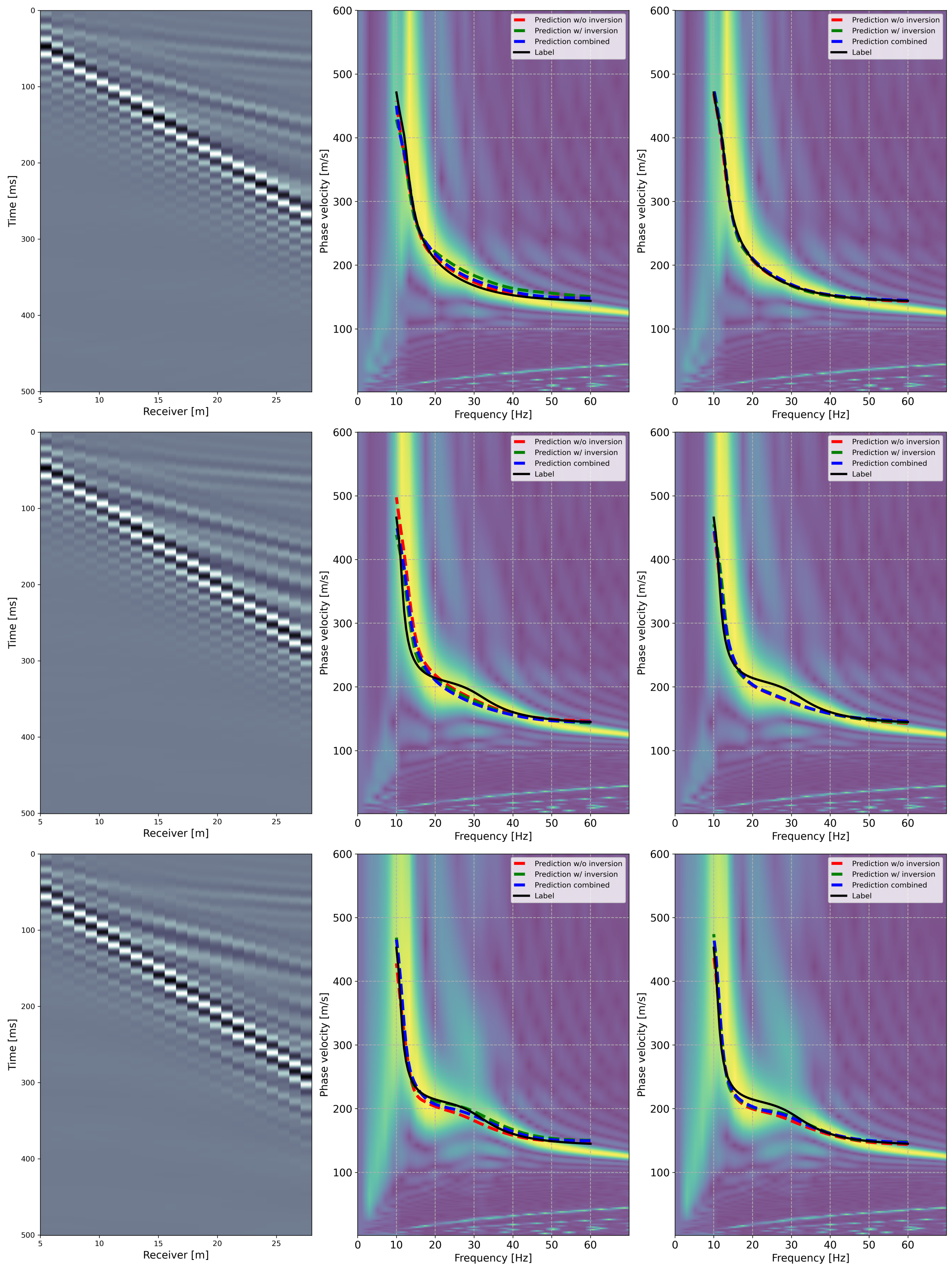

Figure 4.7: **Left**: Common-shot gathers generated using the velocity model in Figure 4.4 with the source located at 10 m, 60 m, and 120 m (from top to bottom, respectively). **Middle**: The corresponding dispersion spectra, calculated using the PS method by Park et al. (1998) and the predictions of the CNN trained on shot gathers. **Right**: Predictions for the CNN trained on dispersion spectra. The solid black curve represents the dispersion curve for the velocity model extracted at the midpoint of the shot gather (label). The coloured curves represent the predictions for the CNN trained on different datasets.

Table 3 shows the average root-mean-square error (RMSE) between the predicted and label dispersion curves at different locations across the benchmark model. The RSME is calculated as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}, \tag{4.1}$$

where N is the number of samples of the dispersion curve, $\hat{y}$ the label, and y the prediction.

We define the left part of the model as the shot positions until 40 m, the middle part as the shot locations between 50 m to 80 m, and the right part as locations between 90 m to 120 m. Similar observations can be made as described for the three selected shots above. Notably, the CNN trained on the combined shot-gathers dataset exhibits a higher average RMSE compared to the one trained solely on shot gathers with velocity inversion. We attribute this to the better predictions on the laterally varying part of our benchmark model by the CNN trained on shot gathers with velocity inversion only. However, when excluding the section with lateral velocity changes from considerations, the CNN trained on the combined datasets satisfactorily predicts both settings. Overall, we observe that the training on dispersion spectra proves more accurate, with the only exception being the prediction on the right model side using the CNN trained on shot gathers with velocity inversion.

Table 3: Average RMSE between predictions and label for different sections of the model in Figure 4.4 using CNN models trained on different data. The left error is calculated for the shot locations up to 40 m, the middle error for shot locations between 50 m and 80 m, and the right error for shot locations between 90 m and 120 m. The total error is the average RMSE across all shots.

Training on shot gather				
	Left Error	Middle Error	Right Error	Total Error
w/o inversion	8.3	15.6	10.1	11.7
w/ inversion	13.4	10.7	7.0	10.4
combined	10.1	13.3	9.1	11.0
Training on dispersion spectrum				
w/o inversion	3.4	9.0	8.7	7.2
w/ inversion	2.9	8.8	7.3	6.5
combined	2.5	7.9	7.2	6.0

Given that our different CNN models are trained on synthetic, noise-free data, we also conducted tests on noisy data to evaluate their generalization capabilities. For this, Gaussian noise is added to the shot gathers of our benchmark model, creating testing data with Signal-to-Noise ratios (SNR) of 10 and 5. To test our assumption from Subsection 4.1 that dropout increases the generalization of our CNN models, we retrain the CNN models with dropout rates of 0.15 and 0.3.

For this, we only consider the CNNs trained on the combined dataset (with and without velocity inversion). For displaying purposes, we will only show the results for the shot gather locations at 10 m and 120 m, as the prediction on the model section with lateral variations has shown to be outside the scope of the prediction capabilities. For an SNR of 10 (Figure 4.8), we find that the prediction quality for the CNN trained on shot gathers (middle column) significantly decreases without making use of dropout. Comparing the predictions for dropout rates of 0.15 and 0.3, we observe that the lower dropout rate of 0.15 yields better results. This improvement is particularly evident for the shot gathers modelled for the benchmark model section with velocity inversion (lower row). The predictions for the CNN trained on dispersion spectra (right column) are similar regardless of the dropout rate, and are comparable to the predictions obtained on shot gathers without noise (right column in Figure 4.7). Comparing the underlying dispersion spectra of Figures 4.7 and 4.8, it is evident that the added noise in the shot gather has minimal influence on the quality of the dispersion spectra due to the random distribution of the Gaussian noise. Consequently, the prediction quality remains largely unaffected.

For a lower SNR of 5 (Figure 4.9), the quality of predictions for the CNN trained on shot gathers (middle column) drastically decreases, independently of the used dropout rate. The predictions for the CNN trained on dispersion spectra continue to show good results, despite a noticeable degradation in the quality of the corresponding dispersion panels compared to those in Figures 4.7 and 4.8.

Table 4 summarizes the average RMSE between the predicted dispersion curves for noisy shot gathers with SNR of 10 and 5 and the labels. The error is the average RMSE for all shots across all the sections of our benchmark model.

Table 4: Average RMSE between predictions and labels using different dropout rates obtained for noisy shot gathers with SNR of 10 and 5. The CNN models are trained on the combined dataset. The total error is the average RMSE for all shots.

	Training on shot gathers		Training on dispersion spectra	
Dropout rate	SNR 10	SNR 5	SNR 10	SNR 5
0	19.0	30.3	6.9	14.2
0.15	12.9	40.1	9.0	12.3
0.3	17.1	25.9	8.8	12.9

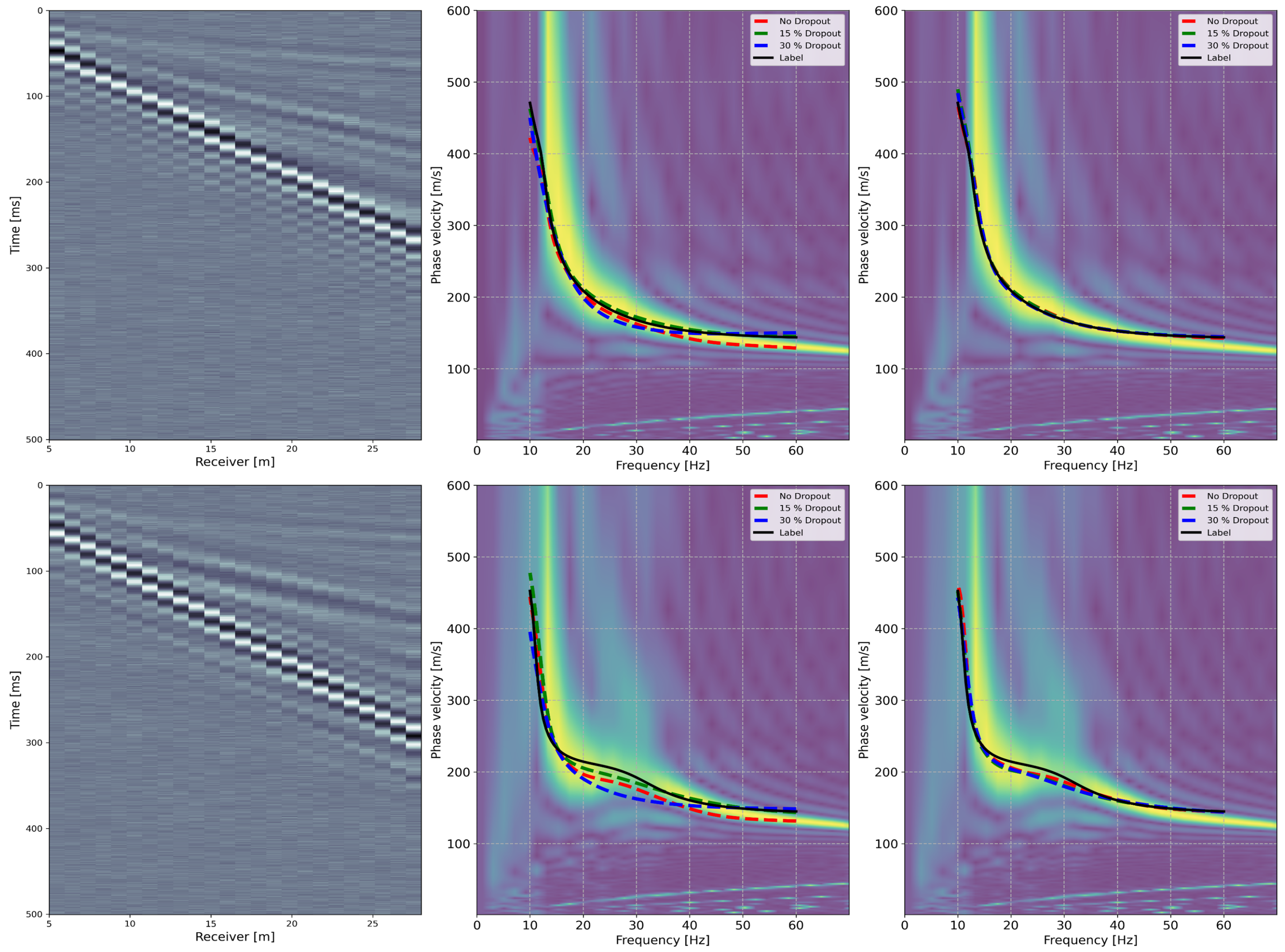

Figure 4.8: **Left column:** Noisy common-shot gathers generated using the velocity model in Figure 4.4 with the sources located at 10 m and 120 m (top and bottom rows, respectively) with Gaussian noise added to achieve an SNR of 10. **Middle column:** The corresponding dispersion spectra, calculated using the PS method and the predictions of the CNN trained on the shot gathers. **Right column:** Predictions of the CNN trained on dispersion spectra. The solid black curve represents the dispersion curve for the velocity model extracted at the midpoint of the shot gather (label). The coloured, dashed curves represent the predictions for the CNN using different dropout rates.

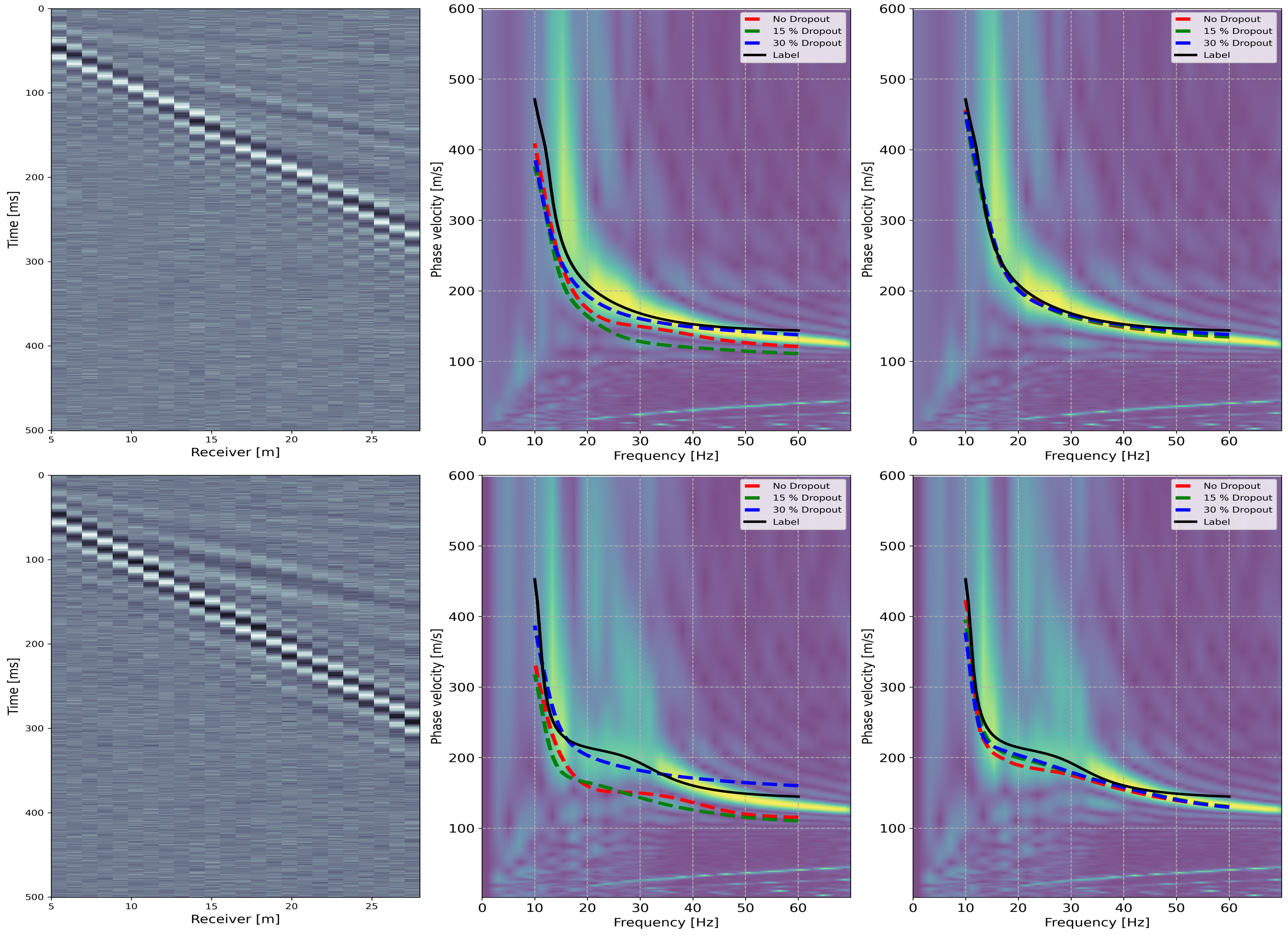

Figure 4.9: **Left column:** Noisy common-shot gathers generated using the velocity model in Figure 4.4 with the sources located at 10 m and 120 m (top and bottom rows, respectively) with Gaussian noise added to achieve an SNR of 5. **Middle column:** The corresponding dispersion spectra calculated using the PS method and the predictions of the CNN trained on shot gathers. **Right column:** Predictions of the CNN trained on dispersion spectra with different dropout rates. The solid black curve represents the dispersion curves for the velocity model extracted at the midpoint of the shot gather (label). The coloured, dashed curves represent the predictions for the CNN using different dropout rates.

4.2.2 Higher Mode Model

The second geological scenario is taken from Socco and Strobbia (2004) (Figure 6b therein) and consists of a simple 5-layer model, with v_s increasing by 100 m/s every layer from an initial v_s of 100 m/s (4.10, left). Figure 4.10 presents the common-shot gather generated using this velocity model (middle) and the corresponding dispersion spectra (right), including the theoretical dispersion curves of the fundamental mode (solid black line) and the first higher mode (solid red line). We extended the recording time of our synthetic shot gathers to 1 s to ensure the entire surface-wave train is captured. This model produces a jump of the dispersion maxima to the first higher mode between 10 and 20 Hz (bright circle in Figure 4.10, right), which complicates the conventional picking process.

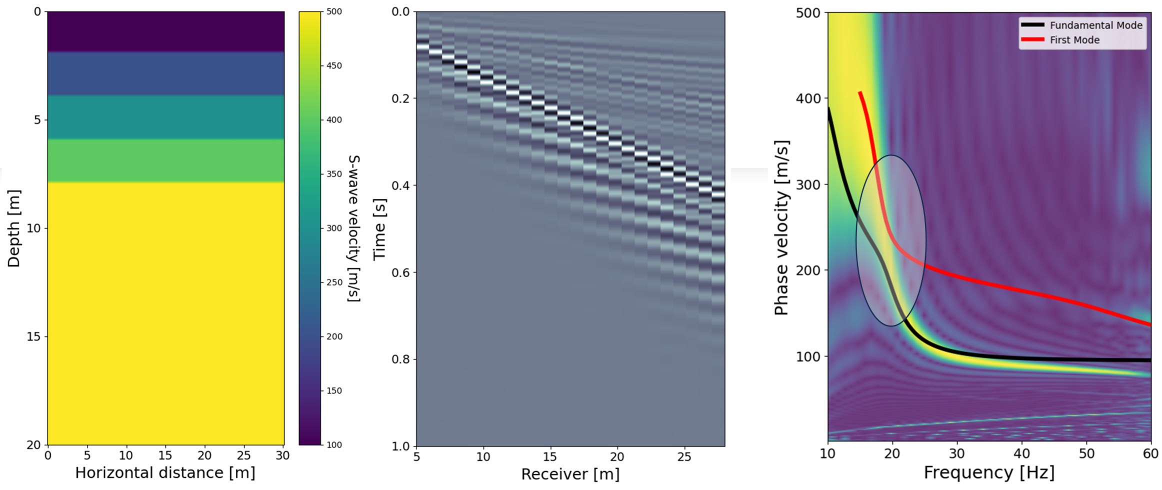

Figure 4.10: **Left**: Shear wave velocity model from Figure 6b in Socco and Strobbia (2004). **Middle**: Common-shot gather generated for the velocity model on the left. **Right**: Corresponding dispersion spectra and the theoretical dispersion curves of the fundamental mode (black solid curve) and the first higher mode (red solid curve). The bright circle indicates the area where the maximum energy in the dispersion spectra jumps to the first higher mode for lower frequencies. Note the deviation of the fundamental mode to the underlying dispersion spectra for frequencies above 30 Hz.

For this example, we extended the training of the CNN to predict both the fundamental and the first higher mode. The parameters used for the random model generation are listed in Table 5. We choose these parameters because a minimum velocity $v_{s1,min}$ of 70 m/s allows for a grid spacing of 0.2 m, which in turn results in a modelling time step of 0.5 ms (see Equation 3.6 and 3.5). Although this increases the computational costs compared to the FD settings used for the random model generation for the benchmark model used previously, they remain within reasonable limits.

Table 5: Modelling parameters used for the random model generation for models with high-energy first higher mode, inspired by Figure 6 from Socco and Strobbia (2004). The resulting velocity profiles and dispersion curves are shown in Figure 4.11

$v_{s1,min}$	$v_{s1,max}$	g	n	z_{max}	d_{min}	d_{max}
70 m/s	200 m/s	75 m/s/m	5	30 m	0.5 m	3 m

Figure 4.11 shows the dispersion curves for the fundamental (left) and first (middle) higher modes calculated from the generated models and the corresponding v_s profiles (right). One fundamental problem we encounter when including higher modes is the need for the same label dimension across all training data. Different velocity profiles result in different cut-off frequencies for the first mode. Specifying a priori the frequency range for which the first higher mode is calculated can lead to different label dimensions. We train our network to predict the first higher mode between 15 and 60 Hz. After removing models with mismatching label dimensions, only 3589 out of the original 10000 models remain. To account for the smaller dataset, we adopt a batch size of 4.

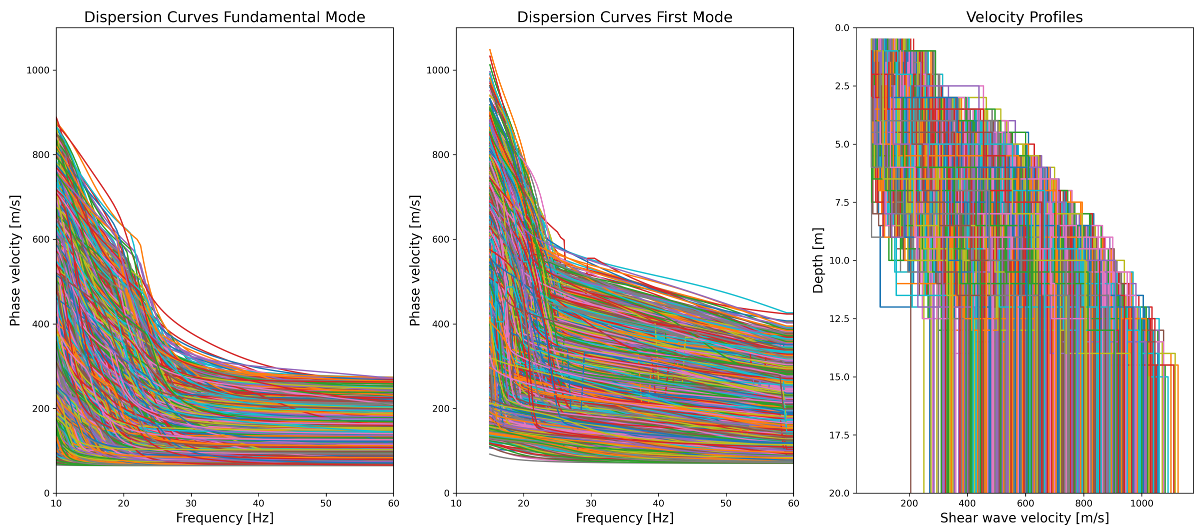

Figure 4.11: **Left**: Fundamental-mode dispersion curves of the randomly generated 1-D geological models used as the training dataset for the prediction of higher modes. **Middle**: First mode dispersion curves of the randomly generated 1-D geological models used as the training dataset for the prediction of higher modes. **Right**: The corresponding v_s profiles.

We repeat the previous experiments of adding Gaussian noise to the shot gathers of our reference model in Figure 4.10, generating test shot gathers with SNR of 10 and 5. Based on our previous results, we also implement dropout with a rate of 0.15 into our CNN. The predictions for our CNN trained on shot gathers or dispersion spectra are shown in Figure 4.12.

We observe that the predictions on our noise-free reference shot gather are very similar for both CNN, trained on shot gather and dispersion spectra (Figure 4.12, top row). While

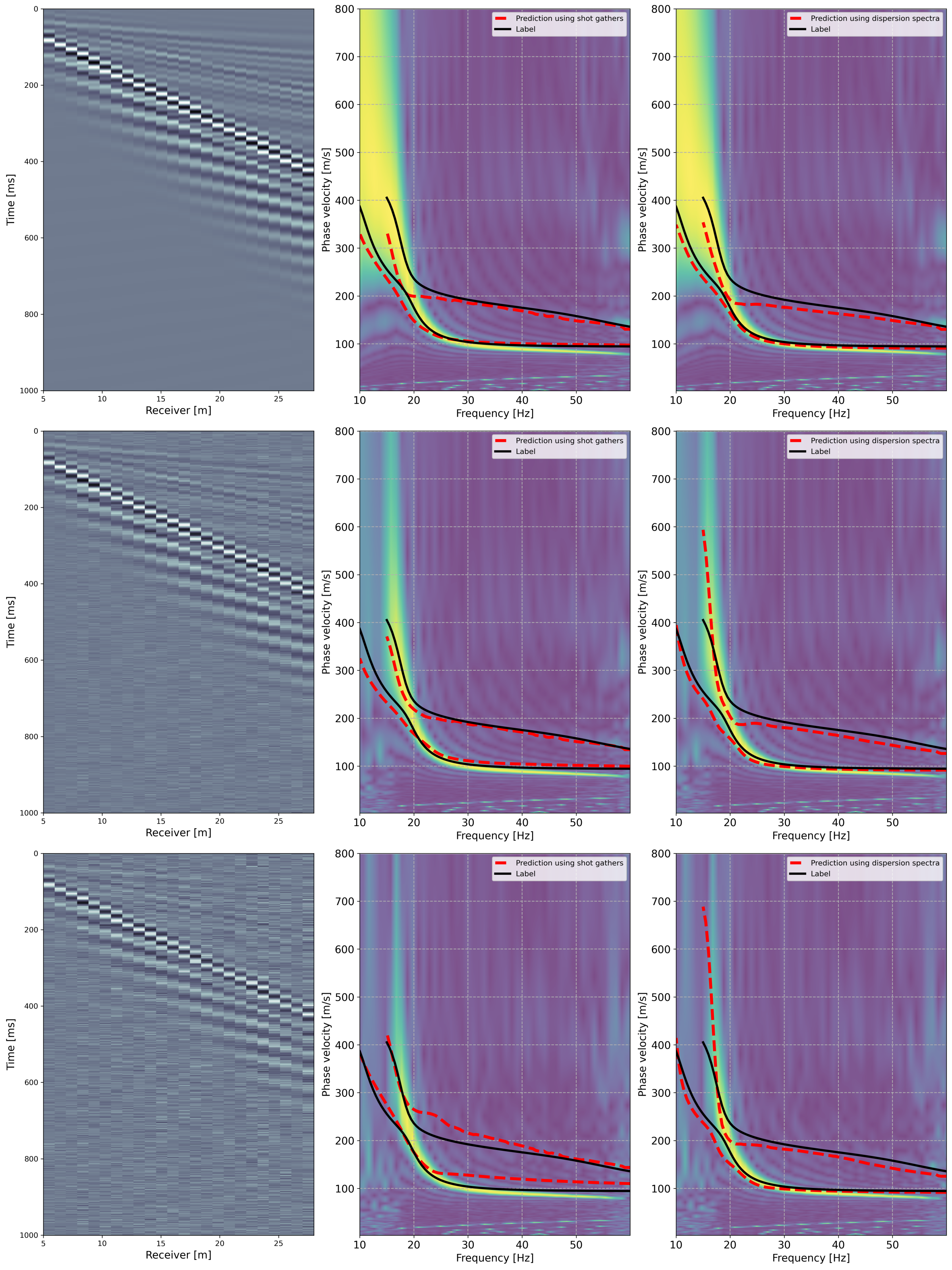

Figure 4.12: **Left**: Common-shot gathers generated using the velocity model in Figure 4.11 with no noise, a SNR of 10 and a SNR of 5 (from top to bottom, respectively). **Middle**: Their corresponding dispersion spectra, calculated using the PS method. The dashed red curves represent the predictions for the CNN trained on shot gathers. The solid black curves represent the fundamental and first higher mode for a velocity model extracted at the midpoint of the shot gather. **Right**: Predictions of fundamental and first higher mode for the CNN trained on dispersion spectra.

the fundamental is well recovered in both cases, the velocity of the first higher mode is underestimated for lower frequencies from 15 Hz to 25 Hz. For an SNR of 10 (Figure 4.12, middle row), the prediction quality for the CNN trained on shot gathers increases, particularly for the first higher mode between 15 Hz and 20 Hz compared to the prediction for the noise-free shot gather. In contrast, while recovering the fundamental mode, the CNN trained on the dispersion spectra shows large deviations for low frequencies of the first higher mode. In the case of stronger noise with an SNR of 5 (Figure 4.12, lower row), the prediction of the CNN trained on shot gathers starts to overestimate the phase velocities for the fundamental and higher modes from 25 Hz and 20 Hz, respectively. For the CNN trained on dispersion spectra, we observe that the prediction of the fundamental mode remains consistent, as previously noted in the preceding subsection. However, the prediction for the first higher mode between 15 Hz and 20 Hz further decreases, which coincides with the change in the underlying dispersion spectrum. Additionally, since higher modes are less evident in the dispersion spectra, it is expected that the prediction of the first mode for the CNN trained on dispersion spectra would produce poorer results than for the fundamental mode. In Table 6 the RMSE for the fundamental and first higher mode are summarized.

Table 6: RMSE for the predictions of the fundamental and first higher modes displayed in Figure 4.12.

	Training on shot gathers		Training on dispersion spectra	
SNR	Fund. mode	First mode	Fund. mode	First mode
∞	15.6	28.0	11.4	25.1
10	16.6	15.2	11.6	32.3
5	18.3	20.1	13.4	51.8

4.3 Field Data Results

After evaluating the proposed methodology on synthetic data, in this subsection, we present the results obtained when applying the methodology to field data. In contrast to the previously added Gaussian noise with known statistical properties, field data comes with noise that can not be characterized a priori, such as environmental noise and noise due to badly coupled or defective receivers.

Additionally, in our synthetic experiments, we make use of a fixed source signature, e.g., a wavelet with known frequency and phase content. In the field, the source signature is influenced by the surface and subsurface properties. Furthermore, the repeatability of the seismic source is not guaranteed, depending on the field operators acquiring the data, and thus possibly varying significantly between shots on the same acquisition line. To address this, we generated multiple synthetic datasets with different source signatures for the training of our CNN. We apply a fixed dropout rate of 0.15 to the training of all our CNNs used for the prediction of field data.

4.3.1 Wallingford Field Data

The Wallingford data consists of three seismic lines acquired by Fugro for MASW for testing purposes. The dataset was acquired on a sports field next to the Fugro office in Wallingford. The area was grass-covered and mostly flat. The line analyzed in this thesis was acquired using a 24-channel landstreamer with 2 m geophone spacing and a constant minimum source offset of 2 m. A total of 61 shots were acquired every 2 m in a roll-along fashion, where the receivers were moved simultaneously with the source, resulting in a total profile length of 170 m. A sledgehammer striking vertically on a steel plate was used as the seismic source. At every shot location, three shots were stacked to increase the SNR. The recording length was 0.5 s with a sampling frequency of 8 kHz.

Since the data was already processed and inverted by Fugro to obtain a v_s model of the site, we use these results to guide our random model generation. The inversion resulted in a v_s velocity of around 200 m/s for the uppermost layer, increasing up to 900 m/s for a depth of 30 m. For a smoother increase in velocity, we generate our velocity models with a fixed layer thickness of 1 m. To allow for more variation in the generated models, we randomly vary the maximum gradient between 10 m/s/m and 40 m/s/m, with a minimum $v_{s1,min}$ of 100 m/s and maximum $v_{s1,max}$ of 400 m/s for the first layer, respectively. This also helps to prevent overfitting the training data to the prior knowledge of the expected velocity profile. The parameters for the random model generation are listed in Table 7. To ensure reasonable computational time, we assume a homogenous half-space below 30 m depth.

Table 7: Modelling parameters used for the random model generation for the prediction on field data. The resulting velocity profiles and dispersion curves are shown in 4.13

$v_{s1,min}$	$v_{s1,max}$	g	n	z_{max}	d_{min}	d_{max}
100 m/s	400 m/s	10-40 m/s/m	30	30 m	1 m	1 m

We visually analyzed the $f - v$ spectra and amplitude spectra of all 61 shots to determine the appropriate frequency range for predicting the dispersion curves. All shots contain sufficient energy in the frequency range between 20 Hz and 50 Hz, wherefore we limited our CNN predictions to this range. The theoretical dispersion curves and v_s profiles used for training are shown in Figure 4.13.

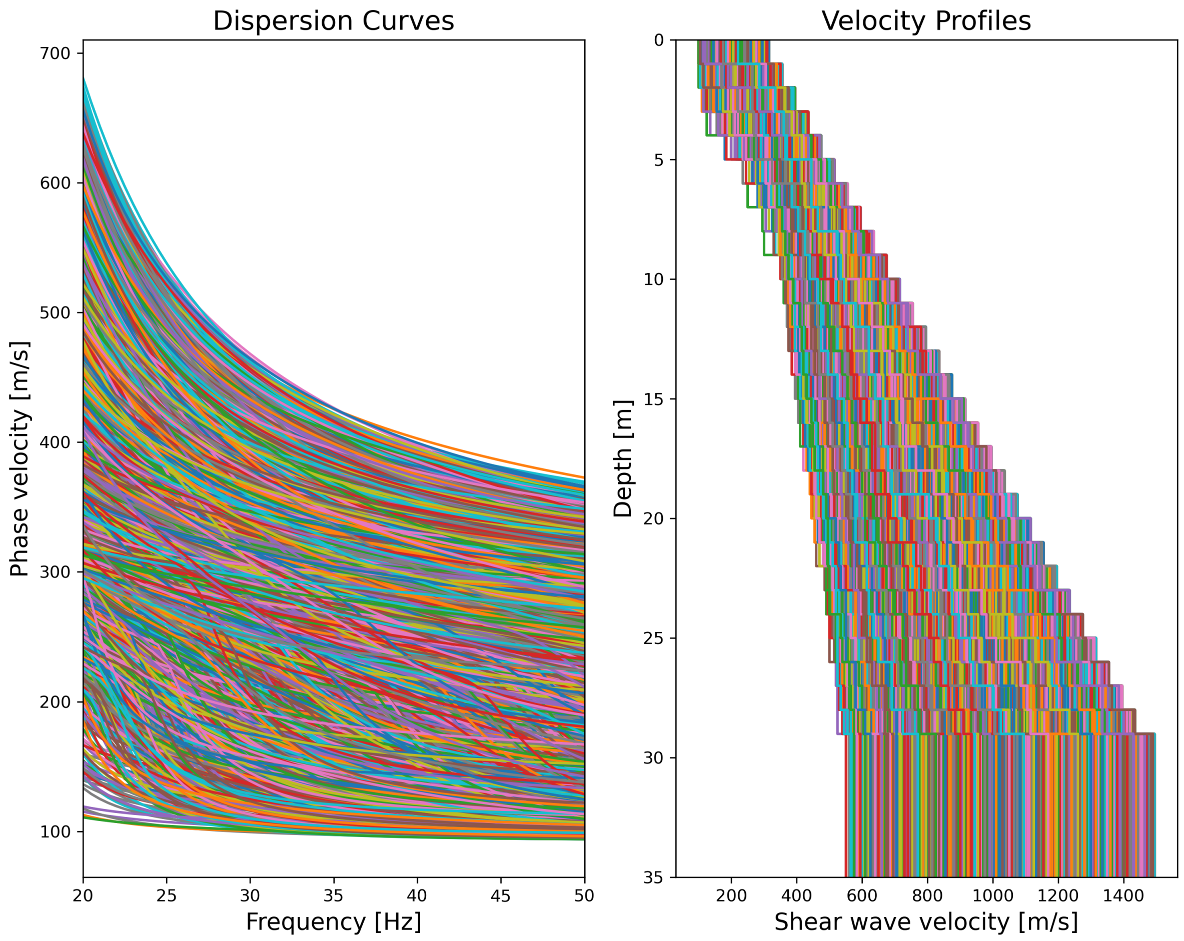

Figure 4.13: **Left**: Fundamental-mode dispersion curves of the randomly generated 1-D geological models used as the training dataset for the prediction on field data. **Right**: The corresponding v_s velocity profiles.

We use the traditional approach of picking the maximum amplitude in the $f - v$ spectrum and correcting for outliers to extract dispersion curves from the field shot gathers. These dispersion curves serve as labels for each field shot gather. Subsequently, we use these labels to evaluate the accuracy of the predictions made by our convolutional neural networks (CNNs).

Since our approach is based on the structural similarity of images (either shot gather or dispersion spectrum), the wavelet used for the forward modelling of our synthetic shot gathers potentially impacts the quality of the predictions on field data. Therefore, we use three different wavelets to model shot gathers for the randomly generated velocity profiles.

The first wavelet is a flat-spetrum wavelet with frequencies of 5-10-60-70 Hz, covering the frequency content of the field data.

For the second wavelet, we estimate a zero-phase statistical wavelet from the field data. For this, we autocorrelate the second trace for all shots (as the first trace shows strong low-frequency noise), calculate the square root of the amplitude spectrum of each autocorrelation, and stack the resulting amplitude spectra (Edgar and Van Der Baan, 2011). Using the inverse Fourier transform, we obtain our zero-phase statistical wavelet, which we then smooth using a Gaussian window. Finally, we apply a low-pass filter to only include frequencies up to 80 Hz, to ensure that the stability and numerical-dispersion criteria are met.

As the third wavelet, we use a minimum-phase wavelet adjusted to our field data. At first, we attempted to supply the minimum phase to our zero-phase statistical wavelet following the steps by Cui and Margrave (2014) and Margrave and Lamoureux (2018) by calculating

$$W(f) = |W(f)| \, e^{i\phi_m(f)}, \tag{4.2}$$

where the minimum-phase $\phi_m(f)$ is calculated by

$$\phi_m(f) = H\left(\ln |W(f)|\right). \tag{4.3}$$

In Equations 4.2 and 4.3, $W(f)$ is the Fourier transform of our desired minimum-phase wavelet $w(t)$, $|W(f)|$ the square root of the amplitude spectrum of the autocorrelated signal, and H the Hilbert transform. While this procedure is effective when applied to the unfiltered statistical wavelet, it proved unsuccessful after filtering. This was due to the significant high-frequency content (up to 300 Hz) present in the unfiltered statistical wavelet, which was necessary to be filtered to avoid numerical dispersion for the FD modelling. Instead, we approximate the amplitude spectrum with a Ricker wavelet with a peak frequency of 50 Hz, matching with the peak frequency of the amplitude spectrum of the statistical wavelet. We then calculate the minimum-phase equivalent with Equations 4.2 and 4.3 and low-pass filter the minimum-phase wavelet to only contain frequencies up to 80 Hz. We also attempted to extract a direct estimate of the source signature from a second line acquired at the Wallingford test site, where shots were acquired next to receivers. However, this was not successful as these traces were contaminated with strong noise. The wavelets used for the FD modelling of our shot gathers are shown in

Figure 4.14.

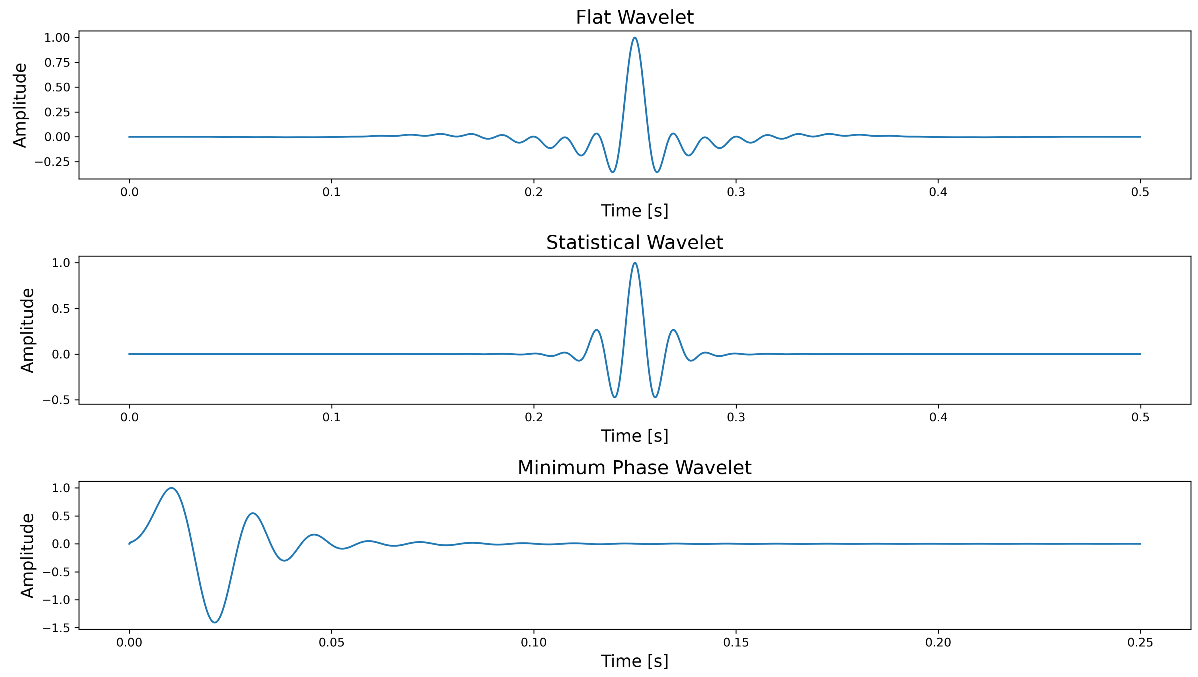

Figure 4.14: The different wavelets used for the forward modelling of the synthetic shot gathers for the training of the CNN. All wavelets are filtered to contain only up to 80 Hz due to computational constraints. **Top**: Flat-spectrum wavelet with frequencies of 5-10-60-70 Hz. **Middle**: Statistical wavelet estimated from the second trace of all field shot gathers. **Middle**: Minimum-phase transformed Ricker wavelet with a peak frequency of 50 Hz.

Figure 4.15 shows the predictions for the CNN trained on shot gathers, modelled with the different wavelets. Out of the 61 shots, we select 2 shots on which the prediction quality is good and 2 shots where the predictions perform poorly. We observe no systematic improvement in the predictions using a particular wavelet, and different wavelets performed better on different shots. Overall, the predictions are mostly constrained to a 50 m/s interval around the label, with the exception being low frequencies of 20 Hz to 25 Hz, where errors of up to 200 m/s are observed more regularly (as seen in Figure 4.15). The results obtained using the different estimated wavelets do not confirm our assumption that the wavelet choice largely impacts the results when estimating dispersion curves from field data. As we were not able to directly measure the source signature, future research should investigate the influence of using a more accurate estimate of the source signature for the forward modelling. If our results are confirmed, strong limitations of the proposed method will be removed, as a generic flat-spectrum wavelet can be utilized across different datasets. Therefore, the expensive process of estimating the source signature will not be required.

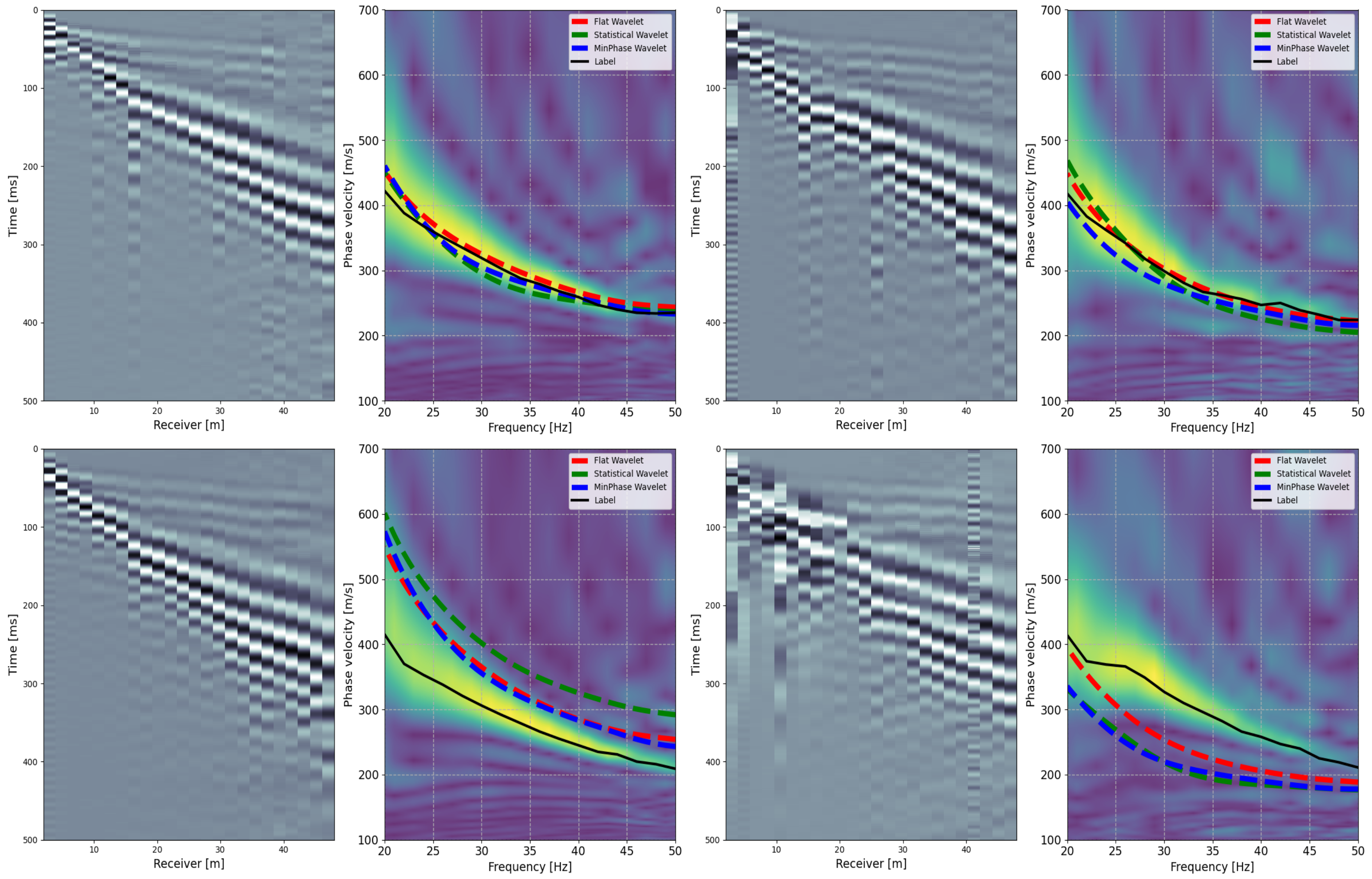

Figure 4.15: Exemplary results for the Wallingford field data using different wavelets to model the training data. **Top**: Common-shot gathers and their corresponding dispersion spectra where the predicted dispersion curves show good agreement with the dispersion spectra. **Bottom**: Common-shot gathers and their corresponding dispersion spectra where the predicted dispersion curves show strong deviations from the underlying dispersion spectra. The solid black curve represents the conventionally obtained label. The coloured, dashed curves represent the predictions for the CNN trained with different wavelets.

Since the resulting dispersion spectra for shots modelled with different wavelets are very similar, we only train the CNN on the dispersion spectra obtained from the shot gathers modelled with the flat-spectrum wavelet. The predictions for the CNN trained on dispersion spectra are shown in Figure 4.16. Compared to the selected predictions using shot gathers as training data in Figure 4.15, we observe that the predictions are much closer to the labels and match the underlying dispersion panel. This coincides with our observations made for the synthetic examples, which proved to be less sensitive to random noise, and, in this case, also to actual field noise. Additionally, it is important to keep in mind, that no estimate of the wavelet is needed, as we just used a flat wavelet with a suitable frequency range.

The average RMSE for the predictions using shot gathers modelled with various wavelets and using dispersion spectra as training data is presented in Table 8. Additionally, a visual inspection was conducted, since traditional picking would also be based on visual evaluation of the dispersion spectrum. Based on this assessment, the predictions were categorized as sufficient or insufficient, with the respective percentages listed in Table 8. Our model trained on dispersion spectra has the smallest RMSE and the highest percentage of accepted predictions.

Table 8: RMSE for the prediction on the Wallingford field data using shot gathers modelled with different wavelets and dispersion spectra.

	RSME	% Acceptable
Flat Wavelet	41.3	27.8
Statistical Wavelet	39.0	14.7
MinPhase Wavelet (7x7)	36.9	21.3
MinPhase Wavelet (3x7)	27.8	34.4
Transfer Learning	22.3	34.4
Dispersion Spectra	11.5	93.4

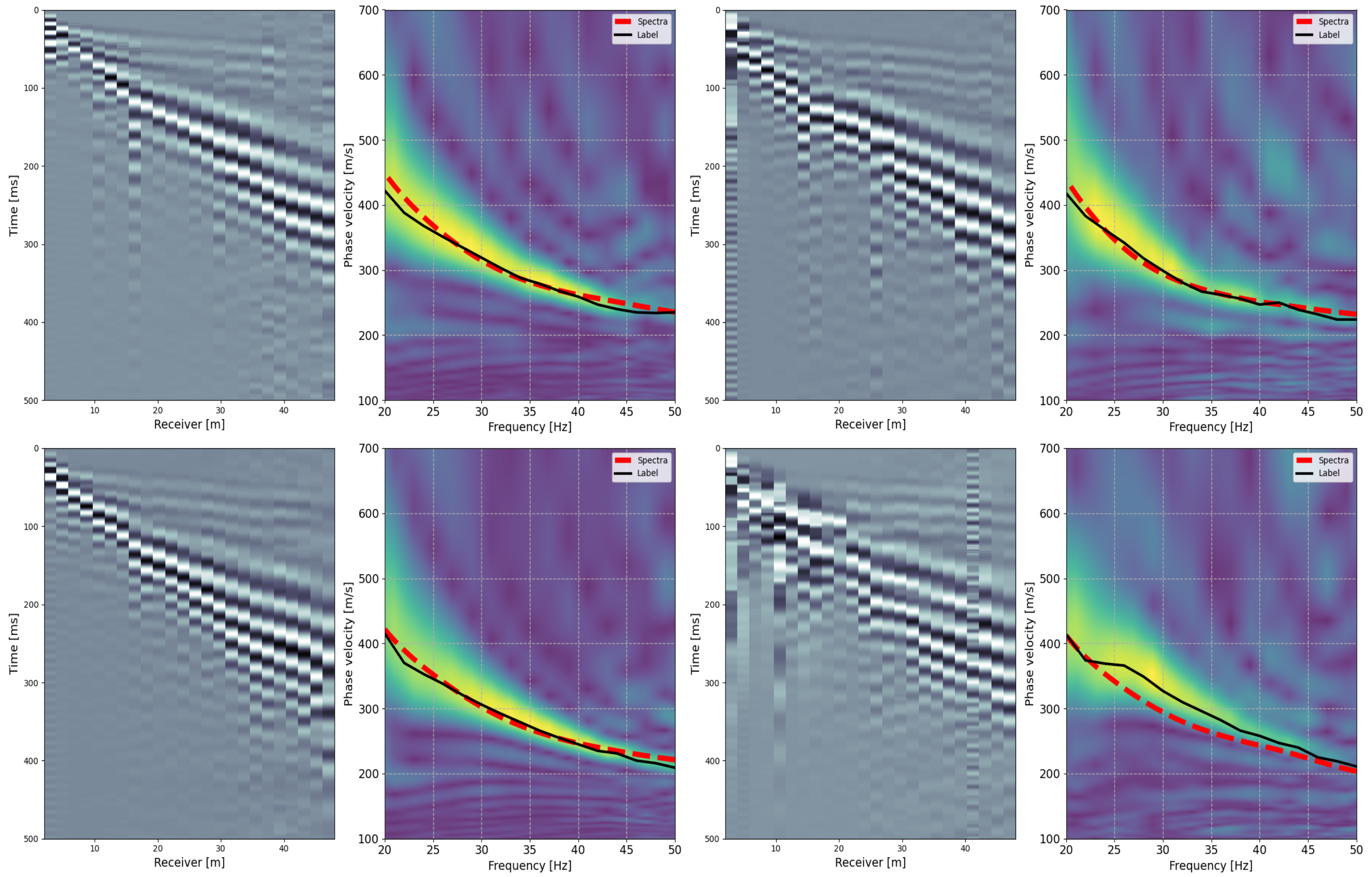

Figure 4.16: Exemplary results for the Wallingford field data using dispersion spectra as training data. The same shot gathers and their corresponding dispersion spectra as in Figure 4.15 are displayed. The solid black curve represents the conventionally obtained label. The red dashed curve represents the prediction for the CNN trained in dispersion spectra.

4.3.2 CNN Architecture Modifications

The results presented in this subsection stem from considerations that arose after obtaining the previous results. The CNN architecture employed in this thesis is a slightly modified version of ResNet-18. As illustrated in Figure 3.3, we use a 7x7 kernel in the first convolutional layer. Considering only 24 receivers were modelled, this kernel size might not be optimal for the application on shot gathers. Employing a too-large filter size in the spatial dimension, coupled with a stride of 2, may be inadequate to properly capture the spatio-temporal characteristics of the geometric dispersion of surface waves. Therefore, we propose using rectangular kernels to account for the difference in space and time dimensions, testing a kernel size of 3x7 for the CNN training on shot gathers.

Given that the minimum phase wavelet shows the lowest RSME and the intermediate percentage of acceptable results among the different wavelets, we train the modified architecture on the shot gathers modelled with this wavelet.

Figure 4.17 shows the results for using the original 7x7 kernel size (red dashed line) and the proposed 3x7 kernel size (green dashed line) for the same four selected shot gathers as in Figure 4.15. We find that training on the modified kernel size for the first layer leads to clear improvements, particularly for the shot gathers in which previously the prediction performed poorly (bottom row in Figure 4.15). This is also evident in Table 8, where we observe a decrease in average RSME and an increase in acceptable predictions.

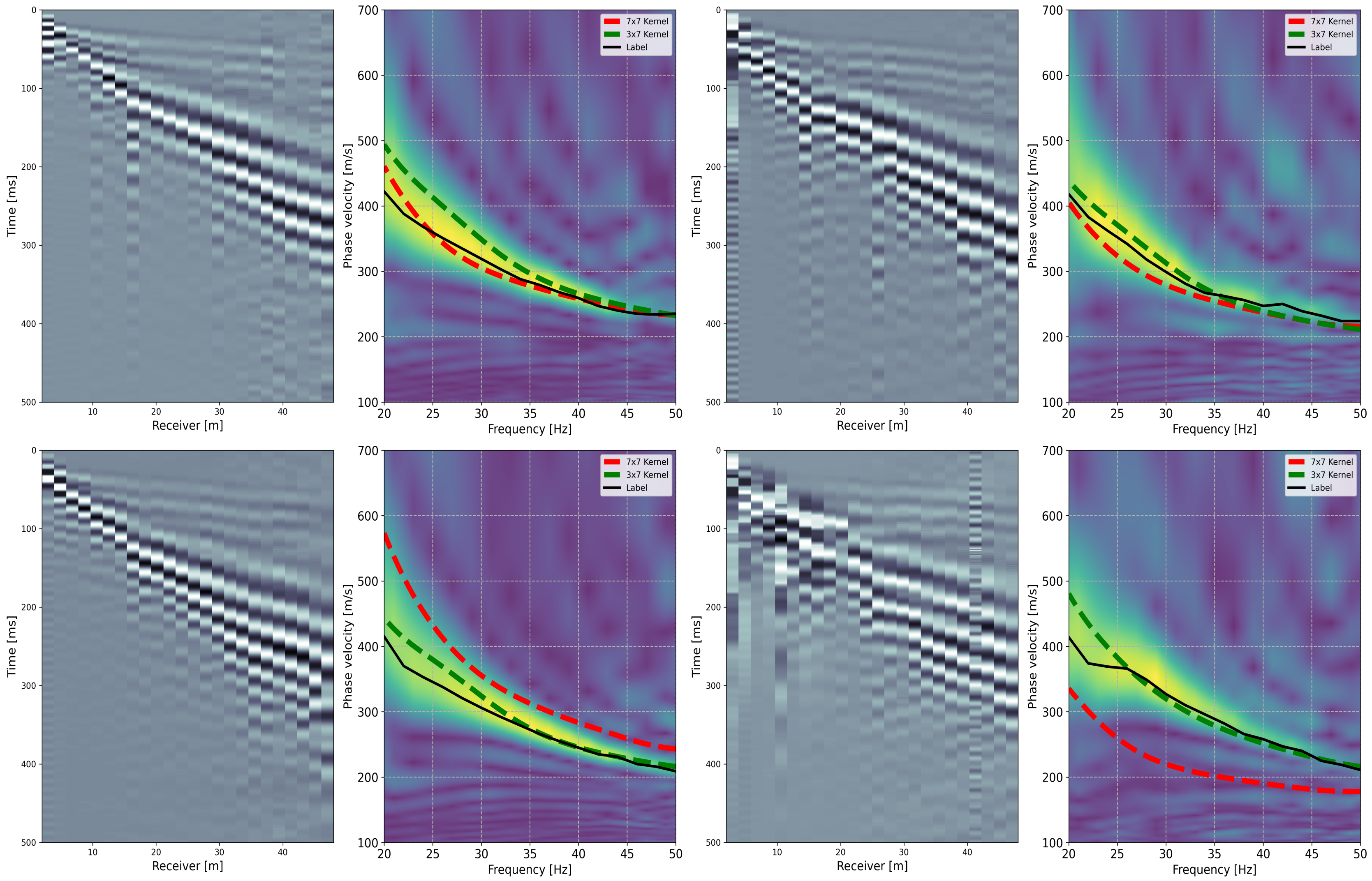

Figure 4.17: Exemplary results for the Wallingford field data using shot gathers modelled with the minimum-phase wavelet as training data. The same shot gathers and their corresponding dispersion spectra as in Figure 4.15 are displayed. The solid black curve represents the conventionally obtained label. The red dashed line represents the prediction for the CNN with the original 7x7 kernel size in the first convolutional layer, whereas the green dashed curve is predicted using a modified kernel size of 3x7.

4.3.3 Transfer Learning

In the previous sections, our CNN is trained solely on synthetic data. To enhance the model's performance and include parts of our field data, we apply transfer learning. Transfer learning is a technique that aims to improve an already trained model by re-training it with data of the target domain (Zhuang et al., 2020). We sample 25 % of our field data (excluding the ones we chose for displaying) by using every fourth shot. Randomly sampling our field data leads to undersampling certain parts of our seismic line, which decreases the prediction quality of those sections due to the retrained model overfitting to the other sections. To further decrease the possibility of overfitting the training data, we only allow updating the parameters of the last convolutional layer (blue blocks in Figure 3.2) and the fully-connected layer. We retrain our previously obtained CNN model with the modified kernel from Subsection 4.3.2. The results for the same shot gathers as in Figure 4.17 are shown in Figure 4.18. The dashed green line indicates the predicted dispersion curve using transfer learning. For comparison, the dashed red line indicates the predicted dispersion curve results without transfer learning. While the average RMSE decreases and the percentage of acceptable predictions stays constant (see Table 8), we observe that some estimations (e.g., Figure 4.18, top right) are worse after transfer learning, most likely due to a poorer prediction quality in the undersampled sections of our seismic line.

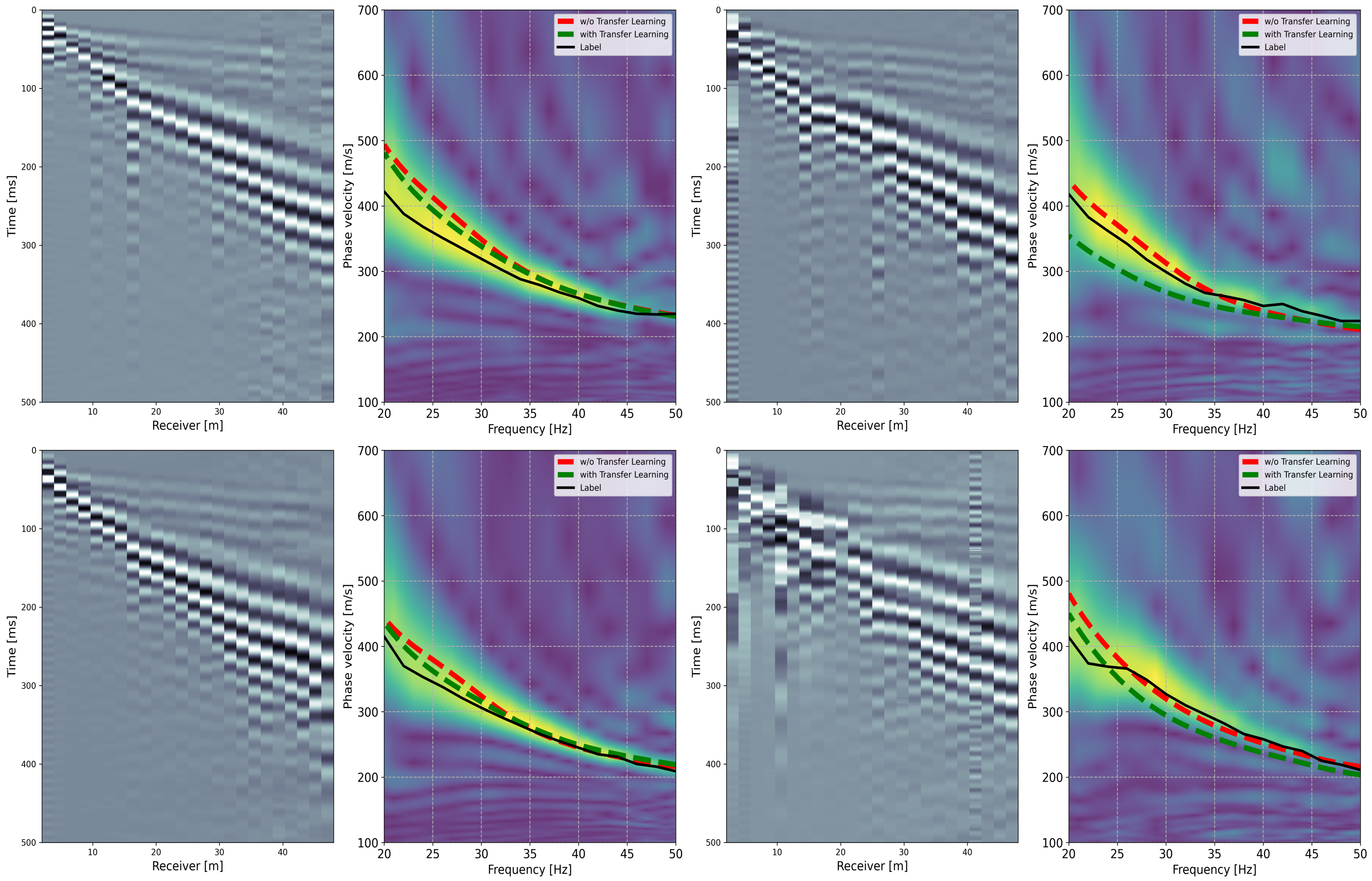

Figure 4.18: Exemplary results for the Wallingford field data using shot gathers modelled with the minimum-phase wavelet as training data. The same shot gathers and their corresponding dispersion spectra as in Figure 4.15 are displayed. The solid black curve represents the conventionally obtained label. The red dashed line represents the prediction for the CNN with a 3x7 kernel size without making use of transfer learning, whereas the green dashed curve is predicted using transfer learning with every 4th shot of the field dataset.

5 Discussion

In this section, we discuss and analyze the effectiveness and limitations of our approach based on the results obtained in the previous section. We examine the forward modelling and model generation techniques, the differences observed between training on shot gathers versus dispersion spectra, and analyze sources of error encountered during the study. Additionally, we address computational considerations that impacted our methodology. We also propose potential optimizations and areas for future research.

5.1 Forward Modelling and Model Generation

We show that an approach using only synthetic data to train a CNN can successfully be applied to field data. Nevertheless, certain assumptions led to less realistic synthetic models. Addressing this could enhance the generated models and improve the method's performance. First of all, we use an elastic FD scheme to generate our synthetic seismic data. However, real media might not strictly fulfil the assumption of pure elasticity and require the implementation of a visco-elastic FD scheme. Using the visco-elastic FD scheme to capture the attenuation of higher frequencies with source distance properly might lead to increased reliability of the CNN, particularly when trained directly on shot gathers.

The use of tapered boundaries in our experiments does not always completely suppress reflections from the model domain boundaries. More tapering points were judged as too computationally demanding. This issue can be observed in Figure 4.12 for the noise-free shot gather (upper row), where low-amplitude artefacts with a positive slope can be observed. The implementation of PML is an alternative that could improve both, computational cost and reduction of side reflections. While modelling visco-elastic media and using PML boundaries can lead to improvements in the synthetic data, the actual effect on the prediction quality of the CNNs is subject to future research.

We have also shown that combining different training datasets can extend the prediction to new geological conditions. However, the prediction quality of the model may degrade with the combined datasets compared to using a more tailored dataset.

One fundamental requirement for the random model generation is prior knowledge of the area to which the method is applied. In this work, significant time has been spent selecting the parameters for the random model generation for the different velocity settings. To make this approach feasible for its application in industry, it is essential to develop routines that make the random model generation more effective, otherwise, time savings over conventional methods cannot be attained. For this, the automatic creation of specialized datasets, for example, based on different gradients or the geologies of different areas, could tackle this issue. In practice, all models could be applied to the field data, and based on a metric measuring the quality of the prediction, the best model is selected. This would indicate the geological setting even if no prior knowledge is available. Alternatively, a preliminary analysis could be carried out to select an appropriate model to carry out the prediction on the whole dataset, similar to what was done in this thesis.

5.2 Training on Shot Gathers versus Dispersion Spectra

Our findings support Chamorro et al. (2024) results indicating that CNNs can be applied to predict dispersion curves directly from shot gathers. We identified and addressed certain limitations in their work and extended it by incorporating more variability in the training datasets, testing the influence of noise, and comparing the training on shot gathers and dispersion spectra. The estimation of dispersion curves from field data using shot gathers as training data is currently insufficient for industrial application. On the other hand, we find that using dispersion spectra rather than shot gathers for training CNNs leads to more accurate estimations on field data and on noisy synthetic shot gathers. Only when predicting the first higher mode, we observe that the change in the dispersion spectrum due to increasing noise leads to a strong overestimation of the phase-velocity for the first higher modes for low frequencies (Figure 4.12). The conventional approach for dispersion curve picking is based on analyzing the energy distribution of the dispersion spectra since they are more easily interpretable and less visually complex than the corresponding shot gathers. This reduction in complexity also translates to the prediction quality using CNNs, where the task of learning the correlation between maxima in the dispersion spectrum and the dispersion curve leads to better results.

For the training on shot gathers and dispersion spectra, different preprocessing steps can be applied to influence the outcome of the training and prediction process. The PS method transforms the shot gathers to the phase-velocity spectra. The size of the phase-velocity and frequency bins can impact the dispersion spectrum and thus the training and prediction process. These can be adjusted by applying zero-padding to produce sufficient resolution. As discussed by Rahimi et al. (2021), the choice of transformation technique depends on the site conditions and strongly influences the quality of the dispersion spectra. The FDBF method is recommended to cover most site conditions to achieve high-quality dispersion spectra. Hence, we propose the use of the FDBF method in future research when applying CNNs to dispersion spectra.

Furthermore, we applied our CNN to field shot gathers which were only subjected to trace normalization. On the one hand, characterizing noise and filtering the field data could lead to substantial stabilization of the prediction process by enhancing the similarity of the field with the synthetic data. On the other hand, *data augmentation*, commonly applied in DL, could be utilized to make the synthetic data more similar to the field data. When using data augmentation, the original training data is duplicated and modified, for example, in the case of shot gathers, by muting traces or adding, in the best case, noise with field characteristics (Lii et al., 2022; James et al., 2023). This is an important step since we have shown in synthetic data that increasing noise decreases the prediction quality. At the same time, by using data augmentation the training dataset is substantially increased, and having multiple shot gathers or dispersion spectra with the same label prevents overfitting. Compared to the generation of additional synthetic data, data augmentation is computationally cheap and does not require the use of additional disk space, as it can be implemented directly during the training process.

Instead of adding noise to the training data to make the synthetic data more similar to the field data, in our work, we integrated dropout into the CNNs architecture. We showed that this step increases the generalization capabilities of the model when estimating the dispersion curves from the same shot gather with different noise levels (Figure 4.9). This can be explained by dropout introducing noise to the training process itself, without explicitly adding noise to the training data. By randomly disabling parts of the feature maps, the network is trained on multiple versions of each shot gather, each with a different distribution of disabled neurons, simulating the effect of noise.

For the estimation of the fundamental mode using dispersion spectra as training data, the noise added to the shot gathers only showed minimal influence on the results (right column in Figures 4.8 and 4.9). As the noise we apply is white Gaussian noise, having the same power distribution across all frequencies, integrating over time and space using Equations 2.3 and 2.6 for the PS transform minimizes the influence of the noise on the dispersion spectra. Therefore, this also minimizes the influence of noise on the dispersion curve estimation. Correlated noise can have a stronger influence on the dispersion spectra. Such noises need to be added to the training data or filtered from the field data before dispersion curves are estimated.

Another important consideration is the adaptability of models trained on shot gathers or dispersion spectra to different acquisition geometries. Training on shot gathers requires training data with the same acquisition parameters as the field data, including the number of receivers, receiver spacing, sampling interval, and recording length. Although it is theoretically possible to model synthetic shot gathers with all acquisition parameters oversampled and subsequently resample them to match the actual acquisition parameters before training the CNN, this approach introduces an additional step in the workflow, thereby increasing the complexity of the process. These limitations are less strict when directly training on dispersion spectra, as the phase-velocity and frequency bin size can be easily adjusted after acquisition to remain consistent across different acquisition geometries.

Furthermore, unsupervised machine learning techniques, such as clustering algorithms, are more reliable for applications in the industry where reproducibility is crucial for providing clients with well-founded results. Clustering algorithms align closely with conventional approaches for dispersion curve picking by partitioning the dispersion spectra into clusters of high energy corresponding to different modes, visually coinciding with the manual picks of a professional operator. Additionally, these methods require no assumptions regarding parameters such as Poisson's ratio, $\frac{v_p}{v_s}$-ratio, prior geological knowledge, wavelet used for modelling, dimensional matching, normalization choices, or CNN architecture. Although not presented in this thesis, we carried out preliminary tests with DBSCAN, as described in Rovetta et al. (2021). Our results show that while easily implemented, the tuning of hyperparameters is not trivial in case of stronger noise and closely separated higher modes. A combination of both approaches could be implemented, where the DL approach is utilized to guide the clustering by searching around the predicted dispersion curves.

5.3 Error Discussion

During our research, we encountered two main unresolved challenges.

The first challenge involves the calculation of the higher-mode dispersion curves. Higher modes only exist above certain cut-off frequencies, sometimes resulting in no solution for frequencies within the specified frequency range for the randomly generated models. Therefore, the frequencies at which the theoretical dispersion curves are computed must be determined beforehand, significantly constraining the availability of training data for predicting higher modes and leading to inefficient use of computational resources. Addressing this limitation is crucial for future research to ensure more efficient and accurate computations. Additionally, we faced the problem of root skipping when calculating the theoretical dispersion curves for higher modes, which introduced irregularities in the labels. This can be mitigated, by decreasing the phase-velocity increment for root finding. However, this substantially increases computational costs to impractical levels. We partially addressed this issue by applying a median filter, which improved but did not eliminate these artefacts. Consequently, the challenges of insufficient training data and labelling errors persist, affecting the quality of our estimations of higher modes.

The second challenge relates to the discrepancy between theoretical dispersion curves and the underlying dispersion spectra calculated using the PS method. An alternative implementation of the PS method was tested, yielding identical results. Given that the v_s model for our synthetic data is known, we calculated the minimum Rayleigh-wave velocity, thereby attributing the observed error to the PS transform instead of the dispersion curve calculation. The precise origin of this discrepancy remains unidentified. However, if it represents a systematic problem inherent to the PS method, it suggests a potential underestimation of the phase velocities when selecting the maxima in the dispersion spectra, as done in the conventional approach for dispersion curve picking. Future research should aim to identify the cause of this discrepancy and determine if it is also present when using other transformation methods. Nevertheless, for training on dispersion spectra, this discrepancy does not present an issue, as the true label used for training is independent of the input dispersion spectrum.

5.4 Computational Considerations

The codes we used in this work are not highly optimized and certain changes could greatly increase their efficiency. The computational cost for the random model generation, the forward modelling using *fdelmodc*, as well as the training of the CNN are highly variable, depending on various factors discussed in this subsection.

In our experiments, the generation of 10000 geological models and calculating their corresponding fundamental mode dispersion curve required approximately 20 minutes. Including the calculation of the first higher mode significantly increased this computation time due to the root-finding process. To address this, we reduced the phase-velocity increment by a factor of five. In this case, the computation time was extended to around 7 hours, with root skipping still present.

The FD modelling of 10000 shot gathers for a domain with 30 m horizontal and 50 m vertical distance and 150 taper points, using a grid spacing of 0.25 m, a time-step of 0.3 ms, a recording length of 0.75 s, and a maximum frequency of 70 Hz took around 14 hours. This computational time can further increase depending on the survey design, potentially requiring a larger model domain or adjustment of the grid spacing and time-step according to Equations 3.5 and 3.6 for different velocity ranges and source frequencies. Since the randomly generated models are independent of each other, this process can be parallelized using multiple CPUs. Additionally, there is potential for optimizing the current version of the code. All geological models are initialized with the same grid size and modelled using the same time step. Instead, based on the dispersion and stability criteria in Subsection 3.2.2, each model could be initialized with individual grid sizes and time steps, based in its minimum and maximum velocity ($c_{s;min}$, $c_{p;max}$). This, in addition to parallelization, can make creating much larger datasets feasible.

The training time of the CNN depends on the dataset size, the data dimensions, and the used CNN architecture. Generally, the computational cost can be balanced by using a larger batch size (if the GPU has enough capabilities to do so). With the specified hardware (see Section A.1.1) and CNN architecture (see Section 3.3), training took around 4 hours for 10000 shot gathers (or dispersion spectra) over 200 epochs. Using Early Stopping, training generally stopped after around 2 hours. The CNN architecture and input data can be optimized to achieve similar accuracy with lower costs. Furthermore, only the required training data for each batch can be loaded during the training process rather than all at once. This can effectively remove any limitations on memory, and allow the use of more training data.

6 Conclusions

In this thesis, we explored the application of CNNs for the prediction of dispersion curves. We compared the performance of CNNs trained on shot gathers versus dispersion spectra, highlighting the strengths and limitations of each approach.

Our results indicate that CNNs trained on dispersion spectra generally provide more accurate and reliable predictions, particularly when dealing with noisy shot gathers. This is largely attributed to the reduced complexity of dispersion spectra compared to shot gathers and the less severe impact of random noise after transformation into the dispersion spectra, which simplifies the learning task for the neural network. Nevertheless, for predicting the first higher mode, we observed that using shot gathers as training data provides more stable predictions, also when noise is added.

We modeled synthetic shot gathers utilizing various wavelets, adapting them to our field data. Despite this, the use of different wavelets did not demonstrate a consistent improvement, and training on dispersion spectra still yielded superior results. To further enhance the accuracy of CNNs trained on shot gathers, we modified the kernel size of the first layer and applied transfer learning. Both adjustments resulted in notable improvements.

From a computational standpoint, generating synthetic training data and training the CNN are the two primarily resource-intensive tasks. We conclude that optimizing these processes through parallelization and adaptive grid sizing and time steps can significantly reduce computation times, making the approach more feasible for industrial applications.

We identified several directions for future research to build upon our findings:

- Optimize the model generation process to decrease computational cost.
- Apply data augmentation to the training data using noise with similar characteristics to that in the field data, improving the similarity of synthetic and field data and thus the training quality.
- Utilize different transformation techniques for the training on dispersion spectra, namely FDBF, to cover most site conditions to achieve high-quality dispersion spectra.
- Utilize unsupervised ML techniques, such as DBSCAN, which more closely resemble the conventional approach of dispersion curve picking and extraction (see Rovetta et al. (2021) and Wang et al. (2021)).

Addressing these areas, future research can further improve the accuracy, efficiency, and applicability of CNNs for dispersion curve prediction, ultimately aiding the field of geophysical and seismic data analysis.

References

Aki, K., & Richards, P. G. (1980). *Quantitative seismology: Theory and methods*. W. H. Freeman.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, 1–6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

Araya-Polo, M., Jennings, J., Adler, A., & Dahlke, T. (2018). Deep-learning tomography. *The Leading Edge*, *37*(1), 58–66. https://doi.org/10.1190/tle37010058.1

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Chamorro, D., Zhao, J., Birnie, C., Staring, M., Fliedner, M., & Ravasi, M. (2024). Deep learning-based extraction of surface wave dispersion curves from seismic shot gathers. *Near Surface Geophysics*, nsg.12298. https://doi.org/10.1002/nsg.12298

Choi, R. Y., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., & Campbell, J. P. (2020). Introduction to machine learning, neural networks, and deep learning. *Translational Vision Science & Technology*, *9*(2), 14. https://doi.org/10.1167/tvst.9.2.14

Cui, T., & Margrave, G. (2014). Seismic wavelet estimation. *CREWES Research Report*, *26*. Retrieved July 29, 2024, from https://www.crewes.org/Documents/ResearchReports/2014/CRR201418.pdf

Dai, T., Xia, J., Ning, L., Xi, C., Liu, Y., & Xing, H. (2021). Deep learning for extracting dispersion curves. *Surveys in Geophysics*, *42*(1), 69–95. https://doi.org/10.1007/s10712-020-09615-3

Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022, June 28). Activation functions in deep learning: A comprehensive survey and benchmark. Retrieved June 26, 2024, from http://arxiv.org/abs/2109.14545

Dunkin, J. W. (1965). Computation of modal solutions in layered, elastic media at high frequencies. *Bulletin of the Seismological Society of America*, *55*(2), 335–358. https://doi.org/10.1785/BSSA0550020335

Edgar, J. A., & Van Der Baan, M. (2011). How reliable is statistical wavelet estimation? *GEOPHYSICS*, *76*(4), V59–V68. https://doi.org/10.1190/1.3587220

Foti, S. (2015). *Surface wave methods for near-surface site characterization*. CRC Press, Taylor & Francis Group.

Gardner, G. H. F., Gardner, L. W., & Gregory, A. R. (1974). FORMATION VELOCITY AND DENSITY—THE DIAGNOSTIC BASICS FOR STRATIGRAPHIC TRAPS. *GEOPHYSICS*, *39*(6), 770–780. https://doi.org/10.1190/1.1440465

Haskell, N. A. (1953). The dispersion of surface waves on multilayered media*. *Bulletin of the Seismological Society of America*, *43*(1), 17–34. https://doi.org/10.1785/BSSA0430010017

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition [Version Number: 1]. https://doi.org/10.48550/ARXIV.1512.03385

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012, July 3). Improving neural networks by preventing co-adaptation of feature detectors. Retrieved July 7, 2024, from http://arxiv.org/abs/1207.0580

Hussien, M. N., & Karray, M. (2016). Shear wave velocity as a geotechnical parameter: An overview. *Canadian Geotechnical Journal*, *53*(2), 252–272. https://doi.org/10.1139/cgj-2014-0524

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift [Version Number: 3]. https://doi.org/10.48550/ARXIV.1502.03167

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. E. (2023). *An introduction to statistical learning: With applications in python*. Springer.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2017, February 9). On large-batch training for deep learning: Generalization gap and sharp minima. Retrieved July 20, 2024, from http://arxiv.org/abs/1609.04836

Kingma, D. P., & Ba, J. (2017, January 29). Adam: A method for stochastic optimization. Retrieved June 29, 2024, from http://arxiv.org/abs/1412.6980

Kumar Thota, S., Duc Cao, T., & Vahedifard, F. (2021). Poisson's ratio characteristic curve of unsaturated soils. *Journal of Geotechnical and Geoenvironmental Engineering*, *147*(1), 04020149. https://doi.org/10.1061/(ASCE)GT.1943-5606.0002424

Lay, T., & Wallace, T. C. (1995). *Modern global seismology*. Academic Press.

Li, H., Yang, W., Zhang, X., Wei, X., & Xu, X. (2022). A ResNet-based method for complex channel interpretation in seismic volumes. *IEEE Geoscience and Remote Sensing Letters*, *19*, 1–5. https://doi.org/10.1109/LGRS.2022.3223422

Lii, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827

Lisman, J. (2015). The challenge of understanding the brain: Where we stand in 2015. *Neuron*, *86*(4), 864–882. https://doi.org/10.1016/j.neuron.2015.03.032

Luu, K. (2021, December 12). *Disba: Numba-accelerated computation of surface wave dispersion* (Version v0.6.1). Zenodo. https://doi.org/10.5281/ZENODO.5775195

Macukow, B. (2016). Neural networks – state of art, brief history, basic models and architecture [Series Title: Lecture Notes in Computer Science]. In K. Saeed & W. Homenda (Eds.), *Computer information systems and industrial management* (pp. 3–14, Vol. 9842). Springer International Publishing. https://doi.org/10.1007/978-3-319-45378-1_1

Margrave, G. F., & Lamoureux, M. P. (2018, December 31). *Numerical methods of exploration seismology: With algorithms in MATLAB®* (1st ed.). Cambridge University Press. https://doi.org/10.1017/9781316756041

Nielsen, M. (2015). *Neural networks and deep learning*. Determination Press.

Nolet, G., & Panza, G. F. (1976). Array analysis of seismic surface waves: Limits and possibilities. *Pure and Applied Geophysics PAGEOPH*, *114*(5), 775–790. https://doi.org/10.1007/BF00875787

O'Shea, K., & Nash, R. (2015, December 2). An introduction to convolutional neural networks. Retrieved June 30, 2024, from http://arxiv.org/abs/1511.08458

Park, C. B., Miller, R. D., & Xia, J. (1999). Multichannel analysis of surface waves. *GEOPHYSICS*, *64*(3), 800–808. https://doi.org/10.1190/1.1444590

Park, C. B., Miller, R. D., & Xia, J. (1998). Imaging dispersion curves of surface waves on multi-channel record. *SEG Technical Program Expanded Abstracts 1998*, 1377–1380. https://doi.org/10.1190/1.1820161

Prieto, A., Prieto, B., Ortigosa, E. M., Ros, E., Pelayo, F., Ortega, J., & Rojas, I. (2016). Neural networks: An overview of early research, current frameworks and new challenges. *Neurocomputing*, *214*, 242–268. https://doi.org/10.1016/j.neucom.2016.06.014

Rahimi, S., Wood, C. M., & Teague, D. P. (2021). Performance of different transformation techniques for MASW data processing considering various site conditions, near-field effects, and modal separation. *Surveys in Geophysics*, *42*(5), 1197–1225. https://doi.org/10.1007/s10712-021-09657-1

Rasamoelina, A. D., Adjailia, F., & Sincak, P. (2020). A review of activation function for artificial neural network. *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 281–286. https://doi.org/10.1109/SAMI48414.2020.9108717

Ren, L., Gao, F., Williamson, P., & McMechan, G. A. (2021). On application issues of automatic dispersion curves picking by machine learning. *First International Meeting for Applied Geoscience & Energy Expanded Abstracts*, 1836–1840. https://doi.org/10.1190/segam2021-3594524.1

Ren, L., Gao, F., Wu, Y., Williamson, P., Wang, W., & McMechan, G. A. (2020). Automatic picking of multi-mode dispersion curves using CNN-based machine learning.

SEG Technical Program Expanded Abstracts 2020, 1551–1555. https://doi.org/10.1190/segam2020-3427827.1

Rovetta, D., Kontakis, A., & Colombo, D. (2021). Application of a density-based spatial clustering algorithm for fully automatic picking of surface-wave dispersion curves. *The Leading Edge*, *40*(9), 678–685. https://doi.org/10.1190/tle40090678.1

Shearer, P. M. (2019). *Introduction to seismology* (Third edition). Cambridge University Press.

Socco, V., & Strobbia, C. (2004). Surface-wave method for near-surface characterization: A tutorial. *Near Surface Geophysics*, *2*(4), 165–185. https://doi.org/10.3997/1873-0604.2004015

Socco, V., Foti, S., & Boiero, D. (2010). Surface-wave analysis for building near-surface velocity models — established approaches and new perspectives. *GEOPHYSICS*, *75*(5), 75A83–75A102. https://doi.org/10.1190/1.3479491

Thomson, W. T. (1950). Transmission of elastic waves through a stratified solid medium. *Journal of Applied Physics*, *21*(2), 89–93. https://doi.org/10.1063/1.1699629

Thorbecke, J. W., & Draganov, D. (2011). Finite-difference modeling experiments for seismic interferometry. *GEOPHYSICS*, *76*(6), H1–H18. https://doi.org/10.1190/geo2010-0039.1

Thorbecke, Jbrackenhoff, & Koffieklopper. (2020, November 13). *JanThorbecke/OpenSource: Release for MME manuscript r2* (Version 2.1.2). Zenodo. https://doi.org/10.5281/ZENODO.3363570

Wang, Z., Sun, C., & Wu, D. (2021). Automatic picking of multi-mode surface-wave dispersion curves based on machine learning clustering methods. *Computers & Geosciences*, *153*, 104809. https://doi.org/10.1016/j.cageo.2021.104809

Wu, B., Meng, D., Wang, L., Liu, N., & Wang, Y. (2020). Seismic impedance inversion using fully convolutional residual network and transfer learning. *IEEE Geoscience and Remote Sensing Letters*, *17*(12), 2140–2144. https://doi.org/10.1109/LGRS.2019.2963106

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020, June 23). A comprehensive survey on transfer learning. Retrieved July 28, 2024, from http://arxiv.org/abs/1911.02685

Zywicki, D. J., & Rix, G. J. (2005). Mitigation of near-field effects for seismic surface wave velocity estimation with cylindrical beamformers. *Journal of Geotechnical and Geoenvironmental Engineering*, *131*(8), 970–977. https://doi.org/10.1061/(ASCE)1090-0241(2005)131:8(970)

A Appendix

A.1 System specifications

A.1.1 Hardware

All computations presented in this thesis were carried out on the same device with the following specifications:

- CPU: 12th Gen Intel Core i7-12850HX, 2100 Mhz, 16 Cores, 24 Logical Processors
- GPU: NVIDIA RTX A3000 12GB Laptop GPU
- RAM: 32 GB, 4800 MHz
- Storage Drive: KIOXIA XG8 1 TB SSD NVMe

A.1.2 Software

While the main operating system of the notebook is Windows 10, *fdelmodc* is a Unix-based software, wherefore the Windows Subsystem for Linux 2 (WSL2) (version 2.2.4.0.) is used to create a Linux environment (Ubuntu 20.04.6 LTS). As WSL gets assigned only half of the available memory, we manually increased it to 24 GB. Coding was done with the WSL extension for Visual Studio Code using Python. The different tasks carried out in this thesis are organized into different Jupyer Notebooks, adopting the structure from Chamorro et al. (2024).

We utilize Miniconda as an environment management system. The environment created to run all the notebooks can be found HERE. During the beginning of the thesis, a lot of time was spent on debugging many dependencies of the different packages. Therefore, setting up a new environment with the provided library versions is encouraged.

A.2 Data Accessibility

To reproduce the results presented in this thesis, all necessary codes are made available on GitHub. The training data and testing models can be generated with this set of codes, if the necessary dependencies are installed (see Appendix A.1.2). Storing 10000 shot gathers for the training, for example for the Wallingford field data, requires 5.6 GB, but is always dependent on the chosen number of receivers, recording length, and sampling interval. We also want to express our gratitude to Fugro for making the field data publicly available.

Be aware, that due to the stochastic nature of the random model generation and the CNN training, obtained results might differ from the results presented in this thesis.

A.3 Training Curves for the ResNet-18 models used in this thesis

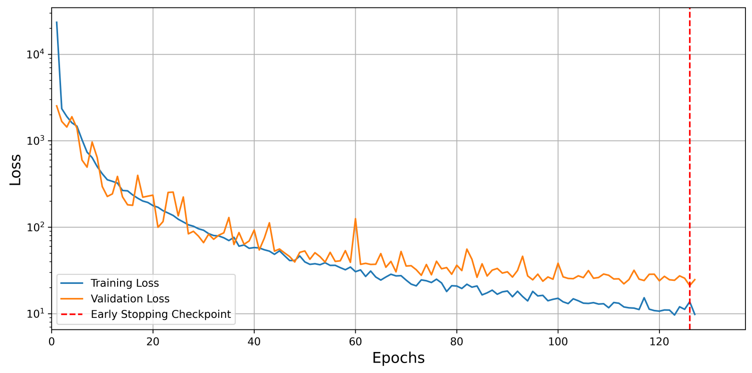

Figure A.1: Training and validation loss over epochs for the CNN trained on shot gathers without inversion created for the benchmark model in Figure 4.4. The final validation loss is 23.4 after 126 epochs.

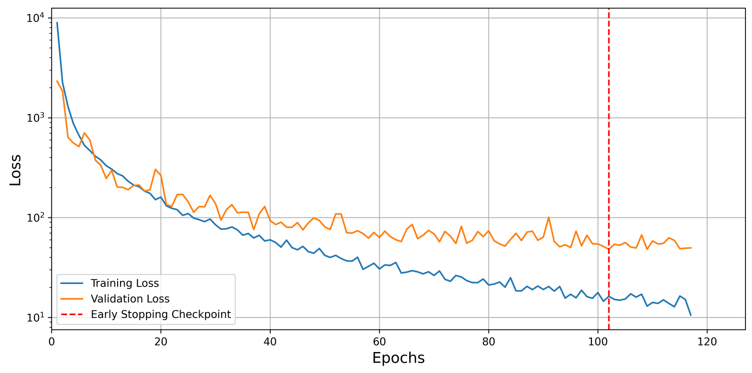

Figure A.2: Training and validation loss over epochs for the CNN trained on shot gathers with inversion created for the benchmark model in Figure 4.4. The final validation loss is 48.1 after 102 epochs.

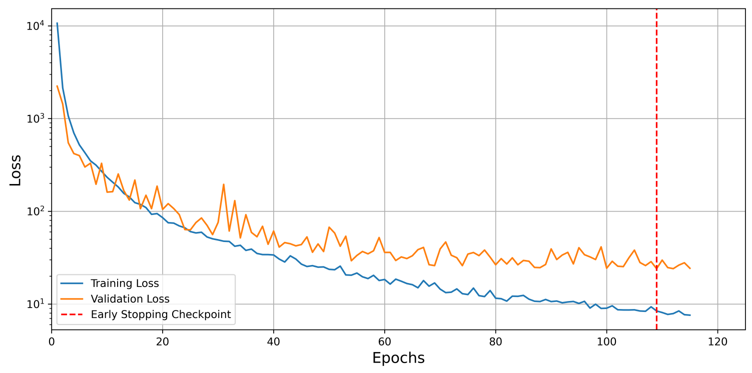

Figure A.3: Training and validation loss over epochs for the CNN trained on the combined shot gathers with and without inversion created for the benchmark model in Figure 4.4. The final validation loss is 25.2 after 109 epochs.

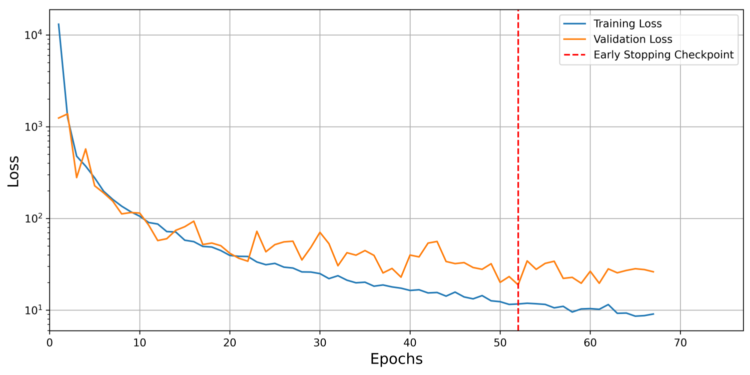

Figure A.4: Training and validation loss over epochs for the CNN trained on dispersion spectra without inversion created for the benchmark model in Figure 4.4. The final validation loss is 19.8 after 52 epochs.

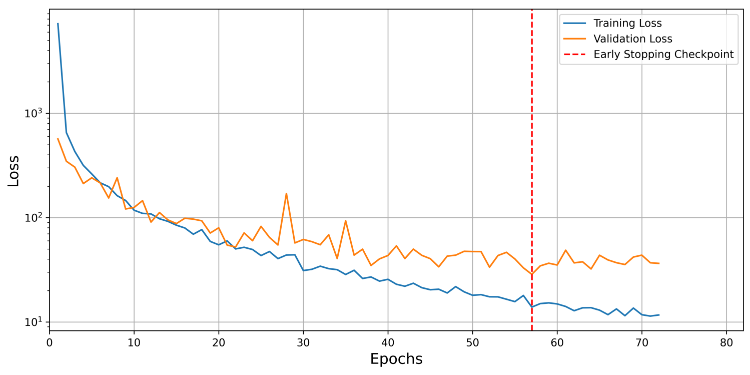

Figure A.5: Training and validation loss over epochs for the CNN trained on dispersion spectra with inversion created for the benchmark model in Figure 4.4. The final validation loss is 28.4 after 57 epochs.

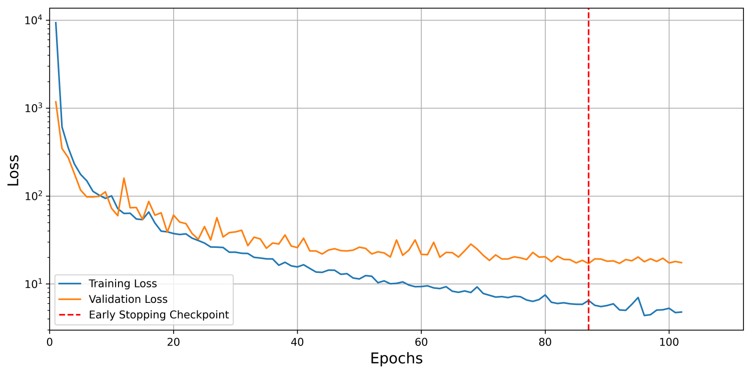

Figure A.6: Training and validation loss over epochs for the CNN trained on the combined dispersion spectra with and without inversion created for the benchmark model in Figure 4.4. The final validation loss is 17.8 after 87 epochs.

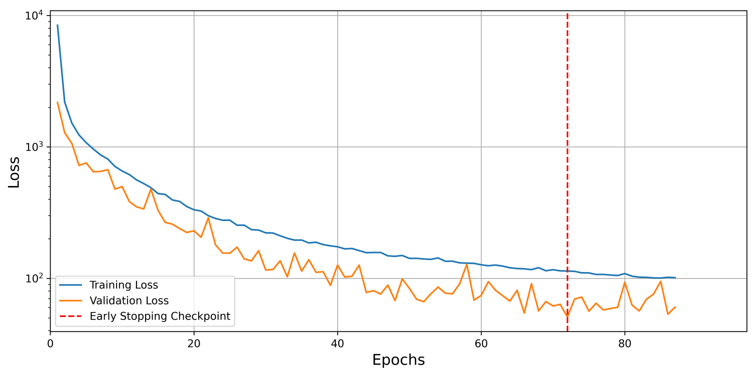

Figure A.7: Training and validation loss over epochs for the CNN with a dropout rate of 0.15 trained on the combined shot gathers (with and without inversion) created for the benchmark model in Figure 4.4. The final validation loss is 50.8 after 72 epochs.

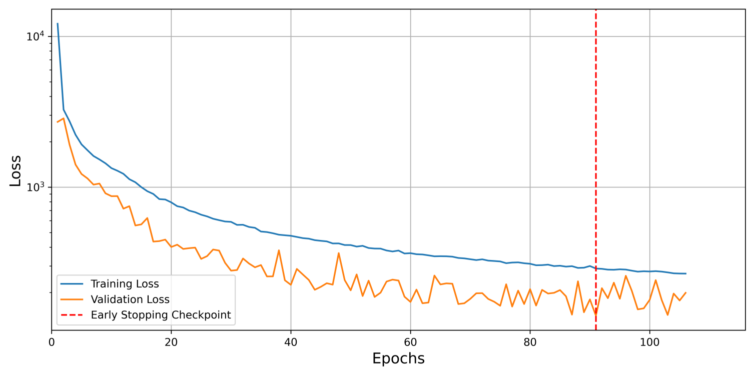

Figure A.8: Training and validation loss over epochs for the CNN with a dropout rate of 0.3 trained on the combined shot gathers (with and without inversion) created for the benchmark model in Figure 4.4. The final validation loss is 127.3 after 91 epochs.

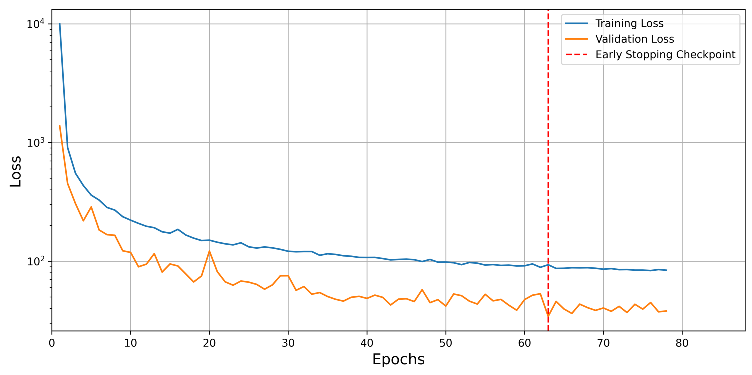

Figure A.9: Training and validation loss over epochs for the CNN with a dropout rate of 0.15 trained on the combined dispersion spectra (with and without inversion) created for the benchmark model in Figure 4.4. The final validation loss is 33.1 after 63 epochs.

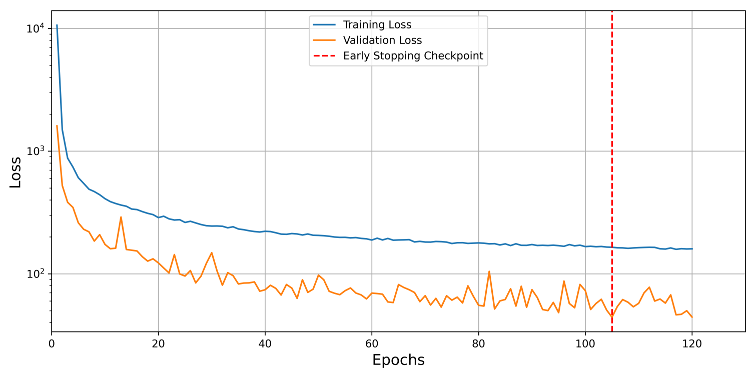

Figure A.10: Training and validation loss over epochs for the CNN with a dropout rate of 0.3 trained on the combined dispersion spectra (with and without inversion) created for the benchmark model in Figure 4.4. The final validation loss is 44.2 after 105 epochs.

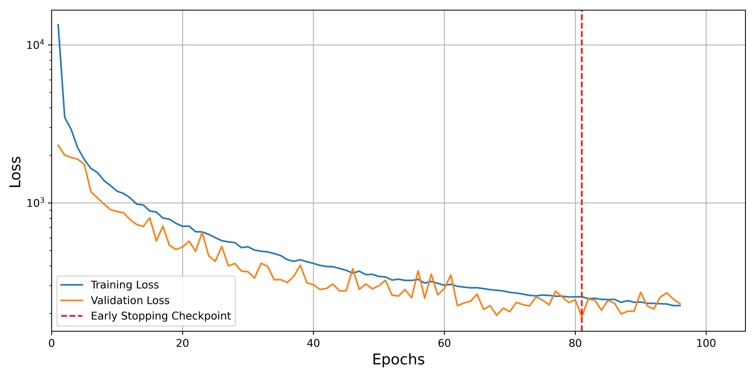

Figure A.11: Training and validation loss over epochs for the CNN with a dropout rate of 0.15 trained on the shot gathers created for the higher-mode model in Figure 4.10. The final validation loss is 192.0 after 81 epochs.

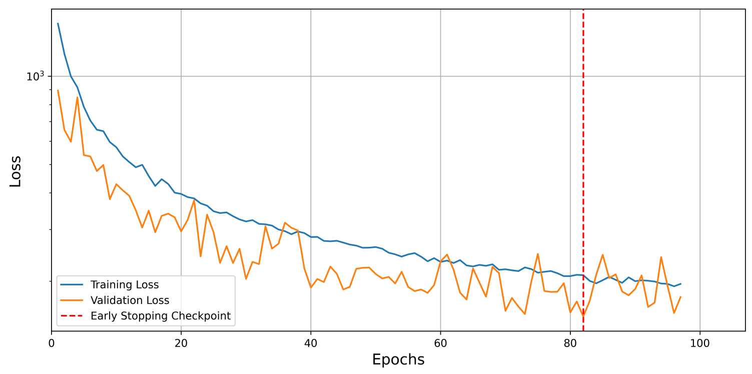

Figure A.12: Training and validation loss over epochs for the CNN with a dropout rate of 0.15 trained on the dispersion spectra created for the higher-mode model in Figure 4.10. The final validation loss is 125.8 after 82 epochs.

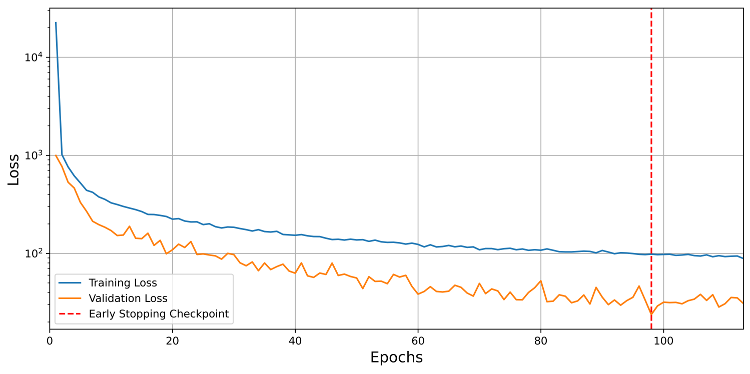

Figure A.13: Training and validation loss over epochs for the CNN trained on shot gathers for the Wallingford field data using the statistical wavelet. The final validation loss is 23.0 after 98 epochs.

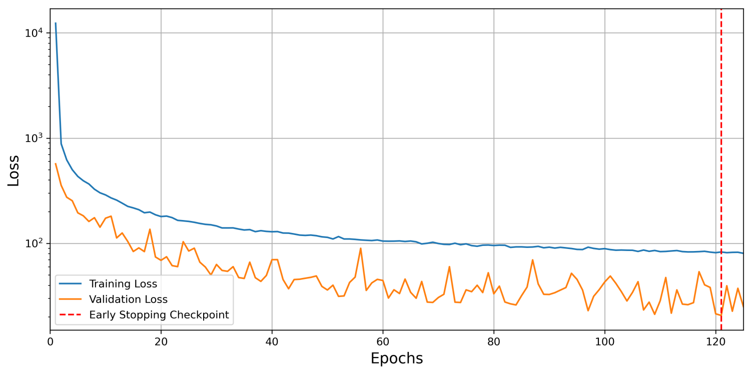

Figure A.14: Training and validation loss over epochs for the CNN trained on shot gathers for the Wallingford field data using the minimum-phase wavelet. The final validation loss is 20.5 after 121 epochs.

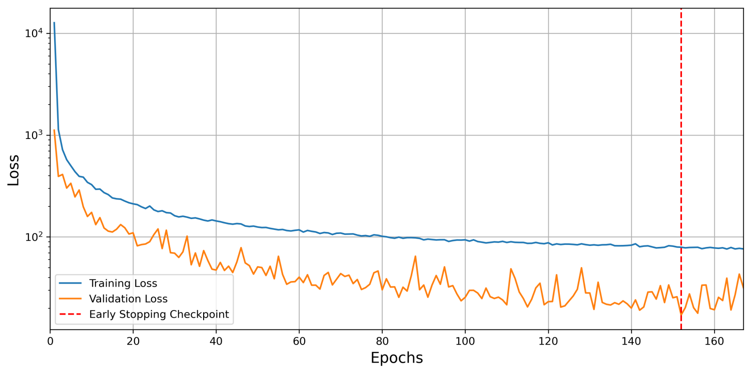

Figure A.15: Training and validation loss over epochs for the CNN trained on dispersion spectra for the Wallingford field data. The final validation loss is 17.2 after 152 epochs.

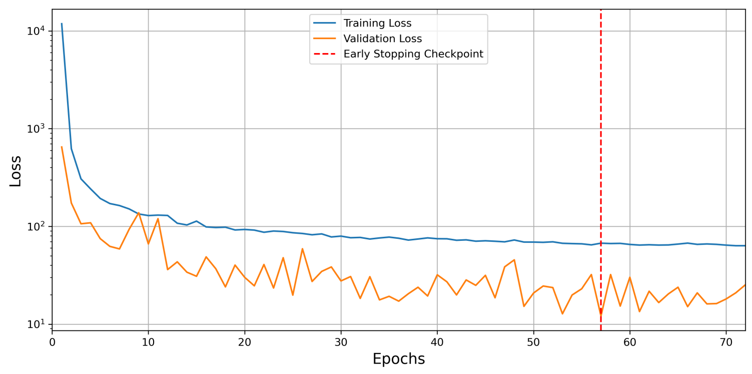

Figure A.16: Training and validation loss over epochs for the CNN trained on dispersion spectra for the Wallingford field data. The final validation loss is 13.4 after 57 epochs.

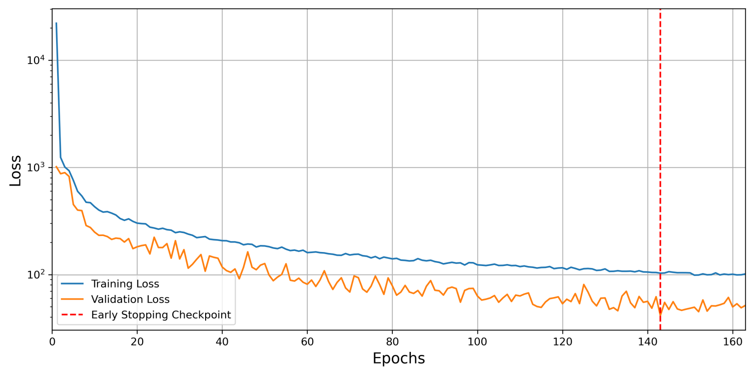

Figure A.17: Training and validation loss over epochs for the modified CNN, with a 3x7 kernel in the first convolutional layer, trained on shot gathers for the Wallingford field data using the minimum-phase wavelet. The final validation loss is 40.8 after 143 epochs.

A.4 Research Module Report

See the following pages.

Research Module Report

for the master thesis on

Machine Learning for Automatic Picking of Dispersion Curves

Kilian Glatz

Master's Thesis Principal Supervisor

Dr. Deyan Draganov

Associate Professor at TU Delft

Master's Thesis Local Supervisors

Dr. Florencia Balestrini and Dr. Myrna Staring

Fugro Innovation & Technology B.V.

Module representative

Prof. Dr. Florian Wellmann

Professor at RWTH Aachen

RWTH Aachen

02.02.2024

1 Abstract

We provide a comprehensive review of Multichannel Analysis of Surface Waves (MASW) principles, focusing on theoretical foundations and workflow. We emphasize the critical role of spectral resolution, influenced by acquisition geometry and geological conditions, and discuss challenges such as mode mixing.

We review recent publications on supervised and unsupervised Machine Learning (ML) dispersion-curve picking algorithms, highlighting their advantages and disadvantages in an engineering context. While unsupervised clustering algorithms show success, concerns for engineering purposes arise because this approach relies on clear mode separation. Supervised methods, if provided with tailored training data, offer promising results. Particularly, training a convolutional neural network (CNN) model directly on shot gathers eliminates the need for transformations in other domains and overcomes associated limitations. We propose implementing this approach for multi-mode dispersion-curve picking, comparing its performance with unsupervised methods using dispersion spectra as training data.

2 Introduction

This report is part of the *Research Module in Applied Geophysics* undertaken at RWTH Aachen during the winter semester of 2023/2024. The goal of this report is to establish the fundamental knowledge needed for the subsequent master's thesis which will be written at *Fugro Innovation & Technology B.V.*, Nootdorp, Netherlands, in cooperation with the TU Delft.

In the first part, as preparation and for completeness of the master thesis, a literature review is performed. The most important properties of surface waves and the necessary steps in Multichannel Acquisition of Surface Waves (MASW) are summarized. This part aims to offer non-professional readers the essential background knowledge required to comprehend the content and context of the master's thesis.

The second part involves an analysis of existing literature on the automation of dispersion-curve picking using Machine Learning (ML). The advantages and disadvantages of supervised and unsupervised ML for dispersion curve picking are highlighted. The most promising approach and the associated limitations will be identified and subsequently implemented in the master's thesis. The identified limitations will then be addressed and refined using the latest advancements in ML.

3 Multichannel Acquisition of Surface Waves

Surface-wave methods have become increasingly popular in the last two decades. While traditionally considered noise in reflection seismic, recent developments made the use of surface waves for near-surface applications more attractive. For a more extensive literature review and best-practice guidelines of MASW in an engineering context, the reader is referred to Socco and Strobbia (2004), Socco et al. (2010), and Foti et al. (2018). The following sections summarize the most important properties of surface waves and steps for MASW (conceptually shown in Figure 1).

As most of the published scientific literature about near-surface velocity-model retrieval treat Rayleigh waves (Socco et al., 2010), the following sections will focus on this type of surface wave, but most concepts can easily be transferred to other types of surface waves. Special focus will be put on acquisition parameters and transformations influencing the dispersion curve picking to highlight problems that must be considered using ML for automated dispersion curve picking.

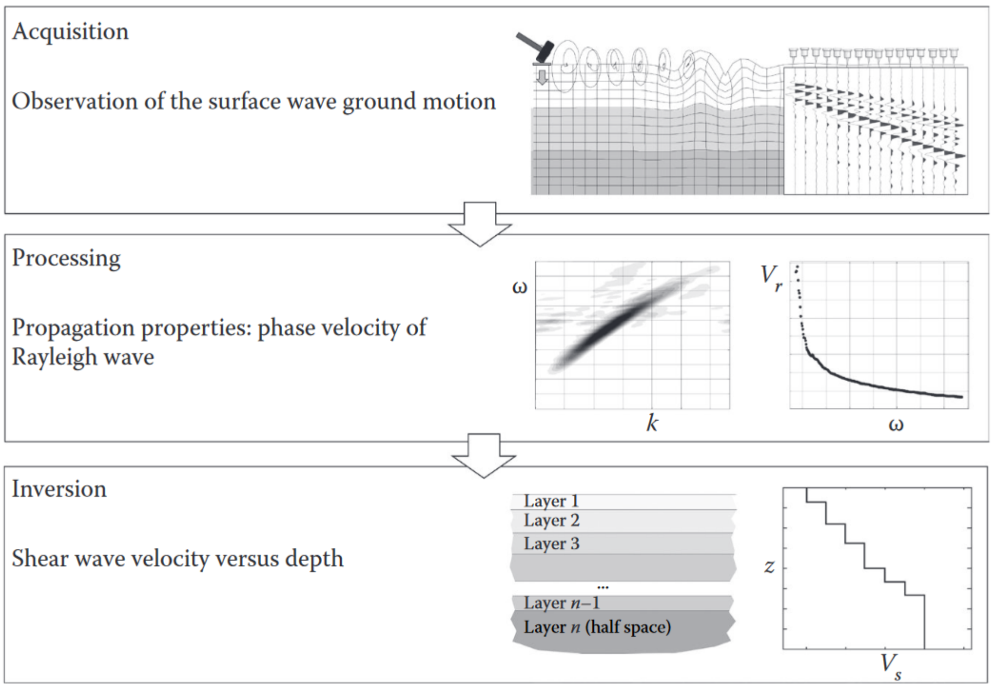
Acquisition
Observation of the surface wave ground motion
Processing
Propagation properties: phase velocity of Rayleigh wave
ω
k
V_r
ω
Inversion
Shear wave velocity versus depth
Layer 1
Layer 2
Layer 3
...
Layer n–1
Layer n (half space)
z
V_s

Figure 1: Conceptual flow of MASW from Foti et al. (2018). **Upper:** Acquisition is carried out in the field and the data is recorded as seismic traces using geophones. **Middle:** The collected data is transformed in a domain where the phase velocity as a function of frequency can be identified. **Lower:** The retrieved dispersion curve is inverted to get the S-wave velocity profile of the subsurface. The three steps are described in more detail in chapters 3.2, 3.3, and 3.4, respectively.

3.1 Properties of Surface Waves

Rayleigh waves (and surface waves in general) differ from body waves in many aspects, making them particularly useful in near-surface applications. In contrast to body waves, where a solution to the seismic wave equation also exists in whole spaces, solutions for surface waves only exist in a medium that has a free surface (Shearer, 2019).

Rayleigh waves propagate along the surface with a combination of compressional and vertical polarized particle motion (Foti, 2015; Shearer, 2019). For a homogenous medium, the particle motion of the fundamental mode is retrograde at the surface (opposite to the direction of Rayleigh wave propagation), changing first to purely vertical and then to prograde motion with increasing depth. In the case of a vertically heterogeneous medium, the particle motion strongly depends on the actual velocity distribution and the mode that is considered. It can also be observed that the total amplitude of particle motion decays with depth and it is commonly assumed to vanish within roughly one wavelength (Socco and Strobbia, 2004; Foti, 2015).

The fact that different wavelengths show different penetration depths forms the basis of surface-wave analysis. In a vertically heterogeneous medium, the phase velocity at which each frequency propagates depends on the velocity distribution of the subsurface. This leads to the existence of dispersion curves, which are curves of phase-velocity as a function of wavelength or frequency. Figure 2 from Foti et al. (2018) shows a qualitative sketch of this property for a two-layer half-space. While short wavelengths propagate predominantly with the velocity of the uppermost layer, larger wavelengths increasingly approach the velocity of the lower half-space.

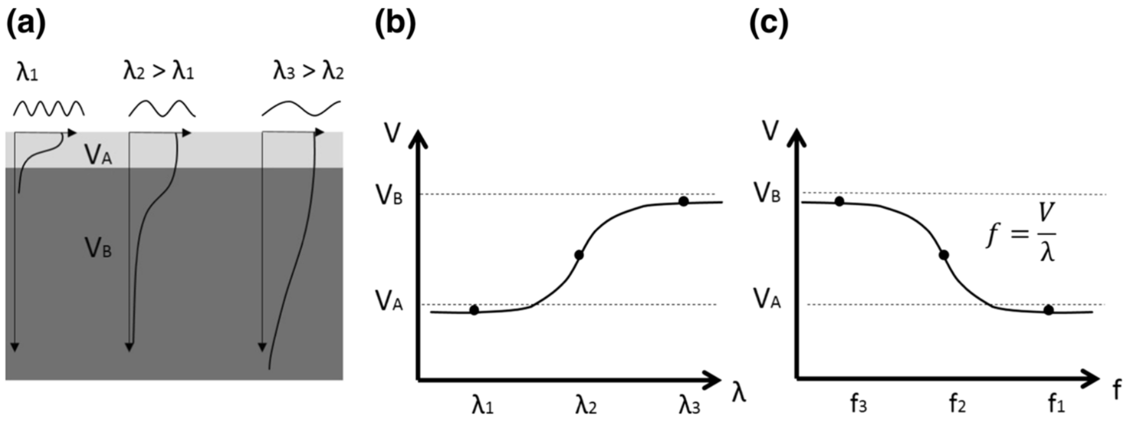

Figure 2: Qualitative sketch of geometric dispersion after Foti et al. (2018). **(a)** shows the amplitude decay (penetration depths) of three different wavelength in a two-layer half-space for the fundamental mode, **(b)** and **(c)** show the corresponding dispersion curves in the wavelength- and frequency-phase velocity domain, respectively.

Another important property of surface waves is the energy distribution during propagation. In a homogenous half-space the energy is distributed over the curved surface of a cylinder, resulting in the energy density decaying proportionally to one over the distance from the source location. Body waves in contrast have their energy distributed over the curved surface of a half-sphere, and thus an energy-density decay proportional to the inverse of the square of the distance from the source point. This leads to the very important conclusion, that surface waves start to dominate the wavefield quickly and provide higher S/N-ratio in records that could also be used for P-wave refraction seismics (Socco and Strobbia, 2004).

In the previous figures and consideration only the fundamental mode of Rayleigh waves were taken into account. Assuming a layered subsurface, the Rayleigh secular equation is

$$F(k, f) = 0, \quad F_R\left[\lambda(z), G(z), \rho(z), k_j, f\right] = 0 \tag{1}$$

where λ and G are the Lamé parameters, ρ is the mass density, k is the wavenumber and f is the frequency (Socco and Strobbia, 2004; Foti, 2015). For a given frequency f, there are multiple values for the wavenumber k_j where a solution exists for equation 1, resulting in the existence of modal curves. The existence of modal curves can cause complications during the processing and inversion of surface wave data making it important to take them into account when planning the survey design. The complications in the analysis of surface waves due to modal curves will be discussed in the following sections.

3.2 Acquisition

The success of MASW relies heavily on the careful selection of acquisition parameters, including the choice of sources, receivers, and their respective configurations. The data acquired during a MASW survey should provide a high signal-to-noise ratio (S/N), allow for modal separation, and allow for uncertainty quantification (Socco and Strobbia, 2004).

The choice of a source depends on the specific goals of the survey and the desired penetration depth for subsurface investigation. Vertical vibrating sources allow for accurate control of the frequency band and high S/N but come at the cost of being expensive and not as easy to operate (Foti et al., 2018).

Impact sources are very popular because they are easy to operate and still cover a considerable frequency band, therefore being very cost-efficient. However, impact sources like sledgehammers or weight-drop systems show limited energy in the low-frequency band ($f < 8$ Hz) limiting the investigation depth (Foti et al., 2018). Passive approaches using ambient noise can overcome this limitation, as described at the end of this chapter.

The source position regarding the closest and furthest receiver has to be considered because of near- and far-field effects (Park et al., 1999). If the offset is too short, the plane-wave assumption used in most dispersion analysis approaches is not valid leading to an underestimation of phase velocities at low frequencies (Foti et al., 2018). At too large offsets, the attenuation of high-frequency components of the surface waves possibly reduces the S/N to an undesirable level.

Receivers and the receiver array configuration play a crucial role in capturing the surface-wave signals generated by the sources. For Rayleigh wave surveys, vertical geophones with a natural frequency of 4.5 Hz are typically used to adequately sample the expected frequency band (Foti et al., 2018).

In MASW, multiple receivers are used and, in practice, the number of receivers is usually limited by the available equipment on hand. This limitation necessitates careful arraylength and receiver-spacing considerations. Array length refers to the distance between the first and last receiver in a linear array. The array length determines the wavenumber resolution and therefore the ability for mode separation (Socco and Strobbia, 2004). The comparison of a short and a long array is shown in Figure 3 and shows that short array lengths decrease the wavenumber resolution reducing the modal discriminability. On the other hand, if lateral velocity variations are present, shorter arrays improve the validity of the lateral homogeneity assumption typically used during the inversion procedure (Socco and Strobbia, 2004; Foti, 2015).

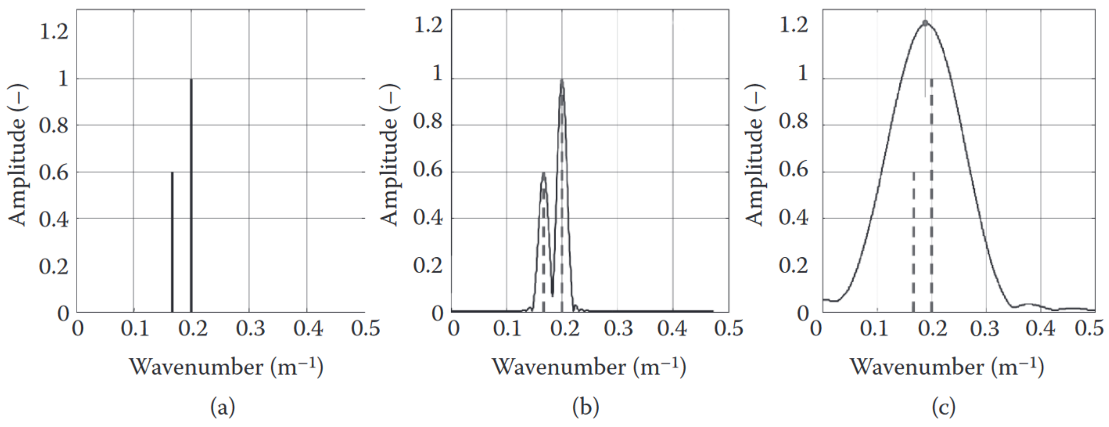

Figure 3: Example of influence of array length on spectral resolution. **(a)** Ideal wavenumber spectrum for theoretical infinite array length. **(b)** Using a long array leakage can be observed, but the two events are still resolved. **(c)** The high spectral leakage for a short array makes it impossible to distinguish the two events. Figure from Foti (2015).

The receiver spacing between adjacent receivers has to be chosen to sample the short wavelength according to the Nyquist-Shannon theorem. If the receiver spacing is too large aliasing occurs which may reduce the quality of the dispersion curve preventing the correct identification of different modes (Foti et al., 2018). Using a limited number of receivers, a compromise between array length and receiver spacing (wavenumber resolution and spatial aliasing) has to be made.

In contrast to active methods, where controlled sources generate surface waves, passive approaches make use of ambient noise as sources. Passive methods are particularly advantageous in situations where controlled sources are impractical or when information about the deep subsurface is of interest since ambient noise typically contains lower frequencies. Combining passive and active approaches can provide improved results over a wider frequency range, as shown by Park et al. (2005) and Hu et al. (2022). Acquisition layouts in passive approaches require special geometries to account for the multi-directionality from which signals can arrive and also require special processing approaches.

3.3 Processing

The foundation of traditional surface-wave analysis is the extraction of dispersion curves, which provide information about the material properties of the subsurface. Dispersion curves display the relationship between phase velocity and frequency, and their accurate determination is essential for subsequent inversion and interpretation. The extraction process involves transforming the acquired wavefield data into a domain where the dispersion curve can be identified and analyzed, usually in the phase velocity-frequency or wavenumber-frequency domain (Socco and Strobbia, 2004).

Dispersion curves can be picked manually or automatically from the transformed wavefield data. Manual picking involves a professional visually identifying and selecting the peaks corresponding to the highest amplitudes in the dispersion curve. However, this approach is susceptible to subjectivity and can be time-consuming.

One criterion commonly used in automatic picking algorithms is picking the highest energy (Foti, 2015) in the transformed domain. However, problems might arise because the highest amplitude might not correspond to the fundamental mode. Additionally, the lack of spectral resolution might lead to an erroneous apparent dispersion curve (Foti et al., 2018). Consequently, visual inspection remains a crucial step to ensure the accuracy of the picked dispersion curve, particularly in cases where modes might overlap or are misidentified. Figure 4 shows examples of apparent dispersion curves using the maximum-energy approach for different subsurface models and how possible errors get introduced into the analysis.

The choice of transformation method also plays an important role for the quality of the dispersion analysis. The work by Rahimi et al. (2021) provides valuable insights into the performance of different transformation techniques.

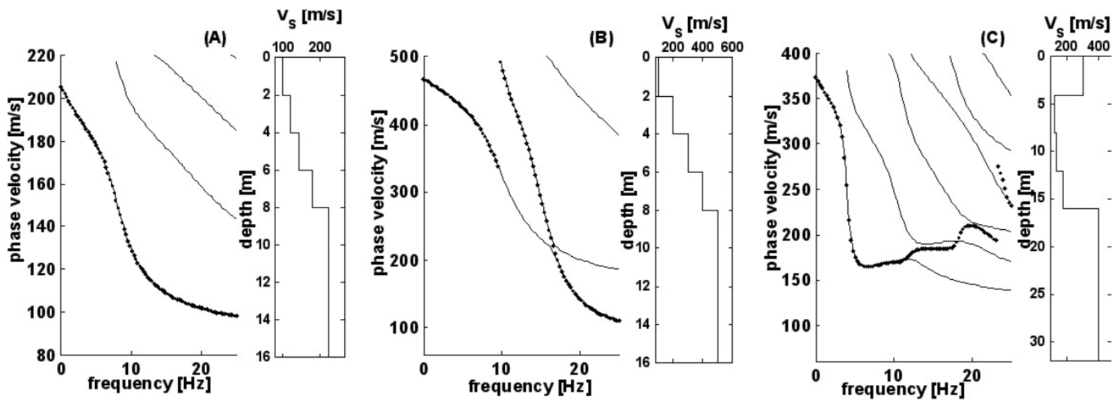

Figure 4: Examples of modal curves for different velocity models (next to dispersion curves). Black dots correspond to the maximum energy at each frequency. **(A)** Case with a slow increase of S-wave velocity with depth and the maximum energy at the fundamental mode. **(B)** In this case, the increase in velocity is larger and the maximum energy jumps to the first higher mode at around 10 Hz. Also note the close separation of the fundamental and first higher mode at around 17 Hz. In the case of velocity inversion, as depicted in **(C)**, the maximum energy continuously jumps to higher modes creating an continuous apparent dispersion curve. Figure from Socco and Strobbia (2004).

3.4 Inversion

Inversion is the final step in the analysis of surface waves, and to estimate the subsurface parameters that explain the measured data. In MASW, the data to be inverted is the phase velocity of rayleigh waves as a function of frequency, while the related parameters to be estimated are the velocity and/or attenuation profile (Socco and Strobbia, 2004). Surface-wave inversion is characterized by its non-linear and non-unique nature (Foti et al., 2018). This inherent ambiguity makes it crucial to adopt robust inversion strategies and, if available, incorporate additional information to constrain and refine the inversion results. Joint inversion approaches that combine surface-wave data with other geophysical methods or borehole data can enhance the reliability and accuracy of the inversion results (Foti, 2015).

Inversion strategies can be categorized into local and global search methods. Local search methods focus on iteratively refining the inversion model within the vicinity of an initial model, searching for the best-fitting parameters in a constrained region by minimizing an objective function. On the other hand, global search methods explore the entire model space to identify the globally optimal model. The choice between these strategies depends on the complexity of the subsurface and the available computational resources.

MASW data is typically inverted to derive a local 1D velocity model (Foti et al., 2018). In this case, the inversion process estimates layer thickness and S-wave velocity, as these parameters most significantly influence the dispersion curve (Foti et al., 2018). Mass densities and P-wave velocities are often assumed a priori, simplifying the inversion process and focusing on the key parameters that govern surface-wave behavior.

An alternative approach to surface-wave analysis is full-waveform inversion (FWI), which tries to find a model that is able to fit the entire waveform of the recorded data. This approach enhances the accuracy of subsurface velocity models and allows for the resolution of lateral changes in subsurface velocities, as shown by Pan et al. (2019). However, it is important to note that while FWI offers superior results, it comes with high computational costs, making it more resource-intensive compared to the previously described conventional inversion approache.

4 Machine Learning for Dispersion Curve Picking

ML is a rapidly evolving field that leverages computational models to enable systems to learn and make predictions or decisions without explicit programming. With the ever-increasing capabilities of gathering higher-quality data, ML provides a useful tool in handling these vast amounts of data. One of the fundamental categorizations in machine learning is the division between supervised and unsupervised learning algorithms. Each type serves distinct purposes, addressing different challenges in data analysis and pattern recognition. Recently, both types of ML approaches have been adopted for the analysis of surface waves. In the following sections, different approaches for the automation of dispersion curve picking are reviewed.

4.1 Supervised Machine Learning

Supervised ML relies on labeled training datasets, meaning that the dataset contains input data (features) and the correct output (label). These labeled datasets are used to train the model, enabling it to make predictions on new datasets.

One of the first applications of ML in the context of surface-wave dispersion curve picking was conducted by Alyousuf et al. (2018) in the context of near-surface corrections for reflection datasets. The features and labels in this study are represented by phase velocity-frequency plots of identical dimensions. In this context, the label consists of ones and zeros, with ones denoting the manually picked dispersion curve. Subsequently, the trained model was used to pick dispersion curves on additional gathers which then were inverted to generate a 1D S-velocity model. This model was subsequently used to structurally constrain a P-wave velocity model derived from (first arrival) seismic refraction tomography. The efficacy of this workflow was successfully validated using a field dataset from a Wadi in Saudi Arabia.

It is crucial to note that the application of this methodology did not encompass higher modes, and the dispersion curve labels employed in the training data lacked ground truth. This poses a potential challenge, as the dispersion images displayed in the work of Alyousuf et al. (2018) show overlapping modes and elevated noise levels. Consequently, the manual picks in the training data are prone to errors, which can subsequently impact the accuracy of the solution. Further consideration and refinement of the training data, especially concerning the absence of ground truth for dispersion curves and the exclusion of higher modes, may contribute to enhancing the robustness of the proposed approach.

Ren et al. (2021) used a similar approach tested on an Ocean Bottom Nodes (OBN) data set. They used a CNN to directly output the dispersion curves values from the input f-k-domain data. Important improvements compared to Alyousuf et al. (2018) are the incorporation of the first higher mode and the use of transfer learning. In transfer learning, the network model is first trained on synthetic data and then on manually picked dispersion curves. This allows for the incorporation of a broad range of possible synthetic models with ground truth and greatly reduces the number of needed labeled field samples. Furthermore, they pointed out the non-determinism of ML and that further analysis was needed in the error quantification due to ground truth picking or noise level in the data.

Chamorro et al. (2023) showed an approach using CNN which directly maps the shot gathers to dispersion curves. This has the advantage that the wavefield transformation to a different domain is avoided, which can cause errors as discussed in Rahimi et al. (2021). The training data is numerically modeled using an elastic finite-difference algorithm incorporating prior knowledge of the test site. Because the source characteristics in the synthetic and field data differ, this approach requires convolving the respective wavelets with the other data set to make the data compatible. The application on a field data set acquired for engineering purposes (short array length of 23 m) results in a subsurface model consistent with prior knowledge of the area.

It is crucial to emphasize that the successfulness of supervised ML in all the mentioned publications heavily depends upon the quality of the training data. Synthetic training data provide the advantage of having a ground truth dispersion curve but need to be tailored to the site with respect to the expected lithology and data quality of the field data.

4.2 Unsupervised Machine Learning

Unsupervised learning deals with unlabeled datasets, where the algorithm aims to discover inherent patterns or structures within the data without explicit guidance. Because no labeled data is needed, unsupervised algorithms do not require manual intervention and thus run fully automatic, making them highly attractive for production (Rovetta et al., 2021). Existing unsupervised ML dispersion curve-picking algorithms are mostly based on clustering algorithms (Khosro Anjom et al., 2024).

Rovetta et al. (2021) introduced an approach to dispersion curve picking using the Density-Based Spatial Clustering Algorithm (DBSCAN). DBSCAN clusters points into groups based on a defined radius and a minimum amount of points connected via the radius to be considered a cluster. Additional processing steps are necessary before applying DBSACN, one being the selection of the maximum energy for each frequency. However, potential challenges arise when using this approach, as discussed in Section 3.2. The lack of spectral resolution may lead to a smooth transition of the highest energy to higher modes, introducing the possibility of erroneous mode identification (see Figure 4 c).

To assess the efficiency of DBSCAN, Rovetta et al. (2021) conducted analyses on both synthetic and field data sets. The synthetic data utilized the SEG Advanced Modeling Corporation Arid model, known for its complexity and inclusion of high noise levels (Oristaglio, 2015). Despite the complex model with high noise levels, the application of DBSCAN gave good results.

Since picking the highest energy for each frequency only allows for picking a single mode, other conditions have to be implemented to allow for multi-mode identification. Wang et al. (2021) applied Gaussian Mixture Models (GMM) instead of the maximum energy criterion to pre-cluster the dispersion images into two clusters: dispersion energy and background noise. Subsequently, DBSCAN is applied and the maximum amplitude for each cluster (mode) is picked to obtain the dispersion curves.

5 Discussion, Conclusion and Outlook to master thesis project

The master's thesis focuses on subsurface characterization in an engineering context, utilizing relatively short array lengths in the range of a few tens of meters. As discussed in Chapter 3, shorter array lengths can compromise spectral resolution, posing challenges in dispersion curve identification, especially when dealing with closely spaced modes. This brings forth a couple of questions to be answered at the start of the master thesis project: whether it suffices for production purposes to reliably pick only the fundamental mode, or if the existing inversion workflow also incorporates information from higher modes. Seeking insights from experienced professionals and analyzing dispersion images from field data for engineering purposes should be conducted to understand energy distribution across modes and assessing spectral resolution on real data sets. Even if focusing initially on the fundamental mode, the algorithm developed in this master thesis should be adaptable to include higher modes, as research (Chamorro et al., 2023) has demonstrated improved reliability and accuracy when higher modes are incorporated.

Unsupervised Machine Learning operates automatically without the need for labeled data, presenting advantages in terms of cost and time efficiency. However, it comes with the challenge of difficult uncertainty assessment. Publications by Rovetta et al. (2021) and Wang et al. (2021) utilize a maximum amplitude approach for picking dispersion curves from mode clusters, but this method has drawbacks, including susceptibility to mode mixing and noise influence. Dispersion curve picking was carried out on sections of kilometer-long receiver arrays for oil and gas exploration, making it questionable wether this approach will be successful in engineering applications.

Supervised algorithms, on the other hand, benefit from having a ground truth for training, which can be modified with noise to better simulate realistic data. Approaches such as using transfer learning (Ren et al., 2021) or purely synthetic data (Chamorro et al., 2023) as training data reduce the need for manually labeled data significantly.

We find the direct training on shot gathers, as demonstrated by Chamorro et al. (2023), very promising, as it offers the possibility to skip the transformation of the shot gather to a different domain and evade associated complications. Conducting performance and robustness analyses, whether to train the data directly on shot gathers or on dispersion spectra, can offer deeper insights into selecting the most suitable supervised algorithm. Much effort has to be put in the creation of suitable training data to cover a broad range of possible geological settings anyway, which could be used as a basis to compare both approaches in the master thesis.

Summarizing, it can be said that if mode mixing is a recurring phenomenon in engineering-applications data, clustering methods may have clear disadvantages, and supervised methods become preferable.

Lastly, it is essential to acknowledge that although ML methods excel at handling large volumes of data, the dispersion curves derived through ML are not inherently grounded in wave theory or other related physical principles. Nonetheless, ML-generated results can offer valuable results in practical applications, if the results are approached with a critical perspective. Consequently, professional review is inevitable in ensuring the reliability and applicability of ML-derived dispersion curves.

References

Alyousuf, T., Colombo, D., Rovetta, D., & Sandoval-Curiel, E. (2018). Near-surface velocity analysis for single-sensor data: An integrated workflow using surface waves, AI, and structure-regularized inversion. *SEG Technical Program Expanded Abstracts 2018*, 2342–2346. https://doi.org/10.1190/segam2018-2994696.1

Chamorro, D., Zhao, J., Birnie, C., Staring, M., Moritz, F., & Ravasi, M. (2023). Deep learning-based extraction of surface wave dispersion curves from seismic shot gathers [Publisher: arXiv Version Number: 1]. https://doi.org/10.48550/ARXIV.2305.13990

Foti, S. (2015). *Surface wave methods for near-surface site characterization*. CRC Press, Taylor & Francis Group.

Foti, S., Hollender, F., Garofalo, F., Albarello, D., Asten, M., Bard, P.-Y., Comina, C., Cornou, C., Cox, B., Di Giulio, G., Forbriger, T., Hayashi, K., Lunedei, E., Martin, A., Mercerat, D., Ohrnberger, M., Poggi, V., Renalier, F., Sicilia, D., & Socco, V. (2018). Guidelines for the good practice of surface wave analysis: A product of the InterPACIFIC project. *Bulletin of Earthquake Engineering*, *16*(6), 2367–2420. https://doi.org/10.1007/s10518-017-0206-7

Hu, S., Zhao, Y., Ma, Z., Zhu, L., Ge, S., Zhang, W., & Wu, F. (2022). Combined analysis of active and passive surface waves for 2d characterization of a landfill. *Journal of Applied Geophysics*, *206*, 104832. https://doi.org/10.1016/j.jappgeo.2022.104832

Khosro Anjom, F., Vaccarino, F., & Socco, L. V. (2024). Machine learning for seismic exploration: Where are we and how far are we from the holy grail? *GEOPHYSICS*, *89*(1), WA157–WA178. https://doi.org/10.1190/geo2023-0129.1

Oristaglio, M. (2015). SEAM update: The arid model — seismic exploration in desert terrains. *The Leading Edge*, *34*(4), 466–468. https://doi.org/10.1190/tle34040466.1

Pan, Y., Gao, L., & Bohlen, T. (2019). High-resolution characterization of near-surface structures by surface-wave inversions: From dispersion curve to full waveform. *Surveys in Geophysics*, *40*(2), 167–195. https://doi.org/10.1007/s10712-019-09508-0

Park, C. B., Miller, R. D., Ryden, N., Xia, J., & Ivanov, J. (2005). Combined use of active and passive surface waves. *Journal of Environmental and Engineering Geophysics*, *10*(3), 323–334. https://doi.org/10.2113/JEEG10.3.323

Park, C. B., Miller, R. D., & Xia, J. (1999). Multichannel analysis of surface waves. *GEOPHYSICS*, *64*(3), 800–808. https://doi.org/10.1190/1.1444590

Rahimi, S., Wood, C. M., & Teague, D. P. (2021). Performance of different transformation techniques for MASW data processing considering various site conditions, near-field effects, and modal separation. *Surveys in Geophysics*, *42*(5), 1197–1225. https://doi.org/10.1007/s10712-021-09657-1

Ren, L., Gao, F., Williamson, P., & McMechan, G. A. (2021). On application issues of automatic dispersion curves picking by machine learning. *First International Meeting for Applied Geoscience & Energy Expanded Abstracts*, 1836–1840. https://doi.org/10.1190/segam2021-3594524.1

Rovetta, D., Kontakis, A., & Colombo, D. (2021). Application of a density-based spatial clustering algorithm for fully automatic picking of surface-wave dispersion curves. *The Leading Edge*, *40*(9), 678–685. https://doi.org/10.1190/tle40090678.1

Shearer, P. M. (2019). *Introduction to seismology* (Third edition). Cambridge University Press.

Socco, L., Foti, S., & Boiero, D. (2010). Surface-wave analysis for building near-surface velocity models — established approaches and new perspectives. *GEOPHYSICS*, *75*(5), 75A83–75A102. https://doi.org/10.1190/1.3479491

Socco, L., & Strobbia, C. (2004). Surface-wave method for near-surface characterization: A tutorial. *Near Surface Geophysics*, *2*(4), 165–185. https://doi.org/10.3997/1873-0604.2004015

Wang, Z., Sun, C., & Wu, D. (2021). Automatic picking of multi-mode surface-wave dispersion curves based on machine learning clustering methods. *Computers & Geosciences*, *153*, 104809. https://doi.org/10.1016/j.cageo.2021.104809