Developing Deep Learning Tools in Earthquake Detection and Phase Picking

by

Hao Mai

A thesis submitted to the University of Ottawa
in partial fulfillment of the thesis requirement for the degree of

Doctor of Philosophy
in
Department of Earth and Environmental Sciences,
Faculty of Science,
University of Ottawa

Examining Committee

The following served on the Examining Committee for this thesis.

External Examiner: Miao Zhang
Assistant Professor,
Dalhousie University

Internal Member(s): Claire Perry
Associate Adjunct Research Professor/Research Scientist,
University of Ottawa/Natural Resources Canada

Mareike Adams
Associate Adjunct Research Professor/Seismic Analyst,
Carleton University/Natural Resources Canada

Dariush Motazedian
Professor,
Carleton University

Supervisor(s): Pascal Audet
Professor,
University of Ottawa

Declaration of Authorship

I hereby certify that the submitted thesis represents results from my own ideas and investigations. This work was completed while I was registered as a candidate for the degree of Doctor of Philosophy at the University of Ottawa, under the supervision of Pascal Audet. Furthermore, I hereby certify that I have not obtained a degree elsewhere on the basis of the research presented in this submitted work. Contemporary doctoral research in the geosciences typically involves collaborative efforts, and this is particularly evident in theses consisting of separate journal articles, such as the one presented here. This thesis consists of three research chapters, two of which have been published as journal articles. Each research chapter includes some contribution from various co-authors. The following discussion elaborates on my specific contributions to these chapters.

Chapter 2 is a journal article titled:
QuakeLabeler: A Fast Seismic Data Set Creation and Annotation Toolbox for AI Applications
Hao Mai and Pascal Audet

I conceptualized this study, as well as developed and tested software coded in the Python programming language, with the goal of simplifying and expediting the process of seismic data set creation and annotation for machine learning applications in seismology. I developed QuakeLabeler, an open-source Python package that streamlines the process of retrieving seismograms from multiple online data centers, querying online human-reviewed catalogs, signal processing, annotation, and data distribution analysis. I implemented various file export formats and designed an interactive command-line interface with three alternative execution modes to provide customized dataset solutions for different types of applications. The code is also packaged with comprehensive tutorials and full API description, for potential extension and use in other applications. My co-author and I wrote the article; I mainly contributed to the software development, data processing, and methodological aspects. This paper is published in *Seismological Research Letters* (Mai and Audet, 2022).

Chapter 3 is a journal article titled:
Blockly Earthquake Transformer: A Deep Learning Platform for Custom Phase Picking
Hao Mai, Pascal Audet, H.K. Claire Perry, S. Mostafa Mousavi and Quan Zhang

I conceptualized this study, as well as developed and tested software coded in the Python programming language. I was the lead developer of the Blockly Earthquake Transformer (BET), a deep learning platform tailored for efficient adaptation of phase pickers in seismology. I designed the interactive dashboard that allows users to customize the deep learning model to their specific dataset. I implemented the transfer learning module, which extends the application of the model to more specific seismic phases, and I contributed to the fine-tuning module, which improves model performance by allowing customization of the model architecture. In addition, I contributed to the development of visualization tools and methods for generating publishable figures from the platform. My co-authors and I collaborated on the writing of the article, where I contributed primarily to the software development and methodology. This paper is published in *Artificial Intelligence in Geosciences* (Mai et al., 2023).

Chapter 4 is a journal article titled:
Investigating the Impact of Dataset Architecture on Deep Learning Models' Performance for Automated Phase-Picking
Hao Mai, Pascal Audet, H.K. Claire Perry and Clément Estève

I conceptualized and led this study, which focused on the performance evaluation of deep learning (DL) models for automated phase picking tasks. I developed and implemented the necessary software using the Python programming language to retrain two popular DL pickers, PhaseNet and EQTransformer. By conducting experiments with different sizes of training datasets and evaluating their impact on phase-picking accuracy using a consistent validation set, I gained insight into the effects of dataset architecture on DL model performance. In addition, I investigated factors such as data preprocessing, standardization, picking methods, and data distributions to understand their influence on training and model performance. The results of this study provide valuable guidance for determining the optimal size of training datasets and selecting appropriate DL models for future studies. As of June 2023, this paper is in preparation for submission to *Journal of Geophysical Research: Solid Earth*.

Declaration Regarding the Use of Artificial Intelligence (AI)

I wish to acknowledge the role of several artificial intelligence (AI) tools, specifically OpenAI's Chat-GPT, DeepL Write, and GitHub Copilot, in the preparation of this thesis. These AI models have served as efficient aids in non-substantive aspects of the work. They have been particularly beneficial in saving considerable time on grammar corrections, as

English is not my first language. In addition, they have helped improve plot formatting, debug LaTeX and Python code, and maintain a high standard of presentation. This has allowed me to focus more on the substantive and creative aspects of my research.

I hereby certify that these AI tools were strictly limited to support roles. All ideas, work, and textual content in this thesis are entirely my own original creations. No content in this work is the result of any bias or point of view introduced by these AI tools. They did not generate or influence any of the ideas or thought processes reflected in this thesis.

In using these AI tools in this way, I have carefully followed the University of Ottawa's guidelines for the responsible use of such technologies in academic work. I affirm that my use of these AI tools is fully consistent with the principles of academic integrity. The use of these tools does not violate academic standards, but rather supports the academic process by allowing for a more efficient use of time and resources. I am fully prepared to answer any questions or concerns that may arise in this regard.

Hao Mai
June, 2023

Abstract

With the rapid growth of seismic data volumes, traditional automated processing methods, which have been in use for decades, face increasing challenges in handling these data, especially in noisy environments. Deep learning (DL) methods, due to their ability to handle large datasets and perform well in complex scenarios, offer promising solutions to these challenges. When I started my Ph.D. degree, although a sizeable number of researchers were beginning to explore the application of deep learning in seismology, almost no one was involved in the development of much-needed automated data annotation tools and deep learning training platforms for this field. In other rapidly evolving fields of artificial intelligence, such automated tools and platforms are often a prerequisite and critical to advancing the development of deep learning. Motivated by this gap, my Ph.D. research focuses on creating these essential tools and conducting critical investigations in the field of earthquake detection and phase picking using DL methods. The first research chapter introduces QuakeLabeler, an open-source Python toolbox that facilitates the efficient creation and management of seismic training datasets. This tool aims to address the laborious process of producing training labels in the vast amount of seismic data available today. Building on this foundational tool, the second research chapter presents Blockly Earthquake Transformer (BET), a deep learning platform that provides an interactive dashboard for efficient customization of deep learning phase pickers. BET aims to optimize the performance of seismic event detection and phase picking by allowing easy customization of model parameters and providing extensions for transfer learning and fine-tuning. The third and final research chapter investigates the performance of DL pickers by examining the effect of training data size and deployment settings on phase picking accuracy. This investigation provides insight into the optimal size of training datasets, the suitability of DL pickers for new target regions, and the impact of various factors on training and on model performance. Through the development of these tools and investigations, this thesis contributes to the application of DL in seismology, paving the way for more efficient seismic data processing, customizable model creation, and a better understanding of DL model performance in earthquake detection and phase-picking tasks.

Acknowledgements

I still clearly remember that rainy night in October 2018, on a train to Chengdu, the home of pandas in Sichuan, China. Relying on the intermittent cell phone signal in the mountainous region, I received an email from Pascal Audet welcoming me to his geophysics research group at the University of Ottawa. At that time, I was torn between staying in China to work after my master's degree or pursuing a Ph.D. in North America. As machine learning was just taking off in China, several companies I interviewed with offered attractive packages, and I felt that I had spent too long in school and it was time to work and give back to my family. Now that I am about to complete my Ph.D., I am deeply grateful to my parents and my entire family for their unwavering support in helping me pursue a doctoral degree. I thank all my family members for giving me the chance to finish what I want to do.

Ottawa has been a wonderful chapter in my life, where I've met many kind people. Ottawa citizens are so nice that I want to spend the rest of my life in this city. I'm very grateful to my supervisor, Pascal Audet, who not only guided me step by step to realize the research I've wanted to do since my undergraduate days, but also helped me grow in many ways. Thank you Pascal, you are the best advisor. At the University of Ottawa, I met a group of enthusiastic professors and colleagues who helped me quickly integrate into the work and life here. I am grateful to Glenn Antony Milne and Mareike Adams for serving as my comprehensive exam examiners and for providing substantial academic guidance, even during the worst period of the pandemic lockdown. I am very grateful to Claire Perry, who, although I got to know you late, provided much guidance and suggestions for my research. I would also like to thank Claude Farley for teaching me how to lead students in field trips and lab sessions, and Lisa-Robin Murphy for helping me get acquainted with everything in the department. And to my colleagues with whom I work day and night, Clement, Jeremy, Stephen, Morgan, Pasan, Taylor, Erica, and Quan, you are the best companions! Thanks also to those who selflessly helped me in research and work, thank you Mostafa Mousavi, YuanYuan Ma, Xiaoming Sun for your valuable advice on my research.

Finally, I can't end this section without mentioning my best friends, the most important people in my life, whose companionship and encouragement have brought me to where I am today. I am grateful to Crystal, a girl much younger than me, who taught me many excellent qualities. Thank you for studying and living with me when I came to Canada alone and for helping me get acquainted with everything. I have many such dear friends in Ottawa, including Emily, Charmaine, Kevin, Annie, Gideon, Zhezhe, Yuan, Kun, Kai, Wenyan and Bruce. Thank you for your help and companionship in life. Of course there are my homies back in China. Thanks to Chao Tang, my close friend for 17 years, for your

unconditional support and belief in me at all times. Thanks to Li Wang and Jiki Zhang for reminding me to stay focused when I face setbacks or distractions. And to Dongyang Liu, Weijie Bai and Liang Wang, my goofy friends who have brought me endless joy. Even though we are thousands of miles apart, we chat every day! Thanks to Haoyuan, Xinke, and all my soccer teammates! Finally, to my wife, Sherry, thank you, you are the best!

To Sherry, TC and @KELECRYS,
who make my world magical.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Deep Learning: A Game Changer in Diverse Fields

In recent years, deep learning has rapidly grown and to become a game-changer in many fields of science and technology, such as image and speech recognition (Wang et al., 2020; Nassif et al., 2019), natural language processing (Otter et al., 2020), and computer vision (Shrestha and Mahmood, 2019). The advancements in hardware and software have enabled deep learning algorithms to process and analyze large amounts of data, leading to breakthroughs in many industries. For instance, in healthcare, deep learning is being used to analyze medical images and improve diagnostic accuracy (Esteva et al., 2019). In finance, it is being used to detect fraudulent transactions and predict stock prices (Huang et al., 2020). In transportation, it is being used to develop self-driving cars and improve traffic flow (Veres and Moussa, 2019). Deep learning has drastically transformed how we live and work.

Although a plethora of impressive models have been proposed recently, one cannot overlook the fact that the application of deep learning in seismology is still in its infancy (Mousavi and Beroza, 2022a). It is thus appropriate at the beginning of this thesis to present some exceptional applications of deep learning in various disciplines outside the geosciences. Examining the progress and successes of deep learning in other rapidly evolving fields can provide valuable insights for seismologists looking to make further progress. For example, PhaseNet, a U-net structure originally applied to biomedical image segmentation tasks, has been repurposed for earthquake detection (Zhu and Beroza, 2019). The U-network is capable of accurately identifying seismic phase arrivals within continuous seismic recordings, analogous to cell boundaries in biomedical images (Du et al., 2020).

Similar anology has paved the way for the introduction of numerous deep learning solutions in seismology (Jiao and Alavi, 2020). Therefore, it is beneficial to understand the most common deep learning models in other fields, as they may hold the potential to provide meaningful insights for the seismological community.

The following are some of the most significant projects in the field of deep learning:

1. AlphaGo: Developed by Google DeepMind, AlphaGo is a deep learning model that notably defeated human world champions at the strategic board game Go (Silver et al., 2016). Known to be more complex than games like chess, Go posed a significant challenge to computer algorithms. AlphaGo uses a combination of two deep neural networks, the "policy network" and the "value network," to strategize its moves (Li et al., 2020). These networks are trained using a dataset consisting of thousands of expert human Go games, as well as data generated by the networks playing against each other. The AlphaGo model represents a major breakthrough in artificial intelligence, demonstrating the power of deep learning algorithms for complex and strategic tasks.

2. DeepFace: Facebook's DeepFace is a deep learning model capable of recognizing human faces in images with near-human accuracy (Taigman et al., 2014). The model uses a convolutional neural network (CNN) architecture trained on a large dataset of faces. Achieving an impressive accuracy of 97.35% on the LFW (Labeled Faces in the Wild) dataset (Huang et al., 2008), DeepFace has been widely used in various computer vision tasks, such as face recognition in mobile applications (Soyata et al., 2012).

3. U-Net: Invented by Olaf Ronneberger, Philipp Fischer, and Thomas Brox at the University of Freiburg, U-Net is a deep learning algorithm typically used for biomedical image segmentation tasks (Ronneberger et al., 2015). U-Net's architecture is symmetric, consisting of a contracting path (the encoder) and an expanding path (the decoder). The model is particularly well suited to image segmentation tasks, where the goal is to label each pixel in an image (Krithika alias AnbuDevi and Suganthi, 2022). It is also adept at handling large amounts of input and output data and images of high resolution and complexity (Du et al., 2020).

4. ResNet(Residual Network): Developed by a team at Microsoft Research Asia, ResNet is a convolutional neural network (CNN) architecture designed to solve the vanishing gradient problem common in deep neural networks (He et al., 2016). The architecture includes a series of "residual blocks" that allow the network to learn residual functions

with respect to the layer inputs. ResNet's design facilitates the training of deeper networks without the risk of overfitting, and its architecture is widely used in industry because of its simplicity, trainability, and performance (Wu et al., 2019).

5. YOLO (You Only Look Once): Developed by Joseph Redmon and Ali Farhadi at the University of Washington, YOLO is an innovative real-time object detection system that uses a single neural network to predict bounding boxes and class probabilities within an image (Redmon et al., 2016). Due to its real-time performance and high accuracy, YOLO has been adopted for diverse applications including self-driving cars, security systems, and robotics (Jiang et al., 2022b).

6. Chat-GPT: Created by OpenAI, Chat-GPT is a pre-trained large language model that generates text content that mimics human responses (Floridi and Chiriatti, 2020). It uses a transformer architecture, a type of neural network that encodes input language through an attention mechanism to process sequential data (Han et al., 2021). Chat-GPT's ability to generate human-like text has led to its use in various natural language processing tasks, including language translation, text rephrasing, content creation, and even debugging programming code. It is arguably one of the most valuable deep learning models to be released in 2022 (Firat, 2023).

It is important to note that this list does not include all the advances in deep learning. In fact, the pace of progress in deep learning research is rapid and ongoing, driven by by increases in computing power, the availability of massive datasets, and the continued development of innovative architectures and algorithms (Alzubaidi et al., 2021).

When I started the second round of dissertation revisions in May 2023, Chat-GPT had just released version 4.0 (Sanderson, 2023). In this iteration, Chat-GPT has become an almost universal assistant. It offers assistance in creating resumes, writing poems, composing emails, etc (Liu et al., 2023). In the academic field, it provides invaluable insights, from correcting grammar to summarizing the content of academic papers to suggesting potential directions for future research (Bubeck et al., 2023). The emergence of Chat-GPT highlights the potential of AI not only to replace many time-consuming and labor-intensive traditional tasks, but also to bring creative ideas to our work (Hill-Yardin et al., 2023; McGee, 2023; Surameery and Shakor, 2023). AI assistants like Chat-GPT are likely to proliferate in various industries in the coming years, significantly increasing productivity. One can imagine a future where AI-assisted analysts help process raw seismic data, leaving human seismic analysts to focus on the final interpretation decisions. This could dramatically improve both the speed and quality of routine seismic data processing.

1.2 Understanding Deep Learning: From Basic Building Blocks to Complex Networks

Deep learning may seem complex, but it's not too complicated to understand or put into practice. Deep learning is a specific way of building artificial neural networks (ANNs). ANNs were first proposed in the 1940s and 1950s (Agatonovic-Kustrin and Beresford, 2000). In design, ANNs can learn from data in a way similar to how human brain cells work (Yegnanarayana, 2009). The essence of a deep learning model is a very deep (and complicated) ANN structure consisting of layers of interconnected nodes called artificial neurons(Gershenson, 2003). In this way, a deep learning model can train a mapping from data to desired prediction results. For example, consider a deep learning model used for image recognition. The model is trained to map pixel data (the input) to object labels (the desired prediction) (Horak and Sablatnig, 2019). As another example, in seismology, a deep learning model could be trained to map seismic waveform data (the input) to earthquake events (the desired prediction) (Bao et al., 2021) (see Figure 1.1).

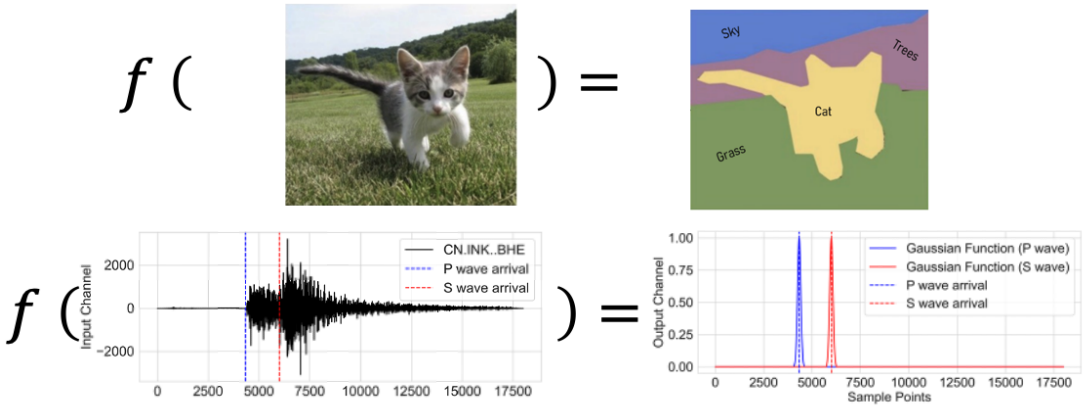


Figure 1.1: **Deep learning: mapping data to prediction results.** Top figure shows in image recognition, deep learning is to find a mapping from image data to object names. Bottom figure shows in seismology, deep learning is to find a mapping from seismogram to potential event probabilities.

An artificial neuron, also known as a perceptron, is the basic building block of a deep learning framework (Noriega, 2005). It consists of input connections, a linear function, a nonlinear (activation) function, and output connections (Figure 1.2). The input connections receive inputs from other neurons or the input channel of data. The linear function

takes the inputs $x_1, x_2, ..., x_n$, multiplies the weights $w_1, w_2, ..., w_n$ and adds a bias term b, then sends the numerical values to the activation function. The activation function is used to introduce nonlinearity into the network, such as sigmoid, tanh, and ReLU functions (Sharma et al., 2017). Without an activation function, a neural network can only model linear relationships and cannot model nonlinear relationships present in the data. The output connections send the output of the neuron to other neurons or to the final output channel of the network (Blundell et al., 2015). In this way, the simple computation inside the neurons eventually builds into a comprehensive network architecture. During training, millions of parameters (weights and biases) in a deep learning model iteratively find their optimal values from the input data. This is the microscope of how neurons work much like brain cells, i.e. millions of trainable parameters working together to have the ability to "understand" the data provided and make rational predictions for new input data.

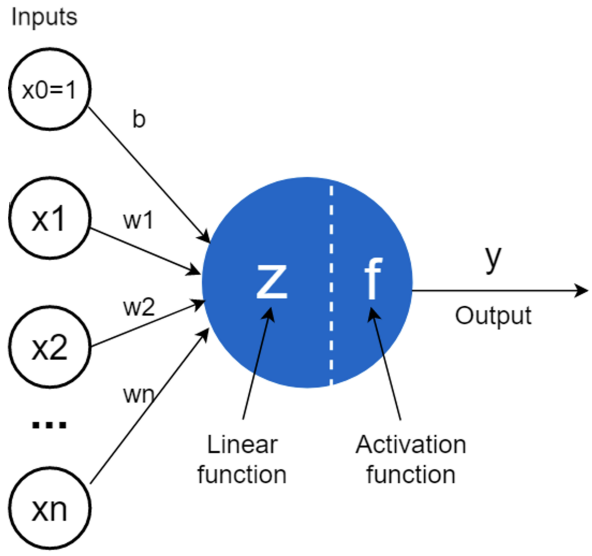


Figure 1.2: **Schematic of an artificial neuron.** Artificial neurons (also called perceptrons) are the simplest elements in a deep learning model. They are inspired by biological neurons that are found in the human brain.

The macroscopic framework for deep learning is the layered neural network (see Figure 1.3). The structure of a neural network is determined by the number of layers (also called depth), the number of neurons in each layer (also called nodes), and the connections between the neurons. The input layer, which is the first layer of a neural network, is where

the input data is sent. The final prediction or classification is made by the output layer, which is the last layer in the algorithm. Between the input and output layers, there may be one or more hidden layers that process the data and extract features from the input data (Stathakis, 2009). .

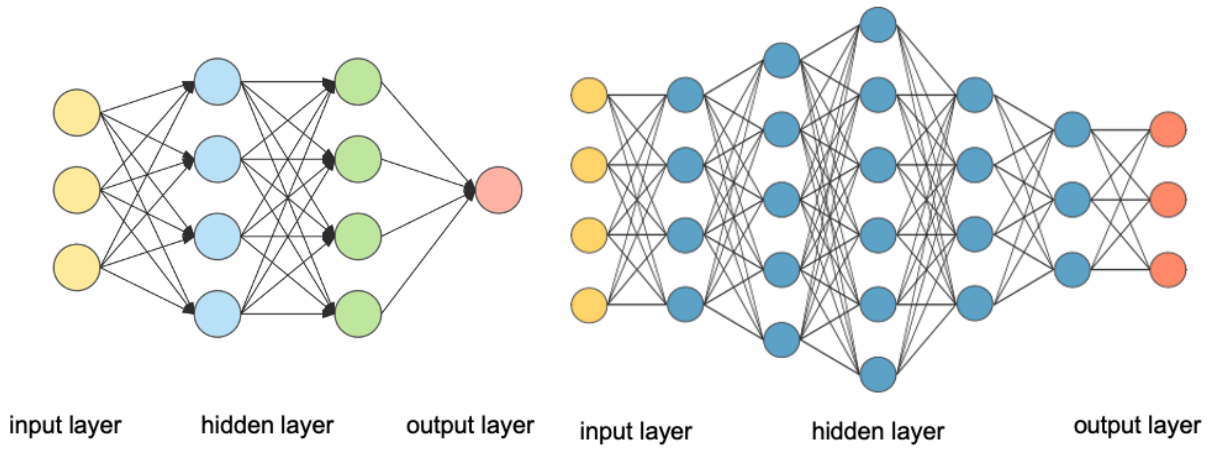


Figure 1.3: **Schematic diagrams of neural networks.** This diagram illustrates two types of neural networks. The left diagram shows a simpler network with three input nodes, two hidden layers, and one output node. The right diagram shows a more complex network with four input nodes, five hidden layers, and three output nodes. These diagrams illustrate the variety of neural network structures determined by the number of layers (depth), the number of neurons (nodes), and the connections between neurons. The optimal configuration of hidden layers and neurons remains an open question in deep learning research.

Over the years, the architecture of deep learning networks has evolved significantly, becoming more complex and sophisticated to handle a wider range of tasks and data types. This evolution has led to the development of new network structures that enhance the learning capabilities of these models. One such structure is the self-attention mechanism, which allows the model to focus on different parts of the input data depending on their relevance to the task at hand (Shaw et al., 2018). This mechanism has proven particularly useful in tasks involving sequential data, such as natural language processing, where the context of a word may depend on words that appear much earlier or later in the sequence (Soydaner, 2022). Another notable development is the introduction of ResNets. These

structures contain shortcut connections that allow the gradient to be more effectively backpropagated to earlier layers, mitigating the problem of vanishing gradients in deep networks. ResNets have been instrumental in enabling the training of very deep networks, leading to improved performance on a variety of tasks (He et al., 2016). Despite these advances, determining the optimal structure for a deep learning network, including the number of layers, the number of neurons in each layer, and the specific types of structures to include, remains a challenging problem. Research in this area is ongoing, and while there is no universally accepted solution yet, the continued development of new structures and training techniques promises to further enhance the capabilities of deep learning models in the future (Simoulin and Crabbé, 2021).

Building a deep model framework is only one step in deep learning, the next steps are training, testing, and deployment (Paul et al., 2021). During training, the model learns to interpret the data by adjusting the weights and biases of the neurons. The goal is to minimize the difference between the predicted output and the actual output (also called the ground truth) (Géron, 2022). This process uses an optimization algorithm, such as stochastic gradient descent (SGD), which updates the weights and biases based on the discrepancy between the predicted and actual output (Bottou, 2012). The error is quantified by comparing the model's predictions with the actual output for each corresponding input-output pair in the training dataset(Murphy, 2012).

As training progresses, the optimization algorithm iteratively refines the weights and biases using the gradients of the error function with respect to these parameters. The computation of these gradients is accomplished by the backpropagation algorithm, a method that propagates the error back through the network, systematically updating the weights and biases (Amari, 1993). The algorithm continues to adjust the weights and biases until the error is minimized, or until a predefined stopping criterion is met, marking the end of the training phase.

The trained model is then tested on another dataset, often called the validation or test dataset. This dataset is different from the one used in the training phase and allows for the evaluation of the model's generalization ability - its ability to perform well on unseen data (Eelbode et al., 2021). The performance of the model during this testing phase provides a more realistic assessment of its effectiveness.

After successful testing, the model is ready for deployment, where it can be used to solve real-world problems(Guo et al., 2019). This could involve integrating the model into an existing software application or using it to inform decision making in fields as diverse as healthcare, finance, geoscience, and beyond. In this way, the deep learning model becomes a powerful tool that transforms data into actionable insights(Provost and Fawcett, 2013;

Shrestha and Mahmood, 2019).

In summary, building a deep learning model is not just about building a network architecture. It's a comprehensive process that involves the careful orchestration of several steps, from design and training to testing and deployment (see Figure 1.4), all aimed at creating a model that can effectively learn from data and make accurate predictions(Wan et al., 2019).

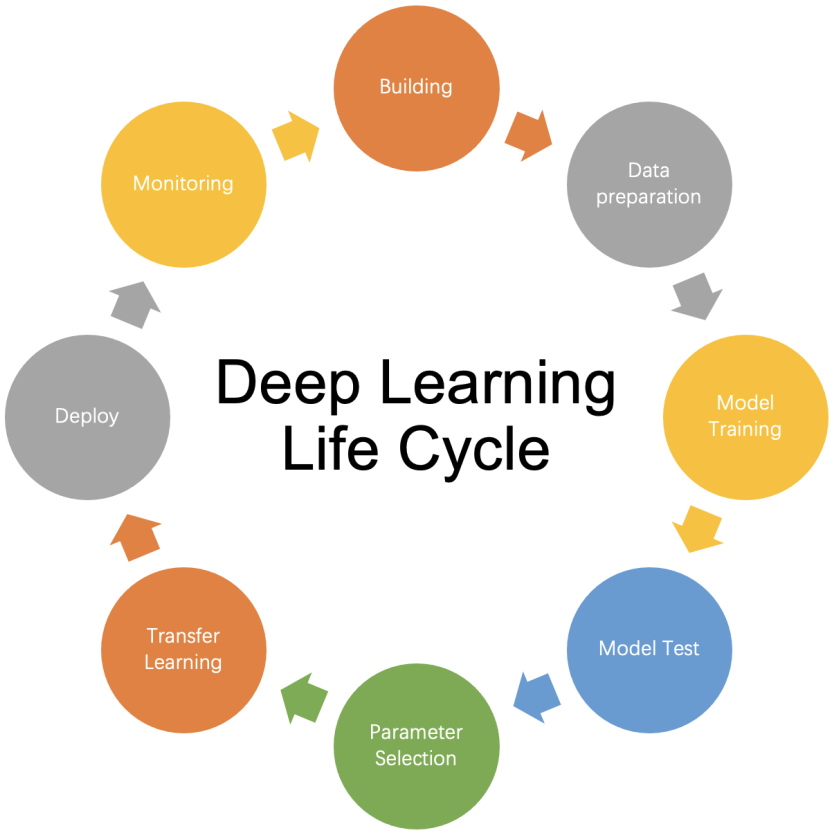


Figure 1.4: **The life cycle of deep learning .** This diagram illustrates the deep learning life cycle, which includes the key stages of data preparation (collection and preprocessing), model training and validation, deployment, and ongoing monitoring. This cycle provides a comprehensive overview of the process involved in developing and implementing a deep learning solution.

1.3 Understanding the Seismological Problems: Earthquakes, Detection Techniques, and Catalogues

In the field of seismology, the term "earthquake" encompasses more than just the natural disasters commonly recognized by the public. Physically, an earthquake results from the sudden release of energy in rocks, creating ground motion (McLaskey, 2019). This energy propagates outward from the fault line in the form of seismic waves. More precisely, seismology is the study of seismic waves. The source of seismic waves is not limited to fault ruptures. It also includes volcanic eruptions, glaciers and landslides, as well as blasts, hydraulic fracturing and air guns made by human activities (Hill et al., 2002; Nettles and Ekström, 2010; Rodrıguez et al., 1999; Holland, 2013; ZHAO et al., 2008; Doocy et al., 2013).

With the ongoing enhancement of the global network of seismic monitoring stations, there is a notable rise in the number of detectable earthquakes each year (Weatherill et al., 2016). It's estimated that over half a million earthquakes are detected around the globe annually, of which around 20% can be felt by humans, and approximately 100 events have the potential to cause harm to human life and property.

Highly destructive earthquakes are a current concern in seismology (Yang and Yao, 2021). Recently, more and more governments and researchers have recognized the need to accelerate the development of earthquake early warning systems (EEW). Earthquake early warning is the rapid detection of earthquakes, real-time estimation of the shaking hazard, and notification of expected shaking. Owing to the constraints of present-day scientific technology, we are yet to achieve the ability to precisely forecast earthquakes in the short term (Cremen and Galasso, 2020; Allen et al., 2009). Consequently, the alerts furnished by EEW systems take on an outsized significance in the quest to safeguard lives and property. This is particularly significant for the west coast of Canada, where, according to various geological and geophysical data, there's a very real possibility of a major earthquake striking within the next hundred to two hundred years (Schlesinger et al., 2021). In addition, the Ottawa River Valley and the Saint Lawrence Seaway are also in urgent need of an EEW system due to the high seismic risk based on historical records and considering the concentration of population and infrastructure(Seywerd et al., 2022; Kohler et al., 2020; Crane et al., 2021, 2019).

When we shift our focus to small magnitude earthquakes (also called small events), a very different but fascinating topic emerges. It is certain that countless small earthquakes evade detection every day, slipping past seismic analysts and remaining absent from catalogues (Shebalin et al., 2020). Such small seismic events are often imperceptible

to humans, and their relatively small amplitudes can easily be obscured in seismograms by background noise (Ross et al., 2019). Nevertheless, these small events deserve attention. Accurate detection of small earthquakes helps to enrich existing earthquake catalogues (Brodsky, 2019). Studying these events can improve scientists' understanding of earthquake mechanisms, allow for more nuanced interpretations of tectonics, and pave the way for future breakthroughs in earthquake forecasting(Ide, 2019).

We are now aware that the term "earthquake" envelops more than just catastrophic natural phenomena. Earthquakes can vary in magnitude and source, and can even be human-induced (Campbell et al., 2020; Li et al., 2017). With such a plethora of event information sporadically surfacing on the seismometers at different seismic stations, how do we identify earthquake events, i.e., how do we find seismic signals on a seismogram? The problem of robustly detecting earthquakes is fundamental in seismology. Earthquake detection methods are technical operations that help seismologists identify an earthquake. This problem is often treated simultaneously with seismic phase picking, because we usually determine the type of earthquake event during earthquake detection (Xie et al., 2020). Traditional earthquake detection methods rely primarily on manual seismic phase picking (by human seismic analysts), a process that involves identifying the arrival times of different types of seismic waves on a seismogram (Tozzi et al., 2020). This process can be time-consuming and prone to human error, especially when dealing with small events of low magnitude that are often buried in background noise. However, it remains the most reliable method of earthquake detection to date. Especially when earthquake centers publish their official catalogues, human review is indispensable (Dryhurst et al., 2022).

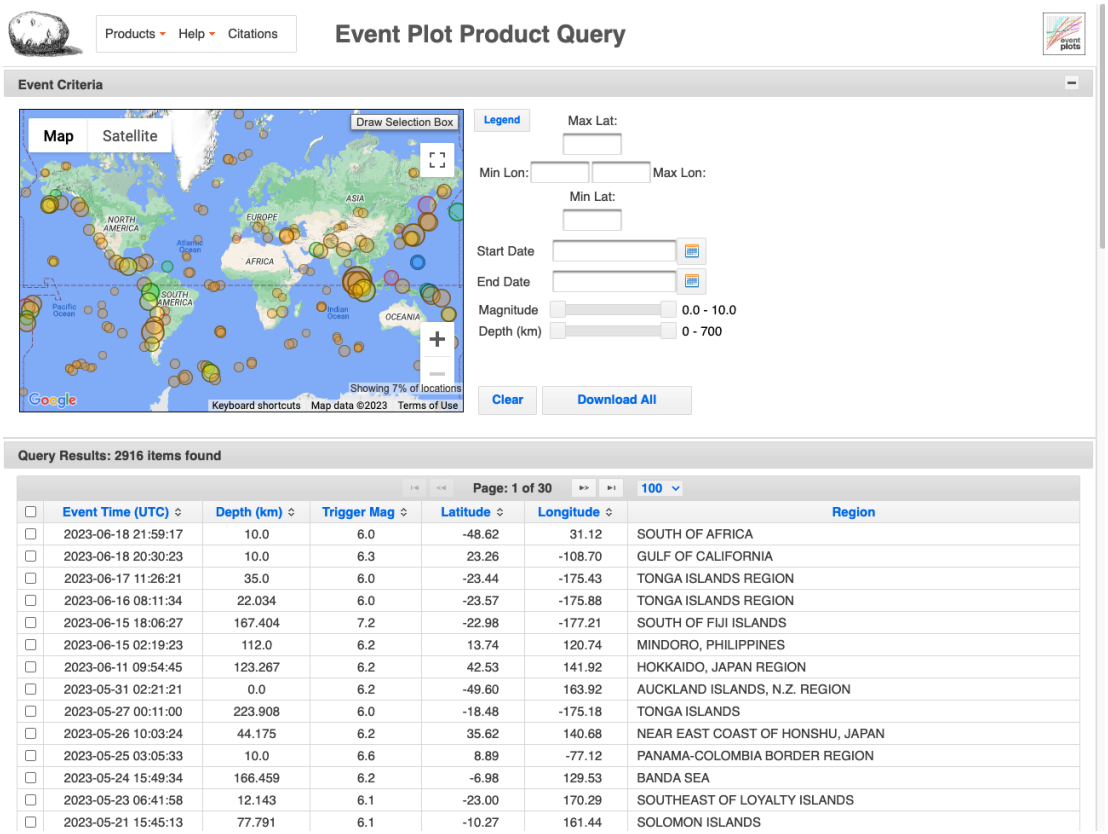
The development of automatic detection techniques, which began about a decade ago, including the Short-Term Average/Long-Term Average (STA/LTA) and Akaike Information Criterion (AIC) methods, marked a significant advance by enabling faster and more objective discrimination of seismic events (Trnkoczy, 2009). STA/LTA and AIC methods have been widely used in various earthquake detection scenarios, such as EEW, microseismic monitoring, and picking first arrivals in industrial-level seismic data, among others (D'Angelo et al., 2020; Long et al., 2019). However, these conventional automatic methods are often hampered by their limited detection capabilities in noisy environments, and the optimization of threshold selection relies heavily on the expertise of seismologists (Hongli and Han, 2020).

In recent years, the incorporation of deep learning techniques has shown immense potential for improving the accuracy and efficiency of earthquake detection (Mousavi and Beroza, 2022a). However, the stability of deep learning-based methods is still under scrutiny, as they sometimes detect more small events, but also introduce an increased number of false positives. In addition, the practical implementation of deep learning methods in real-world

operations is still a work in progress (Zhu et al., 2022b). Nevertheless, the outlook for deep learning-based automated detection methods to replace traditional STA/LTA and AIC methods is quite optimistic. After all, deep learning techniques have broken through the limitations of traditional detection methods and brought to light more minor seismic events (Wang et al., 2019b).

After processing the raw seismic data of a study area and obtaining the detected events, the next critical step is to establish an earthquake catalogue for that region. Earthquake catalogues serve as indispensable tools in seismology by offering complete records of seismic events within a specified time frame and region (Rovida et al., 2020). Typically, these catalogues contain information about each event's date, time, location, depth, and magnitude, as well as other relative data like the type of faulting and the earthquake's focal mechanism or moment tensor solutions (Rovida et al., 2019). Earthquake catalogues are fundamental resources for research in seismic hazard assessment, the study of large earthquake triggering and clustering, and the inference of the Earth's internal structure (Herrmann and Marzocchi, 2021). Besides, seismologists can benefit from earthquake catalogues to analyze the distribution, frequency, magnitude, and effects of earthquakes. Many organizations maintain earthquake catalogues, such as the United States Geological Survey (USGS) which maintains the ANSS Comprehensive Earthquake Catalog (ComCat). This catalogue contains earthquake source parameters and other products produced by contributing seismic networks (Barnhart and Wolfe, 2022). Similar organizations which publish global earthquake catalogues are International Seismological Centre (ISC) and IRIS (Incorporated Research Institutions for Seismology). ISC is an organization that collects and compiles data on earthquakes from over 130 agencies worldwide (Storchak et al., 2020). The main purpose of the ISC is to compile the ISC Bulletin, which is regarded as the definitive record of the Earth's seismicity (Di Giacomo et al., 2021). IRIS provides access to earthquake catalogues through its Data Management Center (DMC) (Trabant et al., 2012). Various relevant products are provided by IRIS to better understand the catalogues (Quinteros et al., 2021). For example, the IRIS DMC's event plot product (see Figure 1.5) is a suite of plots that are automatically generated earthquakes larger than magnitude 6 (DMC, 2011).

However, the construction of reliable and complete earthquake catalogues is a challenging task, especially for individual researchers. It requires not only the accurate detection of seismic events but also their precise location and the determination of their characteristics (Liu et al., 2020). Manual inspection and analysis of seismic data for catalogue construction is a labor-intensive and time-consuming task. Moreover, catalogues often suffer from incompleteness and bias, especially for small events that can be difficult to detect and accurately locate (Jiang et al., 2021). The application of deep learning methods, can help



Event Time (UTC)	Depth (km)	Trigger Mag	Latitude	Longitude	Region
2023-06-18 21:59:17	10.0	6.0	-48.62	31.12	SOUTH OF AFRICA
2023-06-18 20:30:23	10.0	6.3	23.26	-108.70	GULF OF CALIFORNIA
2023-06-17 11:26:21	35.0	6.0	-23.44	-175.43	TONGA ISLANDS REGION
2023-06-16 08:11:34	22.034	6.0	-23.57	-175.88	TONGA ISLANDS REGION
2023-06-15 18:06:27	167.404	7.2	-22.98	-177.21	SOUTH OF FIJI ISLANDS
2023-06-15 02:19:23	112.0	6.2	13.74	120.74	MINDORO, PHILIPPINES
2023-06-11 09:54:45	123.267	6.2	42.53	141.92	HOKKAIDO, JAPAN REGION
2023-05-31 02:21:21	0.0	6.2	-49.60	163.92	AUCKLAND ISLANDS, N.Z. REGION
2023-05-27 00:11:00	223.908	6.0	-18.48	-175.18	TONGA ISLANDS
2023-05-26 10:03:24	44.175	6.2	35.62	140.68	NEAR EAST COAST OF HONSHU, JAPAN
2023-05-25 03:05:33	10.0	6.6	8.89	-77.12	PANAMA-COLOMBIA BORDER REGION
2023-05-24 15:49:34	166.459	6.2	-6.98	129.53	BANDA SEA
2023-05-23 06:41:58	12.143	6.1	-23.00	170.29	SOUTHEAST OF LOYALTY ISLANDS
2023-05-21 15:45:13	77.791	6.1	-10.27	161.44	SOLOMON ISLANDS

Figure 1.5: **Screenshot of the IRIS DMC's event plot product.** The IRIS DMC's event plot product is a suite of plots that are automatically generated following all earthquakes of magnitude 6 or greater. The plot suite uses all open broadband data, or a sub-selection, available at the IRIS DMC at the time the product was generated.

improve the completeness and reliability of earthquake catalogues by automating the detection and characterization of seismic events, even those of small magnitude (Jiang et al., 2022a).

Understanding the seismic problems related to earthquakes, their detection, and cataloguing is crucial in the field of seismology. Deep learning techniques can help tackle these challenges, providing more efficient and accurate solutions. However, their application to seismology is not without its own challenges and issues, which will be discussed further in this thesis.

1.4 Incorporating Deep Learning in Seismology

1.4.1 The Importance of Benchmark Datasets

The creation and use of benchmark datasets is fundamental to the advancement of deep learning research in seismology. Not only do they ensure the reliability and validity of research, but they also foster collaboration and innovation within the seismological community (Mousavi et al., 2019a). As more high-quality and diverse benchmark datasets become available, they will undoubtedly play an increasingly important role in the future of seismological research. Their importance lies in several key aspects:

- **Model Evaluation and Comparison**: Benchmark datasets provide a standard against which different models can be evaluated and compared (Magrini et al., 2020). By testing different models on the same dataset, researchers can objectively evaluate their performance in terms of accuracy, efficiency, and generalizability. This allows the scientific community to identify the most promising models and approaches, thereby advancing the field.

- **Promoting Reproducibility**: Reproducibility is a cornerstone of scientific research. In deep learning, access to the same dataset allows researchers to replicate experiments and validate results reported by others. This not only ensures the integrity of the research, but also promotes transparency and trust within the scientific community (Sun et al., 2020).

- **Enabling Collaborative Improvement**: Benchmark datasets serve as common ground for researchers around the world (Banbury et al., 2021). By working on the same dataset, researchers can share their results, learn from each other, and collaboratively improve models and techniques. This kind of collective effort accelerates the progress of deep learning applications in seismology.

- **Easing Model Generalization**: Benchmark datasets often include a wide range of geologic settings and event types, which helps to train models that generalize well to unseen data (Saad et al., 2021). This is particularly important in seismology because seismic events are inherently unpredictable and vary widely in location, depth, and magnitude.

- **Training and Testing New Models**: Benchmark datasets provide a wealth of data needed to train deep learning models. These datasets allow the models to learn

complex patterns in the seismic data and adjust their parameters to minimize errors. The same datasets, or portions of them, can be used to test the models, providing an estimate of how well the models have learned and how they might perform on new, unseen data (Gu et al., 2022).

1.4.2 Currently Available Seismic Benchmark Datasets

As mentioned in the previous section, benchmark datasets provide a common ground for researchers to compare and improve different methodologies. In the field of seismology, several benchmark datasets have been curated and serve as invaluable resources for researchers around the world. Here are some of the most important:

1. **STanford EArthquake Dataset (STEAD)**: is a global dataset of seismic signals for AI research (Mousavi et al., 2019a). It contains local earthquake waveforms and seismic noise waveforms free of earthquake signals, totaling approximately 1.2 million time series or more than 19,000 hours of recordings. The dataset has been collected, quality controlled, and processed using advanced techniques to ensure accurate labeling and robust models. STEAD is the first published dataset which offers new and unprecedented opportunities for seismology and AI research by providing high-quality, large-scale, and global data.

2. **Machine Learning Asset Aggregation of the PDE (MLAAPDE):** is a comprehensive dataset that includes a waveform archive and a feature-labeled catalog, making it a valuable resource for training machine learning models in seismology (Yeck et al., 2021). The data, primarily from the Preliminary Determination of Epicenters (PDE), span the period from July 2013 to December 2020. The PDE catalog, which characteristically includes earthquakes above $M4.5$ globally and above $M2.5$ in the United States, also includes smaller events which fall below $M2.5$. MLAAPDE data files consist of catalogs in CSV and waveform archives in HDF5 format. To further assist researchers, the Python module 'neic-mlaapde' is provided for easy data selection and loading.

3. **INSTANCE:** contains 1.2 million 3C waveform traces from about 50,000 earthquakes and over 130,000 noise traces recorded by the Italian National Seismic Network and other networks from January 2005 to January 2020 (Michelini et al., 2021). The dataset is representative of the seismicity and crustal structure of Italy and the surrounding area, covering a wide range of earthquakes with magnitudes from 0.0 to 6.5, depths from 0 to 550 km, and epicentral distances from 0 to 700 km. This

dataset, available in HDF5 format, also includes different seismic sources such as crustal, subduction and volcanic events, making it versatile for various applications including seismic monitoring, earthquake detection and seismic hazard assessment.

4. **DiTing:** is a comprehensive seismological artificial intelligence training dataset constructed from the 2013-2020 seismic cataloging reports of the China Earthquake Networks Center (Xue et al., 2021). It has the largest known total time length and contains 2,734,748 three-component waveform traces from 787,010 regional seismic events. These waveforms are accompanied by corresponding P- and S-phase arrival time labels and 641,025 P-wave first-motion polarity labels. With a variety of descriptive parameters such as epicentral distance, back azimuth, and signal-to-noise ratios, DiTing provides a rich resource for data-driven seismological research (Zhao et al., 2022). The dataset supports multiple applications, including earthquake detection, seismic phase picking, earthquake magnitude prediction, and early warning systems, thereby promoting the use of AI in seismology.

It is important to note that while these datasets provide a solid foundation, the continued development of more comprehensive, diverse, and accessible datasets is key to further advancing deep learning applications in seismology.

1.4.3 Recent Advances and Applications of Deep Learning Methods in Earthquake Detection

Deep learning has the potential to revolutionize seismology by providing solutions to a variety of challenges and enhancing existing methods. Recent review articles have highlighted the innovative applications of deep learning in areas such as data processing automation, forward problems, inverse problems, and exploratory data analysis (Mousavi and Beroza, 2022a). In each of these areas, deep learning can improve the performance, efficiency, and accuracy of seismological methods and models. However, in this section I do not intend to discuss exhaustively the applications of deep learning in all branches of seismology. Instead, I will use this limited space to present some of the most successful applications of deep learning in seismology to date. As you will see in the following reading, these examples focus primarily on data processing automation, or more specifically, earthquake detection (and phase picking).

So what are the benefits? The impressive performance of deep learning in earthquake detection (and phase picking) suggests that it has the potential to become the next generation of seismic data processing automation. It could fully or partially replace manual

processes and significantly improve the efficiency of existing earthquake processing methods (Saad et al., 2021). Moreover, the mature applications in earthquake detection (and phase picking) can be quickly transferred to other related research areas, such as seismic event monitoring, seismic event discrimination, earthquake early warning, and various post-processing studies (Zhu et al., 2022b).

PhaseNet is currently one of the most widely used and mature methods for earthquake detection. It's a deep neural network-based method for seismic phase picking, which involves measuring the arrival times of P and S waves within an earthquake signal (Zhu and Beroza, 2019). PhaseNet takes three-component seismic waveforms as input and generates probability distributions of P-arrivals, S-arrivals, and noise as output. It's trained on a large dataset of analyst-labeled P and S arrival times from the Northern California Earthquake Data Center. When applied to the waveforms of known earthquakes, PhaseNet achieves much higher picking accuracy and recall rates than existing automatic picking methods. It can also be applied to continuous data for earthquake detection by extracting arrival times from the peaks of probability distributions. Based on my personal experience, I believe that PhaseNet has several advantages: 1. It's lightweight, with fast training and deployment speeds. 2. It's sensitive to small earthquake events hidden in high background noise (although this can sometimes lead to a higher number of false events). 3. It has a high success rate for transfer learning, requiring fewer training samples.

The second model I'd like to discuss is the Earthquake Transformer. Similar to PhaseNet, the Earthquake Transformer is a deep neural network based method for simultaneous earthquake detection and phase picking (Mousavi et al., 2020). It identifies earthquake signals and measures the arrival times of P- and S-waves within them. The Earthquake Transformer takes single-station, three-component seismic waveforms as input and generates probability distributions of earthquake signals, P-arrivals, and S-arrivals as output. Earthquake Transformer is trained on a large dataset of globally distributed earthquake and noise waveforms from the STanford EArthquake Dataset (Mousavi et al., 2019a). Earthquake transformer incorporates a hierarchical attention mechanism to capture the local and global features within the earthquake waveforms. Earthquake transformer outperforms previous deep learning and traditional methods in terms of detection and picking performance, generalization, and computational efficiency. PhaseNet and Earthquake Transformer serve as prototypes for most of the deep learning and transfer learning methods published to date. The more complex network structure of the Earthquake Transformer gives it a higher level of understanding of earthquake waveforms(Mousavi et al., 2020). The advent of the Earthquake Transformer has further enhanced the automation capabilities of earthquake detection, marking a significant advancement in the field of data processing automation in seismology.

In addition to existing deep learning models, we're seeing the emergence of deep learning platforms in earthquake detection research. SeisBench is a prime example. It's an open source software package for developing and applying machine learning methods in seismology (Woollam et al., 2022). SeisBench provides a unified interface for accessing both state-of-the-art models and benchmark datasets for various seismological tasks such as earthquake detection, phase picking, magnitude estimation, and source inversion. SeisBench also provides a range of common processing and data augmentation operations through an API. SeisBench is designed to be extensible, and community participation is encouraged to extend the package. SeisBench aims to facilitate faster model development, fair comparison, and wider adoption of machine learning techniques within the seismological community.

Deep learning platforms like SeisBench are critical to the seismology community for several reasons. First, they provide a unified and standardized environment for developing and testing machine learning models. This standardization is key to fair and meaningful comparisons between different models and techniques (Mittal, 2019). Second, these platforms often come with pre-loaded benchmark datasets, saving researchers the time and effort of collecting and cleaning their own data. This allows them to focus more on model development and less on data preparation (Mu and Zeng, 2019). Third, deep learning platforms facilitate collaboration and knowledge sharing within the community (Bisong, 2019). They allow researchers to build on each other's work, accelerating the pace of innovation and discovery in the field. Finally, these platforms are helping to democratize machine learning in seismology. By providing user-friendly interfaces and comprehensive documentation, they make advanced machine learning techniques accessible to researchers who may not have a background in computer science or data science (Ouyang et al., 2019). This broadens the pool of people who can contribute to advances in the field.

1.5 Thesis Objectives

The application of deep learning in earthquake detection faces a number of significant challenges, which this thesis aims to address. Among these challenges, the issue of data availability stands out. At present, data preparation is a significant challenge. It is the first and most time-consuming step in the deep learning life cycle (Strodthoff et al., 2020). For most developers or researchers, the independent preparation of large-scale standard data sets within a reasonable research timeframe is nearly impossible. Despite the availability of several publicly accessible benchmark datasets, the shortage of high-quality training data severely hampers the progress of deep learning in seismology (Mousavi and Beroza,

2023). This problem is exacerbated by the inaccessibility or unavailability of labeled data in some research areas, substandard data quality, significant errors in manual annotation, and lack of labeled data for specific event discrimination tasks such as volcanic activity, human-made explosions, and nuclear tests (Zöller and Huber, 2021; Magrini et al., 2020; Pérez et al., 2020). Thus, one of the primary objectives of this thesis is to tackle the data availability problem in order to facilitate future research.

Furthermore, seismology is experiencing a scarcity of deep learning platforms. Having only one open-source Python toolbox, i.e., SeisBench, is far from sufficient (Strodthoff et al., 2020). If we look at fields where deep learning has advanced rapidly, they all have numerous mature deep learning platforms that make it easier for developers and more efficient to develop new models (Michael Paluszek, 2020). For example, platforms such as Hugging Face are used for image classification, object detection, language modeling, and more (Wolf et al., 2019). Unfortunately, these mature platforms do not support seismic data types. Therefore, another main objective of this thesis is to fill this gap by developing interactive deep learning platform for seismological community. This should accelerate the development of deep learning techniques in seismology and promote the practical application of deep learning technologies.

In addition, in recent years, an increasing number of deep learning models have been proposed to detect earthquake events. These models often perform well on benchmark datasets (Mousavi et al., 2019b; Zhu and Beroza, 2019; Xiao et al., 2021). However, in practical applications, they often fail to achieve the predictive accuracy observed on benchmark datasets (Woollam et al., 2022). This discrepancy is typically due to differences in the data distribution between the training datasets and the actual data. My final thesis objective is to discuss how many waveforms should be included in a new deep learning project and which additional factors (e.g., data preprocessing and standardization, picking method, tectonic setting, etc.) might affect the training and performance of the model. Through the study of these issues, the goal is to provide a guide for determining the optimal size of the training dataset and model selection for future studies.

1.6 Summary of Thesis Chapters

This thesis is divided into three main research chapters. The first research chapter aims to accelerate the data preparation phase of deep learning in seismology by focusing on the development and application of an open source toolbox for fast seismic dataset creation and annotation for AI applications called QuakeLabeler. This tool differs from published benchmark datasets in that it is designed to customize, build, and manage seismic training

datasets, including processing and visualization. The functionality of QuakeLabeler, which includes retrieving seismograms from multiple online data centers, querying online human-reviewed catalogs, signal processing, data augmentation, annotation, and data distribution analysis, is thoroughly explored. This component of the thesis has been published as an article in the journal *Seismological Research Letters*.

In the second research chapter, I developed a new Python platform called: Blockly Earthquake Transformer (BET), a deep learning platform designed for efficient adaptation of deep learning phase pickers. BET implements the Earthquake Transformer (EqT) as its base model and provides transfer learning and fine tuning extensions. This no-code platform helps researchers design EqT-like model frameworks. BET's interactive dashboard, which allows model customization based on a specific dataset, and the transfer learning module, which extends the application of a deep learning P and S phase picker to more specific phases, are discussed in detail. This component of the thesis has been published as an article in the journal *Artificial Intelligence in Geosciences*.

The third and final research chapter addresses how the architecture of seismic datasets affects the performance of deep learning models for automated earthquake detection. This chapter explores the issue of deep learning model performance by investigating the effect of increasing sample size and examining different deployment settings applied to new data. The insights gained from this study provide a guide for determining the optimal size of the training dataset and model selection for future studies. This component of the thesis is under internal review and will be submitted as an article to the *Journal of Geophysical Research: Solid Earth*.

In summary, this thesis provides a comprehensive exploration of the application of deep learning in seismology, from the creation and management of seismic datasets to the customization of phase pickers and the understanding of how seismic datasets influence the performance of deep learning models.

Chapter 2

QuakeLabeler: A Fast Seismic Dataset Creation and Annotation Toolbox for AI Applications.

2.1 Abstract

The production and preparation of datasets are essential steps in Machine Learning (ML) applications. With the increasing volume and scale of available ML techniques in seismology, annotating seismograms or seismic features has become time consuming and tedious for many researchers. Furthermore, most methods train and validate on unique data subsets, which hampers independent performance evaluation and comparison. To solve this problem, we develop an open-source Python package called QuakeLabeler to customize, build and manage seismic training datasets, including processing and visualization. QuakeLabeler has tight pipeline functions, which include retrieving seismograms from multiple online data centers, querying online human-reviewed catalogues, signal processing, data augmentating, annotating (i.e., making samples) and analyzing data distribution. In addition, relevant statistical graphs and human-readable files can be generated. Various file export formats are supported, such as Seismic Analysis Code (*.sac), mini Standard for Exchange of Earthquake Data (*.mseed), and NumPy (*.npz). This toolbox is packaged with an interactive command-line interface. Three alternative running modes (beginner, advanced and benchmark) are implemented, intended to offer specific dataset solutions for different types of applications, i.e., quick-start-recipes for simple ML solutions, advanced design for customized project training and benchmark bulletins for model comparison.

2.2 Introduction

Artificial intelligence (AI), machine learning (ML) and deep learning (DL) approaches are now well established in many research fields, including seismology (Voulodimos et al., 2018; Zhao et al., 2019; Kong et al., 2019; Fawaz et al., 2019; Tajbakhsh et al., 2020). Datasets, the primary ingredients of any AI/ML algorithm, are the only source of information for AI algorithms to unlock the concealed information or recognize complex patterns (Dahlquist et al., 2021). Part of making the decision of whether a proposed AI/ML model is right for a project comes down to the type, amount and reliability of labeled data. Unfortunately, having access to a suitable labeled dataset in a given research field and/or region can be difficult. Many research and industrial projects do not publicly release their database, which hampers reproducibility and limits progress in AI/ML research (Sarkar et al., 2020).

Collecting and preparing a large enough, high-quality dataset is the first and foremost step in any AI/ML project; however, creating a useful dataset is never simple, even for data specialists. A general step to building a dataset includes data collection, pre-processing, data augmentation, labeling (annotation), data formatting, creation of data subsets (training, test, validation sets), analysis of data distribution, etc. (Géron, 2019). Popular benchmark datasets provide well-calibrated, high-quality and reliably labeled data, which have largely facilitated research in many of the most cutting-edge AI fields. For instance, ImageNet provides a large-scale hierarchical image classification dataset, which has helped thousands of AI projects in computer vision to train and evaluate their AI/ML models using the same standard inputs (Deng et al., 2009). Amazon reviews (Mudambi and Schuff, 2010), a vast text dataset from Amazon containing over 45 million Amazon reviews, fuels AI approaches such as automated recommendation systems, leading to progress in natural language processing (Torfi et al., 2020).

In seismology, a few solutions are available to build datasets for AI/ML use. The first solution is a "do-it-yourself" dataset creation strategy (Vandeput, 2021). Like many engineering fields, most seismological problems are unique enough that augmenting or extending an existing database is impossible. Different researchers have specific expertise and focus on different research regions. Therefore, most AI/ML projects in seismology are based on custom-design dataset schemes. Custom design of the dataset ensures the data format is 100% suitable for the particular AI model of interest. One of the drawbacks is that this approach is oftentimes a trial-and-error process, and is extremely cumbersome to recreate for other users. Furthermore, it is difficult to evaluate whether the dataset covers the full scope of the solution space, or if there are sufficient or accurate enough labeled data for the resulting model to recognize all prospected patterns. Another promising solution is to use published, well-calibrated global seismic datasets. These large-scale,

human-reviewed datasets contain high volumes of labeled data from seismic data archives and earthquake catalogues, including hypocenter location, magnitude, arrival times, etc. The most widely-used open-source seismic dataset for AI use is the STanford EArthquake Dataset (STEAD) (Mousavi et al., 2019a). STEAD provides ˜1.2 million 3-component local earthquake and seismic noise waveforms from all around the world. Fabrizio et al. offer another similar 3-component local earthquake benchmark dataset (Magrini et al., 2020), which also provides global scale, 3-component earthquake and noise waveforms for AI use.

Several advances in the application of ML techniques in seismology are based on these benchmark datasets. For instance, newly proposed earthquake detection (Mousavi et al., 2020), location (Mousavi and Beroza, 2019; Ristea and Radoi, 2021), and magnitude estimation (Mousavi and Beroza, 2020), synthetic seismogram generation(Gatti and Clouteau, 2020), and benchmark performance evaluation (Soto and Schurr, 2021; Liao et al., 2021) algorithms use STEAD. However, apart from STEAD, other mature datasets for AI use are seldom available publicly and are mainly for internal use in commercial companies and research institutes. It is worth mentioning that some active online data science communities also provide seismic datasets, for example the Kaggle Dataset (`www.kaggle.com/datasets`), a subsidiary of Google LLC. Currently, Kaggle has 120 available datasets relevant to earthquake research, including datasets from official publishers such as US Geological Survey (USGS) and Los Alamos National Lab (LANL). However, these datasets only provide event catalogues; none of the datasets directly offer seismograms, which is a significant limitation.

The seismological community has access to very large volumes of raw seismic data published by various online archiving centers, e.g. Incorporated Research Institutions for Seismology (IRIS), US Geological Survey (USGS), etc. However, transforming raw seismic data to labeled datasets is the greatest challenge for the fast deployment of AI datasets in seismology. The manual labeling of seismic data is labour-intensive, time-consuming, and can be inaccurate or subjective. In order to facilitate reproducible and easily extendable AI/ML research in seismology, it is becoming urgent to develop a user-friendly dataset production tool. In other research fields, applying an automated labeling approach is a popular way to save researchers and developers countless hours. Mature auto-labeling applications are provided by various commercial companies such as Amazon SageMaker Ground Truth (AWS, 2021), MATLAB Ground Truth Labeler (MATLAB, 2021a), LabelBox (Labelbox, 2021), etc. These automated annotation tools mainly serve in image segmentation and classification, object detection and self-driving vehicles. Unfortunately, automated labeling for AI/ML datasets has not yet been implemented in seismology.

This article introduces QuakeLabeler, an open-source Python package that enables

users to customize and build seismic datasets at any scale for AI/ML applications. QuakeLabeler was born from the need for seismologists and developers who are not AI specialists to easily, quickly, and independently build and visualize their training dataset. Current functionalities include retrieving seismograms from multiple online data centers, querying online human-reviewed catalogues, performing signal processing, augmentating datasets, annotating (i.e., making samples), analyzing data distribution, etc. The toolbox helps all levels of AI developers and researchers build their own earthquake datasets. QuakeLabeler runs on an interactive command-line interface. Users are able to design datasets with little knowledge of programming and data querying. By design, QuakeLabeler enhances reproducibility in AI seismic research.

2.3 Implementation

QuakeLabeler is completely written in Python. Over the years, Python has become one of the most popular open-source programming languages used for machine learning, especially deep learning. The vast majority of seismologists contribute their AI applications in the Python ecosystem. These seismic AI applications, combined with other versatile and powerful scientific computing and deep learning packages in the Python community, facilitate new geophysical workflows and applications. QuakeLabeler also benefits from several mature Python packages. These packages and their functionality for QuakeLabeler are described in Table 2.1. Currently, QuakeLabeler offers promising possibilities for other Python projects, easily bridging seismic data to AI applications.

Table 2.1: Package dependencies

Package Name	Usage in QuakeLabeler
ObsPy	Support for I/O, fetching data and signal processing.
Request	Query human-reviewed catalogue from ISC.
PyGMT	Make publication quality maps and figures.
Termplotlib and Matplotlib	Visualize data distributions.

Essentially, QuakeLabeler conducts a tight pipeline of functions to help users convert raw seismic data into valuable training datasets for machine learning through professional collection and annotation techniques. In addition, it also includes several utilities that output necessary analytical figures, maps and meta data, respectively. Figure 2.3 introduces the workflow of QuakeLabeler. By design, QuakeLabeler immensely reduces manual operations. Once the user inputs the specific dataset structure and processing parameters

via command-line interactive tools, QuakeLabeler builds the requested seismic datasets without the need for human interaction. This aspect is particularly useful for researchers with little to no prior experience in Python.

2.3.1 Running Modes

QuakeLabeler provides three running modes targeted for different types of users and research goals (Tab. 2.2). The features of the different running modes and their relevant applications are briefly described below.

Table 2.2: Running modes

Mode Name	Descriptions
Beginner	Pre-defined datasets at different scales: small, medium, large.
Advanced	Custom search region, time period, dataset scale, pre-processing steps, sample format, etc.
Benchmark	Standard seismic datasets for different data quality.

Beginner mode

The ***Beginner mode*** is aimed at AI/ML novices to quickly start developing their models. In many scenarios, researchers or AI learners only wish to extract small samples to develop and test their models. However, the original training datasets for their application are often unavailable (i.e., not shared openly) or too large to download and prepare in a short time. Furthermore, users may wish to apply published AI methods to their own specific research area, which might not be covered by the original datasets that were used to train those AI models. QuakeLabeler's quick-start recipes offer solutions for these shortcomings, which list specific dataset structures of the current popular open-source AI/ML models on GitHub (Fig.2.1). Following these recipes, users can rapidly produce samples for AI practice at different scales (small, medium or large dataset sizes). We also invite developers to contribute their dataset design ideas to enrich QuakeLabeler's recipes.

Advanced mode

The ***Advanced mode*** is suitable for experienced seismologists and data scientists interested in designing a specific structure for their dataset. In this mode, QuakeLabeler

Figure 2.1: **Screen capture of quick-start-recipes menu.** This option is offered in the Beginner running mode, to rapidly deploy popular dataset structures.

initiates an interactive ***Custom Design Dialog (CDD)*** interface. The CDD consists of a list of specifications for the dataset, with pre-set values available by default. The CDD makes QuakeLabeler more versatile than static, pre-bundled benchmark datasets by providing the end user with the control in creating unique datasets in a reproducible manner. For instance, seismograms in STEAD have a fixed sample size defined by 1 minute-long, 3-component time series with a sampling rate of 100 Hz (i.e., 6000 points in each sample). In QuakeLabeler, these parameters are selected by the user, thus allowing flexibility for producing seismic labels, and creating datasets that can be adapted to any specific AI/ML application.

The CDD module is divided into two steps. In each step, the CDD provides a brief introduction and specific usage as well as default parameters. In the first step, the CDD prompts the user to select their geographic region and time range of interest (Tab.2.3) with default values displayed as templates. For instance, the default search region options is to select a rectangular-shaped area with exact longitude and latitude values, i.e., stn_bot_lat = 40.0, stn_top_lat=55.0, stn_left_lon=-130.0, stn_right_lon=-120.0 (Fig.2.2). Selecting the time range for the dataset requires the user to type in a pair of start time and end time. For instance, startyear=2010, startmonth=01, startday=01, starttime=00:00:00.

Figure 2.2: **Screen capture of custom design dialog.** This option is offered in the Advanced running mode, for users with specific needs for unique datasets.

Table 2.3: Custom design dialog #1, which specifies the source region and time range of interest.

Custom	Options (User design parameters)	Default
Region	STN, GLOBAL, RECT, CIRC, FE, POLY	RECT
Time	Start time, End time	00:00:00 - 23:59:59
Date	Start date, End date	2010-01-01- 2010-12-31

The second step of the CDD controls every processing detail within the dataset. Multiple options are provided to let the users decide their preference for the production of the dataset. This step includes three sub-steps related to specifying the size of the dataset, the waveform processing options, and the output formats, respectively (Tab.2.4). A detailed description of this step is presented in the following section. We note that, by default, QuakeLabeler will not perform any signal processing and maintain raw data through the workflow. Finally, QuakeLabeler creates and saves a log file of all user inputs through the workflow, such that other users can reproduce a particular dataset exactly. This feature will promote fast development and testing of AI/ML models through enhanced reproducibility of the datasets.

Table 2.4: Custom design dialog #2, which defines the structure of the dataset. For more details, please consult QuakeLabler's online documentation.

Sub-section	Options (User design parameters)	Default
Dataset Size	Sample amount, sample length	10,000 ; 5,000 points
Waveform format	filter, detrend, denoise, re-sampling	None
Export file format	SAC, MSEED, NPY, MAT	SAC

Benchmark mode

The ***Benchmark mode*** generates datasets similar to STEAD in order to be easily used in model comparison. In this mode (Fig.2.11), users can select alternative built-in benchmark bulletins to rapidly deploy well-organized datasets. All relevant information, graphs, and documents are generated simultaneously. QuakeLabeler aims to publish as many benchmark datasets as possible to provide one-stop data solutions for many seismological applications. The current version focuses on natural earthquake events at different spatial scales and geographic regions. We invite contributions for datasets of other types of seismicity scenarios, e.g. induced earthquakes, volcano seismicity and glacial quakes.

2.4 Workflow

After selecting the running mode and specifying the various options, QuakeLabeler's workflow (Fig.2.3) includes three steps: 1) Data collection; 2) Signal processing and annotation; and 3) Export and visualization.

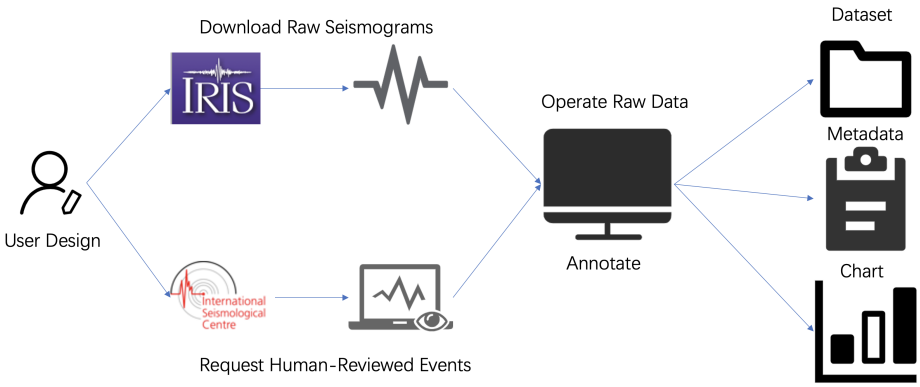


Figure 2.3: **QuakeLabeler's workflow, outlining the dataset creation process.**

2.4.1 Data Collection

QuakeLabeler does not provide any pre-bundled, static dataset (in contrast to STEAD, for instance). Instead, all available data are requested from the International Federation of Digital Seismograph Networks (FDSN). Data collection includes two parts: querying human-reviewed catalogues and requesting raw seismogram data. In each part, knowledge of the seismic station distribution within the region of interest is required. Obtaining accurate and reliable earthquake catalogues with complete meta information (i.e., hypocenter, magnitude, arrival time picks for several seismic phases) is a crucial part of the labeling process. To this end, QuakeLabeler retrieves human-reviewed catalogues from the International Seismological Centre (ISC) web service and extracts paired event-station information. This type of data is also sometimes referred to as *features* in some AI/ML applications (Jiao and Alavi, 2020; Saad et al., 2020; Matin and Pradhan, 2021). In this paper, we will refer to the paired event-station information as meta data, which gets stored in a Pandas DataFrame object in QuakeLabeler.

Following this step, QuakeLabeler collects raw seismograms from specific FDSN clients based on the meta data. Fetching raw seismograms is the most time-consuming step in QuakeLabeler, and highly depends on the user's connectivity and the amount of requested data. By design, QuakeLabeler can filter out bad data traces (e.g., seismograms containing data gaps) and manage valid raw data to a local drive. Each available event-based seismogram is also accompanied by a "non-signal" noise waveform to provide a complete

and balanced dataset. Noise waveforms are collected from pre-event time window (60 minutes before an event) in the catalogue. Noise waveforms are processed similarly to the earthquake seismograms and are tagged as "noise" in the following annotation procedure.

2.4.2 Signal Processing and Annotation

Raw seismograms are then manipulated to produce ground-truth labels using a series of signal processing steps. This is the most complex and tedious part of data preparation in any AI/ML study. QuakeLabeler can help researchers navigate this step by providing several signal processing tools. Most of these functions are implemented in ObsPy, which is the standard Python package for processing seismic data (Beyreuther et al., 2010).

Here we recommend 2 sets of signal processing and annotation workflows that are most popular in AI seismic projects. Different workflows will impact training performance, therefore we encourage users to generate different types of datasets and compare their effects on their AI/ML models. Repetitive trials can help converge to the optimal training dataset, which should cover enough waveform patterns for any specific project.

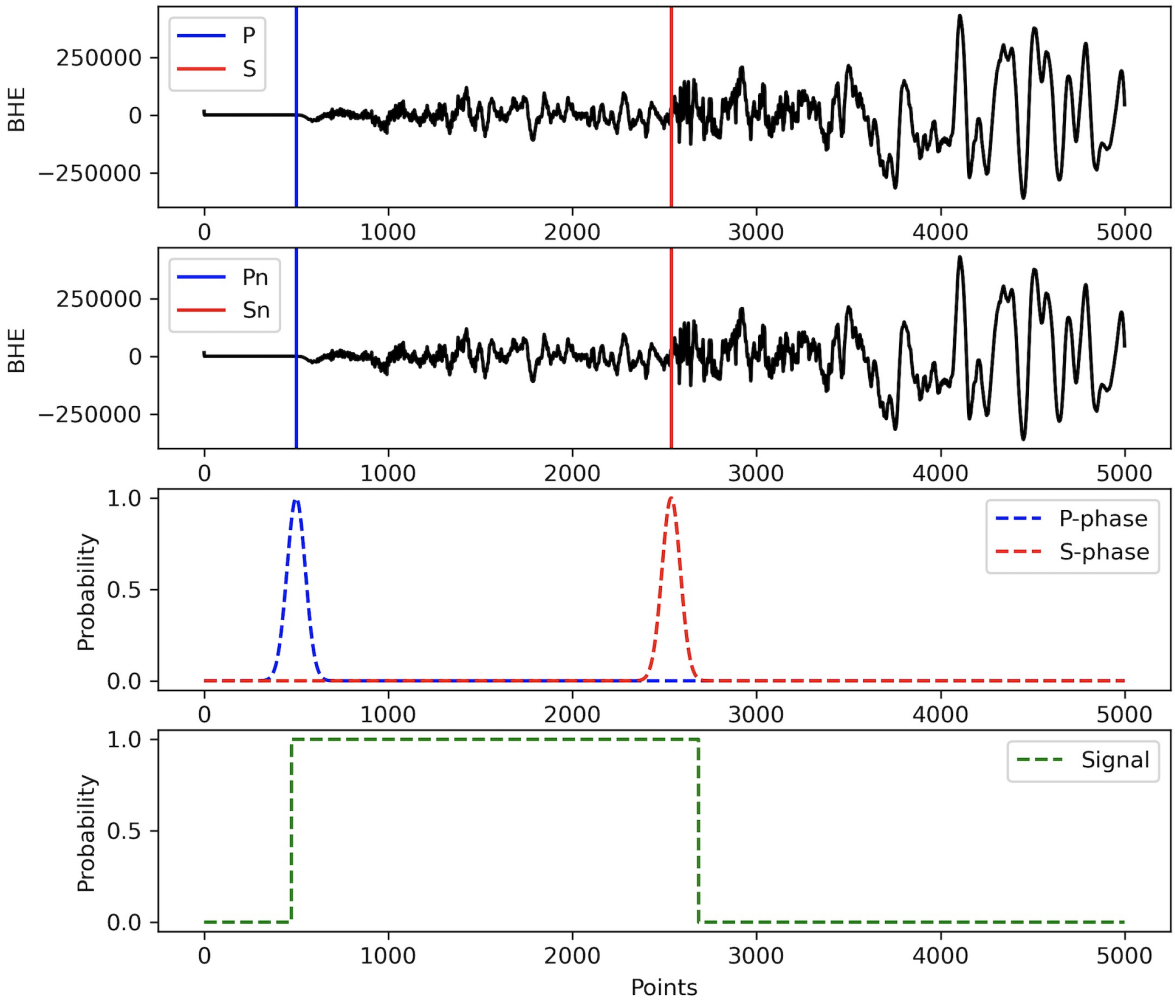


Figure 2.4: **Schematic of annotation options.** From top to bottom: 1) simple phase labels for classification; 2) specific phase labels for classification; 3) prediction probabilities for phase picking; and 4) prediction probabilities for signal detection.

The first workflow performs only basic manipulations to the raw waveforms. By default, we use QuakeLabeler to randomly crop the waveforms near each arrival time of interest (e.g., P&S phases) and cut the trace to a fixed length (e.g., 5000 points). This random cropping step helps reduce over-fitting. Fixing a uniform sample length for all waveforms is the easiest input/output (I/O) design for most AI models. We do not remove the mean and trend or apply any filter, which would inevitably change the original waveforms. Many

supporters for this workflow believe that AI/ML models should be training based on the original data as much as possible (Chu et al., 2016; Neutatz et al., 2021). However, others claim it is important to clean the data before training models (Li et al., 2019; Brownlee, 2020). Therefore, our second recommended workflow is to manipulate the raw seismograms in a similar way to what is done by analysts in routine processing (Havskov and Ottemoller, 2010), which includes demeaning, detrending, resampling, band-pass filtering, etc.

Annotation can take many forms in QuakeLabeler. To build labeled datasets for classification and phase picking projects, QuakeLabeler supports a simple labeling mode to annotate each waveform with P-phase, S-phase and Noise tags. Alternatively, QuakeLabeler can annotate by specific phase types such as PcP, ScS, Pg, Sn, etc., thanks to the ISC arrival bulletin's human-reviewed catalogues. Furthermore, QuakeLabeler also implements annotations as probability distribution functions, which are often used in seismic ML projects as output channels (Zhu and Beroza, 2019; Mousavi et al., 2020), where a Gaussian (normal) distribution represents the possibility of arrival. A Rectangular distribution stands for a positive seismic signal (Fig. 2.4). As for other ML tasks such as hypocenter and/or magnitude estimation, QuakeLabeler annotates the waveforms using the paired event-station meta data (Tab. 2.5).

Table 2.5: Example paired station and event information that is provided as meta data in QuakeLabeler.

EVENTID:	14223208,
STA:	I04A,
ARRIVAL_LAT:	43.7941,
ARRIVAL_LON:	-122.4113,
ARRIVAL_ELEV:	731.0,
ARRIVAL_DIST:	3.51,
ARRIVAL_BAZ:	24.9,
ORIGIN_LAT:	40.6274,
ORIGIN_LON:	-124.4529,
ORIGINL_DEPTH:	21.6,
ORIGIN_DATE:	2010-01-10,
ORIGIN_TIME:	00:27:42.10,
EVENT_TYPE:	MS,
EVENT_MAG:	6.3

2.4.3 Export and Visualization

QuakeLabeler supports various export file formats for the labeled waveforms, such as Seismic Analysis Code (*.sac), mini Standard for Exchange of Earthquake Data (*.mseed), NumPy (*.npz) and MATLAB (*.mat). The labeled data are automatically written into the target folder in a local directory. Data are randomly assigned into 2 different sub-folders as a training sub-set and a validation sub-set. By default, the training dataset includes 80% of the data for model training; the validation dataset takes the rest of the samples to test the model's performance. For each sub-folder, the labels are completely independent and none of the training samples can leak to the validation sub-set.

The most significant advantage of the export module is that QuakeLabeler can create flexible types of annotated waveforms (Fig.2.5). For example, running a quick-start-recipe in beginner mode, EQTransformer (Mousavi et al., 2020) and PhaseNet's(Zhu and Beroza, 2019) original waveform format can be applied. In this way, the generated dataset can be directly used as input into AI training or validation without editing the structure and format of the dataset. Moreover, users are allowed to fully customize waveform annotation in the advanced mode. Flexibility in export options allows QuakeLabeler to serve various processing methods and codes.

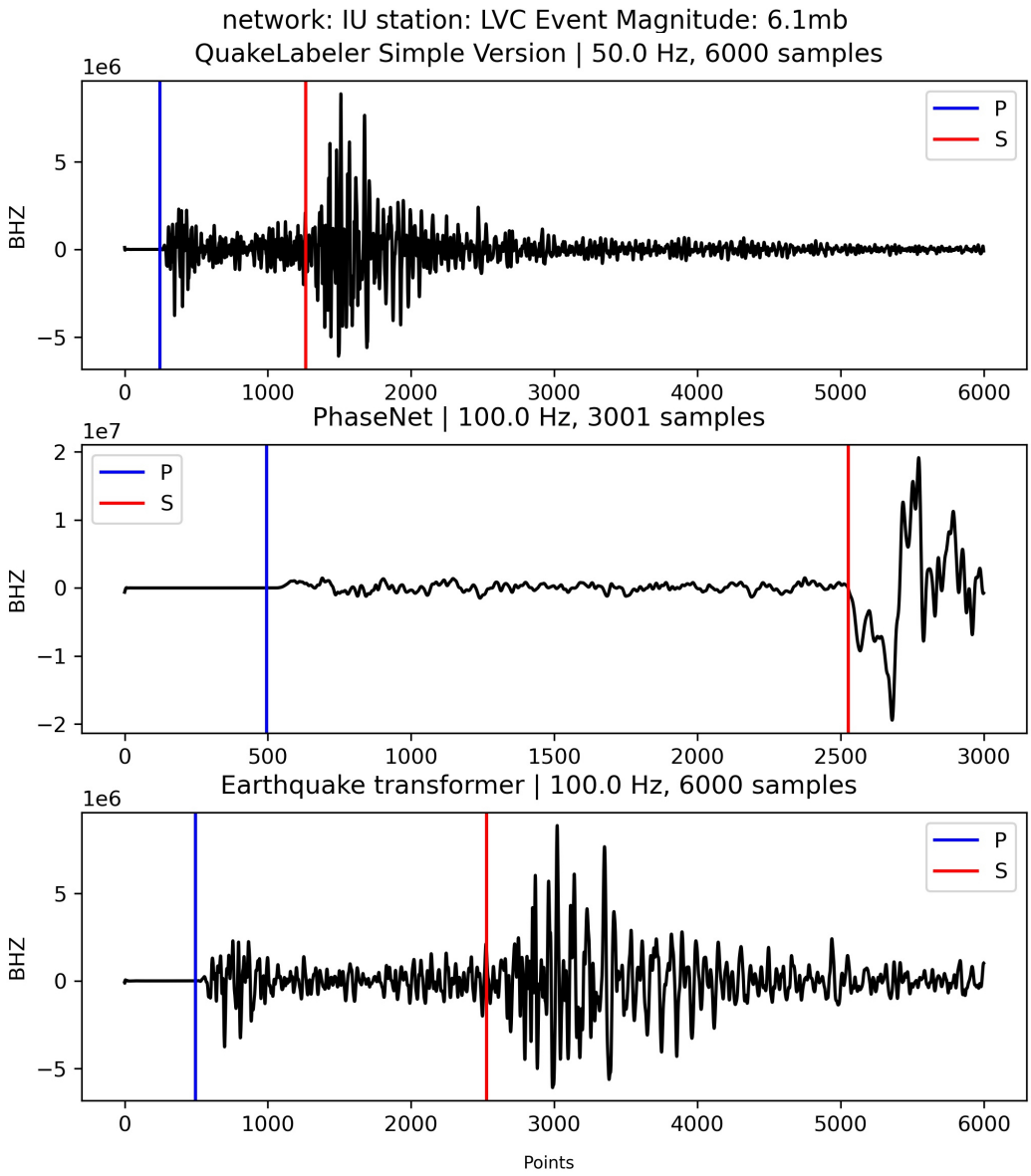


Figure 2.5: **Examples of annotated waveforms.** The workflow for EQTransformer format applies a band-pass filter from 1-45 Hz; the other workflows have minimal pre-processing. These format options can be directly applied in the Beginner Mode.

In the final stage, QuakeLabeler exports several relevant graphs and human-readable documents to help users understand and visualize the properties of their dataset using

charts, graphs, maps and meta data. Moreover, the meta data will be saved as comma-separated-value (.csv) files for further external processing. In QuakeLabeler, users can examine the magnitude distribution from a histogram (Fig. 2.7); observe station locations (Fig. 2.8) and events (Fig. 2.9) on maps; and check specific samples of the dataset (Fig. 2.6). These extensions provide ways to visualize and understand trends, outliers, and patterns in the dataset.

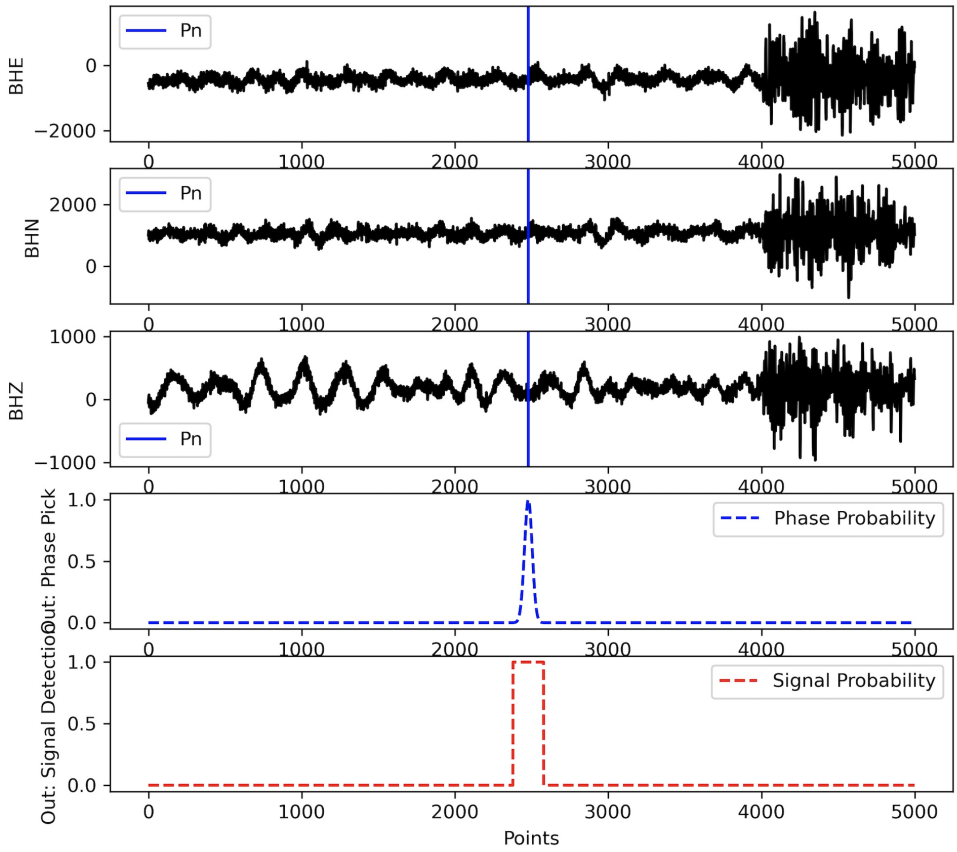


Figure 2.6: **Example of ground-truth labels for of training sample for Pn-wave arrival.** Top 3 sub-figures are 3-component seismograms, bottom 2 sub-figures are output probability distribution for phase picking and signal detection, respectively.

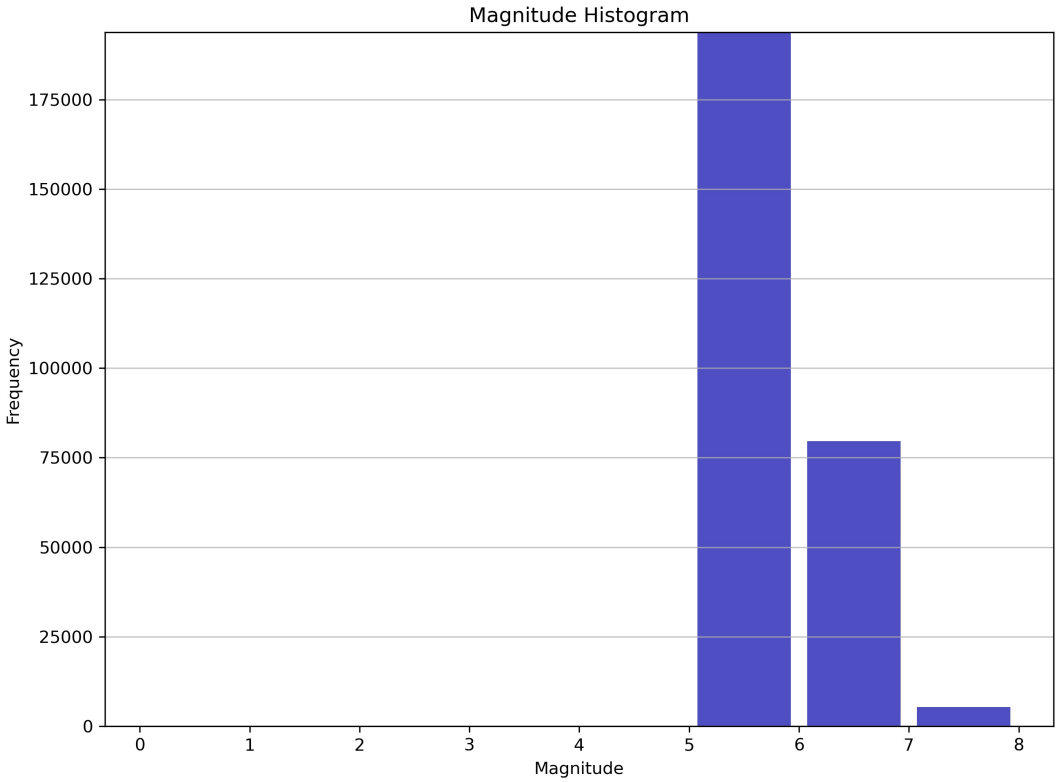


Figure 2.7: **Example of magnitude distribution of global earthquakes $M > 5$ that occurred in 2018, from one of the available benchmark datasets.**

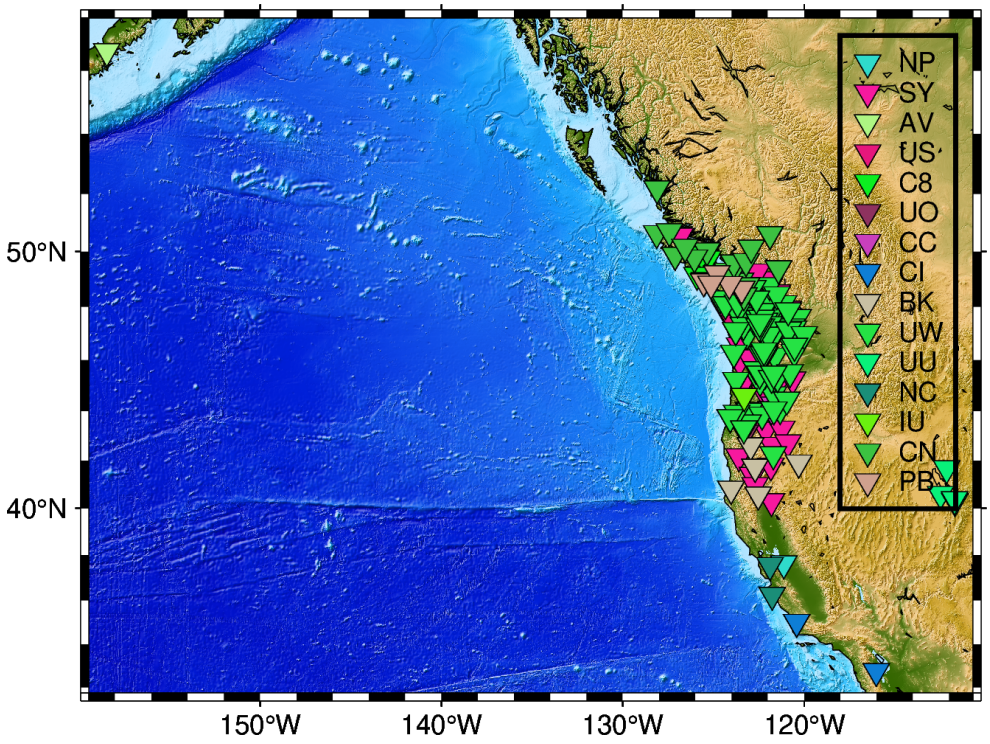


Figure 2.8: **Example of station distribution (inverted triangles) in the Pacific Northwest region of North America, taken from the Cascadia benchmark dataset.**

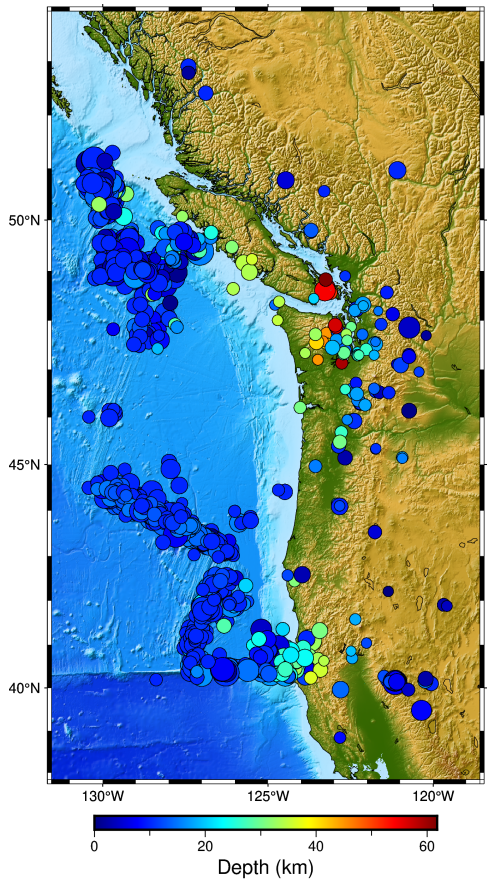


Figure 2.9: **Example of earthquake distribution extracted from the Cascadia benchmark dataset. Magnitudes scales with symbol size.**

2.5 Prospective Applications

As mentioned above, QuakeLabeler does not provide a static benchmark dataset but is instead a dataset creation tool that offers a large range of prospective applications. Many seismological tasks can be connected to QuakeLabeler dataset service, i.e., denoising, event detection, phase picking, magnitude estimation, hypocenter determination, etc. Here we provide examples where QuakeLabeler can help researchers develop their AI models.

2.5.1 Phase Picking

Here we present a case where QuakeLabeler is used to create a dataset for training a phase picking AI model to improve the detectability of small earthquakes. We use QuakeLabeler's advanced running mode to design our training dataset for the Cascadia subduction zone, which is characterized by a wide variability of earthquake focal depths. We select a ten-year period (2008-2018) as our time window of interest and a rectangular research area (W110°0′, W140°0′, N30°0′, N60°0′). All types of ambiguous phase picking cases are covered in this dataset due to the variability in earthquake types, source environments and onshore-offshore observational bias. For instance, long-duration P wave coda and the long time lag between P and S phases that occur in this data set often hamper a robust determination of S-wave picks. We take the lightest signal processing approach, which only applies random arrival labels, and crop samples at the same length. This sample form is similar to cropped raw seismograms containing weak seismic signals that are difficult or impossible to identify visually. The addition of small events allows our training procedure to self-learn unseen signal features from natural intricate seismic patterns.

A U-Net model is applied to this dataset. The U-Net creates a mapping from five input channels (three-component seismograms plus two filtered traces) to three-channel probability functions describing the current data point as a P arrival, S arrival, or non-arrival (Mai and Audet, 2020). We compare our proposed model with PhaseNet and AIC picker from Obspy (Krischer et al., 2015). The results show that our model is highly robust to small magnitude, complex source wavelet patterns, and low signal-to-noise ratio cases (Fig.2.10). Our prediction results match those in calibrated earthquake catalogues and demonstrate their high precision and generalization performance.

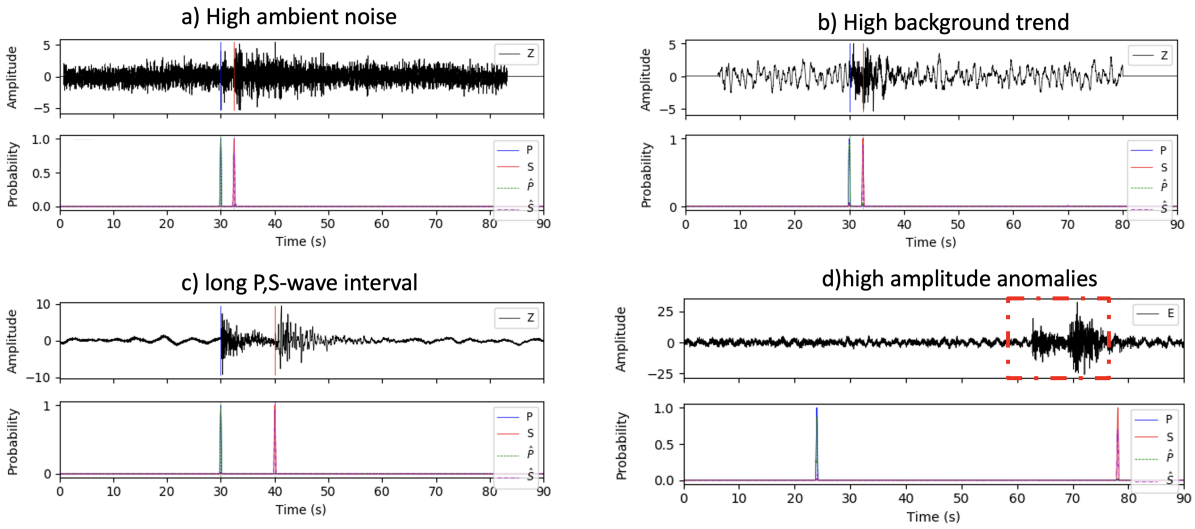


Figure 2.10: **Examples of automated arrival time picking using training data from the Cascadia dataset available in the benchmark mode.** These examples highlight the variable signal environments in the training data for small-magnitude earthquakes.

2.5.2 Model Comparison

Comparing different seismic AI models in QuakeLabeler's benchmark mode can be more straightforward and objective than previously published static benchmark datasets. With the benefit of flexible export options, QuakeLabeler can reproduce training datasets in popular formats used in established AI/ML methods (Mousavi et al., 2019b; Zhu and Beroza, 2019; Mousavi et al., 2020). This provides straightforward insights into which model is best suited for a specific field or application of interest. To help illustrate this, we select the 2010 Cascadia subduction zone seismicity dataset available in the benchmark mode, to test the accuracy of several popular approaches in earthquake detection and phase picking tasks. This dataset includes 9,213 individually (human-)reviewed, body-wave onset times (P- and S-wave arrivals) on a global scale. Over 30,000 annotated traces can be applied for model comparison.

Model comparisons may include, for instance, seismic signal detection and phase picking for P- and S- wave arrivals. To compare signal detection performance, we applied three deep learning models: EQTransformer, PhaseNet, CRED as well as a traditional AIC picker(Mousavi et al., 2019b; Zhu and Beroza, 2019; Mousavi et al., 2020; Akazawa,

2004). For the phase picking task, we test EQTransformer, PhaseNet and AIC in P-phase and S-phase, respectively. We set QuakeLabeler to produce test datasets in each method's unique data format to ensure adequately formatted output (e.g., EQTransformer and PhaseNet dataset formats). CRED's and AIC's sample formats are easily set based on their original paper's descriptions(Mousavi et al., 2019b; Zhu and Beroza, 2019; Mousavi et al., 2020; Akazawa, 2004) using QuakeLabeler's custom waveform options. Within two hours (with a modest connectivity and a desktop computer), the required four datasets are successfully collected. We run the test procedure in each method and generate the following performance summary (Tab. 2.6). In this comparison, EQTransformer (Mousavi et al., 2020) yields the best accuracy in both detection and phase picking tasks, which shows that EQTransformer has better performance in identifying earthquake signals at the Cascadia subduction zone than any other method. Comparison results testify that deep learning models have better detection ability than traditional methods. Although automated S-phase picking remains challenging, these AI-based approaches are worthwhile in massive seismic detection tasks.

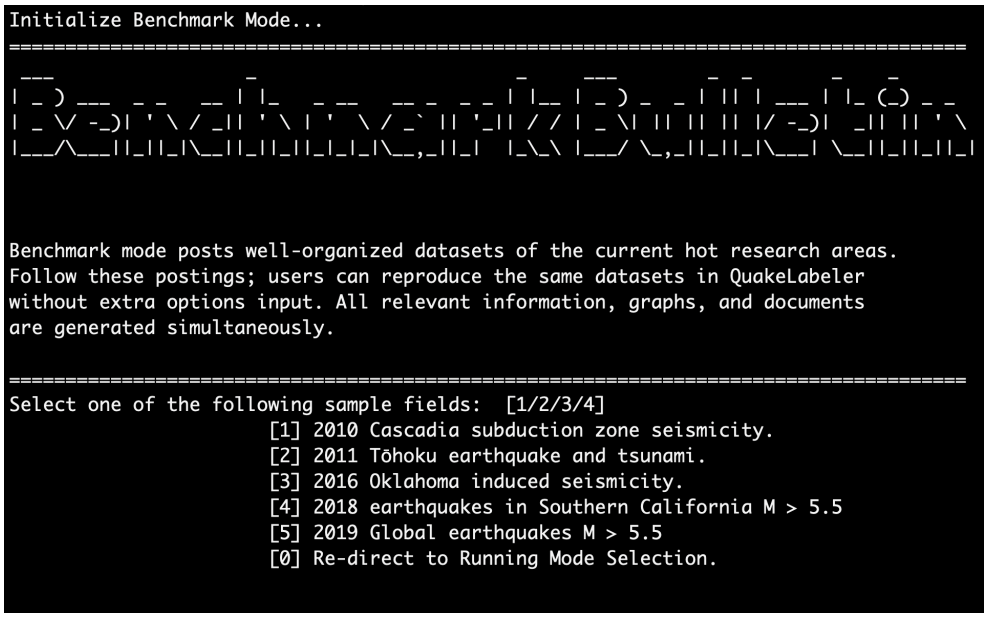


Figure 2.11: **Screen capture of the benchmark mode menu.** Benchmark models help rapidly produce well-organized datasets, and includes various graphs, statistical data, etc.

Table 2.6: Model performance

MODEL	Detection accuracy	P-phase accuracy	S-phase accuracy
EQTransformer	0.98	0.99	0.94
PhaseNet	0.92	0.94	0.90
CRED	0.91	——	——
AIC	0.74	0.77	0.68

2.5.3 QuakeLabeler with Google Colab

One of the limitations of QuakeLabeler is its reliance on good connectivity for querying and downloading large amounts of seismic data. For instance, downloading the previously published benchmark datasets (i.e., large HDF5 format documents) on a local device can be time-consuming and some users' connectivity may be unreliable for this task. To remediate this, QuakeLabeler supports execution through a web browser with the help of Google Colaboratory. Google Colab is a free Jupyter notebook environment running entirely in the cloud (Nelson and Hoover, 2020). Most importantly, Colab supports the most popular machine learning libraries, which can be easily loaded after creating a dataset by QuakeLabeler. The most significant advantage is that the dataset can be saved in Google Drive instead of a local machine, which enhances collaborative access and alleviates connectivity issues.

Running QuakeLabeler in Colab's hosted Jupyter notebook service may also be more stable and efficient. In our stability testing, QuakeLabeler in Colab is faster than QuakeLabeler in the bash version (Tab. 2.7). Moreover, spontaneous breaks in connectivity will not affect the execution of QuakeLabeler when executed in Colab. Furthermore, because computation is done on remote cloud servers, there are no minimum requirements for the local device (Carneiro et al., 2018). For example, running QuakeLabeler on mobile phones and tablets is possible. For further details about QuakeLabeler with Google Colab, please consult the QuakeLabeler introduction Google notebook: https://github.com/maihao14/Earthquake-Arrivals-Dataset-for-AI/blob/main/QuakeLabeler_with_Colab.ipynb

Table 2.7: Average Execution Time Table

Running Environment	Initialization	Global Search	Advanced Search	Total Time
Google Colab	7:00	4:30	7:35	50:45
Python Script	00:20	17:50	25:00	65:20
Interactive user interface	00:25	18:12	27:45	67:25

2.6 Conclusion

QuakeLabeler was born from the need for seismologists and developers who are not AI specialists to easily, quickly, and independently build and visualize reproducible training datasets. The software provides three running modes to reach different types of users and research goals. Users can design their datasets to target different spatial scales of seismicity, i.e., local ($< 400KM$), regional ($400 - 2000KM$) and teleseismic ($> 2000KM$), and magnitude scales from microseismic studies to large fault ruptures. QuakeLabeler can be successfully used in studies of noise attenuation, earthquake detection, phase picking, earthquake location, magnitude estimation, etc. We invite contributions for new benchmarked datasets (e.g., volcano-seismic activity, glacial quakes, induced seismicity). The current version of QuakeLabeler is accessible under https://maihao14.github.io/QuakeLabeler/. A detailed documentation and extensive tutorial is online at https://maihao14.github.io/QuakeLabeler/index.html.

2.7 Data and Resources

Earthquake and noise waveforms are request from the Incorporated Research Institutions for Seismology data management center (IRIS DMC, http://www.iris.washington.edu/ds/nodes/dmc/). The human-reviewed event catalogues are requested from the International Seismological Centre (ISC, https://doi.org/10.31905/D808B830). QuakeLabeler is available as a repository on GitHub (https://maihao14.github.io/QuakeLabeler/).

2.8 Declaration of Competing Interests

The authors acknowledge there are no conflicts of interest recorded.

2.9 Acknowledgments

This research was funded by a Discovery Grant (RGPIN-2018-03752) from the Natural Science and Engineering Research Council of Canada (PA).

Chapter 3

Blockly Earthquake Transformer: A Deep Learning Platform for Custom Phase Picking.

3.1 Abstract

Deep-learning (DL) algorithms are increasingly used for routine seismic data processing tasks, including seismic event detection and phase arrival picking. Despite many examples of the remarkable performance of existing (i.e., pre-trained) deep-learning detector/picker models, there are still some cases where the direct applications of such models do not generalize well. In such cases, substantial effort is required to improve the performance by either developing a new model or fine-tuning an existing one. To address this challenge, we present Blockly Earthquake Transformer(BET), a deep-learning platform for efficient customization of deep-learning phase pickers. BET implements Earthquake Transformer as its baseline model, and offers transfer learning and fine-tuning extensions. BET provides an interactive dashboard to customize a model based on a particular dataset. Once the parameters are specified, BET executes the corresponding phase-picking task without direct user interaction with the base code. Within the transfer-learning module, BET extends the application of a deep-learning P and S phase picker to more specific phases (e.g., Pn, Pg, Sn and Sg phases). In the fine-tuning module, the model performance is enhanced by customizing the model architecture. This no-code platform is designed to quickly deploy reusable workflows, build customized models, visualize training processes, and produce publishable figures in a lightweight, interactive, and open-source Python toolbox. The BET package is

available as a GitHub repository at https://github.com/maihao14/BlocklyEQTransformer.

3.2 Introduction

In the last five years, the global seismological community has witnessed the emergence of deep-learning (DL) as a promising candidate for undertaking large-scale seismic processing and modelling tasks in modern seismology (Mousavi and Beroza, 2022a). Among many other seismological applications, DL algorithms for earthquake signal detection and seismic phase picking have had huge success, thanks to the existence of large-scale and publicly-available archived waveforms and phase labels (Mousavi and Beroza, 2022b). DL models are designed to learn intricate patterns concealed in seismic waveforms, identify seismic signals and their relations with desired targets such as seismic wave arrival times (i.e., P-arrival, S-arrival, etc.) without human intervention. One obvious application of DL phase pickers is in the automation of seismic data processing in near real-time earthquake monitoring workflows that are routinely operated by regional, national and international network operators to improve the accuracy and completeness of earthquake and seismic event catalogues (Yeck et al., 2021). In addition to improving magnitude-frequency distributions, which are critical for seismic hazard assessment and forecasting (Beroza et al., 2021), DL phase pickers may have utility in verification seismology, specifically in the identification of P- and S-phases of very low-yield (e.g., $10^{-4} - 10^{-1}kT$), low-magnitude (e.g., $m_b 1 - 3$) nuclear explosions. Such events are best recorded at local distances ($< 150 - 200$ km) by regional seismic network operators (Koper, 2019). Algorithms based on DL phase pickers like the ones presented here can expedite the creation of reliable earthquake catalogs. Recent reviews of the state-of-the-art applications show that DL-based models outperform classical characteristic function–based models in many scenarios, such as fast detection of P and S arrivals in a global benchmark dataset, and for finding S-waves that are obscured by the coda of earlier phases (Mousavi et al., 2019a; Woollam et al., 2022; Münchmeyer et al., 2022; Ma et al., 2020, 2023).

Although DL applications have shown promising performance and resulted in interesting outcomes, the path to developing more effective DL models is not always clear and efforts have been uneven. The model development procedure is often fraught with difficulties with the consequence that any new DL project requires the investment of massive amounts of labour and research resources. DL-based approaches for seismic data processing face similar issues, but most of the steps have unique prerequisites that cannot be migrated from previous work. Standard DL workflows can be summarized in four steps:

1. Pre-processing: This step prepares the training and validation data, and includes data collection, cleaning, and labeling. Benchmark seismic datasets such as STEAD (Mousavi et al., 2019a), INSTANCE (Michelini et al., 2021), and DiTing (Zhao et al., 2022) provide high-quality labeled data for training and model building. However, the number of these benchmark datasets and their applications is limited. Alternatively, applying a custom seismic sample toolbox such as QuakeLabeler (Mai and Audet, 2022) can generate labeled datasets for training based on users' demands. The quality and distribution of the training data ingested by the DL algorithm have a large impact on the performance of the resulting model. Good training data should have qualities such as relevance, minimal labeling errors, wide distribution representing various real-world classes, few missing or repeated values, etc.

2. Training: In the training step, the pre-processed seismic data are passed to a specific DL architecture to find patterns within the seismic signals and learn to pick seismic phase arrival times. Many DL models have been proposed recently, including U-Net models such as BasicPhaseAE and PhaseNet (Woollam et al., 2019; Zhu and Beroza, 2019). Convolutional and Recurrent Neural Network (CNN and RNN, respectively) earthquake event detectors include CRED and DeepPhasePick (Mousavi et al., 2019b; Soto and Schurr, 2021), as well as more complex architectures like Earthquake Transformer (EqT) that adds self-attention layers (Mousavi et al., 2020).

3. Validation: Before building a final phase picker model, the validation process allows testing the model against data that have not been used for training. In general, 20% of the entire dataset are retained and excluded from the training dataset. The performance of the model on this validation set represents how likely the model is to succeed in correctly predicting phases in the face of independent seismic data. A trained model could either be approved (i.e., deemed satisfactory), tuned, or rejected based on the validation results (Livieris et al., 2020). The validation stage should not be time-intensive; however, if the model is rejected, the project returns to the training stage to revise the architecture and parameters. This training and validation loop continues until a model is approved.

4. Deployment: The approved DL picker moves on to the model development stage and is then applied to different target regions/datasets, or at various scales ranging from local and regional scales to global scales, depending on the prescribed architecture. It is common that a performing DL model fails to yield satisfactory results using a new benchmark dataset or a new target region (e.g., Woollam et al., 2022), meaning the model has weak generalization. This may be due to different data distributions between the training dataset and the data collected from the regions of interest; DL

models cannot guarantee performance if the new dataset is not statistically similar to the training dataset. Such generalization issues almost always arise, and there is always a limit for the performance of a DL model (Münchmeyer et al., 2022).

To shorten the development cycle of DL phase pickers, transfer learning (TL) has been shown to be an effective solution for efficient reproducibility. TL builds new DL picker models from pre-trained models using partially or fully pre-trained hyper-parameters (weights) and network architectures to reduce the training time. This is a rapidly growing field in DL research, which reduces the massive computation requirements and time resources, and improves upon data-deficient tasks. A few early attempts have proven the success of TL, especially benefiting research projects with too few data to train a full-scale model (Lapins et al., 2021; Zhu et al., 2022a). For instance, TL was utilized in Italy to enhance the earthquake ground shaking predictions using only a limited amount of training events (a few hundreds) (Jozinović et al., 2022). Another example is the Phase-Net model that was re-trained using TL to augment the accuracy of mesoscale monitoring systems in seismic phase picking (Chai et al., 2020). Nevertheless, in seismic phase picking applications, TL only addresses specific situations, such as extending the model's application to other regions or seismic phase types (Zhu et al., 2022a). On the other hand, if a need arises to re-train and deploy new models for specific data/regions, users need to consider all steps of the model-building cycle and their technical requirements, as outlined above. Although TL is an effective and flexible approach, its implementation is often not straightforward. Seismologists who are not familiar with neural networks and/or common programming languages used in machine learning (e.g. Python) often find it time-consuming and cumbersome to apply TL.

In other research fields that face similar challenges, the solution is often to use professional interactive platforms to design, train, and develop industry-level AI applications (Huang et al., 2017; Gibson et al., 2018; Ma et al., 2019). Interactive DL platforms accelerate related research, and reduce the number of repeated steps and duplicated efforts among different research groups. For instance, in natural language processing, notable projects include NVIDIA Transfer Learning Toolkit (Gopalakrishnan et al., 2017) and EasyTransfer (Wang et al., 2019a). These platforms allow users to load existing models from built-in libraries, configure training settings, and perform training, validation and deployment with minimal programming operations. To the best of our knowledge, no such platform exists in the seismological community. The only python toolkit with similar functions is Seisbench, which implements previously published models that can be loaded to reproduce the built-in models and datasets (Woollam et al., 2022). However, if users want to use Seisbench to create new models or apply them to new data, code rewriting becomes an unavoidable

part of the process; i.e., users need to learn the Seisbench API to write custom codes with similar PyTorch syntax as Seisbench's examples (Woollam et al., 2022).

To address these challenges and accelerate the adoption and development of DL pickers in seismology, we propose an easy-to-use, no-code, interactive platform – the Blockly Earthquake Transformer (BET) – for customizing DL pickers. BET uses Earthquake Transformer (Mousavi et al., 2020) as its base model. It provides a user-friendly interface to create new models and perform TF and fine-tuning using an interactive approach. BET's high flexibility and simplicity make it possible for users with a wide range of background knowledge and expertise to design and adjust DL models without in-depth DL knowledge. Blockly Earthquake Transformer is open source and is released under the MIT License. In the following section, we present the BET infrastructure and its possible configurations. Next, we provide three demos demonstrating BET's efficiency, effectiveness, and scalability. Finally, some potential applications of BET are discussed.

3.3 Methods

BET is entirely written in Python, combining Jupyter interactive widgets and Voilà to render a stand-alone web application (Silaparasetty, 2020). The graphical interface of BET is shown in Figure 3.1. The BET dashboard allows users to design, analyze, and deploy industry-level phase-picking projects with little to no prior experience in Python or machine learning. For advanced users, built-in models and functions are easy to extend or rewrite through revisable APIs that can be found in the underlying Jupyter notebook.

Essentially, BET executes a tight pipeline of modules, which includes uploading datasets and pre-trained models (optional), setting up model configurations, training, validating and deploying. Additionally, built-in modules support automatic pre-processing of datasets, monitoring training performance, and saving the training history (which can be reloaded to continue training or to restore a previous model version) and diagnostic figures. Based on these utilities, users can either launch and apply pre-trained models to new datasets or design new DL or TL models in a short time frame.

3.3.1 Data Management

Preparing a good quality dataset for neural network training and model building is a challenging and labour-intensive task. It is often complicated to set up a robust quality control

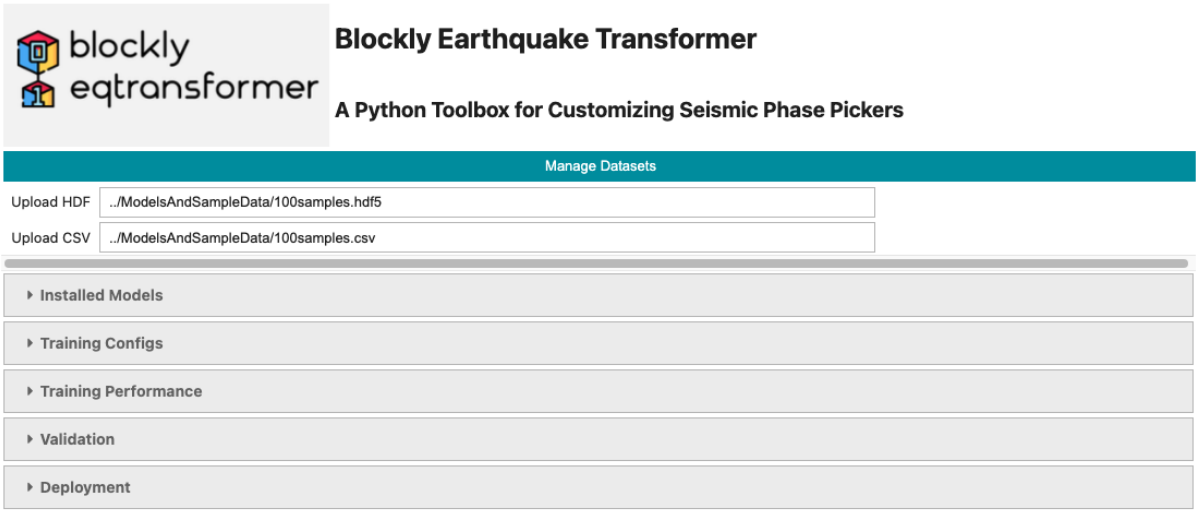


Figure 3.1: **Screen Capture of the Blockly Earthquake Transformer Dashboard.** BET provides a complete interactive workflow. Users can run BET's modules by defining their parameters interactively.

procedure to eliminate erroneous labels and make the distribution of training data as diverse as possible. The BET data management module allows users to upload seismograms, which will be saved in a single Hierarchical Data Format version 5 (HDF5) archive file. The related information (metadata) should be supplied in a comma-separated values (CSV) file. If the file format of the uploaded dataset is compatible with STEAD (Mousavi et al., 2019a), BET can automatically parse it. However, if there are any complications, it is recommended to use QuakeLabeler (Mai and Audet, 2022) first to convert the dataset into the STEAD format. If the configuration of the uploaded data (e.g., sample length, input channel size) conflicts with the model configuration requested, BET's data management module will initialize the built-in pre-processing functions to attempt to fit the dataset into the model's prerequisites, if possible. For instance, if the length of raw seismic traces is less than the model's pre-defined channel length, the data augmentation module will be activated to adaptively duplicate a fragment of the trace away from the seismic signals to pad to the required sample length (see an example in Figure 3.2); if the length of raw seismic traces is non-uniform and beyond the required input shape, a trimming method is applied to crop the trace where seismic signals exist (Figure 3.2).

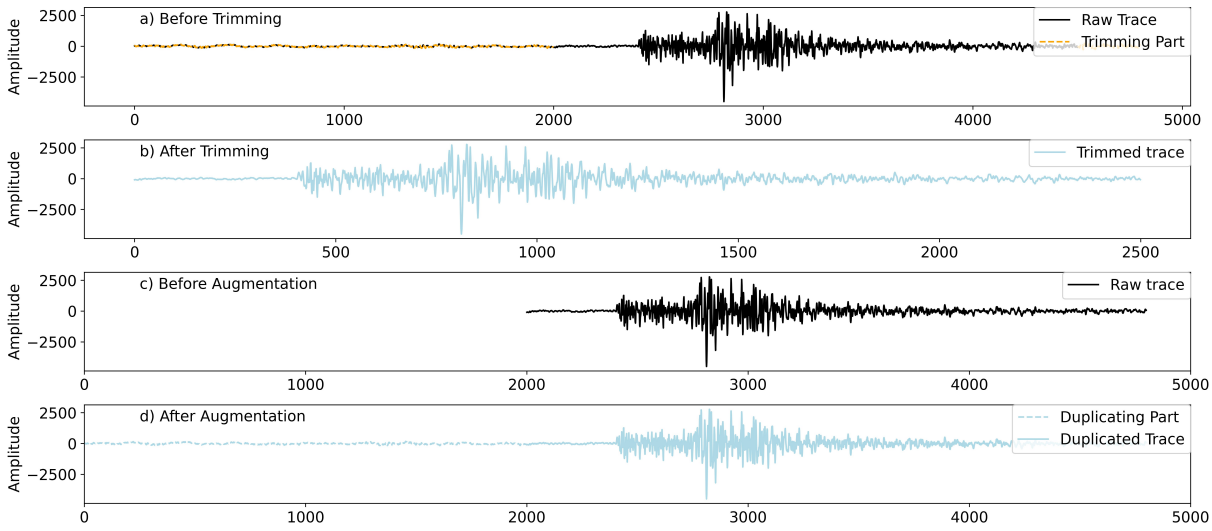


Figure 3.2: **Example of trimming and data augmentation.** In 3.2 a, the model's input size requires 2500 sample points, but the raw data has 4500 sample points per trace. The BET trimming module adaptively trims 2000 non-signal sample points from both ends of the raw trace. 3.2 b is the trimmed trace which fits the input size. In 3.2 c, another model's input size requires 4,800 sample points, but the raw trace only has 2,500 sample points per trace. The BET data augmentation module automatically adaptively duplicates 2,300 non-signal sample points that are pre-appended to the raw trace. 3.2 d shows the trace after data augmentation.

3.3.2 Model setup

Two types of model installation options are available in BET: 1) load a pre-trained model, and 2) create a new model. The first option will initialize a DL model based on the selected pre-trained network structure and hyper-parameters. The advantage of this option is that training data size can be minimal since the common seismic signal characteristics have been previously understood by the pre-trained models. Later, users can apply transfer learning, fine-tuning, and/or new model methods to customize models using a "warm start" (i.e., an optimal solution to a simpler optimization problem that sets initial values based on parameter information gained from a previous training dataset). The option to create new models is targeted to advanced users who wish to explore new architectures in addressing more complex tasks, for which current DL pickers face obstacles. In either case, users can use the BET dashboard to create model architectures without any coding involved (Figure 3.3).

Transfer learning and fine-tuning

Transfer learning and fine-tuning options help users to optimize the performance of a pre-trained model to a new data set (i.e., a new target region, new epicentral distance ranges, or phase types) with minimum effort. They minimize the model's revision procedures, but often produce high-quality models that result in performance enhancement. In transfer learning and fine-tuning, most of the hidden layers are set as frozen layers, which means that their hyper-parameters won't be updated during training; frozen layers will not cost further training time. Therefore, most of the computation resources are instead dedicated to the learning of new unique features representing a particular task or dataset at the very bottom hidden layer (where more high-resolution features are extracted from the data).

Note that the transfer learning and fine-tuning modes request users to select one of BET's supported pre-trained models (hyper-parameters combined) in the model setup dashboard. This is the most convenient way to start a new seismic signal detection/picker project when data and/or computation resources are limited. The default transfer learning mode is the easiest way to start a new project because it only requests users to design input/output (I/O) channels and set training strategies. If the transfer learning mode cannot realize the user's performance requirements, BET offers the fine-tuning mode to activate more frozen layers and make small and precise adjustments of the model's hyper-parameters. This results in improved models but at the cost of an increase in training time.

Creating new models

BET allows designing new DL models based on Earthquake Transformer(EqT)'s network architecture (Mousavi et al., 2020), which has been one of the most successful deep learning approaches, both for seismic signal detection and phase-picking tasks. In recent years, many transfer learning applications have shown that EqT is well suited for the migration to different phase-picking tasks after re-training or light-weight model revisions (Xiao et al., 2021; Zhang et al., 2022; Jiang et al., 2021). Figure 3.4 shows the general EqT framework, which is used by default by BET. Users can revise the framework by specifying different arguments via this interactive form, e.g., numbers of filters, kernel size, padding function, activation method, amount of 1D convolutional neural network (CNN) blocks and bi-directional and uni-directional long-short-term memory blocks. These arguments control the framework of the newly created DL model, which will generate an EqT-like DL picker. This customization can be done via the BET dashboard, which means that no programming is required. Furthermore, BET also provides the complete API hidden in its Jupyter

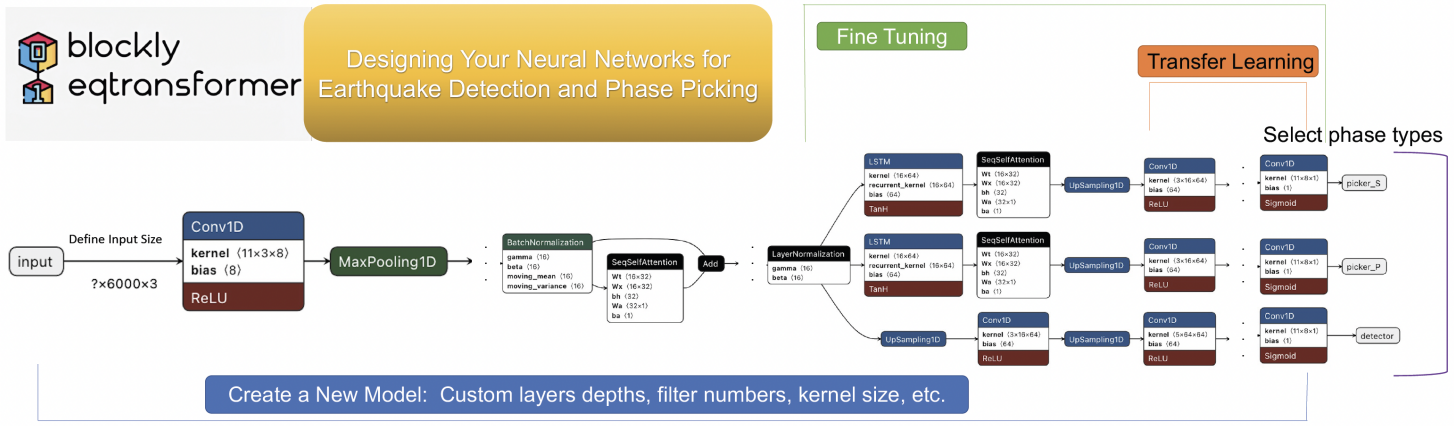


Figure 3.3: **Schematic diagram of BET's model setup framework.** BET provides a visual interface for building and training neural networks. By default, the 'Create a New Model' module allows users to select the number of layers, the number of units in each layer, etc., to fully design a new model. 'Transfer Learning' and 'Fine Tuning' modules can load pre-trained model architectures and weights, then select to re-train part of the layers to improve the model's accuracy in a short time.

notebook, giving advanced users the possibility to rewrite or extend the other parts of the network architecture, e.g., adding new types of neural network units.

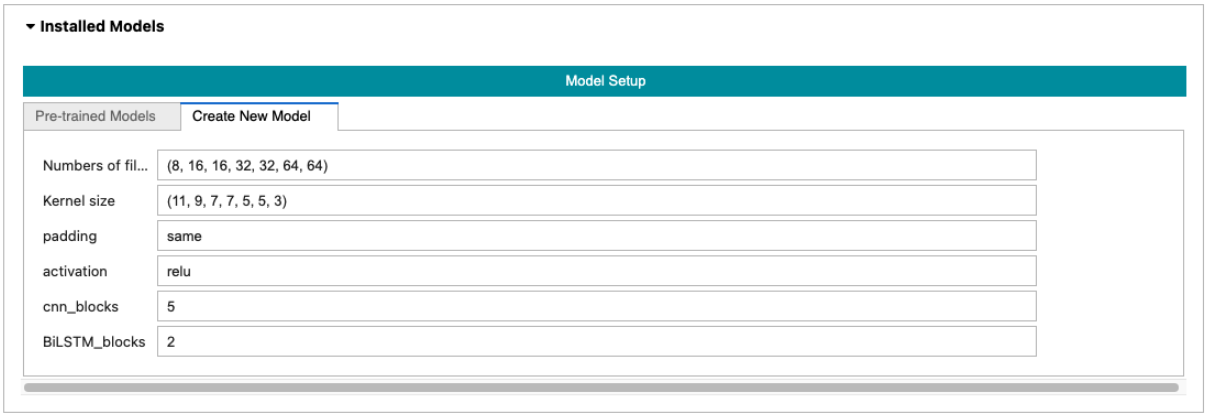


Figure 3.4: **Screen Capture of BET's Model Setup dashboard, with the "Create New Model" tab activated.** Users can customize the model architecture via this dashboard to build new models at different scales.

3.3.3 Training Configuration

Designing a training configuration is the last step before training a DL model, and can be done interactively through the BET dashboard ("Training Configs"; see Fig. 3.5). Most of these arguments are straightforward, and their descriptions are shown in Table 3.1. The most important part of the training configuration is the designing of input/output (I/O) channels. Here we give a brief overview to help users understand the mechanism of I/O channels in a DL picker.

Input channel

In a DL project, prepared datasets must be compatible with I/O channels of the pre-trained model, e.g., the EqT model receives 3-component seismic traces with 6000 sample points each ($6000 * 3$). If the new datasets do not satisfy the model's input shape, errors will be raised during training unless the data have been correctly trimmed or augmented (Mousavi et al., 2020). This always hinders the quick application of pre-trained DL models to new data; fortunately, BET allows users to customize the input data shape. For instance, a

Table 3.1: Training Configuration

Argument Name	Description
Train-Test Split Ratio	Retained data used to test model performance.
Drop Rate	Dropping out rates (to avoid over-fitting).
Batch Size	Amount of training examples utilized in one iteration.
Epoch	One training iteration.
Patience	The number of epochs to wait before an early stop if there is no progress.
Output Name	Folder name of the generated DL/TL model.

single trace input (e.g., $5000 * 1$) or multiple input channels (e.g., $6000 * 5$) are allowed. Once users complete the parameter input form, BET will automatically rewrite the input data format. To offer more flexibility, BET will also check the compatibility between the training data and the model's input shape, and (if possible) automatically pre-process the training data to ensure compatibility. Various input data sampling rates are acceptable, making it possible to utilize a variety of datasets in the DL models.

Output channel

The output channels represent the prediction probabilities of each identified seismic signal or phase existing at each sample point. BET defines output channels from a pre-defined list of phase types (or labels) shown in Figure 3.5. Based on the supported labels, various output combinations can be formed. As a default, BET offers three major categories of output channels: seismic signal detector, P-phase picker and S-phase picker. Seismic signal detector distinguishes between seismic signals and noise. Note that, in contrast to the original EqT model output that annotates the full length of the seismic signal on the seismograms, the seismic signal detector channel in BET annotates a very limited time window (about 1-2 seconds), which is similar to the phase picker channels. Experimental results showed that, in this way, convergence of the model is faster during training. For the phase picker channels, users can select the default P and S pickers, or additional channel identifiers, i.e., P, Pg, Pn, S, Sg, Sn, etc, if those labels are available in the training metadata (i.e., catalog picks). BET will automatically rewrite the output channels based on these selections. Alternatively, users can use the default P and S labels as surrogate for other binary labels, for example, blasts and natural earthquakes, to start training a new model for the discrimination of anthropogenic and natural events.

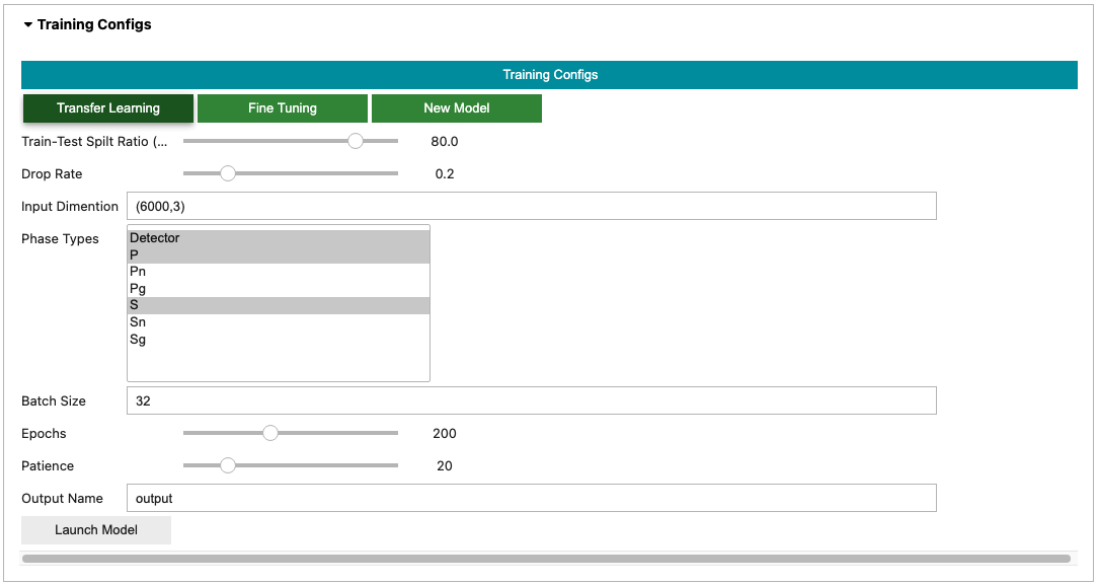


Figure 3.5: **Screen Capture of BET's Training Configuration Dashboard.** In this dashboard, users can select different training options, i.e., TL, fine-tuning or training a new model, and other training related arguments.

3.3.4 From Performance to Deployment

Training performance

BET has a built-in visualization tool to display the training performance and monitor the loss function of all defined phase types. The loss function is a method to evaluate the difference between predicted seismic arrival times and the ground truth labels (authentic phase labels), where loss is proportional to error rate (Christoffersen and Jacobs, 2004). For example, in Figure 3.6, the loss function decreases to a very low rate and remains stable, which means a successful training procedure is complete and the trained model has a very low probability of making an erroneous prediction in arrival time. We note that not all training procedures will be successful and generate ideal loss functions like the ones shown in Figure 3.6. For a well-trained dataset, generally 25 training iterations (epochs) or greater are required to minimize loss for P- and S-wave pickers and detectors. It is common to encounter loss functions that remain high or fluctuate. In these cases, users need to consider redesigning the network architecture and retraining it.

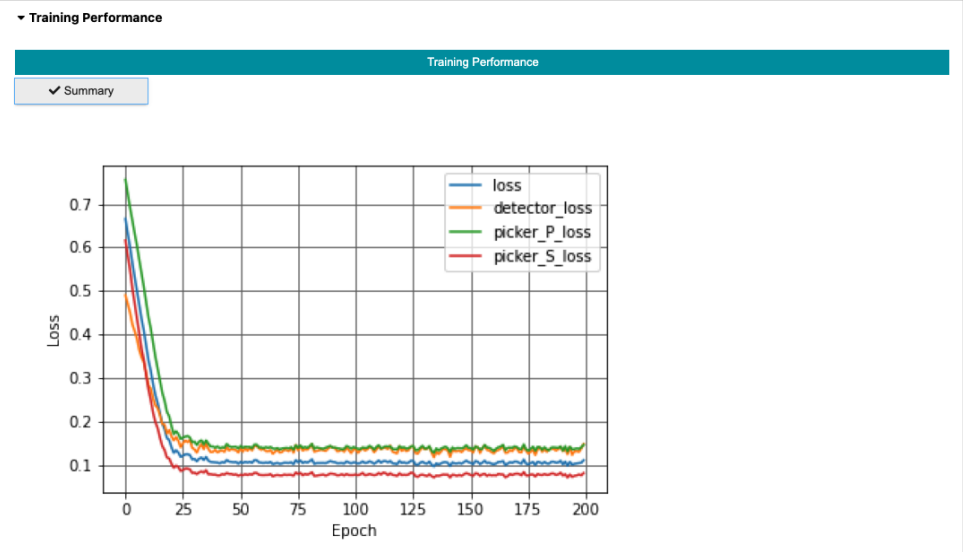


Figure 3.6: **Screen capture of training performance.** The loss function for each output channel and the overall loss decrease linearly with epoch until they reach a stable value, where loss no longer decreases.

Validation and deployment

Validating the newly trained DL model is essential to ensure its performance and accuracy (Eelbode et al., 2021). During validation, a sufficient quantity of samples in proportion to the overall size of the dataset, but not considered in the training set, is used to validate the model's performance. These samples are preserved from the split dataset. If the validation results are as good as in the training dataset, the model has not been over-fitted and the new picker is ready to be deployed. Although there are no standards for evaluating a model, BET provides three types of supporting files to help users evaluate their trained models. In the specified output folder, users can find detailed test results to evaluate the model's performance, i.e., a summary table (*.CSV) containing all the picked results, a text file (*.TXT) reporting the configuration parameters for prediction and model performance, and a figure folder containing the picked arrivals (default: 10 examples of picks). BET also offers a deployment dashboard to quickly apply newly trained pickers to large-scale datasets (see Figure 3.7). When the deployment is complete, BET exports several sample graphs and human-readable documents to help users visualize and understand the model predictions (see Figure 3.13). These extensions provide ways to visualize and understand model predictions, avoid repetitive labour and let users focus on analyzing performance.

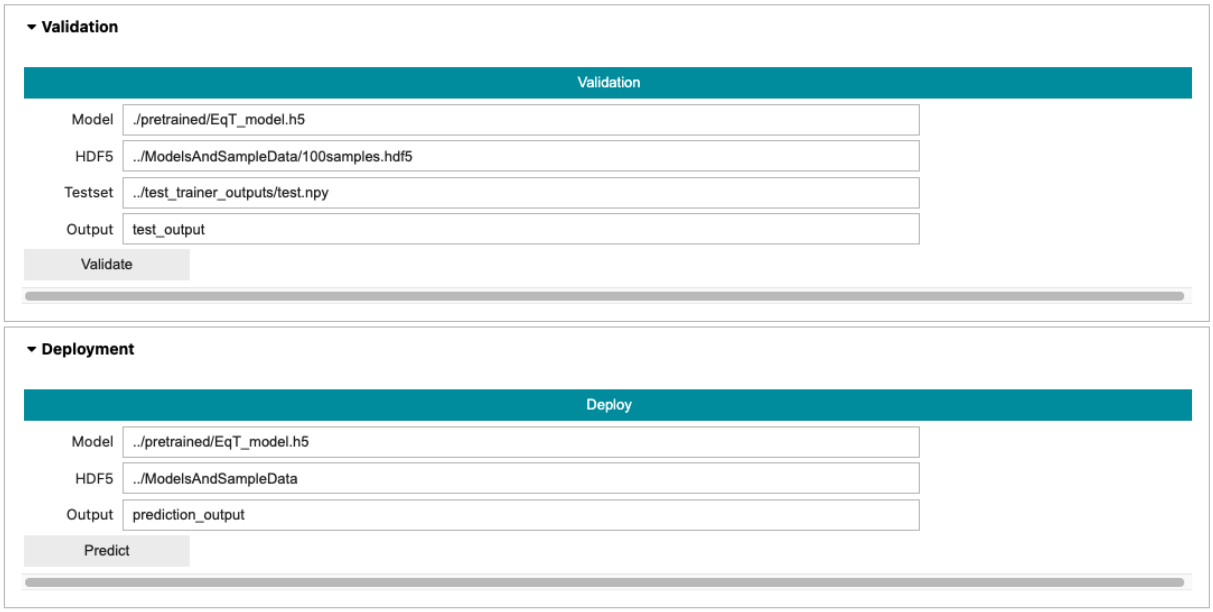


Figure 3.7: **Screen Capture of Validation and Deployment Dashboard.** When the user finishes a training process, the validation and/or deployment dashboard can be activated based on training results, e.g., input training output folder and other user-defined arguments.

3.4 Prospective Applications

In this section, we present three examples that cater to novice, intermediate and advanced users, respectively, to illustrate the flexibility and applicability of BET for phase-picking projects. For novice users, BET provides a convenient way to understand how deep learning models can be deployed in earthquake detection and phase-picking projects. For intermediate users, BET can be used to train new deep learning and/or transfer learning models to address specialized phase-picking tasks. For advanced users, they can fully customize deep-learning models to create novel deep-learning pickers. The focus of this section is to give users a quick overview of BET's capabilities, rather than to showcase the accuracy of any particular trained model. We will focus on various case studies in future work.

3.4.1 Deploying pre-trained models for earthquake detection and phase picking

Using a pre-trained model can be a good way to get started with deep learning, especially for novice users with little experience in DL model training. To deploy a pre-trained EqT model on a new dataset, the user may skip the training and validation steps, and start with the deployment module. BET provides built-in, pre-trained models and associated files, such as weights. Alternatively, users can download pre-trained models, obtained from other transfer learning applications using the EqT framework. The user then defines the model's file path on the BET deployment interface (see Fig. 3.8), and BET will load the model, the associated weights and the configuration. Other than choosing a pre-trained model, users only need to pass the new dataset through the model and use the probability output to make predictions. On the user's end, BET requires the prepared dataset directory be defined (see Fig. 3.8). The input dataset should be similar to the STEAD dataset (Mousavi et al., 2019a) to avoid running into errors. Building a dataset for deep learning may be challenging for novice users. To remediate this, BET provides a tutorial notebook to convert seismic data to a STEAD-like dataset. Alternatively, users can use QuakeLabeler to generate a dataset that matches the STEAD format (Mai and Audet, 2022). As described previously, BET has an auto-preprocessing module to adjust the input dataset (e.g., reshape the input size, drop data with insufficient information, etc.) to avoid errors during deployment.

Here we use a built-in dataset in STEAD format to showcase how to use the pre-trained EqT model to pick events. This dataset is the same example dataset as the original Earthquake Transformer package (Mousavi et al., 2020); it contains 99 human-reviewed local earthquake events. To try this example, users can skip to the deployment interface and press the 'Predict' button. BET will then load the pre-trained EqT model automatically (see Fig. 3.8). Typically, it takes 1-2 minutes to load the pre-trained model in the backend, depending on the available computational resources. In the second step, BET will auto-extract waveforms from the user input dataset directory's HDF5 file and send them to the model's input channel. The pre-trained EqT model predicts the probabilities of the P-phase and S-phase on each data point. In the final step, all detected events are saved in a CSV file with some example visuals stored in a separate folder.

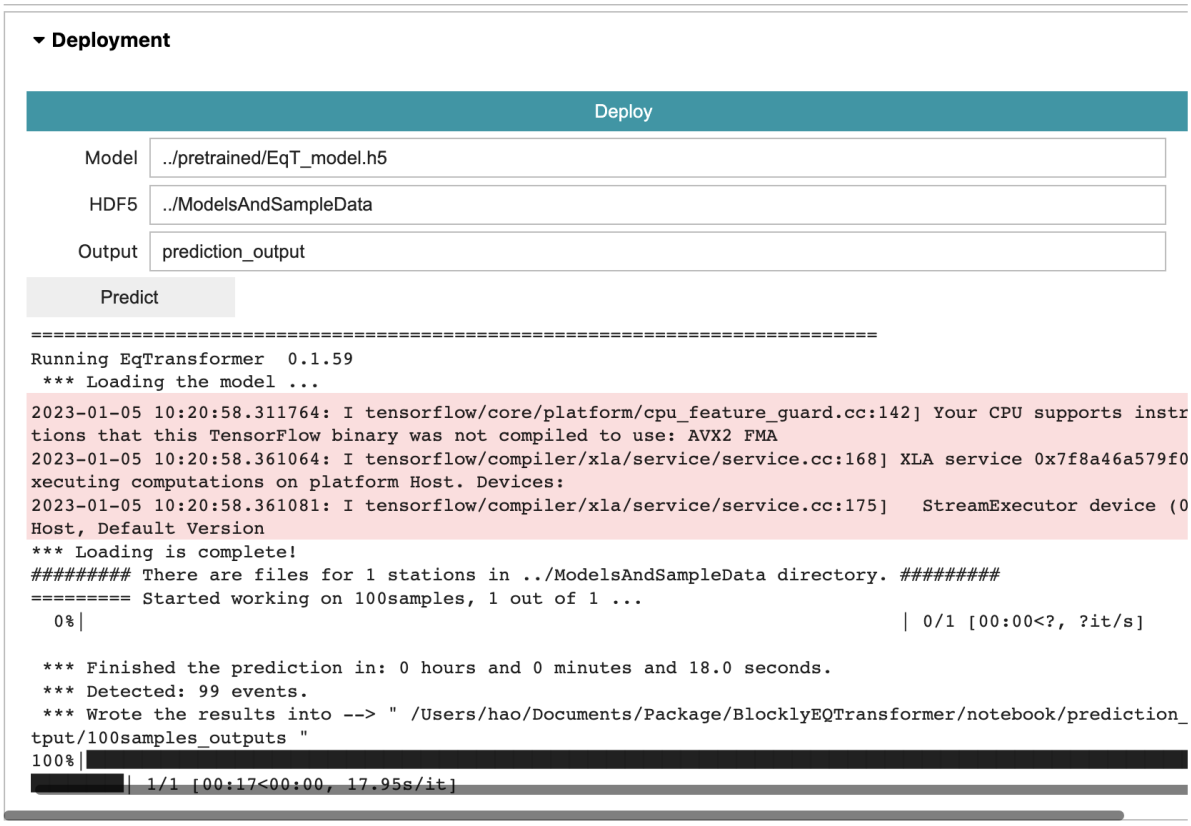


Figure 3.8: **Screenshot of using a pre-trained model to detect and pick earthquake events.** Users can use BET's built-in pre-trained models (and weights) to predict earthquake events. Following the procedure outlined in this paper, the user defines the file path and selects the 'Predict' button. BET automatically detects the earthquake events in the dataset and generates results and example visuals in the output folder. The deployment stage generally takes a few minutes, depending on the size of the dataset and computational resources.

3.4.2 Training models on new datasets

Deploying an AI picker in a new scenario (e.g., different input channel requirements, specific regions with no prior training data, etc.) may require the laborious adaptation of previously published DL or TL models. BET is designed to facilitate the training of new deep-learning models on user-defined datasets. The first step for training a new model

is dataset preparation in the STEAD format. We illustrate the training functionality in BET by collecting training data waveforms from the Canadian National Seismograph Network, CNSN (Natural Resources Canada, 1975) and events from the National Earthquake DataBase, NEDB (Natural Resources Canada, 1985) for Cascadia Subduction Zone seismicity between 2015-2019 (see Data and Resources). For the training dataset, a subset of 726 earthquakes, including 27,219 manual picks of magnitude ≥ 2 (see Fig. 3.9) are extracted from the NEDB between 2015-01-01 and 2019-12-31. A magnitude threshold of 2 was chosen based on previous studies (Yeck et al., 2021). We first use this dataset to evaluate the pre-trained EqT model for this new region (Cascadia) and use transfer learning to improve the model.

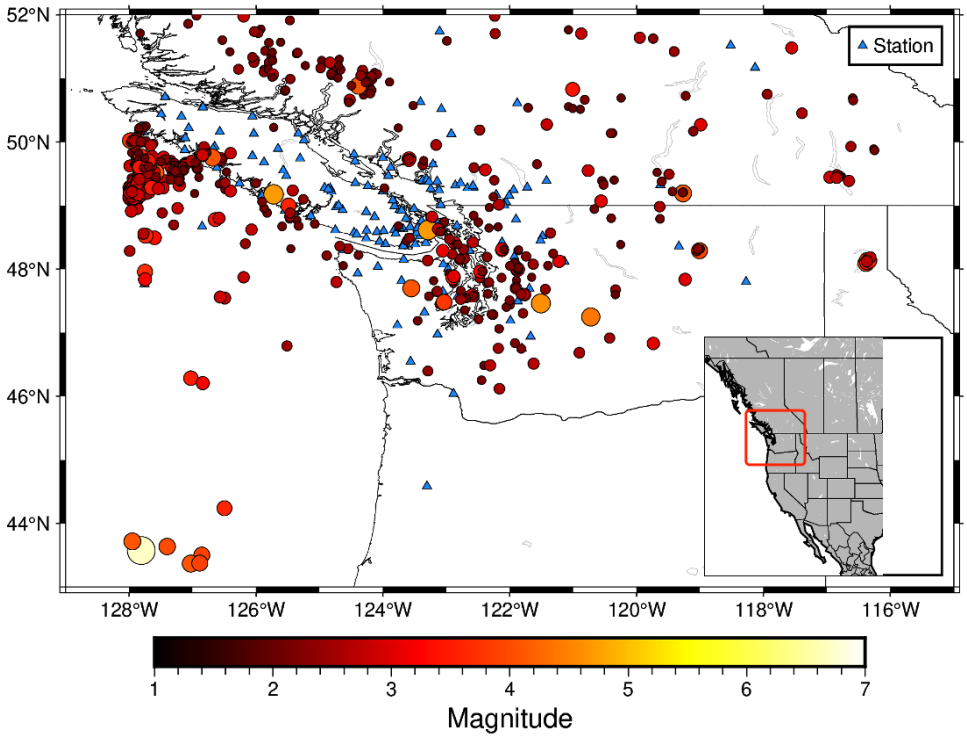


Figure 3.9: **Geographic data distribution of earthquakes of magnitude M $\geq$ 2, from the training dataset extracted from the NEDB catalog (Natural Resources Canada, 1985) from 2015-01-01 to 2019-12-31.** Blue triangles represent the seismic stations. Circles represent earthquakes, with size and color scaled to event magnitudes. The NEDB dataset contains single trace P- and S-labeled, manually-picked phases and is used in this study as a training dataset to introduce BET's training module.

The second step is to define the architecture of the model. Here we select the 'Transfer Learning' function for illustration. In the new dataset, each sample uses one channel of input trace; the vertical component for P arrivals and one horizontal component for S arrivals. This input structure differs from most published DL models, making direct application of the pre-trained EqT model challenging. However, for transfer learning on the EqT model, the user needs only to specify that the input shape is that of the original EqT model(i.e., $6000 * 3$) and set up other arguments as shown in Table 3.2. The new dataset will be automatically formatted as the EqT model, and fed into training in BET's backend. In this case, EqT pre-trained model weights are still acceptable as a warm start to accelerate model convergence.

The transfer learning and fine-tuning options are suitable for users who do not have access to arrays of GPUs for computation. A single desktop-level GPU (e.g., GTX 1080I) or workstation with CPU cores (e.g., 8 cores of Intel i7) can complete a training step in a few hours. When training is complete, the next step is to test the accuracy of the trained picker in the validation dashboard. We use the validation dataset (2,721 samples), which gets automatically split when loading the dataset, to test BET's prediction accuracy. Performance comparison of the original EQTransformer (V1.59) model and the BET newly-trained transfer-learning model (Table 3.3), demonstrates that the new picker has improved detection ability. The original EqT model's low accuracy suggests that the NEDB data distribution differs from the STEAD dataset. Figure 3.10 shows a predicted result from the newly trained picker. This example illustrates a promising method for improving picking in a region where the DL model has never been applied.

Table 3.2: Model Configuration for Cascadia Picker

Model Configuration	
Input Dimension	(6000, 3)
Train-Test Spilt Ratio	0.8
Training Mode	Transfer learning
Drop Rate	0.2
Phase Types	P, S
Batch Size	100
Epochs	100
Patience	20

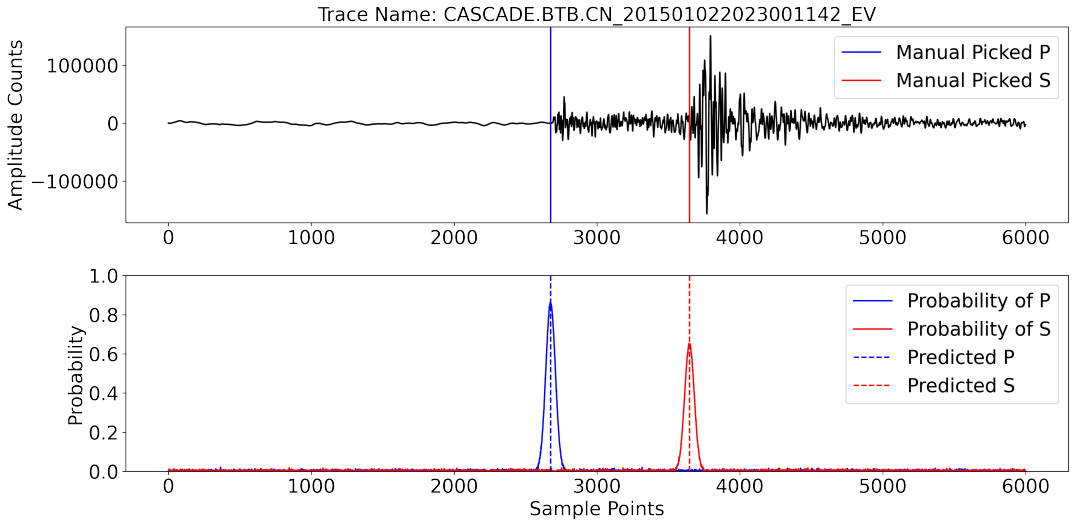


Figure 3.10: **Representative predicted P and S arrival in Cascadia test dataset.** The trained picker correctly identifies P and S phase from a N-component seismic trace. The human-reviewed P and S arrival times are taken from the NEDB catalog at station CN.BTB for a magnitude 2.0 event on 2015-01-02 at an epicentral distance of 78 km. The picker predicts a high probability (>0.6) at the correct sample position.

Table 3.3: Model Performance on the CNSN/NEDB Validation Dataset

Model	P-phase accuracy	S-phase accuracy
EQTransformer (V1.59)	73.9%	66.6%
BET New Trained Picker	92.3%	88.6%

3.4.3 Beyond P- and S-phase picking

Here we present a case where BET is used to train a comprehensive AI phase picker to extend the detectability of P and S phase arrivals and further include direct crustal phases (Pg and Sg) and/or Moho refracted phases (Pn and Sn). We use BET's fine-tuning mode to create the model. In this example, we select all phase types in the configuration dashboard (i.e., P, Pg, Pn, S, Sg, Sn), define the input channel as $6000*3$ and use default values for all training settings. We download three months (i.e., from June to September 2013) of data from the U.S. Geological Survey (USGS) machine learning dataset (Cole and Yeck, 2022) and convert it into STEAD format. This USGS dataset is publicly available and contains a wide range of human-reviewed samples of the above 6 phase types. In this project, we set 0.8 as the training-test-split ratio to produce 31,254 training samples and 7,814 validation samples (the data distribution map is shown in Fig. 3.11). In the USGS dataset, each phase is centered in the trace, i.e., the original waveform is sampled at 40 Hz with a start time 60 seconds before the first P-wave arrival, so the arrival time is at the 2,400 sample point. Therefore, the original ground truth labels are fixed at a static position, which will cause problems for the model in understanding the spatial features of the picking tasks. However, once the data loading function is activated, BET will automatically pre-process the dataset in its backend, augment sample length to the input shape and randomize arrival positions to avoid biased training.

In this example, training reaches its best performance after only 7 epochs, at which point the training is stopped automatically (after 2 hours 20 minutes of training time). The pre-trained model (EqT version 1.59 model) provides the basic network structure and a warm start. In this training solution, the total number of hyper-parameters is 378,667, but there are only 4,871 trainable hyper-parameters. The 373,796 non-trainable hyper-parameters are frozen layer units. This significantly reduces the training time and required data size. The training is mainly limited to fine-tuning the new TL model to capture the characteristics of new seismic phases (i.e., Pg, Pn, Sg, and Sn) by updating the learned features at the final layers of the EqT baseline model, enabling the model to distinguish between these phases. According to Table 3.4, the last three epochs of loss value are relatively low (below 0.0420855), indicating a successful training despite a low number of

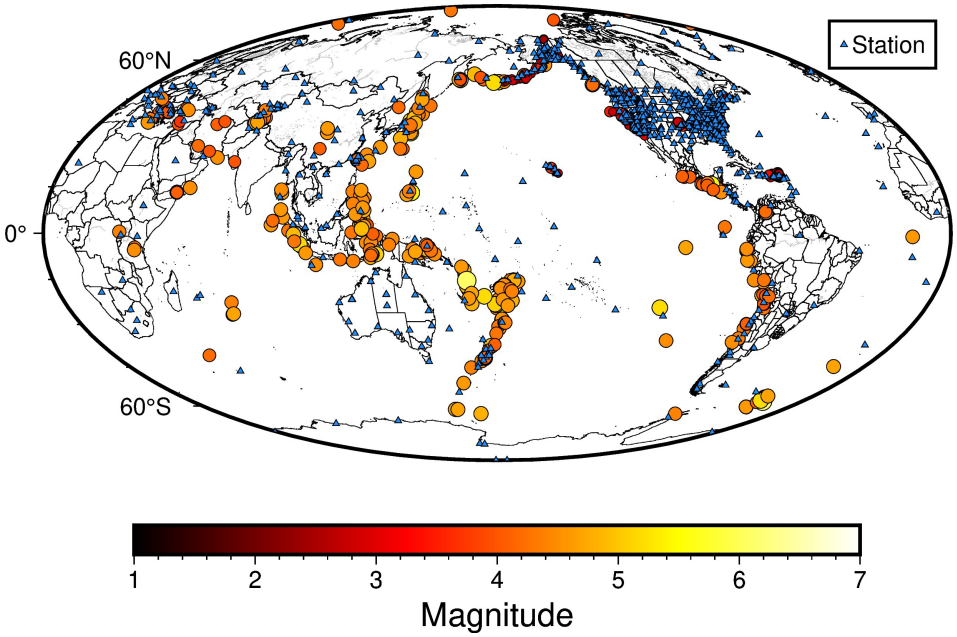


Figure 3.11: **Geographic Data Distribution of the USGS dataset (Cole and Yeck, 2022) used in this application.** The blue triangles represent the available seismic stations. The circles represent earthquakes color-coded by magnitude. The USGS dataset contains human-reviewed P, Pn, Pg, S, Sn and Sg labeled samples, and is used here as a training dataset to test BET's applicability.

iterations and training samples. The accuracy in the validation set also indicates that this newly trained phase picker can correctly detect P, Pg, Pn, S, Sg, Sn phases in the new data in a similar way as the usual P and S phases. The output probability channel predicts relatively high values (i.e., Pn, Pg channel > 0.8; Sn, Sg channel > 0.6) at the location of seismic arrivals for the correct phase output channels (see Fig. 3.12), with other channels correctly predicting very low probabilities (see Fig. 3.13).

3.5 Extensibility

BET allows users to customize various types of DL pickers. Based on different types of datasets, BET can train models that encompass regional to teleseismic events and identify signal types such as natural earthquakes, explosions, volcanic or even vehicle tremors. In phase-picking tasks, phase types are no longer limited to only P and S phases, and can be

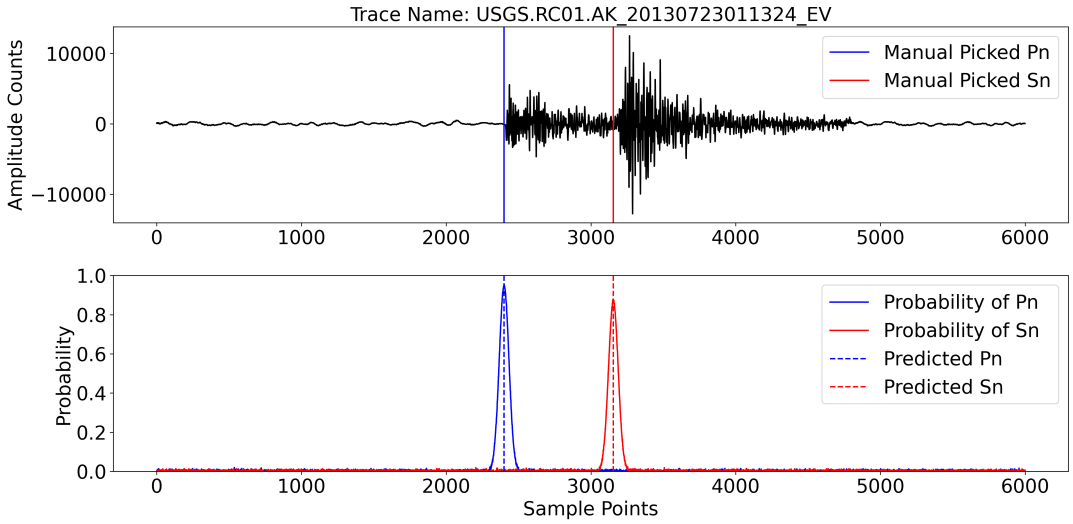


Figure 3.12: **Representative predicted Pn and Sn arrivals in the USGS dataset (Cole and Yeck, 2022).** The fine-tuned picker successfully detects a Pn-arrival and a Sn-arrival at station AK.RC01 from the USGS validation dataset. This event is human-reviewed with a magnitude of 2.9 recorded at a distance of 161 km from the epicenter.

extended to P, Pg, Pn, S, Sg, and Sn. Furthermore, advanced users can implement other phase types (i.e., to distinguish between natural earthquakes and blasts) by expanding BET's output channels at the scripting level. BET's Jupyter notebook can be used to replace the **P** and **S** types with **Natural Earthquakes** and **Blasts**, respectively. Then, by running voila (Silaparasetty, 2020) to reinitialize BET, these 2 new channels can be utilized for training. More complex extensions are also allowable in BET's scripting level and outlined in the user manual.

3.6 Conclusion

The application and development of DL models are steadily growing in seismology. BET provides a user-friendly platform for engaging a broader range of users, including those without machine learning or coding experience. BET allows researchers to explore DL pickers without having to worry about in-house computational power and other technical concerns. BET will prove useful in improving model performance and lowering the magnitude of completeness of existing seismic catalogues. Similar packages can be devel-

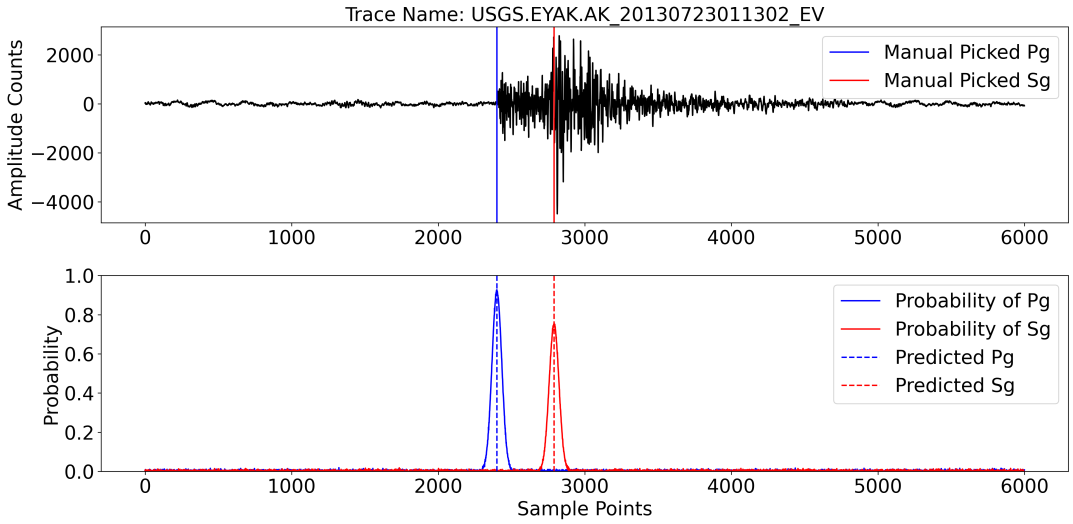


Figure 3.13: **Representative predicted Pg and Sg arrivals in the USGS dataset (Cole and Yeck, 2022).** The custom BET picker successfully detects a Pg-arrival and a Sg-arrival at station AK.EYAK from the USGS validation dataset. This event is human-reviewed with a magnitude of 2.9 recorded at a distance of 65 km from the epicenter.

oped for other seismological tasks, including those in exploration seismology, where fine-tuning pre-trained models by synthetic data is even more common due to the scarcity of large-scale labeled training datasets. Other applications of BET are in the discrimination between natural earthquakes and anthropogenic events, where separate training datasets may be developed for each event types. This package will be both practical for students to understand deep learning applications and convenient for research groups to deploy large-scale phase-picking tasks. The BET package is available as a GitHub repository at https://github.com/maihao14/BlocklyEQTransformer. We encourage users to contribute trained models to enrich the library of BET models.

3.7 Declaration of Competing Interests

The authors acknowledge there are no conflicts of interest recorded.

Table 3.4: Transfer Learning Performance in USGS Dataset

Phase Type	Last Value of Loss Function	Accuracy in Validation
P	0.0032257	99.5%
Pg	0.0100786	97.8%
Pn	0.0058863	98.2%
S	0.0113646	93.6%
Sg	0.0269472	94.5%
Sn	0.0420855	92.4%

3.8 Data and Resources

The (Canadian) National Earthquake Database may be found at https://www.earthquakescanada.nrcan.gc.ca/stndon/NEDB-BNDS/bulletin-en.php (last accessed June 2023). At this time, the online database is searchable only from 1985 to the present and returns only the solutions. Solutions and phase information for older earthquakes may be obtained by contacting nrcan.earthquakeinfo-infoseisme.rncan@canada.ca. Waveform data and/or station metadata from the CNSN, may be retrieved via FDSN dataselect webservice https://earthquakescanada.nrcan.gc.ca/fdsnws/dataselect/1/, via FDSN station webservice https://earthquakescanada.nrcan.gc.ca/fdsnws/station/1/, or via Station Book https://earthquakescanada.nrcan.gc.ca/stndon/CNSN-RNSC/stnbook-cahierstn/index-en.php.

The Global Earthquake Machine Learning Dataset: Machine Learning Asset Aggregation of the PDE (MLAAPDE) from USGS may be found at https://www.sciencebase.gov/catalog/item/6127b30fd34e40dd9c05094c (last accessed December 2022). USGS dataset provides 6 types of human-reviewed phases, i.e., P, Pg, Pn, S, Sg, Sn.

3.9 Acknowledgments

This research was funded by a Discovery Grant (RGPIN-2018-03752) from the Natural Science and Engineering Research Council of Canada (PA). The authors thank Stephen Crane, Laurel Sinclair, two anonymous reviewers and the editor for their reviews of this manuscript. This is NRCan publication number 20220610.

Chapter 4

Investigating the Impact of Dataset Architecture on Deep Learning Models' Performance for Automated Phase-Picking

4.1 Abstract

The use of deep learning (DL) in earthquake detection and phase-picking tasks has produced transformative results in recent years. Driven by large seismic datasets, DL pickers hold much promise in improving the accuracy of automated picks. However, the regionalization of seismic velocity and attenuation models makes the application of pre-established phase pickers to new target regions challenging if the input seismic data distribution is not reflected in the original training datasets. Furthermore, transfer learning of DL pickers may not be possible due to the lack of reliable human-reviewed waveforms for training. Perhaps the greatest challenge is that seismologists have no a priori knowledge of the number of waveforms required for model training to achieve their desired phase-picking accuracy and model residuals, or which proposed DL pickers can be applied directly to a new target region without re-training. In this study, we explore the issues of DL model performance by investigating the effect of increasing training sample sizes and examining different deployment settings applied to new data. To this end, we retrain two of the most popular DL pickers, PhaseNet and EQTransformer, using training datasets of various size and then test the phase picking accuracy with the same validation set. From this study, we

gain insight into how many waveforms should be included in a new DL project and which additional factors (e.g., data preprocessing and standardization, picking method, tectonic setting, etc.) might affect training and model performance. Our study provides a guide for determining the optimal size of the training data set and model selection for future studies.

4.2 Introduction

The rapidly increasing volume of digital data in global earthquake monitoring operations presents a growing risk of human error and increased labor (Havskov and Ottemoller, 2010). To address these concerns, automated workflows have been implemented to replace or assist time-consuming manual operations, such as seismic discrimination and phase picking (Gao et al., 2022). However, due to the complexity of earthquake waveforms, monitoring still heavily relies on the careful eye of seismic analysts (Mousavi and Beroza, 2022a). The application of deep learning (DL) to earthquake monitoring has gained popularity as a result of its improved performance compared to other automated methods, such as short-time-average to long-time-average trigger (STA/LTA) and Akaike information criterion, especially in noisy environments (Dimililer et al., 2021; Trnkoczy, 2009; Li et al., 2017). DL has been particularly successful in earthquake phase detection, leading to more comprehensive catalogues and a better understanding of Earth structure (Reichstein et al., 2019). In the past five years, numerous DL models have been proposed for event detection and phase picking in various scenarios, including natural earthquakes, volcano monitoring, and ocean-bottom seismic data processing (Mousavi and Beroza, 2023; Ma et al., 2020; Lapins et al., 2021; Bornstein et al., 2023; Ma et al., 2023; Zhang et al., 2021b). Successful DL models, such as Generalized Phase Detection, PhaseNet, and Earthquake Transformer (EQTransformer), have been widely adopted and implemented through transfer learning (Ross et al., 2018; Zhu and Beroza, 2019; Mousavi et al., 2020; Liu et al., 2020). Open-source DL platforms like SeisBench and Blockly Earthquake Transformer are further accelerating the development of new models (Woollam et al., 2022; Mai et al., 2023).

A key principle of DL models is that their high generalizability depends on the availability of large amounts of seismic data for training. For example, the STanford EArthquake Dataset (STEAD), which has been used to train PhaseNet and EQTransformer, contains over 200 thousand 3-component seismic waveforms (Mousavi et al., 2019a). Most successful DL pickers and their transfer learning models are trained on STEAD or similar published benchmark datasets, such as the Global Earthquake Machine Learning Dataset (MLAAPDE), the GEOFON (GEOFOrschungsNetz) dataset, the Italian seismic dataset

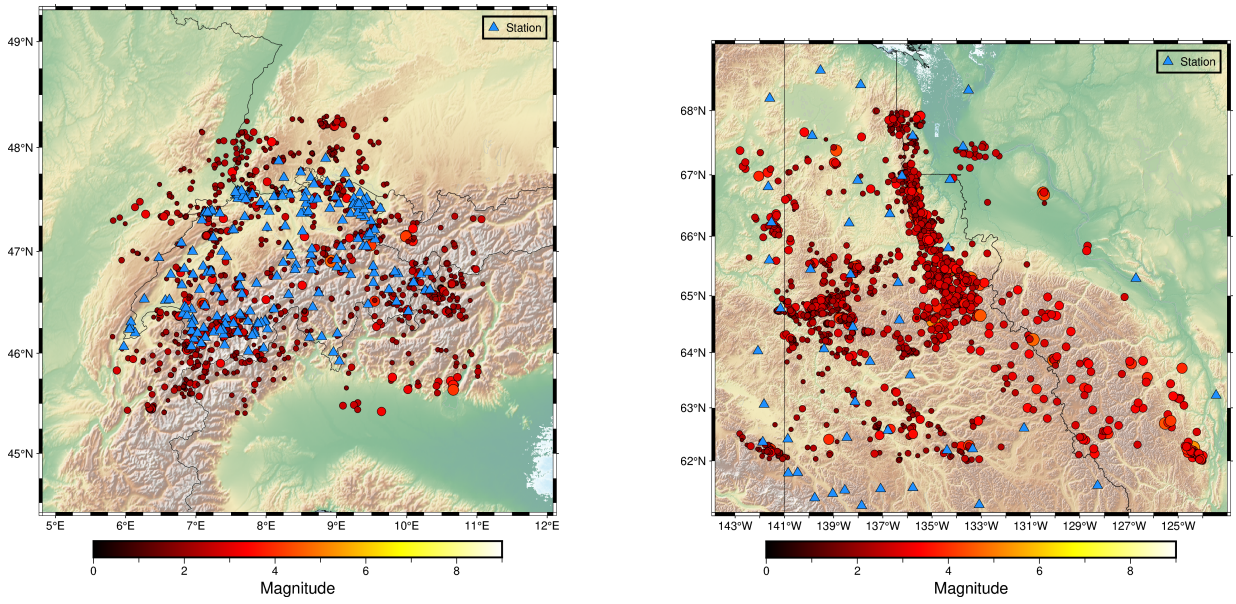
(INSTANCE), and the Chinese seismic benchmark dataset (DiTing) (Yeck et al., 2021; Hanka et al., 2000; Michelini et al., 2021; Zhao et al., 2022). Although these DL models have effectively addressed earthquake detection and automated phase picking problems on these benchmark datasets, directly implementing a published DL model in new target region remains challenging. Detection capability cannot be guaranteed even when the models are pre-trained on massive global benchmark datasets (Münchmeyer et al., 2022).

In practice, the entire development cycle of a DL model is lengthy and resource intensive (Chen et al., 2020). First, the collection of seismic data is both time-consuming and labor-intensive. Creating an entirely new benchmark dataset within a limited research time frame is nearly impossible for most researchers (Whang and Lee, 2020; Mai and Audet, 2022). In addition, it may not be feasible to provide manually labeled data in quantities equivalent to a benchmark dataset for certain specific research areas. Even if researchers manage to prepare a large dataset, training remains a challenge due to the large number of trainable parameters in deep learning models (Alahmari et al., 2020). Combined with computer memory requirements to handle large datasets, this task typically requires several days or even months of training time on high-performance GPUs (Hestness et al., 2019). Such extensive training is prohibitive for most researchers and significantly hinders the proposal of new models or their application in practical scenarios. Although some studies explore the limited training size of seismic data and apply transfer learning to solve overfitting and model error in seismic phase arrival detection, the effect of training data volume on the performance of DL pickers remains unclear (Lapins et al., 2021; Li et al., 2022), and many practical issues remain to be addressed.

In other research areas facing similar challenges, such as image classification in medical applications, numerous studies provide discussions on various effects on the accuracy of the DL models, such as training size, data pre-processing, training configurations, threshold settings, and more (Cho et al., 2015; Pérez-García et al., 2021; Chilimbi et al., 2014; Brownlee, 2018). To address these open questions and help seismologists make decisions for their DL projects, we conduct a series of tests to explore the effect of training size, waveform pre-processing, and the selected picking function in the overall model performance on new data. We then present empirical suggestions regarding which DL model is best suited for a new data set, and whether or not transfer learning is necessary. In this paper, we use Canadian seismic data as an example of a new dataset. Our goal is not to produce an optimal DL model for this dataset, but rather to provide practical suggestions for researchers planning to use DL methods in their work.

4.3 Data and Models

In this study, we make use of two different datasets of labeled waveforms from Switzerland and Canada to train various model architectures and to provide geographical diversity. The first is a published benchmark earthquake dataset called the ETHZ dataset, which compiles the earthquake catalog for Switzerland and surrounding regions (Woollam et al., 2022). The ETHZ dataset contains 36,743 waveform examples, including both regional earthquakes and noise. The ETHZ dataset is available through SeisBench (Woollam et al., 2022). The second dataset, called the Mackenzie dataset, is collected from the Canadian National Earthquake Database and consists of seismic data from publicly available networks throughout the Mackenzie Mountains in Yukon and the Northwest Territories, Canada. The Mackenzie dataset is the first published Canadian earthquake dataset for machine learning (released with this paper, see Data and Resources). The data in the Mackenzie dataset have never been included in any current publicly available benchmark seismic dataset or exposed to DL models. Thus, the Mackenzie dataset can be considered as new data to serve in our study, especially to provide accurate empirical results for researchers who want to apply pre-trained DL models using Canadian seismic data. The Mackenzie dataset compiled in this study consists of 54,547 human-reviewed phases of 1,666 natural earthquakes that occurred between January 1, 2015, and December 31, 2022. The seismic events captured in this dataset range in magnitude from 0.0 to 5.3 (Fig. 4.1). To maintain compatibility with the ETHZ dataset, we formatted the Mackenzie dataset using the SeisBench data format.



(a) (b)

Figure 4.1: **Geographical Distribution of the ETHZ and Mackenzie datasets.** Panel a shows the distribution of the ETHZ dataset, a benchmark earthquake dataset that compiles the earthquake catalog for Switzerland and surrounding regions (Woollam et al., 2022). Panel b presents the Mackenzie dataset, the first published Canadian earthquake dataset for machine learning, collected from the Canadian National Earthquake Database. This dataset includes seismic data from various networks deployed throughout the Mackenzie Mountains in Yukon and the Northwest Territories, Canada (see Data and Resources).

Numerous deep learning (DL) models have been proposed for earthquake detection and phase picking tasks, including transfer learning models tailored to specific scenarios (Mousavi and Beroza, 2023; Ma et al., 2020; Lapins et al., 2021; Bornstein et al., 2023; Ma et al., 2023). To maintain objectivity and ensure reproducibility in our experiments, we select six DL models from SeisBench as baseline models for an initial evaluation of model architectures. We used the Mackenzie test dataset to test the performance of the six models. A limited preprocessing workflow (resampling to 100 Hz, and removing trend and instrument response) was applied prior to testing according to the models' requirements (Woollam et al., 2022). Preliminary tests show that PhaseNet and EQTransformer outperformed the other DL models in the Mackenzie dataset (see Table 4.1). Therefore, we focus our investigations on the PhaseNet and EQTransformer model architectures. In addition, we use the same EQTransformer framework but trained on the ETHZ dataset

Table 4.1: Preliminary Test Results of the Six Existing Deep Learning Models in the Mackenzie Mountains, Canada

Model Name	Training set	Accuracy	Reference
BasicPhaseAE	N. Chile	7.8%	Woollam et al., 2019
CRED	N. California	10.7%	Mousavi, Zhu, et al., 2019
DPP	N. Chile	8.2%	Soto & Schurr, 2021
EQTransformer	STEAD	45.8%	Mousavi et al., 2020
GPD	S. California	13.2%	Ross et al., 2018
PhaseNet	STEAD	21.5%	Zhu & Beroza, 2019

(called the ETHZ model) as a third model for comparison (Woollam et al., 2022). We include the ETHZ model because we want to compare the performance in the same network architecture but trained on different training datasets.

Our analysis therefore includes the following models: PhaseNet, a U-net architecture with 23.3k trainable parameters, trained on 113k samples from the STEAD dataset; EQTransformer, a deep neural network with an attention mechanism with 376k trainable parameters, trained on the global STEAD dataset; and finally, the ETHZ model, which uses the same network architecture as EQTransformer, but is trained on the ETHZ dataset, a smaller dataset than STEAD (only 36,743 waveform samples), encompassing the Swiss Alps (Zhu and Beroza, 2019; Mousavi et al., 2020).

4.4 Experiments on Direct Deployments

One of our primary objectives in this study is to determine whether existing models can be directly applied to a new study area without labeled data. If existing models can accurately predict new data without additional training, there would be no need to develop new DL models for seismic data in this region, significantly reducing development time. This would be most beneficial to research regions where human picked seismic data are not available or not feasible.

Another objective of this study is to provide a more comprehensive understanding of existing models and improve their applicability and performance on new data through finding their optimal configurations. Our experiments will examine the following aspects:

1. The effect of different data settings on prediction accuracy, including:

Table 4.2: Different Data Settings in Direct Application Tests

Setting Name	Band-Pass Filtered	Detrended	Resampled to 20 Hz	Resampled to 100 Hz	Denoised
Raw Data	✗	✗	✗	✗	✗
Preprocessed Data	✓	✓	✗	✗	✗
Resampled 20 Hz	✓	✓	✓	✗	✗
Resampled 100 Hz	✓	✓	✗	✓	✗
Denoised	✓	✓	✓	✓	✓

   (a) The need for resampling prior to model prediction;

   (b) The impact of preprocessing (detrending, filtering) on prediction accuracy;

   (c) The potential improvement of denoising methods (e.g., DL denoiser (Zhu et al., 2019));

2. The prediction accuracy of different seismic phase types (P, Pn, Pg, S, Sn, Sg);

3. The "picking" mode of predictions.

The experimental design is similar to the cross-domain comparison tests in SeisBench (Münchmeyer et al., 2022). We use the original PhaseNet, EQTransformer, and ETHZ models from SeisBench as three different pre-trained models, and the full Mackenzie dataset as the test set. In each model, we test five different settings applied to the test set to evaluate their potential impact on model performance. These settings, listed in Table 4.2, are the most common data processing steps employed prior to using these DL pickers. We also compare two picking functions based on the best performing model. The picking functions are: 1) picking the time when the probability value exceeds the default threshold (referred to as the "start_time" mode in SeisBench), and 2) picking the time when the probability curve reaches its peak (referred to as the "peak_time" mode in SeisBench). The "start_time" mode is the most common approach and is typically implemented using the "trigger_onset" module from ObsPy. In contrast, the original PhaseNet and EQTransformer used the "peak_time" mode (see Figure 4.2).

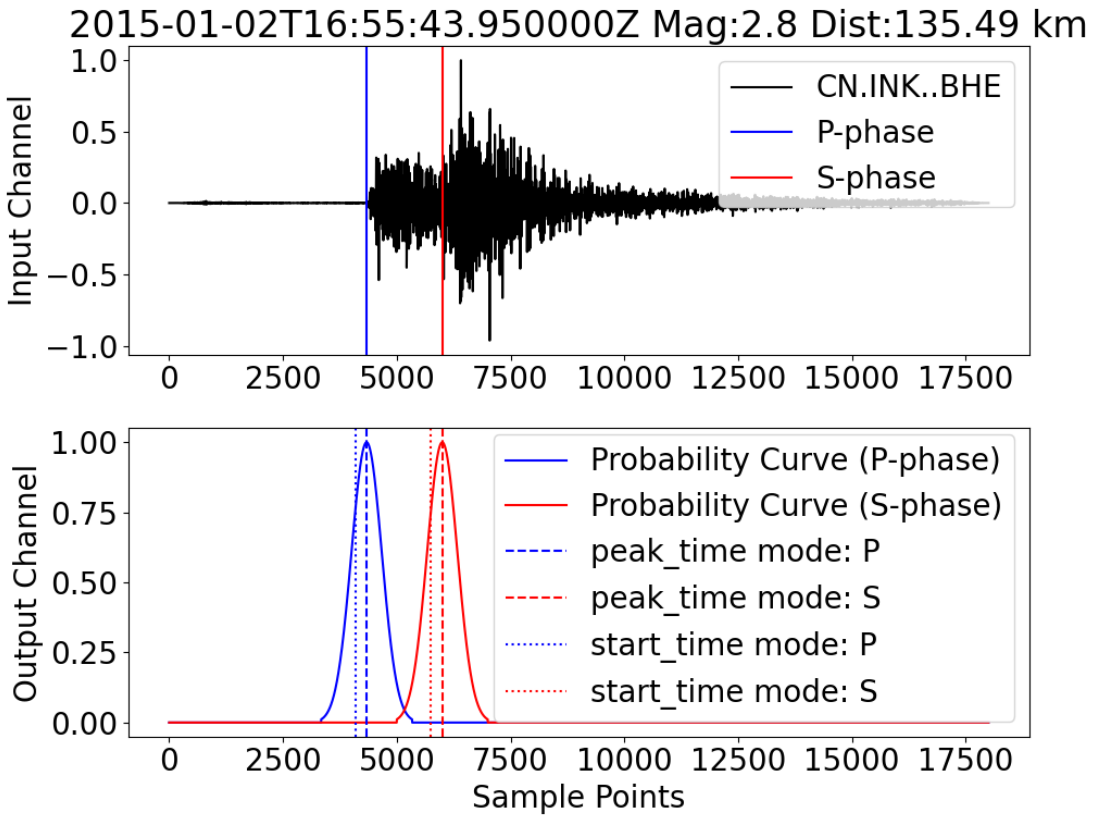


Figure 4.2: **Comparison of the impact of different picking functions on model performance.** This figure explains the difference in how the two existing "picking" functions work on the probability curves to predict arrival times. The two picking functions are: 1) the **start_time** mode, which picks the time when the probability value exceeds the default threshold; and 2) the **peak_time** mode, which picks the time when the probability curve reaches its peak.

Figure 4.3 shows the phase-picking results of the three different models on the Mackenzie dataset and the effect of the various data settings. Overall, the predictions of the three models are significantly less accurate than those on the benchmark dataset on which the models were trained, highlighting the challenges associated with applying pre-trained models to new data. Below, we highlight our main results and findings.

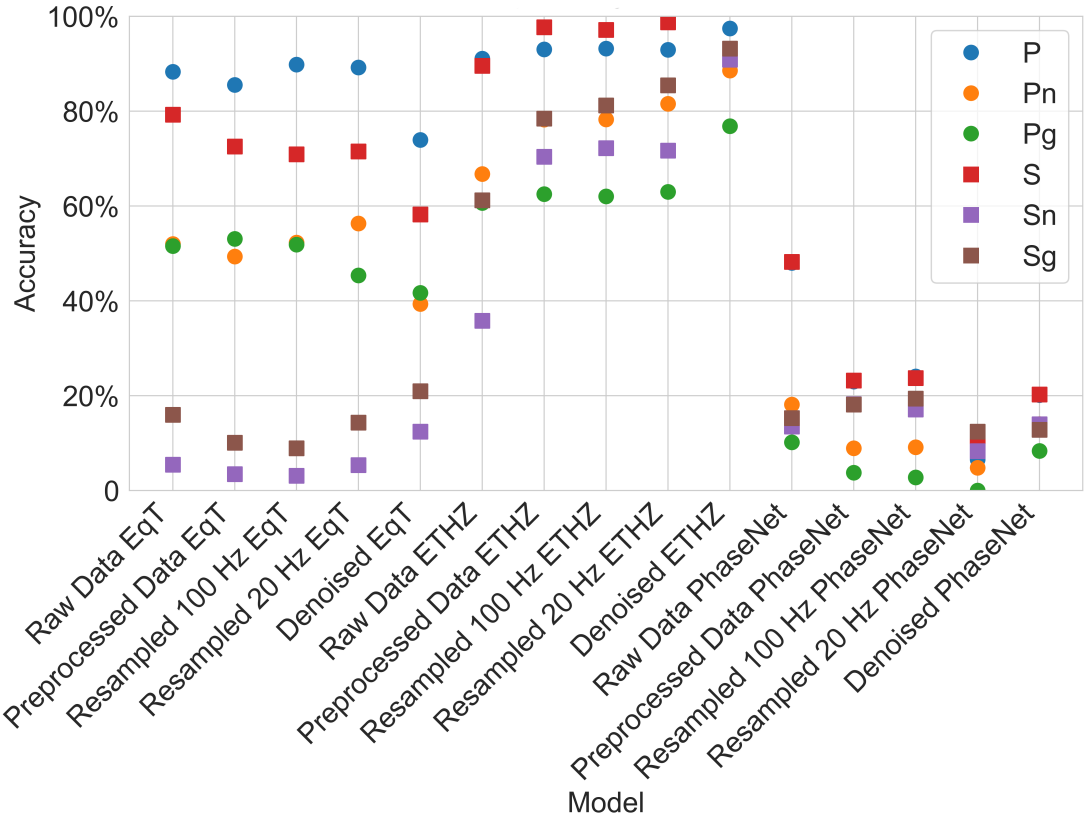


Figure 4.3: **Phase identification results in Mackenzie, Yukon.** Each scatter point represents the accuracy of a specific phase type in a particular model. For each model, we provide five distinct training settings. Note that ETHZ model is a transfer-learning model of original EqT model.

4.4.1 Impact of Data Preprocessing and Standardization

The results presented in Figure 4.3 indicate that data pre-processing, noise removal, and other standardization methods have a limited impact on improving the accuracy of predicting new data, and may even decrease the accuracy of certain phases. The Mackenzie dataset used in the test contains primarily 40 Hz waveform data, with a few 100 Hz or 200 Hz waveform data, and all of the data are raw and unprocessed. In contrast, both the original ETHZ and STEAD training datasets have been resampled to 100 Hz and pre-processed, i.e. detrended, band-pass filtered, etc. (Woollam et al., 2022). From the comparison of

prediction accuracy for each model with different data settings shown in Figure 4.3, only in the ETHZ model do preprocessing, resampling, and noise reduction significantly improve the accuracy of the Pn and Pg phases. In other experiments, there is little difference in prediction performance between raw data and other data settings. Furthermore, test results for the PhaseNet model show that excessive data processing methods can reduce the accuracy of P and S by 20%.

4.4.2 Influence of Picking Method

One aspect that is often overlooked when applying DL models to new data concerns the selection of the "picking" method for predictions. The actual output of the DL model is a probability curve representing the phase arrival at each timestamp. It is common practice to extract the onset time using the trigger_on function described in the Methods section, i.e., the time when the probability curve reaches a preset threshold. Our test results show that using this start_time mode to predict results usually leads to earlier arrival times than manually picked ones, while the peak_time mode used in the original PhaseNet and EQTransformer studies does not have this issue (Figure 4.4). This is because during the training step, DL models use a Gaussian function to create the output channel, with the peak value located at the human-reviewed onset time. Consequently, in the deployment stage, the probability curve predicted by the DL model also resembles a Gaussian function, with the position of the peak value being closer to the true arrival time. Based on the test results, we recommend using the peak_time mode to predict the pick times in practice.

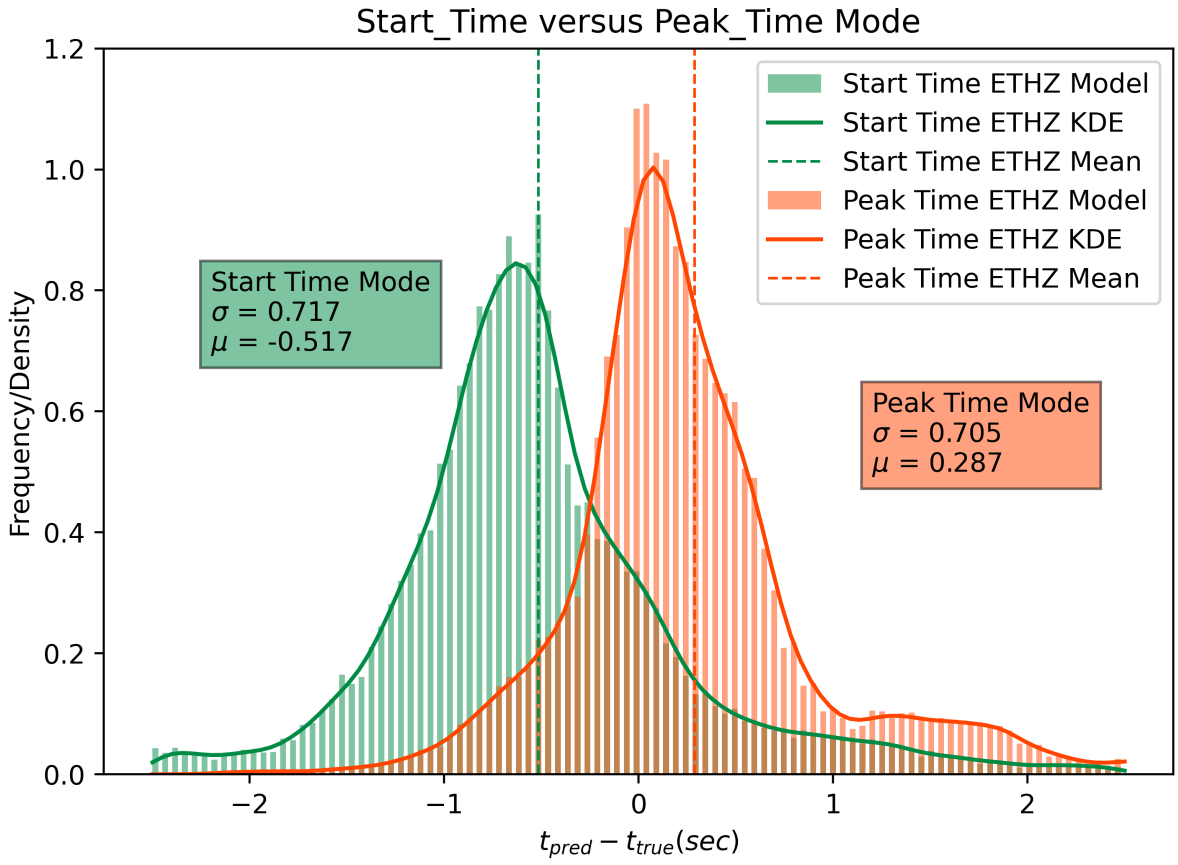


Figure 4.4: **Comparison of pick residuals using different picking strategies.** The green color histogram shows the distribution of residuals based on trigger-generated pick times, while the orange color histogram shows the distribution based on the time of the highest probability. These histograms provide a visual representation of the differences between the predicted pick times and the human-reviewed onset times for each strategy.

4.5 Experiments on Training Size

To examine the influence of training size and assess the necessity of using large-scale benchmark datasets for training new models, we apply varying sizes of the ETHZ dataset to train PhaseNet and EQTransformer architectures. We use the same ETHZ test set provided by SeisBench to ensure the performance is reliable (Münchmeyer et al., 2022).

All experimental conditions, except for the training set size, were kept consistent with the default training parameters in SeisBench. For each training sample size, the subsets maintain a strict inclusion relation (i.e., the current smaller subset was a strict subset of the next larger set), as the exact waveforms in each subset are randomly selected from the next larger set. Our data collection strategy aims to mimic the real annual earthquake catalogue, i.e., each training dataset contains more small events than large ones. To complete the experiments with limited computing power and time, we use a learning rate of 0.01, a default number of epochs of 100, and stop the training if the loss function does not improve after 10 epochs. To ensure the reliability of our analysis, we rigorously maintain a clear separation between the training and test sets.

For each training size examined, we calculate the receiver operating characteristic (ROC) curve, which shows the true positive rate versus the false positive rate. The ROC curve represents how well a model can distinguish between different groups (i.e., earthquakes and noise). An ideal ROC curve rapidly reaches a true positive rate of one, after which point it no longer changes. To quantify the differences in ROC curves for various models, we calculate the area under the ROC curve (AUC), which for an ideal case is equal to one, and is approximately 0.5 for poor model predictions (i.e., equal number of true and false positives).

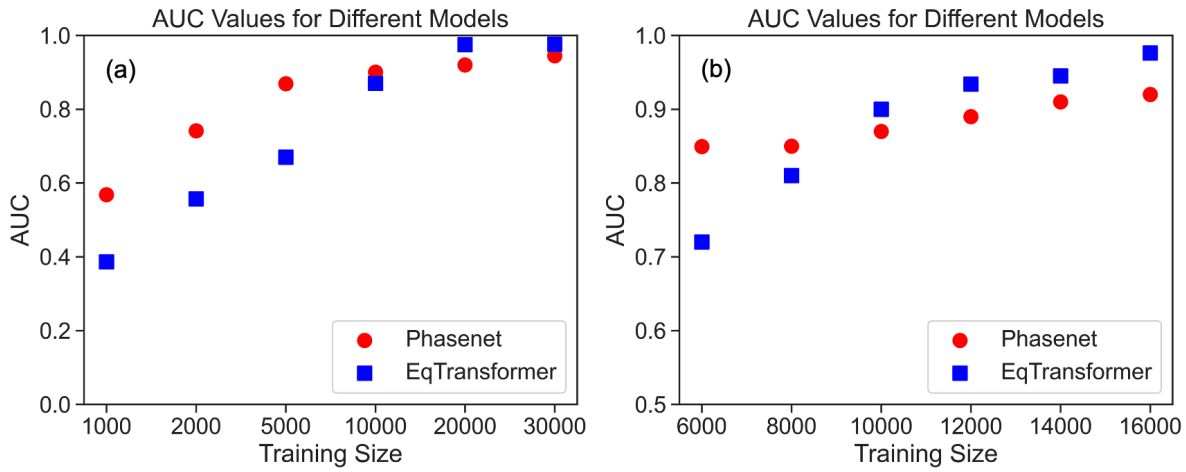


Figure 4.5: **Performance of the 2 models at different training sizes, as measured by the area under the receiver operating characteristic curve (AUC) value.** The AUC value provides an overall measure of the model's ability to discriminate between positive (earthquake) and negative (non-earthquake) samples. The closer the AUC value is to 1.0, the better the model's performance. The results indicate that the PhaseNet model achieved convergence after 5000 samples, as evidenced by the stable AUC values over larger training sizes.

We perform two sets of experiments. In the first set, we evaluate the phase picking performance of each model at increasing scales, i.e., $1000, 2000, 5000, 10000, 20000, 30000$. The results in Figure 4.5a show that when the size of the training set reaches approximately 10,000 waveform data, both DL models converge and achieve the same accuracy as training with the full benchmark data set that involves a much longer computation time. In addition, PhaseNet converges faster than EQTransformer. This is likely because the trainable parameters of the PhaseNet network structure are only one-tenth of those of EQTransformer, requiring fewer training samples for completion.

In the second set of training experiments, we zoom in around the training size of 10,000 and use equal intervals (2000 training samples) ranging from 6000 to 16000 samples to determine the minimum training size required for a model to converge to satisfactory performance (i.e., when accuracy in the test set reaches 90%). The test results show that both models can be successfully trained with a sample size of 12,000 for this particular dataset. Further increasing the amount of data only slightly improves the model performance. We further note that, once this threshold is reached, the EQTransformer model performs better

than the PhaseNet model for this dataset.

4.6 Discussion and Recommendations

4.6.1 Applying a Dataset to Existing Models

The EQTransformer trained on the ETHZ dataset achieved the highest overall accuracy of the three models examined here when applied to the Mackenzie dataset. Because the ETHZ dataset is significantly smaller than the STEAD dataset used to train the other two models, this result suggests that the size of the training data is not the sole determinant of prediction accuracy. Instead, the similarity between the distributions of the training dataset and the new target dataset becomes more critical when the number of training waveforms exceeds a certain minimum threshold. While the ETHZ and Mackenzie datasets sample regions that are completely independent as they are located on different continents, they are both located in mountainous regions within a large orogenic setting. In contrast, the STEAD dataset, which is a global dataset, has a very different data distribution because it samples a variety of tectonic settings.

Interestingly, when trained on the same global STEAD dataset, the P-wave prediction results of EQTransformer are far superior to those of PhaseNet (although both models have poor predictions for other phases). This indicates that the EQTransformer model may have a superior ability to capture certain data characteristics, as it has a more stable and effective performance in picking the first arrival. The test results from both the ETHZ and Mackenzie datasets suggest that the EQTransformer model performs better than PhaseNet when the stations are located in mountainous regions. Therefore, we recommend the use of the EQTransformer model in scenarios where the seismic data are collected from stations located in such regions. However, this recommendation is based on the specific context of these tests, and further research may be needed to confirm this finding in other geographic or tectonic settings.

4.6.2 Understanding Model Failures

We turn our attention to exploring the specific factors of the datasets that can influence prediction results, including earthquake focal depth, magnitude, event-station distance and azimuth. Our goal is to identify the factors that should be emphasized in future dataset preparation to improve model performance. To do this, we use the ETHZ model with the

optimal data setting (called resampled to 100 Hz in the previous chapter, i.e. detrended, resampled to 100 Hz, and bandpass filtered to 3- 20 Hz) and categorize predictions as "good picks" if they have an absolute difference of less than 2.5 seconds when compared to manually labeled data, and as "bad picks" if no results are predicted or the prediction error exceeds 2.5 seconds. If the distribution of predicted good and bad picks is different for a particular factor, this indicates its strong influence in the prediction accuracy.

Our observations highlight that event-station distance is the only factor influencing the model's prediction accuracy (Figure 4.6). Specifically, when the event-station distance exceeds 2 degrees, or $\sim$200 km, the incidence of bad picks increases significantly, despite the abundance of training data in the 200-400 km range within the training dataset (Figure 4.6c). Consequently, future dataset preparation should consider including more labeled data from this event-station distance range to better capture the waveform characteristics. In contrast, we did not observe significant differences in the distribution of good and bad picks associated with the other three potential factors, namely the azimuth, magnitude, and focal depth. For example, the models did not provide better predictions for events that humans can more easily detect, such as those with high magnitude or shallow focal depth.

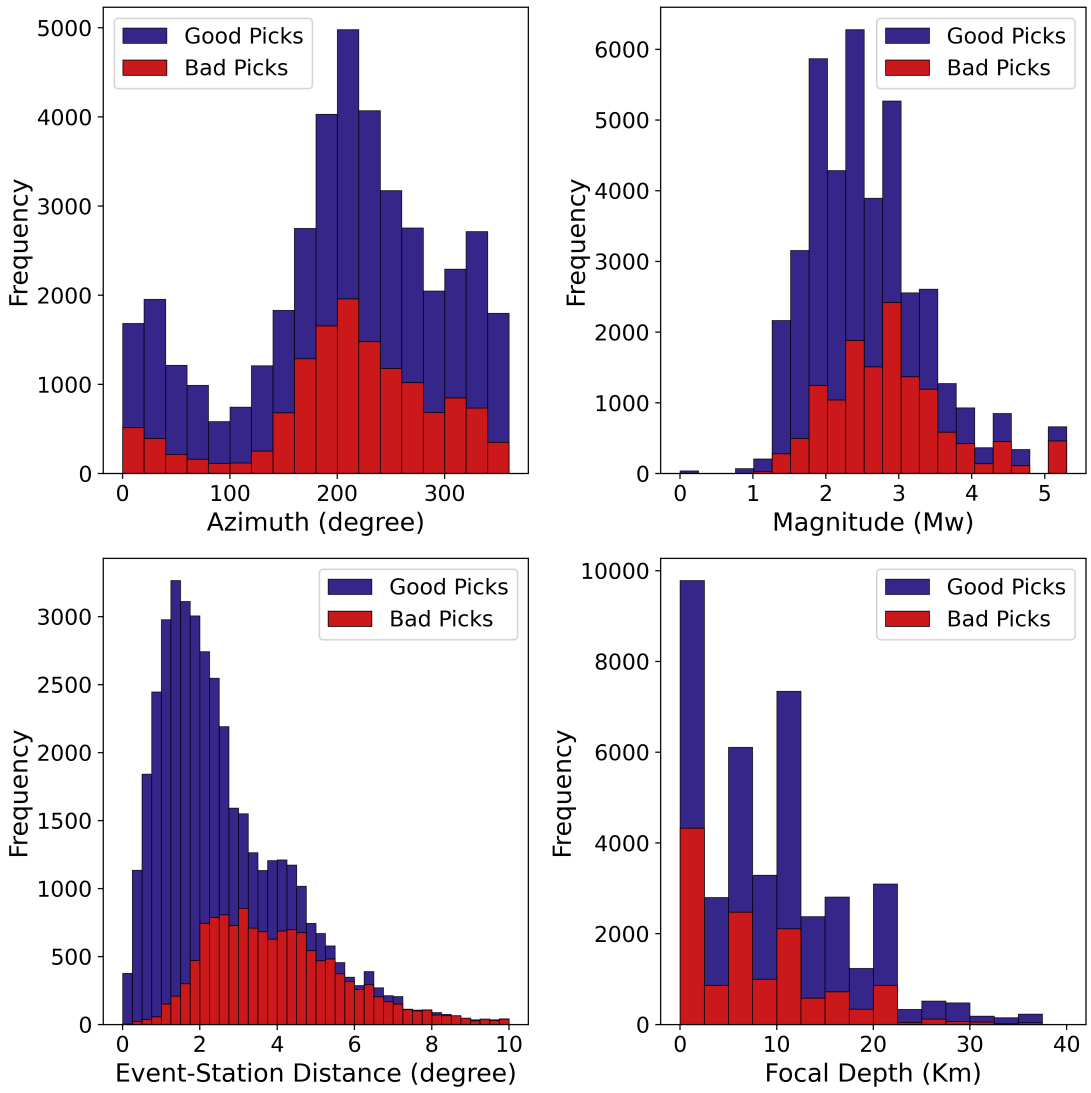


Figure 4.6: **Distribution of model predictions separated by catalogue parameters.** Good picks refer to cases where the predicted arrival time is within 2.5 seconds of the human-reviewed onset time. Bad picks represent missing picks or predictions with a difference greater than 2.5 seconds from the human-reviewed arrivals. A similar distribution pattern between good and bad picks suggests that the corresponding factor has little influence on model prediction. Our tests indicate that only the event-station distance significantly affects the model's performance. Specifically, we observe a decrease in prediction accuracy when the distance exceeds 2 degrees, or ∼200 km.

4.6.3 Training a New Model

In practical projects, it is necessary to consider the balance between the number of training samples, training time, and computational cost. In research areas where manually annotated samples are scarce, we recommend using the PhaseNet model because it requires fewer training samples. With only 5000 samples, PhaseNet can train a model with an accuracy rate of 85%, which is useful for solving practical problems. However, when more training data are available, EQTransformer will provide a better prediction accuracy than PhaseNet.

To substantiate these claims, we train a new model on the Mackenzie dataset using the PhaseNet and EQTransformer model structures to improve the recognition accuracy of Pg, Pn, Sg, and Sn phases. To validate the effectiveness of the training size suggested by our results, we train the two models using two different sample sizes of 6,000 and 12,000. We assess model performance by calculating the ROC and the precision-recall curves. We also visually assess performance by plotting the pick residuals (Fig. 4.7). The recall curve shows how good a model is at catching all the events. For example, a high recall means that the model is good at catching most earthquakes, i.e. it misses few events. We also visually assess performance by plotting the pick residuals (Fig. 4.8)

Figures 4.7 and 4.8 show that the model training results were consistent with our expectations. PhaseNet converges after training with 6,000 samples and shows further performance improvement at 12,000 samples. EQTransformer, on the other hand, shows better prediction accuracy than PhaseNet at 12,000 samples, although it does not fully converge at 6,000 samples (see Figure 4.8). Once trained with 12,000 samples, the EQTransformer model has higher accuracy and lower residuals than the corresponding PhaseNet model.

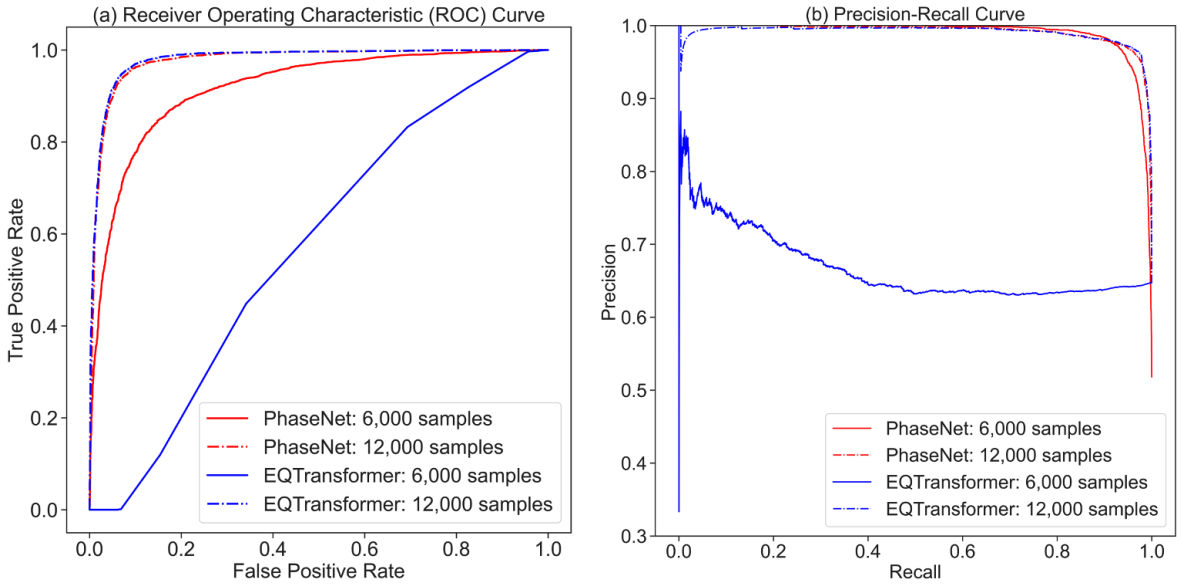


Figure 4.7: **Training performance of different models.** a) Receiver operating characteristic (ROC) curves for the 2 models with different training sizes. b) Precision versus recall curves. EQTransformer trained with 12,000 samples yields the best performance, while EQTransformer trained with 6,000 samples would not be considered as a successfully trained model .

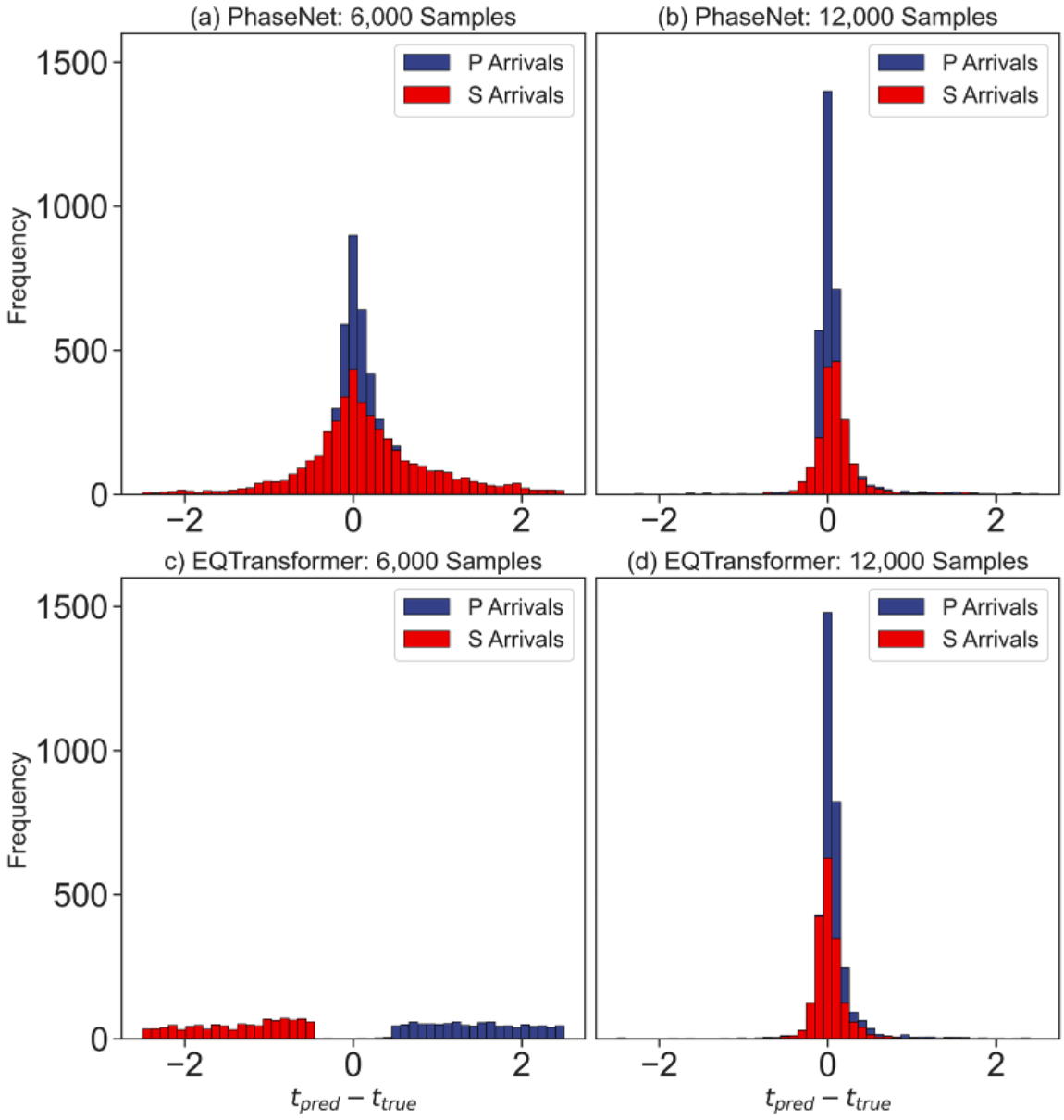


Figure 4.8: **Comparison of pick residuals using different training sizes in the Mackenzie dataset.** Pick residuals are defined as the differences between the predicted arrival times and the human-reviewed onset times. a-b) Distribution of residuals for the PhaseNet model trained with 6,000 and 12,000 samples, respectively. c-d) Distribution of residuals for the EQTransformer model trained with 6,000 and 12,000 samples, respectively. The low frequency residual distribution in c) implies that this model is not fully trained.

4.7 Conclusion

In this study, we investigated the performance of two prominent deep learning models, PhaseNet and EQTransformer, for predicting seismic phase arrival times. We also investigated the factors that influence model performance, including the impact of different data manipulations and picking strategies on model prediction accuracy. Our analysis showed that the various pre-processing steps do not significantly influence the results. However, the picking method is a critical aspect that influences picking accuracy. We found that the peak_time mode used in the original PhaseNet and EQTransformer studies provides a more accurate prediction of arrival times than the start_time mode obtained using the trigger_on function. This is primarily due to the Gaussian probability function used by the deep learning models during the training phase, which aligns the peak value with the human-reviewed onset time.

Additionally, we investigated the influence of training set size on model performance. Our results on the ETHZ benchmark dataset showed that both PhaseNet and EQTransformer models can converge and achieve satisfactory performance with a training set size of 12,000 waveforms. PhaseNet was found to converge faster than EQTransformer, requiring fewer samples for training. This is likely due to the smaller number of trainable parameters in the PhaseNet network structure. In practical applications where manually annotated samples are limited, we recommend using the PhaseNet model as it can at least achieve an accuracy rate of 85% with only 5,000 samples. We further applied the insights from our analysis to train new models on the Mackenzie dataset using both PhaseNet and EQTransformer structures. The results obtained from training with different sample sizes were consistent with our expectations, confirming the findings of our study.

Finally, this research provides valuable practical guidelines for DL pickers, sheds light on the factors that influence their accuracy, and suggests optimal strategies for further applications. By understanding the importance of selection methods and identifying the most effective training set size, we can improve the performance of these models in seismological research and real-world applications, ultimately contributing to a better understanding of seismic catalogues and derived data products.

4.8 Declaration of Competing Interests

The authors acknowledge there are no conflicts of interest recorded.

4.9 Data and Resources

The Python scripts, deep learning models, and the new training dataset for the Mackenzie Mountains used in this work can be accessed from the GitHub repository "How-many-samples-do-we-need" at https://github.com/maihao14/How-many-samples-do-we-need. The repository serves as a resource for researchers aiming to replicate or extend this study.

The (Canadian) National Earthquake Database may be found at National Earthquake Database (last accessed June 2023). At this time, the online database is searchable from 1985 to present and returns only the solutions. Solutions and phase information for older earthquakes may be obtained by contacting nrcan.earthquakeinfo-infoseisme.rncan@canada.ca. Waveform data and/or station metadata from the CNSN may be retrieved via FDSN dataselect webservice here, via FDSN station webservice here, or via Station Book here.

4.10 Acknowledgments

This research was funded by a Discovery Grant (RGPIN-2018-03752) from the Natural Science and Engineering Research Council of Canada (PA). The authors thank the Canadian Hazard Information Service for access to earthquake catalogues and seismic waveforms.

Chapter 5

Thesis Conclusion

5.1 Conclusion

In this thesis, I have undertaken a comprehensive exploration of the application of deep learning in seismology, ranging from the creation and management of seismic datasets, to the customization of phase pickers, to understanding how seismic datasets influence the performance of deep learning models. Since the main conclusions and contributions are already summarized within each chapter, I focus here on limitations and potential opportunities for future research that further develop the ideas presented in this thesis.

The first part of this thesis introduced QuakeLabeler, a tool I developed to address the need for seismologists and developers who may not be AI specialists to easily, quickly, and independently build and visualize reproducible training datasets. QuakeLabeler provides a versatile platform for designing datasets targeting different spatial scales of seismicity and magnitude. It can be used in various studies, including noise attenuation, earthquake detection, phase picking, earthquake location, and magnitude estimation. The development of such tools is crucial in the field of seismology, where the generation and preparation of datasets are essential steps in deep learning applications.

Currently, QuakeLabeler supports dataset generation for PhaseNet, EQTransformer, and SeisBench, which are the most popular published deep learning models/platforms. QuakeLabeler can be implemented on a local machine or in an online Python environment, such as Google Colab (Bisong and Bisong, 2019). By default, QuakeLabeler collects only human reviewed arrival catalogs from ISC. Other organizations could be accessible by changing code settings in the backend, as well as loading catalogues from local drive. In

the future, QuakeLabeler should be planted online, i.e. on its own website, for more flexible use. Another potential application is that QuakeLabeler has the chance to be equipped with interactive annotation function, labeling the events when users simultaneously click on the seismogram. This feature has been implemented in many commercial companies that provide image annotation service (Torralba et al., 2010). Finally, QuakeLabeler's current focus is on earthquake detection, similar to many published benchmark datasets. As deep learning methods evolve, other labeling goals such as magnitude estimation, location prediction, etc. should be fully developed.

In the second part of this thesis, I introduced Blockly Earthquake Transformer (BET), an easy-to-use platform that I designed to appeal to a broader range of users, including those without machine learning or coding experience. BET allows researchers to explore deep learning pickers without having to worry about in-house computing power and other technical concerns. It is a valuable tool in improving model performance and lowering the magnitude of completeness of existing seismic catalogues. BET's versatility allows for the customization of different types of deep learning pickers to accommodate various datasets formats and identify a diverse range of seismic signals.

Like QuakeLabeler's current development environment, BET is mainly implemented locally. Rendering it online would alleviate installation concerns for beginner users and allow them to concentrate on designing new deep learning pickers. However, unlike QuakeLabeler, BET is a deep learning training platform, which means that if users want to create very deep and complex neural network architecture, they will need considerable computational resources during training (if they want to train new models, not transfer learning from a pre-trained model). Therefore, this project may need funding for its long-term support, as GPU rental is still expensive (Carneiro et al., 2018).

The third and final part of this thesis explored the performance of two prominent deep learning models, PhaseNet and EQTransformer, for predicting seismic phase arrival times of seismic phases for a new dataset that was not used for training. This study provided valuable insights into the factors that influence model performance, including the impact of different picking strategies and the optimal training set size for effective model training. The results of this study provide practical guidelines for deep learning pickers, shedding light on the factors that influence their accuracy and suggesting optimal strategies for further applications.

The third research chapter is just the beginning of this kind of investigation. Many potential factors that could influence deep learning methods are still unclear (Yip et al., 2020). There is no rule of thumb on how to find a perfect balance between high accuracy and high generalization ability (Neyshabur et al., 2017). Or even more challenging, how

to balance the ability to detect small events and allowing artifacts (false positive rate). When researchers are engaged in proposing novel deep learning algorithms, they should also consider the likelihood that these DL pickers can be robustly applied to routine seismic data processing (Zhang et al., 2021a).

In conclusion, the research conducted in this thesis contributes significantly to the field of seismology by providing practical tools and insights for the application of deep learning. By understanding the importance of selection methods and identifying the most effective training set size, we can improve the performance of these models in seismological research and real-world applications, ultimately contributing to a better understanding of seismic events and benefiting future research. The open-source nature of the tools developed in this thesis, such as QuakeLabeler and BET, also encourages community involvement and further development, promising continued growth and innovation in the field of seismology.

5.2 Final thoughts

Personally, I believe that deep learning will be the next generation of automation methods, eventually replacing traditional picking methods and maybe even changing the whole workflow of seismic data processing. However, the road is not even, as more resources should be invested and new collaborations should be established to put deep learning models into practice. If these deep learning models can be implemented in the routine workflow, considerable time of analysts and seismologists will be saved to give them the opportunity to focus on exploring new understanding of earthquakes.

The development of benchmark datasets, open source toolboxes, and sharing of educational tutorials should be encouraged. This will accelerate the progress of deep learning in our seismological community. From my own experience, these developments cost a lot of time and no profit for the developers. However, if there are few tools for seismological researchers, the development cycle of new deep learning models will be much longer, and valuable research time would be wasted on repetitive preparation work.

In the near future, AI-assisted research should be considered an important topic in seismology. It's very likely that interactive AI assistants will be widely used in almost every scenario in the future. Consider Microsoft's recent announcement that its AI assistant Copilot will be launched in Windows 11 and cooperate with users under all Office software, even the whole operating system (Dakhel et al., 2023). Or even more specifically in Earth science fields: an AI-powered geographic information system (GIS) called Autonomous GIS has been introduced to the GIScience community, making spatial analysis easier,

faster and more accessible to a broader audience (Li and Ning, 2023). The development of an AI assistant could be of great help to seismic analysts, e.g. in managing the day-to-day operations of large projects, ensuring optimal quality control, identifying problems and defining ideal solutions, and so on. However, this requires much more investment of human intelligence and computing power. I hope this will attract more attention and become a reality.

References

S. Agatonovic-Kustrin and R. Beresford. Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. *Journal of pharmaceutical and biomedical analysis*, 22(5):717–727, 2000.

T. Akazawa. A technique for automatic detection of onset time of p-and s-phases in strong motion records. In *Proc. of the 13th World Conf. on Earthquake Engineering*. Vancouver, Canada, 2004.

S. S. Alahmari, D. B. Goldgof, P. R. Mouton, and L. O. Hall. Challenges for the repeatability of deep learning models. *IEEE Access*, 8:211860–211868, 2020.

R. M. Allen, P. Gasparini, O. Kamigaichi, and M. Bose. The status of earthquake early warning around the world: An introductory overview. *Seismological Research Letters*, 80(5):682–693, 2009.

L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8: 1–74, 2021.

S.-i. Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5 (4-5):185–196, 1993.

AWS. Amazon sagemaker ground truth, 2021. URL https://aws.amazon.com/sagemaker/groundtruth/.

C. Banbury, V. J. Reddi, P. Torelli, J. Holleman, N. Jeffries, C. Kiraly, P. Montino, D. Kanter, S. Ahmed, D. Pau, et al. Mlperf tiny benchmark. *arXiv preprint arXiv:2106.07597*, 2021.

Z. Bao, J. Zhao, P. Huang, S. Yong, and X. Wang. A deep learning-based electromagnetic signal for earthquake magnitude prediction. *Sensors*, 21(13):4434, 2021.

W. D. Barnhart and C. Wolfe. An overview of anss. In *AGU Fall Meeting Abstracts*, volume 2022, pages S22D–0188, 2022.

G. C. Beroza, M. Segou, and S. Mostafa Mousavi. Machine learning and earthquake forecasting—next steps. *Nature communications*, 12(1):1–3, 2021.

M. Beyreuther, R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann. Obspy: A python toolbox for seismology. *Seismological Research Letters*, 81(3):530–533, 2010.

E. Bisong. *Building machine learning and deep learning models on Google cloud platform*. Apress, Berkeley, CA, 2019.

E. Bisong and E. Bisong. Google colaboratory. *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners*, pages 59–64, 2019.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

T. Bornstein, D. Lange, J. Münchmeyer, J. Woollam, A. Rietbrock, G. Barcheck, I. Grevemeyer, and F. Tilmann. Pickblue: Seismic phase picking for ocean bottom seismometers with deep learning. *arXiv preprint arXiv:2304.06635*, 2023.

L. Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

E. E. Brodsky. The importance of studying small earthquakes. *Science*, 364(6442):736–737, 2019.

J. Brownlee. *Better deep learning: train faster, reduce overfitting, and make better predictions*. Machine Learning Mastery, 2018.

J. Brownlee. *Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python*. Machine Learning Mastery, 2020.

S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

N. M. Campbell, M. Leon-Corwin, L. A. Ritchie, and J. Vickery. Human-induced seismicity: Risk perceptions in the state of oklahoma. *The Extractive Industries and Society*, 7(1): 119–126, 2020.

T. Carneiro, R. V. M. Da Nóbrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, and P. P. Reboucas Filho. Performance analysis of google colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, 6:61677–61685, 2018.

C. Chai, M. Maceira, H. J. Santos-Villalobos, S. V. Venkatakrishnan, M. Schoenball, W. Zhu, G. C. Beroza, C. Thurber, and E. C. Team. Using a deep neural network and transfer learning to bridge scales for seismic phase picking. *Geophysical Research Letters*, 47(16):e2020GL088651, 2020.

Z. Chen, Y. Cao, Y. Liu, H. Wang, T. Xie, and X. Liu. A comprehensive study on challenges in deploying deep learning based software. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 750–762, 2020.

T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman. Project adam: Building an efficient and scalable deep learning training system. In *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*, pages 571–582, 2014.

J. Cho, K. Lee, E. Shin, G. Choy, and S. Do. How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? *arXiv preprint arXiv:1511.06348*, 2015.

P. Christoffersen and K. Jacobs. The importance of the loss function in option valuation. *Journal of Financial Economics*, 72(2):291–318, 2004.

X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang. Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 international conference on management of data*, pages 2201–2206, 2016.

H. Cole and W. L. Yeck. Global earthquake machine learning dataset: Machine learning asset aggregation of the pde (mlaapde), 2022. URL https://www.sciencebase.gov/catalog/item/6127b30fd34e40dd9c05094c.

S. Crane, C. Perry, D. Motazedian, and J. Adams. Synthetic ground motions at quebec city from charlevoix earthquakes using empirical green's functions. In *12th Canadian Conference on Earthquake Engineering*, Quebec City, 2019.

S. Crane, N. Bell, H. Seyerd, and J. Adams. Assessing the performance of earthquake early warning in eastern canada using historical earthquakes. In *AGU Fall Meeting Abstracts*, volume 2021, pages S15A–0228, 2021.

G. Cremen and C. Galasso. Earthquake early warning: Recent advances and perspectives. *Earth-Science Reviews*, 205:103184, 2020.

E. Dahlquist, M. Rahman, J. Skvaril, and K. Kyprianidis. Ai overview: Methods and structures. *AI and Learning Systems: Industrial Applications and Future Directions*, page 3, 2021.

A. M. Dakhel, V. Majdinasab, A. Nikanjam, F. Khomh, M. C. Desmarais, and Z. M. J. Jiang. Github copilot ai pair programmer: Asset or liability? *Journal of Systems and Software*, 203:111734, 2023.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

D. Di Giacomo, J. Harris, and D. A. Storchak. Complementing regional moment magnitudes to gcmt: A perspective from the rebuilt international seismological centre bulletin. *Earth System Science Data*, 13(5):1957–1985, 2021.

K. Dimililer, H. Dindar, and F. Al-Turjman. Deep learning, machine learning and internet of things in geophysical engineering applications: An overview. *Microprocessors and Microsystems*, 80:103613, 2021.

I. DMC. Data services products: Eventplots maps, record sections & other plots for m6.0+ events. https://doi.org/10.17611/DP/EP.1, 2011.

S. Doocy, A. Daniels, C. Packer, A. Dick, and T. D. Kirsch. The human impact of earthquakes: a historical review of events 1980-2009 and systematic literature review. *PLoS currents*, 5, 2013.

S. Dryhurst, F. Mulder, I. Dallo, J. R. Kerr, S. K. McBride, L. Fallou, and J. S. Becker. Fighting misinformation in seismology: Expert opinion on earthquake facts vs. fiction. *Frontiers in Earth Science*, 10:937055, 2022.

G. Du, X. Cao, J. Liang, X. Chen, and Y. Zhan. Medical image segmentation based on u-net: A review. *Journal of Imaging Science and Technology*, 64:1–12, 2020.

N. D'Angelo, G. Adelfio, A. D'Alessandro, and M. Chiodi. A fast and efficient picking algorithm for earthquake early warning application based on the variance piecewise constant models. In *Computational Science and Its Applications–ICCSA 2020: 20th International Conference, Cagliari, Italy, July 1–4, 2020, Proceedings, Part VII 20*, pages 903–913. Springer, 2020.

T. Eelbode, P. Sinonquel, F. Maes, and R. Bisschops. Pitfalls in training and validation of deep learning systems. *Best Practice & Research Clinical Gastroenterology*, 52:101712, 2021.

A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

M. Firat. How chat gpt can transform autodidactic experiences and open education. *Department of Distance Education, Open Education Faculty, Anadolu Unive*, 2023.

L. Floridi and M. Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.

W. Gao, Y. Wang, and S. Yu. An interactive system based on the iasp91 earth model for earthquake data processing. *Applied Sciences*, 12(22):11846, 2022.

F. Gatti and D. Clouteau. Towards blending physics-based numerical simulations and seismic databases using generative adversarial network. *Computer Methods in Applied Mechanics and Engineering*, 372:113421, 2020.

A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022.

C. Gershenson. Artificial neural networks for beginners. *arXiv preprint cs/0308031*, 2003.

E. Gibson, W. Li, C. Sudre, L. Fidon, D. I. Shakir, G. Wang, Z. Eaton-Rosen, R. Gray, T. Doel, Y. Hu, et al. Niftynet: a deep-learning platform for medical imaging. *Computer methods and programs in biomedicine*, 158:113–122, 2018.

K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and building materials*, 157:322–330, 2017.

J. Gu, X. Meng, G. Lu, L. Hou, N. Minzhe, X. Liang, L. Yao, R. Huang, W. Zhang, X. Jiang, et al. Wukong: A 100 million large-scale chinese cross-modal pre-training benchmark. *Advances in Neural Information Processing Systems*, 35:26418–26431, 2022.

Q. Guo, S. Chen, X. Xie, L. Ma, Q. Hu, H. Liu, Y. Liu, J. Zhao, and X. Li. An empirical study towards characterizing deep learning development and deployment across different frameworks and platforms. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 810–822. IEEE, 2019.

K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021.

W. Hanka, A. Heinloo, and K.-H. Jaeckel. Networked seismographs: Geofon real-time data distribution, 2000.

J. Havskov and L. Ottemoller. *Routine data processing in earthquake seismology: with sample data, exercises and software*. Springer Science & Business Media, 2010.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

M. Herrmann and W. Marzocchi. Inconsistencies and lurking pitfalls in the magnitude–frequency distribution of high-resolution earthquake catalogs. *Seismological Research Letters*, 92(2A):909–922, 2021.

J. Hestness, N. Ardalani, and G. Diamos. Beyond human-level accuracy: Computational challenges in deep learning. In *Proceedings of the 24th symposium on principles and practice of parallel programming*, pages 1–14, 2019.

D. P. Hill, F. Pollitz, and C. Newhall. Earthquake-volcano interactions. *Physics Today*, 55 (11):41–47, 2002.

E. L. Hill-Yardin, M. R. Hutchinson, R. Laycock, and S. J. Spencer. A chat (gpt) about the future of scientific publishing. *Brain, behavior, and immunity*, pages S0889–1591, 2023.

A. A. Holland. Earthquakes triggered by hydraulic fracturing in south-central oklahoma. *Bulletin of the Seismological Society of America*, 103(3):1784–1792, 2013.

Z. L. Hongli and Z. Han. Comparasion of three micros-eismic first arrival picking methods, based on ratio of short-term average and long-term average, polarization method and aic method. *Global Geology*, 39(03):649–655, 2020.

K. Horak and R. Sablatnig. Deep learning concepts and datasets for image recognition: overview 2019. In *Eleventh international conference on digital image processing (ICDIP 2019)*, volume 11179, pages 484–491. SPIE, 2019.

G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

J. Huang, J. Chai, and S. Cho. Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China*, 14(1):1–24, 2020.

L. Huang, X. Dong, and T. E. Clee. A scalable deep learning platform for identifying geologic features from seismic attributes. *The Leading Edge*, 36(3):249–256, 2017.

S. Ide. Frequent observations of identical onsets of large and small earthquakes. *Nature*, 573(7772):112–116, 2019.

C. Jiang, L. Fang, L. Fan, and B. Li. Comparison of the earthquake detection abilities of phasenet and eqtransformer with the yangbi and maduo earthquakes. *Earthquake Science*, 34(5):425–435, 2021.

C. Jiang et al. A detailed earthquake catalog for banda arc–australian plate collision zone using machine-learning phase picker and an automated workflow. *The Seismic Record*, 2(1):1–10, 2022a.

P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma. A review of yolo algorithm developments. *Procedia Computer Science*, 199:1066–1073, 2022b.

P. Jiao and A. H. Alavi. Artificial intelligence in seismology: Advent, performance and future trends. *Geoscience Frontiers*, 11(3):739–744, 2020.

D. Jozinović, A. Lomax, I. Štajduhar, and A. Michelini. Transfer learning: Improving neural network based prediction of earthquake ground shaking for an area with insufficient training data. *Geophysical Journal International*, 229(1):704–718, 2022.

M. D. Kohler, D. E. Smith, J. Andrews, A. I. Chung, R. Hartog, I. Henson, D. D. Given, R. de Groot, and S. Guiwits. Earthquake early warning shakealert 2.0: Public rollout. *Seismological Research Letters*, 91(3):1763–1775, 2020.

Q. Kong, D. T. Trugman, Z. E. Ross, M. J. Bianco, B. J. Meade, and P. Gerstoft. Machine learning in seismology: Turning data into insights. *Seismological Research Letters*, 90 (1):3–14, 2019.

K. D. Koper. The importance of regional seismic networks in monitoring nuclear test-ban treaties. In *AGU Fall Meeting Abstracts*, volume 2019, pages S14B–05, 2019.

L. Krischer, T. Megies, R. Barsch, M. Beyreuther, T. Lecocq, C. Caudron, and J. Wassermann. Obspy: A bridge for seismology into the scientific python ecosystem. *Computational Science & Discovery*, 8(1):014003, 2015.

M. Krithika alias AnbuDevi and K. Suganthi. Review of semantic segmentation of medical images using modified architectures of unet. *Diagnostics*, 12(12):3064, 2022.

I. Labelbox. Ground truth labeler, 2021. URL https://labelbox.com/.

S. Lapins, B. Goitom, J.-M. Kendall, M. J. Werner, K. V. Cashman, and J. O. Hammond. A little data goes a long way: Automating seismic phase arrival picking at nabro volcano with transfer learning. *Journal of Geophysical Research: Solid Earth*, 126(7): e2021JB021910, 2021.

J. Li, M. Ran, W. Tong, Y. Cao, L. Cai, X. Ou, and T. Yang. Transfer-learning-based svm method for seismic phase picking with insufficient training samples. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

P. Li, X. Rao, J. Blase, Y. Zhang, X. Chu, and C. Zhang. Cleanml: A benchmark for joint data cleaning and machine learning [experiments and analysis]. *arXiv preprint arXiv:1904.09483*, page 75, 2019.

X. Li, X. Shang, A. Morales-Esteban, and Z. Wang. Identifying p phase arrival of weak events: The akaike information criterion picking application based on the empirical mode decomposition. *Computers & Geosciences*, 100:57–66, 2017.

Z. Li and H. Ning. Autonomous gis: the next-generation ai-powered gis. *arXiv preprint arXiv:2305.06453*, 2023.

Z. Li, C. Zhu, Y.-L. Gao, Z.-K. Wang, and J. Wang. Alphago policy network: A dcnn accelerator on fpga. *IEEE Access*, 8:203039–203047, 2020.

W.-Y. Liao, E.-J. Lee, D. Mu, P. Chen, and R.-J. Rau. Arru phase picker: Attention recurrent-residual u-net for picking seismic p-and s-phase arrivals. *Seismological Research Letters*, 2021.

M. Liu et al. Rapid characterization of the july 2019 ridgecrest, california, earthquake sequence from raw seismic data using machine-learning phase picker. *Geophysical Research Letters*, 47(4):e2019GL086189, 2020.

Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, et al. Summary of chatgpt/gpt-4 research and perspective towards the future of large language models. *arXiv preprint arXiv:2304.01852*, 2023.

I. E. Livieris, S. Stavroyiannis, E. Pintelas, and P. Pintelas. A novel validation framework to enhance deep learning models in time-series forecasting. *Neural Computing and Applications*, 32(23):17149–17167, 2020.

Y. Long, J. Lin, B. Li, H. Wang, and Z. Chen. Fast-aic method for automatic first arrivals picking of microseismic event with multitrace energy stacking envelope summation. *IEEE Geoscience and Remote Sensing Letters*, 17(10):1832–1836, 2019.

Y. Ma, D. Yu, T. Wu, and H. Wang. Paddlepaddle: An open-source deep learning platform from industrial practice. *Frontiers of Data and Domputing*, 1(1):105–115, 2019.

Y. Ma, S. Cao, J. W. Rector, and Z. Zhang. Automated arrival-time picking using a pixel-level network. *Geophysics*, 85(5):V415–V423, 2020.

Y. Ma, D. Eaton, N. Igonin, and C. Wang. Machine learning-assisted processing workflow for multi-fiber das microseismic data. *Frontiers in Earth Science*, 11, 2023.

F. Magrini, D. Jozinović, F. Cammarano, A. Michelini, and L. Boschi. Local earthquakes detection: A benchmark dataset of 3-component seismograms built on a global scale. *Artificial Intelligence in Geosciences*, 1:1–10, 2020.

H. Mai and P. Audet. A U-Net for Weak Earthquake Detection. In *AGU Fall Meeting Abstracts*, volume 2020, pages S051–04, December 2020.

H. Mai and P. Audet. Quakelabeler: A fast seismic data set creation and annotation toolbox for ai applications. *Seismological Society of America*, 93(2A):997–1010, 2022.

H. Mai, P. Audet, M. Mousavi, C. Perry, and Q. Zhang. Blockly Earthquake Transformer: A deep learning platform for custom phase picking. *Artificial Intelligence in Geosciences*, in press, 2023.

S. S. Matin and B. Pradhan. Earthquake-induced building-damage mapping using explainable ai (xai). *Sensors*, 21(13):4489, 2021.

MATLAB. Ground truth labeler, 2021a. URL https://www.mathworks.com/help/driving/ref/groundtruthlabeler-app.html.

R. W. McGee. Is chat gpt biased against conservatives? an empirical study. *An Empirical Study (February 15, 2023)*, 2023.

G. C. McLaskey. Earthquake initiation from laboratory observations and implications for foreshocks. *Journal of Geophysical Research: Solid Earth*, 124(12):12882–12904, 2019.

S. T. Michael Paluszek. *MATLAB machine learning toolboxes*, pages 25–41. Apress, 2020.

A. Michelini, S. Cianetti, S. Gaviano, C. Giunchi, D. Jozinović, and V. Lauciani. Instance–the italian seismic dataset for machine learning. *Earth System Science Data*, 13(12): 5509–5544, 2021.

S. Mittal. A survey on optimized implementation of deep learning models on the nvidia jetson platform. *Journal of Systems Architecture*, 97:428–442, 2019.

S. M. Mousavi and G. C. Beroza. Bayesian-deep-learning estimation of earthquake location from single-station observations. *arXiv preprint arXiv:1912.01144*, 2019.

S. M. Mousavi and G. C. Beroza. A machine-learning approach for earthquake magnitude estimation. *Geophysical Research Letters*, 47(1):e2019GL085976, 2020.

S. M. Mousavi and G. C. Beroza. Deep-learning seismology. *Science*, 377(6607):eabm4470, 2022a.

S. M. Mousavi and G. C. Beroza. Machine learning in earthquake seismology. *Annual Review of Earth and Planetary Sciences*, 51:2023, 2022b.

S. M. Mousavi and G. C. Beroza. Machine learning in earthquake seismology. *Annual Review of Earth and Planetary Sciences*, 51, 2023.

S. M. Mousavi, Y. Sheng, W. Zhu, and G. C. Beroza. Stanford earthquake dataset (stead): A global data set of seismic signals for ai. *IEEE Access*, 7:179464–179476, 2019a.

S. M. Mousavi, W. Zhu, Y. Sheng, and G. C. Beroza. Cred: A deep residual network of convolutional and recurrent units for earthquake signal detection. *Scientific reports*, 9 (1):1–14, 2019b.

S. M. Mousavi, W. L. Ellsworth, W. Zhu, L. Y. Chuang, and G. C. Beroza. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature communications*, 11(1):1–12, 2020.

R. Mu and X. Zeng. A review of deep learning research. *KSII Transactions on Internet and Information Systems (TIIS)*, 13(4):1738–1764, 2019.

S. M. Mudambi and D. Schuff. Research note: What makes a helpful online review? a study of customer reviews on amazon. com. *MIS quarterly*, pages 185–200, 2010.

J. Münchmeyer, J. Woollam, A. Rietbrock, F. Tilmann, D. Lange, T. Bornstein, T. Diehl, C. Giunchi, F. Haslinger, D. Jozinović, et al. Which picker fits my data? a quantitative evaluation of deep learning based seismic pickers. *Journal of Geophysical Research: Solid Earth*, 127(1):e2021JB023499, 2022.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE access*, 7:19143–19165, 2019.

Natural Resources Canada. Canadian National Seismograph Network [dataset]. *International Federation of Digital Seismograph Networks*, page https://doi.org/10.7914/SN/CN, 1975.

Natural Resources Canada. Canadian National Earthquake Database [dataset]. *Canadian Hazards Information Service*, page http://doi.org/10.17616/R3TD24, 1985.

M. J. Nelson and A. K. Hoover. Notes on using google colaboratory in ai education. In *Proceedings of the 2020 ACM conference on innovation and Technology in Computer Science Education*, pages 533–534, 2020.

M. Nettles and G. Ekström. Glacial earthquakes in greenland and antarctica. *Annual Review of Earth and Planetary Sciences*, 38:467–491, 2010.

F. Neutatz, B. Chen, Z. Abedjan, and E. Wu. From cleaning before ml to cleaning for ml. *Data Engineering*, page 24, 2021.

B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

L. Noriega. Multilayer perceptron tutorial. *School of Computing. Staffordshire University*, 4:5, 2005.

D. W. Otter, J. R. Medina, and J. K. Kalita. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2):604–624, 2020.

W. Ouyang et al. Imjoy: an open-source computational platform for the deep learning era. *Nature methods*, 16(12):1199–1200, 2019.

M. Paul, S. Ganguli, and G. K. Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34: 20596–20607, 2021.

F. Pérez-García, R. Sparks, and S. Ourselin. Torchio: a python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning. *Computer Methods and Programs in Biomedicine*, 208:106236, 2021.

F. Provost and T. Fawcett. Data science and its relationship to big data and data-driven decision making. *Big data*, 1(1):51–59, 2013.

N. Pérez et al. Eseismic: Towards an ecuadorian volcano seismic repository. *Journal of Volcanology and Geothermal Research*, 396:106855, 2020.

J. Quinteros et al. Exploring approaches for large data in seismology: User and data repository perspectives. *Seismological Research Letters*, 92(3):1531–1540, 2021.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, and N. Carvalhais. Deep learning and process understanding for data-driven earth system science. *Nature*, 566 (7743):195–204, 2019.

N.-C. Ristea and A. Radoi. Complex neural networks for estimating epicentral distance, depth, and magnitude of seismic waves. *IEEE Geoscience and Remote Sensing Letters*, 2021.

C. Rodrıguez, J. Bommer, and R. Chandler. Earthquake-induced landslides: 1980–1997. *Soil Dynamics and Earthquake Engineering*, 18(5):325–346, 1999.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Z. E. Ross, M.-A. Meier, E. Hauksson, and T. H. Heaton. Generalized seismic phase detection with deep learning. *Bulletin of the Seismological Society of America*, 108(5A): 2894–2901, 2018.

Z. E. Ross, D. T. Trugman, E. Hauksson, and P. M. Shearer. Searching for hidden earthquakes in southern california. *Science*, 364(6442):767–771, 2019.

A. Rovida, M. Locati, R. Camassi, B. Lolli, and P. Gasperini. Italian parametric earthquake catalogue. http://emidius.mi.ingv.it/CPTI15-DBMI15, 2019. Accessed: 21-06-2023.

A. Rovida, M. Locati, R. Camassi, B. Lolli, and P. Gasperini. The italian earthquake catalogue cpti15. *Bulletin of Earthquake Engineering*, 18(7):2953–2984, 2020.

O. M. Saad, A. G. Hafez, and M. S. Soliman. Deep learning approach for earthquake parameters classification in earthquake early warning system. *IEEE Geoscience and Remote Sensing Letters*, 2020.

O. M. Saad, G. Huang, Y. Chen, A. Savvaidis, S. Fomel, N. Pham, and Y. Chen. Scalodeep: A highly generalized deep learning framework for real-time earthquake detection. *Journal of Geophysical Research: Solid Earth*, 126(4):e2020JB021473, 2021.

K. Sanderson. Gpt-4 is here: what scientists think. *Nature*, 615(7954):773, 2023.

A. Sarkar, Y. Yang, and M. Vihinen. Variation benchmark datasets: update, criteria, quality and applications. *Database*, 2020, 2020.

A. Schlesinger, J. Kukovica, A. Rosenberger, M. Heesemann, B. Pirenne, J. Robinson, and M. Morley. An earthquake early warning system for southwestern british columbia. *Frontiers in Earth Science*, 9:684084, 2021.

H. Seywerd, A. L. Bird, L. Nykolaishen, L. McKee, S. Crane, J. Adams, and D. A. McCormack. Natural resources canada's national earthquake early warning program–an update. In *AGU Fall Meeting Abstracts*, volume 2022, pages NH33A–06, 2022.

S. Sharma, S. Sharma, and A. Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316, 2017.

P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

P. N. Shebalin, C. Narteau, and S. V. Baranov. Earthquake productivity law. *Geophysical Journal International*, 222(2):1264–1269, 2020.

A. Shrestha and A. Mahmood. Review of deep learning algorithms and architectures. *IEEE access*, 7:53040–53065, 2019.

N. Silaparasetty. Introduction to jupyter notebook. In *Machine Learning Concepts with Python and the Jupyter Notebook Environment*, pages 91–118. Springer, 2020.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

A. Simoulin and B. Crabbé. How many layers and why? an analysis of the model depth in transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 221–228, 2021.

H. Soto and B. Schurr. Deepphasepick: a method for detecting and picking seismic phases from local earthquakes based on highly optimized convolutional and recurrent deep neural networks. *Geophysical Journal International*, 227(2):1268–1294, 2021.

T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman. Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture. In *2012 IEEE symposium on computers and communications (ISCC)*, pages 000059–000066. IEEE, 2012.

D. Soydaner. Attention mechanism in neural networks: where it comes and where it goes. *Neural Computing and Applications*, 34(16):13371–13385, 2022.

D. Stathakis. How many hidden layers and nodes? *International Journal of Remote Sensing*, 30(8):2133–2147, 2009.

D. A. Storchak et al. Rebuild of the bulletin of the international seismological centre (isc)—part 2: 1980–2010. *Geoscience Letters*, 7(1):1–21, 2020.

N. Strodthoff et al. Deep learning for ecg analysis: Benchmarks and insights from ptb-xl. *IEEE Journal of Biomedical and Health Informatics*, 25(5):1519–1528, 2020.

Z. Sun, D. Yu, H. Fang, J. Yang, X. Qu, J. Zhang, and C. Geng. Are we evaluating rigorously? benchmarking recommendation for reproducible evaluation and fair comparison. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 23–32, 2020.

N. M. S. Surameery and M. Y. Shakor. Use chat gpt to solve programming bugs. *International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455-5290*, 3(01):17–22, 2023.

Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

N. Tajbakhsh, L. Jeyaseelan, Q. Li, J. N. Chiang, Z. Wu, and X. Ding. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis*, 63:101693, 2020.

A. Torfi, R. A. Shirvani, Y. Keneshloo, N. Tavaf, and E. A. Fox. Natural language processing advancements by deep learning: A survey. *arXiv preprint arXiv:2003.01200*, 2020.

A. Torralba, B. C. Russell, and J. Yuen. Labelme: Online image annotation and applications. *Proceedings of the IEEE*, 98(8):1467–1484, 2010.

R. Tozzi, F. Masci, and M. Pezzopane. A stress test to evaluate the usefulness of akaike information criterion in short-term earthquake prediction. *Scientific Reports*, 10(1):1–9, 2020.

C. Trabant, A. R. Hutko, M. Bahavar, R. Karstens, T. Ahern, and R. Aster. Data products at the iris dmc: Stepping stones for research and other applications. *Seismological Research Letters*, 83(5):846–854, 2012. doi: https://doi.org/10.1785/0220120032.

A. Trnkoczy. Understanding and parameter setting of sta/lta trigger algorithm. In *New manual of seismological observatory practice (NMSOP)*, pages 1–20. Deutsches GeoForschungsZentrum GFZ, 2009.

N. Vandeput. 12 machine learning. In *Data Science for Supply Chain Forecasting*, pages 109–121. De Gruyter, 2021.

M. Veres and M. Moussa. Deep learning for intelligent transportation systems: A survey of emerging trends. *IEEE Transactions on Intelligent transportation systems*, 21(8): 3152–3168, 2019.

A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

Z. Wan, X. Xia, D. Lo, and G. C. Murphy. How does machine learning change software development practices? *IEEE Transactions on Software Engineering*, 47(9):1857–1871, 2019.

J. Wang, Y. Chen, H. Yu, M. Huang, and Q. Yang. Easy transfer learning by exploiting intra-domain structures. In *2019 IEEE international conference on multimedia and expo (ICME)*, pages 1210–1215. IEEE, 2019a.

J. Wang, Z. Xiao, C. Liu, D. Zhao, and Z. Yao. Deep learning for picking seismic arrival times. *Journal of Geophysical Research: Solid Earth*, 124(7):6612–6624, 2019b.

Z. Wang, J. Chen, and S. C. Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2020.

G. Weatherill, M. Pagani, and J. Garcia. Exploring earthquake databases for the creation of magnitude-homogeneous catalogues: tools for application on a regional and global scale. *Geophysical Journal International*, 206(3):1652–1676, 2016.

S. E. Whang and J.-G. Lee. Data collection and quality challenges for deep learning. *Proceedings of the VLDB Endowment*, 13(12):3429–3432, 2020.

T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

J. Woollam, A. Rietbrock, A. Bueno, and S. De Angelis. Convolutional neural network for seismic phase classification, performance demonstration over a local seismic network. *Seismological Research Letters*, 90(2A):491–502, 2019.

J. Woollam, J. Münchmeyer, F. Tilmann, A. Rietbrock, D. Lange, T. Bornstein, T. Diehl, C. Giunchi, F. Haslinger, D. Jozinović, et al. Seisbench—a toolbox for machine learning in seismology. *Seismological Society of America*, 93(3):1695–1709, 2022.

Z. Wu, C. Shen, and A. Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133, 2019.

Z. Xiao, J. Wang, C. Liu, J. Li, L. Zhao, and Z. Yao. Siamese earthquake transformer: A pair-input deep-learning model for earthquake detection and phase picking on a seismic array. *Journal of Geophysical Research: Solid Earth*, 126(5):e2020JB021444, 2021.

Y. Xie, M. Ebad Sichani, J. E. Padgett, and R. DesRoches. The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. *Earthquake Spectra*, 36(4):1769–1801, 2020.

C.-X. Xue, H. Lin, X.-Y. Zhu, J. Liu, Y. Zhang, G. Rowley, J. D. Todd, M. Li, and X.-H. Zhang. Diting: a pipeline to infer and compare biogeochemical pathways from metagenomic and metatranscriptomic data. *Frontiers in Microbiology*, 12:698286, 2021.

H. Yang and S. Yao. Shallow destructive earthquakes. *Earthquake science*, 34(1):15–23, 2021.

W. L. Yeck, J. M. Patton, Z. E. Ross, G. P. Hayes, M. R. Guy, N. B. Ambruz, D. R. Shelly, H. M. Benz, and P. S. Earle. Leveraging deep learning in global 24/7 real-time earthquake monitoring at the national earthquake information center. *Seismological Society of America*, 92(1):469–480, 2021.

B. Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.

M. Y. Yip, G. Lim, Z. W. Lim, Q. D. Nguyen, C. C. Chong, M. Yu, V. Bellemo, Y. Xie, X. Q. Lee, H. Hamzah, et al. Technical and imaging factors influencing performance of deep learning systems for diabetic retinopathy. *NPJ digital medicine*, 3(1):40, 2020.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021a.

M. Zhang, M. Liu, T. Feng, R. Wang, and W. Zhu. Loc-flow: An end-to-end machine learning-based high-precision earthquake location workflow. *Seismological Society of America*, 93(5):2426–2438, 2022.

X. Zhang, M. Zhang, and X. Tian. Real-time earthquake early warning with deep learning: Application to the 2016 m 6.0 central apennines, italy earthquake. *Geophysical Research Letters*, 48(5):2020GL089394, 2021b.

M. Zhao, Z. Xiao, S. Chen, and L. Fang. Diting: A large-scale chinese seismic benchmark dataset for artificial intelligence in seismology. *Earthquake Science*, 35:1–11, 2022.

M.-H. ZHAO, X.-L. QIU, P. WANG, S.-H. XIA, Y.-M. LI, H.-L. XU, C.-M. YE, and Y. KANG. Large volume air-gun sources and its seismic waveform characters. *Chinese Journal of Geophysics*, 51(2):400–408, 2008.

Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.

J. Zhu, L. Fang, F. Miao, L. Fan, J. Zhang, and Z. Li. Deep learning and transfer learning of earthquake and quarry-blast discrimination: Applications to southern california and eastern kentucky. *Authorea Preprints*, 2022a.

W. Zhu and G. C. Beroza. Phasenet: a deep-neural-network-based seismic arrival-time picking method. *Geophysical Journal International*, 216(1):261–273, 2019.

W. Zhu, S. M. Mousavi, and G. C. Beroza. Seismic signal denoising and decomposition using deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 57 (11):9476–9488, 2019.

W. Zhu, K. S. Tai, S. M. Mousavi, P. Bailis, and G. C. Beroza. An end-to-end earthquake detection method for joint phase picking and association using deep learning. *Journal of Geophysical Research: Solid Earth*, 127(3):e2021JB023283, 2022b.

M.-A. Zöller and M. F. Huber. Benchmark and survey of automated machine learning frameworks. *Journal of artificial intelligence research*, 70:409–472, 2021.