



# DeepRFQC: automating quality control for P-wave receiver function analysis using a U-net inspired network

Sina Sabermahani  \*<sup>1</sup>, Andrew Frederiksen <sup>1</sup>

<sup>1</sup>Department of Earth Sciences, University of Manitoba, Winnipeg, Canada, R3T 2M6

**Author contributions:** *Conceptualization:* Sina. *Formal Analysis:* Sina, Andrew. *Writing - Original draft:* Sina. *Visualization:* Sina, Andrew. *Supervision:* Andrew. *Funding acquisition:* Andrew.

**Abstract** This paper introduces DeepRFQC, an automated method for quality control in P-wave receiver function analysis. Leveraging a U-Net inspired deep learning model, which has previously shown promise in denoising and phase detection, DeepRFQC efficiently distinguishes usable from noisy receiver functions. We examine data recorded by stations located in Archean and Paleoproterozoic regions of northern Canada, including seismic events from 1990 to 2023, which are expanded for training purposes by data augmentation techniques. With 1,508,449 trainable parameters, the DeepRFQC model attains 96.6% validation accuracy on a test dataset from the X5 seismic network. The model's global applicability is substantiated through supplementary analyses conducted on seismic stations situated in diverse tectonic settings. However, optimal performance is achieved when utilizing a dataset that has undergone water-level deconvolution and subsequent bandpass filtering within the range of 0.05 to 0.5 Hz. Consistent and plausible results from H- $\kappa$  stacking also validate this method. As manual quality control is a major bottleneck in receiver function processing, automated methods such as this one will allow for efficient examination of large data sets.

**Non-technical summary** Checking large amounts of seismic data for quality using receiver functions can be challenging and time-consuming, especially when done manually. However, a new method called DeepRFQC uses deep learning to automate this quality control process for P-wave receiver function analysis, making it much more efficient. DeepRFQC uses a deep learning model inspired by U-Net, trained on data from Paleoproterozoic regions in northern Canada. By enhancing the data and using over 1.5 million parameters, DeepRFQC achieves a high accuracy of 96.6% in validation tests. The primary strength of DeepRFQC lies in its capacity to automate a traditionally time-intensive process, providing an efficient solution for analyzing large seismic datasets. By streamlining the evaluation of extensive data, this approach has the potential to significantly conserve researchers' time and resources. Comprehensive testing and evaluation, including comparisons with existing models, variations in input parameters, and exploration of different hyperparameter configurations, have demonstrated DeepRFQC's robustness, adaptability, and its potential for broad adoption in the field of receiver function analysis.

Production Editor:  
Gareth Funning  
Handling Editor:  
Abhineet Gupta  
Copy & Layout Editor:  
Théa Ragon

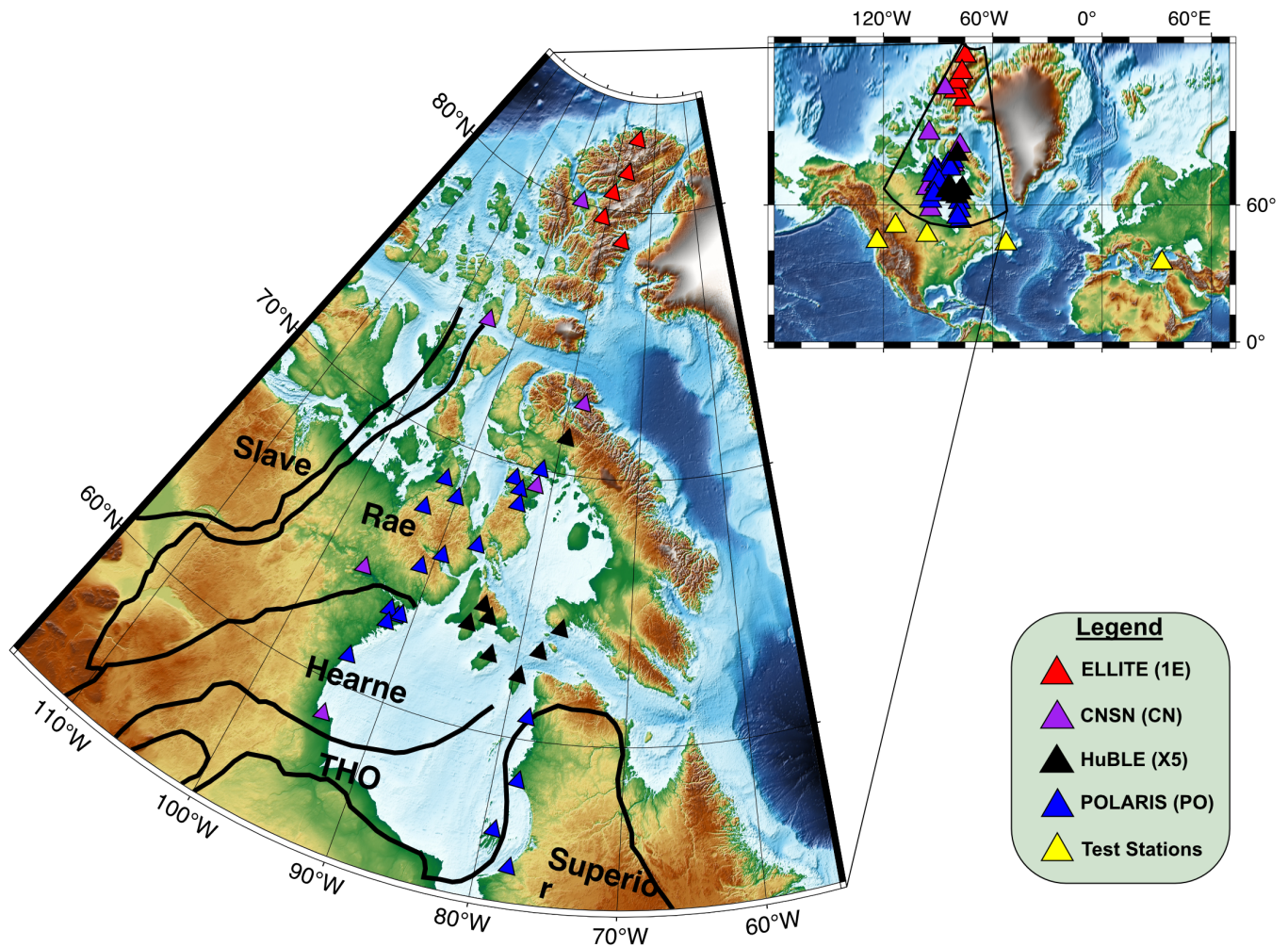
Received:  
March 26, 2024  
Accepted:  
October 10, 2024  
Published:  
November 4, 2024

## 1 Introduction

The receiver function (RF) technique is passive seismic imaging that plays a crucial role in understanding crustal and upper mantle structure. Receiver functions are sensitive to sharp changes in crustal and upper mantle properties (Vinnik, 1977; Hansen and Schmandt, 2017), complementing other techniques, such as surface wave analysis, which are more sensitive to smooth variations (Bensen et al., 2007; Shen et al., 2013; Zanjani et al., 2019; Dreiling et al., 2020), and contribute to understanding tectonic processes (Vinnik et al., 2004; Rodriguez and Russo, 2020). The arrival time of P-to-S converted phases after the direct P-wave corresponds to the depth of the interface at which the conversion occurred and the amplitude of this phase indicates the velocity contrast at the interface (Lawrence and Shearer, 2006; Ramadanti, 2023). To create a receiver function

signal, we first rotate a recording of a teleseismic earthquake into the ZRT (Vertical-Radial-Transverse) coordinate system, and then deconvolve the vertical component from the radial or transverse. Deconvolution can be performed in the time or frequency domains. Some established techniques include iterative time-domain deconvolution (Kikuchi and Kanamori, 1982; Ligorria and Ammon, 1999) as well as frequency-domain multitaper (Park and Levin, 2000), and water-level deconvolution (Ammon, 1992). Receiver functions are strongly affected by noise on either of the components, which can be amplified by the deconvolution process. Deconvolution has the potential to amplify noise in a frequency-dependent manner, complicating the precise interpretation of converted waves. Quality control in P-wave receiver functions is essential for ensuring the accuracy and reliability of seismic imaging and interpretation. This quality control, distinguishing acceptable data from unacceptable data, is a time-consuming and

\*Corresponding author: [sabermas@myumanitoba.ca](mailto:sabermas@myumanitoba.ca)

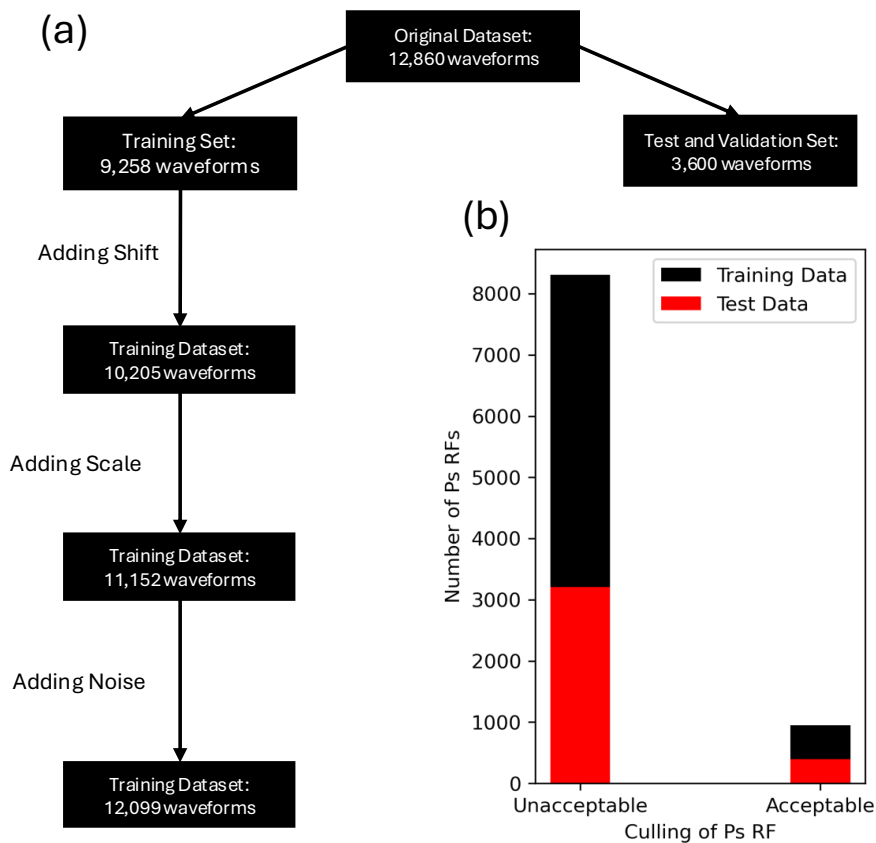


**Figure 1** Illustrating the distribution of seismograph stations used in this study, overlaid on major tectonic boundaries (adapted from Whitmeyer and Karlstrom, 2007). Black triangles indicate stations belonging to the X5 network, our test dataset for model development. Yellow triangles represent additional test stations located throughout Canada and in Turkey.

tricky process, especially in larger studies. Since receiver functions require visible Ps conversions to be useful, they require careful examination of what are often weak arrivals in the P coda. Recognizing these challenges, automated RF quality control (QC) procedures become invaluable tools. Comprehensive analyses necessitate careful examination of each individual radial trace, which can be time-consuming for large datasets. Automated RF QC not only saves considerable human effort but also facilitates the efficient analysis of larger datasets, thereby enhancing the reliability and robustness of the findings (Hopper et al., 2017). To date, various semi-automated and automated receiver function quality control methods have been introduced, offering significant time savings in receiver function analysis. Crotwell and Owens (2005) introduced the EarthScope Automated Receiver Survey (EARS), a freely available Java-based package. In contrast to EARS which relies on phase similarities, Yang et al. (2016) introduced a semi-automated method. This method is based on three categories of metrics: (1) deconvolution attributes, (2) characteristics of individual receiver functions and (3) statistical attributes of station gathers.

In addition to classical approaches, automated quality control can be done using AI-based (Artificial Intel-

ligence) methods. With the expanding applications of machine learning (ML) and its ability to identify patterns better than humans in some cases (Gong et al., 2022), the potential of ML-based models for quality control is increasingly recognized. Machine learning based approaches, supervised and unsupervised, have demonstrated their capabilities to closely mimic human behavior. In the field of seismology, these approaches have shown comparable or even superior performance to humans in extracting information from seismic data. Two tasks that have received significant attention are data denoising and phase detection (Zhu and Beroza, 2019; Mousavi et al., 2019; Adler et al., 2021). For receiver function quality control, both supervised (Gong et al., 2022) and unsupervised (Krueger et al., 2021) machine learning approaches have been used. Gong et al. (2022) created four deep learning architectures to automate the quality control of receiver functions. They concluded that a model combining CNN (Convolutional Neural Network) and LSTM (Long Short Term Memory) layers demonstrated the highest performance. However, while their model excelled in the areas used for training (three distinct tectonic settings in China), it may not perform optimally in other regions and updating the weights via transfer learning (meaning adjust-



**Figure 2** a) Illustration of the dataset division into training and test sets, including the expansion of training data after augmentation. b) Distribution of data labeled as acceptable and unacceptable in both the training and test sets.

ing the model based on new dataset) would need to be considered for usage in other area and tectonic settings. In this study we focus on Archean and Paleoproterozoic regions of northern Canada. We tested the model of Gong et al. (2022) to quality control RFs but found this method to be insufficient. Consequently, there arose a need to either update their model’s weights through transfer learning on our dataset or to develop a new model that outperforms it. We developed a new model inspired by U-net (Ronneberger et al., 2015) aiming for a simple yet capable model suitable for handling large datasets. In the following sections, we will first provide a brief overview of our data collection procedure and data preparation. Subsequently, we will elaborate on the methodologies employed in this study. In the “Experimental Setup” section, we will detail the model parameters and the process of fine tuning those parameters. Following that, we will present the results and evaluate them using a well-established technique in receiver function analysis. Additionally, we will investigate the sensitivity of our model to various hyperparameters, training data and choice of optimizer. We will then discuss the findings and compare the performance of our model with that of Gong et al. (2022). Finally, we will conclude by summarizing the results and providing comparisons.

## 2 Data Collection and Preparation

The dataset used in this study comprises seismic events recorded by stations located within the latitude range of 49.8° to 80°N and longitude range of 105° to 75°W, situated amidst the Rae and Hearne tectonic provinces to the northwest, and the Superior craton to the southeast, all of which are Archean in age. Between these cratons lies the Proterozoic Trans-Hudson Orogen, a region of considerable tectonic interest (Whitmeyer and Karlstrom, 2007, Figure 2). A substantial number of the deployed instruments were originally deployed to investigate the tectonics of the Trans-Hudson as part of the HuBLE experiment (Bastow et al., 2011). The study area contains several smaller-scale structural features of interest, prominent among which are the Wager Bay and Chesterfield faults (Snyder et al., 2015). Events with magnitudes ranging from 5.5 to 9 were examined, limited to seismic events occurring within 30° to 90° from the center of the study area, a range in which teleseismic P-waves are not triplicated and arrive well separated from other arrivals. The temporal scope of the dataset spans from January 1, 1990, to November 1, 2023. Stations involved in this study are shown in Figure 1.

To assemble this comprehensive dataset, seismic data were sourced from two primary repositories: the IRIS (Incorporated Research Institutions in Seismology) and FDSN (Federation of Digital Seismograph Networks) (Scripps Institution of Oceanography, 1986) web server

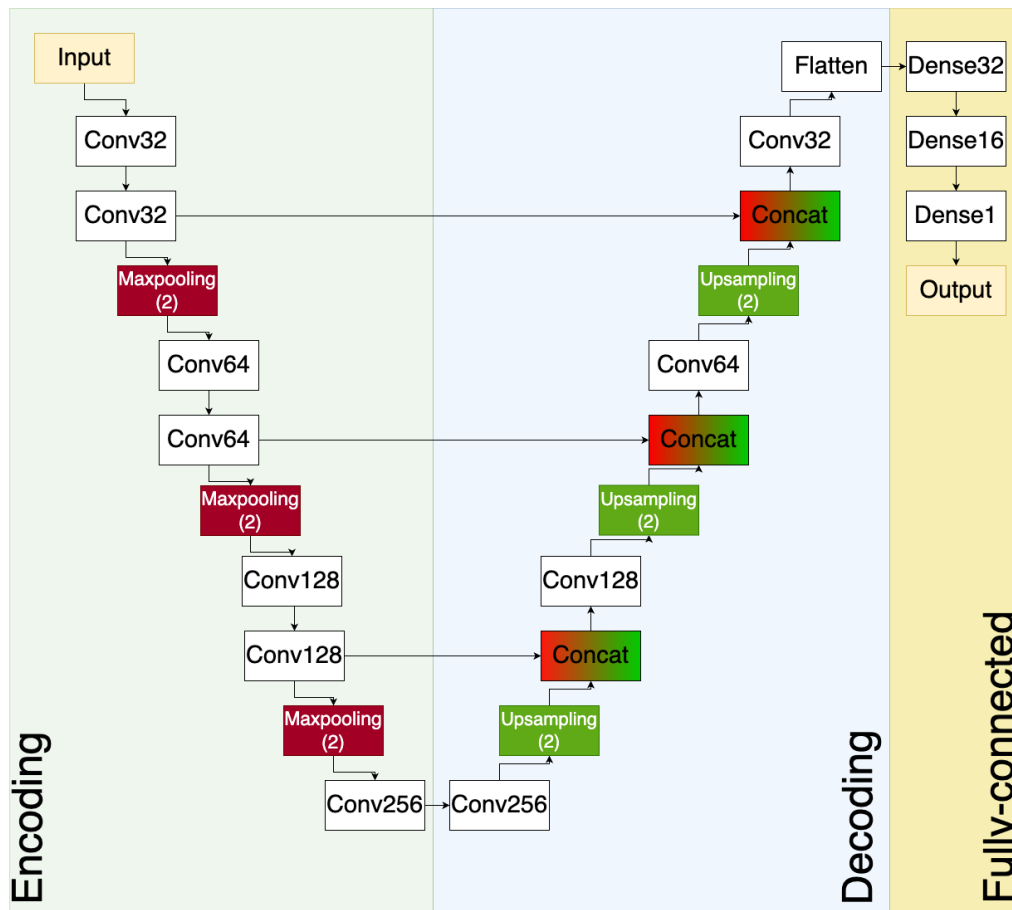
and the server maintained by [Natural Resources Canada \(1975\)](#). The creation of our database was followed by a phase in which we converted raw seismic data into receiver function signals. We began by reorienting the waveforms from the ZNE (Vertical, North, East) to the ZRT (Vertical, Radial, Transverse) coordinate system. After reorientation, we constrained the signal's frequency spectrum to between 0.05 and 0.5 Hz using a bandpass filter. Then, to eliminate the source characteristics from the radial and transverse components, we applied water-level deconvolution ([Ammon, 1992](#)) with a water-level parameter set to  $10^{-2}$ . While frequency domain techniques are more susceptible to sidelobe residuals at the base of RF peaks due to numerical artifacts from the Fast Fourier Transform (FFT; [Kind et al., 2020](#)), the decision was made to proceed with this faster and simpler approach. The steps explained above have been performed using a code written in Python which incorporates classes and functions from `RfPy` ([Audet, 2020](#)). The use of water-level deconvolution is common practice as it introduces a minimum value for the bottom terms in the spectral division within the frequency domain, thus ensuring the stability of the deconvolution process ([Wilson and Aster, 2005](#)). Within this study's context, the transverse component is not examined further. However, in future work on this dataset, attention will be directed towards analysis of these transverse components, which can provide information on crustal anisotropy. We successfully performed deconvolution on 12,860 waveforms (Figure 2). These waveforms were subsequently divided into two distinct categories: the training dataset, which comprised 9,258 waveforms, and the validation and test dataset, consisting of 3,600 waveforms (Figure 2). The validation and test dataset notably included the X5 network, which comprises stations SHWN, CRLN, CTSN, DORN, MANN, MARN, NOTN, and SHMN. Using data augmentation, we artificially increase the size and diversity of the dataset. To augment the dataset, three commonly employed and effective techniques were utilized: introducing "new" data by adding noise to the original dataset, as well as shifting and scaling the dataset (Figure 2a). Retaining the original data alongside the dataset augmented with added noise doubles the size of our dataset. This approach, as demonstrated by [Chang et al. \(2022\)](#) has been shown to improve the performance of deep learning models. To do this, white noise with a peak amplitude of  $10^{-2}$  was added to the entire dataset, thereby doubling the quantity of waveform data available for analysis. After normalizing each waveform by its maximum, the magnitudes of the waveforms ranging between -1 and 1. Introducing noise with a maximum amplitude of  $10^{-2}$  corresponds to 1% of the maximum amplitude of the waveforms. In addition to noise augmentation, two additional augmentation techniques were selectively applied to datasets categorized as acceptable data, as the volume of data deemed low quality was sufficient. One of these techniques is temporal shifting, termed rolling, where waveform data were translated temporally within a range of -5 to +5 seconds, wrapping around any overlap ([Shorten and Khoshgoftaar, 2019](#)). The second technique, scaling, involved adjusting the

waveform amplitude by a scaling factor ranging from 0.9 to 1.1, which equates to an amplitude variation of -10% to +10% ([Iwana and Uchida, 2021](#)). Each augmentation technique was applied to the dataset labeled as acceptable, with each technique adding 958 traces to the dataset, for a total of 12,134 traces collectively in the training set.

### 3 Methodology

DeepRFQC, our model, is built upon a U-Net ([Ronneberger et al., 2015](#)) inspired architecture with 1,508,449 trainable parameters, for a volume of 13.2 MB. While these parameters/weights may not hold significant meaning individually, they effectively perform their intended function when appropriately positioned within the network. The network comprises two main components: a U-Net-inspired section and a set of fully connected layers (Figure 3). In the U-Net-inspired segment, the architecture follows an encoding-decoding structure with skip connections ([Ronneberger et al., 2015](#)). The encoding phase captures hierarchical features, while the decoding phase reconstructs the output. Skip connections facilitate direct connections between corresponding layers in the encoding and decoding phases, promoting the preservation of positional information (position of features in the signal), and aiding in the efficient learning of intricate patterns ([Ronneberger et al., 2015](#)). The inclusion of fully-connected layers can further enhance the model's ability to learn complex feature hierarchies, combining high-level semantic information with detailed spatial features. This architecture allows DeepRFQC to effectively extract and analyze complex features in P-wave receiver function data, contributing to its accuracy and robustness. In the initial step of training, we input our data into the model through a 1D convolutional layer with 32 filters, a stride of one, and a kernel size optimized through testing different values. In the encoding branch, the number of filters doubles after each max-pooling layer, while in the decoding phase, it is halved after each upsampling layer.

In the process of optimizing our network, we had to work within practical resource constraints. As such, we were unable to train the network exhaustively across countless hyperparameter configurations. Instead, we conducted a broad search over various network hyperparameters to identify the most promising candidate for further refinement. This involved making incremental adjustments to the selected candidate, iteratively bringing it closer to our desired model with the expected accuracy level. During this step, we employed a meticulous approach, referred to as "Panda" ([Basha and Rajput, 2019](#)), which allowed us to closely monitor the network's performance and identify the configuration most suitable for our objectives. Following the establishment of the network, preprocessing steps were undertaken to prepare the dataset for consumption by the network. After the creation of RF signals, the next preprocessing step involved manual labeling. For this labeling process, we developed a Python program to examine the radial receiver function obtained for each



**Figure 3** Illustration of the architectural configuration of DeepRFQC, encompassing a total of 27 layers, which incorporate Convolutional, Max-Pooling, Up-sampling, and fully connected layers. Above, “Conv” denotes 1-D convolution in the layers, with the adjacent number indicating the number of filters. “Maxpooling(2)” signifies a maxpooling (a type of down sampling) layer with a factor of 2, and similarly upsampling is performed by a factor of 2. “Concat” layers indicate the concatenation of their lower layers with the output of data from the Encoding branch. The role of “Flatten” is to convert the previous layer into 1-D form, and since the outputs are already 1-D, it doesn’t affect them, although it can be utilized if considering 2-D data such as STFT results. “Dense” layers represent hidden layers, and the accompanying numbers indicate the number of neurons.

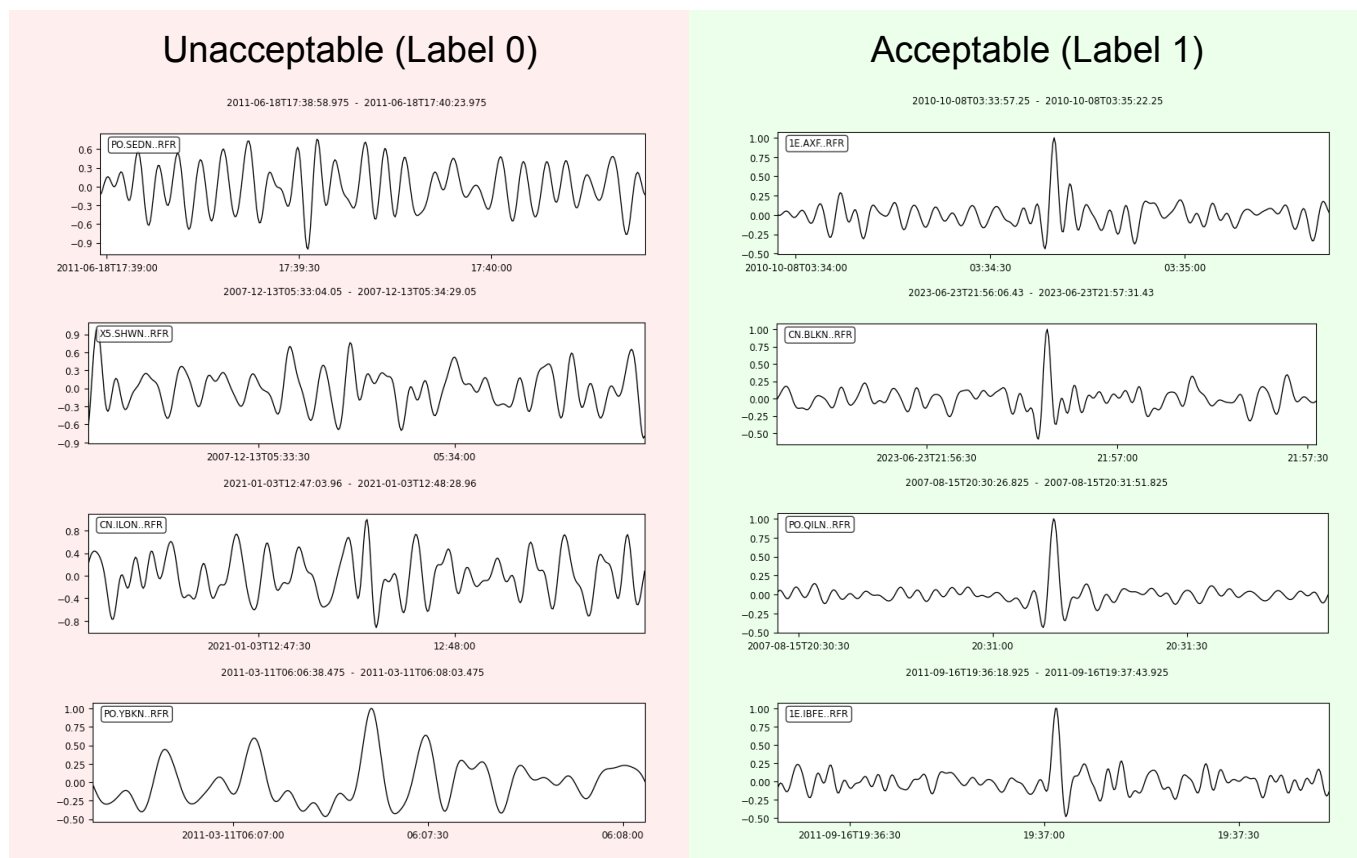
event-station pair and determine whether it met the criteria for suitability in receiver function analysis. Figure 4 shows examples of seismic receiver function signals labeled as 0 (unacceptable) and 1 (acceptable). Unacceptable signals often have indistinct initial P-wave arrivals and high noise levels surpassing anticipated phases. Conversely, acceptable signals have clear P-wave arrivals and minimal noise before this phase. Initial waveform assessment focused on the segment preceding the P arrival. If characterized by noise and high amplitudes, it was promptly labeled as unacceptable (0). If the pre-P phase had lower noise levels, subsequent phases were manually inspected for distinctive Ps conversions. Our code facilitated this by highlighting segments for ease of identification, and records showing clear recognition of these phases were labeled as acceptable (1) for subsequent analysis.

After preparing the dataset and inputting it into DeepRFQC, we evaluated the network using the X5 seismic network. The network X5 was used as both test and validation set. Utilizing a Python code implemented with TensorFlow, we predict the labels for event-station pairs within the X5 dataset. To address concerns about labeling quality, we validate our results using the H- $\kappa$  stack-

ing method. The H- $\kappa$  method, introduced by Zhu and Kanamori (2000), is a seismic analysis technique that aims to determine the Moho depth (the crust-mantle boundary) and the Vp/Vs ratio (the ratio of compressional wave velocity to shear wave velocity) by stacking RF energy at the expected arrival times of the Moho conversion and its associated multiples.

## 4 Experimental Setup

In the first phase of our model training, we established key parameters to govern the process. Each waveform in our dataset was set to a size of 424 samples (equivalent to 84.8 seconds, 42.4 seconds before and after the P-arrival estimated by TauPy (The ObsPy Development Team, 2022), providing a standardized basis for subsequent analyses (the time length follows Audet, 2020; Audet et al., 2020). To enhance the dataset and introduce variability, we employed data augmentation by generating noise with magnitudes ranging from 0 to  $10^{-2}$ , rolling between -5 to +5 sec and scaling by -10 to 10%. Through experimentation with convolutional neural network (CNN) architecture, we explored kernel sizes (the dimensions of the filter used for sliding over



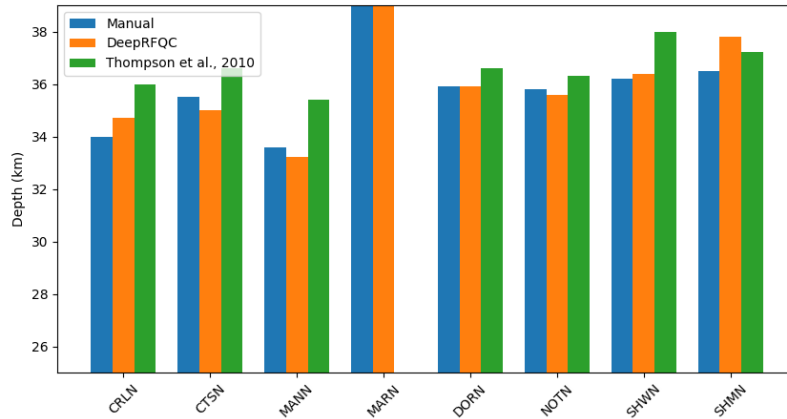
**Figure 4** Samples of acceptable and unacceptable waveforms labelled for feeding the deep neural network model. On the acceptable side, we see a lower noise level before the P-wave arrival (the phase exactly located at the center) and some obvious phases after it.

input data, determining the region it considers for feature extraction) of 3, 5, 7 and 9 and 11 samples. Following a thorough assessment, a kernel size of 5 was chosen due to its superior accuracy. The optimal batch size for effective training was determined to be 256, facilitating efficient processing of data batches. It is worth noting that batch sizes of 64, 128, 256, 512, and 1024 were experimented with during the evaluation process. We conducted an evaluation of four learning rate and learning decay pairings, denoted as (learning rate, learning decay). The configurations tested were  $(10^{-6}, 10^{-8})$ ,  $(10^{-5}, 10^{-7})$ ,  $(10^{-4}, 10^{-6})$ ,  $(10^{-3}, 10^{-5})$ , and  $(10^{-2}, 10^{-4})$ . The pairing that yielded the best results was  $(10^{-4}, 10^{-6})$ . Another important hyperparameter is initialization methods, that is, the first assumptions of weights in the network. We tried several different options, including starting with all zeros, all ones, random numbers, and specific initialization schemes called Glorot uniform (Glorot and Bengio, 2010), He normal, and He uniform (He et al., 2015). After evaluating the results of the first training stage, we decided to use He uniform initialization for the rest of the training. Using the optimum values selected in the previous part, the model underwent extensive training over 200 epochs, allowing it to progressively learn and adapt. To prevent overfitting, an early stopping mechanism based on validation accuracy was implemented. This mechanism, with a patience of 20 epochs and a minimum expected improvement of  $10^{-3}$ , ensured that the model ceased training

when further improvements were marginal. Our overarching validation accuracy criterion aimed for a target of 96%. This value was found after a trial-and-error procedure, as lower values can be easily observed, while for higher values, the model diverges or ceases improving before reaching the target.

## 5 Results

Upon initiating the network training with the specified parameters outlined in the preceding sections, our model achieved a validation accuracy of 96.6%. Both training and validation accuracy converged to the same value of 97%, indicating the absence of overfitting. After applying DeepRFQC to the X5 network's receiver function data (the whole test set; while in the training part, validation accuracy is determined from a part of the test set according to the batch size), we achieved noteworthy results with an accuracy exceeding 93%. When comparing these outcomes with those from Gong et al. (2022) on earthquakes recorded at Chinese seismic stations, our results demonstrate comparable performance. Regarding our findings, it is important to note that the slightly lower prediction accuracy, in contrast to the validation accuracy, can be attributed to two factors: firstly, a smaller size was selected for the validation set (because of batch processing) compared to the test set, and secondly, the samples from the validation set were chosen randomly, and the distribution of



**Figure 5** Comparison of Moho depth from H- $\kappa$  stacking with different sources of RF: manual, DeepRFQC, and from Thompson et al. (2010). For station MARN, no depth was reported by Thompson et al. (2010)

these chosen samples can influence the accuracy of the model. Conversely, in the prediction phase, the entire test set is utilized. Table 1 reveals that out of 1,778 waveforms, the model incorrectly predicted 123 instances and correctly predicted 1,655 instances.

Measure	Test set	Training set
True Positive	334	-
True Negative	1321	-
False Positive	99	-
False Negative	24	-
Accuracy	0.931	0.975
Precision	0.771	-
Recall	0.933	-
F1-score	0.845	0.934

**Table 1** Performance Metrics for Test and Training Sets

In Table 1, we provide other statistical measures like precision, recall and F1-score (Sokolova et al., 2006) as our data set was unbalanced and needed to be considered carefully for this imperfection. This F1-score was introduced as it is a more reliable measure to judge the outcome of the network given the imbalance in the data set. Equations 1 to 3 show how we calculated these measures:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

$$\text{F1-score} = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (3)$$

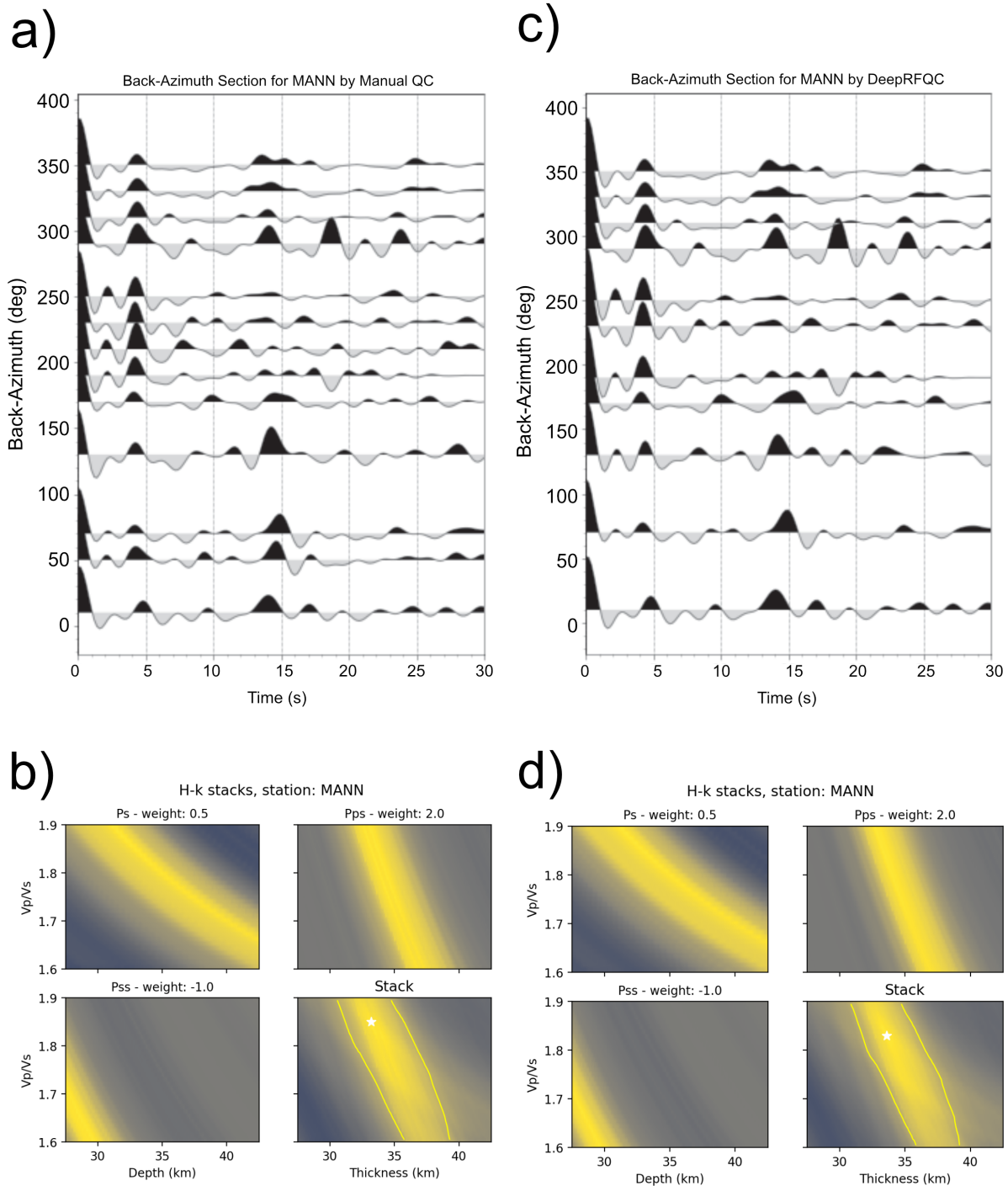
The equations above use TP, FP, TN, and FN to denote the number of true positive, false positive, true negative, and false negative predictions, respectively.

## 5.1 H- $\kappa$ stacking test

Testing the reliability of the trained model involves not only analyzing statistical results but also conducting a

real-world application using the obtained data. The output of the network was employed as input data for H- $\kappa$  stacking, a method introduced by Zhu and Kanamori (2000) to measure bulk crustal properties from receiver functions. In this method, a search is conducted across a range of Moho depths (H) and Vp/Vs ratios ( $\kappa$ ; specifically,  $20 \text{ km} < H < 50 \text{ km}$  and  $1.5 < \kappa < 2.1$  in this implementation) for each waveform. For every H- $\kappa$  pair, the projected travel times are computed for the major Moho-interacting phases (Ps, Pps, and Pss) using the ray parameter of each trace. The amplitudes of the traces at the projected times are then combined to form the H- $\kappa$  stacked value at that particular point. Executing this procedure for all conceivable values in the H- $\kappa$  grid produces the final stack, showcasing a peak when arrivals are optimally aligned. To improve the stacking process, we employed phase-weighted stacking following the guidelines of Schimmel and Paulssen (1997). Additionally, a P-wave velocity of 6.0 km/s was assumed for stacking purposes. The results for each phase are subsequently stacked using weights which may be adjusted at each station (for example, at MANN the Pss phase was poor, and its weight was set to zero). Figure 5 shows the Moho depths extracted from H- $\kappa$  stacking through different sets of RFs including manual quality control from this study, automated quality control from this study, and published depths from Thompson et al. (2010). A detailed comparison is available in Table S1, presenting the values for H,  $\kappa$ , H-error,  $\kappa$ -error, and the number of waveforms utilized for stacking at each individual station. As can clearly be seen, manual and automated analyses return essentially the same values but there is a systematic difference between manual-automated pair and Thompson et al. (2010) that could be attributed to a different crustal Vp assumption (6.5 km/s for Thompson et al., 2010).

Figures 6 and 7 illustrate the results of H- $\kappa$  stacking conducted at the MANN and NOTN stations, with additional data provided in the supplementary materials (Figures S1 to S6). It is worth noting the similarity between the results obtained from manual quality control and those from analyzing with DeepRFQC. While the



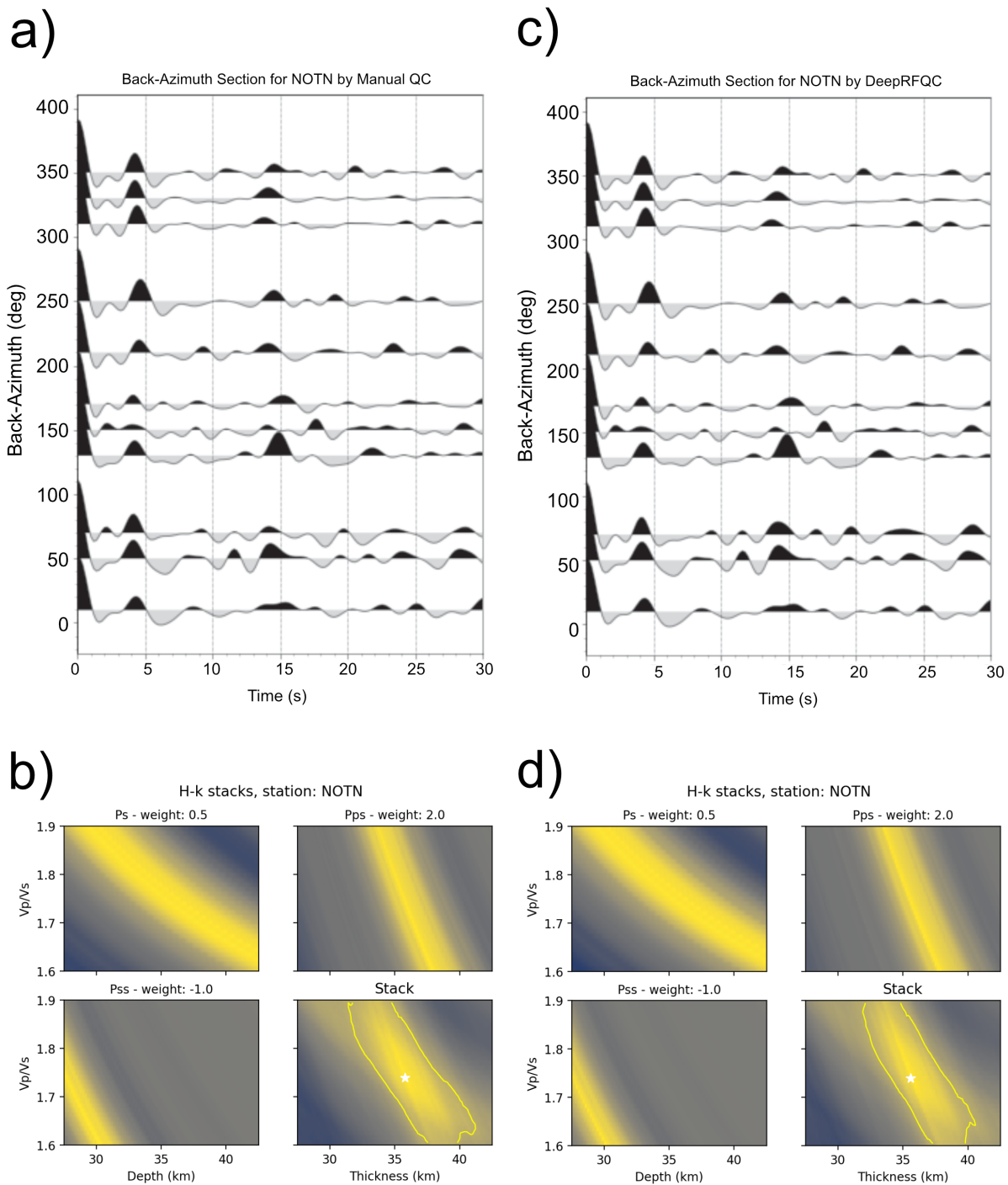
**Figure 6** Results of the analysis for station MANN, a, and c) back-azimuth section plots for manual and automated QC by DeepRFQC, respectively, b and d) H- $\kappa$  stacking for manual and automated QC by DeepRFQC, respectively.

automatic quality control method performs outstandingly, it will not outperform manual quality control, as it is trained on a human-labeled dataset. The automatic method builds upon and relies on the manual quality control process. As evident in Figures 6 and 7 (a, c), the Ps phase is discernible at around 4 seconds for both stations shown. A consistent pattern is observed across the other six stations, as illustrated in the supplementary material.

Additionally, station MANN exhibits a subtle precursor phase before Ps, evident in the traces from the automated quality control results. A similar observation

holds for station NOTN, where a minor phase preceding Ps is identifiable in the manual QC results, particularly at back azimuths of 70° and 150°. Interestingly, this phase is clearly discernible in automated QC data at a back azimuth of 150°. Regarding reverberations, the Pps phase is clearly discernible around 14 seconds at stations MANN and NOTN. However, for back azimuths of 170° and 190° at station MANN, the trend in back azimuth is lost in both manual and automated QC results. Interestingly, the variation with back azimuth at this station follows a pattern resembling two periods of a sinusoid function, consistent across both manual and auto-





**Figure 7** Results of the analysis for station NOTN, a, and c) back-azimuth section plots for manual and automated QC by DeepRFQC, respectively, b and d) H- $\kappa$  stacking for manual and automated QC by DeepRFQC, respectively.

mated QC results, that is an indicator of anisotropy beneath the station. A similar consistent trend in the arrival of Pps is observable at station NOTN.

## 5.2 Investigating data and hyperparameter influence

In this section, we analyze the model’s sensitivity to training set, data augmentation and hyperparameters. To assess sensitivity to the training set, we removed each seismic network and retrained the whole process using optimal hyperparameters. This allowed us to ob-

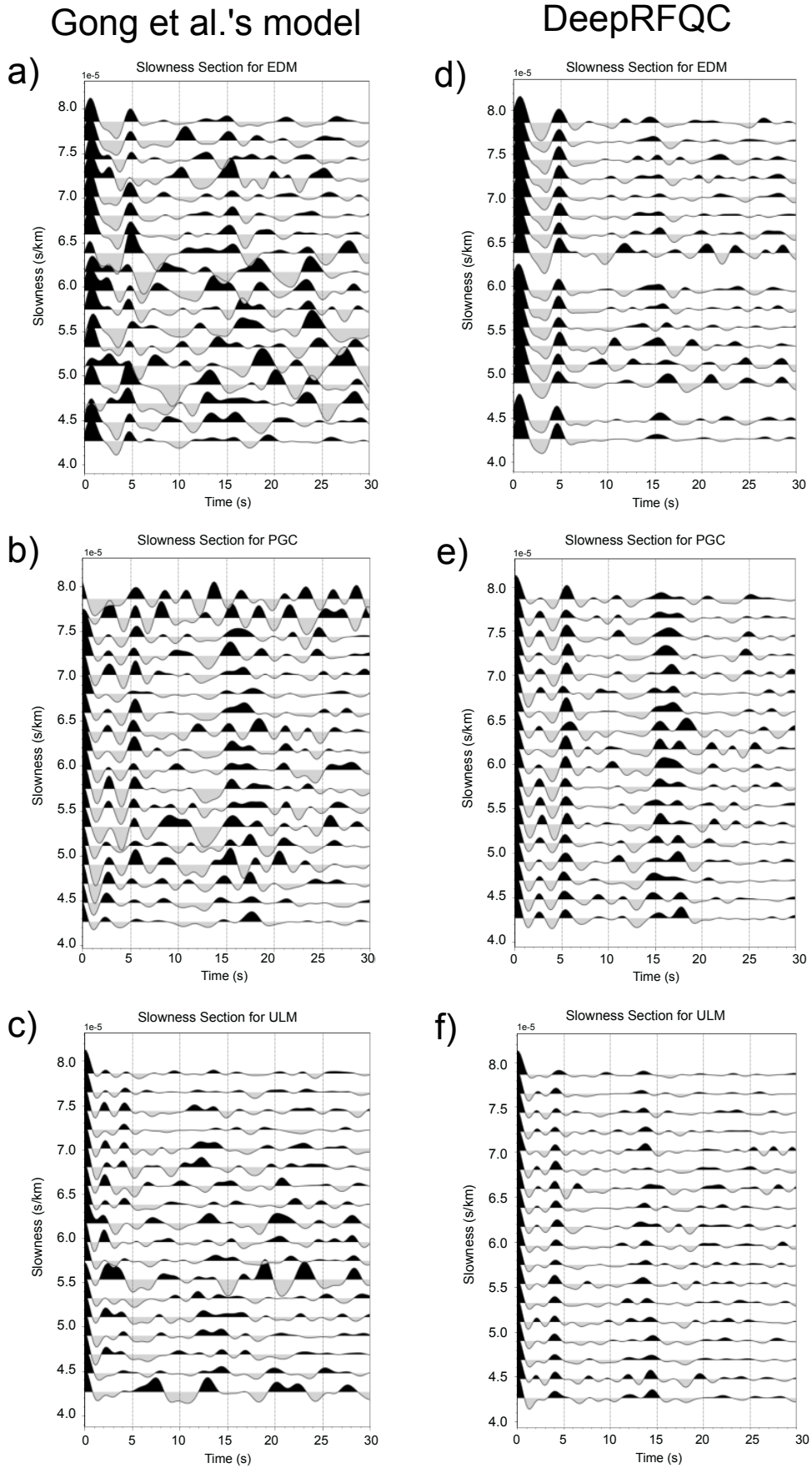
serve how the model performance would be affected. Taking F1-score as the performance metric, we found that the F1-score is the highest for PO (77%) and lowest for 1E (0.81%). This is as expected, as 1E has the least amount of data, resulting in a minimal impact on the training set (Table 2). The impact of data augmentation methods becomes evident when examining model performance. When shift and noise augmentations are removed, performance noticeably drops (Table 2). Removing scaling augmentation, on the other hand, does not significantly reduce performance. However, it does lead to an increase in overfitting, as evi-

Object	Training Loss	Training Accuracy	Training F1-score	Validation Loss	Validation Accuracy	Validation F1-score
<b>Removing network to see the sensitivity of the model to each network</b>						
PO	0.0175	0.9803	0.9527	0.0404	0.9448	0.7753
CN	0.0262	0.9649	0.9077	0.0297	0.9585	0.8000
1E	0.0243	0.9692	0.9186	0.0296	0.9593	0.8130
<b>Removing each data augmentation method</b>						
Shift	0.1174	0.8789	0.6901	0.1103	0.8897	0.6701
Scaling	0.0191	0.9750	0.9153	0.0290	0.9622	0.8381
Noise	0.1383	0.8465	0.7300	0.0352	0.9514	0.7977
<b>Testing different Hyperparameters</b>						
Kernel Size: 3	0.0364	0.9524	0.8738	0.0263	0.9647	0.8394
Kernel Size: 5	0.0184	0.9775	0.9408	0.0261	0.9667	0.8529
Kernel Size: 7	0.0145	0.9823	0.9533	0.0294	0.9625	0.8327
Kernel Size: 9	0.0392	0.9461	0.8573	0.0295	0.9572	0.8226
Kernel Size: 11	0.1875	0.8125	0.0000	0.1113	0.8897	0.0000
LR, decay: $10^{-6}, 10^{-8}$	0.0655	0.9148	0.7738	0.0359	0.9536	0.7976
LR, decay: $10^{-5}, 10^{-7}$	0.0619	0.9301	0.8240	0.0357	0.9578	0.8216
LR, decay: $10^{-4}, 10^{-6}$	0.0269	0.9665	0.9119	0.0316	0.9650	0.8463
LR, decay: $10^{-3}, 10^{-5}$	0.1875	0.8125	0.0000	0.1103	0.8897	0.0000
LR, decay: $10^{-2}, 10^{-4}$	0.1875	0.8125	0.0000	0.1103	0.8897	0.0000
Batch Size: 64	0.0241	0.9677	0.9149	0.0273	0.9642	0.8390
Batch Size: 128	0.1875	0.8125	0.0000	0.1103	0.8897	0.0000
Batch Size: 256	0.0377	0.9478	0.8626	0.0256	0.9661	0.8498
Batch Size: 512	0.0769	0.9014	0.7460	0.0347	0.9542	0.8061
Batch Size: 1024	0.0804	0.9134	0.7720	0.0411	0.9567	0.8098
<b>Testing different Initializers</b>						
Initializer: Zeros	0.2469	0.8125	0.0000	0.2454	0.8897	0.0000
Initializer: Ones	0.2469	0.8125	0.0000	0.2454	0.8897	0.0000
Initializer: Random Normal	0.1181	0.8151	0.0883	0.0461	0.9306	0.7950
Initializer: Glorot Uniform	0.0189	0.8151	0.0883	0.0461	0.9306	0.5629
Initializer: He Normal	0.0189	0.9755	0.9355	0.0268	0.9647	0.8418
Initializer: He Uniform	0.0268	0.9656	0.9092	0.0261	0.9653	0.8481
<b>Testing the performance of different optimizers</b>						
Adagrad	0.2341	0.6084	0.4654	0.1398	0.8772	0.5424
Adadelata	0.1289	0.8403	0.4861	0.0989	0.9025	0.4311
Nadam	0.1875	0.8124	0.0000	0.1103	0.8897	0.0000
RMSProp	0.0258	0.9658	0.9099	0.0277	0.9642	0.8398
SGD	0.1769	0.7878	0.5733	0.1423	0.8753	0.5469
ADAM	0.0334	0.9556	0.8817	0.0261	0.9644	0.8428

**Table 2** Performance metrics (Loss, Accuracy, and F1-score) on training and validation sets with varied hyperparameters and data sensitivity tests.

denced by the significant difference between training set F1-score (91%) and validation set F1-score (83%). To evaluate the impact of hyperparameters, we tested various values while fixing the remaining parameters at their previously identified optimal values. During the evaluation, we recalculated the optimal values to confirm that they indeed resulted in superior performance, which the results confirm. Our analysis of kernel sizes reveals that kernels of size 3, 5, 7, and 9 achieve the highest F1-scores on the validation set. Although F1-score is relatively comparable for all four sizes, the F1-score of the training set is higher for kernel sizes of 5 and 7, and we chose the size 5 since the difference between validation and training set was smaller and the chance of overfitting is lower. It should be mentioned that for kernel sizes of 9 and 11, the model training performance dramatically diverges. We further investigated the impact of the learning rate (LR) and its decay (Decay) hyperparameter pair. To ensure training pro-

gressed beyond 100 iterations, we chose a decay rate set at 1% of the learning rate. This strategy facilitated a smoother decrease in the learning rate throughout the iterations, preventing the model from overlooking minima where it performed well (local minima). It can easily be seen that the pairs of  $10^{-5}, 10^{-7}$  and  $10^{-4}, 10^{-6}$  are the best ones while the latter takes the lead by almost two percentage points. Our analysis showed that batch sizes of 64 and 256 achieved superior performance compared to other tested values. While the F1-score of validation set is close for both batch sizes, we opted for 256. The decision was influenced by two factors; first, the slight improvement in validation accuracy and F1-score, and second, the potentially lower risk of overfitting associated with a larger batch size. The final hyperparameter investigated was the initializer. We evaluated several built-in TensorFlow initializers: zeros (initializes all weights to zero), ones (initializes all weights to one), random normal, Glorot uniform



**Figure 8** Comparison of slowness section plots of Gong et al. (2022)'s model and DeepRFQC for three selected stations in Canada with varying degrees of similarity to the training dataset; a and d) EDM, b and e) PGC and c and f) ULM.

(Glorot and Bengio, 2010), He normal, and He uniform (He et al., 2015). Upon analyzing the performance of each initializer, it is evident that zeros and ones do not function effectively in this context. Setting initializer to random normal leads to diverging model performance. Among the remaining three options, He normal and He uniform demonstrate superior performance with He uniform being the preferable choice due to its potential for preventing overfitting. In our optimizer testing phase, we trained our model using multiple optimizers to assess their impact on model performance. We tested six optimizers and found that RMSProp (Mukkamala and Hein, 2017) and Adam (Kingma and Ba, 2014) outperformed the others, achieving F1-scores of 83.9% and 84.2% respectively. Our analysis revealed that for this specific task, the choice between Adam and RMSProp as optimizers does not yield a significant difference in performance.

## 6 Discussion

DeepRFQC, an automated quality control for P-wave receiver function data, addresses the labor-intensive nature of trace examination in larger studies. Utilizing a convolutional deep learning model inspired by U-Net, automated QC procedures not only reduce human effort but also enhance the efficiency of analyses for larger datasets. The U-Net architecture's proficiency in capturing intricate seismic data features proves crucial in accurately discerning usable data from noise, optimizing the overall quality assessment process. In contrast to Gong et al. (2022)'s prior deep learning model, which explored four different architectures and identified convolutional and long-short memory as the best, our model exhibits a slight improvement in accuracy. The additional advantage of our model for Canadian applications lies in its foundation on data from Canadian seismic stations, capturing the nuances of associated geological structures. Significantly, our model is characterized by a smaller volume (fewer weights), enhancing efficiency, especially when handling larger datasets. Automating the quality control process becomes particularly time-saving when managing a vast number of waveforms. In the worst-case scenario, it can efficiently identify potential acceptable waveforms, serving as a preliminary filter for further analysis. Based on our dataset, it's noteworthy that around 90% of the collected data were labeled as noise (Figure 2). Consequently, employing automated algorithms can potentially save nine times the effort when compared to manually inspecting and labeling the entire dataset, as one would only need to double-check the output of the network. Additional considerations involve the incorporation of statistical measures such as the F1-score and the assessment of factors like True Positive and True Negative. To ensure a careful reliance on the results, we specifically isolated network X5. For stations within this network, our algorithm has more instances of retaining unacceptable (i.e. noisy) waveforms in the final RF stacks (False Positive) than omitting acceptable waveforms from the final RF (False Negative). This indicates a relatively lower risk of losing information waveforms. It is important to acknowledge

that our waveform labeling criteria were stringent, and some waveforms marked as unacceptable may perform reasonably. Examining the outcomes of False Positives (Figure 8), it becomes apparent that certain instances may be deemed somewhat acceptable. The decision to label them as unacceptable led to a reduction in the F1-score to 84% in the test set. Interestingly, in the training set, where the model encounters a smaller proportion of such waveforms, it excels, achieving a higher F1-score. To validate assertions regarding False Positive and F1-score, we conducted H- $\kappa$  stacking on the intact results of both automated and manual QC, providing a basis for result comparison. Figures 6 and 7 (b and d) depict the H- $\kappa$  stacking outcomes for manual and automated QC. Evidently, the results exhibit a high degree of similarity, with nearly identical H values extracted for all stations. While  $\kappa$  is not as robust as H, a slight variation is noticeable between manual and automated QC for stations CTSN, SHWN, and SHMN, falling within the acceptable error range. A systematic difference is observed in the calculated values when comparing these outcomes with a prior study conducted in the region (Thompson et al., 2010). This difference can be primarily attributed to dissimilarities in the assumed Vp (velocity of P-wave), with 6.5 km/s in Thompson et al. (2010) compared to 6.0 km/s in our study. Additionally, differences in processing methods, such as the utilization of different deconvolution techniques, may contribute to the observed differences.

### 6.1 Additional Test

Since our training and test sets were situated in geographically proximate regions with similar tectonic characteristics, we decided to validate our results by testing them on three stations located outside this area. The first station, ULM in Manitoba, is situated on Archean bedrock, while the second station, EDM in Edmonton, is located within the Western Canada Sedimentary Basin. The third station, PGC, is the farthest away, located in the southern part of Vancouver Island in proximity to the Cascadia Subduction Zone. At each station, we collected data spanning the past three years and subjected it to automated quality control using DeepRFQC. The results depicted in Figure 8, exhibit promising clarity, revealing distinct phases. Upon examining Figure 8.a, station EDM, one can observe a delayed P pulse, particularly for events farther from the stations (higher slowness), consistent with what we generally expect from a sedimentary basin. At station PGC, the Ps phase stands out prominently, while other phases are less discernible, attributable to the area's complex structure (Figure 8.b) due to subducting Juan de Fuca and Gorda plates along the entire Cascadia forearc (Bloch et al., 2023). At station ULM, both the Ps and Pps phases are clear, with the Pps phase becoming recognizable at around 18 seconds for slowness values lower than 6.25 s/km (Figure 8.c). It is also obvious from comparative plots that our model significantly outperformed Gong et al. (2022)'s model at all three of these stations. One of the reasons for this outperformance can be attributed to using data augmentation which enriches the train-

ing set with a good variety of waveforms. Data quantity and diversity are most important factors of a successful training process, which is why good augmentations can give rise to higher performance. Three additional tests were conducted to evaluate various deconvolution techniques, assess the effectiveness of quality controlling waveforms based solely on signal-to-noise ratio (SNR), and investigate the global applicability of DeepRFQC. For the deconvolution comparison, we applied both water-level and multitaper techniques to data recorded by the St. John's seismic station (SJNN) (Figure S7). Obviously, the algorithm works for both techniques while it is more efficient on water-level as it extracts more acceptable data. Figure S8 shows the results of using SNR for quality control on the same station. We observed that results became acceptable at an SNR threshold of 9, yielding 41 waveforms, while DeepRFQC retained 49 waveforms with more realistic results. Specifically, for waveforms around 50° back azimuth, the SNR method shows Ps arrival before 5 seconds, whereas DeepRFQC results indicate arrival slightly after 5 seconds. The latter is more realistic considering the later arrivals in other back azimuth ranges. The third test has been done on a 6-year data record from station ANTO (Ankara, Turkey). We applied DeepRFQC to RFs created using two different settings: a) bandpass filtered between 0.05 and 0.5 Hz and deconvolved by the water-level approach, and b) bandpass filtered between 0.05 and 1.0 Hz and deconvolved by the multitaper approach. The results shows that DeepRFQC not only can be used globally, but also can be used with other settings like different frequency ranges or deconvolution methods.

## 7 Conclusion

Our study introduces DeepRFQC, an automated quality control method inspired by the U-Net architecture, demonstrating notable success in evaluating teleseismic receiver function data. The method's effectiveness is evident through consistent and favorable results observed from the  $H-\kappa$  stacking process across most stations. This marks a significant stride in streamlining the workflow of teleseismic receiver function assessment. The study's findings present avenues for future research, particularly the need to address outliers and enhance the method's robustness. The integration of deep learning into automated quality control processes for teleseismic receiver functions yields significant benefits. It streamlines the most time-consuming aspect of this type of data analysis, resulting in enhanced efficiency. Furthermore, our model demonstrated superior performance compared to traditional, non-machine learning automated quality control metrics such as signal-to-noise ratio (SNR). Lastly, it establishes a standardized quality control procedure, ensuring a reliable means of preserving data integrity in seismological studies. This research contributes to the evolving landscape of automated quality control in seismology, emphasizing the significance of refining methods for greater adaptability across diverse datasets. The low-quality waveforms that are retained in RFs at some

stations using DeepRFQC emphasize the continuous need for improvement and adaptation of automated processes. Furthermore, it is imperative to note that the model exhibits peak performance when applied to datasets that have undergone water-level deconvolution and subsequent bandpass filtering within the frequency range of 0.05 to 0.5 Hz.

## 8 Acknowledgment

We express our gratitude to FDSN and Natural Resources Canada for providing seismic data crucial to this research. Special thanks to the developers of Python packages, including NumPy, Pandas, RFPy, and TensorFlow, for their essential contributions, enhancing the efficiency and success of our study. We also extend our sincere appreciation to the anonymous reviewers and the editor, whose insightful comments and constructive feedback significantly improved the quality and clarity of our research.

## 9 Open Research

Seismic data used here can be accessed from the Natural Resource Canada ([natural-resources.canada.ca/](https://natural-resources.canada.ca/)) and FDSN websites ([www.fdsn.org/](https://www.fdsn.org/)). Processing involved the use of TensorFlow (Abadi et al., 2015), RFPy (Audet, 2020), Numpy (Harris et al., 2020), Pandas (The Pandas development team, 2024), and Matplotlib. Additionally, we utilized GMT (Wessel et al., 2019) and Draw.io for their respective advantages to create plots and maps. Our codes are accessible on GitHub (publicly available) and Zenodo (Sabermahani and Frederiksen, 2023).

## References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. <https://www.tensorflow.org/>.
- Adler, A., Araya-Polo, M., and Poggio, T. Deep learning for seismic inverse problems: toward the acceleration of geophysical analysis workflows. *IEEE Signal Processing Magazine*, 38(2):89–119, 2021. doi: 10.1109/msp.2020.3037429.
- Ammon, C. J. A comparison of deconvolution techniques, 1992. United States.
- Audet, P. RFPy: Teleseismic receiver function calculation and post-processing (0.1.0), 2020. doi: 10.5281/zenodo.4302558.
- Audet, P., Schutt, D., Schaeffer, A. J., Cubley, J. F., Pellerin, L., and Buitier, S. J. H. Moho variations across the Northern Canadian Cordillera. *Seismological Research Letters*, 91(6), 2020. doi: 10.1785/0220200166.
- Basha, S. M. and Rajput, D. S. Aspects of deep learning: hyperparameter tuning, regularization, and normalization. In *Intelligent systems*, pages 171–188. Apple Academic Press, 2019.
- Bastow, I. D., Kendall, J., Helffrich, G., Thompson, D., Wookey, J., Brisbourne, A., and Snyder, D. B. The Hudson Bay lithospheric experiment. *Astronomy & Geophysics*, 52(6):6.21–6.24, 2011. doi: 10.1111/j.1468-4004.2011.52621.x.

- Bensen, G. D., Ritzwoller, M. H., Barmin, M. P., Levshin, A. L., Lin, F., Moschetti, M. P., Shapiro, N. M., and Yang, Y. Processing seismic ambient noise data to obtain reliable broad-band surface wave dispersion measurements. *Geophysical Journal International*, 169(3):1239–1260, 2007. doi: 10.1111/j.1365-246X.2007.03374.x.
- Bloch, W., Bostock, M. G., and Audet, P. A Cascadia Slab Model from Receiver Functions. *Geochemistry, Geophysics, Geosystems*, 24(10):e2023GC011088, 2023. doi: 10.1029/2023GC011088.
- Chang, D., Zhang, G., Yong, X., Gao, J., Wang, Y., and Wang, W. Deep learning using synthetic seismic data by Fourier domain adaptation in seismic structure interpretation. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. doi: 10.1109/lgrs.2022.3218911.
- Crotwell, H. P. and Owens, T. J. Automated Receiver Function Processing. *Seismological Research Letters*, 76(6):702–709, 2005. doi: 10.1785/gssrl.76.6.702.
- Dreiling, J., Tilmann, F., Yuan, X., Haberland, C., and Seneviratne, S. W. M. Crustal Structure of Sri Lanka Derived from Joint Inversion of Surface Wave Dispersion and Receiver Functions Using a Bayesian Approach. *Journal of Geophysical Research: Solid Earth*, 125(5):e2019JB018688, 2020. doi: 10.1029/2019JB018688.
- Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, 2010.
- Gong, C., Chen, L., Xiao, Z., and Wang, X. Deep learning for quality control of receiver functions. *Frontiers in Earth Science*, 2022. doi: 10.3389/feart.2022.921830.
- Hansen, S. and Schmandt, B. P and S wave receiver function imaging of subduction with scattering kernels. *Geochemistry Geophysics Geosystems*, 18(12):4487–4502, 2017. doi: 10.1002/2017gc007120.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., and ... Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2.
- He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. doi: 10.1109/ICCV.2015.123.
- Iwana, B. K. and Uchida, S. An empirical survey of data augmentation for time series classification with neural networks. *Plos One*, 16(7):e0254841, 2021. doi: 10.1371/journal.pone.0254841.
- Kikuchi, M. and Kanamori, H. Inversion of complex body waves. *Bulletin of the Seismological Society of America*, 72(2):491–506, 1982. doi: 10.1785/BSSA0720020491.
- Kind, R., Mooney, W. D., and Yuan, X. New insights into the structural elements of the upper mantle beneath the contiguous United States from S-to-P converted seismic waves. *Geophysical Journal International*, 222(1):646–659, 2020. doi: 10.1093/gji/g-gaa203.
- Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2014. doi: 10.48550/arXiv.1412.6980.
- Krueger, H. E., Gama, I., and Fischer, K. M. Global patterns in cratonic mid-lithospheric discontinuities from Sp receiver functions. *Geochemistry, Geophysics, Geosystems*, 22(6), 2021. doi: 10.1029/2021GC009819.
- Lawrence, J. and Shearer, P. A global study of transition zone thickness using receiver functions. *Journal of Geophysical Research Atmospheres*, 111(B6), 2006. doi: 10.1029/2005jb003973.
- Ligorria, J. P. and Ammon, C. J. Iterative deconvolution and receiver-function estimation. *Bulletin of the Seismological Society of America*, 89(5):1395–1400, 1999. doi: 10.1785/BSSA0890051395.
- Mousavi, S., Sheng, Y., Zhu, W., and Beroza, G. Stanford earthquake dataset (STEAD): a global dataset of seismic signals for AI. *IEEE Access*, 7:179464–179476, 2019. doi: 10.1109/access.2019.2947848.
- Mukkamala, M. C. and Hein, M. Variants of RMSProp and Adagrad with Logarithmic Regret Bounds. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2545–2553, 2017. <https://proceedings.mlr.press/v70/mukkamala17a.html>.
- Natural Resources Canada. Canadian National Seismograph Network. [Data set]. Natural Resources Canada, 1975. doi: 10.7914/SN/CN.
- Park, J. and Levin, V. Receiver functions from multiple-taper spectral correlation estimates. *Bulletin of the Seismological Society of America*, 90(6):1507–1520, 2000. doi: 10.1785/0119990122.
- Ramadanti, F. Preliminary results on receiver function study in Mt. Merapi, Central Java, Indonesia, 2023. doi: 10.1088/1755-1315/1227/1/012049.
- Rodriguez, E. E. and Russo, R. M. Southern Chile crustal structure from teleseismic receiver functions: Responses to ridge subduction and terrane assembly of Patagonia. *Geosphere*, 16(1): 378–391, 2020. doi: 10.1130/GES01692.1.
- Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N., Hornegger, J., Wells, W., and Frangi, A., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, Cham, 2015. doi: 10.1007/978-3-319-24574-4\_28.
- Sabermahani, S. and Frederiksen, A. DeepRFQC: automating quality control for P-wave receiver function analysis using a U-net inspired network, 2023. doi: 10.5281/zenodo.10087652.
- Schimmel, M. and Paulssen, H. Noise reduction and detection of weak, coherent signals through phase-weighted stacks. *Geophysical Journal International*, 130(2):497–505, 1997.
- Scripps Institution of Oceanography. IRIS/IDA Seismic Network. International Federation of Digital Seismograph Networks. Other/Seismic Network, 1986. doi: 10.7914/SN/II.
- Shen, W., Ritzwoller, M. H., Schulte-Pelkum, V., and Lin, F.-C. Joint inversion of surface wave dispersion and receiver functions: A Bayesian Monte-Carlo approach. *Geophysical Journal International*, 192(2):807–836, 2013. doi: 10.1093/gji/ggs050.
- Shorten, C. and Khoshgoftaar, T. M. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(60), 2019. doi: 10.1186/s40537-019-0197-0.
- Snyder, D. B., Craven, J. A., Pilkington, M., and Hillier, M. J. The 3-dimensional construction of the Rae craton, central Canada. *Geochemistry, Geophysics, Geosystems*, 16(10):3555–3574, 2015. doi: 10.1002/2015GC005957.
- Sokolova, M., Japkowicz, N., and Szpakowicz, S. Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. In Sattar, A. and Kang, B. H., editors, *AI 2006: Advances in Artificial Intelligence*, volume 4304 of *Lecture Notes in Computer Science*, pages 1015–1021. Springer, Berlin, Heidelberg, 2006. doi: 10.1007/11941439\_114.
- The ObsPy Development Team. ObsPy 1.4.0 (1.4.0), 2022. doi:

10.5281/zenodo.6645832.

The Pandas development team. Pandas-dev/pandas: Pandas (v2.2.2), 2024. doi: 10.5281/zenodo.10957263.

Thompson, D. A., Bastow, I. D., Helffrich, G., Kendall, J.-M., Wookey, J., Snyder, D. B., and Eaton, D. W. Precambrian crustal evolution: Seismic constraints from the Canadian Shield. *Earth and Planetary Science Letters*, 297(3-4):655–666, 2010. doi: 10.1016/j.epsl.2010.07.021.

Vinnik, L. P. Detection of waves converted from P to SV in the mantle. *Physics of the Earth and Planetary Interiors*, 15(1):39–45, 1977. doi: 10.1016/0031-9201(77)90008-5.

Vinnik, L. P., Reigber, C., Aleshin, I. M., Kosarev, G. L., Kaban, M. K., Oreshin, S. I., and Roecker, S. W. Receiver function tomography of the central Tien Shan. *Earth and Planetary Science Letters*, 225(1-2):131–146, 2004. doi: 10.1016/j.epsl.2004.05.039.

Wessel, P., Luis, J. F., Uieda, L., Scharroo, R., Wobbe, F., Smith, W. H. F., and Tian, D. The Generic Mapping Tools Version 6. *Geochemistry, Geophysics, Geosystems*, 20(11), 2019. doi: 10.1029/2019GC008515.

Whitmeyer, S. and Karlstrom, K. E. Tectonic model for the Proterozoic growth of North America. *Geosphere*, 3(4):220–259, 2007. doi: 10.1130/GES00055.1.

Wilson, D. C. and Aster, R. C. Seismic imaging of the crust and upper mantle using regularized joint receiver functions, frequency–wave number filtering, and multimode Kirchhoff migration. *Journal of Geophysical Research: Solid Earth*, 110(B5), 2005. doi: 10.1029/2004jb003430.

Yang, X., Pavlis, G. L., and Wang, Y. A Quality Control Method for Teleseismic P-Wave Receiver Functions. *Bulletin of the Seismological Society of America*, 106(5):1948–1962, 2016. doi: 10.1785/0120150347.

Zanjani, A. A., Zhu, L., Herrmann, R. B., Liu, Y., Gu, Z., and Conder, J. A. Crustal Structure Beneath the Wabash Valley Seismic Zone from the Joint Inversion of Receiver Functions and Surface-Wave Dispersion: Implications for Continental Rifts and Intraplate Seismicity. *Journal of Geophysical Research: Solid Earth*, 124(7):7028–7039, 2019. doi: 10.1029/2018JB016989.

Zhu, L. and Kanamori, H. Moho depth variation in southern California from teleseismic receiver functions. *Journal of Geophysical Research*, 105(B2):2969–2980, 2000. doi: 10.1029/1999JB900322.

Zhu, W. and Beroza, G. C. PhaseNet: a deep-neural-network-based seismic arrival-time picking method. *Geophysical Journal International*, 216(1):261–273, 2019. doi: 10.1093/gji/ggy423.

The article *DeepRFQC: automating quality control for P-wave receiver function analysis using a U-net inspired network* © 2024 by Sina Sabermahani is licensed under CC BY 4.0.