Machine learning based denoising for waveform similarity analysis

Bachelor Thesis

Levin Stiehl (5396585)

Supervisors: Dr. Jonas Folesky, Dr. Jörn Kummerow

Geologische Wissenschaften
Freie Universität Berlin
Geophysik

23.10.2023

Zusammenfassung

Viele Felder der geologischen Wissenschaften nutzen seismische Signale, um wichtige Informationen über den Untergrund zu erhalten. Ihre Analyse kann allerdings aufgrund von Rauschen schwierig sein. Die präzise Identifizierung der Eigenschaften seismischer Signale stellt daher eine anspruchsvolle Aufgabe dar. Dieses Problem ist besonders ausgeprägt, wenn das Signal ein geringes Signal-Rausch-Verhältnis hat. Eine neu entwickelte Methode für diesen Zweck ist DeepDenoiser, ein neuronales Netzwerk, das Deep-Learning-Techniken einsetzt, um Rauschen von seismischen Signalen zu entfernen.

Diese Arbeit konzentriert sich auf die Bewertung der Anwendbarkeit von DeepDenoiser im Rahmen der Wellenform-Ähnlichkeitsanalyse. Ihr spezieller Fokus liegt auf der Klassifizierung wiederkehrender seismischer Ereignisse, die oft als ”Repeater” bezeichnet werden. Die Identifizierung von Repeater ist entscheidend, da sie Einblicke in die Aktivitäten von Störungszonen in großen Tiefen bieten. Um dies zu erreichen, zielt die Studie darauf ab, die Fähigkeiten und Grenzen von DeepDenoiser durch die Reproduktion der Originalergebnisse anhand seismischer Daten aus Nordchile (IPOC) umfassend zu verstehen. Die Untersuchung beinhaltet das Überlagern von Rauschen auf seismische Signale, um einen Datensatz von synthetischen Erdbeben zu erstellen, wobei Signal-Rausch-Verhältnisse, maximale Amplituden und Kreuzkorrelationsverbesserungen nach dem Entrauschen verglichen werden. Anschließend wird der Einfluss von DeepDenoiser auf Kreuzkorrelationsmessungen bei sich wiederholenden Erdbeben Sequenzen bewertet. Durch die Anwendung von DeepDenoiser auf identifizierte Repeater wird untersucht, wie sich die Kreuzkorrelationskoeffizienten in Bezug auf die Entfernung ändern.

Die Studie hebt die erhebliche Wirksamkeit der Rauschunterdrückung mittels maschinellem Lernen hervor. Es zeigt sich jedoch, dass die Nützlichkeit von DeepDenoiser von kontextuellen Nuancen abhängt. Während es in der Verbesserung von Kreuzkorrelationswerten für niedrige Signal-Rausch-Verhältnisse in synthetischen Daten versiert ist, überträgt sich dieser Erfolg nicht einheitlich auf authentische Repeaters. Ein signifikanter Anteil von 40% aller Repeater Sequenzen werden durch DeepDenoiser verbessert. Insbesondere Repeater außerhalb des Hauptüberwachungsgebietes von 70°W - 69°W zeigen jedoch selten Verbesserungen durch DeepDenoiser. Die Kreuzkorrelationswerte ändern sich um etwa ±0,1. Diese nuancierte Beobachtung unterstreicht die inhärente Komplexität in der Wechselwirkung verschiedener Faktoren. Eine universelle Empfehlung für die Verwendung von DeepDenoiser zur Identifizierung von Repeater kann daher nicht gegeben werden und ein neues Trainieren ist eventuell angebracht.

Abstract

Seismic signals, which carry important subsurface information, can be difficult to distinguish due to the presence of seismic noise. Precise identification of seismic signal characteristics can be particularly challenging, and this problem becomes especially pronounced when dealing with low signal-to-noise ratios.

A recent method addressing this issue is DeepDenoiser, a neural network that uses deep learning techniques to remove noise from seismic signals. This research is centred on a thorough assessment of DeepDenoiser's potential within the framework of waveform similarity analysis. Its specific focus is directed towards the categorization of repeating seismic events, commonly referred to as "repeaters". Identifying repeaters is vital, as they offer insights into fault zone activity at great depths. To achieve this, the study aims to comprehensively understand the capabilities and limitations of DeepDenoiser by replicating its results using seismic data from northern Chile (IPOC). The investigation involves superimposing noise on seismic traces to create a synthetic dataset, comparing signal-to-noise ratios, maximum amplitudes, and cross-correlation improvements post-denoising. Subsequently, the research evaluates DeepDenoiser's impact on cross-correlation measurements for repeating earthquake sequences. By applying DeepDenoiser to identified repeaters, the study investigates how cross-correlation coefficients change concerning distance.

The study highlights the considerable effectiveness of machine learning-based denoising, but it is apparent that the usefulness of DeepDenoiser is dependent on contextual nuances. While it is proficient in improving cross-correlation values for low signal-to-noise ratios in synthetic data, this success does not translate uniformly to authentic repeater data. A significant 40% of all repeating earthquake sequences were improved through DeepDenoiser. Repeaters located outside the 70°W - 69°W main monitored region in particular do not consistently show improvements through DeepDenoiser. The cross-correlation values alter by around ±0.1. This nuanced observation underscores the inherent complexity involved in the interaction of various factors. Formulating a universal recommendation for using DeepDenoiser to identify repeaters remains a challenging task and retraining might be necessary.

Table of Contents

1. Introduction

In the field of seismology, the extraction of meaningful seismic signals is a frequent challenge due to the intricate interplay between signal and noise. Seismic signals, crucial carriers of subsurface information, often find themselves intertwined within a backdrop of seismic noise. This problem becomes particularly significant when dealing with low signal-to-noise ratios (SNR), which can make seismic signals unidentifiable (Bormann and Wielandt, 2013). Hence, there is a pressing need to develop more advanced techniques to mitigate this problem. One recent method for addressing this challenge is DeepDenoiser (Zhu et al., 2019), a neural network that uses deep learning techniques to remove noise from seismic signals.

Waveform similarity analysis, for example, is applied in the classification of repeating earthquakes (repeaters, RE). These repeaters represent recurring releases of seismic energy from distinct structures sharing the same location. Cross-correlation (cc) is one of the methods used for identifying repeaters (Uchida and Bürgmann, 2019). Among other applications, repeaters may be used to measure the average slip rate of a fault zone at great depth by measuring their occurrence times and cumulative slip (see Figure 2.1; Uchida, 2019).

However, when not all or faulty repeaters in a repeating earthquake sequence (RES) are correctly identified, the calculations become erroneous. This can happen if the event is disturbed by a large amount of noise. The accurate identification of RE is therefore very important. For this reason, I test the application of DeepDenoiser with regard to denoising for waveform similarity analysis on the example of RE.

The objectives of this thesis are twofold. The primary goal of this thesis is to gain a deeper understanding of DeepDenoiser's behaviour, limitations and features. For this, I recreate the results from the original publication using data from northern Chile. By superimposing noise traces on seismic traces I produce a new synthetic dataset and compare the SNR, maximum amplitude and cc improvement with DeepDenoiser.

The second goal is to evaluate DeepDenoiser's application for waveform similarity analysis. Noise may affect cross-correlation in measuring similarities between two different events. Nevertheless, cc is essential when assessing repeaters. DeepDenoiser will be applied to events classified as RE. I compute the cc for different RES at several stations of different distance. Creating a dataset of cross-correlation coefficient values in relation to distance and the hypocenter's location. In particular, I will focus on the question of how the cc for those events reacts if computed after denoising. Is there a clear pattern? Can I improve detection, or is there a more complex systematic approach?

If DeepDenoiser proves effective in improving the identification of RE, it has the potential to be of great benefit to future studies in this field.

The focus of this thesis is limited to the originally trained version of DeepDenoiser.

1.1. Data and Regional Setting

The study region is located in northern Chile. In this area, the Nazca and South American plates converge, resulting in the subduction of the Nazca plate beneath the South American plate at a rate of 68 mm/year, with a direction of N76°E. This convergence occurs within a boundary zone spanning 500-1000 km, generating the towering peaks and active volcanoes in the Andes region (Norabuena et al., 1998).

Subduction zone forearcs are the most seismically active regions worldwide with regard to both seismic moment release and seismicity rate (Sippl et al., 2018). These areas are renowned for producing powerful earthquakes, such as the Iquique earthquake with M_w 8.1 (Hayes et al., 2014).

At the subduction interface, where two tectonic plates affect each other, most of the seismic activity takes place. Earthquakes are also recorded in the continental crust and expand into a deeper zone about 20-25 km below the interface, in the oceanic mantle (Folesky et al., 2021). The selected RES include examples out of all of these areas.

Seismic data from a network of 23 broadband landstations are used in this thesis, managed by the Integrated Plate Boundary Observatory Chile (IPOC). IPOC is a joint effort between European and South American institutions and researchers who organise and operate a distributed network of instruments and research initiatives focused on studying earthquakes and deformation along the Chilean continental margin. This extensive network spreads over 700 kilometres from 17.6°S to 24.6°S, covering a vast area (IPOC, 2018).

The dataset was recorded over a period of 14 years from 2007-2021, recording over 180,000 events (Sippl et al., 2023). The EIDA web service of GFZ Potsdam provided the corresponding 100 Hz three-component waveform data (GFZ German Research Centre for Geosciences and Institut des Sciences de l'Univers-Centre National de la Recherche CNRS-INSU, 2006).

The Institute of Geological Sciences' General and Applied Seismology Group, located in Berlin, identified 2254 RES. Out of this catalogue, 56 RES containing 10-30 events each were selected. The median location of the chosen RES is shown in Figure 1.1. The magnitudes of the events range from $M_w = 1.0$ to $M_w = 3.1$.

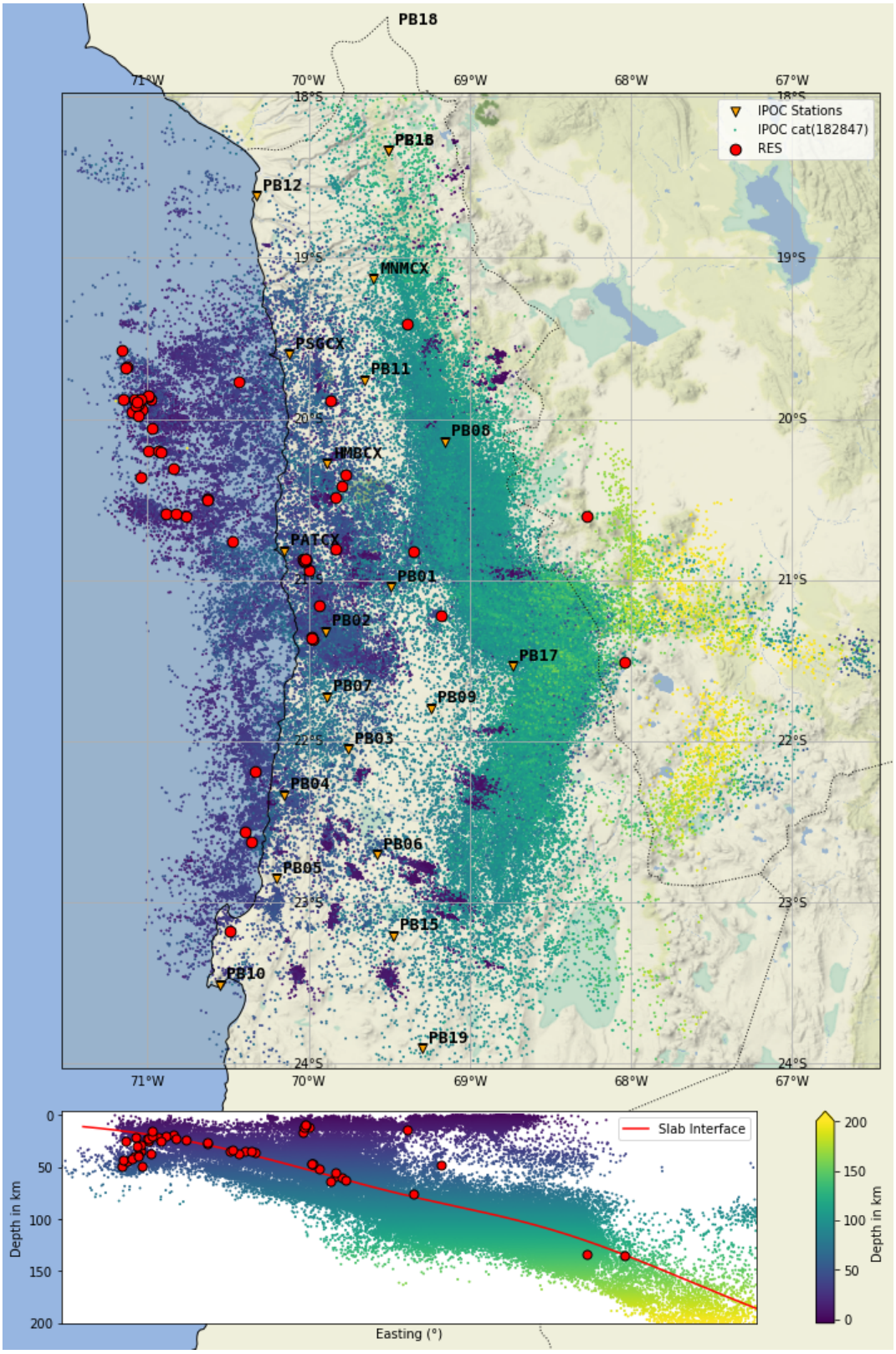


Figure 1.1.: Map of the surveyed region in northern Chile. The 56 RES used in this thesis are marked in red. Events from the catalogue of Sippl et al., 2023 are colour coded according to their depth. The 23 IPOC stations are plotted in orange. Figure altered after Dr. Folesky (personal communication, 2023).

2. Theoretical Background

2.1. Repeating Earthquakes

Repeating earthquakes, also known as repeaters, share the same location, but have different times of occurrence. Recurring seismic energy releases from distinct structures, such as slip on a fault patch, seems to be the cause of repeating earthquakes. Repeating earthquakes can be observed along many plate boundary fault systems, but are most common on creeping plate boundary faults, where seismic patches are loaded by surrounding slow slip, as shown in Figure 2.1 (Uchida and Bürgmann, 2019).

There is no standard definition of repeaters, but two distinct methods are commonly used to identify repeaters: either by precisely determining the hypocenter to detect actual colocation, or by confirming the similarity of the waveforms. Identifying repeaters by event location usually requires a minimum overlap of 50-70%. However, due to limited station coverage or timing accuracy, this method is not always possible. Waveform similarity analysis typically uses the cross-correlation coefficient and coherence. To qualify as a repeater, the cross-correlation coefficient thresholds typically range from 0.90 to 0.98, while coherence is between 0.95 to 0.98 (Uchida and Bürgmann, 2019).

Strict criteria ensure that only genuine repeating events are included, but may miss events due to changes in rupture propagation, noise, or uncertainties in source area and location. A more loose threshold can include more true repeating events, but may also include some non-repeating events. To achieve the right balance of thresholds, a combination of criteria may also be used (Lengliné and Marsan, 2009).

Therefore, it is advisable to consider the criteria and thresholds used to identify repeaters in a catalogue.

RE has numerous applications. In general, they can be used to estimate changes in temporal properties of fault zones, volcanic systems or Earth's core. Moreover, they can assist in interpreting laboratory experiments and informing studies of earthquake predictability (Uchida and Bürgmann, 2019). The average slip rate of a fault zone at a great depth can be measured with the help of repeaters by measuring their occurrence times and cumulative slip (see Figure 2.1; Uchida, 2019). Linking earthquake activity with spatial and temporal changes in the slip rate (Nadeau and McEvilly, 1999).

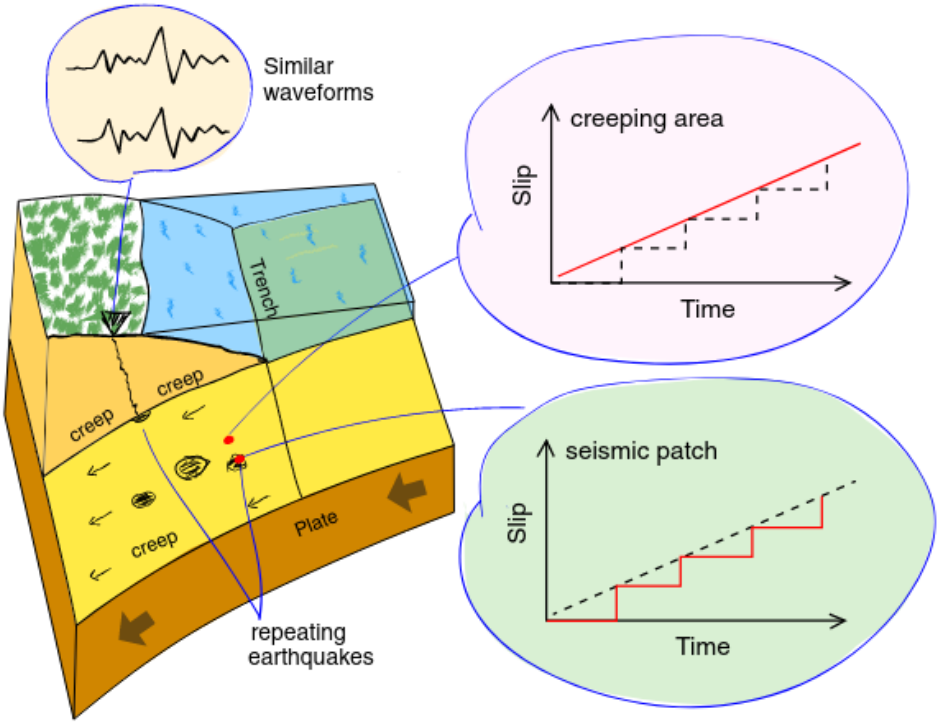


Figure 2.1.: Schematic representation of the process of repeating earthquakes in a subduction zone. The seismic patches (black spots) are loaded by the creep of the plate boundary, producing repeating earthquakes with similar waveforms. As the creeping area and the seismic patch are adjacent to each other, the long-term cumulative slip remains the same, despite potential slip in the interseismic period. This figure was created by Uchida, 2019.

2.2. Signal-to-Noise Ratio

The quality of seismic signals can be measured objectively using the signal-to-noise ratio (SNR). This quantifies the strength of the seismic signal compared to background noise and is expressed as a ratio. The SNR in decibels (dB) can be expressed as:

$$SNR = 10 \log_{10} \frac{\sigma_{signal}}{\sigma_{noise}}, \tag{2.1}$$

where σ_{signal} is the standard deviation of waveforms after the first arrival and σ_{noise} is the standard deviation of waveforms before the first arrival (Zhu et al., 2019).

2.3. Cross-Correlation

The cross-correlation coefficient $C(\tau)$ serves as a similarity measure for two time series, f_x and f_y, consisting of N discrete samples. As described in section 2.1, it was used for example to detect repeating earthquakes. cc at a particular lag time τ can be stated as (Uchida and Bürgmann, 2019):

$$C(\tau) = \frac{1}{N} \sum_{t=1}^{N} f_x(t) f_y(t+\tau). \tag{2.2}$$

The maximum lag time used for the analysis determines how much $f_y(t+\tau)$ is shifted in comparison to $f_x(t)$. This is done to find the lag time with the highest cross-correlation. The length of the time series may also affect the cross-correlation. The longer the time window, the less likely a high cross-correlation is to occur (Uchida, 2019).

However, since the signal is represented in the discrete-time domain, which is not computationally convenient, there is often a need to transform it into the frequency-domain. There are two methods for doing this: the Discrete Fourier Transform (DFT) and the Fast Fourier Transform (FFT). The DFT is a sequence used to transform a discrete-time sequence of finite length into a discrete-frequency sequence of finite length (Anand, 2022, p. 195).

FFT includes all algorithms that are computationally more efficient versions of DFT. They achieve this by exploiting the symmetry and periodicity properties of the phase factor (Anand, 2022, p. 247).

Obspy's function for cross-correlation, 'obspy.signal.cross_correlation.correlate', chooses automatically the fastest method between direct time domain cross-correlation and FFT cross-correlation to compute the cross-correlation function (Team, 2022).

2.4. DeepDenoiser

DeepDenoiser is a denoising method based on a deep neural network (DNN) that claims to be able to separate the signal of interest from the noise and improve various applications, such as event detection.

It was trained and tested using 30-second seismograms from the HN* broadband channels of North California Seismic Network (NCSN). The dataset consists of 56,345 earthquake waveforms with high SNRs as signals and 179,233 seismograms with non-earthquake waveforms as noise. Both datasets underwent random division into training, validation, and test sets. Noisy seismograms for training were created by adding noise samples to the signals. The deep neural network receives input from the short-time Fourier transform (STFT), accessing both the real and imaginary parts of the data. The DeepDenoiser network generates prediction targets of signal and

noise masks. Validation sets are employed for optimising hyperparameters to prevent over-fitting. Test sets are used to assess the performance of the network (Zhu et al., 2019).

DeepDenoiser's architecture is a convolutional neural network (CNN), consisting of 22 fully convolutional layers with descending and then ascending sizes. Convolutional neural networks are typical used for image analysis and are great at detecting patterns, their ability to reduce the number of required neurons has great computationally advantage (Aghdam and Heravi, 2017). Every convolutional layer detects different patterns. It filters the input for that specific pattern and feed-forward the result (feature map) to the next layer. The convolutional layer dimension is noted in frequency bins × time points × channels. The number of filters in a layer determines the number of channels in the output. The training procedure sets the weights of the filters (Zhu et al., 2019).

A rectified linear unit (ReLU) is a basic yet effective non-linear activation function. It sets all negative values of a feature map to zero, producing a non-linearity (Aghdam and Heravi, 2017).

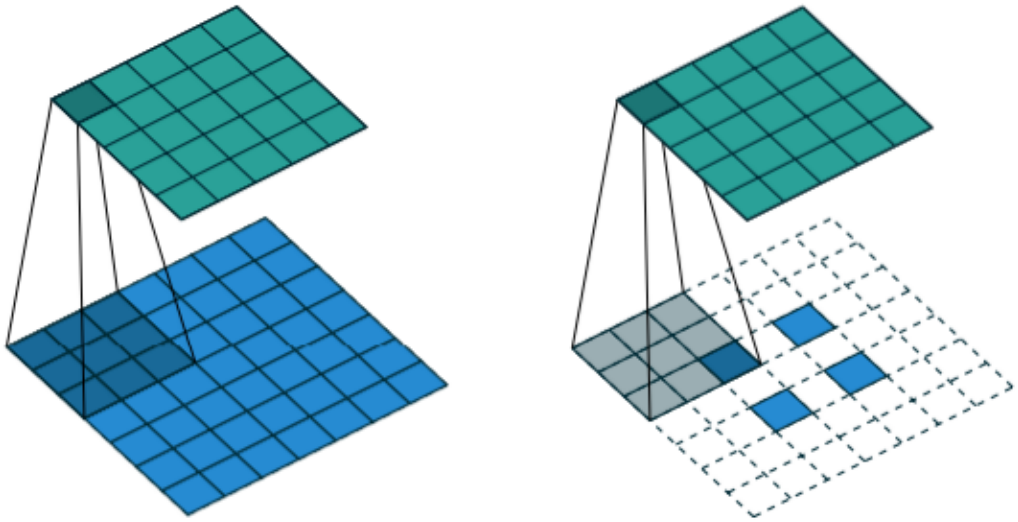


Figure 2.2.: The two different inputs (blue tiles) of 7x7 and 2x2 both generate a 5x5 feature map (green tiles). In one case, the feature map is resized to be larger and in the other case, the feature map is resized to be smaller. Convolution comprises the information within the 3x3 filter (darker tiles) into a singular tile, leading to a reduction in dimensions corresponding to the strides (convolution steps). Deconvolution expands the input by adding zeros as padding (white tiles), resulting in an output larger than the original input. This figure was created by Dumoulin and Visin, 2018.

Because strides specify the convolution steps, they determine by how much the

dimension of the feature map shrinks, but not the number of feature maps. Therefore, a 2x2 stride decreases the image dimension from 4x26x64 to 2x13x64 (Aghdam and Heravi, 2017).

Skip connections enable the layer's output to be directed not only to the next layer but also to the layer lower down the network. Resolving the issue of the vanishing gradient and improving convergence during training (Adaloglou, 2020).

Batch normalisation increases the learning rates by minimising changes in the distribution of network activations resulting from the change in network parameters during training (Ioffe and Szegedy, 2015).

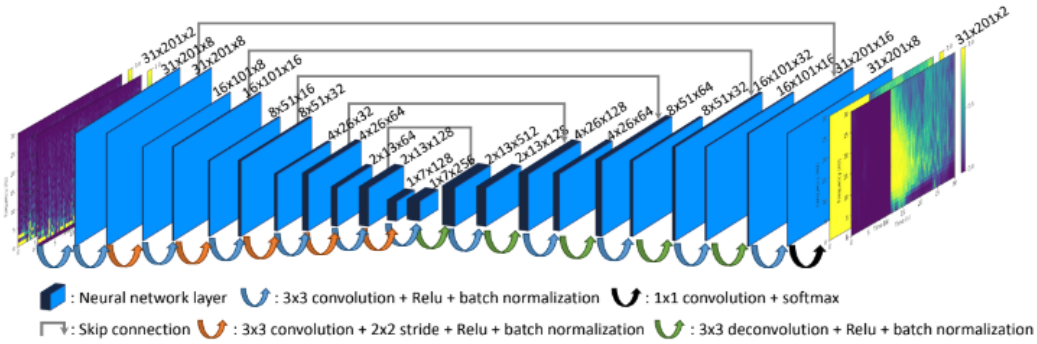


Figure 2.3.: DeepDenoiser DNN architecture. Both the real and imaginary components of the time-frequency representation of noisy data are used as the input. The signal mask and noise mask are both outputs of the DNN. The network layers are displayed as blue rectangles, with their dimensions shown as frequency bins x time points x channels. The different kinds of operations are indicated by arrows. The input data undergoes downsampling via 3x3 convolution layers with 2x2 strides before passing through upsampling deconvolution layers. Skip connections and batch normalisation are utilised to enhance convergence during training. In the final layer, a softmax normalized exponential function is utilised to predict the signal and noise masks. This figure was created by Zhu et al., 2019.

The entire DeepDenoiser noise reduction process is briefly described as follows. To denoise noisy data in the time domain, it is first transformed into the time-frequency domain using STFT. The data is then input into the DNN, resulting in the production of a noise mask and a signal mask. These masks are subsequently applied to the noisy data to filter out the seismic signal and noise. Finally, the data is transformed back into the time domain by using the inverse STFT (Zhu et al., 2019).

DeepDenoiser was showcased by improving earthquake detection.

To avoid error-prone results, the denoised waveform requires a component of pure noise. Analysing waveforms composed solely of signals contributes to errors.

I will use the term ”denoised signal” to refer to signals that are filtered by DeepDenoiser, and ”raw/noisy” signal to refer to signals that are not filtered by DepDenoiser, even if a high-pass or a band-pass filter is applied.

2.5. EQTransformer

EQTransformer (Mousavi et al., 2020) is a deep learning model that aims to simultaneously detect earthquake signals and identify P and S phases in seismic data, by utilising a hierarchical attentive model. It has a multi-task architecture, using a combination of recurrent and convolutional layers. STanford EArthquake Dataset (STEAD), which offers a significant amount of 1 million labelled earthquake and 300,000 non-earthquake signals (ambient and cultural noise), was used to train EQ Transformer. STEAD incorporates earthquake waveforms from diverse geographical regions and tectonic settings.
EQTransformer is intended to be used for single seismic stations' data at close distances (<300 km) and displays better performance for seismic signal identification and first P and S phase picking compared to other methods such as Yews, GPD, PickNet, and others (Mousavi et al., 2020).

3. Synthetic Data

To gain a better understanding of DeepDenoiser's behaviour, limitations, and characteristics, I try to replicate the results of Zhu et al., 2019 using a similar approach to generate the synthetic data, but with a different dataset.

3.1. Data and Processing

Twenty-one 30-second seismic traces are used as the database, which were recorded by the 100 Hz high broad-band channels (HH*) of the IPOC Seismic Network. Consisting of eleven earthquake waveforms, each with a very high SNR, as the signal samples and ten non-earthquake waveforms which were used as the noise samples. A high-pass filter of 0.2 Hz is applied to all data.

The goal is to transform these seismic signals and noise into a new synthetic earthquake signal with specific signal-to-noise ratios by superimposing noise traces on seismic traces (Figure 3.1), creating a new synthetic dataset for analysis.

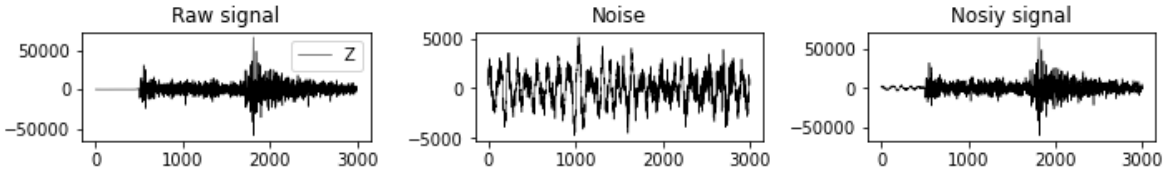


Figure 3.1.: Example of adding noise to a Z-component waveform, (left) high SNR signal, (middle) noise, (right) noisy signal (combination of high SNR signal and noise).

By rearranging Equation 2.1 for σ_{noise} (σ_{target}), I can determine the necessary noise adjustments (x_{factor}) to achieve the desired signal-to-noise ratio for the noisy signal:

$$\sigma_{target} = \frac{\sigma_{signal}}{10^{\frac{SNR}{10}}} \tag{3.1}$$

$$x_{factor} = \frac{\sigma_{target}}{\sigma_{noise}}. \tag{3.2}$$

The value of σ_{target} refers to the necessary standard deviation of the noise. SNR is the desired signal-to-noise ratio for the noisy signal. σ_{signal} is the calculated standard deviation of the signal. σ_{noise} is the calculated standard deviation of the noise. The x_{factor} represents the multiplier for the noise trace and determines how much the noise needs to be amplified or reduced to achieve the desired synthetic earthquake signal.

This process is carried out for every combination of signal and noise, covering all 44 SNR values. The value of each data point is the average of 110 different events.

The SNR is calculated using the Equation 2.1. The standard deviation of the noise is calculated using the first five seconds of the noisy signal prior to the initial arrival. The signal's standard deviation is calculated using the 25 seconds of the noisy signal following the first arrival.

To determine the normalised maximum amplitude change, the difference between the absolute maximum amplitude of the noisy signal and the absolute maximum amplitude of the reference signal (raw/denoised raw signal) is divided by the absolute maximum amplitude of the reference signal (raw/denoised raw signal).

I calculate the maximum correlation between the noisy data and the reference data (raw/denoised raw data) using the 'obspy.signal.cross_correlation.correlate' (Team, 2022) function with zero-lag and normalisation by the overall standard deviation. The initial five seconds of the traces are disregarded to solely incorporate the event into the calculation.

The denoised data follows the same procedure as the noisy data, except that it undergoes additional filtering via DeepDenoiser.

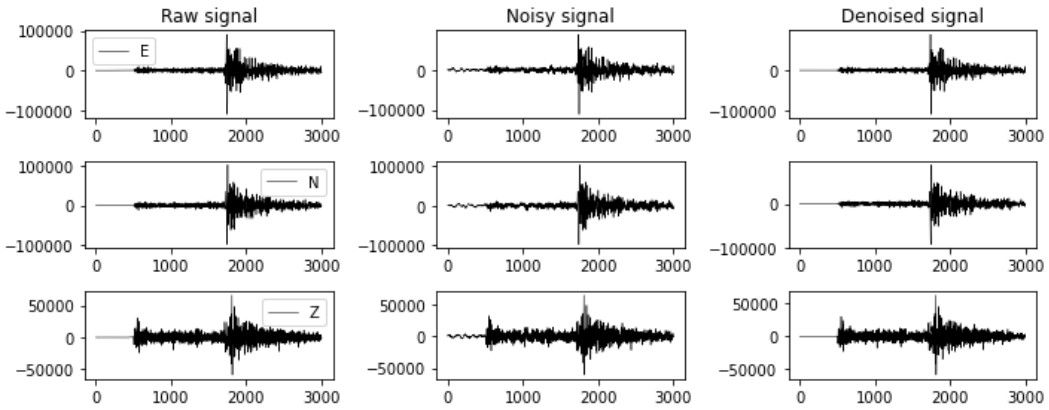


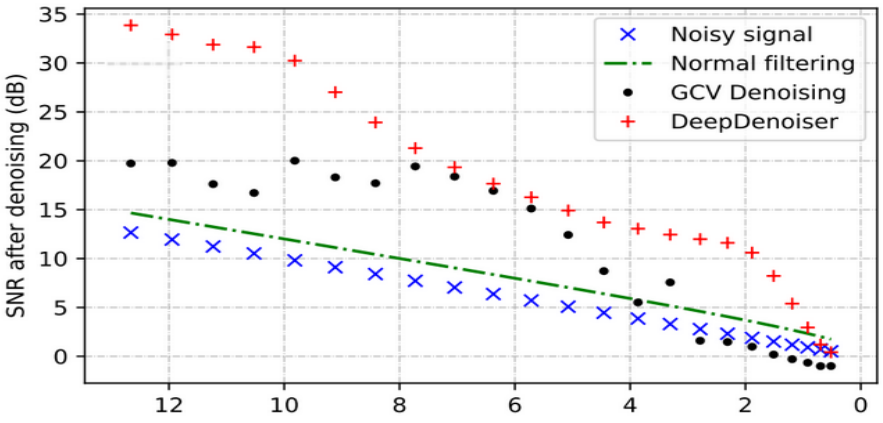
Figure 3.2.: Example of creating the synthetic earthquake signals both for noisy and denoised data, (left) high SNR signal, (middle) noisy signal (combination of high SNR signal and noise), (right) denoised signal.

3.2. Results

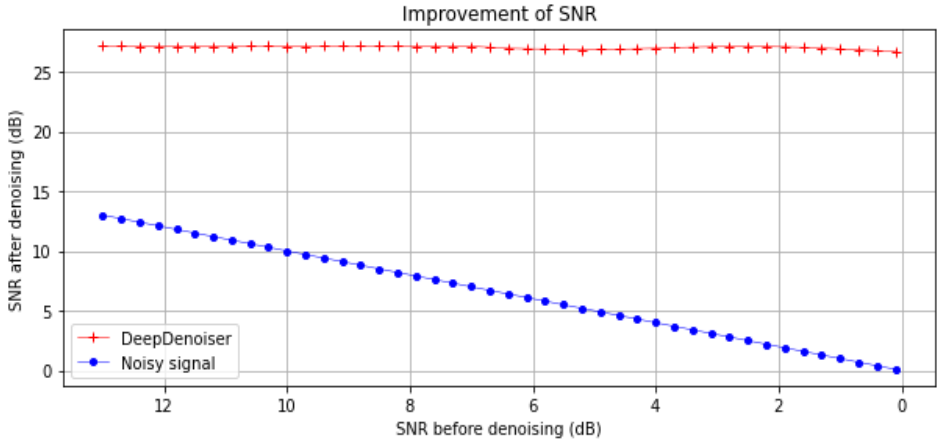
I measure the SNR improvements between the denoised signals and noisy signals, as well as changes in the maximum amplitude and the cross-correlation coefficient for signals with an SNR of 0.1 - 13 dB in 0.3 dB steps (Figure 3.3 - 3.5). There are two different reference points used to evaluate DeepDenoiser in Figure 3.4 and 3.5. In the first case (plus marker), the original signal is used as the reference. In the second case (dot marker), the reference signal undergoes denoising using DeepDenoiser. Because even though I only use high SNR signals they still have some noise included. So if theoretical DeepDenoiser would perfectly separate noise and signal there would still be a slight discrepancy in the data shown. Also I can analyse at what point SNR improvement has no effect on the results given by DeepDenoiser.

3.2.1. Signal-to-Noise Ratio

Figure 3.3 shows that DeepDenoiser significantly improves the SNR even if the original signal has very low SNR. Though there is no stronger improvement at higher SNRs, the signal is boosted to an SNR of about 27 dB. The outcomes are notably more effective for SNRs under 9 dB, especially in the very low SNR spectrum, compared to the original paper. However, in the original, there is a steady improvement with increasing SNR that surpasses my results for higher SNR.



(a) original



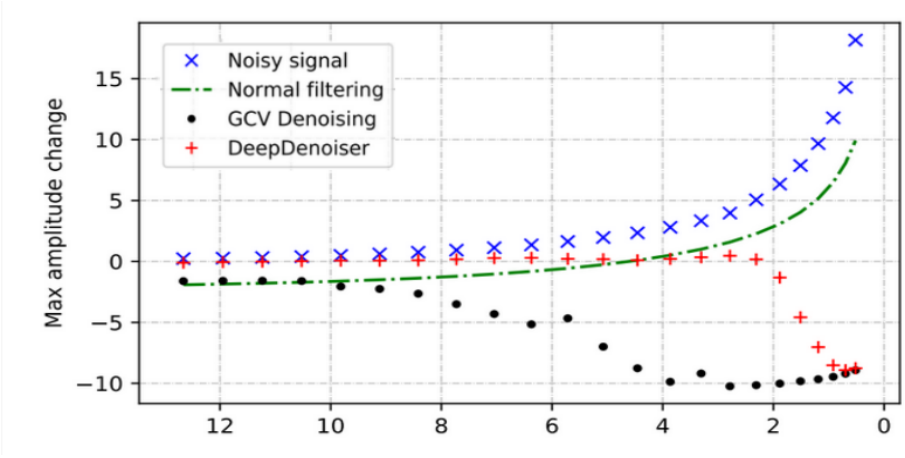
(b) new

Figure 3.3.: Comparisons between the original results (a) of Zhu et al., 2019 and the newly replicated results (b) are shown in terms of SNR improvement after applying DeepDenoiser. For each SNR between 0.1 and 13 dB, 110 synthetic earthquakes with exactly that SNR are created in 0.3 dB steps. The average SNR of all 110 synthetic earthquakes before (noisy data, blue dots) and after (DeepDenoiser, red pluses) denoising with DeepDenoiser represents each data point. GCV denoising (black dots) and normal filtering (green dash-dotted line) from the original graph are not relevant in this comparison and can be disregarded.

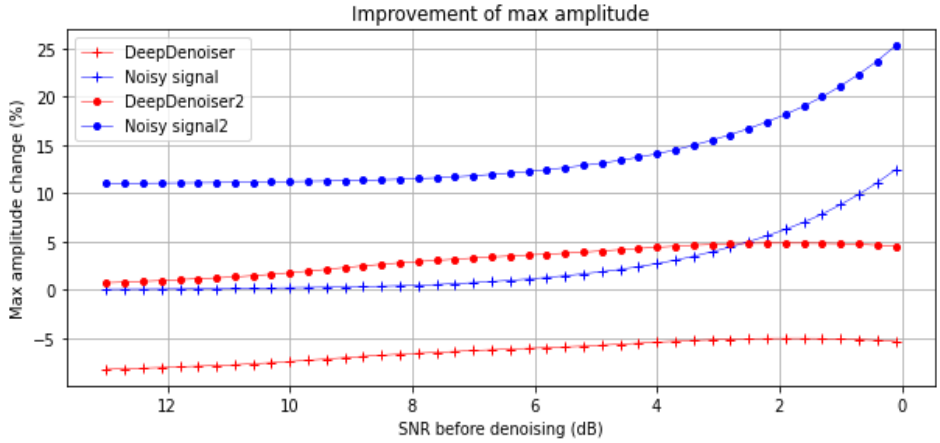
3.2.2. Maximum Amplitude

Figure 3.4 shows that in the first case the amplitude will always be reduced by about 5%, which becomes more severe as the SNR increases. There is no perfect convergence even with high SNR as with the original data. In the second case, where the reference is changed, there is a very good approximation in the high SNR range. The curve is very similar except that the amplitude is first amplified by 5% and then slowly

approaches zero. For SNR levels up to approximately 12 dB, the amplitude reduction accelerates. DeepDenoiser has problems separating noise from signal and tends to filter out the real signal, which is indicated by the reduction in amplitude.



(a) original

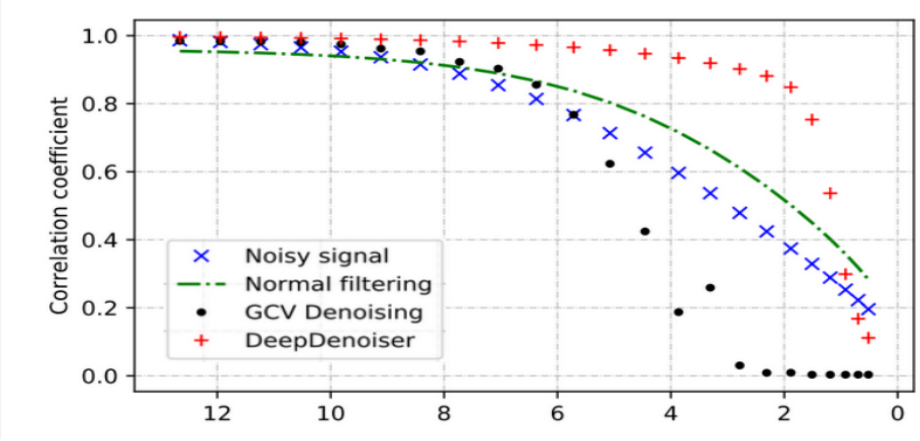


(b) new

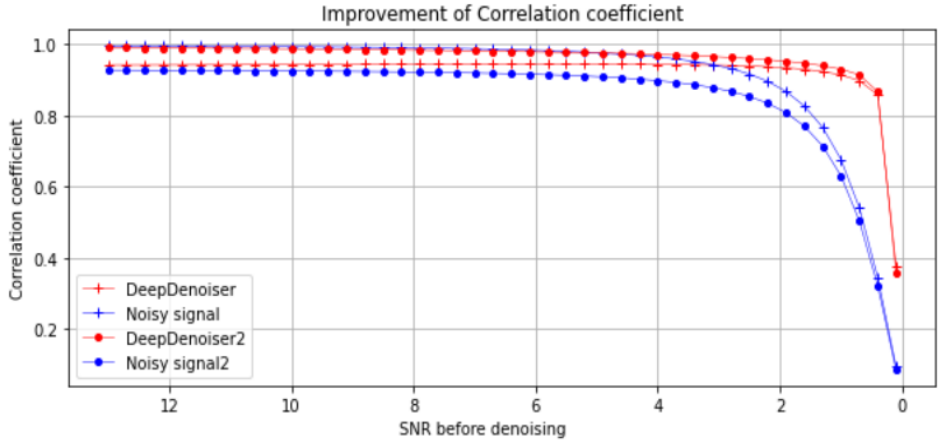
Figure 3.4.: Comparisons between the original results (a) of Zhu et al., 2019 and the newly replicated results (b) are shown in terms of changes in maximum amplitude after applying DeepDenoiser. For each SNR between 0.1 and 13 dB, 110 synthetic earthquakes with exactly that SNR are created in 0.3 dB steps. In case one, the average normalised maximum amplitude change in respect to the raw signal of all 110 synthetic earthquakes before (noisy data, blue pluses) and after (DeepDenoiser, red pluses) denoising with DeepDenoiser represents each data point. In case two, the average normalised maximum amplitude change in respect to the denoised raw signal of all 110 synthetic earthquakes before (noisy data, blue dots) and after (DeepDenoiser, red dots) denoising with DeepDenoiser represents each data point. GCV denoising (black dots) and normal filtering (green dash-dotted line) from the original graph are not relevant in this comparison and can be disregarded.

3.2.3. Cross-Correlation

The unit of the X-axis in Figure 3.5 is not in dB, as the original graph is erroneously not in dB either. Similar to the original, in the low ranges, there is a significant enhancement in cc. Further improvement ceases above an SNR of three. As the noise signal converges to one, the DeepDenoiser signal remains constant, at around 0.93. However, this changes in the second case. There is now a very fast convergence to one, which means that DeepDenoiser produces reproducible results even when the signal is significantly more noisy.



(a) original



(b) new

Figure 3.5.: Comparisons between the original results (a) of Zhu et al., 2019 and the newly replicated results (b) are shown in terms of changes in cc after applying DeepDenoiser. For each SNR between 0.1 and 13, 110 synthetic earthquakes with exactly that SNR are created in 0.3 steps. In case one, the average cc in respect to the raw signal of all 110 synthetic earthquakes before (noisy data, blue pluses) and after (DeepDenoiser, red pluses) denoising with DeepDenoiser represents each data point. In case two, the average cc in respect to the denoised raw signal of all 110 synthetic earthquakes before (noisy data, blue dots) and after (DeepDenoiser, red dots) denoising with DeepDenoiser represents each data point. GCV denoising (black dots) and normal filtering (green dash-dotted line) from the original graph are not relevant in this comparison and can be disregarded.

According to my findings, the following assumptions can be made about the behaviour of DeepDenoiser with regard to the data from Chile:

- SNR will be improved in all cases.
- DeepDenoiser reduces the amplitude.
- Even with a high SNR, DeepDenoiser cannot perfectly separate signal from noise.
- CC of signals with low SNR is significantly improved.

The same procedure was applied to higher magnitude earthquakes in the range of $M_w = 4$ to $M_w = 5$, as shown in Figure A.1, generating similar results. Although slight variations exist, the key points are the same. As a result, it is unlikely that the magnitude of these events has a substantial impact on the behaviour of the DeepDenoiser.

3.3. Limitations

Due to the considerably smaller amount of data and the lower variation, which can only be partially compensated for by combining the data, it is probable that the outcomes would alter, if other random events were chosen. Nevertheless, comparable observations should still be made. One of the most significant impacts on the results is the use of complementary filters, such as a high-pass filter of 1Hz (Figure A.2), which can produce significantly different outcomes in some aspects. There is no single ideal filter, but various appropriate filters exist, depending on the evaluation criterion. I opted for a 0.2 Hz high-pass filter as it exhibits the strongest improvement of cc in the low SNR range and generally the highest cc values in the second case. However, the maximum cc value in the first case is lower compared to when I use a 1 Hz high-pass filter. The high-pass filter of 1 Hz only marginally outperforms in terms of SNR improvement, starting at approximately 7 dB. Regarding the maximal amplitude, there is a decrease in amplitude reduction as the SNR increases, but generally, there is no superior approximation.

4. Repeating Earthquakes Sequences

In a second step, DeepDenoiser will be applied to events classified as repeating earthquakes. The objective is to investigate the impact of denoising on the obtained cross-correlation values, which serve as the standard for identifying RES. Specifically, the focus will be on understanding how the cc values for these events respond when computed after denoising. Are there clear patterns? Can detection be improved, or is there a more complex systematic?

The signal quality deteriorates as the distance increases. It is reasonable to assume that cc improvement may be feasible, particularly at longer distances, since the synthetic data indicates that enhancement took place at low SNR levels.

4.1. Data and Processing

The analysis examines 56 RES that have between 10 and 30 recurrences, out of the 2254 RES that were identified. These recurrences were recorded by a maximum of 23 monitoring stations and I have access to the records in the form of four-minute long waveforms. The processing objective is to generate seismograms for each RES at each station, which comprise of 35-second traces including two seconds of pure noise at the start, that can be combined into a single cc value. The two seconds pure noise is to get better results with DeepDenoiser.

To identify and accurately slice the traces, I employ a combination of the autopicker EQTransformer and cross-correlation. This approach ensures identification of RE while minimizing the risk of misidentifying other events recorded during the same time period.

For each RES, I choose the event with the most recorded data to be the master stream. Subsequently, I use the EQTransformer on the master stream to pick all recorded events.

To perform cc calculations, I employ a randomly selected second stream recorded at the same station as the reference.

Next, both streams are filtered using a band-pass filter of 1 - 10 Hz. This is according to Wolke, 2022 the optimal filter for this specific data. Only the HHZ traces every streams are used for the analysis.

For every picked P Phase I slice the master event in a 35 seconds long trace using the peak time minus two seconds. With the ’obspy.signal.cross_correlation.correlate_template’ function and the ’obspy.signal.cross_correlation.xcorr_max’ (Team, 2022) function the max cc and the shift between the sliced master trace and the refrencetrace is calculated.

The peak time of the pick with the highest max cc minus two seconds is used as the timestamp for the slice of the master trace in the following steps. If there is no pick with a max cc higher than 0.4 or no reference stream exists, the station is disregarded from the analysis. That is because to get the seismogram the sliced master trace is required.

First, every trace recorded at the station will be detrended and filtered using a band-pass filter of 1 - 10 Hz. For the DeepDenoiser seismograms the traces are also filtered using DeepDenoiser. To correctly slice the traces the master event is used. With the 'obspy.signal.cross_correlation.correlate_template' function and the 'obspy.signal.cross_correlation.xcorr_max' function the position of the repeating earthquake is determined. Every trace is sliced in 35 seconds long traces.

Then for every combination of traces, the cc is calculated using the 'obspy.signal.cross_correlation.correlate' (Team, 2022) function. The median of all cc values serves as the cc value for this station.

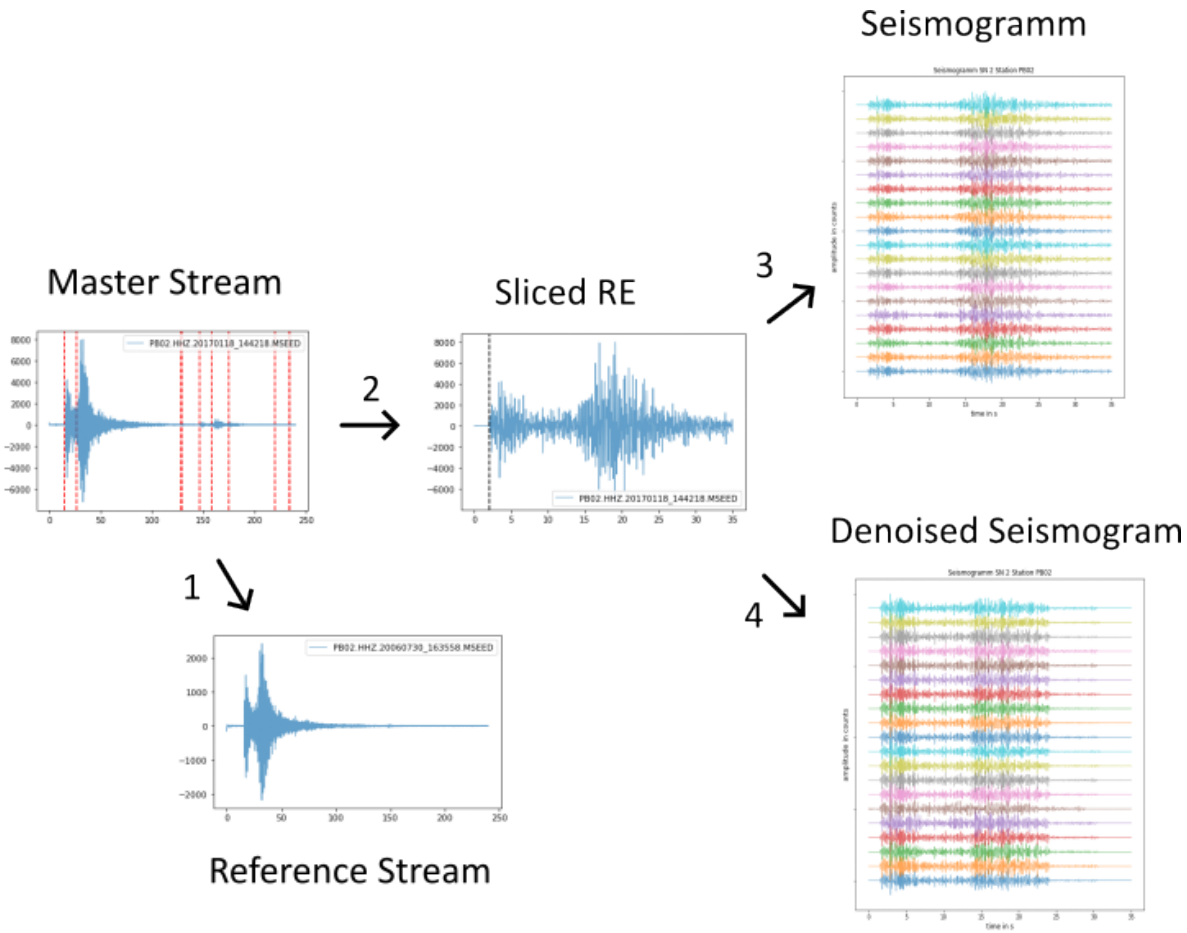


Figure 4.1.: Data flow diagram for creating seismograms. 1: Each pick is compared to the reference trace and the max cc is calculated. 2: The pick with the highest max cc is determined as the master trace. 3: Each trace is sliced to closely correspond with the master trace. 4: Each denoised trace is sliced to closely correspond with the master trace.

For each RES, the median of all latitudes and longitudes is determined as the common origin. The median is used because the original localisation of some events was incorrect. Using the 'geopy.distance.distance' ("GeoPy: Python Geocoding Toolbox", 2023) function, the flat distance (fd) between the hypocenter of the events and the stations is calculated. To take the depth (dep) of the hypocenter into account the following equation is used to calculate the Euclidean distance (dis), which gives a good approximation:

$$dis = \sqrt{fd^2 + dep^2}. \tag{4.1}$$

This process provides a dataset of cross-correlation coefficients in relation to the distance from the hypocenter.

4.2. Results

Through the process described above, 706 DeepDenoiser and 706 raw data points were generated. Each data point represents all recorded events at a station from that RES. Figure 4.2 and 4.3 shows the distribution of the cc values in relation to the Distance. Most events were recorded at a distance of 100 to 200 km. There is a slight exponential decrease of the cc with increasing distance, which is expected due to geometric spreading. In addition, the range of cross-correlation coefficients also increases with increasing distance and more Data points have very low cc. Considering the included high distances, the general data quality is good.

The DeepDenoiser data points and the Raw data points in Figure 4.2 are both fitted with the following fit function:

$$f(x) = a - b^{cx}. \tag{4.2}$$

I have tested various fitting functions and found this particular function to be the most fitting.

For events within a 90 km distance there is a general improvement with the Deep-Denoiser filter. However, for events further away there is a decrease in cc. This contradicts the hypothesis derived from the results in the previous chapter.

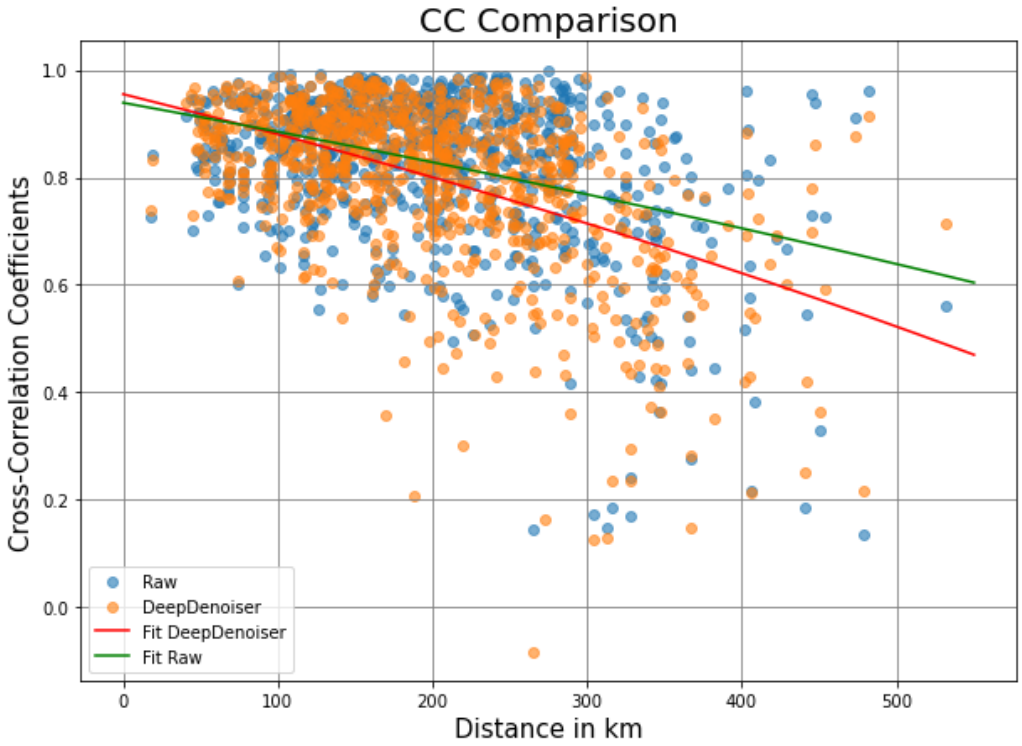


Figure 4.2.: CC comparisons between DeepDenoiser and raw, distance dependent. The cc of each RES at each station before and after denoising with DeepDenoiser is displayed. The fits are obtained by fitting the exponential decay Function 4.2 to the DeepDenoiser and raw data.

In Figure 4.3 the first violin plot shows a slight improvement with the DeepDenoiser filter, both in density estimate and higher mean. For every other violin plot, there is a decline to be noted. The DeepDenoiser data points never have the highest recorded cc in any step.

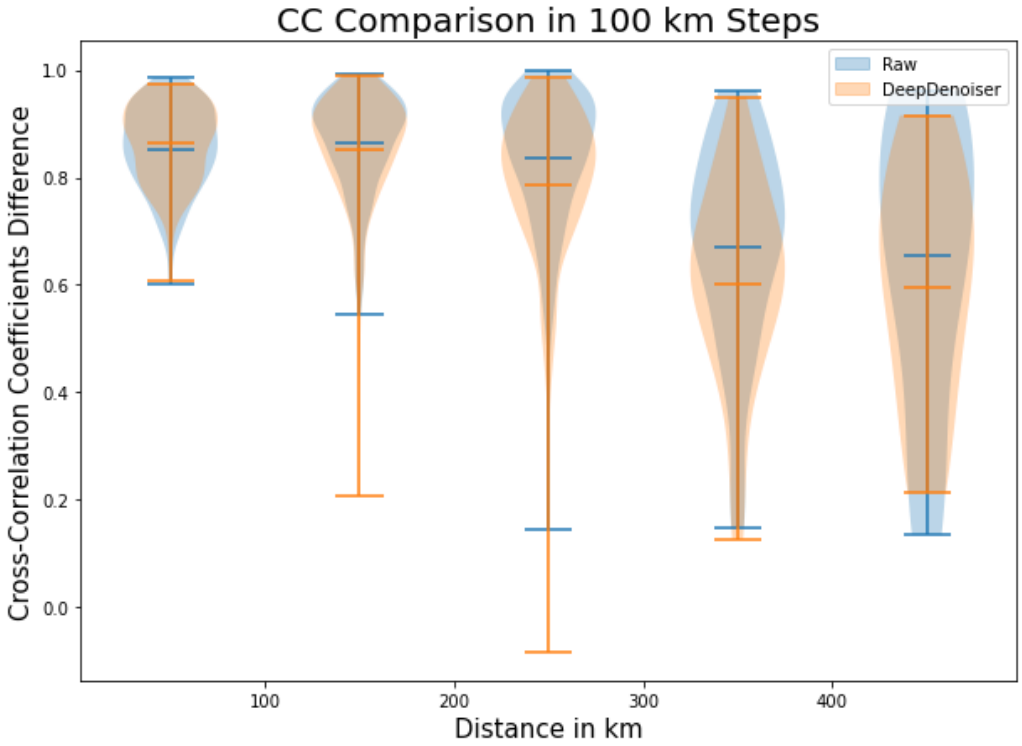


Figure 4.3.: CC comparisons between DeepDenoiser and raw, with the density estimate for each 100 km step, including extrema and means. The cc of each RES at each station before and after denoising with DeepDenoiser is calculated and displayed as violin plots including only events at that distance.

The difference between DeepDenoiser data points and raw data points are depicted in Figure 4.4 and 4.5. Most values are concentrated within the range of -0.1 to 0.1, with a linear decreasing tendency. Again, it can be seen that there is more improvement in the very short range, which then decreases with increasing distance. In the further distances above 300 km, there are more cases of strong improvement, although the mean value remains in the negative range. So in general, the hypothesis that DeepDenoiser will improve the CC at great distances cannot be proven. However, for certain specific cases, it may still hold true.

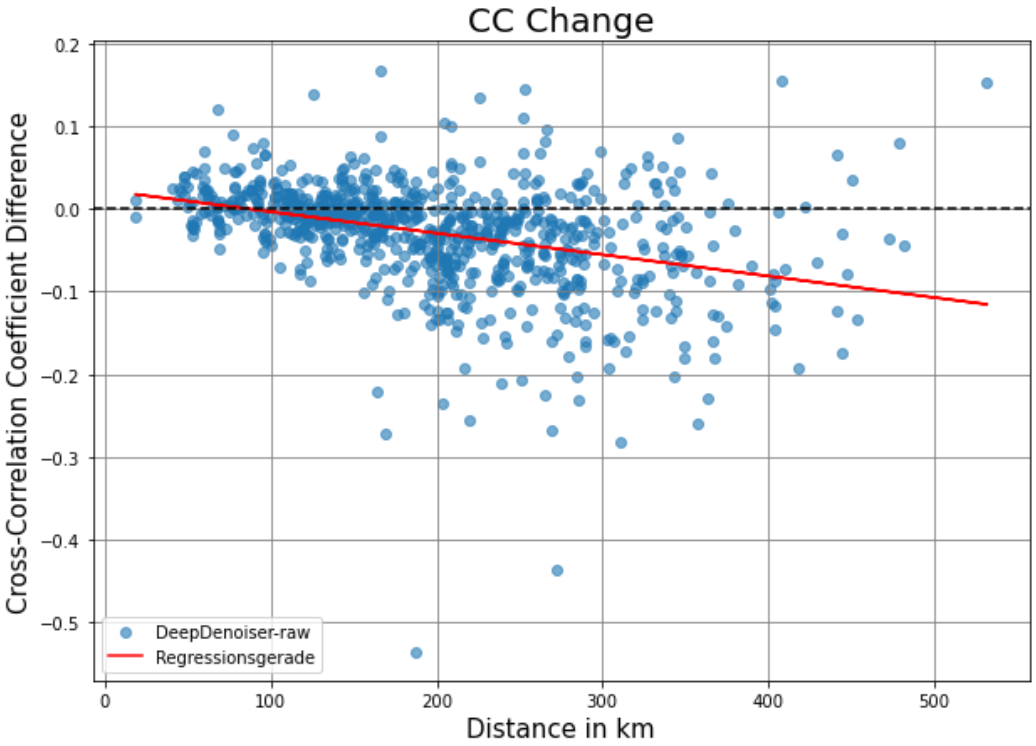


Figure 4.4.: DeepDenoiser - raw cc difference, distance dependent. The cc of each RES at each station before and after denoising with DeepDenoiser is calculated. The resulting difference between these values is displayed. A linear regression line is fitted to the data. Positive values indicate improvements with DeepDenoiser whereas negative values indicate worsening by DeepDenoiser.

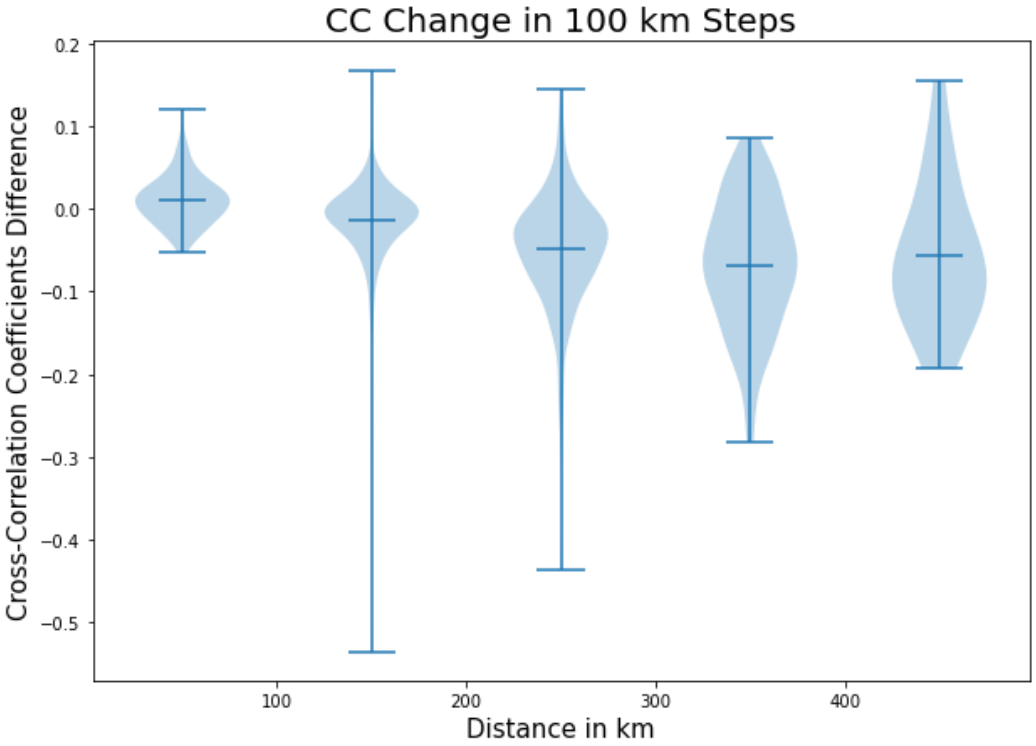


Figure 4.5.: DeepDenoiser - raw cc difference, with the density estimate for each 100 km step, including extrema and means. The cc of each RES at each station before and after denoising with DeepDenoiser is calculated. The resulting difference between these values is displayed as violin plots including only events at that distance. Positive values indicate improvements with DeepDenoiser whereas negative values indicate worsening by DeepDenoiser.

To assess whether DeepDenoiser contributes to detection, I examine the change of cc concerning each individual RES. To do so, I aggregate all data points from Figure 4.4 that correspond to a single RES (as depicted in Figure 4.6). I calculate the average cc for the RES, including only the top three cc values, for both the raw and DeepDenoiser data. The difference in cc is then plotted in Figure 4.7. It is a common practice in RE identification to include only the best stations when evaluating whether an event qualifies as an RE. The individual RES show significantly different results, spanning from -0.1 to 0.1. This suggests that the RES position has a certain influence on the outcome. The RES can be roughly separated into two different groups. In the western cluster, off the coast of Chile, there are mainly neutral to negative RES, while the rest of the RES are mainly positive.

The absence of nearby stations in the western cluster may be a contributing factor to this phenomenon. Previous findings demonstrate a decrease in cc as distance increases, though the influence of other unknown factors cannot be excluded. There appears to be no clear correlation between the depth of hypocenters and changes in cc.

The travel paths of earthquakes become increasingly complex as they occur at greater depths, resulting in more intricate waveforms. In some instances, phase conversion may occur due to the boundary between the crust and mantle in this region being approximately 50 km deep (Sodoudi et al., 2011). However, the extent to which this influences DeepDenoiser is difficult to assess due to the limited number of very deep RES available. At least in the case of the two earthquakes that occurred at a depth of approximately 140 km, DeepDenoiser improved the results.

In the case where only the top three stations are considered (Figure 4.7), the results differentiate. There still remains a big difference between the two groups. The western cluster shows some positive RES, but mainly neutral to negative RES. The remaining stations now demonstrate a more mixed result. In total 23 RES improved after the application of DeepDenoiser. That is a success rate of approximately 40%.

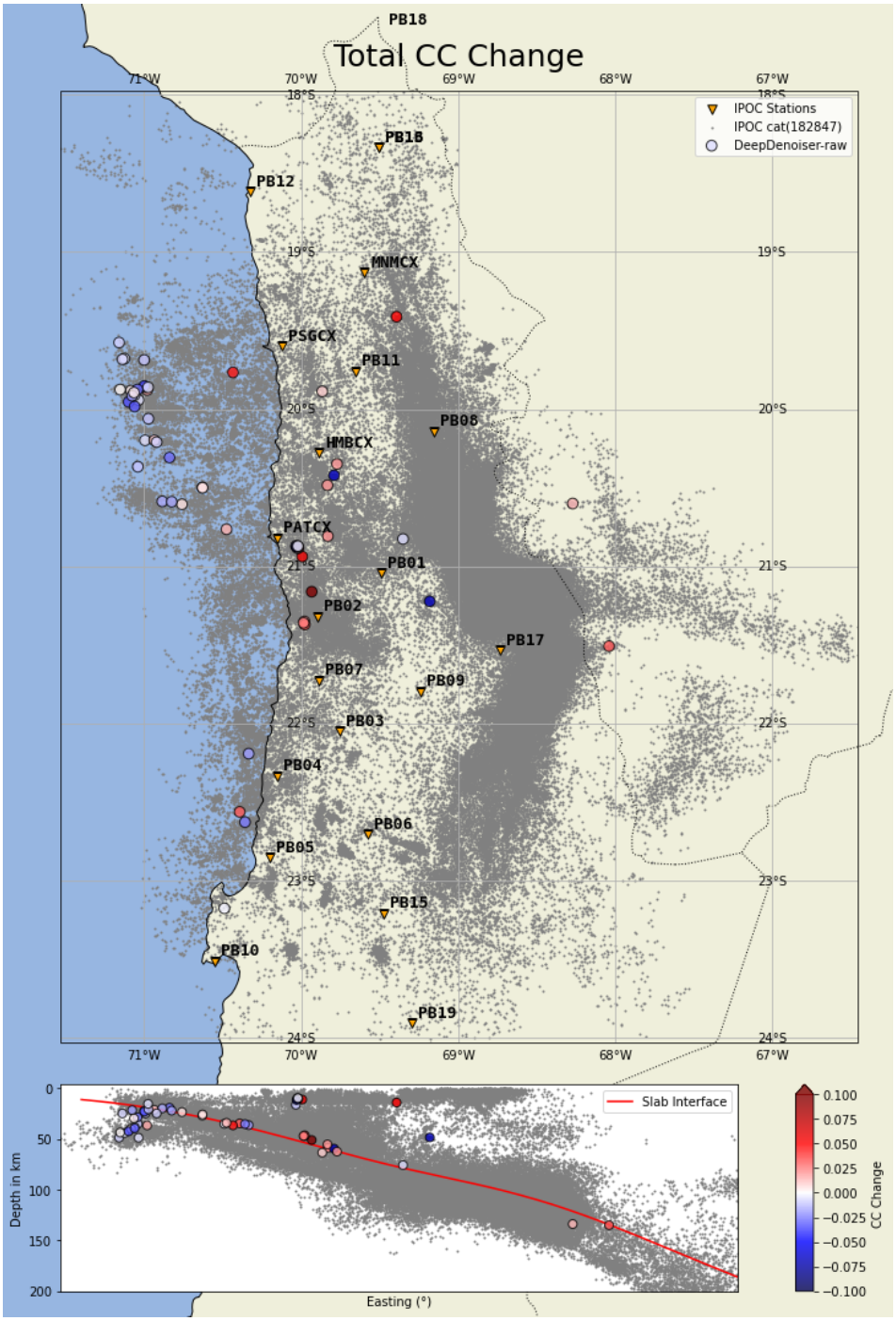


Figure 4.6.: Map of the surveyed region in northern Chile, with the 56 RES colour coded according to their total cc change. Positive values (highlighted in red) indicate improvement due to DeepDenoiser, while negative values (highlighted in blue) signify worsening. Events from the catalogue of Sippl et al., 2023 are shown in grey, and the 23 IPOC stations are shown in orange. Figure altered after Dr. Folesky (personal communication, 2023).

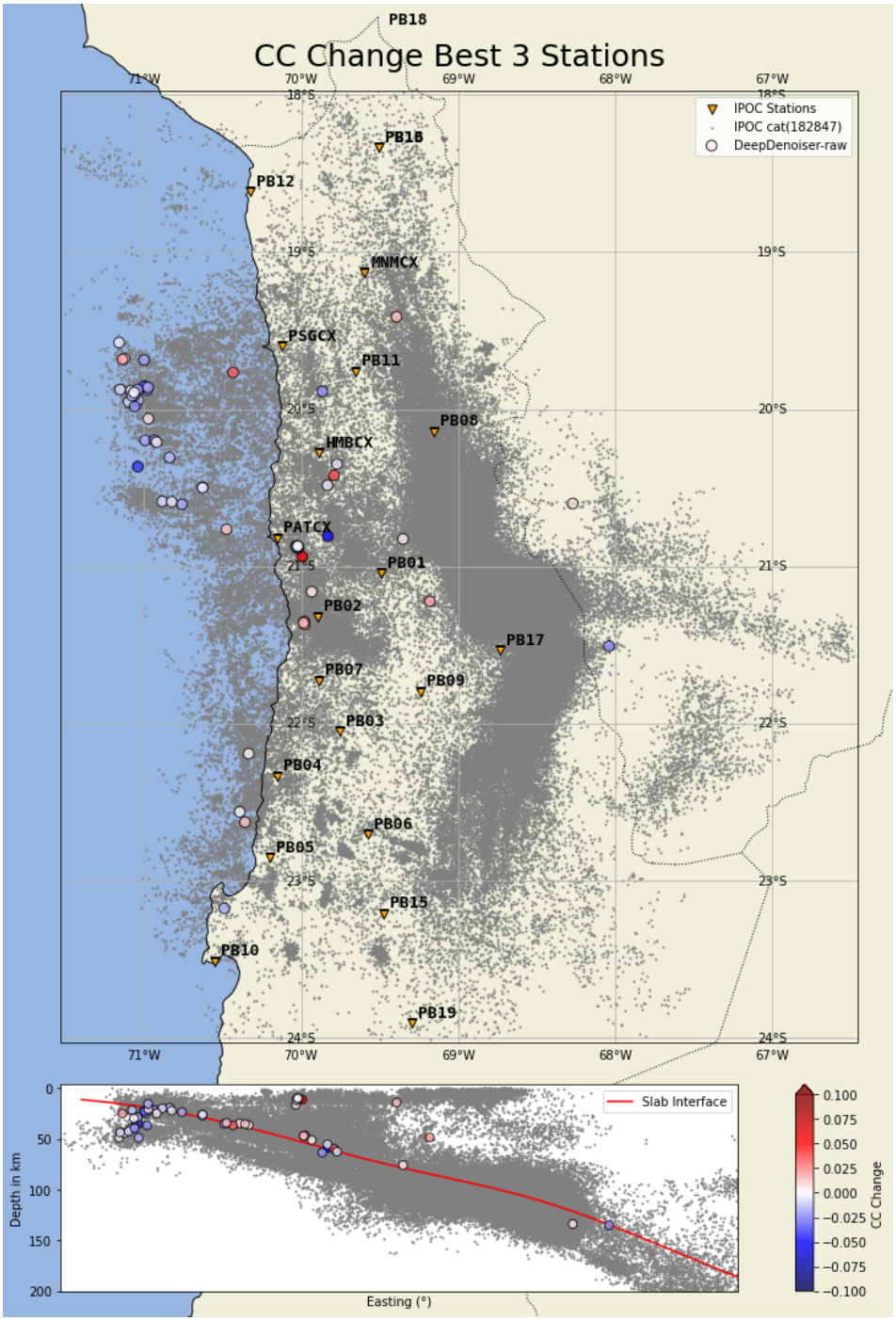


Figure 4.7.: Map of the surveyed region in northern Chile with the 56 RES colour coded according to their cc change for only the best three stations. Positive values (highlighted in red) indicate improvement due to DeepDenoiser, while negative values (highlighted in blue) signify worsening. The Events from the catalogue of Sippl et al., 2023 are shown in grey, and the 23 IPOC stations are shown in orange. Figure altered after Dr. Folesky (personal communication, 2023).

4.3. Limitations

Even though I created a robust system for selecting master events and creating the seismograms, with multiple quality controls, it is possible that genuine repeating earthquakes may not be identified due to the necessary automatic nature of the process. In cases where heavy noise is present, either over the master trace or reference trace, a random event may receive a better cc. In cases where the master events can be correctly identified, but the rest of the traces recorded at the station is of poor quality, the results become distorted, resulting in outliers in the data set.

Similarly to synthetic data, the additional use of a filter (band-pass filters of 1-10 Hz) has an impact on the results and can change the outcome depending on whether it was used before or after DeepDenoiser and what range it filters. The creation of a seismogram and its inclusion in the dataset can be affected by various criteria that need to be considered carefully.

The impact of different earthquake causes on DeepDenoiser's behaviour is difficult to assess, as the specific trigger process is unknown for each RES.

It is important to note that I use the best cc values rather than the nearest stations for calculating cc change in Figure 4.7. While there may be some differences, I believe the impact is not significant.

5. Discussion and Conclusion

I discovered that machine learning-based denoising can be helpful for waveform similarity analysis. However, DeepDenoiser in its current form is not universally suitable for this task. While it achieves a significant enhancement in cc for low signal-to-noise ratios in synthetic data. This behaviour cannot be fully identified within the repeater data.

Nonetheless, I observe a general improvement in RES recorded by stations up to approximately 90 km away. Repeaters located outside the main monitored region of IPOC of 70°W - 69°W, often do not show improvement with DeepDenoiser. The cross-correlation values show alterations of around ±0.1. In practice, the waveform similarity analysis typically considers only the top-performing stations. Therefore, there may still be value in applying DeepDenoiser to improve the identification of RE. A significant 40% of all RES were improved through DeepDenoiser.

In my analysis, I examined the impact of distance and SNR on cross-correlation, and discovered correlations. However, I also observed a more complex system that influences DeepDenoiser's results.

During the analysis, it was observed that DeepDenoiser tends to separate P and S waves for events with longer travel times (see Figure A.3), treating them as separate events. This behaviour may be attributed to the fact that DeepDenoiser was trained using traces with a maximum length of 30 seconds. While waveform similarity analysis was not the primary application envisioned for DeepDenoiser, this study highlights the prospect of its adoption as a promising use case within this specialized domain. Because DeepDenoiser was trained on earthquakes recorded by the NCSN, primarily consisting of shallow (0-15 km) crustal earthquakes. The difference in the tectonic setting plays a big role in the difference between the original results of Zhu et al., 2019 and mine.

DeepDenoiser behaviour is likely influenced by the following factors:

- Distance to measuring station,
- Frequency spectrum,
- Hypocenter location,
- Signal-to-noise ratio,
- Travel time duration.

Earthquake magnitude is an unlikely factor of influence. The effects of depth and various causes of earthquakes remain unknown. Other potential factors cannot be completely ruled out.

While DeepDenoiser has the potential to improve waveform similarity analysis, the varied results obtained prohibit a universal recommendation for using DeepDenoiser

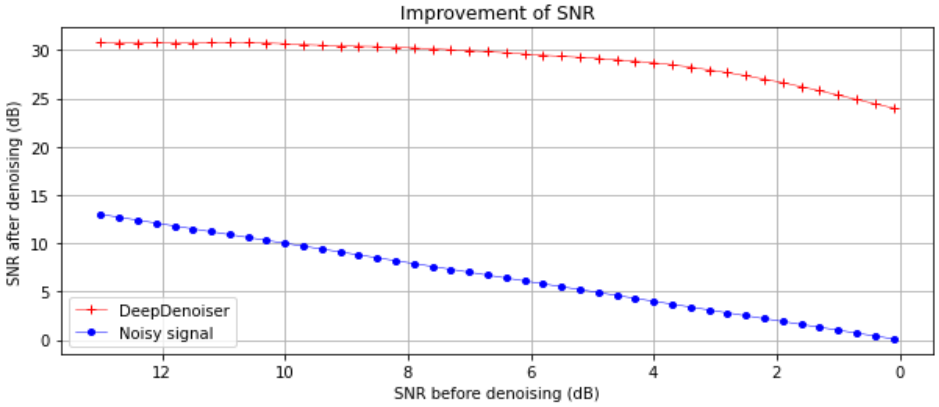
to enhance RE identification. When using machine learning-based methods, it is important to note that DNNs can learn but not think. The training process and the dataset must be taken into account when evaluating the application of these models.

6. Outlook

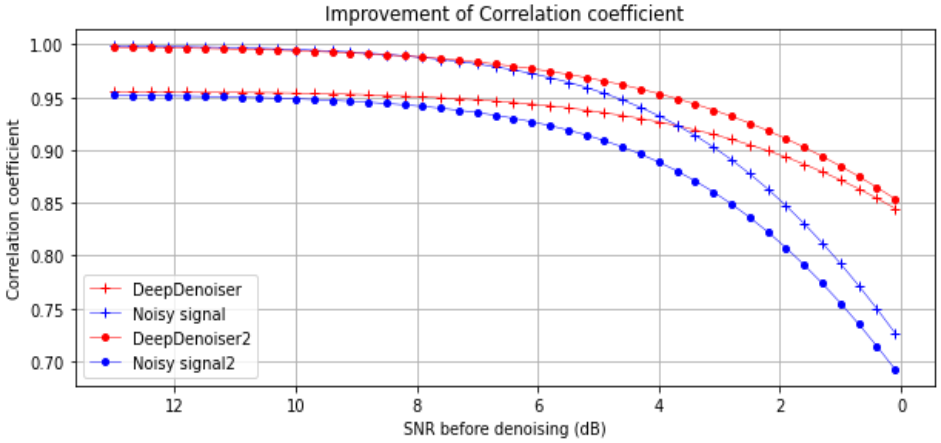
To develop the potential of machine learning based denoising for waveform similarity analysis, further research is necessary. As the interaction between deep neural networks and earthquake waveforms is very complex and hard to predict, exploring various mitigating factors is important to better understand their behaviour.

In the case of the Chilean data, a retraining of DeepDenoiser could potentially aid in the improved identification of repeating earthquakes and assist with the creation of a more extensive catalogue. Alternatively, the development of a specialised method for waveform similarity analysis might be necessary. To further test DeepDenoiser capabilities for waveforms similarity analysis. A potential test could be reclassifying the RES catalogue and comparing the results with the current catalogue. How many new repeaters are discovered? How many false detections? Another more general questions that should be considered in future research is, if it is feasible to develop a globally applicable method, or if these methods should always be considered on a regional basis? Machine learning is advancing rapidly, and I am convinced that it will become an important tool in many fields in the near future. Therefore, it is worth pursuing this topic further.

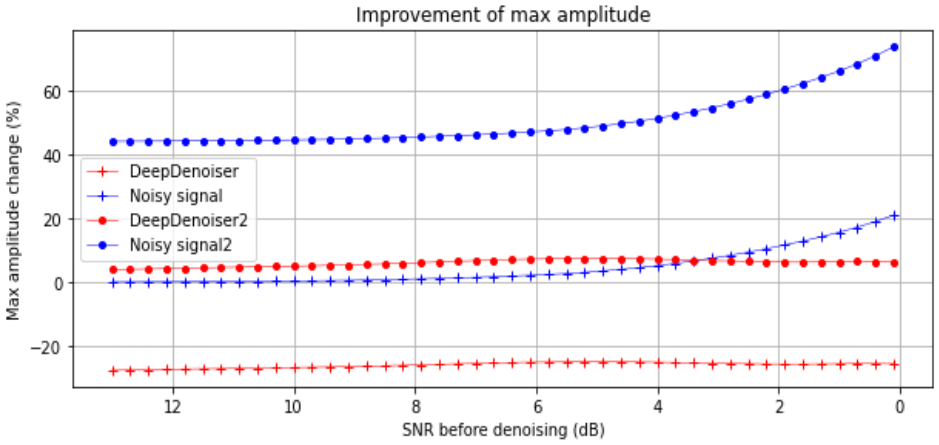
A. Appendix



(a) SNR

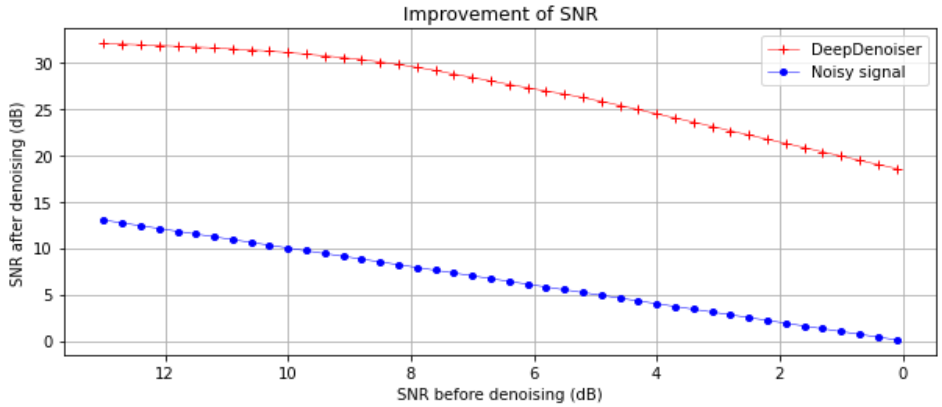


(b) CC

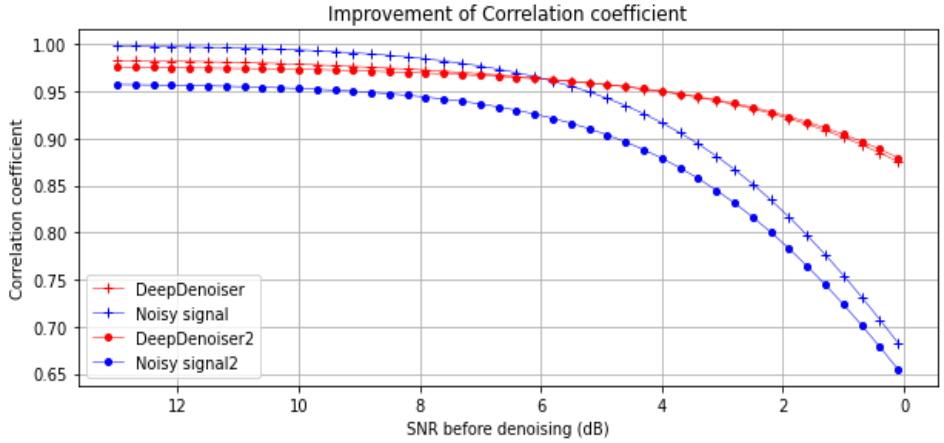


(c) MA

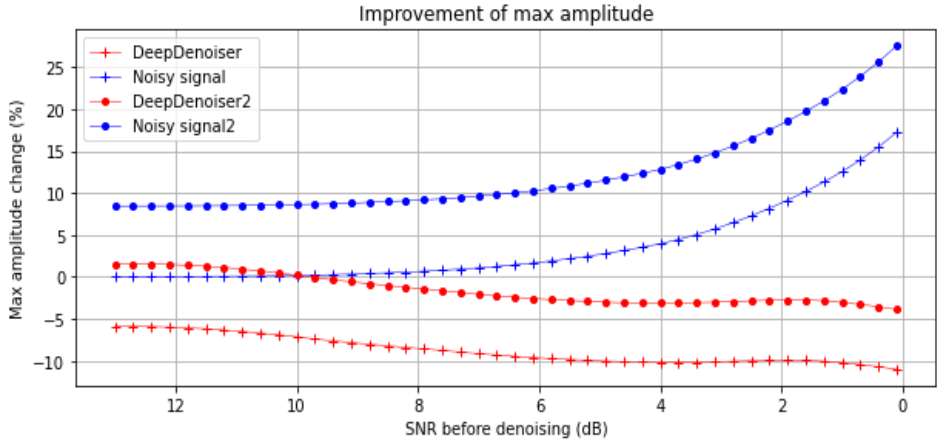
Figure A.1.: Alternative synthetic plots featuring earthquakes of greater magnitude ($M_w > 4$ - $M_w < 5$) and a high-pass filter of 0.2 Hz.



(a) SNR

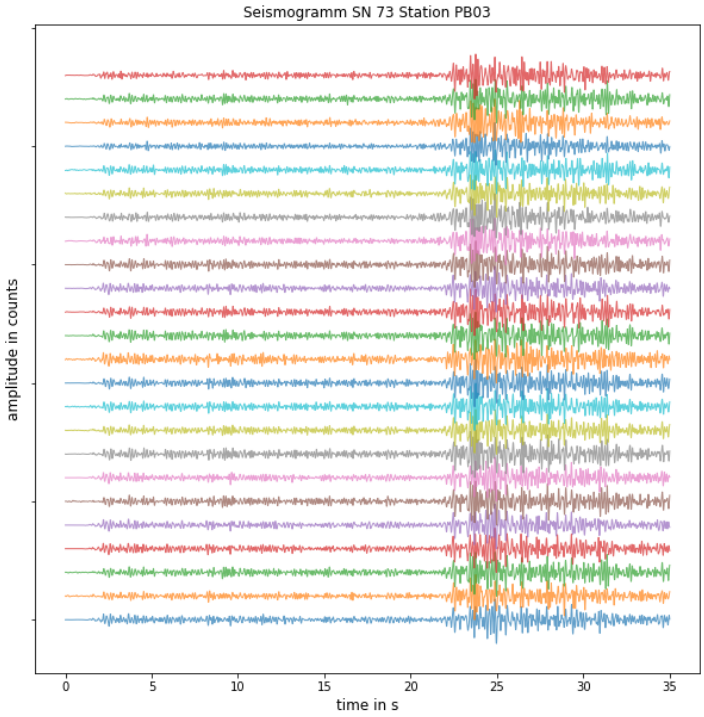


(b) cc

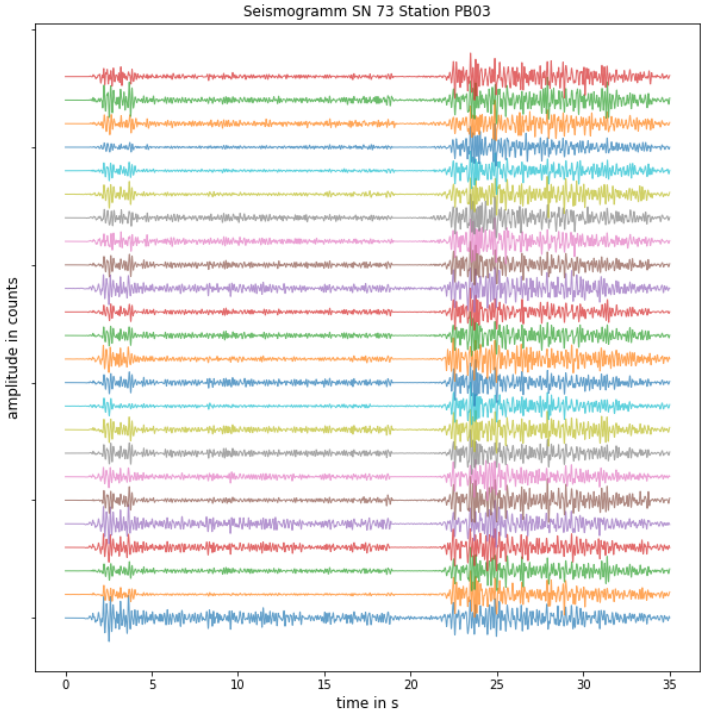


(c) maximum amplitude

Figure A.2.: Alternative synthetic plots with a high-pass filter of 1 Hz.



(a) raw



(b) DeepDenoiser

Figure A.3.: Example of DeepDenoiser separating P and S waves. Z-component seismograms of RES 73 recorded at station PB03.

B. Appendix

See https://github.com/forgso/DeepDenoiser-for-waveform-similarity-analysis for the full Python code used in this thesis.

```
import seisbench.models as sbm
import matplotlib.pyplot as plt
from obspy import read, UTCDateTime, Stream
import numpy as np
from obspy.signal.cross_correlation import correlate

model = sbm.DeepDenoiser.from_pretrained("original")

def deepDenoiser(dt,dt2=30, snr=2, dtn=dtn1):
    """
  Applies denoising to a seismic signal within a specified time window.

  Args:
      dt (obspy.UTCDateTime): Start time for the signal extraction.
      dt2 (int, optional): Duration of the signal extraction in seconds. Default is 30 seconds.
      snr (float, optional): Signal-to-noise ratio (SNR) for noise addition. Default is 2.
      dtn (obspy.UTCDateTime): Start time for the noise signal extraction.

  Returns:
      annotations (obspy.Stream): Stream with annotations after denoising.
      ste (obspy.Stream): Original seismic signal within the specified time window.
      noisy_data (obspy.Stream): Noisy signal with added noise.
  """

    #slice signal
    ste = st.slice(dt, dt + dt2)
    #slice noise
    stn = st.slice(dtn, dtn + dt2)

    #add noise
    noisy_data = add_noise_to_stream(ste, stn, snr)
    #signal denoisen whith DeepDenoiser
    annotations = model.annotate(noisy_data)

    return annotations, ste, noisy_data

#%%
def add_noise_to_stream(stream, noise, snr):
    """
   Adds noise to a seismic stream with the desired SNR.

    Args:
```

```
43          stream (obspy.Stream): Original seismic stream.
44          noise (obspy.Stream): Stream containing the noise signal.
45          snr (float): Desired signal-to-noise ratio (SNR).

47      Returns:
48          noisy_stream (obspy.Stream): Stream with added noise based on
        the specified SNR.
49      """

51      # Check if the number of traces in the stream object and in the
        noise object are the same
52      if len(stream) != len(noise):
53          raise ValueError("The number of traces in the stream and in
        the noise must match.")

55      # Create an empty list for the modified traces
56      modified_traces = []

58      # Add the noise to each trace in the stream object
59      for trace, noise_trace in zip(stream, noise):
60          # Check if the sampling rate of the traces matches
61          if trace.stats.sampling_rate != noise_trace.stats.
        sampling_rate:
62              raise ValueError("The sampling rate of the traces must
        match.")

64          # Calculate the scaling factors based on the SNR and the
        standard deviations        std_signal = np.std(trace.data[500:])
65          std_noise = np.std(noise_trace.data)
66          print('std_noise',std_noise,'std_noise 500', np.std(
        noise_trace.data[:500]))
67          # dB or not
68          #std_target = std_signal / snr
69          std_target = std_signal / (10**(snr / 10))
70          adjustment_factor = std_target / std_noise

72          # Scale the noise based on the SNR
73          scaled_noise = noise_trace.data * adjustment_factor

75          calculate_snr(trace.data[500:], scaled_noise.data)

77          # Add the noise to the trace
78          noisy_data = trace.data + scaled_noise
79          noisy_trace = trace.copy()
80          noisy_trace.data = noisy_data

82          # Add the modified trace to the list
83          modified_traces.append(noisy_trace)

85      # Create a new stream object with the modified traces
86      noisy_stream = Stream(traces=modified_traces)

88      return noisy_stream
89  #%%
90  def calculate_snr(signal, noise):
```

```
91      """
92      Calculates the signal-to-noise ratio (SNR) between a signal and
        noise.
94      Args:
95          signal (numpy.ndarray): Signal data array.
96          noise (numpy.ndarray): Noise data array.
98      Returns:
99          snr (float): Signal-to-noise ratio (SNR) in dB.
100     """
102     # Standard deviations of the signal and noise
103     std_signal = np.std(signal, axis=0)
104     std_noise = np.std(noise, axis=0)
106     #  SNR in decibels (dB)
107     snr = 10 * np.log10(std_signal / std_noise)
108     #snr = std_signal / std_noise
110     print(snr)
112     return snr
114 #%%
116 def Cross_Corelation(deno_da,evt_da):
118     """
119     Computes the cross-correlation between two seismic data arrays.
121     Args:
122         deno_da (numpy.ndarray): Denoised seismic data array.
123         evt_da (numpy.ndarray): Original seismic data array.
125     Returns:
126         corr (float): Maximum cross-correlation value at zero lag.
127     """
129     #cross Corelation zero lag machen
130     corr = max(correlate(deno_da[500:], evt_da[500:], 0, normalize= '
        naive')) #zero Lag
131     print(corr)
133     return corr
```

Source Code B.1: Excerpt of the Python code used to generate the synthetic data in the 3 chapter.

References

Adaloglou, N. (2020). Intuitive explanation of skip connections in deep learning. *https://theaisummer.com/*. https://theaisummer.com/skip-connections/

Aghdam, H. H., & Heravi, E. J. (2017). Guide to convolutional neural networks : A practical application to traffic-sign detection and classification / hamed habibi aghdam, elnaz jahani heravi.

Anand, R. (2022). *Digital signal processing : An introduction / r. anand.* Mercury Learning; Information,

Bormann, P., & Wielandt, E. (2013). Seismic signals and noise. https://api.semanticscholar.org/CorpusID:51833666

Dumoulin, V., & Visin, F. (2018). A guide to convolution arithmetic for deep learning.

Folesky, J., Kummerow, J., & Shapiro, S. A. (2021). Stress drop variations in the region of the 2014 m w 8.1 iquique earthquake, northern chile. *Journal of geophysical research. Solid earth*, *126*(4).

Geopy: Python geocoding toolbox. (2023).

GFZ German Research Centre for Geosciences & Institut des Sciences de l'Univers-Centre National de la Recherche CNRS-INSU. (2006). Ipoc seismic network. integrated plate boundary observatory chile - ipoc. *Other/Seismic Network*. https://doi.org/10.14470/PK615318

Hayes, G. P., Herman, M. W., Barnhart, W. D., Furlong, K. P., Riquelme, S., Benz, H. M., Bergman, E., Barrientos, S., Earle, P. S., & Samsonov, S. (2014). Continuing megathrust earthquake potential in chile after the 2014 iquique earthquake. *Nature (London)*, *512*(7514), 295–298.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv.org*.

IPOC. (2018). Integrated plate boundary observatory chile [Online; accessed September 24, 2023]. https://www.ipoc-network.org/welcome-to-ipoc/

Lengliné, O., & Marsan, D. (2009). Inferring the coseismic and postseismic stress changes caused by the 2004 mw = 6 parkfield earthquake from variations of recurrence times of microearthquakes. *Journal of Geophysical Research - Solid Earth*, *114*(B10), B10303–n/a. https://doi.org/10.1029/2008JB006118

Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y., & Beroza, G. C. (2020). Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature communications*, *11*(1), 3952–3952.

Nadeau, R. M., & McEvilly, T. V. (1999). Fault slip rates at depth from recurrence intervals of repeating microearthquakes. *Science (American Association for the Advancement of Science)*, *285*(5428), 718–721.

Norabuena, E., Leffler-Griffin, L., Mao, A., Dixon, T., Stein, S., Sacks, I. S., Ocola, L., & Ellis, M. (1998). Space geodetic observations of nazca-south america convergence across the central andes. *Science (American Association for the Advancement of Science)*, *279*(5349), 358–362.

Sippl, C., Schurr, B., Asch, G., & Kummerow, J. (2018). Seismicity structure of the northern chile forearc from >100,000 double□difference relocated hypocenters. *Journal of geophysical research. Solid earth*, *123*(5), 4063–4087.

Sippl, C., Schurr, B., Münchmeyer, J., Barrientos, S., & Oncken, O. (2023). Catalogue of earthquake hypocenters for northern chile from 2007-2021 using ipoc (plus auxiliary) seismic stations. *GFZ Data Services*. https://doi.org/10.5880/GFZ.4.1.2023.004

Sodoudi, F., Yuan, X., Asch, G., & Kind, R. (2011). High-resolution image of the geometry and thickness of the subducting nazca lithosphere beneath northern chile. *Journal of Geophysical Research: Solid Earth*, *116*(B4). https://doi.org/https://doi.org/10.1029/2010JB007829

Team, T. O. D. (2022). *Obspy 1.4.0* (Version 1.4.0). Zenodo. https://doi.org/10.5281/zenodo.6645832

Uchida, N. (2019). Detection of repeating earthquakes and their application in characterizing slow fault slip. *Progress in Earth and Planetary Science*, *6*(1), 1–21.

Uchida, N., & Bürgmann, R. (2019). Repeating earthquakes. *Annual review of earth and planetary sciences*, *47*(1), 305–332.

Wolke, M. (2022). Repeating Earthquakes in northern Chile - True repeaters or just similar earthquakes?

Zhu, W., Mousavi, S. M., & Beroza, G. C. (2019). Seismic signal denoising and decomposition using deep neural networks. *arXiv.org*.

Declaration of Authorship

I hereby certify that this thesis has been composed by me and is based on my own work, unless stated otherwise. No other person's work has been used without due acknowledgement in this thesis. All references and verbatim extracts have been quoted, and all sources of information, including graphs and data sets, have been specifically acknowledged.

Date: _____ Signature: _____