Suppression of wind turbine noise from seismological data using nonlinear thresholding and denoising autoencoder

Janis Heuel · Wolfgang Friederich

Received: 1 March 2022 / Accepted: 9 May 2022 / Published online: 2 July 2022 © The Author(s) 2022

Abstract Seismologists found a significant deterioration in station quality after installation of wind turbines (WTs), which led to conflicts between WT operators and seismic services. We compare different techniques to reduce the disturbing signals from WTs at seismological stations by selection of an affected station. WT noise and earthquake signals have overlapping frequency bands, and thus spectral filtering cannot be used. As a first method, we apply the continuous wavelet transform on our data to obtain a time-scale representation. From this representation, we estimate a noise threshold function either from noise before the theoretical P-arrival or using a noise signal from the past with similar ground velocity conditions at the surrounding WTs. As a second method, we use a denoising autoencoder (DAE) that learns mapping functions to distinguish between noise and signal. In our tests, the threshold function performs well when the event is visible in the raw or spectral filtered data, but it fails when WT noise dominates. The use of the threshold function and pre-noise can be applied immediately to real-time data and has low computational cost. Using a noise model from our prerecorded database at the seismological station does not improve the result and is more time consuming. In contrast, the DAE is able to remove WT noise even when the event is completely covered by noise. However, the DAE must be trained with typical noise samples and high signal-to-noise ratio events to distinguish between signal and interfering noise.

Keywords Neural networks · Seismic noise · Wavelet transform · Fourier analysis

1 Introduction

Over the last years, the installation of wind turbines (WTs) increased in Germany due to a substantial development of renewable energy. Owing to negative effects on humans, WTs are often installed in areas with low population density. Because of low anthropogenic noise, these areas are also well suited for sites of seismological stations. As a consequence,
WTs are often installed in the same areas as seismological stations. Several studies investigated the imprints of WT noise on seismological recordings (Styles et al. 2005; Saccorotti et al. 2011; Stammler and Ceranna 2016; Estrella et al. 2017; Neuffer and Kremers 2017; Zieger and Ritter 2018; Neuffer et al. 2019; Lerbs et al. 2020; Neuffer et al. 2021) and characterised the wavefield of the disturbing WT signals. All mentioned studies come to the conclusion that the data quality is deteriorating due to the expansion of WTs.

Stammler and Ceranna (2016) analysed time spans before and after installation of WTs at the Gräfenberg array (GRF) in Germany and show a strong dependence of local wind speed and noise spectra after the installation of new WTs. They conclude that the WT signals reduce the sensitivity of the GRF stations in a frequency range between 1 and 7 Hz, which is crucial for the analysis of seismic events. Furthermore, they noticed that installations of WTs within a radius of 5 km around a seismological station lead to a worsening in station quality. Therefore, Stammler and Ceranna (2016) suggest to prohibit additional installations within a distance of 5 km. Neuffer and Kremers (2017), Estrella et al. (2017), Zieger and Ritter (2018), and Neuffer et al. (2019) further substantiate this suggestion by estimating power spectral density (PSD) amplitude decay curves in the frequency band between 1 and 10 Hz. The suggestions by Stammler and Ceranna (2016) and the amplitude decay curves of other studies led to definitions of protection radii and radii of shared interest of the WT operators and the seismological data services, e.g. in the states of Bavaria (Windenergie-Erlass - BayWEE 2016) and North Rhine-Westphalia (Windenergie-Erlass NRW 2018). Thus, some stations are protected within a radius of 5 to 15 km, where the installation and operation of wind turbines is prohibited. The protection zone around a seismological station always depends on the importance of the station (e.g. 15 km for stations, which are used for the Comprehensive Nuclear-Test-Ban Treaty (Windenergie-Erlass - BayWEE 2016)), and the local geological and topographical settings (Estrella et al. 2017). At other sites, the WT constructors have to inform the seismological data services when planning new WTs close (several kilometres) to seismological stations and they may have to pay for a new station, when the station quality is degraded by the new WT.

Due to the large protection radii, a lot of potential area is lost that could be used for wind energy. In order to achieve decarbonisation of the energy sector for both private and industrial consumption, it is important that as much available land as possible can be used for renewable energies and especially wind energy.

Since the WT noise and the earthquake signals share a common frequency band, spectral filtering, based on the Fourier transform, which is normally used to suppress the noise, cannot be used. Furthermore, spectral filtering alters the waveform and can affect subsequent analyses. Mousavi et al. (2016) and Langston and Mousavi (2019) modified time scale coefficients after transforming time series data using the continuous wavelet transform (CWT) to effectively suppress noise. Even when signal and noise share a common frequency band, their method can separate signal and noise, but the signal is always visible in the noisy raw data in the examples shown, which is not always true for real data. Sometimes, the earthquake signal is completely covered by noise and therefore not visible in either the time or the time-frequency domain. Zhu et al. (2019), Mandelli et al. (2019), and Tibi et al. (2021) have developed a new approach that uses a convolutional denoising autoencoder (DAE) to suppress white noise, a variety of coloured noise, and non-earthquake signals at seismological stations. During the training process, the DAE learns to distinguish between signal and noise by simultaneously learning a variety of sparse representations from the input time-frequency coefficients and functions that map these representations into two individual mask functions for signal and noise, respectively. Zhu et al. (2019) and Tibi et al. (2021) used the short-time Fourier transform (STFT) to calculate the time-frequency coefficients. Mandelli et al. (2019) applied their DAE on 2D common shot gathers.

In our study, we attenuate noise from WTs at a seismological station by using two different denoising techniques. As a first method, we use nonlinear thresholding (Langston and Mousavi 2019), where time-frequency coefficients exceeding a certain threshold are attenuated. As a second method, we use a modified DAE (Zhu et al. 2019; Mandelli et al. 2019; Tibi et al. 2021), which takes a large training dataset to learn how to separate signal and noise. The DAE uses either the STFT as done by Zhu et al.
(2019) and Tibi et al. (2021) or the CWT to transform the noisy raw data into the time-frequency or time-scale domain. STFT and CWT have different resolutions (e.g. Mallat 2009), so CWT may be able to suppress noise more accurately, which was not tested in previous studies. Nonlinear thresholding can be applied on real data without any training, whereas the DAE has first to be trained. We compare raw data, bandpass filtered data, the results of the nonlinear thresholding, and the DAE, using different techniques for the estimation of the threshold and using STFT and CWT to determine the time-frequency coefficients for the DAE.

The paper is structured as follows: First, we present the selection of a suitable seismological station that is deteriorated by the installation of new WTs. Second, we introduce the different denoising techniques, where we focus first on nonlinear thresholding, the CWT, and how we tested different noise models to suppress the noise. Afterwards we introduce the DAE, show the differences to the existing DAEs by Zhu et al. (2019) and Tibi et al. (2021), and present the dataset that we used to train the DAE. In the last part, we show the results of the denoising techniques on real seismological data.

2 Site selection

To find a seismological station, which is negatively affected by surrounding WTs, we compute hourly power spectral density (PSD) curves for several stations in the state of North Rhine-Westphalia (NRW, Fig. 2a) and bin those curves by the local wind speed by the method as proposed by Stammler and Ceranna (2016). The results are average PSD curves for each wind speed bin. Figure 1 shows the results of the year 2012 (Fig. 1a) and 2019 (Fig. 1b) for the vertical channel of station RN.BAVN close to Haltern am See, Germany (Fig. 2b) in a frequency range between 1 and 10 Hz. Both seismic events and WT noise are in this frequency range (Stammler and Ceranna 2016).

Fig. 1 Wind-dependent power spectral density (PSD) noise spectra (in dB) for the vertical component of station RN.BAVN near Haltern am See between 1 and 10 Hz of the years 2012 (a) and 2019 (b). Colours indicate different wind speeds measured at the nearest weather station of the German Weather Service. Between the years 2012 and 2017, five new wind turbines were installed near the seismological station (Fig. 2), leading to a deterioration of the station quality.
When the spectral lines in Figure 1 are close together, there is no evidence for any correlation of wind speed and noise level (Stammler and Ceranna 2016). In 2012 (Fig. 1a), there exists already some wind dependency, which might result from a WT installed in 2009 or from trees in the surroundings (Fig. 2b), which also emit short-term seismic noise (Bormann and Wielandt 2013). In contrast in 2019 (Fig. 1b), there are distinct noise peaks at 3.2 Hz and 4.2 Hz which grow with wind speed. The comparison of the yellow PSD curves (Fig. 1) shows a deterioration of 20 dB between 2012 and 2019, which is a tenfold increase compared to the original noise amplitude. During this time period, five new WTs of different types were installed around the seismological station (Table 1). PSD curves at other stations where fewer or no WT were installed do not show similar wind dependencies. For this reason, we selected the seismological station RN.BAVN for our study. The geology at this site can be described as unconsolidated sediments and consists of Upper Cretaceous sands.

After selection of the seismological station, we installed short period seismometers (Raspberry Shakes RS3D and Mark L-4C3D) close to each WT (50 to 80 m away from the foundation). The seismological station codes and WT type are given for each station and WT. The green areas are forested, beige areas are used for agriculture, and grey areas are built on

![Fig. 2](https://www.openstreetmap.org)

### Fig. 2 Location of the state NRW in Germany (a) and an overview of the selected site west of Haltern am See (b). Between 2009 and 2017, six WTs (blue stars) were installed in the vicinity of the seismological station RN.BAVN (coordinates are given in parenthesis). For the development and testing of a denoising algorithm, new seismometers were installed at each WT (red triangles). The seismological station codes and WT type are given for each station and WT. The green areas are forested, beige areas are used for agriculture, and grey areas are built on.

### Table 1 WT types, hub heights, maximal power, construction year, and the distance to the affected seismological station (Energieatlas NRW 2022)

| WT type        | Hub height (m) | Max. power (kW) | Construction year | Distance to BAVN (km) |
|----------------|----------------|-----------------|-------------------|-----------------------|
| Senvion 3.2M 114 VG | 143            | 3170            | 2017              | 1.7                   |
| Senvion 3.2M 114   | 143            | 3170            | 2017              | 2.1                   |
| Senvion MM92       | 100            | 2050            | 2014              | 1.1                   |
| Senvion 3.2M 114   | 143            | 3200            | 2015              | 1.7                   |
| Senvion 3.0M 122   | 139            | 3000            | 2015              | 1.6                   |
| Senvion MM92       | 100            | 2050            | 2009              | 1.1                   |
data, the WT owner kindly provided the operation parameters of each WT, except one. At this WT, we did not install a new station (Fig. 2b). The new stations are used to find similar signals in WT noise to get an estimate of the disturbing signal at station RN.BAVN (see Section 3.1).

3 Methods

3.1 The CWT and nonlinear thresholding

Since the short-time Fourier transform (STFT) is limited in time-frequency resolution, we use the CWT to transform signals into time-scale representation. The CWT takes two independent variables, scale $a$ and time lag $\tau$, and is defined as (Mallat 2009)

$$W(a, \tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \Psi^* \left(\frac{t-\tau}{a}\right) dt,$$  \hspace{1cm} (1)

where $\Psi(t)$ is the mother wavelet and the asterisk represents the complex conjugate. Thus, the CWT is a correlation of the time series $x(t)$ with a scaled and dilated wavelet $\Psi(t)$. A wavelet is admissible when it has zero average, i.e.

$$\int_{-\infty}^{\infty} \Psi(t) \, dt = 0,$$  \hspace{1cm} (2)

and when the integral

$$C_\Psi = \int_{0}^{+\infty} \frac{|\hat{\Psi}(\omega)|^2}{\omega} \, d\omega$$  \hspace{1cm} (3)

exists. Here, $\hat{\Psi}(\omega)$ denotes the Fourier transform of the mother wavelet. Transforming from time-scale-domain back into time-domain is done by the double integral (Mallat 2009)

$$x(t) = \frac{1}{C_\Psi} \int_{0}^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{a}} W(a, \tau) \Psi \left(\frac{t-\tau}{a}\right) \frac{d\tau}{a^2} \frac{da}{a^2}.$$  \hspace{1cm} (4)

In this study, we use the implementation of the CWT and its inverse for discrete sequences by Torrence and Compo (1998). An example of an admissible mother wavelet is the Morlet wavelet, which is a Gaussian-modulated sinusoidal pulse (Farge 1992; Torrence and Compo 1998):

$$\Psi(x) = \pi^{-1/4} \exp(i\omega_0 x) \exp \left(\frac{-x^2}{2}\right),$$  \hspace{1cm} (5)

where $\omega_0$ is a nondimensional frequency and is set to 6 (Farge 1992; Torrence and Compo 1998) and $i$ denotes the imaginary unit. The real and imaginary parts of the wavelet are shown in Fig. 3. To compare the results of STFT and CWT, pseudo-frequencies $f_a$ can be determined by the relationship $f_a = f_0 a$ (e.g. Stockhausen 2016), in which $f_0$ is the centre frequency of the scaled wavelet and $a$ is the scale.

To attenuate disturbing noise, Donoho and Johnstone (1994) and Donoho (1995) introduced the concept of nonlinear thresholding in the wavelet-domain using an universal threshold. They assume that the recorded signal $x(t)$ is a superposition of the seismic signal $s(t)$ and some additive noise $n(t)$

$$x(t) = s(t) + n(t),$$  \hspace{1cm} (6)

and present a denoising method in which wavelet coefficients below some scale-dependent threshold $\beta(a)$ are considered as noise and suppressed. Since the noise is usually unknown, Donoho (1995) presents a method of keeping wavelet coefficients if they are greater than a threshold function $\beta(a)$. Otherwise, they are classified as pure noise and are set to zero. This is called hard thresholding. Hard thresholding results in lower signal-to-noise ratio (SNR) if noise outliers are above the threshold function. Alternatively, thresholding can also be done by shrinking all

![Fig. 3 Real (solid) and imaginary (dashed) parts of the Morlet wavelet in time domain (eq. 5)](image-url)
coefficients that exceed the threshold function. This concept is called soft thresholding and is given by (Weaver et al. 1991; Langston and Mousavi 2019)

\[
\hat{W}(a, \tau) = \begin{cases} 
\text{sign}(W(a, \tau))(|W(a, \tau)| - \beta(a)) & \text{if } |W(a, \tau)| \geq \beta(a) \\ 
0 & \text{otherwise}
\end{cases}
\]

where

\[
\text{sign}[W(a, \tau)] = \frac{W(a, \tau)}{|W(a, \tau)|}.
\]

The threshold function \(\beta(a)\) is determined from a noise estimate for each wavelet scale \(a\) and is estimated as (Langston and Mousavi 2019)

\[
\beta(a) = Q_p(|W(a, \tau)|) \quad (P = 0.99),
\]

where \(Q_p\) denotes the quantile function and \(P\) is the quantile value. Following Langston and Mousavi (2019), it is not necessary for the noise estimate to be characterised by a particular distribution function such as a Gaussian distribution. The noise estimate is either determined from a time window before (pre-noise) or after the expected event or from a different time window in the past or future with similar noise conditions. To find such time windows, we use records from the sensors close to the WTs. We hypothesise that, if the noise recorded close to a WT in the event window and a different time window are very similar, this also holds for the noise at the seismological station because the noise propagates through the ground from the WT to the station. Once such a highly similar time window is found in the records close to the WT, we can extract the waveform at the seismological station at the given time and use it as a noise model to determine the threshold function \(\beta(a)\).

To check this hypothesis, we took records form the sensors close to the WTs and calculated the Euclidean distance between the noise in the event window and other windows of equal length before the event. The normalised Euclidean distance of a master signal \(t_i\) and a subsequence \(t_j\) is defined as

\[
D_{ij} = \left[ \frac{\sum_{l=0}^{n-1} ((t_{i+l} - \mu_i) - (t_{j+l} - \mu_j))^2}{\sum_{l=0}^{n-1} (t_{i+l} - \mu_i)^2} \right]^{\frac{1}{2}},
\]

where \(\mu_i\) and \(\mu_j\) are the mean of each subsequence and \(n\) is the length of the master signal \(t_i\). Figure 4a shows an example of a synthetic time series where the similar signals with different amplitudes (black, red, blue, and pink signals in Fig. 4a) are hidden. The black signal is used as the master signal and the distance profile is computed (eq. 10) for this signal (Fig. 4b). The resulting distance profile is zero (global minimum) at the timestamp of the master signal and has local minima at the timestamps of similar signals. These are marked as stars in Fig. 4b.

Fig. 4 Example of a synthetic time series (a), a master signal (black), and the distance profile of the master signal and the time series (b). The red, blue, and pink signals have a different amplitude scaling but the same shape as the master signal and are hidden in the time series in (a). Local minima in the distance profile (black, red, blue, and pink stars in (b)) indicate the location of these similar signals in the time series. When extracting a similar signal using the distance profile, the location of the local minimum is taken as the starting point.
Compared to cross-correlation, which is commonly used in seismic data processing to extract similar signals, Euclidean distance is computationally more efficient for large datasets (Yeh et al. 2016; Zhu et al. 2016; Akbarinia and Cloez 2019). An average distance profile is formed from all distance profiles calculated at each WT with the event window as master signal and the average distance profile is searched for minima. The global minimum is where the master signal is located in the time series. Therefore, the global minimum is truncated and the minimum with the second smallest Euclidean distance is taken as the starting point for the most similar signal. The final step is to extract the waveform at the seismological station at the time of the minimum in the average distance profile, as this waveform follows from similar conditions at all WTs. Figure 5 summarises the workflow.

Figure 6 shows an example of such a similarity search using three WT stations. The red waveforms are the master signals for each WT and the grey waveforms are the most similar signal at the WT in a small range around the minimum of the average distance profile (Fig. 6a–c). Figure 6d shows the extracted waveform which is used as a noise model to estimate the threshold function $\beta(a)$ (eq. 9, red line) and the noisy waveform with an expected event in the middle (grey waveform). After applying the threshold function, the modified CWT coefficients $\hat{W}(a, \tau)$ are transformed back into time domain to obtain the denoised waveform. Figure 7 summarises the dataflow of nonlinear thresholding using either pre-noise or the extracted waveform as a noise model.

3.2 Denoising autoencoder

Zhu et al. (2019) introduced a novel time-frequency denoising technique, where a convolutional neural network (CNN) learns sparse representations of spectrograms and maps this representation into two mask functions for signal $M_S(t, f)$ and noise $M_N(t, f)$, respectively. In general, this method is called a denoising autoencoder (DAE). We start from the time-frequency representation of equation 6:

$$X(t, f) = S(t, f) + N(t, f).$$

The objective is to separate signal and noise by the computation of a denoised signal

$$\hat{S}(t) = TFT^{-1}\{M_S(t, f) \cdot X(t, f)\}$$

and the recovered noise

$$\hat{N}(t) = TFT^{-1}\{M_N(t, f) \cdot X(t, f)\},$$

where $TFT^{-1}$ denotes the inverse time-frequency transform. The DAE learns two mask functions

$$M_S(t, f) = \frac{1}{1 + \frac{|N(t, f)|}{|S(t, f)|}}$$

and

$$M_N(t, f) = \frac{|N(t, f)|}{1 + \frac{|N(t, f)|}{|S(t, f)|}} = 1 - M_S(t, f)$$

for signal and noise, respectively. During the training process, signal and noise samples from a database are added to construct a noisy signal. Since the DAE learns from real data, it is not required that the noise samples are characterised by a specific distribution function. The true mask functions $(M_S(t, f), M_N(t, f))$ interact as labels of the CNN and the DAE has the objective to minimise the expected error $E$ between

---

**Fig. 5** Flowchart of similarity search at WTs and extraction of the waveform at the seismological station. Details are given in the text.
the true mask functions and the predicted mask functions ($\hat{M}_i(t, f)$, $M_N(t, f)$):

$$\text{error} = E \left\| \hat{M}_i(t, f) - M_i(t, f) \right\|_2^2,$$

(16)

where $i$ is either $S$ or $N$ and $\| \cdot \|_2$ denotes the Euclidean norm. The mask functions for signal and noise consist of real values ranging from 0 to 1. The network architecture is similar to those CNNs presented by Zhu et al. (2019) and Tibi et al. (2021) and is shown in Fig. 8. However, the input size and the depths of the CNN differ by the chosen time-frequency representation and we add dropout to each convolutional layer to prevent the neural network from overfitting (Srivastava et al. 2014). We also use skip connections to improve the convergence of the network (Ronneberger et al. 2015; Liu and Yang 2018).

The input layer consists of the real and imaginary parts of the previously estimated time-frequency representation, which are min-max-normalised independently of each other, i.e. the values of the input layers are transformed to the range $[0, 1]$. In the first half of the network (encoder), the input data are converted into a multitude of sparse representations by passing them through a series of 2D convolutional layers with a constant filter size of $3 \times 3$, followed by a rectified linear unit (ReLU) as activation function, batch normalisation (BN) to improve the optimisation of the network (Ioffe and Szegedy 2015; Goodfellow et al. 2016), and a dropout layer at the seismological station (d). The grey waveform is the noisy signal and the red waveform is used to determine the threshold function $\beta(a)$ for the denoising process. The waveforms at the WTs used to determine the distance profile have a length of 6 s (a–c), while the noise model (d) has a length of 60 s to obtain a more representative noise model.

---

**Fig. 6** Example of similarity search of a master signal (red waveforms) at three WTs (a–c) and the resulting extracted waveform (red line in (a–c)) that has the smallest Euclidean distance (distance in a–c) to the master signal and is thus the most similar signal. After finding the most similar signal at each WT in a given time window, the waveforms are extracted.
(blue arrows in Fig. 8). To reduce the size, the output of the layer above is passed through a 2D convolution with constant filter size (3 x 3) and strides, followed again by a ReLU, BN, and a dropout layer (orange arrows in Fig. 8). The stride size depends on the size of the input layer. We used either 2 x 2 or 2 x 3 as stride size for STFT and CWT, respectively. In the second part of the network (decoder), the sparse representations are upsampled using 2D deconvolution (transpose of the convolutional layers) to generate a mapping function for the two output masks. Since the dimension does not fit in every case with transposed convolution, a cropping layer is added to get the correct size (green arrows in Fig. 8). The last hidden layer is convolved with a 1 x 1 filter and passed through a softmax activation function (Goodfellow et al. 2016) to generate the mask functions for signal and noise, respectively. During the training process, the pre-estimated mask functions act as labels and the network learns how

**Fig. 8** Architecture of the neural network adapted from Zhu et al. (2019) and Tibi et al. (2021). The input layer consists of the real and imaginary parts of the time-frequency coefficients of the noisy time series. The outputs are two mask functions for the recovered signal and the noise. The rectangles with different sizes represent the hidden layers. Their size depends on the shape of the input layers and on the stride size. Arrows indicate different operations within the neural network. The input layers are down-sampled by applying 2D convolution and strides and up-sampled by transpose convolution and strides. Since the up-sampled layers do not have to be the same size as the down-sampled layers, these layers are cropped during upsampling. Dots indicate a repetition of the previous operations. Batch normalisation and skip connections are used to improve the convergence of the neural network. A softmax layer is used as the last layer to generate the mask functions for signal and noise.
to map from the real and imaginary part of the
time-frequency coefficients to the sparse represen-
tations, and from there to the output layer by optim-
mising the binary-crossentropy that is used as the
loss-function.

Figure 9 summarises the data flow of the DAE.
First, the noisy time series (Fig. 9a) is transformed
into time-frequency domain either by CWT or
STFT (Fig. 9b). The time-frequency coefficients are
then fed into the previously trained neural network
(Fig. 9c), which has learned to distinguish between
signal and noise during the training process. The
network outputs two individual mask functions for
signal and noise which are element-wise multiplied
with the noisy time-frequency coefficients to get two
time-frequency representations for signal (Fig. 9d(I))
and noise (Fig. 9d(II)). Finally, the time-frequency

3.3 Data and network training

To train the CNN, we use the Stanford Earthquake Data-
set (STEAD) which contains 1,050,000 local earthquakes
(Mousavi et al. 2019). We select signals with high SNR
(SNR ≥ 20 dB) and use 100,000 of these seismograms as a
training dataset. Each waveform is 60 s long and is recorded
with a sampling rate of 100 Hz. More than 100,000 wave-
forms with a duration of 60 s are randomly taken as noise
samples from the affected station. To ensure that the time
window contains only pure noise, the data are checked by
an STA/LTA (Withers et al. 1998) for any spikes and the
periods in which earthquakes have been detected at surrounding stations are excluded. For a more accurate denoising, we further selected noise samples close to the expected onset of the P-phase, which is estimated from onsets at other stations in the seismological network. During the training process, at the beginning of each training iteration (epoch), the earthquake dataset is randomly split into one training dataset comprising 80% of the full dataset and one validation dataset (20%). At the start of each epoch, first the signal is randomly shifted (data augmentation), then a noise sample is randomly selected from the noise dataset, and both signal and noise are scaled independently by a positive random number. As a last step, we add noise and signal to construct a noisy seismogram. By applying these steps, we build a dataset with millions of different signal and noise samples for three component seismograms.

We trained the DAE on an Nvidia Quadro RTX 6000 GPU with a batch size of 16 for CWT and 32 for STFT, for a maximum number of 200 epochs, and we used the Adam optimiser (Kingma and Ba 2014) with a learning rate of 0.001. The input layer has size $121 \times 100 \times 2$ when STFT is used and $3001 \times 200 \times 2$ when CWT is used. With CWT, the data are downsampling by a factor of 2; otherwise, the input layer would be too large for the memory. The loss of the validation data is monitored during the training and training stops when the validation loss does not improve further. The CNN training takes a few hours on a single GPU if STFT is used as the time-frequency representation, and up to several days for CWT. The network is built using the Python package Keras (Chollet 2015) on top of the machine learning platform Tensorflow (Abadi et al. 2015) and contains $\sim 1.2$ million trainable parameters and $\sim 2500$ nontrainable parameters (parameters that are not updated during the training process, e.g. mean and standard deviation in BN layers), when STFT is used as time-frequency representation and $\sim 19$ million trainable and $\sim 10,000$ nontrainable parameters, when CWT is used. The difference results from the different resolutions of STFT and CWT, respectively.

## 4 Results

### 4.1 Performance of the DAE

To evaluate the performance of the trained network, we create a test dataset from events and noise signals, which were not used for training. Our test dataset contains 5000 noisy waveforms. Since the waveforms of STEAD (Mousavi et al. 2019) contain phase onsets, we are able to determine the SNR in decibels (dB) for the raw, noisy, and denoised signal by

$$\text{SNR} = 20 \cdot \log_{10} \left( \frac{A_S}{A_N} \right),$$  \hspace{1cm} (17)$$

where $A_S$ is the root mean square for a 3-s-wide window around the P-arrival and $A_N$ denotes the root mean square for a 3s-wide window before it. The results are shown in Fig. 10a for CWT and in Fig. 10b for STFT as time-frequency transformation, respectively. Both networks are not able to reach the original SNR (grey bars) but in general, both methods result in a higher SNR (blue bars) in comparison to the noisy signals (green bars).

The mean SNR of the denoised data increases by 26.6 dB for both CWT and STFT (Fig. 10). Comparing the denoised histograms, CWT achieves a slightly better result than STFT, as the first standard deviation of CWT is slightly larger than that of the STFT. Furthermore, the denoised test dataset using the CWT has more events in the range 35 to 45 dB (Fig. 10a and b).

We apply both networks on real data at station RN.BAVN. For this, we use all events with a magnitude greater than 1 between 2017 and 2020 in the state NRW, Germany (2421 events), and use the event list of the seismological observatory at the Ruhr-University Bochum. Since many events are fully covered by noise at the seismological station, we estimate the arrival times at the station from analysed events at the surrounding stations. Thus, we are able to determine the SNR before and after denoising. The mean SNR is increased by 4.4 dB (Fig. 10c) and 4.2 dB (Fig. 10d) for CWT and STFT, respectively. Again, the first standard deviation for the CWT is slightly larger than for STFT. The histograms show clearly that many events with low SNR retained a low SNR after denoising. The CWT also performs slightly better on real data (Fig. 10c and d), especially when comparing denoised events with an SNR $\geq 40$ dB.

The improvements in SNR for the test and the real dataset are shown in Fig. 11. Both trained DAE models are able to improve the SNR of the noisy test dataset (Fig. 11a), but applied on the real dataset, the...
The average improvement for the test dataset is 20.6 dB and 4.6 dB or 4.4 dB for the real dataset when either CWT or STFT is used. Few events are improved by more than 30 dB (Fig. 11b). From both datasets, it follows that the DAE using the CWT performs slightly better, especially on real data, as the first standard deviation is slightly larger with the CWT (Fig. 11b).

4.2 Comparison with nonlinear thresholding and bandpass filter

We compared the performance of the DAE with both nonlinear thresholding and a bandpass filter between 5 and 20 Hz (Butterworth filter of order 4) on several selected events. As described in section 3.1, we use two techniques to determine the threshold function:
either a noise window before the first arrival or a noise model estimated from the similar signals at the WTs (eq. 10). Figure 12 shows a waveform where an event is detectable in the raw data (Figs. 12a(I) and 16a(I)). The time-frequency representation (Fig. 12b(I)) shows some high energy around 4 Hz. Applying the bandpass filter leads to an increase in SNR by 6.9 dB (Figs. 12a(II) and 16a(II)) and it also removes all signals, which are not in the passband. Using the threshold function estimated in a 30-s-wide time window before the first arrival yields an increase in SNR by 22.9 dB (Figs. 12a(III) and 16a(III)). Using a noise model only results in an increase of 11.8 dB (Fig. 12a(IV)), but the noise level is lower than in the raw or filtered data. However, the time-frequency representation still contains some high energy spots around 4 Hz, which are not considered as noise by the threshold function. Both trained DAEs (Figs. 12(V), (VI), and 16a (V), (VI)) are able to fully remove the noise before the first arrival, which results in an increase in SNR of around 30 dB. Also, the time-frequency representations do not contain any high energy spots before or after the event.

A second example shows an event that is also visible in the raw data (Figs. 13a(I) and 16b(I)), but whose SNR is not improved by the bandpass filter (Figs. 13a(II) and 16b(II)). Again, using pre-noise for the threshold function suppresses the noise well (Figs. 13a(III) and 16b(III)) and increases the SNR by 14 dB. In comparison, the pre-estimated noise model still has a moderate noise level, which results in an increase in SNR by only 6.4 dB compared to the raw data. The DAE that uses the CWT as time-frequency representation has the highest increase in SNR of 32.9 dB (Fig. 13(V)). Both DAEs suppress the noise well, but for the DAE using the STFT (Figs. 13(VI) and 16b(VI)) some noise remains before the first arrival, which explains the difference in SNR.

In the third example (Figs. 14(I) and 17a(I)), the SNR is low in the raw data and in the time-frequency representation some high energy peaks can be recognised after 30 and 50 s. Applying the bandpass filter on the raw data does not lead to an improvement (Figs. 14(I) and 17a(I)). Both thresholding functions make the event easier to detect (Fig. 14(III) and (IV)), but the resulting waveforms have spikes, which hinders further analyses. The DAEs increase the SNR by more than 20 dB and suppress the noise well (Figs. 14(V), (VI), and 17a(V) and (VI), b(V), and (VI)). Comparing the amplitudes of both waveforms demonstrates that the DAE using CWT has more loss in amplitude than the DAE using STFT.

The final example in Figs. 15 and 17c shows an event that is fully covered by noise. Noise and signal of the raw data (Figs. 15a(I) and 17c(I)) do not differ in amplitude, which results in an SNR close to zero. Also, the bandpass filter neither improves the SNR nor makes the event visible (Figs. 15a(II) and 17c(II)). The time-frequency representations of the raw data and the filtered data do not give a hint
as to when the event might be. Nonlinear thresholding also fails (Fig. 15(III) and (IV)), no matter which method is used. Only the DAEs are able to distinguish between noise and signal and make the event visible and detectable (Figs. 15(V), (VI) and 17c (V), (VI)).

Figures 12 to 17 show the vertical component of the recorded data only. Usually, seismometers record in three directions: one vertical component and two horizontal components in east-west and north-south direction, respectively. However, the horizontal components can be processed in the same manner since the DAE learns to denoise all three components during training. Figure 18 shows an example of a three component seismogram. The left column (Fig. 18a) presents the raw data and the middle and right column the denoised data using the DAE using STFT (Fig. 18b) and the DAE using CWT (Fig. 18c), respectively. The upper row shows the vertical (Fig. 18(I)), the middle row the north-south (Fig. 18(II)), and the lower row the east-west component (Fig. 18(III)). The denoising results do not differ significantly when the horizontal components are
included, and we are able to determine the arrival of the S-phase, which can be recognised by the high amplitudes in the horizontal components.

5 Discussion and conclusions

The objective of this work is the suppression of interfering WT noise in seismological recordings using different noise suppression techniques. For this purpose, we calculated wind-dependent PSD curves at several stations in NRW in order to find a suitable seismological station that is degraded by the installation of new WTs. We found a station close to Haltern am See, where six WTs are in the vicinity of one seismological station (Fig. 2). As a first method, we used nonlinear thresholding to suppress the disturbing noise by defining a threshold function (Fig. 7) either from a noise window before an event or from a previously calculated noise model. To obtain a noise model, we first installed additional seismometers at each WT to record the noise directly at the
source, extracted similar waveforms at each WT using Euclidean distance (Figs. 4, 5, and 6), and used the data recorded at the seismological station RN.BAVN at the time when similar signals were found at the WTs as our noise model (Fig. 6). As a second method, we used a DAE which learns from a training dataset how to distinguish between signal and noise. As a training dataset, we used local earthquakes from STEAD (Mousavi et al. 2019) and took noise samples from the affected seismological station. Furthermore, we tested two different time-frequency transformations (STFT and CWT) for the DAE. Zhu et al. (2019) and Tibi et al. (2021) only used the STFT which has a lower time-frequency resolution than CWT. We applied a bandpass filter between 5 and 20 Hz to compare the different denoising techniques.

The results on real seismological data (Figs. 12 - 18) show that nonlinear thresholding using pre-noise well suppresses the noise, when the event is detectable in the raw or bandpass filtered data. This is consistent with the findings of Mousavi et al. (2016) and Langston and Mousavi (2019). Using pre-noise

---

**Fig. 14** The same as in Fig. 12 but for different waveforms with a lower SNR in the raw data (a(I)). Magnified windows of the P- and S-phase onsets are shown in Fig. 17a and b.
instead of calculating a noise model from the distance profile is more efficient in suppressing noise. Furthermore, the estimation of a noise model is more complex, since additional seismometers at the WTs are required and it is necessary to determine a noise model before denoising an event, which is time consuming. Using pre-noise is therefore superior (to the noise-model determination) as it can almost be done in real time. When denoising events where signal and noise have similar amplitudes (Fig. 14) or the event is fully covered by noise and can neither be recognised in the waveform nor in the time-frequency representation, nonlinear thresholding always fails for our examples. We have found that the resulting waveforms contain many spikes, making them unusable for further seismological analysis.

In our shown examples, the DAE is always able to remove the noise from the raw data and make the event visible in both the time and time-frequency domain. To test our DAE models, we constructed artificial noisy signals by randomly adding signal and noise from our dataset, which was not used in
the training process (Fig. 10a and b). Using STFT or CWT as a time-frequency transform does not result in a big difference in the denoised data. Also both trained DAE models show similar results, when they are applied on the real dataset recorded between 2017 and 2020, which contains 2421 events (Fig. 10c and d). However, for both the test dataset (Fig. 11a) and the real dataset (Fig. 11b), some data are not improved or in some cases noise remains before the P onset (Figs. 16b(VI), 17a (V) and (VI), and 17c (V) and (VI)). This might result from a too high noise level in the constructed noisy dataset or in the real dataset, so that the DAE is not able to remove the noise, i.e. the DAE classifies the selected time window as noise and is not able to find any signal. Here, a too high noise level cannot be specified because the results (Fig. 10-18) do not show systematic limits of the DAE. Finding these limits needs further investigations.

In conclusion, there is no big difference when using CWT or STFT as time-frequency transformation. Furthermore, since CWT results in a larger size for the input layer, the training of the DAE is more time consuming than for STFT. However, both presented DAEs are able to suppress WT noise and they are also applicable on three component data (Fig. 18). The DAE is the only one of the methods investigated that is capable of separating noise and signal well, especially when signal and noise share a common frequency band.
As Zhu et al. (2019) already note, the DAE does not result in a perfect separation of signal and noise, since a perfect separation requires the recovery of two complex spectra for signal and noise. Instead, the DAE learns how to map the real and imaginary parts of the time-frequency domain to two mask functions.
for signal and noise. Furthermore, the mask function only has values between [0, 1], and thus, the DAE is not able to recover signal values which are larger than the noisy signal (Zhu et al. 2019). However, both DAEs do not lead to time shifts or signal deformations and the polarity is not changed (Figs. 16 and 17). How the DAE affects earthquake localisation, magnitude estimation, and other seismological analyses needs to be further investigated, especially when earthquake data are only visible on stations after denoising. The use of a DAE could therefore lead to more earthquake detections.

As a future improvement, one could assume a real and an imaginary output layer for the signal instead of two mask functions for signal and noise, respectively. The DAE is then trained by minimising the error between the true and the predicted signal and may be able to learn more from the phases of the spectrogram.

Once a DAE model has been trained for a station, it can be used as long as the noise at the station does not change. When the noise at the station changes, new training might be necessary. We also tested the DAE on real time denoising. To do this, we split the incoming data from the seismological station into overlapping 60-s windows, denoise each window, and merge the individual windows back into one trace using ObsPy (Beyreuther et al. 2010). Afterwards, the denoised trace can be analysed by common seismological software packages. In general, the DAE can be trained with any type of noise depending on the application. At the moment, the training dataset used (Mousavi et al. 2019) consists only of local earthquake data. Adding regional and teleseismic data to STEAD could be a future improvement.

**Fig. 18** Waveforms for raw data (a), after applying either the DAE using the STFT (b) or the CWT (c) as time-frequency representation for the vertical (I), north-south (II), and east-west (III) components

**Acknowledgements** The authors would like to thank the owners of the wind turbines near Haltern am See (North Rhine-Westphalia, Germany) for their cooperation, sharing of the wind turbines’ operational data, and permission to install new seismometers. Many thanks to the seismology working group at the Ruhr-University Bochum and the anonymous reviewers for proofreading this work and their helpful recommendations.

**Funding** Open Access funding enabled and organized by Projekt DEAL. The MISS research project was funded by the “Europäische Fonds für regionale Entwicklung (EFRE)”.

**Declarations**

**Conflict of interest** The authors declare no competing interests.
References

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous systems. URL https://www.tensorflow.org/, software available from tensorflow.org

Akbarinia R, Cloez B (2019) Efficient matrix profile computation using different distance functions. arXiv preprint arXiv:1901.05708

Beyreuther M, Barsch R, Krischer L, Megies T, Behr Y, Wassermann J (2010) ObsPy: a Python toolbox for seismology. Seismological Research Letters 81(3):530–533. https://doi.org/10.1785/gssrl.81.3.530

Bormann P, Wielandt E (2013) Seismic signals and noise. In: New Manual of Seismological Observatory Practice 2 (NMSOP2), Deutsches GeoForschungsZentrum GFZ, pp 1–62

Chollet F (2015) Keras available at https://github.com/fchollet/keras (last accessed june 2021)

Donoho DL (1995) De-noising by soft-thresholding. IEEE transactions on information theory 41(3):613–627. https://doi.org/10.1109/1.830209

Donoho DL, Johnstone JM (1994) Ideal spatial adaptation by wavelet shrinkage. biometrika 81(3):425–455. https://doi.org/10.1093/biomet/81.3.425

Energieatlas NRW (2022) Energieatlas NRW. URL https://www.energieatlas.nrw.de/site, last accessed: 2022-04-12

Estrella HF, Korn M, Alberts K (2017) Analysis of the influence of wind turbine noise on seismic recordings at two wind parks in Germany. Journal of Geoscience and Environment Protection 5(05):76. https://doi.org/10.4236/gep.2017.55006

Farge M (1992) Wavelet transforms and their applications to turbulence. Annual review of fluid mechanics 24(1):395–458. https://doi.org/10.1146/annurev.fl.24.010192.002143

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press

Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Bach F, Blei D (eds) Proceedings of the 32nd International Conference on Machine Learning, PMLR, Lille, France, Proceedings of Machine Learning Research, vol 37, pp 448–456. URL https://proceedings.mlr.press/v37/ioffe15.html

Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980 URL https://arxiv.org/abs/1412.6980

Langston CA, Mousavi SM (2019) Separating signal from noise and from other signal using nonlinear thresholding and scale-time windowing of continuous wavelet transforms. Bulletin of the Seismological Society of America 109(5):1691–1700

Lerbs N, Zieger T, Ritter J, Korn M (2020) Wind turbine induced seismic signals: the large-scale SMARTIE1 experiment and a concept to define protection radii for recording stations. Near Surface Geophysics. https://doi.org/10.1785/10201900073

Liu JY, Yang YH (2018) Denoising auto-encoder with recurrent skip connections and residual regression for music source separation. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, pp 773–778, https://doi.org/10.1109/ICMLA.2018.00123

Mallat S (2009) A wavelet tour of signal processing: The sparse way. AP Professional, Third Edition, London. https://doi.org/10.1016/B978-0-12-374370-1.X0001-8

Mandelli S, Lipari V, Bestagini P, Tubaro S (2019) Interpolation and denoising of seismic data using convolutional neural networks. arXiv preprint arXiv:1901.07927

Mousavi SM, Langston CA, Horton SP (2016) Automatic microseismic denoising and onset detection using the synchrosqueezed continuous wavelet transform. Geophysics 81(4):V341–V355. https://doi.org/10.1190/geo2015-0598.1

Mousavi SM, Sheng Y, Zhu W, Beroza GC (2019) Stanford earthquake dataset (STEAD): a global data set of seismic signals for AI. IEEE Access 7:179464–179476. https://doi.org/10.1109/ACCESS.2019.2947848

Neuffer T, Kremers S (2017) How wind turbines affect the performance of seismic monitoring stations and networks. Geophysical Journal International 211(3):1319–1327. https://doi.org/10.1093/gji/ggx370

Neuffer T, Kremers S, Fritschen R (2019) Characterization of seismic signals induced by the operation of wind turbines in North Rhine-Westphalia (NRW), Germany. Journal of Seismology pp 1–17. https://doi.org/10.1007/s10950-018-9866-7

Neuffer T, Kremers S, Meckbach P, Mistler M (2021) Characterization of the seismic wave field radiated by a wind turbine. Journal of Seismology. https://doi.org/10.1007/s10950-021-10003-6

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, pp 234–241, https://doi.org/10.1007/978-3-319-24574-4_28
Saccorotti G, Piccinini D, Cauchie L, Fiori I (2011) Seismic noise by wind farms: a case study from the Virgo Gravitational Wave Observatory, Italy. Bulletin of the Seismological Society of America 101(2):568–578. https://doi.org/10.1785/0120100203

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15(1):1929–1958

Stammler K, Ceranna L (2016) Influence of wind turbines on seismic records of the Gräfenberg array. Seismological Research Letters 87(5):1075–1081. https://doi.org/10.1785/0220160049

Stockhausen N (2016) Methoden der digitalen Signalverarbeitung: mit interaktivem Lernprogramm und LabView-Experimenten. John Wiley & Sons

Styles P, Stimpson I, Toon S, England R, Wright M (2005) Microseismic and infrasound monitoring of low frequency noise and vibrations from windfarms: recommendations on the siting of windfarms in the vicinity of Eskdalemuir, Scotland. Keele University (UK), Report for the Ministry of Defence 18

Tibi R, Hammond P, Brogan R, Young CJ, Koper K (2021) Deep learning denoising applied to regional distance seismic data in Utah. Bulletin of the Seismological Society of America. https://doi.org/10.1785/0120200292

Torrence C, Compo GP (1998) A practical guide to wavelet analysis. Bulletin of the American Meteorological society 79(1):61–78. https://doi.org/10.1175/1520-0477(1998)079<0061:APGTWA>2.0.CO;2

Weaver JB, Xu Y, Healy D Jr, Cromwell L (1991) Filtering noise from images with wavelet transforms. Magnetic Resonance in Medicine 21(2):288–295. https://doi.org/10.1002/mrm.1910210213

Windenergie-Erlass - BayWEE (2016) Hinweise zur Planung und Genehmigung von Windenergieanlagen (WEA). Available at: https://www.stmwi.bayern.de/fileadmin/user_upload/stmwi/Publikationen/2016/Windenergie-Erlass_2016.pdf

Windenergie-Erlass NRW (2018) Erlass für die Planung und Genehmigung von Windenergieanlagen und Hinweise für die Zielsetzung und Anwendung. Available at: https://recht.nrw.de/limi/owa/br_bes_text?anw_nr=1&gld_nr=2&ugl_nr=2310&bes_id=38805&val=38805&ver=7&sg=0&aufgehoben=N&menu=1

Withers M, Aster R, Young C, Beiriger J, Harris M, Moore S, Trujillo J (1998) A comparison of select trigger algorithms for automated global seismic phase and event detection. Bulletin of the Seismological Society of America 88(1):95–106. https://doi.org/10.1785/BSSA088010095

Yeh CCM, Zhu Y, Ulanoa L, Begum N, Ding Y, Dau HA, Silva DF, Mueen A, Keogh E (2016) Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In: 2016 IEEE 16th international conference on data mining (ICDM), IEEE, pp 1317–1322, https://doi.org/10.1109/ICDM.2016.0179

Zhu W, Mousavi SM, Beroza GC (2019) Seismic signal denoising and decomposition using deep neural networks. IEEE Transactions on Geoscience and Remote Sensing 57(11):9476–9488. https://doi.org/10.1109/TGRS.2019.2926772

Zhu Y, Zimmerman Z, Senobari NS, Yeh CCM, Funning G, Mueen A, Brisk P, Keogh E (2016) Matrix profile II: exploiting a novel algorithm and GPUs to break the one hundred million barrier for time series motifs and joins. In: 2016 IEEE 16th international conference on data mining (ICDM), IEEE, pp 739–748, https://doi.org/10.1109/ICDM.2016.0085

Zieger T, Ritter JR (2018) Influence of wind turbines on seismic stations in the Upper Rhine Graben, SW Germany. Journal of Seismology 22(1):105–122. https://doi.org/10.1007/s10950-017-9694-9