# How long before strike can we predict earthquakes with an LSTM neural network?

Amaury Charlot
Responsible professor: Elvin Isufi
Supervisor: Mohammad Sabbaqi
EEMCS, Delft University of Technology, The Netherlands

June 19, 2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

**Abstract**

Different methods have been studied to predict earthquakes, but the results are still far from optimal. Due to their seemingly dynamic and unpredictable nature, it has been very hard to find data correlating with earthquakes happening. But recently, various research has been done using neural networks, and some has suggested that it could extract valuable information from preceding seismic data. To get a better sense of how seismic data can contain this information, we need to look at how long before an earthquake seismic precursor signals can exist. This paper uses an LSTM model to perform binary classification of the task: ”Given the seismic wave recordings of N stations during T seconds, will an earthquake happen after H seconds?” By varying the parameter H and studying its effect on the prediction accuracy of the NN, results suggest that sensitive information is very present in the seismic data 10 to 15 minutes before a low-magnitude (less than 2.5 on Richters scale) earthquake strikes. We aim to open the way for further research about precursor-based earthquake prediction using neural networks, showing that LSTM can be a good option. We also hope for further research to dig deeper in understanding what the signals in the seismic data are to further improve earthquake prediction.

# 1 Introduction

An earthquake is a sudden shake of the earth’s surface realizing energy and thereby creating seismic waves [1]. They can be the cause of a lot of damage, including but not limited to surface faulting, soil liquefactions, ground resonance etc. [2]. This has a lot of consequences, ranging from ecological and economical ones to costing human lives, which makes predicting earthquakes very promising. But as of now, it is not possible to precisely predict earthquakes well in advance [3]. Current research, a.o, tries to investigate the possibility of early warning systems, which could be of great use for the population [3].

It is not known exactly how seismic movement preceding an earthquake correlates with it. If research such as [4] can analyze the behavior of seismicity before earthquakes, it does not show a clear link. Recently though, interest has grown in using deep learning techniques to try achieving precursor-based earthquake prediction [5]. For example, Ibrahim et al [6] proposed a comparison of 1D convolutional neural networks (CNN), 2D CNNs as well as recurrent neural networks (RNN). They found that some seismic precursor signal may exist and suggested RNNs to be a preferable option. Other research such as [7] insists on the importance of spatial knowledge and showed that introducing spatial parameters could increase performance, while Q. Wang et al. [8] proposed using a variant of RNNs called long short-term memory (LSTM) to ”learn the spatio-temporal relationship among earthquakes in different locations and make predictions by taking advantage of that relationship”. They found LSTM to effectively exploit that relationship and to make better predictions. Other literature tells us that LSTMs ”are one of the most recent and promising developments in the time-series analysis” [9] and that it tends to perform better than other NNs on unseen earthquake data [10].

To get a better insight on why these neural network approaches seem to work, we need to understand where the correlation comes from. In this regard, this paper will bring answers to the question ”How many seconds before strike can we accurately predict low-magnitude earthquakes?”. This would allow us to confirm that some seismic precursor signal does exist [6] and show us how long before an earthquake it is present.

To do so, the paper will use an LSTM network that works with the data of 36 different stations in New-Zealand. This way the model can exploit the spatial relationship [8] and build on a neural network technique that seems promising. In the next sections, we will present a modelling of the problem and explain how this was then implemented and with which data. We then explain the experiment set-up where we ran the same model multiple times, comparing how seismic data of different timings before an earthquake strike would perform. We hypothesize that the performance of the model will go down when data is taken from longer before, but nothing gives an idea in which measure. The results will then be presented and a conclusion will be drawn. Suggestions, limitations, future work will also be discussed at the end.

# 2 Methodology

## 2.1 Problem modelling

The problem was modelled as follows. Given the seismic wave recordings of N stations during T seconds, the task of the neural network is to binary classify between an earthquake happening and no earthquake happening after H seconds.

## 2.2 Data

The data set used is provided by the International Federation of Digital Seismograph Networks (FDSN) [11] and looks at earthquakes happening between 1999 and 2020 in New-Zealand. Approximately half of the seismic recordings were from before an earthquake, half from normal behaviour.

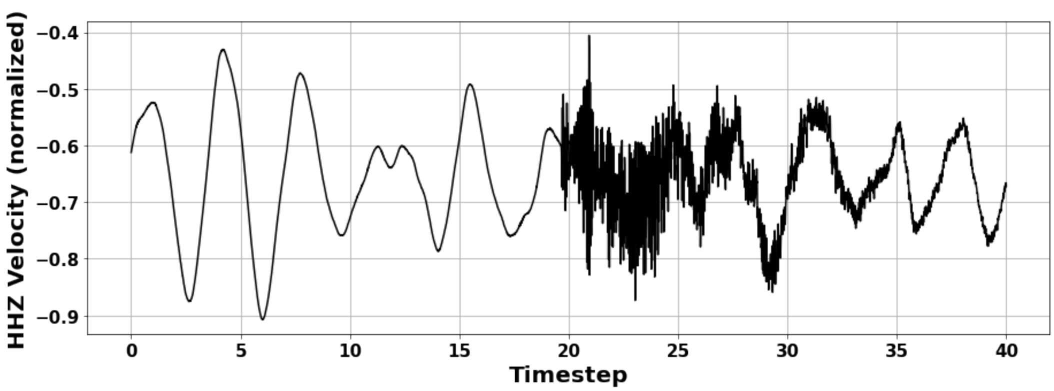



Figure 1: Example of seismic wave recording with an earthquake happening at timestep 20s.

First, all earthquakes with their information were retrieved. Since this paper specifically looks at low-magnitude earthquakes, high-magnitudes earthquakes had to be filtered out. For this, 2.5 was considered as the limit between high and low-magnitude. This limit yields an almost equal balance in the number of low and high magnitude earthquakes. The data of seismic wave recordings before those earthquakes were then downloaded using the API provided by FDSN [11], from 36 available recording stations in New Zealand. This data can be described as the ground's motion in function of the time (also called a seismogram) [12].

Seismic recording from normal behaviour was then retrieved by selecting times in between two time-distant earthquakes.

### 2.3 LSTM model

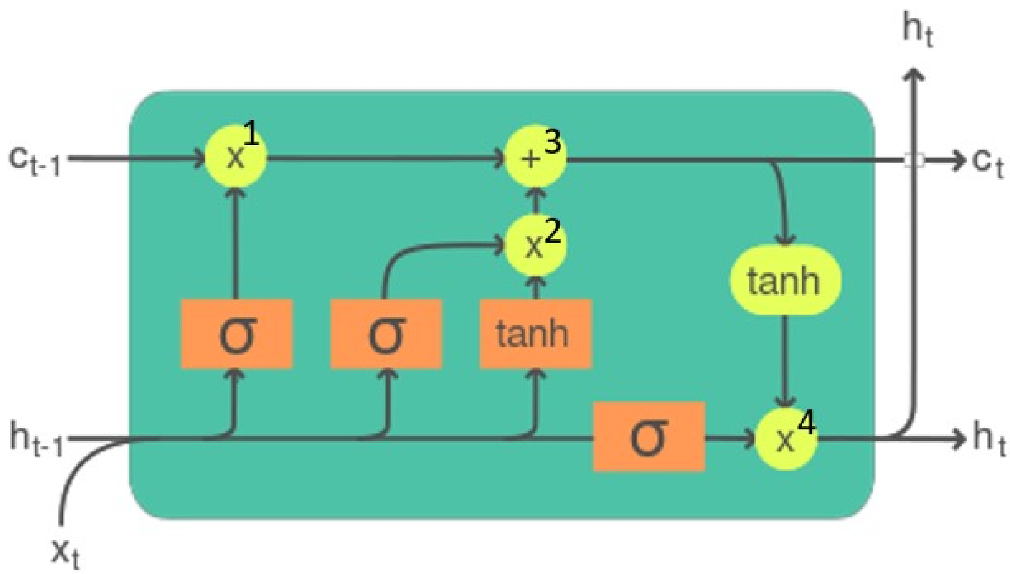



Figure 2: LSTM cell structure: 1) forget gate, 2) learn gate, 3) remember gate, 4) use gate.

As mentioned previously, the neural networked used is an LSTM. These have the advantage of being able to handle time series, and are designed to retain information from before (long memory). LSTM's are similar to classic Recurrent Neural Networks (RNN's), but they have the addition of retaining long-time information. This is achieved with a 4-gate structure [?] of each cell, with a forget gate, a learn gate, a remember gate and a use gate as shown in figure 2. The forget gate acts to discard unuseful long-term information from previous iterations. The learn gate combines a sigmoid and a hyperbolic tangent activation function to learn information which is then combined by the remember gate with the kept long-term information to update the long-term memory. Finally, the use gate combines this memory with the input to make the output prediction.

$$\begin{pmatrix} EQ1, S1 & \cdots & EQ1, S36 & 0/1 \\ \vdots & \ddots & \vdots & \vdots \\ EQn, S1 & \cdots & EQn, S36 & 0/1 \end{pmatrix}$$

Figure 3: Representation of the input.

In our case, the LSTM's input consists of N features which are the seismic recording waves of each of the N used stations. This is shown in figure 3. This matrix represents the structure of the input. Each row is a sample, $EQ_n$ denotes the $n^{th}$ event and $S_m$

the $m^{th}$ station. The last column are the labels 0 or 1, respectively normal behavior and earthquake. It runs through one LSTM cell with a hidden layer size of 2, meaning the model creates a 2-feature output for the long-term memory (denoted c in figure 2) The model was programmed in python using PyTorch [13], an open source machine learning framework

# 3 Experiment

## 3.1 Data pre-processing

Figure 4: The 36 stations are shown by the red dots on the New-Zealand map.

Firstly, the stations to use had to be determined. For this, we retrieved all stations located in the "bounding box" of New-Zealand. These are all stations with location's longitude between 166.104 and 178.990, and latitude between -47.749 and -33.779. This leaves us with 91 stations. Later on, when processing the seismic recording data (see below), some stations were found to not have information for all recordings. Those were discarded. Also some stations, when experimenting with them individually, seemed to have a negative impact on the overall performance. They were again discarded. 36 stations remained, which can be seen in figure 4

Next, earthquake events were filtered. We considered earthquakes with magnitudes from 1 to 2.5, a range in which most of the low-magnitude earthquakes are (figure 5). This left 262 040 earthquakes. To get a sample of manageable size thereof, and based on the logical hypothesis that deeper earthquakes would yield less precise predictions, since the recordings are from ground level, events with depth below 5km were retained, yielding a total of 5674 earthquakes.

Then, for each earthquake, seismic recording data was retrieved. The recordings of the 36 stations were downloaded for the 60 seconds preceding the strike, as well as 11 recordings of 30 seconds each, with the time before strike varying from 60 to 600 seconds with 60 seconds steps (cfr. Section 3.3: Set-up). All wave recordings were normalized.

To retrieve normal seismic data, i.e. recordings that would be classified as 0 (no earthquake happening), all earthquake events were sorted by the time

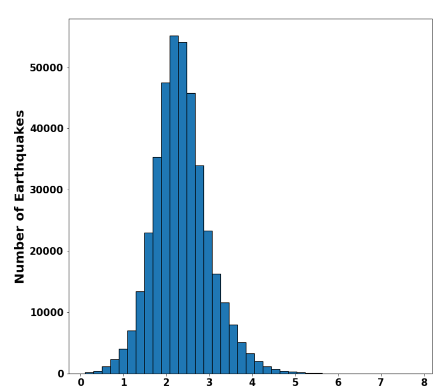



Figure 5: Distribution of the magnitudes of all earthquakes in the original dataset.

since the last earthquake. Then, the 5674 first elements were retained and 30 seconds of seismic recordings were downloaded from the time exactly in the middle of them. This defines normal behaviour as when no earthquake is happening in at least the next 8590 seconds, roughly 71 minutes. This way, our dataset was balanced and normal behaviour should have very little to no information indicating an earthquake would happen after H seconds. Again, all wave recordings were normalized.

**Downsampling**

Considering time and computational resources, one technique to speed up training is downsampling the wave recordings. The original data provided by FDSN [11] is recorded at a frequency of 100 Hz. To find a reasonable downsampling rate, capturing the essential features but still conveying enough information, a small dataset was tested with different sampling rates of 2, 5, 10, 25, 50 and 100Hz, with H=0, H=15 and H=30. Table 1 shows the accuracies for the dataset at each rate, based on which we decided to sample down to 25Hz in the final experiment.

| *Freq.* | H=0 | H=15 | H=30 |
|---|---|---|---|
| 2Hz | 0.76 | 0.76 | 0.72 |
| 5Hz | 0.79 | 0.75 | 0.75 |
| 10Hz | 0.77 | 0.76 | 0.75 |
| 25Hz | 0.80 | 0.77 | 0.76 |
| 50Hz | 0.80 | 0.77 | 0.76 |
| 100Hz | 0.79 | 0.78 | 0.76 |

Table 1

## 3.2 Overfitting

When a deep learning data becomes too specific to its training data, we say it is overfitting. Too much overfitting leads to the model being good at predicting the training set, but not on other, unseen data. To prevent our model from overfitting, we looked at multiple methods, namely drop-out, batch normalization and early stop. Drop-out and batch normalization showed to be useful and were implemented in our LSTM model. To show their impact, a small dataset was run with H=0, T=30; first with nothing added to avoid overfitting, then once each with drop-out and batch normalization. Without any overfitting strategy, the model was clearly overfitting, with the training accuracy still increasing up to 0.96 while the validation accuracy slightly decreased until 0.76

**Dropout**

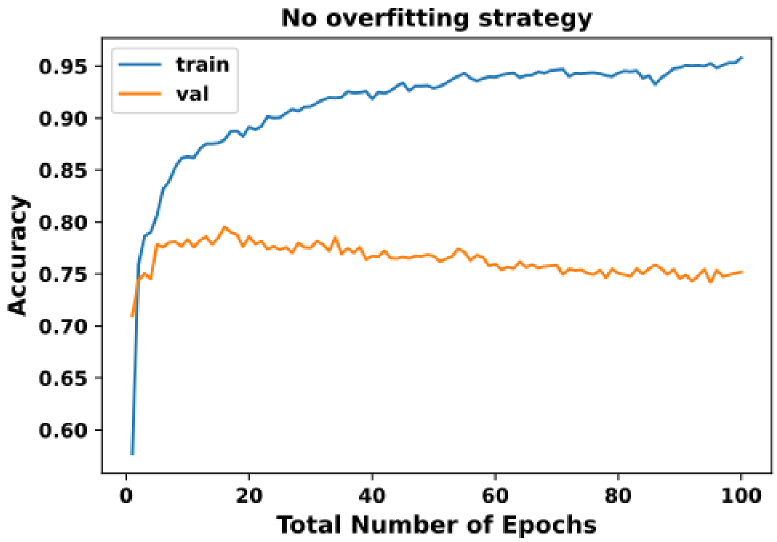
This technique randomly ignores certain neurons in a layer, such that the output predictions do not rely on specific neurons. This way, the model can generalize better. In our implementation, dropout was used in the LSTM layer with a probability of 0.2. As shown in figure 6, this diminishes overfitting considerably, as we can observe that the validation accuracy does not significantly decrease anymore and reaches 0.79 (compared to 0.76).

### Batch normalization

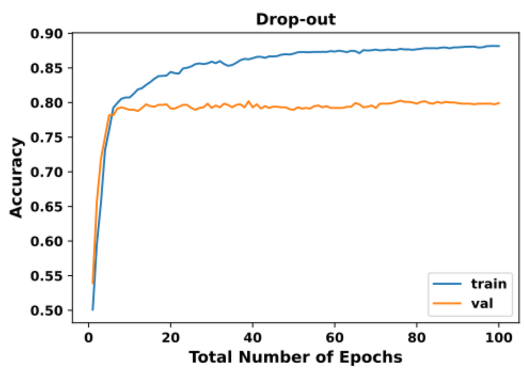
Batch normalization standardizes the input of the layer for each batch. By normalizing all features per batch, the LSTM model gets a more stable accuracy of 0.77, but most importantly the curves show less overfitting and the validation accuracy does not seem to decrease.

### Early-stop

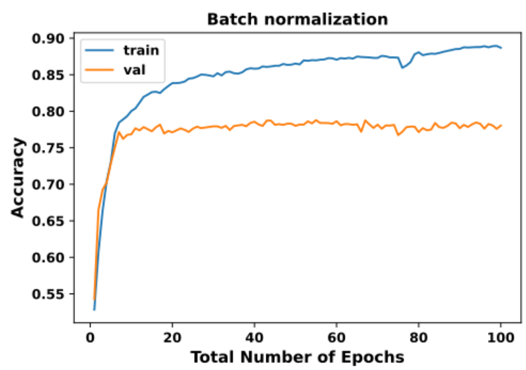
When overfitting, over the number of epochs ran the model will first have an ascending (learning) curve. Up to a certain point, where the model will continue improving on the training data but not on the validation data. Therefore, one way to prevent overfitting is prompting our training loop to stop when reaching that point. However, in our LSTM model the errors made were highly varying from one epoch to the next, and the model did not deteriorate that much in generalizing. This is shown for example in figure 7b, section 4. A few early-stop rules were tested out, but they did not seem effective or impactful. Early-stop was not used in the end.




(a)




(b)




(c)

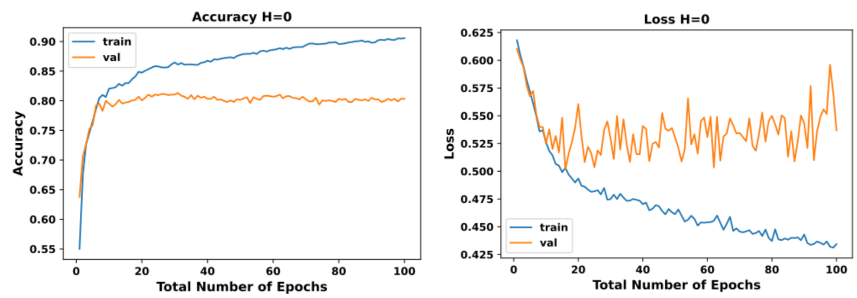
Figure 6: Accuracies of a small dataset with (a) no strategy to avoid overfitting, (b) drop-out and (c) batch normalization. H=0, T=30

### 3.3 Set up

The final experiments were run on the above described dataset, consisting of about 11 thousand instances of 36 30-seconds long seismic wave-recordings as input, half of which were earthquakes and half not. The batch size was 50 and the model was run for 100 epochs with a learning rate of 0.001. The training, testing and validation sets were respectively 60, 20 and 20 percent of the whole dataset. The model had one LSTM layer and a hidden size of two.

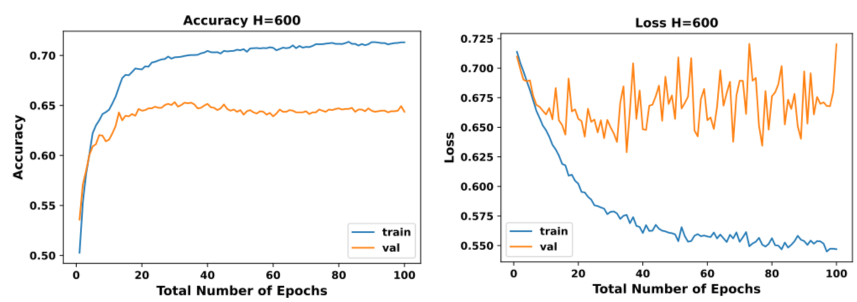
To bring insight to the research question, and with regards to the problem modelling described in section 2.1, the LSTM model had to run on datasets with the same earthquakes with parameter H (time before strike) varying. L (length of recorded sequence) was fixed to 30 seconds. A first experiment was run with H going from 0 to 60 seconds by steps of 5 seconds. A second experiment let H vary from 0 to 600 seconds by steps of 60 seconds. To even out the effect of the random test and validation set allocation, for every H the model was run 5 times and then averaged.

# 4 Results



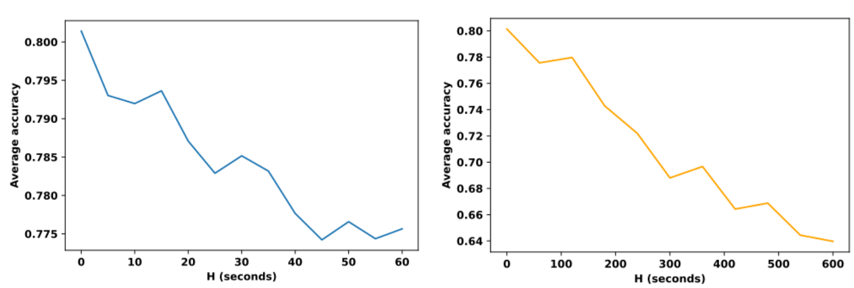

(a) Accuracy (b) Loss

Figure 7: Accuracy and loss plots for H=60s




(a) Accuracy (b) Loss

Figure 8: Accuracy and loss plots for H=600s




(a) H from 0 to 60s (b) H from 0 to 600s

Figure 9: Average accuracies against H varying

The prediction accuracy for this dataset went over 80 percent for H=0s (averaged 0.801). This confirms the ability of LSTM to predict earthquakes given the preceding seismic recordings. At first, the model was run with H varying from 0 to 60 seconds, with 5 seconds steps. This already showed that time before strike had an effect on the accuracy, which gradually dropped to 0.776 with H=60s. Since the accuracy stayed pretty high at 60s, the experiment was run again with H varying from 0 to 600 seconds, with 60 seconds steps. Accuracies continued to drop; with H=600s, the average accuracy was 0.639.

Clearly, we can say that seismic movement retains warning features in at least the 10 minutes preceding a low-magnitude earthquake. If we extrapolate the curve shown in figure 9b, we could argue that even 15 minutes before earthquake can be of interest. To test this, the dataset was run once with H=900s. This yielded an accuracy of 0.573. The dataset was then again run, this time with H=3600s. The validation accuracy should have been close to 0.5, but it showed to be higher than expected, namely 0.554. This indicates that there still was some bias in the dataset.

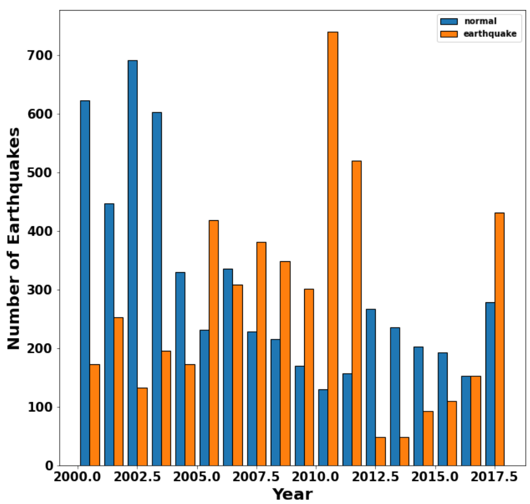



Figure 10: Distribution of the number of samples per year in the dataset. Blue bins are normal behavior, orange bins are earthquake predecessors.

After examination, one answer found was that the normal and earthquake waves did not have the same distribution over the years figure 10. Some changes in any station could thus influence the results, with the LSTM model learning which data was from before or after a change. This could be, for example, recording material deteriorating.

Knowing this, the accuracy gotten from H=900s is not that high. We can thus conclude that useful information to predict low-magnitude earthquakes lies within 10 to 15 minutes before strike.

# 5 Conclusion & Discussion

Firstly, the results presented in the previous section clearly indicate that LSTM can be used with seismic recordings as features to predict the happening of low-magnitude earthquakes. But most importantly, it demonstrates the relation between the time before strike and the accuracy of such a model. The results suggest that seismic recordings preceding a low-magnitude earthquake contain sensitive information in the 10 to 15 minutes preceding the earthquake.

However, these numbers must be considered carefully. They are an indication of what can be used, but not a representation of how accurately we could really predict earthquakes. One limitation is the problem modelling that was used. In order to clearly study only the effect of the time before strike, the model used was very simple and a binary classifier. In each run of the experiment, earthquakes would either happen exactly H seconds after the recordings, or not at all. The reality of the world is more complex. One would not want a program that can only predict if an earthquake does happen after 10 minutes for example, but rather a program that can forecast probabilities of an earthquake happening with time estimations.

Having shown the effect of the time of recording before strike, future work should look deeper into how we can try to predict earthquakes happening after any H seconds. Since earthquakes are a rare event but with big consequences, it is important to find ways to minimize fake alerts. We could ask research questions such as ”Given a continuous stream of seismic recordings, how can we accurately warn for imminent earthquakes with very high probability?”

In a more global way, research around this subject still has a long way to go. This paper does not guarantee in any way that LSTM is the best approach. Different models should also be tested. Also, this research does not look at predicting the location of an earthquake, which opens the way for a whole lot of other questions.

# 6 Responsible Research

One important aspect of research is the reproducibility of the research methods. As such, the methodology and experiment sections describe the data used precisely, such that one could reproduce the experiments. Of course, no two models will produce the exact same results, but the source code of the model used in this paper can be provided upon request.

Another point of attention that is specific to research using machine learning methods, is the bias in the model and/or data. This has been adressed when discussing the results, and it is important to keep in mind that the data might not always be perfect.

# References


[1] Earthquake. https://en.wikipedia.org/wiki/Earthquake. Accessed: April 28th 2022.

[2] Earthquake environmental effects. https://en.wikipedia.org/wiki/Earthquake_environmental_effects. Accessed: April 28th 2022.

[3] Earthquake early warning - fine-tuning for best alerts. https://www.usgs.gov/natural-hazards/earthquake-hazards/science/earthquake-earlywarning-fine-tuning-best-alerts. Accessed: May 5th 2022.

[4] G. Chouliaras. Seismicity anomalies prior to 8 june 2008, mw=6.4 earthquake in western greece. *Nat. Hazards Earth Syst. Sci.*, 9, 2009.

[5] Mazzola G. Isufi, E. Graph-time convolutional neural networks. *2021 IEEE Data Science and Learning Workshop (DSLW)*, 2021.

[6] Park J. Athens N. Ibrahim, M. A. Earthquake warning system: Detecting earthquake precursor signals using deep neural networks. 2018.

[7] Mohsen Y. et al. Spatiotemporally explicit earthquake prediction using deep neural network. *Soil Dynamics and Earthquake Engineering*, 144, 2021.

[8] Q. Wang et al. Earthquake prediction based on spatio-temporal data mining: an lstm network approach. *IEEE Transactions on Emerging Topics in Computing*, 8:148–258, 2020.

[9] S. Pasari A. Bhatia and A. Mehta. Earthquake forecasting using artificial neural networks. *International Archives of the Photogrammetry*, 2018.

[10] Hashmi I. et al. How does a cnn mixed with lstm methods compare with the individual one in predicting earthquakes? 2022.

[11] Fdsn webservice. https://www.geonet.org.nz/data/tools/FDSN. Accessed: May 6th 2022.

[12] Seismometers, seismographs, seismograms - what's the difference? how do they work? https://www.usgs.gov/faqs/seismometers-seismographs-seismograms-whats-difference-how-do-they-work. Accessed: May 28th 2022.

[13] Pytorch. https://pytorch.org/. Accessed: May 28th 2022.