Predicting Micro-Earthquakes with Deep Neural Networks

Kevin Zhu
Supervisor: Mohammad Sabbaqi
Responsible Professor: Elvin Isufi
EEMCS, Delft University of Technology, The Netherlands

May 6, 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

Earthquake prediction is the field of seismology concerned with predicting the time, location, and magnitude of earthquakes within a small time frame, usually defined in terms of minutes or seconds before an event. Such predictions can have a large impact on minimizing the damage caused by these seismic events, by providing early warnings to the affected population and allowing them to respond appropriately. Although the methods used to predict earthquakes are often limited to larger magnitude events, predicting micro-earthquakes is also an important task, especially in locations that are more vulnerable to seismic shocks. Concentrations of localized micro-earthquakes can also hint at larger future seismic events, and their location can be used to locate moving fault lines underground. Deep learning methods perform particularly well in this context, due to their ability to quickly identify patterns in large volumes of data, and Long-Short Term Memory (LSTM) neural networks are very well suited to handling time-sequenced data such as the seismic waves used in earthquake prediction problems. In this paper, an LSTM network is trained to predict micro-earthquakes three seconds before the event, using seismic recordings from the New Zealand dataset: the goal is to find the optimal size of these recordings and understand how different values affect the model. Our results suggest that larger recordings do not provide any benefit in performance and that high levels of accuracy can be reached with smaller samples. This means that in the context of micro-earthquake prediction, primary waves can be easily detected in short recordings, and are likely to travel close to shear waves. This can be attributed to the low strength of signals that micro-earthquakes generate, as they will travel shorter distances than major earthquakes.

1 Introduction

Earthquakes are seismic events that cause shaking on the surface of the earth and generate seismic waves [1]. Although they differ greatly in magnitude, without the proper countermeasures even minor earthquakes can prove to be devastating, causing damage to buildings, infrastructure, and whole communities [2]. The destruction also comes from secondary sources, named "Earthquake environmental effects", and include tsunamis, landslides, soil liquefaction, and more [3].

The goal of earthquake prediction is to provide warnings of potentially damaging earthquakes to the affected population, allowing them to respond appropriately and minimize losses to life and property [4], [5]. However, such warnings are often limited to large seismic events, due to their bigger impact on society. The prediction of micro-earthquakes is on the other hand a relatively minor field, often limited to localized geological surveys instead of disaster prevention [6]. Nevertheless, predicting micro-earthquakes is an important task, especially in locations that are more vulnerable to seismic shocks such as hospitals, research laboratories, historical buildings, and sites prone to landslides. Furthermore, according to [7], concentrations of micro-earthquakes can hint at larger seismic events and volcanic activity, and since they occur on the same fault lines as major earthquakes, they can provide more data to locate fault structures underground [8], [9].

The problem of earthquake prediction is conceptually straightforward, but the increasingly large volumes of data to analyze, paired with the short time frame of computation allowed, drastically increases the complexity of the task. Deep learning methods perform particularly well in this context, thanks to their ability to quickly identify patterns in multi-dimensional data, and have a clear advantage compared to more traditional methods [10]. Long-Short Term Memory (LSTM) based networks in specific are very well suited to han-

dling time sequenced data such as the seismic waves used in earthquake prediction problems [11], [12].

The focus of this research was to train an LSTM neural network to predict micro-earthquakes, using recordings of seismic waves gathered from the New Zealand earthquake dataset [13]. The research question is *What is the optimal size of recordings for predicting micro-earthquakes?*. To answer this question, we want to understand how the size of training samples affects the performance of the model. While larger sizes can provide more information to the network, they also increase the amount of data that needs to be stored, processed, and analyzed. For this reason, smaller viable samples provide more benefits from a logistic point of view.

Related works

Deep learning methods have often been used in the field of earthquake prediction, and are known to have clear advantages over traditional methods, as shown in [10]. To select which model to use for our experiments, we looked at [14] and [15], where they tested the performance of different architectures on the New Zealand earthquake dataset: according to their results, LSTM networks manage to outperform models such as convolutional neural networks (CNN), vanilla recurrent neural networks (RNN) and mixed CNN-LSTM architectures on unseen data. LSTM networks have also been successfully used to predict micro-earthquakes with high performance, as shown in [16], where they trained the model on 1 million earthquakes below 2.5 magnitude. For this reason, we selected LSTM as our model of choice.

2 Methodology

In this section we will describe how we created the dataset used in the experiment, the architecture of the deep learning model, and the performance metrics used to evaluate the final results.

2.1 Dataset

The New Zealand earthquake dataset contains a wide variety of information related to seismological events. These include data about all recorded earthquakes, as well as wave recordings gathered by stations spread out across the country. Each recording station constantly records nearby seismic waves across different channels and frequencies, in the form of continuous, temporally sequenced data. For our purposes, it was necessary to understand how to correctly gather the required data from the New Zealand servers and process it into the desired format to use as input for our model.

2.1.1 Geographical distribution of earthquakes

The first step was to gather information about seismic events that happened within the New Zealand territory. To do so we restricted the geographical area from which we would retrieve data by defining a bounding box delimited by the coordinates shown in Table 1. In Figure 3 we can see the area delimited by the bounding box, as well as the distribution of earthquakes and recording stations within its boundaries.

	Latitude	Longitude
Lower bound	-47.749	166.104
Upper bound	-33.779	178.990

Table 1: Coordinates of the bounding box that delimits the geographical area of New Zealand.

2.1.2 Filtering earthquakes

We then retrieved a list of all earthquakes within the bounding box from the New Zealand dataset, together with information about the epicenter, magnitude, depth and time of the events. We ended up with a list of around 266k events, and their magnitude distribution can be seen in Figure 1. From this list, we had to retrieve the micro-earthquakes, which are usually defined as seismic events below 2.0 magnitude [17]. Most micro-earthquake networks though, use a higher limit to account for error: this includes [16], where they used earthquakes below 2.5 magnitude to train their LSTM model. As such, we defined our upper threshold as 2.5. Earthquakes with magnitude below 0.5 were also excluded from the dataset since the seismic energy released from such events were deemed to be too low to be distinguishable from calm periods (energy equal to or lower than a hand grenade according to [18]).

After filtering out all the earthquakes that did not lie between these two thresholds, we ended up with 173k events. The magnitude distribution of these events can be seen in Figure 2, while their locations can be seen in Figure 4. The average depth of these earthquakes was 32km below ground with a 75th percentile of 33km, which means that the majority of micro-earthquakes were relatively shallow.

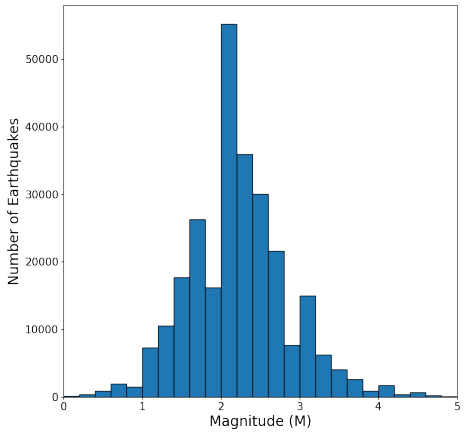

Figure 1: Magnitude distribution of all recorded earthquakes retrieved from the dataset.

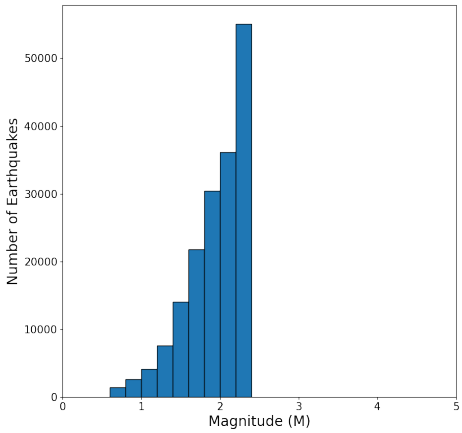

Figure 2: Magnitude distribution of micro-earthquakes in the dataset (between 0.5 and 2.5 magnitude).

Our objective was to train the network to predict micro-earthquakes, and we decided to not include higher magnitude earthquakes in the dataset. The reason for this was that

introducing high magnitude events that are labelled as false values would directly increase the level of complexity of the classification problem, which was not necessary to answer the research question. The model did not need to know how to distinguish micro-earthquakes from high-magnitude earthquakes.

2.1.3 Filtering stations

Next we retrieved data of recording stations active during the 2007-2016 period from the dataset: we ended up with a list of 90 different stations, whose locations are shown in Figure 3. From this figure we can see that a lot of stations are closely clustered on the northern side of the country, making their information redundant for our research. To train the neural network, we also needed the seismic wave recordings from each station in the list for every event, which would result in an unnecessarily large dataset. For this reason, we decided to narrow down the number of stations.

To do so, we filtered them based on the quality of their recordings: we first randomly selected 2000 earthquakes from the dataset, and downloaded recordings from each station for each earthquake, from 60 seconds before the event up until the event itself. We then removed all stations whose recordings contained corrupted data (in the form of NaNs or improbably high/low scalar values) as well as recordings that were shorter than the expected length of 60 seconds. After ensuring that the remaining stations were somewhat evenly distributed across the New Zealand territory we ended up with 38 stations (whose locations are shown in Figure 4). This last step was done to ensure that seismic waves from any location in the country could be picked up by at least one station.

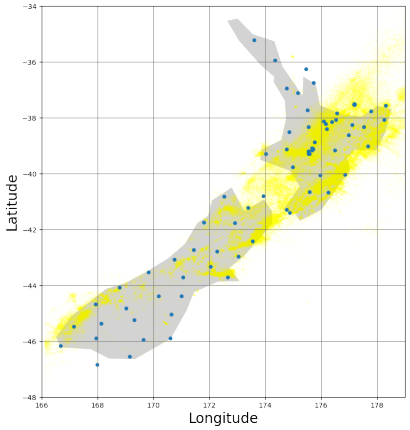

Figure 3: Geographical area delimited by the bounding box and the distribution of all 90 recording stations (blue) and recorded earthquakes (yellow) active between the years 2007-2016 retrieved from the New Zealand dataset.

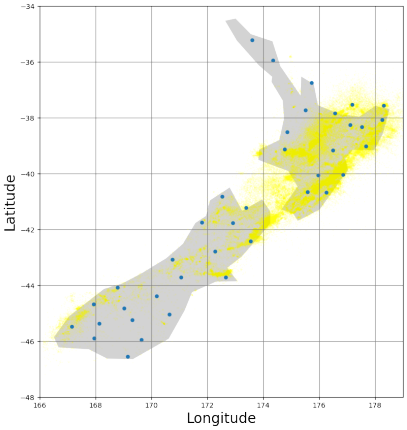

Figure 4: Distribution of the 38 filtered recording stations (blue) and micro-earthquakes (yellow) from the New Zealand dataset.

2.1.4 Gathering calm period recordings

To train the neural network, we also needed recordings of periods without any seismic event, which we called calm periods. These are used as the negative values in our binary classification problem. To retrieve these calm periods, we first sorted the earthquakes in our dataset chronologically and then filtered out all the events that were within 5000 seconds (83 minutes) of each other. According to [19], most earthquakes have a very short duration, ranging from a few seconds up to 30-40 seconds for major earthquakes. The largest earthquake ever recorded, with a magnitude of 9.5, lasted around 10 minutes. As such, the threshold of 5000 seconds was chosen to ensure that recordings of calm periods taken between two consecutive events from the list (around 40 minutes before and after any recorded earthquake) would not contain any seismic waves from an earthquake, with a sizeable margin of error taken into account.

2.1.5 Seismic waves sampling

Recordings of seismic waves were retrieved from the New Zealand dataset in the form of MiniSEED files [20]. These files contain fixed-length records of contiguous time-stamped values, with a length equal to the length of the recording (in seconds) times the frequency (frames per second). While stations in New Zealand record waves in multiple channels and frequencies, we specifically used the 'HHZ' channel for our research [21]. The letters in the acronym 'HHZ' stand for:

- H - High broad band: recording with a frequency between 80-250 HZ.
- H - High gain seismometer: recordings with higher amplitude capacity, although possibly noisier.
- Z - Vertical waves.

The reason why we selected vertical waves is to detect the primary waves (P waves) of earthquakes in the recordings: acting similarly to sound waves, P waves can travel through any type of material very quickly, producing very low damage, and reaching far distances without weakening [22]. This allows them to retain their frequencies when arriving at recording stations, and since epicenters of earthquakes are located deep underground, these waves end up travelling upwards creating a strong vertical movement on the surface. What causes damage during earthquakes are instead the shear waves (S waves), which travel at around half the speed of P waves.

After downloading the recordings, we used the same process as the one used in section 2.1.3 to remove any corrupted sample from the input dataset.

2.2 Deep-learning model

The neural network that was used for the research is a Long-Short term memory (LSTM) neural network. It was used as a binary classifier, returning a value of 1 if the seismic wave recordings used as input preceded an earthquake within the next three seconds, and returning 0 otherwise. LSTMs are models derived from recurrent neural networks (RNN) and their architecture performs particularly well with temporally sequenced data such as the recordings in this dataset.

The model used in the experiment is composed of six main layers, and its architecture can be seen in Figure 5. The first layer is a single cell LSTM (Figure 6), with two features

and the length of the MiniSEED file as input size. The LSTM connects to a Feed-Forward layer composed of alternating ReLu activation functions (Rectified Linear Unit [23]) and two Fully Connected layers. The first layer receives 2 input features and outputs 128 features, while the second layer receives 128 input features and outputs 1 feature [24]. Each fully connected layer is also preceded by a Dropout layer, as described in Section 3.4. Finally, the output is passed through a Sigmoid as the activation function for the binary classification [25].

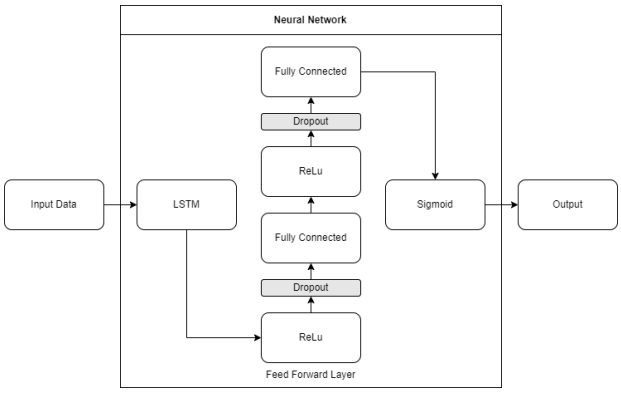

Figure 5: Architecture of the binary classifier neural network model.

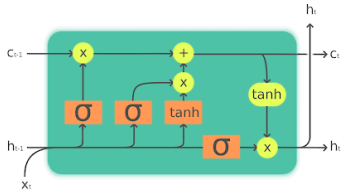

Figure 6: Architecture of a single LSTM cell [26]

2.3 Performance metrics

To evaluate the performance of a binary classifier, we decided to use metrics based on a confusion matrix, which is composed of four scores: True Positives (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN) [27]. Given the context of earthquake predictions, and the severity of damages that a missed warning for an earthquake can cause, we considered false negatives to be the main score to optimize. On the other hand, ignoring the number of false positives could lead to the loss of trust in the warning system by the population. As such, we included two evaluation metrics that aim to optimize false positives and false negatives respectively, with a larger emphasis on the latter.

The first metric is the Positive Predictive Value (PPV, also known as Precision), which refers to the percentage of all positive predictions that are correct (Formula 1) [28], while the second metric is the True Positive Rate (TPR, also knows as recall), which refers to the percentage of all positives in the dataset that were correctly classified (Formula 2) [28]. These two scores are demonstrated in the form of PR curves (Precision-Recall curves), graphs that plot precision on the Y-axis and recall on the X-axis, to illustrate their relationship [29].

$$Precision = \frac{TP}{TP + FP} \quad (1) \qquad Recall = \frac{TP}{TP + FN} \quad (2)$$

To evaluate the overall performance of the model we use Accuracy, shown in Formula 3, and is defined as the percentage of correct predictions over the total number of predictions. Although this metric does not take into account the distribution of different classes in the data, the dataset used for this research is evenly distributed.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

In summary, we use accuracy as the main evaluation metric when comparing different sets of parameters in the experiment. We define the best parameters by looking at the average accuracy, the variance, and the size of the samples with a preference for smaller samples. We then verify the precision and recall values of the optimal parameters by looking at the PR curves of the model.

3 Implementation

In this section we will describe the experimental settings used to reach our final results, as well as the tools and methods used to optimize the parameters of the model.

3.1 Experimental settings

The final input dataset was composed of a total of 10'000 events, of which 50% were micro-earthquakes and 50% were calm periods. This dataset was split into three sets: 60% training, 20% validation and 20% testing. The sets were shuffled and stratified according to their label, ensuring an even distribution of classes which was required for a reliable accuracy value. Some important hyper-parameters to define for the experiment are:

- Length of the recording T (seconds): How many seconds of data we provide the model to obtain the prediction. To answer the research question, we want to understand how this parameter affects the overall performance of the network. Recordings that are too short might not provide enough information, while longer recordings can instead introduce noise to the algorithm. Shorter values are preferred though, as they would result in fewer data required to be stored and analyzed by a fully trained model. The maximum value of T in the experiment was set to 55 seconds.
- Time before the event H (seconds): How far into the future the model can predict the earthquake. For our research question, optimizing this parameter was not considered relevant, and would instead unnecessarily increase the complexity of the task. As such, we used a fixed value of 3 seconds for all our computations.
- Frequency HZ: The number of frames for each second in the recording. This parameter is very closely related to the value T, as such, they needed to be optimized together. Similarly to the value of T, recordings with high frequencies can be noisy and result in an overfitting model, while recordings with low frequencies could lose important features in the data and result in an underfitting model. They also directly affect the size of the dataset, and as such, given a similar level of performance, lower values are preferred. The original sampling rate of the recordings in the dataset is 100HZ.

Experiments were run with the following hyper-parameters: training loops of 1000 epochs, with a batch size of 100 and a learning rate of 0.0001. The values to optimize were T and HZ. Optimizing other hyper-parameters was not considered a priority, since achieving the highest accuracy possible was not the goal of the research. The format of a single input sample is a 2D-array with the shape of $(38, (T * HZ))$, with 38 being the number of stations. For the training process, we decided to use Binary Cross Entropy as the loss criterion (BCELoss [30]), and Adam as the loss optimizer.

3.2 Data pre-processing

During the experiments, the seismic wave recordings in the dataset were pre-processed with different methods to analyze how they affect the performance of the model. These include Scaling (performed by removing the mean from the input vectors and scaling them to unit variance) and Normalizing (performed by scaling the input vectors individually to unit norm). The norm used for normalization was 'l2'. The performance of these methods was tested on a default set of parameters: T value of 30, H value of 3 and a frequency of 50HZ. In Appendix A, we can see a comparison of the validation accuracy of the different pre-processing methods, as well as their PR curves. We decided to use the scaled dataset for the final experiments, due to its higher performance.

3.3 Down-sampling

In digital signal processing, down-sampling is the process in which the band-width of a wave is reduced [31]. This process is done by first reducing the high-frequency signal components with a digital low-pass filter, and then decimating the filtered signal by N (which means keeping only every Nth sample of the signal). The low-pass filter is usually needed to prevent aliasing: a phenomenon that causes different waves to result in the same down-sampled wave by mapping different frequencies to the same sampling points [32].

While multiple types of low-pass filters have been attempted during the down-sampling process, they increased the computation time of the model by a factor of 20+, without any noticeable difference. This was further proven in Section 4.3, where we can see from the shape of the waveforms in the scaled dataset that aliasing would not distort the data. As such, we decided to skip the low-pass filter step and directly decimated the signals to the desired frequency.

3.4 Overfitting solutions

The training process of the model also included methods used to prevent overfitting, namely regularization, early stopping and dropout layers.

Regularisation

This is a method used to reduce the complexity of the weights in the neural network, thus allowing the model to generalize better instead of overfitting [33]. This is done by adding a penalty, which in our case is L2 regularization, to the loss function. Although L2 is often synonym with weight decay in loss optimizers, according to [34] and [35], this is only the case when using Stochastic Gradient Descent. Since our model uses Adam as the loss optimizer, we opted instead for a variant named AdamW, with built-in L2 regularization that is decoupled from weight decay [36].

Dropout

Dropout layers are used to thin out a neural network by randomly resetting neurons during the training phase, which according to [37] can help reduce overfitting by averaging the weights of the whole network. In our model, we added dropout layers before each Fully Connected layer, with a probability value of 0.1 (the probability of resetting a randomly selected neuron every epoch). Since the model ended up not overfitting, we did not increase this value any further.

Early stopping

Early stopping is used to prevent a neural network from overfitting on the training data by halting the training process once the performance on the validation set starts to degrade [38]. We implemented this solution in the code with a patience value of 10, which means that the training loop halts after 10 consecutive epochs in which the loss value has increased over its previous epoch.

4 Results

In this study, we wanted to find out the optimal size of recordings for predicting micro-earthquakes, and how this value affects the performance of the model. In practice, the size of a sample is composed of two separate parameters: the length of the recording in terms of seconds T, and the frequency HZ.

4.1 Grid-search

To answer the research question, we conducted a grid-search with different values of T and HZ over 10 runs each, and the results can be seen in Figure 7. The data suggest that the size of the input does not affect the maximum accuracy that a model can reach, since most sets of parameters are capable of reaching above 0.85 accuracy. The smallest input size of 'T10 - HZ10' (100 data-points) has a slightly lower maximum of 0.81, and while this could be caused by the lack of features in the data, it could also be due to variance in the grid-search. From this, we can deduce that larger input sizes are not strictly better than smaller samples, which goes against the initial hypothesis that more data equals more useful information for the model. Some of the better performing sets of parameters, such as 'T10 - HZ25' (250 data-points) and 'T50 - HZ10' (500 data-points), are much smaller than the default parameters of 'T30 - HZ50' (1500 data-points) that were used in past experiments [14], [15].

What we believe is happening is that P and S waves generated by micro-earthquakes, due to their low magnitude and seismic impact, are travelling a much shorter distance than major earthquakes. As a consequence, only events that are close to recording stations may be detected. Due to the shorter distance from the epicenter, P and S waves are more likely to reach the recording stations close to each other, since P waves do not have the opportunity to gain distance. As a result, long recordings are not necessary for the model to detect the initial shock of vertical P waves, as the S waves will follow soon after. This hypothesis also aligns with the magnitude distribution of earthquakes that we saw in Figure 1: as the magnitude of earthquakes lowers, so does the amount of earthquakes detected, which would normally not be the case since micro-earthquakes are much more frequent.

What we could not find from the results, however, is a clear relationship between the frequency of the samples and the performance of the model: samples at 75HZ, for example, seem to perform more consistently than other frequencies, but according to our findings in Section 4.3, the sampling rate on a scaled dataset should not have such a direct impact on the performance, due to the shape of the waveforms. As for the difference in standard deviation between the different sets of parameters, we explore this issue further in the next section.

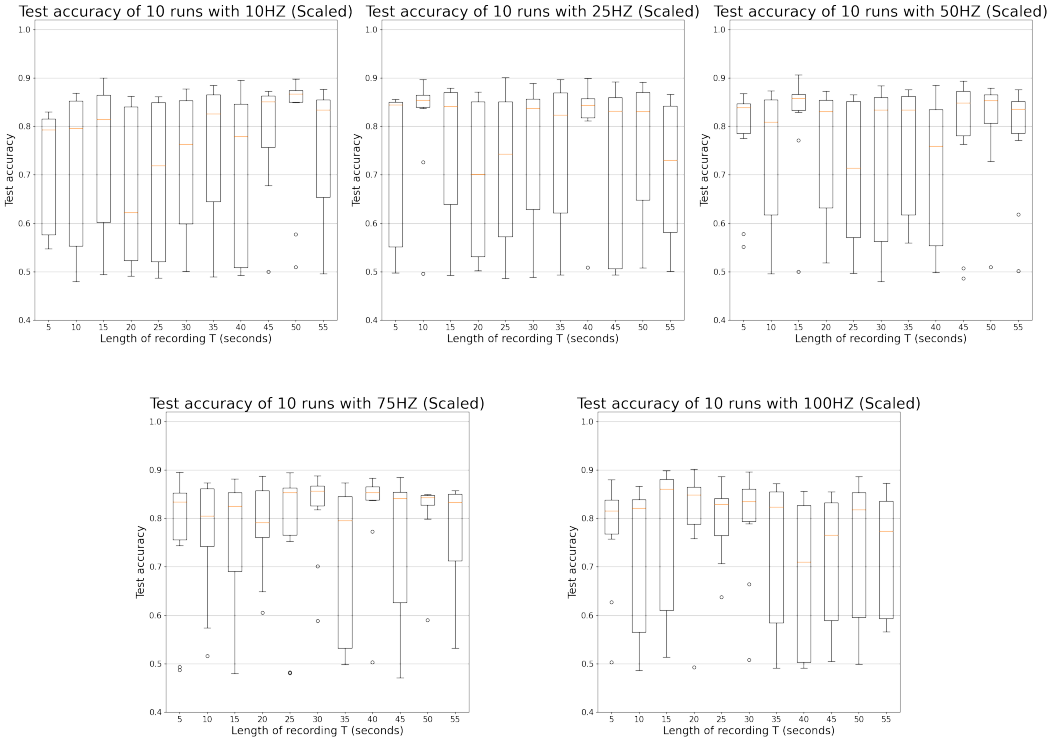

Figure 7: Boxplots of the test accuracy of the model on a Scaled dataset, with different values of T and HZ, over 10 runs.

4.2 Analysis of the best performing parameters

To understand the reason behind the large variance seen in previous results, we decided to analyze the validation accuracy, validation loss and PR curve of one set of parameters over 10 runs, namely 'T10 - HZ25'. We can see from Figures 8 and 9 that the learning curves have unusual shapes, and the model is underfitting. The slow progression in the first few hundred epochs implies a low learning rate, but most runs are characterized by a steep drop in the loss value at one point or another, followed again by a slow but steady convergence. A hypothesis is that the structure of the data makes it very likely for the model to be stuck in a very steep saddle point and that the presence of multiple local minima is what produces the high variance in the data. These points seem to indicate that the learning rate is indeed too low, even with the adaptive learning rate of the Adam optimizer.

In Figure 10 we see the PR curve of the better performing run in the set. If we wanted the model to optimize for the recall value, we can see from the curve that it is capable of approaching 1.0 while still maintaining a precision value above 0.75. This means that with proper hyper-parameter tuning, the model is capable of predicting 0 false negatives on the dataset even with small samples of data. In Table 2, we see the evaluation scores for the same run when optimizing for accuracy instead, which still results in a recall value of 0.974.

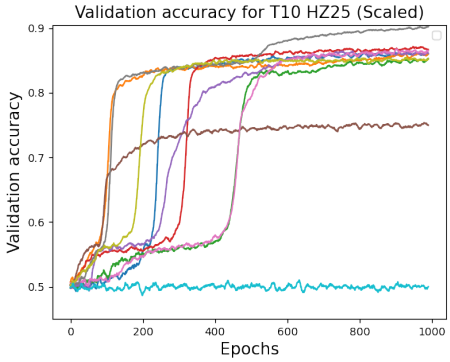

Figure 8: Validation accuracy over 10 runs on a scaled dataset: T of 10, H of 3 and 25HZ.

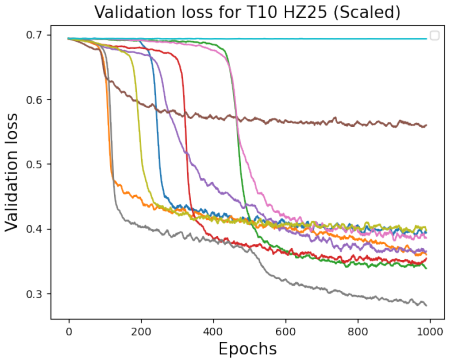

Figure 9: Validation loss over 10 runs on a scaled dataset: T of 10, H of 3 and 25HZ.

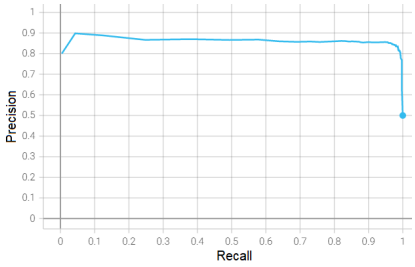

Figure 10: PR curve of the run with highest accuracy with the following parameters: T of 10, H of 3 and 25HZ.

Threshold	Accuracy	Precision	Recall	TP	FP	TN	FN
0.50	0.913	0.869	0.974	995	150	872	27

Table 2: Accuracy, precision, recall and confusion matrix of the run with highest accuracy with the following parameters: T of 10, H of 3 and 25HZ.

4.3 Waveforms analysis

To better understand how earthquake prediction works, we decided to analyze the waveforms of different samples. When looking at the normalized samples in Figure 11, we can find a clear difference between recordings of calm periods and before earthquakes: in the right figure, the high amplitude and level of oscillation of the green wave, recorded from a station that was closer to the epicenter of the earthquake, is very distinct, especially when compared to the raw samples in Appendix B. We can also see how the signal of the green wave attenuates as the time approaches the time of the earthquake, indicating that the vertical oscillation was caused by P waves that reached the station before the more damaging S waves.

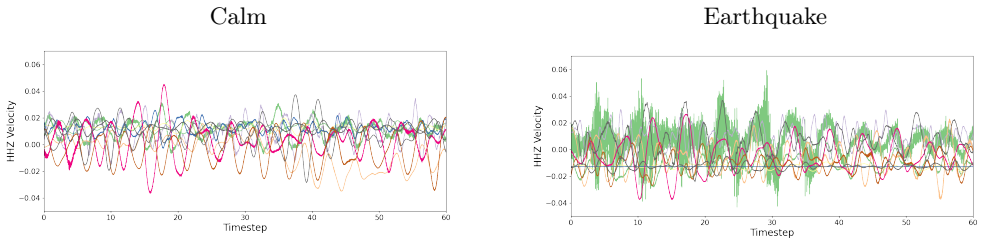

Figure 11: Normalized Dataset. Recordings of 60 seconds from five different stations

The scaled samples in Figure 12, on the other hand, are less intuitive to decipher: most waves became flattened at a value between 0.12 and 0.14, due to their low level of oscillation. An exception is the recording that captured the P wave, which sits at a value of -7.95. While the scaled samples seem to contain less information compared to the normalized samples, according to the results in Appendix A, the model performs similarly to the scaled dataset, possibly since P waves in the data can still be detected.

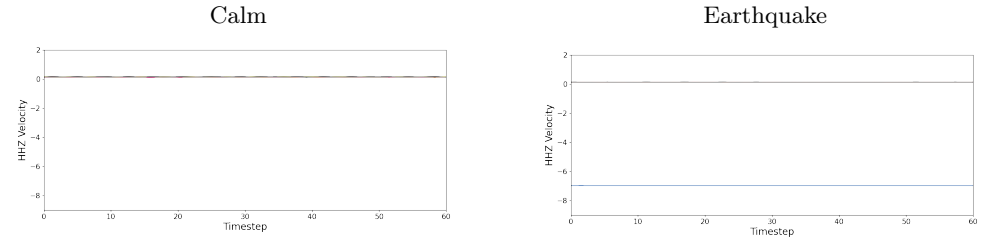

Figure 12: Scaled Dataset. Recordings of 60 seconds from five different stations

5 Responsible research

During the research, the experiments were conducted with the objective to be reproducible and repeatable. All steps taken to gather and process the dataset have been documented in Section 2.1, while the model architecture and experimental settings are described in Sections 2.2 and 3.1. The results in Section 4.1 were acquired through multiple runs to account for variance, although the number of runs was limited by the computation resources available. Provided that the same steps have been taken to recreate the dataset and the same parameters used for the model, results should be similar, accounting for variance.

The source data used for the experiment is open-source and freely available. We acknowledge the New Zealand GeoNet project and its sponsors EQC, GNS Science, LINZ, NEMA, and MBIE for providing data used in this study [39]. The source code of the model used in the experiment is available at `https://github.com/neah1/earthquake_rnn` and can be freely used to reproduce the results presented in this study.

6 Conclusion and Future work

In this paper, we trained a Long-Short term memory neural network to predict micro-earthquakes, using data from the New Zealand earthquake dataset. We wanted to find out the optimal size of recordings for the task, and how this value affects the performance of the model. To do so, we ran a grid-search over different lengths and frequencies of the samples, using recordings up to three seconds before the event. What we found out from the results, is that larger recordings do not seem to provide any benefit in performance to the model. On the contrary, high levels of accuracy were reached from all sizes of samples, although with different levels of consistency. The issue with the variance seems to stem from a low learning rate. Regardless, we can gather from this that in the context of micro-earthquake prediction, short recordings at low frequencies can be used with high accuracy and recall values. This seems to be attributed to the weak signal that micro-earthquakes generate, resulting in primary and shear waves reaching recording stations close to each other.

In summary, the prediction of micro-earthquakes seem to have lower requirements in term of the amount of data to store and analyze, both in term of length and sampling rate, compared to larger magnitude earthquakes. On the other hand, they also require wider networks of stations to cover up for their weak signals, which in turn makes their use strictly localized to areas more vulnerable to seismic damage.

Future work will consider exploring how far in the future it is possible to predict these micro-earthquakes, while still maintaining short sample sizes. Doing so can give insight into how far their seismic waves can travel and their behavior. Another direction would be to train neural networks to directly locate the epicenter of micro-earthquakes based on recordings, which can help map out the coverage of existing recording stations.

References

[1] "Earthquake," Feb. 2022. [Online]. Available: https://en.wikipedia.org/wiki/Earthquake

[2] "Earthquake Magnitude Scale." [Online]. Available: https://www.mtu.edu/geo/community/seismology/learn/earthquake-measure/magnitude/

[3] "Earthquake environmental effects," Nov. 2021. [Online]. Available: https://en.wikipedia.org/wiki/Earthquake_environmental_effects

[4] "Earthquakes - General Interest Publication." [Online]. Available: https://pubs.usgs.gov/gip/earthq1/predict.html

[5] S. E. Minson, M.-A. Meier, A. S. Baltay, T. C. Hanks, and E. S. Cochran, "The limits of earthquake early warning: Timeliness of ground motion estimates," *Science Advances*, vol. 4, no. 3, Mar. 2018. [Online]. Available: https://www.science.org/doi/10.1126/sciadv.aaq0504

[6] W. H. K. Lee, W. H. K. Lee, X. Lee, S. W. Stewart, and S. W. Stewart, *Principles and Applications of Microearthquake Networks*. Academic Press, 1981, google-Books-ID: NjW_CVz7NSoC.

[7] "Microquakes May Hint at the Big Ones." [Online]. Available: https://www.science.org/content/article/microquakes-may-hint-big-ones

[8] S. University, "Stanford AI Technology Detects Hidden Earthquakes - May Provide Warning of Big Quakes," Jan. 2021. [Online]. Available: https://scitechdaily.com/stanford-ai-technology-detects-hidden-earthquakes-may-provide-warning-of-big-quakes/amp/

[9] J. R. Kayal, "Microearthquake activity in some parts of the Himalaya and the tectonic model," *Tectonophysics*, vol. 339, no. 3, pp. 331–351, Sep. 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0040195101001299

[10] H. S. Kuyuk and O. Susumu, "Real-Time Classification of Earthquake using Deep Learning," *Procedia Computer Science*, vol. 140, pp. 298–305, Jan. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918319896

[11] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019. [Online]. Available: https://doi.org/10.1162/neco_a_01199

[12] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series," in *2019 IEEE International Conference on Big Data (Big Data)*, Dec. 2019, pp. 3285–3292.

[13] "GeoNet FDSN webservice." [Online]. Available: https://www.geonet.org.nz/data/tools/FDSN

[14] X. Du, "Short-term Earthquake Prediction via Recurrent Neural Network Models: Comparison among vanilla RNN, LSTM and Bi-LSTM," *Student theses*, 2022. [Online]. Available: http://resolver.tudelft.nl/uuid:ccfcdc1e-bd7c-44cb-a834-5b6b651dc09e

[15] I. Hashmi, "How does a CNN mixed with LSTM methods compare with the individual one in predicting earthquakes?" *Student theses*, 2022. [Online]. Available: http://resolver.tudelft.nl/uuid:c6c6a920-3a1f-477f-9ef5-5cb370a97924

[16] S. M. Mousavi, W. L. Ellsworth, W. Zhu, L. Y. Chuang, and G. C. Beroza, "Earthquake transformer-an attentive deep-learning model for simultaneous earthquake detection and phase picking," *Nature Communications*, vol. 11, no. 1, p. 3952, Aug. 2020, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41467-020-17591-w

[17] "Microearthquake," Jan. 2022. [Online]. Available: https://en.wikipedia.org/wiki/Microearthquake

[18] "Richter scale," Feb. 2022. [Online]. Available: https://simple.wikipedia.org/wiki/Richter_scale

[19] "How long does an earthquake last?" [Online]. Available: https://www.gns.cri.nz/Home/Learning/Science-Topics/Earthquakes/Monitoring-Earthquakes/Other-earthquake-questions/How-long-does-an-earthquake-last

[20] "IRIS: miniSEED." [Online]. Available: http://ds.iris.edu/ds/nodes/dmc/data/formats/miniseed/

[21] "IRIS: SEED Channel Naming." [Online]. Available: https://ds.iris.edu/ds/nodes/dmc/data/formats/seed-channel-naming/

[22] "Seismic Waves." [Online]. Available: https://topex.ucsd.edu/es10/es10.1997/lectures/lecture20/secs.with.pics/node3.html

[23] "Rectifier (neural networks)," Jun. 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Rectifier_(neural_networks)&oldid=1091009197

[24] "Fully Connected Layer: The brute force layer of a Machine Learning model," Mar. 2019. [Online]. Available: https://iq.opengenus.org/fully-connected-layer/

[25] "Sigmoid function," Jun. 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Sigmoid_function&oldid=1091009275

[26] "The fastai book," May 2022, original-date: 2020-02-28T19:26:47Z. [Online]. Available: https://github.com/fastai/fastbook

[27] "Confusion matrix," May 2022. [Online]. Available: https://en.wikipedia.org/wiki/Confusion_matrix

[28] s. saxena, "Precision vs Recall," May 2018. [Online]. Available: https://medium.com/@shrutisaxena0617/precision-vs-recall-386cf9f89488

[29] "Precision-recall curves: what are they and how are they used?" [Online]. Available: https://acutecaretesting.org/en/articles/precision-recall-curves-what-are-they-and-how-are-they-used

[30] D. Godoy, "Understanding binary cross-entropy / log loss: a visual explanation," Feb. 2019. [Online]. Available: https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

[31] "Downsampling (signal processing)," Apr. 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Downsampling_(signal_processing)&oldid=1081129508

[32] "Aliasing and Anti-Aliasing Filter - DSPIllustrations.com." [Online]. Available: https://dspillustrations.com/pages/posts/misc/aliasing-and-anti-aliasing-filter.html

[33] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 Regularization for Learning Kernels," May 2012, number: arXiv:1205.2653 arXiv:1205.2653 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1205.2653

[34] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," arXiv, Tech. Rep. arXiv:1711.05101, Jan. 2019, arXiv:1711.05101 [cs, math] type: article. [Online]. Available: http://arxiv.org/abs/1711.05101

[35] T. van Laarhoven, "L2 Regularization versus Batch and Weight Normalization," Jun. 2017, number: arXiv:1706.05350 arXiv:1706.05350 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1706.05350

[36] "AdamW and Super-convergence is now the fastest way to train neural nets Â· fast.ai." [Online]. Available: https://www.fast.ai/2018/07/02/adam-weight-decay/

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[38] "Early stopping of deep learning experiments | Peltarion Platform." [Online]. Available: https://peltarion.com/knowledge-center/documentation/modeling-view/run-a-model/early-stopping

[39] "GeoNet Data Policy." [Online]. Available: https://www.geonet.org.nz/policy

Appendices

Appendix A Comparison of pre-processing methods

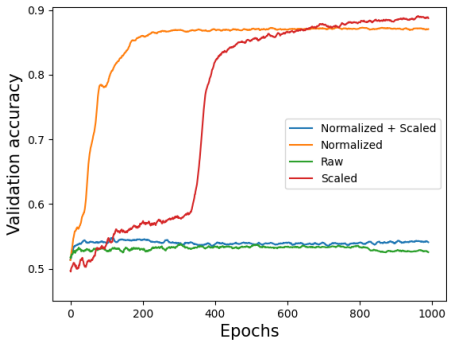

Figure 13: Validation accuracy with different pre-processing methods. Parameters: T of 30, H of 3 and 50HZ

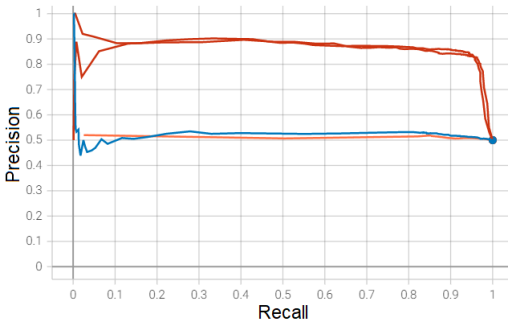

Figure 14: PR curve with different pre-processing methods: Normalized + Scaled (Blue), Raw (Orange), Normalized (Red) and Scaled (Red). Parameters: T of 30, H of 3 and 50HZ

Appendix B Additional waveforms samples

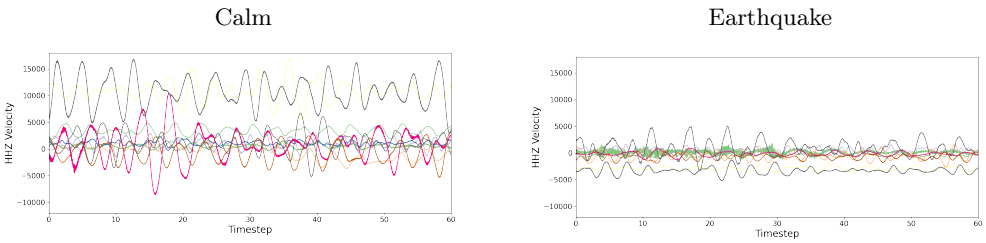

Figure 15: Raw Dataset. Recordings of 60 seconds from five different stations

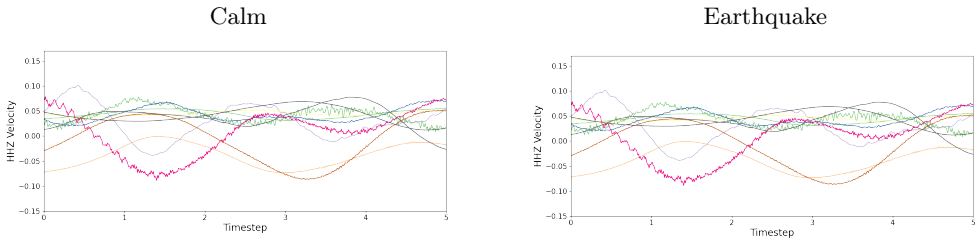

Figure 16: Normalized Dataset. Recordings of 5 seconds from five different stations

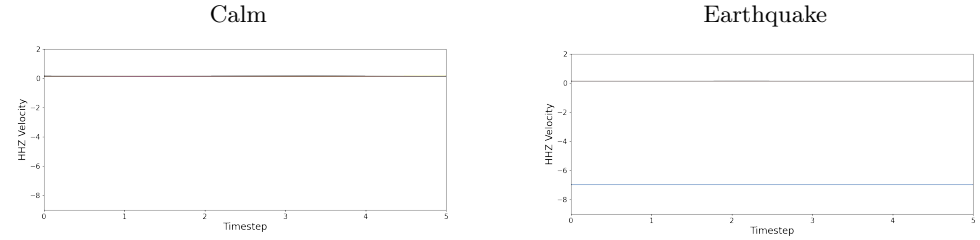

Figure 17: Scaled Dataset. Recordings of 5 seconds from five different stations

Appendix C Boxplots of test accuracy aggregated over T and HZ

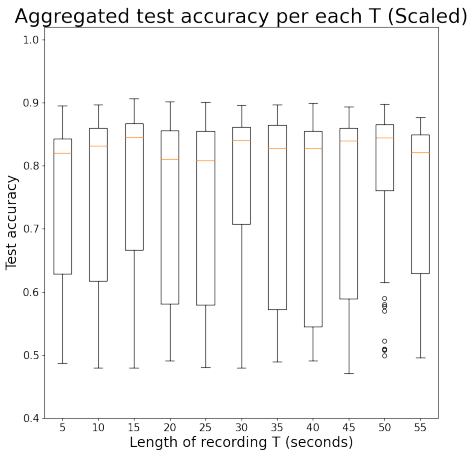

Figure 18: Boxplot with the aggregated test accuracies of different values of HZ for each value of T.

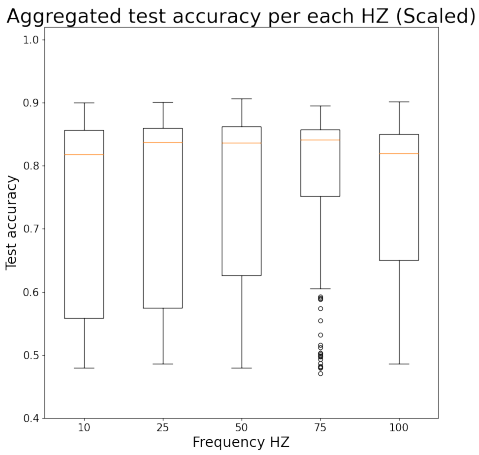

Figure 19: Boxplot with the aggregated test accuracies of different values of T for each sampling rate HZ.