Impact of seismic wave length to detect high-magnitude earthquakes via deep learning

Gancho Georgiev
Supervisor(s): Mohammad Sabbaqi, Elvin Isufi
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

Earthquakes are one of the most destructive natural phenomena, both in terms of human lives, and property damage. Although they are treated as a random phenomenon, the ability to predict them, even few seconds before they occur, could be of great benefit to society. Lots of research has been done on this topic but without any significant results. With the increase of seismic wave measurements data and since in recent years deep learning has solved many difficult problems, this paper aims to answer the impact of the seismic wave length in detecting high-magnitude earthquakes via Long Short-Term Memory (LSTM) neural network. Although the performance of the model was unsatisfactory, given the complex task of predicting earthquakes, as well as the resulting metrics not indicating any significant data in order to extrapolate a certain conclusion, it is worth further researching a duration of seismic waveform recordings of length 30 seconds, with sampling rate between 10 and 20 HZ, as these seismic waves seem to perform relatively best in our research.

1 Introduction

Earthquakes are one of the most destructive natural phenomena. The primary effects of earthquakes are landslides, ground shaking and rupture, tsunamis, fires, flooding and more. Earthquakes take an average of 27,000 lives a year since 1990, according to EM-DAT International Disaster Database [1]. Therefore the ability to predict earthquakes holds great benefit to society by potentially saving lives, reducing property damage, and minimizing economic losses. The ability to predict earthquakes is defined [2] by stating the exact time, location, and magnitude of an upcoming earthquake. The problem with this task is that an earthquake is treated to be an extremely nonlinear or random phenomenon [3], and thus there is no reliable method to predict them.

Nevertheless, there are two main approaches [3] in trying to predict earthquakes. The first approach is trend-based which uses statistical methods to identify periodicity in the occurrences of earthquakes. This approach is more suitable for long-term (years) risk assessments and predictions. The second approach, which this paper focuses on is precursors-based - more suitable for predicting earthquakes in the short-term. In 1989 the International Association for Seismology and Physics of the Earth's Interior (IASPEI) has issued a call to submit earthquake precursors to an evaluation panel for a critical review. By 1997 a total of 40 nominations had been reviewed [4], of which only five cases were selected as significant precursors - (1) foreshocks, (2) preshocks, (3) radon decrease in ground water, (4) ground water level increase, and (5) seismic quiescence before major aftershocks - the precursor this paper is focused on.

In recent years deep learning has solved many seemingly intractable problems, which gives the hope that it may also help predict earthquakes. According to Arnaud Mignan and Marco Broccardo [5], from the period 1994 - 2019, there have been roughly 77 scientific articles regarding earthquake predictions using artificial neural networks, which found two emerging trends - increase in interest of this problem over time, and increase of complexion of ANN models such as Long Short-Term Memory (LSTM) networks [6] and Convolutional Neural Networks (CNNs), which are considered the most complex architecture [5]. The paper concludes that despite the relatively positive results claimed by the articles, far simpler and traditional models offer similar, if not better results. Q.Wang et al. [6] have used a LSTM network with two-dimensional input to learn the spatio-temporal relationship among history earthquake data in different locations and exploit it to make predictions based on it. The accuracy they've achieved is 74.81%. Furthermore they've decomposed their original

LSTM into several smaller ones by dividing the sub-regions into groups which collectively cover their whole area of interest, and train the groups separately. Doing that allowed them to achieve 85.12% accuracy. Another research [7] used a three-layer perceptron neural network with a backpropagation learning algorithm and time-series magnitude data as input to predict earthquake magnitudes in Greece. The accuracy of their model was 80.55% for all seismic events, but only 58.02% for events with magnitude greater than 5.2 on the Richter scale. Another approach [8] used is a highly scalable convolutional neural network for earthquake detection and location from a single raw time-series waveform. The researchers studied the seismicity in Oklahoma, USA, and found that their method detects 17 times more earthquakes than previously catalogues by the Oklahoma Geological Survey, and is orders of magnitude faster than established detection methods such as Autocorrelation, and Fingerprint And Similarity Thresholding (FAST). To achieve that, they've analyzed the waveforms with a collection of nonlinear local filters, that were optimized to select features that are more relevant to classification during the training phase. In doing so they bypass the need to store a growing list of waveforms. Furthermore there are two [3] [9] trend-based researches that compare LSTM with Feed Forward Neural Network (FFNN) to predict earthquakes parameters such as latitude, longitude, depth, magnitude, and time. Both researches concluded that the LSTM outperforms FFNN by 59% in terms of a R^2 score. A third study [10] based on the same five parameters compared the performance of earthquake predictions in Morocco using LSTM and a multi-layer neural network has concluded that LSTM performs better in three metrics - Mean Absolute Error (MAE), Mean Squared Error(MSE), and accuracy, but is slower due to the time it takes to converge to the minimum error. When comparing the performance of three different recurrent neural networks with seismic waveform signals as inputs - vanilla RNN, LSTM, and bidirectional LSTM, it has been discovered that LSTM has the best performance with an averaged accuracy around 66.5%. The networks were designed and trained with 30 seconds of seismic waveform signals data, prior of an earthquake as input. The data for the earthquakes used consists of earthquakes happening between 2016 and 2020 in mainland New Zealand. One may argue that longer duration of seismic waveform recordings may lead to better results, as generally "more data equals better performance/accuracy". However that may not always be the case as "more data" may contain noise that can negatively influence the model, and thus the final result.

Therefore this paper aims to answer what is the optimal duration of seismic wave recordings prior to an earthquake when being used to predict earthquakes via deep learning, and more specifically high-magnitude earthquakes as they are the most destructive and expensive in terms of human loss, damages, and economic losses. Finding the optimal duration threshold of seismic wave recordings can hopefully help further earthquake prediction approaches that use seismic wave data as input to their models and achieve better results. Based on our results, it seems that the optimal seismic wave length is of duration 30 seconds, with a sampling rate between 10 and 20 HZ. Duration of seismic waves less than 15 seconds, and more than 45 seconds, as well sampling rate of 100 HZ and 2 HZ doesn't perform as well, which could be due to the fact that the data is either too much/less and/or has noise and thus our model can't extract any useful features to make predictions. It should be noted that these conclusions are not certain, given the results obtained.

2 Methodology

We obtained seismic waveform recordings of length 60 seconds, 3 seconds before a high-magnitude earthquake occurs between January 1, 2014 and December 31, 2018 in mainland New Zealand. We call the obtained seismic waveform recordings data - precursor data. It is shown in the yellow background at Fig.1.

The 3 seconds buffer (purple background Fig.1) before an earthquake occurs was chosen as the time in advance we would like to predict an earthquake. Initially the considered value was 30 seconds, as that could give enough time for people to potentially be able to react by evacuating or finding a safer spot to hide. However the initial experiments we ran showed that the model was unable to learn anything, and the prediction accuracy was as good as a coin flip. Discussing it with the supervisor, it was proposed to try a smaller value, anywhere between 1 and 5 seconds, with the hope of achieving better results. Thus 3 seconds was chosen as our buffer time.

We ran the model using two approaches. One was to use all the station data altogether to train our model, and the second approach was to train the model station by station. The goal of the research is to find the optimal length of the seismic waveform recordings (precursor data), 3 seconds prior to a start of a high-magnitude earthquake (vertical purple line Fig.1). 5, 15, 30, 45, and 60 seconds (as shown by the vertical green lines Fig.1) have been experimented with in this research.

Since our data is of high frequency - 100 HZ or 100 samples per second, it may contain noise that can negatively influence the result. Therefore we will downsample it, which may lead to risk of lacking useful data features. Thus finding the optimal sampling rate is also an important part in answering our research question. The frequencies that have been tried are 2, 5, 10, 20, 25, 50 and 100 HZ.

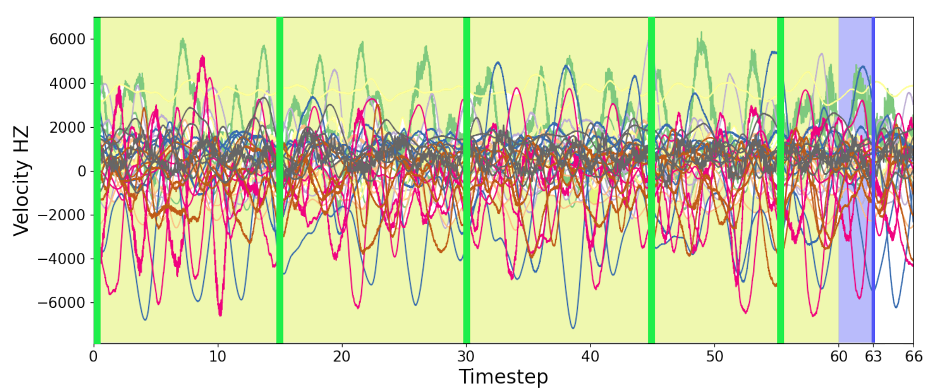

Figure 1: Seismic waveform recordings from 38 stations, 63 seconds prior of an earthquake.

2.1 Neural Network

The neural network we used is Long short-term memory (LSTM). The reason for choosing it is because previous studies (mentioned in the introduction) have concluded that it performs better with the data we are working with compared to other neural networks. Another reason for choosing this model is the nature of our data - time series. Time series is a sequence of data points that occur in a successive order throughout time, as time being the independent variable. LSTM models have proven useful for time series forecasting on multiple occasions [11]. Our LSTM model consists of 1 layer, hidden size of 2 (number of features of the hidden state), utilizing the Adam optimizer with learning rate of 0.001, and weight decay of 7e-5. To compute the binary cross-entropy loss the BCELoss method was used, along with sigmoid activation function, as that is the only activation function compatible with it. The LSTM model was implemented using Python 3.10 open source machine learning framework PyTorch, TensorBoard and matplotlib for visualization, scikit-learn for data preprocessing, numpy and pandas for manipulating data, and the python module pickle for loading and saving data.

3 Implementation

In this section we describe how we processed the raw earthquake data to obtain a balanced dataset for training, validation and testing.

3.1 Area of Interest and Station Filtering

We define a bounding box[1] representing the geographical area spanning over mainland New Zealand that we are interested in, by specifying four coordinates - bottom right and upper left longitudes and latitudes (Table 1).

Coordinal type	**Value**
Minimum Latitude	-47.749
Maximum Latitude	-33.779
Minimum Longitude	166.104
Maximum Longitude	178.990

Table 1: Bounding Box for the New Zealand dataset.

There are a total of 90 seismic stations that were active at some point of time in the period 2014 - 2018 inclusively consisting of weak motion (velocity) measurements at 100 Hz. However many stations had corrupt or missing data, therefore we kept only the stations that have measurements for all the events, which resulted in 38 stations that are displayed with red dots in Fig 2. The 38 stations codes choosen are listed at the bottom of the page.[2]

[1] https://wiki.openstreetmap.org/wiki/Bounding$_{Box}$

[2] 'BFZ', 'BKZ', 'DCZ', 'DSZ', 'EAZ', 'HIZ', 'JCZ', 'KHZ', 'KNZ', 'KUZ', 'LBZ', 'LTZ', 'MLZ', 'MQZ', 'MRZ', 'MSZ', 'MWZ', 'MXZ', 'NNZ', 'ODZ', 'OPRZ', 'OUZ', 'PUZ', 'PXZ', 'QRZ', 'RPZ', 'SYZ', 'THZ', 'TOZ', 'TSZ', 'TUZ', 'URZ', 'VRZ', 'WCZ', 'WHZ', 'WIZ', 'WKZ', 'WVZ'

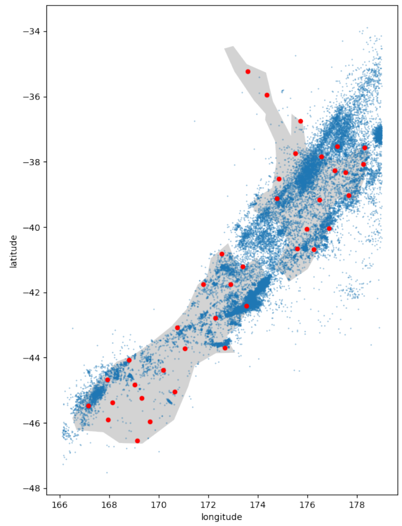

Figure 2: Geographic distribution of 38 seismic stations and earthquakes above 2.5 magnitude during 2014 - 2018

3.2 Earthquake Filtering

We retrieved all the earthquakes in the bounding box for the period 2014 - 2018 inclusively from the International Federation of Digital Seismograph Networks [12]. For each earthquake event, the information we keep is the earthquake's identifier, time, latitude, longitude, magnitude, and depth. A total of 118,060 earthquake events were obtained which can be seen by the blue dots in Fig 2. A distribution of the magnitudes of all the earthquakes is shown on (Fig 3).

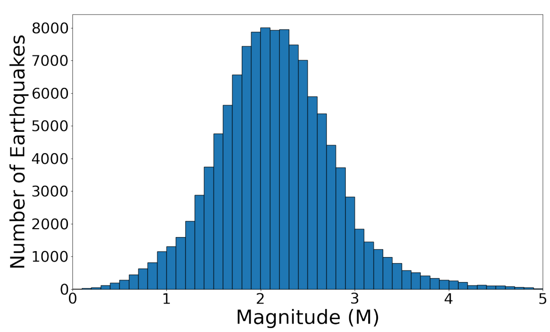

Figure 3: Distribution of magnitudes of the dataset.

Because we are interested only in high-magnitude earthquakes, we first have to define what this paper considers "high-magnitude earthquake", since there is no clear definition in the science community. The U.S. Geological Survey [13] considers earthquakes with magnitude of about 2.0 or less on the Richter scale as microearthquakes. The Michigan Technological University [14] classifies earthquakes with magnitude of 2.5 or less on the moment magnitude scale as ones that are usually not felt, but can be recorded by seismograph (moment magnitude estimates are roughly the same as Richter magnitudes for small to large earthquakes). Additionally in a study [15], "low-magnitude" earthquakes were defined as having magnitude between 2 - 3.5. Based on the above definitions, the histogram (Fig 3), as well as magnitude distributions for other years my colleagues have gathered, we've decided to define high-magnitude earthquakes as ones having a magnitude above 2.5, as that number happens to be roughly at the center of the distributions. Since we are interested in predicting high-magnitude earthquakes that we just defined, filtering out all the earthquakes below 2.5 magnitude resulted in 31,991 earthquakes. Fig 4 shows the waveform recordings of three random high-magnitude earthquakes.

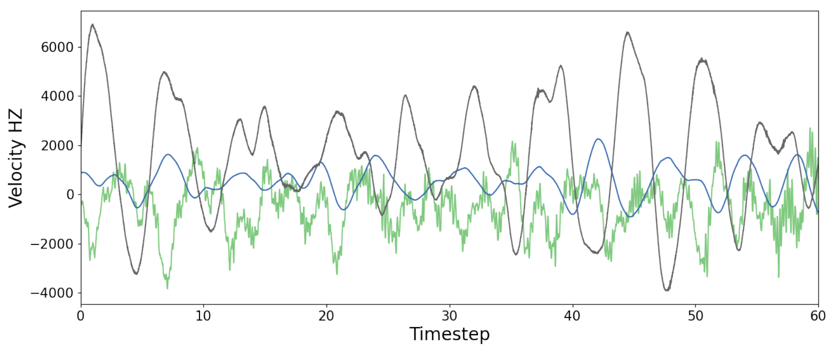

Figure 4: Active waveform recordings of three random earthquakes

3.3 Seismic Waveform

We first retrieved all the seismic waveform recordings of weak motion (velocity) measurements at 100 Hz (channel=HHZ) from the 38 seismic stations located in our area of interest for a duration of 60 seconds (precursor data), 3 seconds prior to the occurrences of the high-magnitude earthquake events we have gathered. This resulted in 29,899 waveform recordings. We refer to these waveform recordings as "active". We then generate "normal" waveform recordings, which are waveform recordings that do not lead to an earthquake. To obtain these recordings, we use an algorithm where we sort all the earthquakes from our dataset by their timestamps, and removed all the earthquakes that are within 5,000 seconds apart from each other, which produced 10,405 events. From that list, for each event we went 2,000 seconds before an occurrence of an earthquake and retrieve 60 seconds of duration of seismic recording, which resulted in 8,902 "normal" recordings. We sanitize the "active" and "normal" waveform recordings, by dropping any recording that has missing values, which produced 19,087 and 4,682 recordings respectively. Since different stations have different ranges of velocity, we normalized the velocities using the Normalizer class of the scikit-learn library to both the "active" and "normal" seismic waveform recordings between the values -1

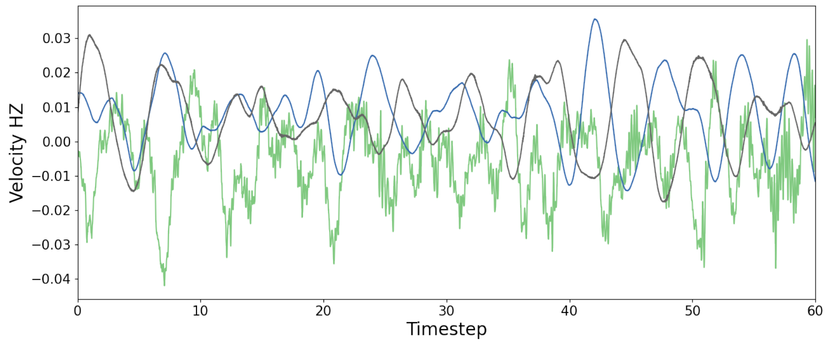

Figure 5: Normalized active waveform recordings of three random earthquakes

and 1, and use those normalized waveform recordings for our model. Normalized waveform recordings of 3 random earthquakes can be seen in Fig 5.

Finally we took an equal amount of "active" and "normal" waveform recordings and label them with 1 and 0 respectively for an earthquake happening or not, and put them together in a dataset we will use to train, validate and test our LSTM model to do a binary classification prediction. Since we ended up with 4682 "normal" recordings, the final dataset will consist of total of 9364 recordings. The dataset was randomly shuffled and split into training, validation and test sets of 60/20/20 ratios respectively with each set consisting of equal number of "active" and "normal" waveform recordings.

3.4 Overfitting and Parameter Tweaking

Our LSTM model was overfitting - achieving good accuracy for the training data set, but producing poor results on unseen data. To combat this we tried using regularization which is a technique used to calibrate machine learning models in order to minimize the adjusted loss function by adding a penalty term to it, and thus prevent overfitting or underfitting. The type of regularization we used was L2, which adds squared magnitude of coefficient as penalty term to the loss function. This was incorporated through the "weight_decay" parameter of our optimization function. While we didn't get rid of overfitting altogether, it helped improve our model for some combinations of precursor data and sampling rate, while for others didn't make much of a difference. Additionally, because of overfitting, to give more objective result about the accuracy of the model, we decided to use K-Fold Cross Validation which splits our dataset into separate training and test subsets. We choose the value of K to be 5 - dividing the dataset into 5 equal parts that our model will run 5 times and each time use a different test set.

Various of parameters were manually tweaked both in the LSTM model as well as in the dataset in order to find optimal results. The list includes:

learning rate - 0.0001, 0.001, 0.005
number of epochs - 50, 80, 100, 120, 150
hidden size - 1, 2, 3, 4
weight decay - 1e-6, 1e-5, 1e-4, 7e-5
precursor data duration - 5, 15, 30, 45, 60 seconds
samples per second - 2, 5, 10, 20, 25, 50, 100 HZ

4 Results and Discussion

Based on various tests, it was concluded that the optimal parameters of the LSTM model are a learning rate of 0.001, 100 number of epochs, hidden size of 2, and weight_decay of 7e-5. The accuracy achieved of different duration of precursor data and its sampling rate is displayed below in Table 2. The highlighted numbers represent the highest accuracy achieved for the given precursor data. Each combination of sampling rate and precursor data was ran 3 times in our model and taken the average, in order to achieve a more accurate result and avoid variance and flukes. We are aware that this amount of times is too little for representative accurate results, and have addressed that issue in the Responsible Research part of the paper.

It can be observed from Table 2 that overall a sampling rate between 10 and 20 samples per second seem to be the optimal rate (with an exception of one - the 5 seconds precursor data that for 25 HZ achieved an accuracy just 0,007 higher than its 10 HZ counterpart). Based on the relatively lower accuracy results for precursor data with high sampling rate of 100 HZ, it seems that it contains too much noise, and our model performs poorly with it. Contrarily, for sampling rate of 2 HZ it seems that there is not enough data, and the model can't extract important features and information from it. Regarding the duration of the precursor data, it can be observed that the 30 seconds of seismic waveform recordings performs best, with 15 sec, and 45 sec coming second and third respectively. Duration smaller than the 30 sec ±15 sec window seems not enough for the model to learn and perform as accurately, and similarly, duration longer than the window is too much data for the model to extrapolate information. Overall, both for the sampling rate and precursor data, the accuracies are so close within each other, that one is unable to make with certainty any definite conclusions.

	Duration of precursor data				
Samples per sec	60 sec	45 sec	30 sec	15 sec	5 sec
100	0,5551	0,5573	0,5858	0,6	0,5816
50	0,6163	0,5801	0,5955	0,6326	0,5704
25	0,5807	0,5979	0,5512	0,6221	**0,609**
20	**0,6217**	0,6041	0,5803	**0,6381**	0,5727
10	0,6166	**0,6253**	**0,6405**	0,6189	0,6022
5	0,6074	0,5968	0,6391	0,5917	0,5329
2	0,5415	0,6229	0,5768	0,5402	0,5373

Table 2: Accuracy table based on duration of precursor data and sampling rate.

The highest accuracy run was 0.6679 and was achieved with a duration of 30 seconds and a sampling rate of 10 samples per second. The model accuracy and loss function of these parameters can be seen in Fig.6 and Fig.7 respectively.

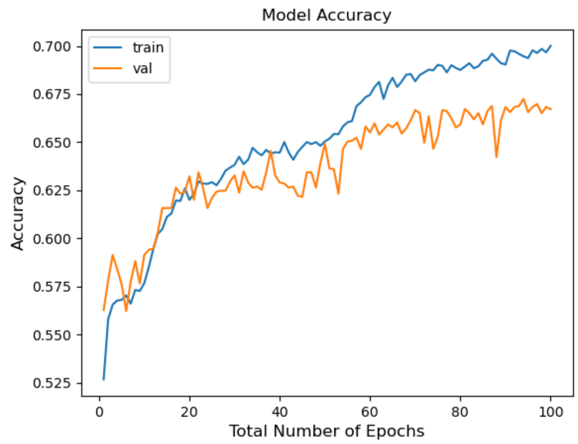

Figure 6: Model accuracy graph of 30 sec duration of precursor data with 10 samples per sec

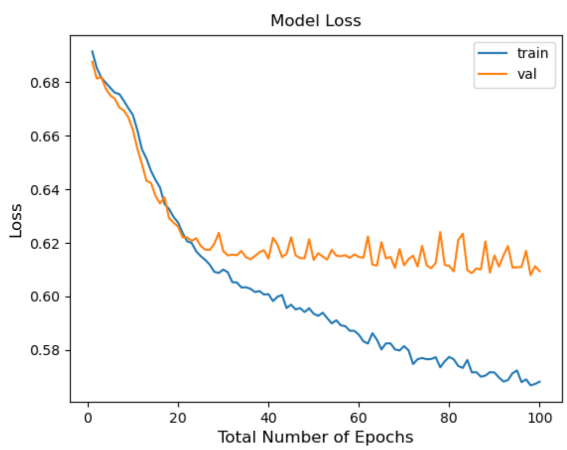

Figure 7: Model loss graph of 30 sec duration of precursor data with 10 samples per sec

Estimating the skill of the model for 30 sec of precursor data and 10 HZ sampling rate by using the k-fold cross validation technique with k = 5, produced an average accuracy of 0.6301, which is near the 0.6405 average accuracy we've obtained earlier. This is a good estimation of the accuracy of the model in general (given that new unseen data will build the same distribution as our observed data). Fig.8 shows the 5-fold accuracy.

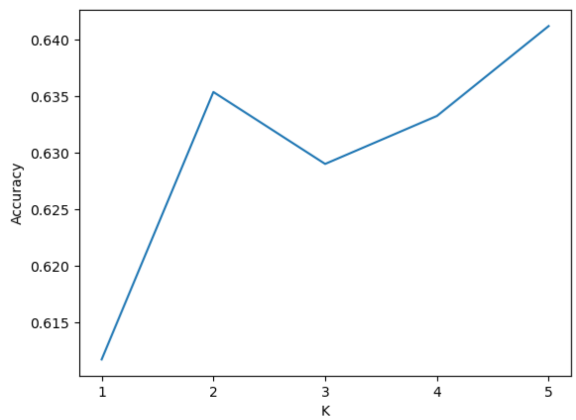

Figure 8: 5-fold accuracy for 30 sec of precursor data and 10 HZ sampling rate

For the second approach - running our model station by station, it was ran 4 times each with 30 seconds of precursor data and 10 HZ sampling, which yielded the average results seen in Table 3 (on page 13). Most stations performed poorly, with most accuracies falling between 0.5 and 0.55. Three stations stood out with high accuracy - DCZ, RPZ, and URZ with accuracies of 0.7161, 0.645, and 0.6668 respectively. An interesting observation when running the data station by station was that the accuracies between different runs were all the same down to two decimal places, which was not the case when running the data with all the stations altogether. Another observation was that for some stations, just 20-40 epochs were needed for the loss and accuracy functions to converge, while for others 100 or even more were needed.

5 Responsible Research

Taking a complex and random natural phenomena such as an earthquake and reducing it to potentially a simple binary classification prediction model, would require lots of thought to important moral and ethical questions, such as what are the acceptable metric values that should be considered trustworthy and beneficial to society, as otherwise may lead to false sense of security. Although the prediction accuracy in our model is low and unsatisfactory, other metrics are important as well, such as the false positives and false negatives. Many false positives (earthquake is predicted, but none occurs) would lead to people panicking and eventually lose trust and confidence in the earthquake prediction model. On the other hand, false negatives (no earthquake warning is given, yet an earthquake occurs) could lead to loss of lives and economic destruction.
When running all the station data altogether, our model was ran just 3 times per specific sampling rate and seismic waveform duration combination. This amount of times is too little to avoid any variance and biases, and therefore produce true results, but was due to the fact partly because they were ran manually, but most importantly because of the limitation of time we had. The law of large numbers theorem [16] suggests that repeating an experiment multiple times until the average result starts to converge will better approximate the true/expected underlying result. Similarly when running the model station by station,

they were ran just 4 times so the above remarks are valid here as well. Therefore the reader should be aware of that if they are to reproduce the experiment. Another point should be made that the optimal duration of seismic wave recordings and downsampling rate may be different from the ones tested, since an exhaustive search with all possible combinations has not been done.
The research is made in a way that is reproducible and repeatable. The dataset that was used is publicly available and can be accessed through the link provided in the references. The preprocessing steps, as well as the architecture of the model and parameters used are also explained in detail. Furthermore the code will be shared, and can be reused.

6 Conclusions and Future Work

Many things can be done for future work to improve on the model and results. Firstly, as explained in the Responsible Research section, the experiments should be ran multiple times, until the average result converges as to get a more representative and true result. Secondly, a hyperparameter optimisation technique such as a grid search or a random search could be utilized, which tests and compares the performance of different combinations of parameters of the LSTM model, such as the number of epochs, batch size, learning rate, hidden size, and optimizer in order to find the optimal parameter values. Thirdly, one can optimize the algorithm that generates "normal" waveform recordings. The current one produces too few recordings, and that number limits the overall size of the dataset as it has to be equally balanced between "active" and "normal" waveform recordings. Therefore improving the algorithm to create a larger amount of "normal" waveform recordings will increase the amount of data we can train, validate and test our model on. Another thing that can be done is to obtain more earthquakes from a longer range of period. In this research we used 5 years of data, but using more data should hopefully increase the accuracy of the model. Additionally, while in our research we only used the channel HHZ that is oriented vertically, trying the other two high broadband and weak motion channels - HHN oriented north-south, and HHE oriented west-east, either individually or in tandem, could produce better results. A different approach could also be tried, such as instead of gathering seismic recordings from every station, to use just one seismic recording - that of the closest station to the earthquake, and train the model on those.

Although the highest average prediction accuracy of 0.6405 was achieved, 3 seconds before a high-magnitude earthquake strucks, with an optimal duration of 30 seconds of precursor data at 10 HZ, this result is unsatisfactory in terms of accuracy. Additionally, the 3 seconds of time prior to an earthquake it tries to predict can hardly make a difference for a person to react given an earthquake is to happen. Yet it could help machines/robots/computers switch off gas/electricity or other things and possibly save further lives and property damage. While a duration window of 30 seconds ± 15 seconds, and a sampling rate of 10 - 20 HZ showed the best results, they cannot be considered as certain. Overall, reliable earthquake prediction model remains an unachievable task for people at this point of time [17]. Throughout the research it was founded that the model performs differently on different set of years, and thus is not a reliable nor accurate enough method given the responsible and important task of predicting earthquakes.

Station codes

	BFZ	DCZ	DSZ	EAZ	HIZ	JCZ	KHZ	KNZ	KUZ	LBZ
Accuracy	0,5543	0,7161	0,4968	0,5600	0,5297	0,5461	0,5350	0,5779	0,5466	0,6213
Precision	0,5904	0,6723	0,4901	0,5707	0,5399	0,5352	0,5285	0,5628	0,5375	0,6826
Recall	0,4065	0,8435	0,5143	0,5258	0,4961	0,7097	0,6472	0,6981	0,6684	0,4534
F1-score	0,4636	0,7481	0,4671	0,5366	0,4942	0,6089	0,5819	0,6232	0,5958	0,5449

	LTZ	MLZ	MQZ	MRZ	MSZ	MWZ	MXZ	NNZ	ODZ	OPRZ
Accuracy	0,5191	0,5281	0,5233	0,5334	0,5397	0,5000	0,5487	0,5122	0,5651	0,4915
Precision	0,5169	0,5365	0,5153	0,5355	0,5413	0,5000	0,5579	0,5179	0,6066	0,4897
Recall	0,5816	0,4131	0,7839	0,5032	0,5212	1,0000	0,4693	0,3528	0,3708	0,4025
F1-score	0,5474	0,4668	0,6218	0,5188	0,5310	0,6667	0,5098	0,4197	0,4602	0,4419

	OUZ	PUZ	PXZ	QRZ	RPZ	SYZ	THZ	TOZ	TSZ	TUZ
Accuracy	0,5715	0,5106	0,5980	0,5281	0,6340	0,6006	0,5752	0,5191	0,5339	0,5959
Precision	0,5556	0,5094	0,5702	0,5250	0,6078	0,5624	0,5755	0,5153	0,5285	0,5834
Recall	0,7150	0,5763	0,7956	0,5900	0,7553	0,9068	0,5731	0,6441	0,6292	0,6706
F1-score	0,6253	0,5408	0,6643	0,5556	0,6736	0,6942	0,5743	0,5725	0,5745	0,6240

	URZ	VRZ	WCZ	WHZ	WIZ	WKZ	WVZ	BKZ
Accuracy	0,6668	0,5636	0,6054	0,5207	0,5371	0,5048	0,5365	0,5704
Precision	0,6041	0,6064	0,6035	0,5288	0,5486	0,5055	0,5331	0,5677
Recall	0,9682	0,3623	0,6144	0,3792	0,4184	0,4396	0,5879	0,5911
F1-score	0,7440	0,4536	0,6089	0,4417	0,4748	0,4703	0,5592	0,5791

Table 3: Results of running the dataset on 30 sec of precursor data and 10 HZ on all 38 stations individually.

References

[1] "Dat: The international disasters database." `https://www.emdat.be/`.

[2] C. R. Allen, "Responsibilities in earthquake prediction," *Bulletin of the Seismological Society of America*, vol. 66, no. 6, p. 2069â2074, 1976.

[3] T. Bhandarkar, V. K, N. Satish, S. Sridhar, R. Sivakumar, and S. Ghosh, "Earthquake trend prediction using long short-term memory rnn," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 2, p. 1304, 2019.

[4] M. Wyss, "Second round of evaluations of proposed earthquake precursors," *Pure and Applied Geophysics PAGEOPH*, vol. 149, no. 1, p. 3â16, 1997.

[5] A. Mignan and M. Broccardo, "Neural network applications in earthquake prediction (1994â2019): Meta-analytic and statistical insights on their limitations," *Seismological Research Letters*, vol. 91, no. 4, p. 2330â2342, 2020.

[6] Q. Wang, Y. Guo, L. Yu, and P. Li, "Earthquake prediction based on spatio-temporal data mining: An lstm network approach," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 1, p. 148â158, 2020.

[7] M. Moustra, M. Avraamides, and C. Christodoulou, "Artificial neural networks for earthquake prediction using time series magnitude data or seismic electric signals," *Expert Systems with Applications*, vol. 38, no. 12, p. 15032â15039, 2011.

[8] T. Perol, M. Gharbi, and M. Denolle, "Convolutional neural network for earthquake detection and location," *Science Advances*, vol. 4, no. 2, 2018.

[9] V. R. Sivaiahbellamkonda, Lavanyasettipalli and M. Vemula, "An enhanced earthquake prediction model using long short-term memory," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, p. 2397â2403, 2021.

[10] A. Berhich, F. Z. Belouadha, and M. I. Kabbaj, "Lstm-based earthquake prediction: Enhanced time feature and data representation," *International Journal of High Performance Systems Architecture*, vol. 10, no. 1, p. 1, 2021.

[11] J. Korstanje, "How to select a model for your time series prediction task [guide]." `https://neptune.ai/blog/select-model-for-time-series-prediction-task`, May 2022.

[12] "Geonet fdsn webservice." `https://www.geonet.org.nz/data/tools/FDSN`.

[13] "The severity of an earthquake." `https://pubs.usgs.gov/gip/earthq4/severitygip.html`.

[14] M. T. University, "Earthquake magnitude scale." `https://www.mtu.edu/geo/community/seismology/learn/earthquake-measure/magnitude/`, Oct 2021.

[15] S. J. Gibbons, M. BÃžttger SÃžrensen, D. B. Harris, and F. Ringdal, "The detection and location of low magnitude earthquakes in northern norway using multi-channel waveform correlation at regional distances," *Physics of the Earth and Planetary Interiors*, vol. 160, no. 3-4, p. 285â309, 2007.

[16] J. Brownlee, “A gentle introduction to the law of large numbers in machine learning.” https://machinelearningmastery.com/a-gentle-introduction-to-the-law-of-large-numbers-in-machine-learning/, Aug 2019.

[17] M. Wyss, “Cannot earthquakes be predicted?,” *Science*, vol. 278, no. 5337, p. 487â490, 1997.