Short-term Earthquake Prediction with Deep Neural Networks

Finding the optimal time prior to earthquake strikes to use in predictions

Glenn van den Belt
Supervisor: Mohammad Sabbaqi
Responsible Professor: Elvin Isufi
EEMCS, Delft University of Technology, The Netherlands

August 22, 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 can have tremendous effects. They can result in casualties, massive damage, and hurt the economy. Therefore, one would like to predict earthquakes as early as possible and with the highest accuracy possible. This paper contains the proposal for the optimal prediction-time, which is the time between the execution of a prediction and the actual earthquake strike, for deep learning models. Only short-term predictions and high-magnitude earthquakes are considered. A prediction means to define whether an earthquake happens or does not happen in an upcoming amount of seconds. A short-term prediction means a forecast to the extent of seconds. A high-magnitude earthquake means an earthquake with a magnitude of 2.5 or higher. This research uses the Long Short Term Memory deep learning model to test the optimal prediction-time value for earthquake predictions. The optimal value for the prediction-time is found by testing the model with different values for the prediction-time and concluding when the model performs best. For prediction-times from moments before the strike until 40 seconds, the model is performing worse compared to higher prediction-times. The model's performance peaks at a prediction-time of 70. When increasing further than 70, the performance decreases until a prediction-time of one hundred. When rising even further, the performance is stabilising. Thus, for predictions with the highest performance, one should use a prediction-time of 70 seconds.

1 Introduction

An earthquake is one of the most dangerous and destructive natural disasters. The occurrence of earthquakes can result in casualties, massive damage, and a sudden downfall in the economy [1]. They often occur without any warning in advance, which leaves not enough time for people to take any measures. In addition, earthquakes might lead to other natural hazards such as tsunamis, snowslips, and landslides [2]. Consequently, even if one can predict an earthquake only moments before it strikes, it could be of incredible value.

Earthquake predictions can be mainly classified into four categories according to the employed methodologies [3]:

1. Mathematical analysis
2. Precursor signal investigation
3. Machine learning algorithms
4. Deep learning methods

Since earthquakes have an intrinsic random nature by themselves, the last two methods have been gaining in popularity over the last couple of years [4].

In previous work, people tried to predict the magnitude of the biggest upcoming earthquake in the next week, in which a test accuracy of 82.0 was achieved using a decision tree [3]. Another work tried to predict whether or not an earthquake with a magnitude six or higher will take place in the upcoming 30 days, in which the best R score is 0.303 [5]. Xiangyu Du tried to predict earthquakes in the FDSN Earthquake dataset [6]. Using a Long Short Term Memory Network, he achieved an average accuracy of 66.5.

This research paper aims to find the optimal short-term prediction-time for high-magnitude earthquake predictions for deep learning models. The prediction-time is defined as the time between the execution of a prediction and the actual earthquake strike. The prediction-time will be referred to as H for the remainder of this paper. A prediction means to define whether an earthquake happens or does not happen in an upcoming amount of seconds. Short-term forecasts are forecasts to the extent of seconds. An optimal number of seconds does not mean having the most considerable accuracy when performing the predictions. Indicating earthquakes more in advance is also considered a precious asset; therefore, this research aims to find a good balance between the two. If an optimal value of H is found, humanity is one step closer to being able to predict earthquakes with higher accuracies and potentially prevent many disastrous situations.

2 Methodology

The following chapter introduces the method used to predict earthquakes, which is ultimately required to determine H's optimal value. In this research, the earthquakes are predicted by a deep learning model. Various deep learning models have been considered in Section 2.1. Deep learning models introduce performance trade-offs, described in section 2.2. The final section 2.3 explains how the optimal value of H can be found.

2.1 Deep learning model

To predict earthquakes with deep learning methods, one must first consider which model best fits the short-term prediction of earthquakes. During the research, the following models have been considered:

1. Long Short Term Memory Network
   The Long Short Term Memory Network (LSTM) is a recurrent neural network architecture. Xiangyu Du achieved an average accuracy of 66.5 on unseen earthquake data coming from the New Zealand dataset using an LSTM model [6].

2. Support Vector Machine
   A study has shown that the Support Vector Machine (SVM) is able to forecast financial time series [7]. Both financial time series data and earthquakes data are very nonlinear and random, which might conclude that the SVM is also suitable for earthquake predictions.

3. Convolutional Neural Network
   Convolutional Neural Network (CNN) has had groundbreaking results in various fields related to pattern recognition. CNN has an excellent performance in machine learning problems [8]. The most significant earthquake magnitude in Taiwan for the upcoming 30 days is predicted using convolution neural networks [5].

4. Mixture
   One could also consider creating a combination of any of the above, as has been done to detect heart rhythm disorders [9]. Combinations of other machine learning algorithms than the aforementioned could be possible as well.

The earthquake data is a time series classification forecasting problem. For this kind of problem, the LSTM has been used many times, such as in oil production forecasting [10]. Besides that, Xiangyu Du used exactly this model to predict earthquakes in the same dataset that is considered during this research. Therefore, the decision has been made to use the LSTM model.

2.2 Performance and trade-offs

In order to determine whether or not the model is producing valuable results, the performance has to be measured. The performance of a deep learning model can be measured in various ways. The first one being considered is the confusion matrix. The confusion matrix deals with the following components:

1. True Positive (TP), which means the model predicts the earthquake correctly.
2. True Negative (TN), which means the model predicts an earthquake is not about to strike.
3. False Positive (FP), which means the model predicts that an earthquake will happen, although it does not occur. This can be costly, for example, when authorities and first responders start evacuating procedures while there is no harm.
4. False Negative (FN), which means the model predicts that an earthquake is not about to strike, although it is happening. There will be no warning for an actual earthquake, resulting in devastating outcomes. A false negative should be considered the most costly outcome.

From the outcomes of a confusion matrix, several other performance metrics can be derived:

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

$Precision = \frac{TP}{TP+FP}$

$Recall = \frac{TP}{TP+FN}$

$F1 - Score = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN}$

Since the F1-Score already integrates the precision and recall metrics, the precision and recall metrics themselves will not be considered a lot. The accuracy and F1-Score will be the primary metrics used during this research. In addition to the metrics above, a deep learning model attribute called loss is considered. Each deep learning model has a cost function which is tried to be minimised. The outcome of the cost function is the loss.

The consideration of the trade-offs of the model's performance and the value of specific input parameters is essential. One might prefer to predict earthquakes minutes earlier with an accuracy slightly less than the situation where one predicts the earthquake moments before the actual strike with higher accuracy. Another essential aspect to consider is the size of the seismological wave. It is possible to feed a very long wave to the model; however, this requires a very high amount of storage capability and processing power. Since the input size of the seismic waveform is not the research question of this paper, a fixed value of 60 will be picked for the input size.

2.3 The optimisation of H

Now that an optimised model exists, finding an optimal value of H is possible. Many different values for H will be fed into the model. The model will be run multiple times and averaged for each value of H to prevent possible outliers. After that, the performance metrics are compared, and optimal values of H will be concluded.

3 Implementation

In the next section, the implementation of the model will be discussed. Data is required to train the model. How this data is retrieved, processed and evaluated will be addressed in section 3.1. After that, details and design choices regarding the LSTM model are examined in 3.2. Lastly, the overfitting issue will be analysed in 3.3.

3.1 Data

Considerable amounts of data are required to train an LSTM model properly. All the data that is used to train the LSTM model is retrieved from the FDSN dataset [11]. The data consist of earthquakes and accessory seismological waves from a particular measurement station in the country. During this research, an earthquake with a magnitude of 2.5 or higher is considered a high-magnitude earthquake. Earthquakes with a magnitude higher than 2.5 are often felt but only cause minor damage [12]. The threshold of 2.5 is chosen because this value results in an approximately balanced split regarding the number of earthquakes. In Figure 1, the distribution of over 400.000 earthquakes according to their magnitude is visualised. Note that the distribution takes the form of a normal distribution with a mean value of roughly 2.5.

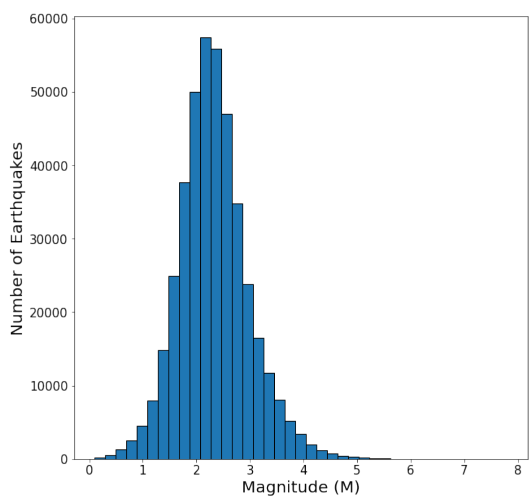

Figure 1: Distribution of earthquakes retrieved from the FDSN dataset according to the magnitude

Data Retrieval

The essential characteristics of earthquakes are the longitude, latitude, magnitude and depth. Before the actual earthquake dataset is built, several actions are taken. At first, a bounding box is defined. An earthquake is only considered if it is within this box. The bounding box represents a rectangular outline of the mainland of New Zealand. Within the box, 38 stations are defined. These 38 stations were chosen since they did not contain as much corrupt data as other stations. The specifications of every station can be found in Appendix A.1. Only the seismic waveforms from these stations are considered. Table 1 shows the coordinates of the bounding box, and in Figure 2, the 38 measurement stations are plotted on the New Zealand map.

Coordinate	Value
Bottom-left longitude	166.104
Bottom-left latitude	-47.749
Top-right longitude	178.990
Top-right latitude	-33.779

Table 1: Specifications of the bounding box

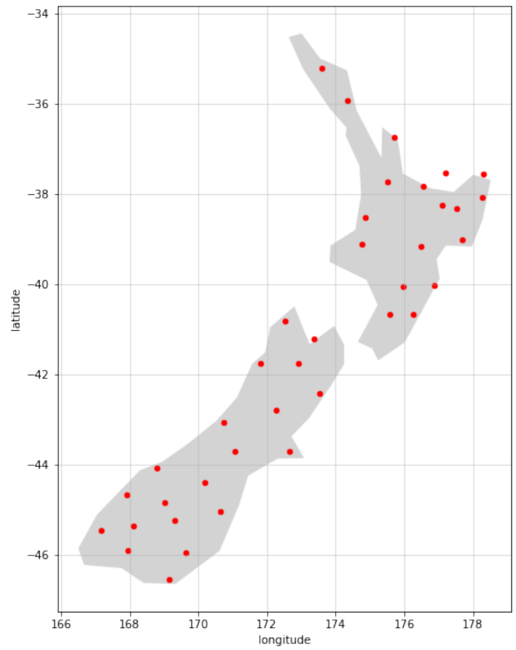

Figure 2: 38 measurement stations plotted on the map of New Zealand

An example of a seismic wave measurement including a specification of H of one particular station without any preprocessing is given in Figure 3.

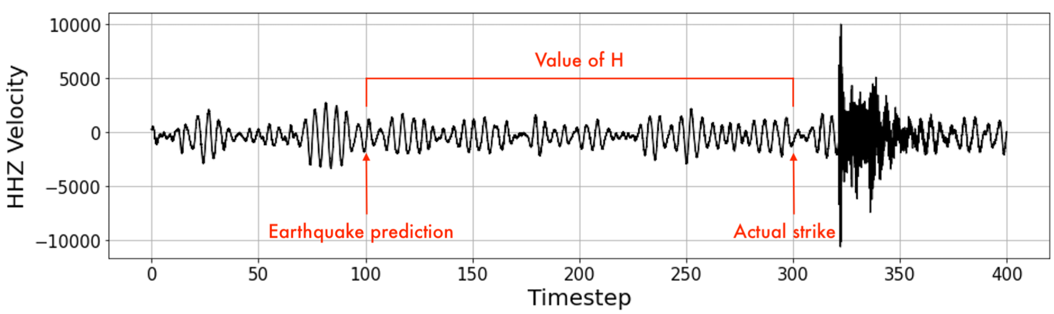

Figure 3: Example of a seismic waveform recording with an earthquake prediction at time step 100, the earthquake strike at time step 300, and an H-value of 200 seconds

This research tries to perform short-term earthquake predictions on high-magnitude earthquakes. In order to retrieve data to train the classifier when it should classify to 1, meaning that an earthquake will happen, the earthquakes with a magnitude of 2.5 or higher are filtered. To retrieve data to train the model to classify for 0, one could consider two options:

1. Train the model with waveforms from earthquakes with a magnitude of 2.5 and lower.
2. Train the model with 'flat' waveforms where no actual earthquake occurs. Flat waveforms are acquired by finding two consecutive earthquakes at least 5000 seconds apart from each other and then retrieving a waveform 2000 seconds before the last of the two earthquakes.

This research aims to find an optimal H and is not aiming to determine what kind of low-earthquakes to use. Since it might be challenging for the model to distinguish between seismic waveforms of an earthquake of a low-magnitude earthquake with a magnitude of 2.4 and a high-magnitude earthquake with a magnitude of 2.6, the decision has been made to only consider the second option with flat waves.

Filtering

Filters are applied to the dataset for better prediction results. As aforementioned, the first filtering method is to leave out the earthquakes and their corresponding waveforms that do not reside within the bounding box.

The following filter is based on the value of the magnitude. Two datasets are created, one consisting of earthquakes with magnitudes of 2.5 and higher and the other containing waves where no earthquake is happening. The distribution of the earthquakes according to their magnitude is already given in Figure 1.

The last filter is based on the depth of an earthquake. Figure 4 shows that most earthquakes have a depth of 200 kilometres or less. Since this research does not deal with the difference between shallow and deep earthquakes, the decision has been made only to consider earthquakes with a depth of 200 kilometres or less to avoid non-similar earthquakes in the dataset.

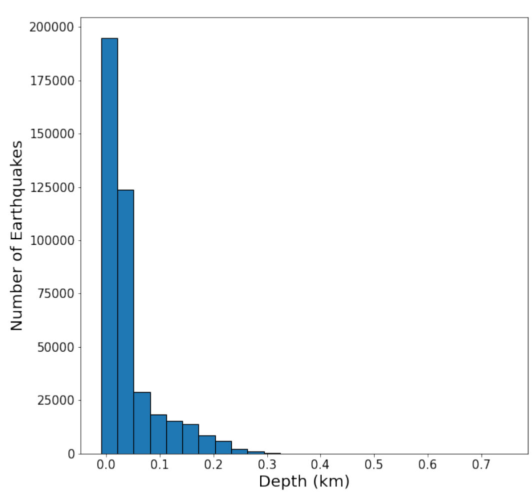

Figure 4: Distribution of earthquakes retrieved from the FDSN dataset according to depth

Preprocessing

Many different preprocessing methods have been considered and will be given in an overview:

1. Sanitize, which removes missing values and drops measurements with waveforms that have an insufficient length and are therefore incomplete.

2. Downsampling, which reduces the number of measurements (frames) per second.

3. Normalizing, which performs a normalization between -1 and 1. Normalizing means to scale the input vectors individually to unit form [13].

4. Scaler, which means to standardize features by removing the mean and scaling to unit variance [14].

When testing the preprocess methods, it turns out that the sanitization deletes roughly 40 percent of a dataset on average, meaning that 40 percent of the dataset is considered corrupted data. The downsampling did not harm the performance of the model a lot, although it does speed up the training time significantly. It turns out that normalizing over the whole dataset resulted in the highest performance. Using only the scaler also resulted in decent outcomes. Combining the normalization method and the scaler gave poor results.

3.2 LSTM design

The model that is used during this research is the LSTM model. The input of the LSTM model is a seismic waveform. The number of frames of these seismic waveforms is the input size of the LSTM cell [15]. The hidden size is set to two and the number of layers to one. The output of the LSTM cell is fed into a Rectified Linear Unit (ReLU) layer. A ReLU is a piecewise linear function that will output the input directly if it is positive; otherwise, it will output zero [16]. The output of the first ReLU layer will go into a fully connected layer. Fully connected layers in neural networks are those layers where all the inputs from one layer are connected to every activation unit of the next layer [17]. The fully connected layer outputs 128 features, which go into a second ReLU layer. The ReLU outputs the 128 features into another fully connected layer, which outputs one feature. This one feature goes into a sigmoid function. The sigmoid function is the logistic function, which maps any real value to the range between 0 and 1. If the value is below 0.5, the classifier will classify to 0 and otherwise to 1. Dropout layers are also added between the aforementioned layers; these will be discussed in section 3.3. In Figure 5, a schematic view of all the layers is represented.

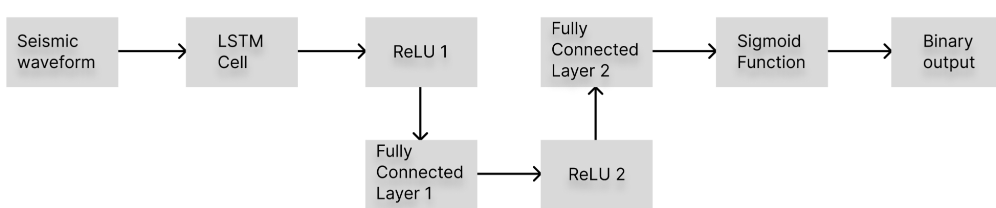

Figure 5: Schematic overview of the layers in the LSTM neural network

In order to test and train the model, 6000 samples are extracted from the dataset. These samples are randomly taken from the dataset built from earthquakes from 2007 until 2011. The data is split so that a part of the samples is used for the model's training, and the other part validates whether the model performs well on unseen data. In Figure 6 and Figure 7, the accuracy and loss are plotted for training and validation data, with the H-value equal to 70. Note that the model is overfitting; thus, the model performs better on the training set than on the validation set. Figure 8 shows the F1-score, which is retrieved using the same data. The following hyperparameters are used when running the model: batch sizes of 50, a down-sample rate of 2, a learning rate of 0.001 and the number of epochs equal to 100. The Binary Cross-Entropy loss function was used because the model is a binary classifier [18].

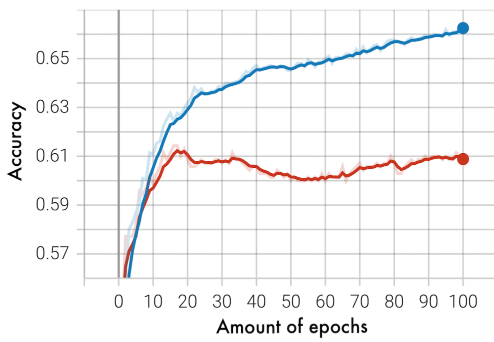

Figure 6: Accuracy of model

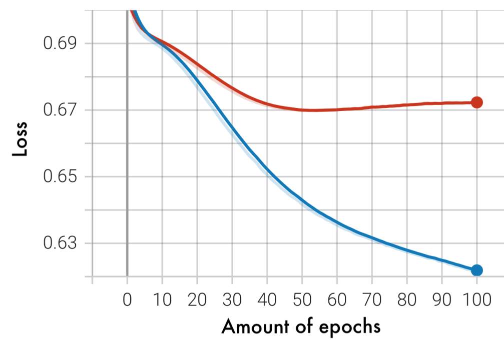

Figure 7: Loss of model

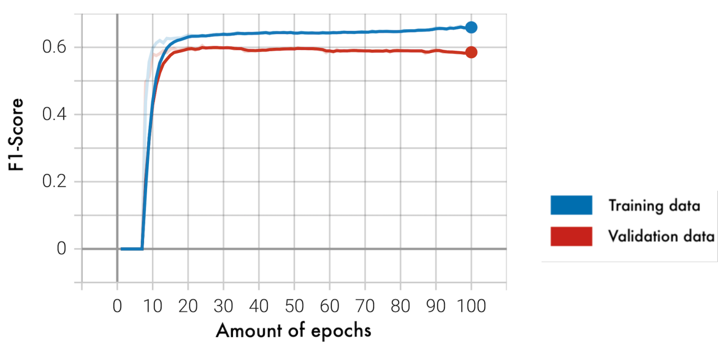

Figure 8: F1-Score of model

3.3 Overfitting

Like many deep learning models, the model used in this research suffers from overfitting. The dropout method is applied to combat overfitting. Dropout is a regularization method that approximates the training of many neural networks with different architectures in parallel [19]. Dropout layers can be placed anywhere between the other layers. The decision is made to put a dropout layer after every fully connected layer, as shown in Figure 9. Every value between 0 and 1 with a step of 0.1 has been tested for the dropout to achieve the highest performance possible. The dropout value of 0.1 resulted in the highest performance and is therefore used for the remainder of this research.

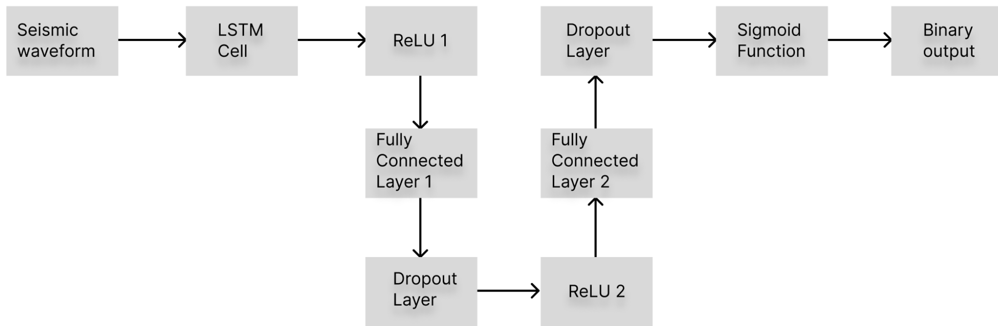

Figure 9: Schematic overview of the layers in the LSTM neural network, including dropout layers

4 Results

This section reports the final results of the appliance of different values for H. In order to train and test the model, a dataset is built from earthquakes from 2007 until 2011. From this dataset, 6000 samples are randomly picked. After preprocessing, a remainder of 3500 samples was left.

The value of H will be iteratively increased by ten seconds to perceive how the model behaves. This procedure is visualised for one seismic waveform in Figure 10.

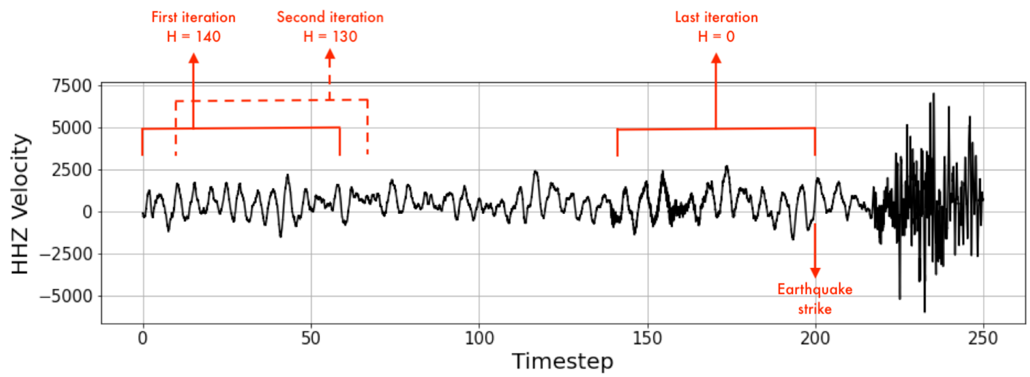

Figure 10: Three iterations to find optimised value of H of a seismic wave from one station

For each value of H, the model has been run ten times. On average, one in twenty iterations returned an outcome where all predictions were either 0 or 1. These outcomes are skipped and not considered for the average accuracy and F1-Score calculation. The accuracy and F1-Score of the validation data of the ten iterations are averaged and displayed in Table 4, Figure 11 and Figure 12.

H-value	0	10	20	30	40	50	60	70	80	90	100
Accuracy	0,57	0,57	0,58	0,55	0,59	0,61	0,58	0,64	0,61	0,60	0,57
F1-Score	0,56	0,57	0,61	0,58	0,57	0,61	0,58	0,63	0,61	0,58	0,56

H-value	110	120	130	140	150	160	170	180	190	200
Accuracy	0,58	0,59	0,58	0,61	0,61	0,61	0,59	0,61	0,59	0,59
F1-Score	0,53	0,55	0,54	0,60	0,60	0,62	0,59	0,59	0,58	0,59

Table 2: Accuracy and F1-Scores of averaged iterations

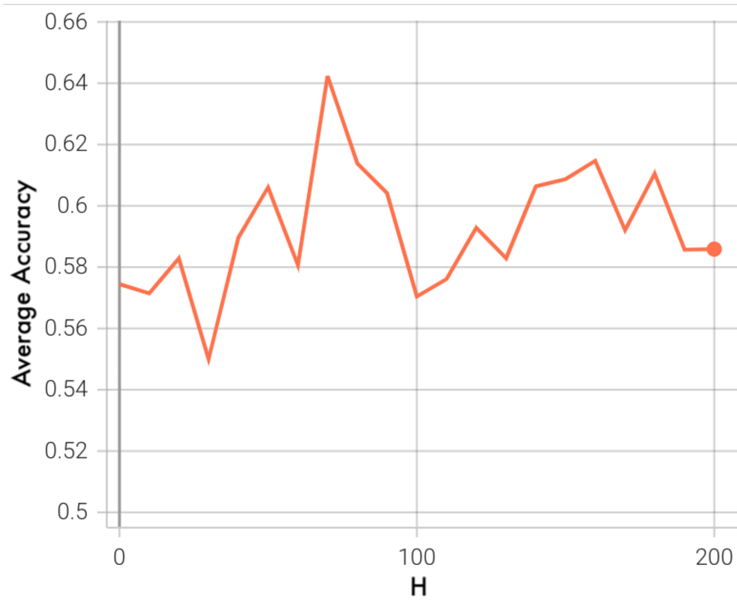

Figure 11: Accuracy plotted against H-value

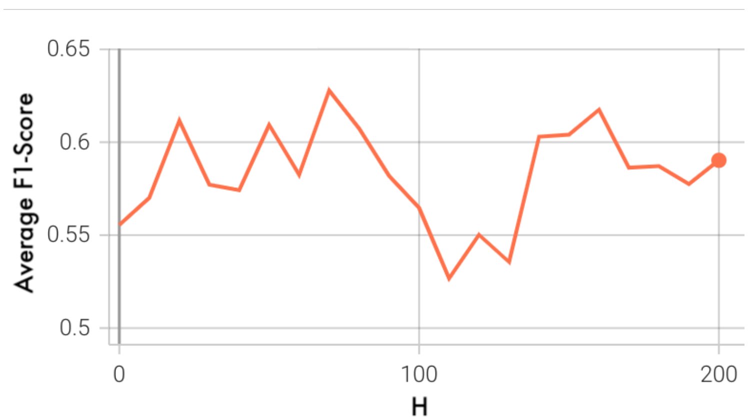

Figure 12: F1-Score plotted against H-value

Several main points can be concluded from the table and the graphs. Firstly, the model does not perform very well at low values of H. The model performs with an accuracy and F1-Score of 0,55 - 0,62 for H = 0 until H = 40. When H is raised further than 40, the performance goes up, peaking at H = 70. When H increases even more than 70, the performance reduces again, and eventually, it will fluctuate around an accuracy value of 0.6 and an F1-Score slightly lower than 0.6.

The precision and recall of the earthquake predictions are visualised in Figure 13. The values are retrieved in the same way as aforementioned. In addition, the recall and precision based on no-earthquake predictions graphs are visualised in Figure 14. The four charts combined can explain what is precisely the cause of the trend of the accuracy and the F1-Score.

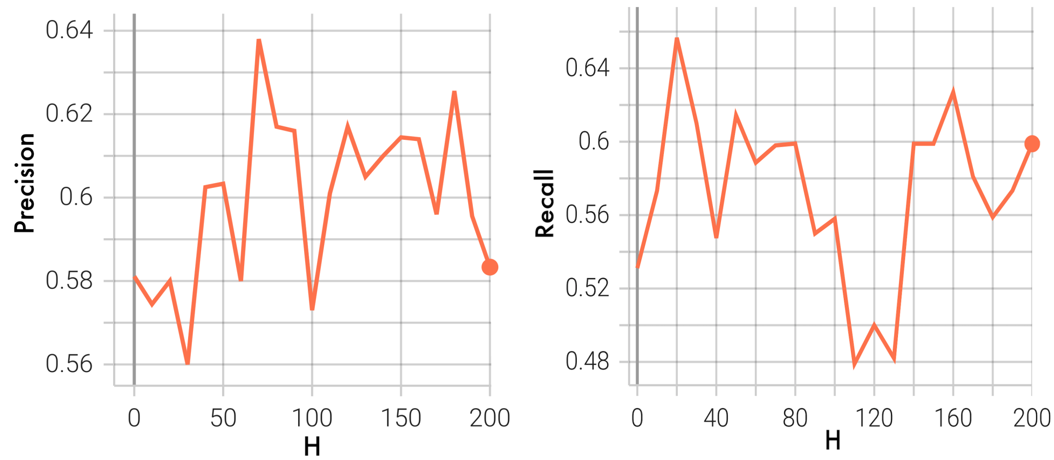

Figure 13: Precision and recall for earthquake predictions plotted against H-value

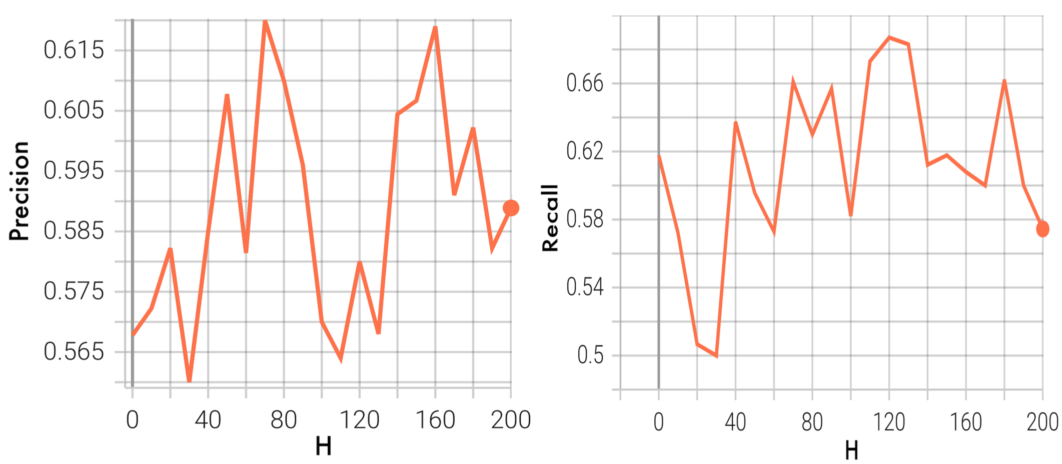

Figure 14: Precision and recall for no-earthquake predictions plotted against H-value

5 Responsible Research

The most significant responsibility during this research is the matter of reproducibility and repeatability. In the Methodology and Implementation section, every step and design choice has been explained and motivated. By clearly explaining every step, everyone with a bit of background in deep learning can implement a deep learning model and follow the same steps that have been taken in this paper. This transparency allows other researchers to adopt the methods in this paper and improve upon them. In addition, the data used in this paper is from the FDSN dataset. This dataset is open-source, so everyone is able to access it for free.

If other researchers consider adopting the methods explained in this paper, they should consider that the results are not guaranteed to work on every other dataset. The FDSN dataset was used during this research, and the methods were not tested on any different dataset.

Besides reproducibility and repeatability, there is also a critical ethical concern. When one implements the solutions described in this paper and tries to predict earthquakes in the real world, one has to be absolutely sure that the software can predict earthquakes correctly. As mentioned in Section 2.2, the price of false negatives, where an earthquake is happening but is not predicted, can be insanely high. The software and responsible implementers will be blamed for this.

6 Discussion

In the following section, the results of the research will be discussed. Possible flaws and particularities will be mentioned. The hypothesis was that the higher the value of H is, the lower the accuracy would become. This was not the case; the performance is lower for smaller values of H. At higher values of H, the performance stabilises. This could be the case for multiple reasons.

1. The dataset contains unexpected patterns. It could be the case that the deep learning model recognises other patterns within the data than whether or not a seismic waveform is an earthquake. As an example, consider the following case: flat earthquake seismic waveforms are retrieved 2000 seconds before the second consecutive earthquake. It could be the case that the model can 'see' that an earthquake will occur based on an event that always happens 2000 seconds before an earthquake. This means that the model is learning based on the 2000 seconds in advance of an earthquake. This could be independent of H's value, and therefore, the effect of changing H is less interpreted.

2. There are different forms of data in the dataset. The performance values are different when running the model on earthquakes from other years. For example, when running the same data using earthquakes from 2021 until 2022, the accuracy was at a maximum of 55 percent. This might indicate that the form of the data within the FDSN dataset is changing throughout the years, which can be very hard for a model to detect. Because the model cannot learn the data correctly, it might result in the value of H having less influence and, therefore, not wholly being able to capture the trend of the performance when changing H-values.

3. It could be the case that the trend described in the Results section is correct.

The precision and recall for earthquake and no-earthquake predictions make it noticeable what is the actual cause for the performance to increase. The peak of the performance resides at an H-value of 70. What strikes is that at H = 70, the precision value for earthquake predictions and recall value for no-earthquake predictions are high. The fact is that a false positive in earthquake predictions is the same as a false negative prediction in no-earthquake predictions. Thus, the amount of false positives in earthquake predictions is decreasing, and therefore the amount of false negatives in no-earthquake predictions is also decreasing. This is the reason for the performance increase at H = 70. Since the performance increase is primarily due to the decrease of false positives, there might be room for more performance by getting to increase the number of true positives.

7 Conclusion and Future Work

The following section concludes the research by summarizing the findings and what is learnt from them.

As seen in the result section, the value of T negatively influences the performance of deep learning models when it is close to 0. The model reaches an accuracy and F1-score of 0,55 - 0,62 for H = 0 until H = 40. When the value of H rises further, the performance increases further, with a peak at H = 70. When the value of H is increased even further than 70, the performance is reduced, and the performance fluctuates eventually around 0.6 for the accuracy and slightly lower than 0.6 for the F1-Score. Thus, if only performance is considered the most important asset, an H-value of 70 should be picked for short-term earthquake predictions. If a longer prediction time is also considered an important asset, the most obvious H-value would be 200, the biggest value that is considered during this research.

As explained in the Discussion section, when the performance is at its peak, the precision for earthquake predictions is high, and the recall value for no-earthquake predictions is high. Thus, the conclusion can be drawn that for the optimal value of H, the decrease of false positives concerning earthquake predictions is causing an increase in performance.

As explained earlier in the Discussion section, the deep learning model performed non-identical on different subsets of the dataset. Therefore, the results shown in this research can not be held as completely reliable.

Regarding future work, it would be of interest to achieve higher performances. A binary classifier with a maximum accuracy of 0.64 is very poor. An interesting first step would be determining whether the dataset contains unexpected patterns, as explained in the Discussion section. In addition, it could be of importance to figure out why the model is performing differently on different datasets within the dataset, as described in the Discussion section.

A Appendices

A.1

Station	Longitude	Latitude	Site name
BFZ	176.246245	-40.679647	Birch Farm
BKZ	176.492544	-39.165666	Black Stump Farm
DCZ	167.153533	-45.464713	Deep Cove
DSZ	171.804614	-41.744961	Denniston North
EAZ	169.308253	-45.231053	Earnscleugh
HIZ	174.855686	-38.512929	Hauiti
JCZ	168.785473	-44.073210	Jackson Bay
KHZ	173.538970	-42.415980	Kahutara
KNZ	177.673669	-39.021755	Kokohu
KUZ	175.720873	-36.745229	Kuaotunu
LBZ	170.184420	-44.385553	Lake Benmore
LTZ	172.271035	-42.781667	Lake Taylor Station
MLZ	168.118407	-45.366544	Mavora Lakes
MQZ	172.653766	-43.706082	McQueen's Valley
MRZ	175.578527	-40.660545	Mangatainoka River
MSZ	167.926399	-44.673334	Milford Sound
MWZ	177.527779	-38.334001	Matawai
MXZ	178.306631	-37.562259	Matakaoa Point
NNZ	173.379477	-41.217103	Nelson
ODZ	170.644622	-45.043982	Otahua Downs
OPRZ	176.554929	-37.844300	Ohinepanea
OUZ	173.596133	-35.219689	Omahuta
PUZ	178.257209	-38.071548	Puketiti
PXZ	176.862145	-40.030644	Pawanui
QRZ	172.529148	-40.825522	Quartz Range
RPZ	171.053865	-43.714608	Rata Peaks
SYZ	169.138823	46.536890	Scrubby Hill
THZ	172.905218	-41.762474	Top House
TOZ	175.501847	-37.730956	Tahuroa Road
TSZ	175.961124	-40.058553	Takapari Road
TUZ	169.631143	-45.953975	Tuapeka
URZ	177.110894	-38.259249	Urewera
VRZ	174.758453	-39.124341	Vera Road
WCZ	174.345043	-35.938642	Waipu Caves
WHZ	167.947031	-45.892428	Wether Hill Road
WIZ	177.189302	-37.524511	White Island
WKZ	169.017562	-44.827021	Wanaka
WVZ	170.736754	-43.074350	Waitaha Valley

Table 3: Specifications of 38 measurements stations in New Zealand

References

[1] Rabia Tehseen, Muhammad Shoaib Farooq, and Adnan Abid. “Earthquake prediction using expert systems: a systematic mapping study”. In: *Sustainability* 12.6 (2020), p. 2420.

[2] Alyona Galkina and Natalia Grafeeva. “Machine learning methods for earthquake prediction: A survey”. In: *Proceedings of the Fourth Conference on Software Engineering and Information Management (SEIM-2019), Saint Petersburg, Russia*. Vol. 13. 2019, p. 25.

[3] Qianlong Wang, Yifan Guo, Lixing Yu, et al. “Earthquake Prediction Based on Spatio-Temporal Data Mining: An LSTM Network Approach”. In: *IEEE Transactions on Emerging Topics in Computing* 8.1 (2020), pp. 148–158. DOI: `10.1109/TETC.2017.2699169`.

[4] T. Bhandarkar et al. “Earthquake trend prediction using long short-term memory RNN”. In: *International Journal of Electrical and Computer Engineering* 9.2 (Nov. 2018), p. 1304.

[5] JP Huang, XA Wang, Y Zhao, et al. “Large earthquake magnitude prediction in Taiwan based on deep learning neural network”. In: *Neural Network World* 28.2 (2018), pp. 149–160.

[6] Xiangyu Du. “Short-term Earthquake Prediction via Recurrent Neural Network Models”. MA thesis. x.du-1@student.tudelft.nl: Delft University of Technology, Jan. 2022.

[7] Francis E.H Tay and Lijuan Cao. “Application of support vector machines in financial time series forecasting”. In: *Omega* 29.4 (2001), pp. 309–317. ISSN: 0305-0483. DOI: `https://doi.org/10.1016/S0305-0483(01)00026-3`. URL: `https://www.sciencedirect.com/science/article/pii/S0305048301000263`.

[8] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. “Understanding of a convolutional neural network”. In: *2017 International Conference on Engineering and Technology (ICET)*. 2017, pp. 1–6. DOI: `10.1109/ICEngTechnol.2017.8308186`.

[9] Philip Warrick and Masun Nabhan Homsi. “Cardiac arrhythmia detection from ECG combining convolutional and long short-term memory networks”. In: *2017 Computing in Cardiology (CinC)*. 2017, pp. 1–4. DOI: `10.22489/CinC.2017.161-460`.

[10] Xuanyi Song, Yuetian Liu, Liang Xue, et al. “Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model”. In: *Journal of Petroleum Science and Engineering* 186 (2020), p. 106682. ISSN: 0920-4105. DOI: `https://doi.org/10.1016/j.petrol.2019.106682`. URL: `https://www.sciencedirect.com/science/article/pii/S0920410519311039`.

[11] GeoNet. *FDSN webservice*. URL: `https://www.geonet.org.nz/data/tools/FDSN` (visited on 04/19/2022).

[12] Michigan Tech. *How Do We Measure Earthquake Magnitude?* 1999. URL: `https://www.mtu.edu/geo/community/seismology/learn/earthquake-measure/magnitude` (visited on 05/06/2022).

[13] scikit-learn developers. *sklearn preprocessing Normalize*. URL: `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html` (visited on 05/30/2022).

[14] scikit-learn developers. *sklearn preprocessing StandardScaler*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html (visited on 05/30/2022).

[15] Michael Phi. *Illustrated guide to LSTMs and GRUs, a step-by-step explanation*. 2018. URL: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21 (visited on 06/22/2022).

[16] Jason Brownlee. *Rectified linear activation function for deep learning neural networks*. 2019. URL: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks (visited on 06/22/2022).

[17] Surya Pratap Singh. *Fully Connected Layer*. URL: https://iq.opengenus.org/fully-connected-layer (visited on 06/22/2022).

[18] Daniel Godoy. *Understanding binary cross entropy log loss*. 2018. URL: https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a (visited on 06/23/2022).

[19] Jason Brownlee. *A Gentle Introduction to Dropout for Regularizing Deep Neural Networks*. 2018. URL: https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks (visited on 05/30/2022).