SPATIO-TEMPORAL FORECASTING OF EARTHQUAKES WITH LONG SHORT-TERM MEMORY

JURRIËN BOOGERT

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

792949

COMMITTEE

Dr. Seyed Mostafa Kia
Dr. Boris Čule

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

June 9th, 2023

WORD COUNT

8798

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to Dr. Seyed Mostafa Kia for his insights and mentorship and care throughout the creation of this thesis. Additionally, I would like to extend my appreciation to my second reader, Dr. Boris Čule, for investing his time into reading my thesis and providing me valuable feedback. I am also grateful to the numerous inspiring professors I encountered this year, providing me their knowledge and inspiration. Furthermore, I would like to thank my beloved partner Jasmine Groenendijk for her unwavering support, patience, and understanding throughout this journey.

SPATIO-TEMPORAL FORECASTING OF EARTHQUAKES WITH LONG SHORT-TERM MEMORY

JURRIËN BOOGERT

Abstract

Earthquakes can potentially cause immense devastation, leading to major suffering, loss of life, and destruction. Accurately predicting earthquakes is critical to society's safety, but this remains a challenge. Numerous studies claim the feasibility of earthquake forecasting using contemporary methods and common metrics, but the accuracy of such predictions is still under question. This research aims to evaluate the possibility of short-term prediction for earthquakes with potential socio-economic impact in Japan, using historical seismic data. The study examines the influence of various factors such as timeframe lengths, and anticipated magnitudes on the predictive performance. The findings indicate that multivariate linear regression outperforms univariate linear regression for smaller magnitudes, suggesting earthquakes are spatially dependent. Both Long Short-Term Memory and Convolutional Neural Network in combination with Long Short-Term Memory models outperform univariate and multivariate linear regression and the persistence model. These models are more adept at capturing non-linear temporal and spatial dependencies across multiple time series. In addition, using longer timesframes results in better predictive performance. However, the performance of CNN-LSTM is comparable with LSTM, implying that explicitly incorporating spatial information does not increase performance. The performance of all models declines when predicting higher magnitudes. Furthermore, the practical utility of these models is limited due to the resemblance between consecutive monthly predictions, when the cutoff magnitude decreases. To address these limitations, further research is required, potentially involving the use of additional or alternative data sources.

1 INTRODUCTION

1.1 *Earthquake Mechanism*

Earthquakes usually happen in specific areas on the surface of the Earth where the boundaries of tectonic plates, which make up the Earth's crust, meet. When the plates move, they become stuck together, building up stress and resulting in deformations in the rocks at either side of the plate boundaries. This deformation stores energy in large amounts, due to the substantial volume of rocks involved. When the fault line ruptures, the stored energy is released in a few seconds, resulting in shock waves and heat, which make up the earthquake (Perez & Thompson, 1994). The released energy travels through the Earth and, upon reaching the surface, it may cause harm to buildings, leading to their collapse and injuries or fatalities. Magnitude represents the actual physical energy release at the source, as measured through instruments. The Richter magnitude scale, developed by Charles Richter in 1936, is the oldest and most widely used scale to measure magnitude and will therefore be used in this study.

1.2 *Relevance*

Earthquakes can potentially be incredibly destructive that occur suddenly, leaving little time for people to react, resulting in serious injuries, death (Pretto et al., 1993), and damage to buildings and infrastructure which in turn could lead to excessive economic losses (Mieler & Mitrani-Reiser, 2018). More than a million deaths are worldwide caused by earthquakes in the past 40 years (Coburn & Spence, 2003). Accurately predicting earthquakes is critical to society's safety, but it is still a challenging issue for seismologists and other researchers (Sobolev, 2015). Although humans cannot prevent earthquakes from happening, timely prediction and safety measures can prevent human suffering and economic loss. Earthquake prediction can be categorized into three main groups, using different methodologies and techniques, which include: 1) mathematical and geophysical analysis, 2) investigating precursors, and 3) machine learning algorithms. The combination of increasingly available amounts of data, incremental computational power and state-of-the-art deep learning algorithms, provide a breeding ground from which researchers and data scientists in particular can approach existing unanswered data-scientific questions, like forecasting earthquakes, with contemporary measures. In the following subsections different approaches will be shortly discussed, which give an overview of the active and broad field of research.

Since it remains a challenge forecasting earthquakes in the near future, still, more research is needed. This study focuses therefore on the short-term prediction of earthquakes. Since earthquake occurrence processes are considered stochastic and non-linear, recent studies mainly focus on the use of neural networks (Galkina & Grafeeva, 2019). To test whether non-linear models perform better than linear models as suggested by the assumption earthquakes are non-linear, this study uses linear regression as a baseline, independently invented by several mathematicians including Adrien-Marie Legendre in 1805, and Sir Francis Galton in the 19th century.

It is assumed seismic activity in one area will impact other locations due to the interconnections of the Earth, and that seismic activity follows specific patterns over time (Kannan, 2014; Wang et al., 2020). Research has been done on spatio-temporal forecasting of earthquakes with various algorithms, from which Recurrent Neural Networks (RNN) (Jordan, 1997; Rumelhart et al., 1985) are well known to be able to model temporal dependencies in time-series data. Long Short-Term Memory (LSTM), first introduced by Hochreiter and Schmidhuber (1997) can incorporate multiple variables and is able to model spatio-temporal data. Various studies made use of LSTM, but none contrasted their findings with linear models, as can be read in Section 2.

An important, but often overlooked fact, is the practical value of the predictions. Each year, more than a million earthquakes occur. That is roughly two earthquakes every minute (Hays, 1990). Most of these earthquakes are of small magnitude, are non-destructive, and do not entail the need to be predicted. It is easy to be accurate when predicting e.g. 'tomorrow there will be an earthquake somewhere on our globe'. According to the United States Geological Survey (USGS) "predictions are so general that there will always be an earthquake that fits". According to a relatively old study, Perez and Thompson (1994) predicted there is a 60% chance of a 7.5 or stronger magnitude earthquake happening on the San Andreas fault in Southern California in the next 30 years, which is a long time in regards to preparation for the general public to act upon and has no practical value in the short term.

This study aims to predict where earthquakes with destructive magnitudes of at least $M4.5$ or $M6$ will happen in the area of Japan, and evaluating the results of linear and non-linear algorithms on metrics as precision, recall, F1-score, Area Under the Curve (AUC) score and area under the Precision-Recall Curve (AUPRC), using various time-frames for this binary classification task. Models using temporal and spatio-temporal information are contrasted and the performance is compared. It appears, there does not exist research on forecasting earthquakes with linear regression, LSTM and CNN-LSTM for the area of Japan – for which data is

collected – with varying timeframes, magnitudes and their implications. Japan is chosen as the area of interest, primarily based on lack of research and secondarily on data availability. Therefore the resulting main research question that this study aims to answer is:

> *To what extent can earthquakes, causing socio-economic damage in the area of Japan, be forecasted short term using historical seismic data?*

In order to answer the research question the following sub-questions are formulated:

SQ1 *To what extent does linear regression using only temporal characteristics perform in forecasting earthquakes?*

SQ2 *To what extent does linear regression using spatio-temporal information perform in forecasting earthquakes?*

SQ3 *To what extent does LSTM using spatio-temporal information influence the performance in forecasting earthquakes?*

SQ4 *To what extent does CNN-LSTM using explicit spatio-temporal information influence the performance in forecasting earthquakes?*

SQ5 *How do various time-frames and cutoff magnitudes influence the performance of the models in forecasting earthquakes?*

2 RELATED WORK

In the pursuit of understanding and predicting earthquakes, various methodologies have been proposed in literature. This section critically reviews the current state of the art in Earthquake Early Warning (EEW), classical methods in earthquake forecasting, and recent advancements in the use of machine learning. The research questions and resulting methodological choices used in this thesis are motivated by the mentioned strengths and limitations of the reviewed literature in the following subsections.

2.1 *Earthquake Early Warning*

The methods mentioned in Section 1.2 can be categorized into two groups: predicting the earthquake rupture before it happens and quickly detecting it after it takes place in order to warn users so that precautionary measures can be taken before major ground motion has initiated at a specific location

(R. M. Allen & Melgar, 2019). The latter is the field of Earthquake Early Warning (EEW). EEW is an alert system that provides users with a binary message of an earthquake happening or not within a matter of seconds. This method is relatively accurate. Though, the time to take precautionary measures is limited. This inspired the main research question. Therefore, this thesis aims to predict further into the future (4 weeks), to provide more time for preparation in order to mitigate the destructive consequences, but not to far into the future, which makes it to generic without practical value.

As most people cannot differentiate between magnitude and intensity, adding this information would only confuse. When a specific threshold is set for a location, alerts are labeled as true or false positives or true or false negatives, depending on whether the alert threshold was accurately or inaccurately predicted to be exceeded or not. Research done by R. Allen (2017) and R. Allen et al. (2018) implies that people are more tolerant of false positives than of false negatives in regards to earthquakes.

The goal of this thesis is to provide a warning system for public use as well, and therefore, the method uses a binary response – if an earthquake with a certain magnitude will happen or not –, and at the same time demonstrating greater tolerance for false positives. This partly results in the fifth sub-question ($SQ5$).

2.2 *Classical methods in earthquake forecasting*

Classical methods in earthquake forecasting – before rupture has taken place – rely on various factors such as geographic deformation, tectonic shifts, seismic patterns, variations in seismic wave speeds across regions, and occurrences of geomagnetic and geo-electric phenomena (Rikitake, 1968). Statistical distributions of cycles are established and used for inference in predicting probabilities long- and medium-term, but predicting short-term occurrences remains a challenge (Rundle et al., 2021; Sobolev, 2015). Therefore, short-term prediction is the focus of this thesis. Kannan (2014) uses a mathematical model using Poisson's distributions in validating spatial connection theory, which assumes earthquakes are related to historical occurrences within a fault zone. Another category in earthquake prediction involves various precursors, like changes in electric and magnetic fields, emission of radon gas, water level, temperature and surface deformation as reported by Cicerone et al. (2009). The authors find that only water level and certain gas emissions show correlations with earthquakes and GPS data can be promising in detecting surface deformation, but more data needs to be gathered first. Deviating animal behavior has been studied as well, but has not been documented scientifically in a quantitative way (Hayakawa et al., 2016).

Classical methods tend to rely on parametric models that assume a specific statistical distribution. They often are based on assumptions, such as linearity, while seismic activities are often non-linear and can involve complex interactions. The high-dimensional nature of geophysical data further strains these conventional techniques, as they lack the capabilities to efficiently process and interpret such complex, multidimensional information. Moreover, classical techniques often struggle with processing large datasets, while machine learning is inherently capable of using big data, making more suitable for large-scale forecasting of earthquakes. Though, Mignan and Broccardo (2019) and Waheed et al. (2020) show that a relatively simple model like logistic regression can be more successful in predicting earthquakes with carefully selected features in contrast with deep learning models. Therefore, this thesis uses linear regression as a basemodel to compare and contrast more contemporary models which results partly in the first and second sub-question ($SQ1$, $SQ2$).

2.3 *Machine learning*

Following on from the previous section, machine learning in predicting earthquakes is an active research field and in this section a recent selection of relevant papers is discussed using these contemporary methods.

Asencio–Cortés et al. (2018) compared four different regressors and their ensembles (Generalized Linear Models, Gradient Boosting Machines, Deep Learning and Random Forest). Random forest achieved the lowest error. Interestingly, their deep learning model – which is not described in detail – had the highest error of all. The authors assign this result due to high parameterization. Because the authors suggest deep learning models overfit and perform worse than the other machine learning models, in this thesis is chosen to compare the relatively simpler linear regression model with a more sophisticated deep learning model and evaluated this claim.

Asim et al. (2017) tested four machine learning algorithms on temporal characteristics of seismic events in binary classification of minimal one $M \geq 5.5$ earthquake happening in a, by faults defined area of the Hindukush region in one month. They chose this area to maintain geological uniformity and facilitate meaningful analysis of correlations between earthquake occurrences and derived seismic features. Eight different seismic features are constructed by transforming predefined sequences of magnitudes mathematically, which is a remarkable choice when using deep learning as mentioned hereafter, since these models are capable of extracting features themselves. Therefore, in this thesis no additional features are engineered to make a fair comparison between linear regression and LSTM models.

Data comes from the catalog from the Center for Earthquake Studies and Hindukush and United States Geological Survey. They compare Pattern Recognition Neural Network (PRNN), which is a simple fully connected Multi-Layer Perceptron (Rosenblatt, 1958), a Recurrent Neural Network (RNN) (Jordan, 1997; Rumelhart et al., 1985), Random Forest (Breiman, 2001), and lastly Linear Programming Boosting (LPBoost) (Demiriz et al., 2002), which makes use of an ensemble of trees and maximizes the area between samples belonging to different classes. The classifiers showed varied performance. LPBoost is more sensitive, while PRNN has the least occurrence of false alarms. Asim et al. conclude that despite being a seemingly random and nonlinear event, the study demonstrates that earthquake events can be modeled based on the geological features of the seismic region. Their method motivates the choice in this thesis to include an connecting geological area defined by active faults, Japan (see main research question).

Wang et al. (2020) assume that seismic events in one area can cause similar events in other areas due to the interconnected nature of the earth, and that seismic activity tends to follow certain patterns over time. This assumption inspired the formulating of the distinction between temporal and spatio-temporal in $SQ1$ and $SQ2$. They chose an LSTM, because it is capable of learning relations between data over a longer time interval. They used earthquake data from mainland China starting 2006 to 2016 with magnitude $M > 2.5$ coming from US Geological Survey (USGS). The time window is set to one month. The lookback-window set to 1 and 10. A temporal-only model yields an overall accuracy of 63.50 percent with true positive and true negative scores of respectively 46.83% and 79.6%. For the spatio-temporal model, they use data from $M > 4.5$ earthquakes (1966-2016) and a 12-month lookback-window. This model achieves an overall accuracy of 74.81%, with true positive and true negative scores of 68.56% and 81.31% respectively. They conclude that "spatio-temporal data correlations do provide better prediction results than only mining temporal data correlations". However, their comparison of different data and magnitudes is a flaw in their study design and therefore, this thesis aims to compare temporal and spatio-temporal models on the same dataset, keeping all conditions constant except for the chosen model, in order to address any differences in performance attributable to the model itself.

Fox et al. (2022) compare four deep learning algorithms used for predicting earthquakes. Earthquake data from the USGS covers Southern California, with events recorded from 1950 to 2022. LSTM (two-layered), Temporal Fusion Transformer (TFT, two distinct LSTM's one for encoder and one for decoder with and attention-based transformer) (Lim et al.,

2021), Science Transformer (ST, using two LSTM's for the encoder and a space-time transformer for the decoder) built by the University of Virginia and AE-TCN Joint Model (combines an auto-encoder and a Temporal Convolutional Network) (Feng & Fox, 2021) are compared. They assume the geometric and geospatial structure are important characteristics of earthquakes, but they have not found the local structure from faults useful in their analysis and make use of a spatial bag where there is variation in space, but it is not necessarily related to the structural distance between regions. In order to test their hypothesis, in this thesis CNN-LSTM is used, to evaluate if the explicit absolute location of seismic events results in better performance than a so called spatial bag with LSTM, inspiring the formulation of the third and fourth sub-question ($SQ3$, $SQ4$).

Jipan et al. (2018) propose a Convolutional Neural Network (CNN) (LeCun et al., 1998). They aim to predict if an $M \geq 6.0$ earthquake will happen in Taiwan, 30 days into the future. The data was gathered from China Seismic Information net ranging between 1970 and 2016. A pretrained CNN model on the Cifar10 dataset was used. The authors hypothesize that destructive earthquakes require a specific preparation period. If the lookback time is too short, the model lacks sufficient data for accurate predictions; conversely, if the window is too large, irrelevant information may hinder the model, leading to inaccurate predictions. This idea is contradictory with the consensus that more data generally results in better performance. Therefore, in this thesis the influence of various lookback-windows is evaluated. This results partly in the fifth sub-question ($SQ5$); it is hypothesized, longer lookback-windows result in better performance.

Nicolis et al. (2021) investigate the use of LSTM in combination with CNN for forecasting the intensity and probability an earthquake with magnitudes of $M \geq 4.0$, $M \geq 5.0$, or $M \geq 6.0$ will happen in a Chile. Using the Chile seismic catalog from 2000-2017, they analyze 86,000 events from the National Seismological Center, recorded over 6,575 days on a 1×1 degree grid. They use Epidemic-Type Aftershock Sequences (ETAS) first introduced by Ogata (1988, 1998) for the intensity estimation per day per grid-pixel. The CNN predicts the location while the LSTM predicts the maximum intensity derived from ETAS. For $M \geq 6.0$ earthquakes, the LSTM outperforms the classical temporal ETAS with an R^2 of 0.66 on the test set. However, the CNN's accuracy is only 45%. Despite this, the authors argue that the model can identify time frames with a heightened likelihood of significant earthquakes. For $M \geq 4.0$, the LSTM's performance declines, with an R^2 of 0.45 on the test set, due to the increased number of events to process resulting in difficulty identifying patterns, thus Nicolis et al. Conversely, the CNN's performance increases to 80%.

This claim contradicts the consensus that more training samples result in better performance. Therefore, this thesis compares the performance of a lower and higher magnitude (see $SQ5$) which results in relatively more and less training samples. The hypothesis is that predicting earthquakes with higher magnitudes will result in more inaccurate predictions and lower performing models.

Both classical methodologies and machine learning techniques have their respective strengths, but they often fall short in accurately predicting short-term seismic activity. Various regions have been studied, but none used Japan as the area of interest in conjunction with linear regression, LSTM and CNN-LSTM. In addition, none explicitly focused on predicting destructive magnitudes with socio-economic consequences. Furthermore, these studies have raised questions about the impact of different time-frames, the choice of cutoff magnitudes, and the benefits of spatio-temporal elements in the predictive models. The subsequent sections of this thesis will systematically explore these factors.

3 METHOD

This section outlines the various components of the process used in this study. It begins by discussing the data sources. Next, it provides a description of the baseline models (persistence and linear regression), followed by a technical explanation of the LSTM and CNN-LSTM architectures. Then, pre-processing and feature engineering steps applied in preparation to be used by the different algorithms, are discussed. Hereafter the evaluation metrics, methods for model comparison, and experimental conditions are described. Finally, the used programming language and frameworks are mentioned.

3.1 *Dataset Description*

Two publicly accessible datasets will be used in this study. The main dataset originates from the online catalog maintained by USGS (US Geological Survey). Data include time, magnitude, latitude, longitude and depth of the hypocenter of in total 69446 earthquakes. The minimum and maximum registered magnitudes are 1.6 and 9.1 respectively with a mean of 4.53 and median of 4.50. The depth ranges from 0m to 686m with a mean of 80.71m and median of 35m. The area of interest is Japan with 50 degrees of latitude (10 to 60N) and 40 degrees of longitude (134 to 174E) between 1973 and 2022, since this time period has a natural looking spread of events and relatively few irregularities based on face value (see Figure 1,2).

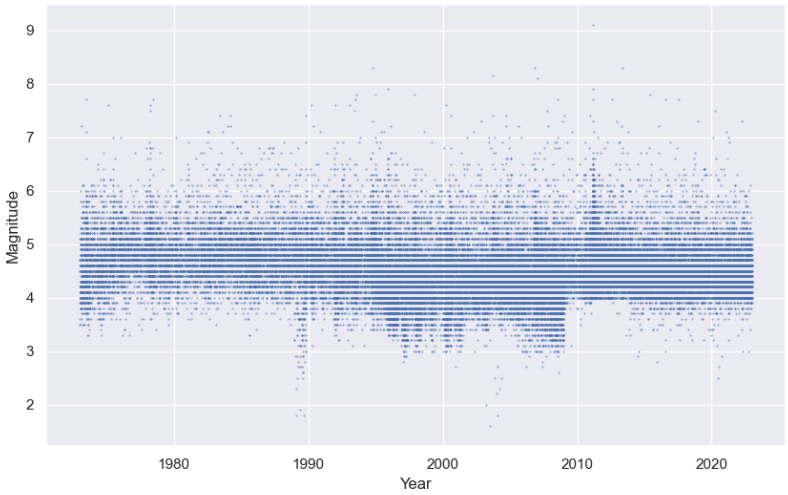
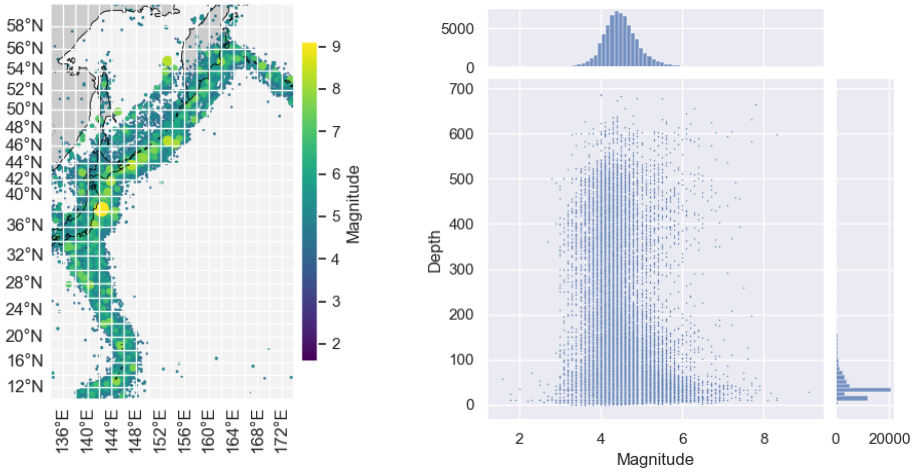


Figure 1: Earthquakes over time within 10 to 60N latitude and 134 to 174E longitude.

The second dataset comes from National Geophysical Data Center and World Data Service (NGDC/WDS) and includes data on approximately 6350 destructive earthquakes that occurred between 2150 B.C. and the present, which meet certain criteria such as causing economical damage ($1 million or more), resulting in 10 or more fatalities, having a magnitude $M7.5$ or greater, causing a Modified Mercalli Intensity X or greater, or generating a tsunami. This dataset is used to define the destructive cut-off magnitude. Figure 14 in the appendix (page 45) shows six categories of socio-economic consequences for the defined area of Japan. Since the total affected increases exponentially with increasing magnitude, a logarithmic scale is used. The smallest magnitude resulting in a positive count for at least one category is of $M4.5$. This magnitude will be used as the minimum value for which the algorithms are evaluated together with a magnitude $M6$. Since higher magnitudes overall result in more severe consequences it is of increasing societal importance to forecast these higher magnitudes with accuracy. Note that the figure suggests there may be missing data, i.e there is a discrepancy between the amounts for categories at $M9$. The data or data controller does not provide an answer. Nevertheless, this notability does not have influence the analysis negatively, since the goal is to define the smallest possible destructive magnitude for any category.



(a) Earthquakes in Japan between 1973 and 2023

(b) Distribution of Depth and Magnitude

Figure 2: Seismic activity in Japan: Geographic and depth/magnitude distribution.

3.2 *Algorithms*

3.2.1 *Persistance Model*

The persistence model can serve as a useful baseline for evaluating the performance of more sophisticated models such as linear regression, LSTM, and CNN-LSTM. In time-series prediction, the persistence model's primary assumption is that the value at the next time step will be the same as the current value. While this approach may appear overly simplistic, it can be effective for certain types of time-series data, especially when there is a strong temporal correlation of one lag.

The persistence model can be mathematically represented as:

$$\hat{y}_{t+1} = y_t \tag{1}$$

where $\hat{y}_{t+1}$ denotes the predicted value at time step $t+1$, and y_t represents the observed value at the current time step t.

3.2.2 *Linear Regression*

Linear regression is a classical fundamental statistical method for modeling the relationship between a dependent variable and one or more independent variables. It uses a linear equation that minimizes the sum of the squared errors between the predicted values and the actual observed values. The method of least squares was first published by Adrien-Marie Legendre in 1805, and by Carl Friedrich Gauss in 1809. As mentioned in Section 1.2,

to test whether non-linear models perform better than linear models as suggested by the assumption earthquakes are non-linear, this study uses linear regression as a baseline. One could argue logistic regression is better suited for a binary classification task, but due to event extraction into a raster form, as can be read in Section 3.3.2, there are multiple samples engineered where there exists only one class. Logistic regression can not be trained on a labeled dataset with samples belonging to only one class. Additionally, the continuous output from linear regression can be converted to a binary target easily by defining a boundary value as is done in this study. Consciously is chosen not to use a Generalized Linear Model like Asencio–Cortés et al. (2018) did in their study, because this model can handle a non-linear relationship between input and output. Linear regression allows quantifying the potential improvement that deep learning models can offer. The linear equation for a multiple linear regression model with n independent variables can be expressed as:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{2}$$

In this equation, $\hat{y}$ is the predicted value of the dependent variable, and $x_1, x_2, \ldots, x_n$ are the independent variables or features. The coefficients $\beta_0, \beta_1, \ldots, \beta_n$ represent the model parameters, where β_0 is the intercept term and $\beta_1, \ldots, \beta_n$ are the weights associated with the independent variables. The goal of linear regression is to find the optimal values for these coefficients, which minimize the error between the predicted and observed values.

3.2.3 *Long Short-Term Memory*

A Simple Recurrent Neural Network (SimpleRNN), explicitly mentioned for the first time by Jordan (1997) and Rumelhart et al. (1985), is a basic type of recurrent neural network (RNN) designed for processing sequential data. It can capture temporal dependencies within the data by maintaining a hidden state that is updated at each time step. Since earthquakes happen to be sequential in nature, it seems logical to choose for a RNN in predicting these phenomena. Long Short-Term Memory (LSTM) networks, first introduced by Hochreiter and Schmidhuber (1997), are a type of RNN specifically designed to handle the vanishing gradient problem that arises in traditional RNNs. As a result, LSTM's are potentially able to capture longer dependencies than more traditional RNNs and since this study aims to evaluate the effect of varying (longer) timeframes LSTMs seem suitable.

LSTMs make use of a gating mechanism composed of input, forget, and output gates, which work in tandem with the cell state to control the

flow of information (see Figure 3). The following equations describe the computations within an LSTM cell:

- **Input gate** (i_t): Determines which information will be added to the cell state. It is governed by the equation:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{3}$$

- **Forget gate** (f_t): Regulates the information that will be discarded from the cell state. It is computed using:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{4}$$

- **Output gate** (o_t): Controls the information from the cell state that will be utilized to update the hidden state. It is calculated as follows:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{5}$$

- **Cell input** (g_t): Represents the candidate information to be added to the cell state, and is given by:

$$g_t = \tanh(W_g x_t + U_g h_{t-1} + b_g) \tag{6}$$

- **Cell state update** (c_t): Updates the cell state by incorporating the input gate, forget gate, and cell input. The equation is:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{7}$$

- **Hidden state update** (h_t): Finally, the hidden state is updated using the output gate and the cell state as follows:

$$h_t = o_t \odot \tanh(c_t) \tag{8}$$

In these equations, x_t is the input vector at time step t, h_t is the hidden state, and c_t is the cell state. The weight matrices W_i, W_f, W_o, and W_g correspond to the input, forget, output, and cell input, respectively, while U_i, U_f, U_o, and U_g are the weight matrices connecting the hidden states to the input, forget, output, and cell input. The bias vectors are represented by b_i, b_f, b_o, and b_g. The sigmoid activation function is denoted by σ, and the hyperbolic tangent activation function is represented by tanh. Element-wise multiplication, or the Hadamard product, is indicated by $\odot$.

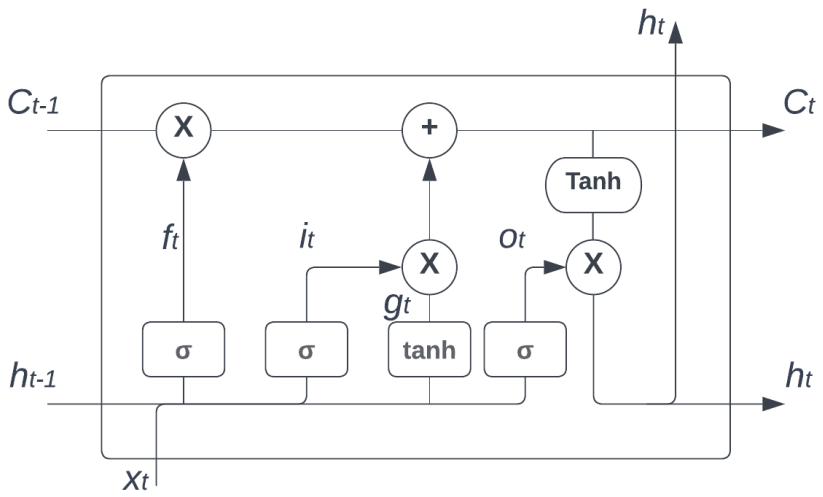


Figure 3: LSTM Cell Architecture

3.2.4 *CNN-LSTM*

Convolutional Neural Networks (CNNs) are designed to process grid-like data, such as images or video, by exploiting local spatial patterns through the use of convolutional layers, first introduced by LeCun et al. (1998). Since the data used in this study is processed and transformed into a grid-like structure or raster, as can be read in Section 3.3, the use of CNN could potentially be beneficial in extracting relative spatial information and explicitly defined longitude, latitude and depth stored in separate channels. A CNN with a 2D convolutional layer (Conv2D) is used as a feature extractor, and its output is then fed into an LSTM for further processing. This combination allows the model to benefit from the CNN's ability to capture local patterns and the LSTM's capacity to model long-range dependencies.

A Conv2D layer applies a set of filters, or kernels, to the input data. Each filter is a small 2D matrix, which is convolved with the input to produce a feature map. The filters are learned by the network during training to extract relevant features. Mathematically, the convolution operation for a single filter can be expressed as:

$$F_{i,j} = \sum_m \sum_n I_{i+m,j+n} K_{m,n} \tag{9}$$

where $F_{i,j}$ is the output feature map at position (i, j), I is the input data, and K is the filter. The indices m and n range over the dimensions of the filter.

After the Conv2D layer, an activation function, such as the Rectified Linear Unit (ReLU), is applied. The output of the activation function can be further processed by pooling layers, which downsample the feature maps while retaining important information.

Once the input data has been processed by the Conv2D layer and subsequent layers, it can be reshaped and fed into an LSTM. This combination has been used in various applications, such as video classification (Karpathy et al., 2014) and time series forecasting with multi-dimensional input data (Ge et al., 2021).

3.2.5 *Loss Function: Focal Loss*

The research goal is to predict where in a pre-defined time window in the future at least one earthquake will occur with at least a magnitude $M4.5$ or $M6$. This is a binary classification task. RNNs with binary targets typically use a sigmoid activation function in the output layer to produce predictions in the range of $(0, 1)$, and this is used in this study as well. Class imbalance could negatively influence the performance of a model, where one class significantly outnumbers the other. In this case, the standard loss function binary cross-entropy, may not be optimal, as the model might become biased towards the majority class. Focal loss is a variant of binary cross-entropy loss, designed to address extreme class imbalance by emphasizing the importance of correctly classifying the minority class.

Focal loss was introduced by Lin et al. (2017) in the context of object detection, specifically for the RetinaNet model. The focal loss function is defined as:

$$L(y, p) = -\alpha y(1-p)^{\gamma} \log p - (1-y)\alpha(1-\alpha)p^{\gamma} \log(1-p) \qquad (10)$$

In this equation, y is the true binary label, and p is the predicted probability of the positive class. The parameters α and γ are introduced to control the contribution of different examples to the loss.

The parameter α balances the importance of positive and negative examples, with typical values in the range $(0, 1)$. When $\alpha = 0.5$, both classes are equally weighted, whereas values closer to 0 or 1 emphasize the importance of negative or positive examples, respectively. The positive class is the occurrence of an earthquake. When using the cutoff magnitude of $M4.5$ or $M6$ the proportion of occurrences is 0.017 and 0.0005, respectively. The inverse of these values for α results in near perfect recall, but precision degrades. Therefore α is empirically set at 0.7 and 0.95, respectively. This results in comparable but slightly more positive predictions in the first case, which is preferred, since studies by R. Allen (2017) and R. Allen et al.

(2018) suggest that people are more tolerant of false positives than false negatives in relation to earthquakes. In the latter case it enables effective training of the LSTM and CNN-LSTM algorithm as well.

The parameter γ is a focusing parameter that adjusts the rate at which easy examples are down-weighted. As γ increases, the contribution of well-classified examples to the loss is reduced, allowing the model to focus more on hard-to-classify examples. γ in our case is empirically set at 1 and 5, respectively.

By introducing the modulating factors $(1-p)^{\gamma}$ and p^{γ}, focal loss assigns lower weights to easy examples and higher weights to hard examples, thus mitigating the impact of class imbalance on the learning process.

3.3 *Experimental Setup*

3.3.1 *Data Collection and Cleaning*

First the minimum ($10°N$, $134°W$), and maximum ($60°N$, $174°W$) degrees latitude, longitude and timeframe (January 3rd, 1973 to December 31st, 2022) are defined via an iterative process. A spatial and temporal area is selected where the data looks relatively regular based on face value. Because sensory instrumentation change over time due to innovation, changing interests and resources (Rikitake, 1968), still there are periods notable in the data where there is deviation (see Figure 1). In the selected data there was a contiguous period where there were zero values present that were clearly outliers since the dataset should only contain positive magnitude values. These zero values were simply deleted from the dataset.

3.3.2 *Feature Engineering*

An important and deliberate step in the pipeline is the transformation of the raw data to an appropriate format for the algorithms to work with and in order to answer the research questions. A visual representation of the complete pipeline is given by Figure 4.

The data is grouped by predefined sub-regions into a longitude bin, latitude bin, depth bin. Time frequency, and the maximum magnitude and mean depth within each group are computed, which is a common practice in earthquake feature engineering (Fox et al., 2022; Nicolis et al., 2021; Wang et al., 2020). Missing values are filled with zeros, since a non-existent earthquake would result in zero magnitude recording. The resulting data is reshaped into a tensor with shape (time, latitude, longitude, channels), where the first channel represents the maximum magnitude, the second channel represents the latitude values, the third channel represents the longitude values, and the fourth channel represents mean depth.

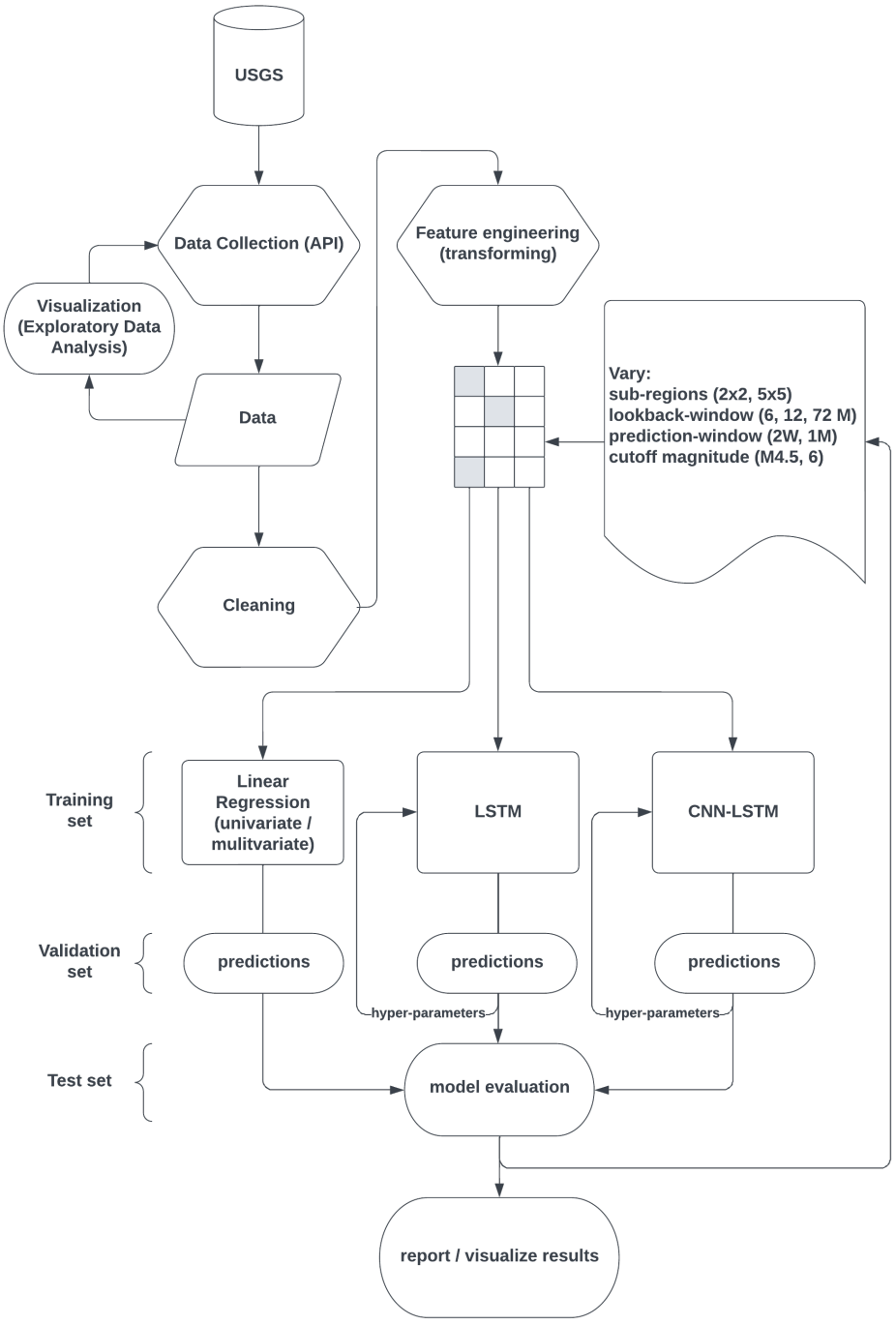


Figure 4: Methodology Flowchart

The raw data is processed into this format to treat the data as a multivariate time series, allowing it to be contrasted with a multiple univariate time series approach and evaluate the performance. In addition this formatting makes it convenient to make predictions for all sub-regions in parallel instead of predicting one at a time in sequence. The assumption is that the sub-regions have their own characteristics in producing earthquakes due to structures of the earth including faults (Carlson et al., 1994; Rundle et al., 2021), and this formatting can be used to test if areas influence the occurrences and prediction of earthquakes in other sub-regions. Individual time-series not only rely on its previous values but also has a level of dependence on other time-series, which is utilized in the prediction of future values. While linear regression and LSTM models only need the first channel containing magnitudes, a CNN-LSTM model can use all channels containing magnitude and location and therefore encodes the spatial information explicitly. These are the processed shapes summarized for the multivariate models:

$$\begin{aligned} \text{Linear Regression:} &\quad (\text{time}, \text{latitude} \times \text{longitude}) \\ \text{LSTM:} &\quad (\text{time}, \text{latitude} \times \text{longitude}) \\ \text{CNN-LSTM:} &\quad (\text{time}, \text{latitude}, \text{longitude}, \text{channels}) \end{aligned}$$

where the channels for the CNN-LSTM model are magnitude, latitude, longitude, and depth. For the multiple univariate variation the same model iterates over all combinations of latitude and longitude, i.e. sub-regions. The predictions are stored and concatenated in a loop until the predictions for every sub-region are made.

3.3.3 *Evaluation Method*

To evaluate time-series Pardo (1992) proposed a method called Walk-forward validation to address the limitations of conventional cross-validation techniques for time-series. Standard cross-validation approaches are ill-suited for time-series due to the presence of temporal correlations, which make using future data points to predict past ones irrational. Figure 5 illustrates the method used in this study. The time series is visualised as a chronological sequence of data points in the form of blocks. The data is divided into a training (70%), validation (15%) and test set (15%) and these subsets are further processed by the walk-forward methodology using a sliding window of observations (depicted in light grey) to predict the next timestep (depicted in dark grey). The sliding window contains a pre-defined number of consecutive data points and moves forward by one timestep, predicting the next timestep. This process continues until

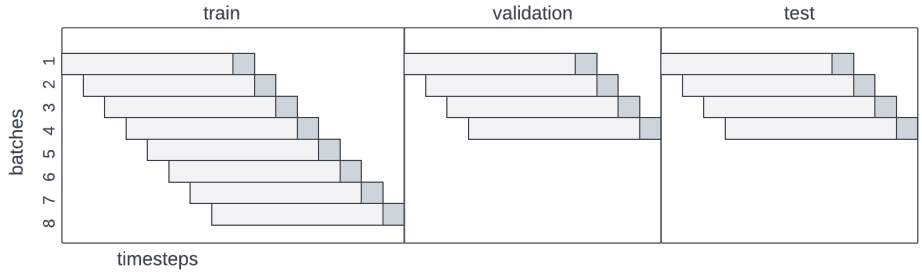


Figure 5: Walk-forward validation using a sliding window

the end of the training set is reached. The trained model is tested on the validation set and based on the performance on this set, hyperparameters are tuned. Finally the tuned model is evaluated on the test set.

The engineered data from the previous subsection needs to be further processed in order to make use of this training and validation method. The resulting input shapes are now:

$$\begin{aligned} \text{Linear Regression:} &\quad (\text{batch}, \text{timesteps} \times \text{latitude} \times \text{longitude}) \\ \text{LSTM:} &\quad (\text{batch}, \text{timesteps}, \text{latitude} \times \text{longitude}) \\ \text{CNN-LSTM:} &\quad (\text{batch}, \text{timesteps}, \text{rows}, \text{columns}, \text{channels}) \end{aligned}$$

The output shapes or targets will be:

$$\begin{aligned} \text{Linear Regression:} &\quad (\text{batch}, \text{latitude} \times \text{longitude}) \\ \text{LSTM:} &\quad (\text{batch}, \text{latitude} \times \text{longitude}) \\ \text{CNN-LSTM:} &\quad (\text{batch}, \text{rows}, \text{columns}) \end{aligned}$$

Where batch is equivalent to samples and the amount of sliding windows. Timesteps is the length of the sliding window. Since the goal is predicting only magnitude for one timestep for all sub-regions, the output shapes do not contain the timesteps and magnitude dimension explicitly.

3.3.4 *Evaluation Metrics*

Time-series prediction in the context of earthquake forecasting involves transforming the time-series data into a supervised learning problem, which can be represented as $y = f(x)$. In this representation, x denotes the historical timesteps, and y corresponds to the target variable. The target variable is a binary response, indicating the presence or absence of an earthquake with $M \geq X$, as discussed in previous sections. To evaluate the performance of an imbalanced binary classification task, this study uses

various metrics, including precision, recall, F1-score, Area Under the Curve (AUC) score and area under the Precision-Recall Curve (AUPRC).

The precision, recall, and F1-score are defined as follows:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{11}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{12}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

The Receiver Operating Characteristic (ROC) curve graphically represents a classifier's performance across various classification thresholds. The curve is plotted with the true positive rate (TPR, or sensitivity) on the vertical axis and the false positive rate (FPR, or 1-specificity) on the horizontal axis. The classifier's ability to discriminate between earthquakes $\geq$ X and $<$ X is captured by the Area Under the Curve (AUC) score. A higher AUC value indicates better discriminatory performance. The TPR and FPR can be defined as:

$$\text{True Positive Rate} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{14}$$

$$\text{False Positive Rate} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}} \tag{15}$$

Given that the majority class (0, indicating $M < \text{X}$) is relatively large, the Precision-Recall (PR) curve is used to assess the performance of the minority class (1, $M \geq \text{X}$). The curve is plotted with recall (or TPR) on the horizontal axis and precision on the vertical axis. The Area Under the Precision-Recall Curve (AUPRC) measures the classifier's performance concerning the minority class, with a higher AUPRC indicating better performance in regards to both precision and recall.

A note must be made about the evaluation of the linear regression model. To facilitate a fair comparison between the continuous target values generated by the linear regression model and the binary output of LSTM and CNN-LSTM models using the same metrics, a transformation is applied to the target values of the linear regression model. By using the same cutoff value $M4.5$ or $M6$ on the predicted magnitudes as used by the other models, the continuous output is converted into a binary format, enabling a consistent evaluation across all models.

3.3.5 *Model Comparison*

Considering the complexity and the many random processes involved in the optimization process of LSTM and CNN-LSTM, training and testing is repeated for a few (10) iterations, resulting in varying performance metrics. The Mann-Whitney U test, an extension of the Wilcoxon rank-sum test, developed by Frank Wilcoxon in 1945, is a non-parametric statistical hypothesis test that will be used to determine if there are significant differences between the sets of F1-scores of the two different models and between the scores of the same model, but trained with different conditions. The test assumes that the data are independent and not normally distributed, which can be concluded by visually inspecting the data – because of small sample size – shown in the appendix (see figure 15, page 46). Further, it makes use of an continuity correction, because the scores are assumed to come from a continuous distribution. It provides a more robust basis for model comparison than simply comparing average performance or the performance of a single run.

Boxplots are used to visually assess whether the performance of the baseline models is different from that of the deep learning models, by identifying the F1-scores as outliers.

3.4 *Experimental Conditions*

In the following two subsections the varying experimental conditions are presented.

3.4.1 *Lookback-window*

The lookback-window is varied from 6 to 12 to 72 months to measure the impact on the performance for the different models. The longer the lookback-window, the smaller the samples to be trained or tested due to a decreasing amount of sliding windows. 72 months is chosen as the maximum lookback-window, based on the size of the dataset. This number results in 19 months to be predicted for the validation and test set. The distribution of classes $(0, 1)$ for the three lookback-windows is respectively 5365 and 1435, 4979 and 1341, 1195 and 325 for the test set. The aggregation frequency to calculate the maximum magnitude is fixed at a day, since it is assumed that more information is preserved when choosing a smaller frequency which could potentially result in better performance. The size of the sub-regions is fixed at 5×5 degrees, resulting in a total of 80 sub-regions. The prediction window is set at a month, since the aim is to predict short-term and it is assumed a longer window would be of less practical value for the general public to act upon. The cutoff magnitude is

set at $M \geq 4.5$, since this is the minimal magnitude resulting in significant socio-economic consequences.

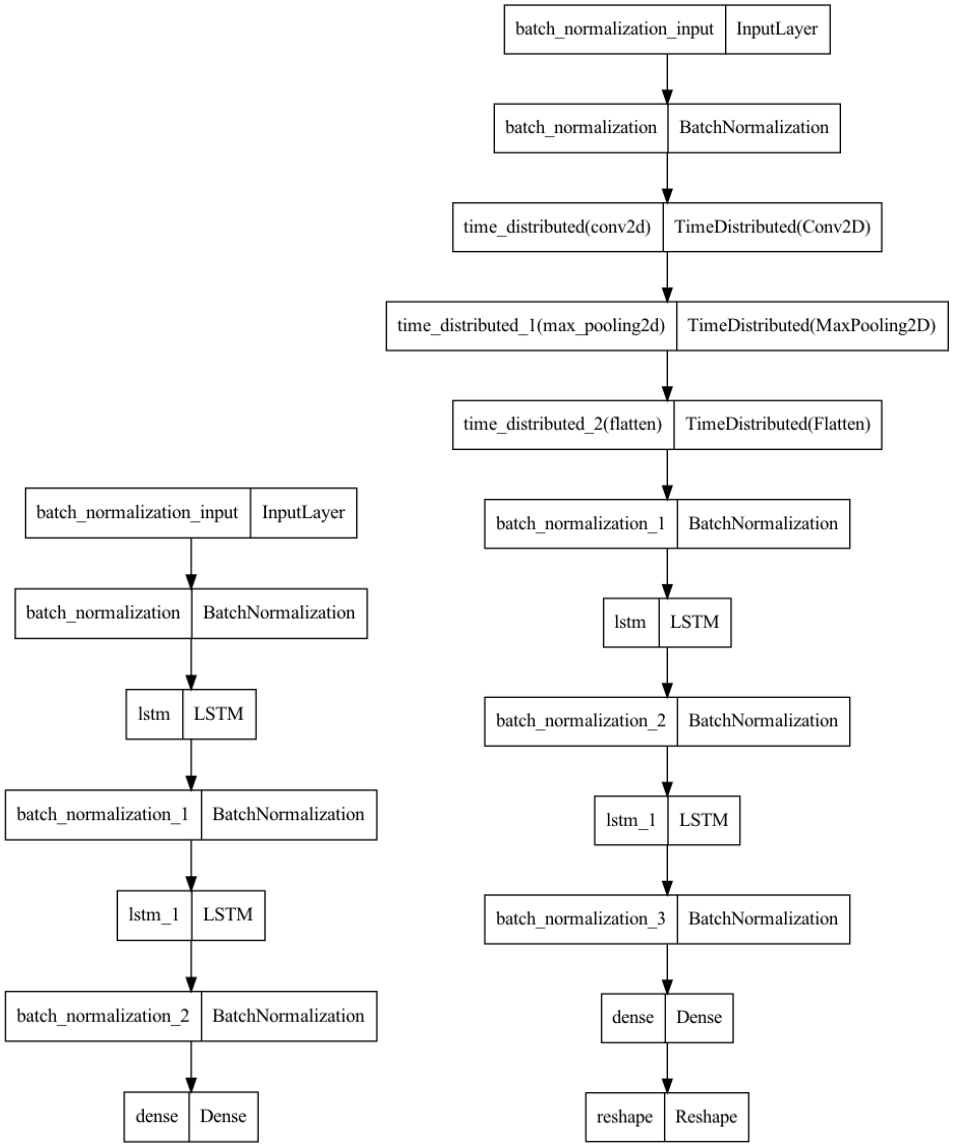
3.4.2 *Cutoff Magnitude*

Lastly, the cutoff magnitude is changed to $M6$. This cutoff leads to an even more imbalanced dataset (1511 and 9 for class 0 and 1 respectively), necessitating adjustments to the focal loss parameters α and γ to .95 and 5 respectively, in order to counteract this imbalance and enable effective training of the LSTM and CNN-LSTM algorithm.

3.5 *Hyper Parameters*

All following hyperparameters are empirically configured for both LSTM models, which share the same configuration and are defined by the highest combination of scores on precision, recall and F1-score in conjunction with the lowest loss on the validation set. Since the loss or error changes based on parameters of the focal loss function, a naive selection of the parameters solely based on lowest loss was not appropriate. Each model consists of two LSTM layers with 64 units each, using the tanh activation function. L1 and L2 regularization is used along with a dropout layer to achieve a less noisy convergence and to make the models generalize better to unseen data. The optimizer is set to Adam, which results in the best conversion and performance. The input data is normalized per batch and after every layer, a batch normalization layer is applied to reduce the likelihood of exploding or vanishing gradients, to facilitate a smoother learning process, and prevents the training to get stuck in local minima, as proposed and validated by Cooijmans et al. (2016) for use in LSTM. Without batch normalization, the models showcased the mentioned problems. Both models conclude with a dense layer.

For the CNN-LSTM model, the input data is first processed by a time-distributed Convolutional2D layer. This is followed by a max pooling layer, which reduces the feature maps by a factor of 2. Subsequently, the data is fed into the LSTM layers. Figure 6 shows the overview of the two models.



(a) Schematic LSTM model overview (b) Schematic CNN-LSTM model overview

Figure 6: Overview of model configuration for LSTM and CNN-LSTM.

3.6 *Programming language and Frameworks*

The following programming language and frameworks will be used:

- Python (Van Rossum & Drake, 2009)
- SKlearn (Pedregosa et al., 2011)
- Tensorflow (Martín Abadi et al., 2015)
- Keras (Chollet et al., 2015)
- Scipy (Virtanen et al., 2020)
- Matplotlib (Hunter, 2007)
- Seaborn (Waskom, 2021)
- Pandas (pandas development team, 2020)
- Numpy (Harris et al., 2020)
- Cartopy (Met Office, 2010 - 2015)

4 RESULTS

In this section, classification performance expressed with metrics mentioned in Section 3.3.4 for the models described in Section 3.2 will be presented. Linear regression is used as a baseline, together with the most naive approach by the persistence model.

4.1 *Lookback-window: Model Comparison*

Table 1a, 1b (complete table in the appendix on page 44), and 1c show the results of the models for a fixed set of parameters. It can been seen all models generalize well or even better beyond the training data on the test set. In response to the first and second sub-question ($SQ1$, $SQ2$), they indicate that multiple univariate linear regression performs the worst across all metrics. In contrast, the persistence model exhibits remarkably high performance when compared to both univariate and multivariate linear regression, particularly in terms of F1-score. This metric is crucial for comparison in cases of class imbalance. Generally, it is observed that when precision is high, recall tends to be relatively low, and vice versa. Consequently, it can be misleading to rely solely on either of these metrics when comparing the performance of different models.

In response to the third and fourth sub-question ($SQ3$, $SQ4$), both LSTM and CNN-LSTM models demonstrate comparable performance and outperform linear regression and the persistence model on all metrics. The boxplots presented in figure 7 show all baseline models are identified as outliers, suggesting they might come from a different population and as such can be regarded as performing differently from the deep learning

Table 1: Performance comparison of different models for one iteration (a and c) on validation(test) set with varying lookback-windows (6, 12, 72 months). The following static parameters were used: frequency aggregation (**day**), sub-region size in degrees (**5x5**), prediction window (**1 month**), and cutoff magnitude (**M** $\geq$ **4.5**).

(a) Precision, Recall, and F1-Score table.

Model	6M			12M			72M		
	P	R	F1	P	R	F1	P	R	F1
Persistence	.75(.76)	.75(.76)	.75(.76)	.76(.76)	.76(.76)	.76(.76)	.80(.78)	.80(.78)	.80(.78)
Univariate LR	.74(.71)	.50(.55)	.60(.62)	.64(.62)	.50(.53)	.56(.57)	.76(.70)	.55(.58)	.63(.63)
Multivariate LR	.84(**.83**)	.52(.54)	.64(.66)	.89(**.87**)	.54(.63)	.67(.73)	.92(**.87**)	.63(.73)	.75(.80)
Multivariate LSTM	.75(.76)	.84(**.88**)	.79(**.81**)	.75(.75)	.84(**.89**)	.79(.81)	.80(.79)	.85(**.87**)	.83(**.83**)
CNN-LSTM	.76(.76)	.84(.87)	.80(**.81**)	.76(.76)	.84(**.89**)	.80(**.82**)	.80(.81)	.82(.81)	.81(.81)

Note: P = Precision, R = Recall, and F1 = F1-Score.

(b) Mean(standard deviation) on test set, calculated for 10 iterations.

Model	6M			12M			72M		
	P	R	F1	P	R	F1	P	R	F1
Multivariate LSTM	.758(.004)	.869(.003)	.81(.001)	.762(.006)	.871(.007)	.813(.001)	.775(.007)	.877(.014)	.823(.006)
CNN-LSTM	.753(.009)	.878(.013)	.81(.001)	.759(.009)	.878(.014)	.814(.001)	.783(.009)	.865(.021)	.822(.007)

(c) ROC and AUPRC table.

Model	6M		12M		72M	
	ROC	AUPRC	ROC	AUPRC	ROC	AUPRC
Persistence	.84(.85)	.78(.78)	.85(.85	.79(.79)	.87(86)	.82 (.80)
Univariate LR	.72(.74)	.68(.68)	.71(.72)	.62(.62)	.75(.75)	.71(.68)
Multivariate LR	.75(.76)	.73(.74)	.76(.80)	.76(.79)	.81(.85)	.82(.83)
Multivariate LSTM	.96(**.97**)	.87(**.89**)	.96(**.97**)	.88(**.88**)	.97(**.97**)	.91(**.91**)
CNN-LSTM	.96(**.97**)	.86(.87)	.96(**.97**)	.87(.86)	.97(**.97**)	.91(.90)

Note: ROC = Area under ROC curve and AUPRC = Area under Precision-Recall curve.

models, except for multivariate linear regression compared with CNN-LSTM with a lookback-window of 72 months. A one-sided Mann-Whitney U test was conducted to determine whether CNN-LSTM had significantly greater F1-scores than LSTM for 72, 12, and 6 months, respectively. There was not a significant difference in the scores for CNN-LSTM and LSTM for 72 and 6 months ($U = 46, 81, 61$, $p = .633, .011, .214$) at an alpha level of 0.05. As the lookback window increases, all models exhibit similar or improved performance, particularly in the case of multivariate linear regression. In response to part of the fifth sub-question ($SQ5$), one-sided Mann-Whitney U tests were conducted to compare the performance of LSTM with different lookback-windows. Significant differences were found when comparing LSTM and CNN-LSTM, repectively, with a lookback-window of 72 months to 12 months ($U = 90, 79$, $p = .001, .016$) at an alpha level of 0.05. Furthermore, LSTM and respectively CNN-LSTM

with a lookback-window of 12 months also significantly outperformed LSTM and CNN-LSTM with a lookback-window of 6 months ($U = 97, 99$, $p < .001, .001$) at an alpha level of 0.05. To illustrate the metrics, confusion matrices are provided (see Figure 10). For all models the false positives and false negatives are relatively low, which suggests the models perform reasonably well. The false negatives for univariate linear regression are comparable to the true positives. Especially LSTM shows the largest difference between true positives and false positives which is reflected in the highest score for recall.

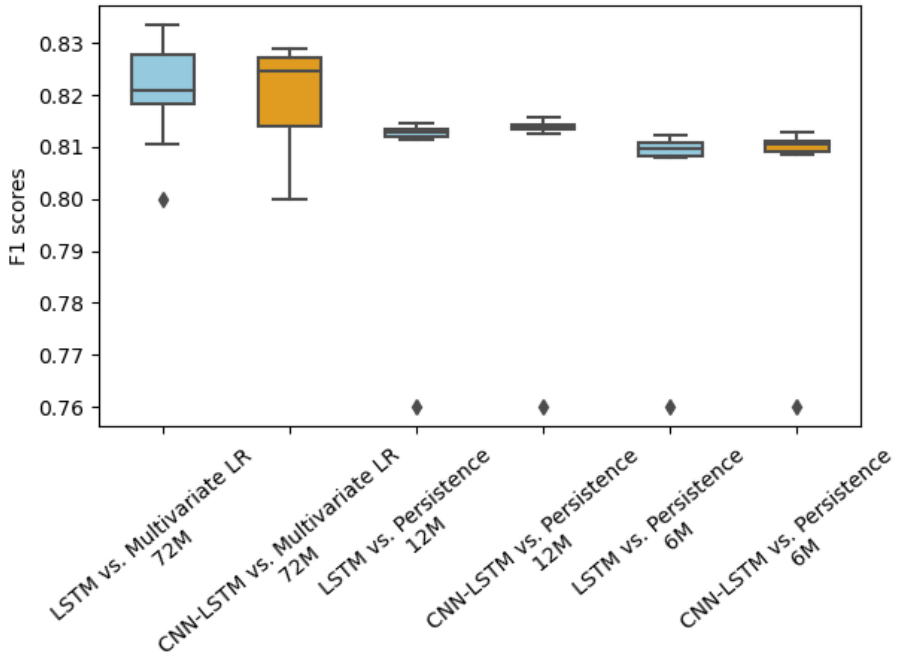
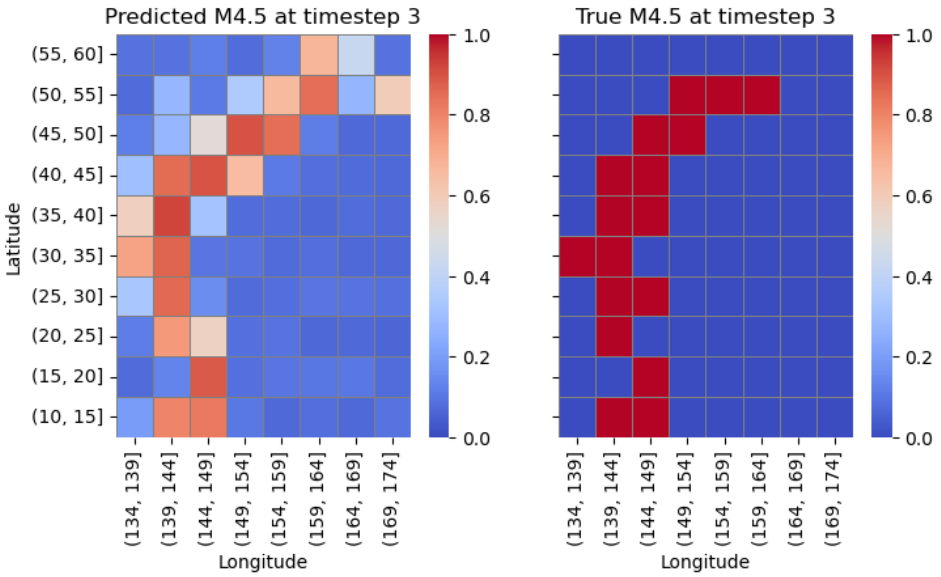


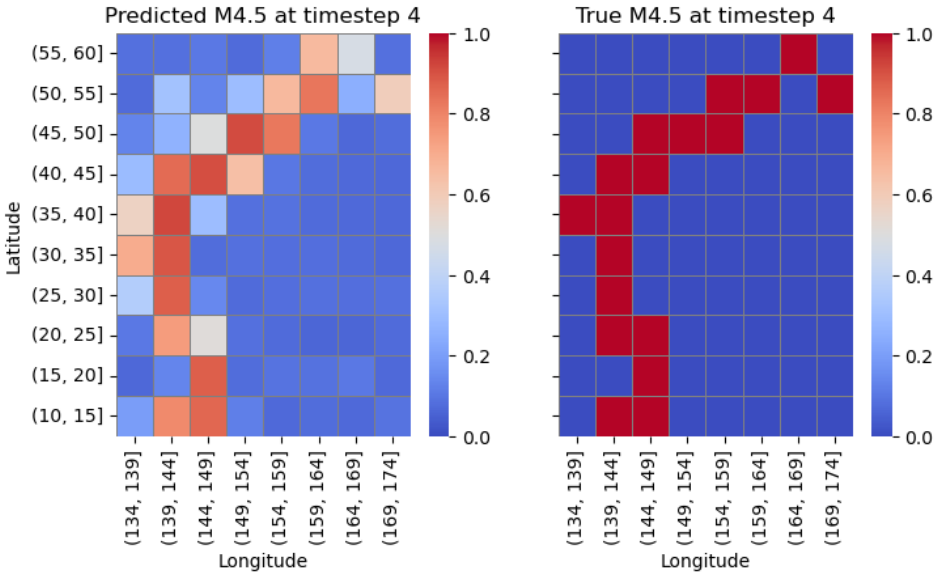
Figure 7: Boxplots using the combined F1-scores (test set) of the basemodel with the highest score and the scores of LSTM or CNN-LSTM. Consequently, all other basemodels (with lower scores) are regarded as outliers. The diamond shaped icon corresponds to the value of the basemodel.

Although these results for LSTM and CNN-LSTM models may suggest their suitability for predicting earthquakes, a closer examination of the visualized predictions reveal that the models generate nearly identical predictions each month (see Figure 8, 9). This pattern seems to minimize the algorithm's loss, since that is the objective function. It is also important to note that the actual occurrences of earthquakes are strikingly similar every month for each region. Therefore, predicting the following month's occurrences holds little value when the patterns remain nearly constant.

To address the shortcomings mentioned in the previous section, two potential solutions are considered: 1) shortening the prediction window, and 2) downscaling the size of the sub-regions. Training CNN-LSTM was found to be infeasible due to memory constraints. Despite several attempts to optimize the CNN-LSTM model and increasing computational resources, the errors persisted. Furthermore, there were no results for multivariate linear regression when using a prediction window of two weeks. The predictions were ill-defined, with the model consistently predicting no earthquake occurrence at all timesteps, due to increased dimensionality, known as the curse of dimensionality which often leads to computational inefficiencies and problems with overfitting, subsequently performing poorly on new, unseen data. Since the two solutions did not address the issue of similarity of predictions for the two weeks for LSTM, it was decided to exclude this analysis from the thesis.

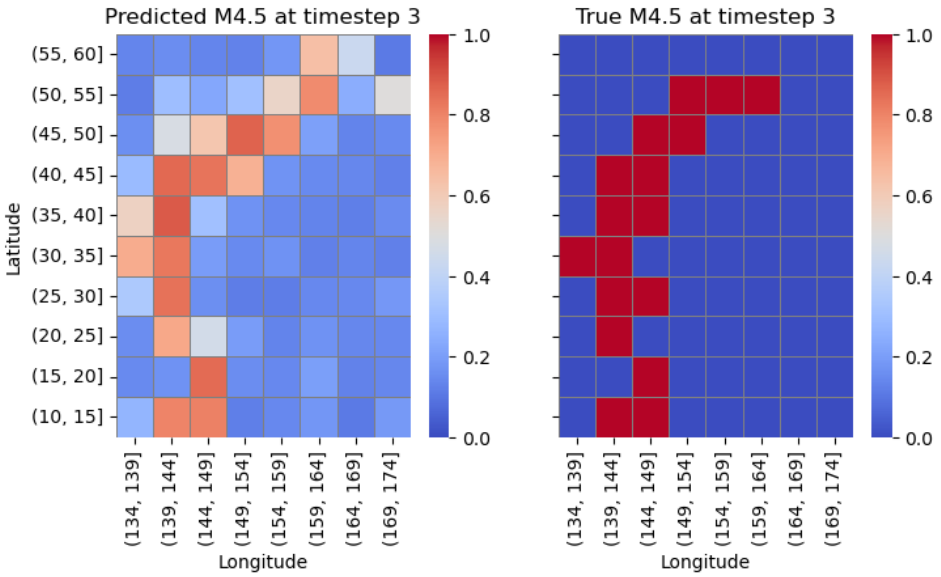


(a) Predictions versus true magnitude at 3rd month

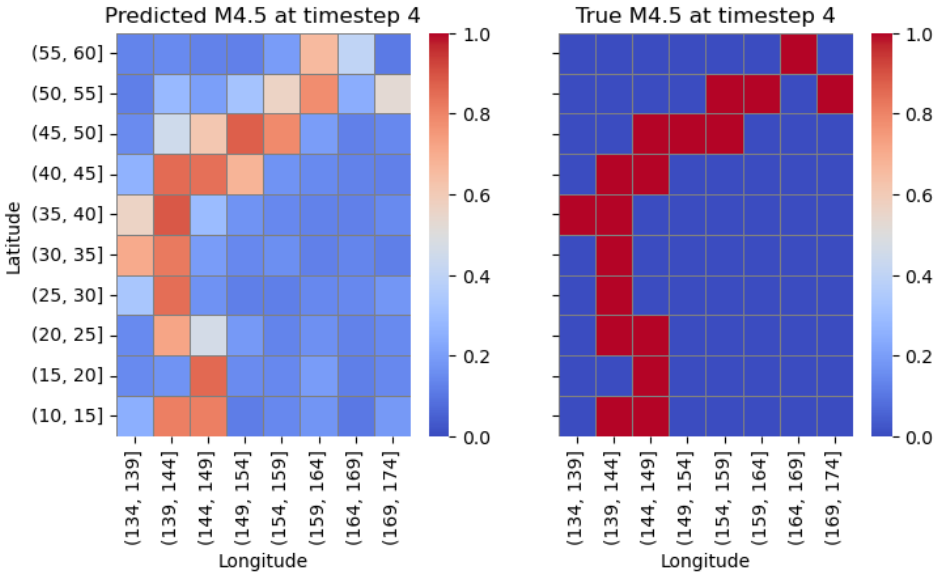


(b) Predictions versus true magnitude at 4th month

Figure 8: Visualisation of the predictions for LSTM at two timesteps, trained with 72 months lookback-window. Left are the predicted probabilities an earthquake with at least magnitude $M4.5$ will happen. Right the true occurances.

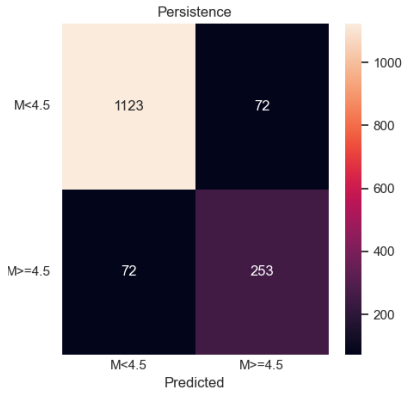


(a) Predictions versus true magnitude at 3rd month

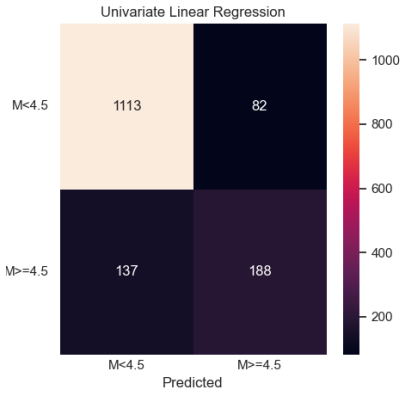


(b) Predictions versus true magnitude at 4th month

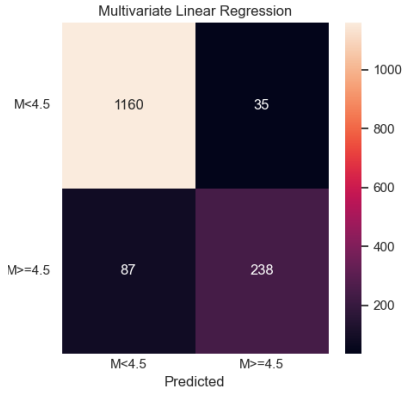
Figure 9: Visualisation of the predictions for CNN-LSTM at two timesteps, trained with 72 months lookback-window. Left are the predicted probabilities an earthquake with at least magnitude $M4.5$ will happen. Right the true occurances.



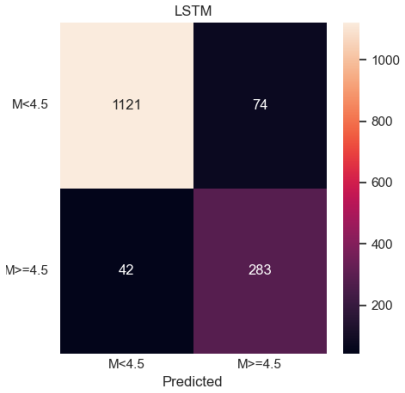
(a) Persistence



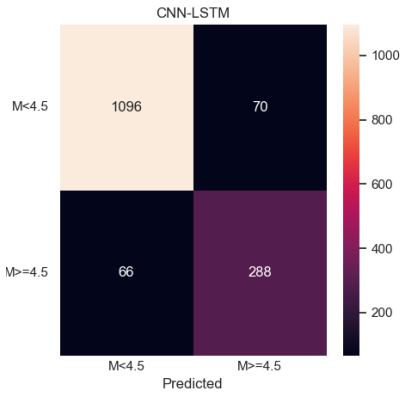
(b) Univariate Linear Regression



(c) Multivariate Linear Regression



(d) LSTM



(e) CNN-LSTM

Figure 10: Confusion Matrices on the test set for five different models with 5x5 sub-regions, 72 months lookback-window, 1 month prediction window and cutoff magnitude M4.5.

4.2 *Cutoff Magnitude: Model Comparison*

In this section, the results for a higher cutoff magnitude of $M6$ are presented in response to the other part of the fifth sub-question ($SQ5$). As observed in table 2a, 2b (complete table in the appendix on page 44), and 2c, there are no results for multivariate linear regression, as the model failed to make any positive predictions. The overall performance for all models is quite low and the precision for earthquakes with at least magnitude $M6$ is especially low, with the CNN-LSTM model achieving relatively better results in terms of recall, but F1-score for LSTM and CNN-LSTM are comparable. When looking at F1-scores all models performance degrades on the test set, compared to the validation set. This maybe due to the very low count of earthquakes above $M6$ and the difference in representation between the datasets. Both LSTM models perform better than the absolute baseline from the persistence model and univariate linear regression. The AUPRC for CNN-LSTM and LSTM model is higher than univariate linear regression. Since the non occurrences increased and are correctly classified as true negative, the ROC-AUC is rather high, but when taking into account only precision and recall, the AUPRC is relatively low as well as F1 scores. A one-sided Mann-Whitney U test was conducted to compare the F1-scores for CNN-LSTM and LSTM. The results did not indicate a significant difference in the scores for CNN-LSTM as compared to LSTM ($U = 65$, $p = .137$) at an alpha level of 0.05. In addition, a one-sided Mann-Whitney U test to compare the F1-scores for LSTM and CNN-LSTM in predicting $M6$ and $M4.5$ earthquakes, revealed a highly significant lower performance of the corresponding models when predicting $M6$ or larger magnitudes (U = 0.0, $p = 9.134 \times 10^{-5}$) at an alpha level of 0.05.

Figure 11, 12 visualizes the result of LSTM and CNN-LSTM. By comparing the colors – corresponding to the probabilities – with the colors from Figure 8, 9 it can be noticed the predictions are less certain, varying more in terms of probabilities, and as a result, the predictions vary more from month to month than previous experiments. From the confusion matrices for the four models on the test set (see Figure 13) it can be seen only the two LSTM models correctly predict 3 and 4 earthquakes of 9 in total, while making 82 and 118 false predictions an earthquake will happen. Here the difference between the values for ROC-AUC and AUPRC are visually illustrated. When taking into account the correctly classified negative class the ROC-AUC is relatively high in comparison with AUPRC, because the models correctly predict the negative class for most timesteps.

Table 2: Performance comparison of different models for one iteration (a and c) on validation(test) set with prediction-window of 1 month and sub-regions scale of 5×5 degrees. The following static parameters were used: frequency aggregation (**day**), lookback-window (**72 months**), and cutoff magnitude ($\mathbf{M} \geq \mathbf{6}$).

(a) Precision, Recall, and F1-Score table.

Model	1M		
	P	R	F1
Persistence	.06(.00)	.06(.00)	.06(.00)
Univariate LR	.02(.00)	.06(.00)	.03(.00)
Multivariate LR	-	-	-
Multivariate LSTM	.04(**.04**)	.24(.33)	.07(**.06**)
CNN-LSTM	.05(.03)	.35(**.44**)	.09(**.06**)

Note: P = Precision, R = Recall, and F1 = F1-Score.

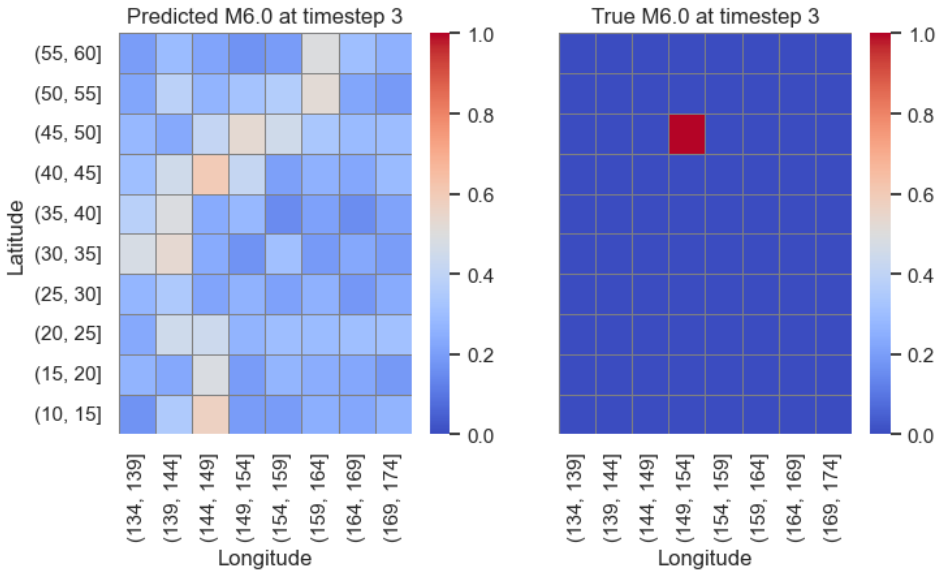
(b) Mean(standard deviation) on test set, calculated for 10 iterations.

Model	1M		
	P	R	F1
Multivariate LSTM	.027(.010)	.233(.105)	.048(.018)
CNN-LSTM	.031(.012)	.478(.172)	.057(.023)

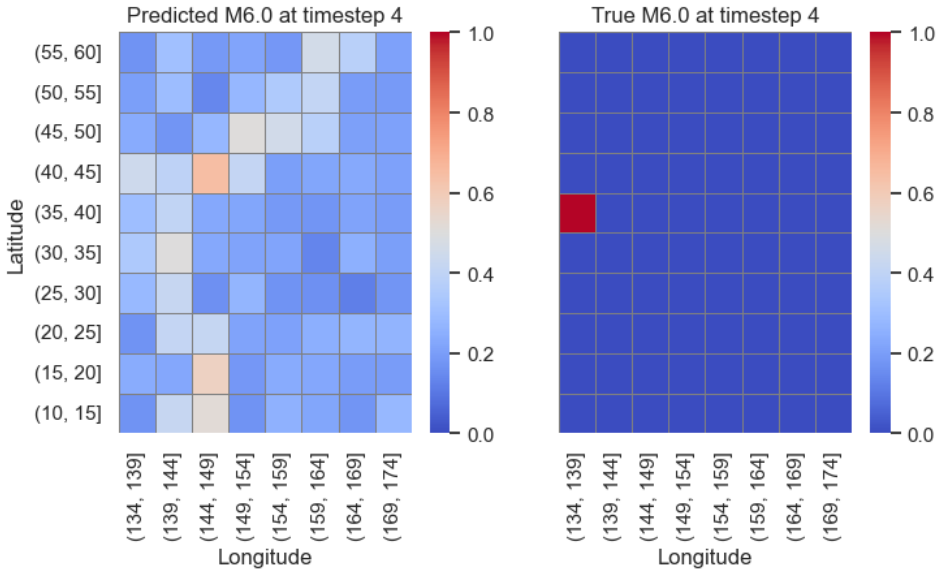
(c) ROC and AUPRC table.

Model	1M	
	ROC	AUPRC
Persistence	.52(.50)	.06(.00)
Univariate LR	.51(.47)	.04(.00)
Multivariate LR	-	-
Multivariate LSTM	.86(.86)	.04(**.02**)
CNN-LSTM	.89(**.88**)	.06(**.02**)

Note: ROC = Area under ROC curve and AUPRC = Area under Precision-Recall curve.

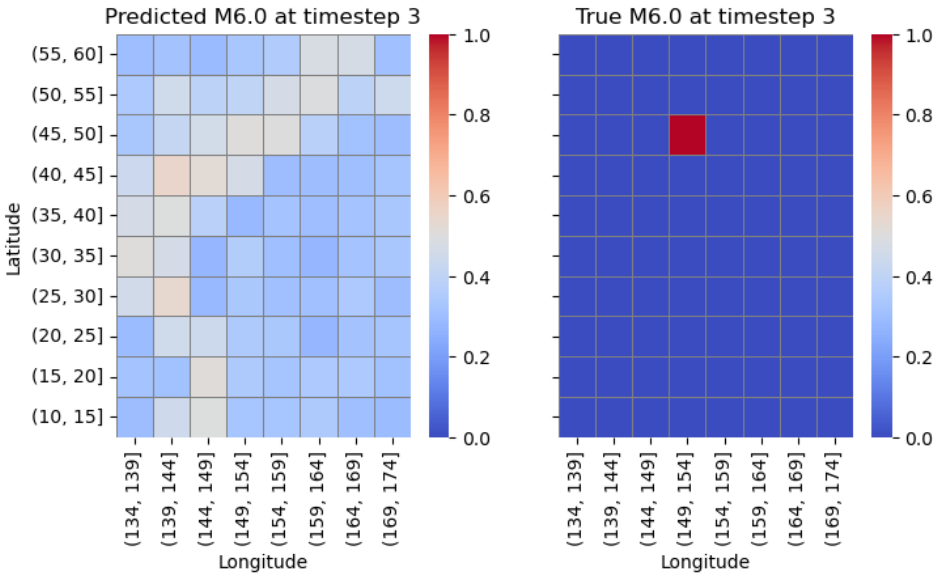


(a) Predictions versus true magnitude at 3rd month

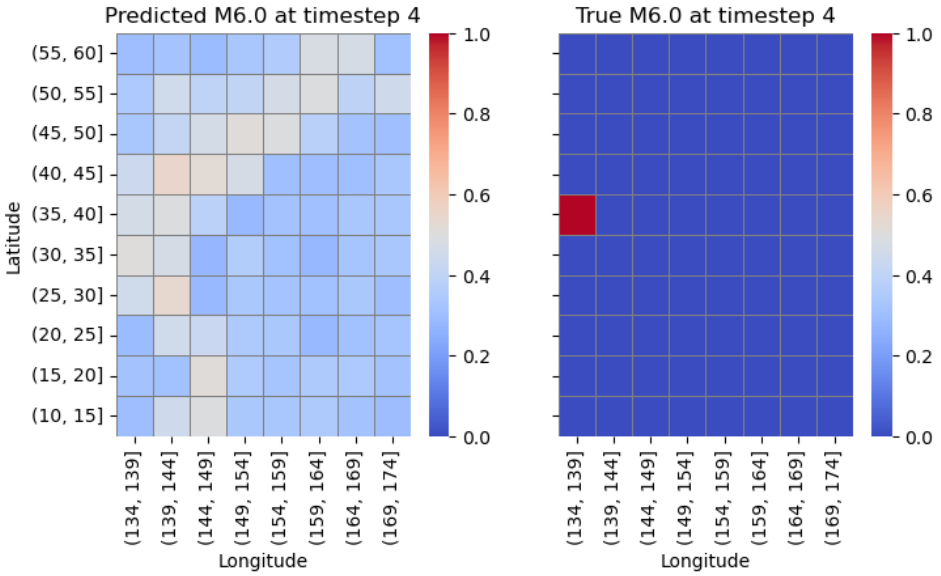


(b) Predictions versus true magnitude at 4th month

Figure 11: Visualisation of the predictions for LSTM at two timesteps, trained with 72 months lookback-window. Left are the predicted probabilities an earthquake with at least magnitude $M6$ will happen. Right the true occurances.

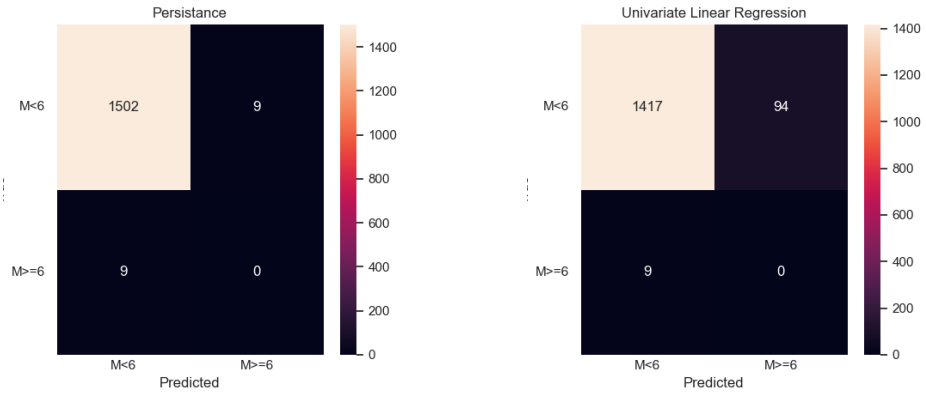


(a) Predictions versus true magnitude at 3rd month



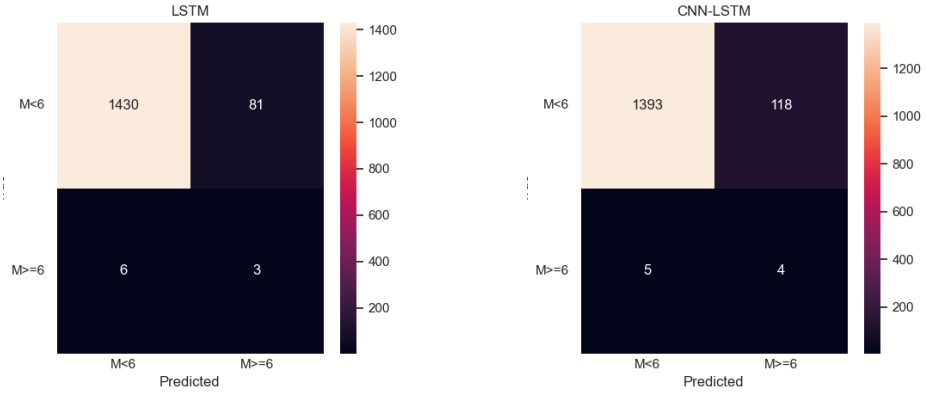
(b) Predictions versus true magnitude at 4th month

Figure 12: Visualisation of the predictions for CNN-LSTM at two timesteps, trained with 72 months lookback-window. Left are the predicted probabilities an earthquake with at least magnitude *M*6 will happen. Right the true occurances.



(a) Persistence

(b) Univariate Linear Regression



(c) LSTM

(d) CNN-LSTM

Figure 13: Confusion Matrices on the test set for four different models with 5×5 sub-regions, 72 months lookback-window, 1 month prediction window and cutoff magnitude $M6$.

5 DISCUSSION

The goal of this study was to predict destructive earthquakes in Japan using linear regression, LSTM, and CNN-LSTM, and to compare their performance across different timeframes, and magnitudes. The main research question focused on the extent to which earthquakes in the area of Japan, can be forecasted short term (4 weeks) using historical seismic data.

In response to the first sub-question ($SQ1$), linear regression using only temporal characteristics demonstrated limited performance in earthquake forecasting. The obtained results indicate that LSTM and CNN-LSTM outperform linear regression and the persistence model across most metrics. Clearly earthquakes show signs of non-linearity, since these models show better performance. In addition, they are able to capture long-term dependencies better than linear regression and the persistence model. However, their practical societal value is particularly limited due to the similarities between consecutive monthly predictions. This is in line with findings from the USGS. According to which "predictions are so general that there will always be an earthquake that fits".

Sub-question two ($SQ2$) addressed the performance of linear regression using spatio-temporal information. While the incorporation of spatial data led to an improvement over using only temporal characteristics, the performance remained sub-optimal when compared with LSTM and CNN-LSTM models. The results of this study support the findings of Wang et al. (2020), who found that spatio-temporal data correlations provide better prediction results than only mining temporal data correlations.

Sub-questions three ($SQ3$) and four ($SQ4$) focused on LSTM and CNN-LSTM models, respectively. Both models outperformed linear regression across most metrics. However, it appears that explicitly modelling spatio-temporal information by CNN-LSTM does not provide added benefit over LSTM. This could be due to the fact that the existing relationship between multivariate time-series is already captured by LSTM. Since including all sub-regions resulted in higher performance it could potentially be beneficial for future research to include a larger area subdivided in smaller regions, maybe even the whole globe, since tectonic plates and their faults are assumed to be interconnected (Kannan, 2014; Wang et al., 2020).

Lastly, the fifth sub-question ($SQ5$) investigated the influence of various time-frames and cutoff magnitudes on the performance of the models.

It was observed that as the lookback-window increased, the performance of all models was similar or improved, particularly for multivariate linear regression. Both LSTM and CNN-LSTM showed an statistically significant increase in performance with longer lookback-windows. This

result is not in line with the literature, which suggests the existence of an optimal lookback-window of 120 days (Jipan et al., 2018).

When evaluating the effect of smaller sub-regions and shorter prediction windows, LSTM emerged as the best performer, but still generated similar predictions for consecutive months. This finding highlights the need for further research to address this limitation and enhance the practical applicability of LSTM and CNN-LSTM in earthquake prediction. A possible cause could be the lack of data. As shown, the raw data from USGS seems normally distributed while the general consensus is that the amount of earthquake occurrences increases with a factor 10 for every unit decrease in magnitude (Gutenberg & Richter, 1955). It is unclear why the data from USGS for our area of interest does not exhibit this pattern. Possibly the used recording instrumentation is not sensitive enough to register smaller motion. Besides, the hypocenters are commonly estimated by combining sensor data from recording stations (Mousavi et al., 2019). It is possible these estimations add extra noise and make it more difficult for the models to capture the signal. Another explanation of the results could be the relatively short timeframe (72 months) in light of supercycles (hundreds of years) in which earthquakes tend to occur (Rundle et al., 2021). Maybe we simply need more data to capture and model existing patterns happening over multiple centuries.

The results also indicate that the performance of all models for earthquakes with at least magnitude $M6$ was relatively low and more uncertain, which may be due to the limitations of the models, the data used, or a combination. The distribution of classes became very imbalanced and there were possibly not enough earthquakes of these magnitudes to train the models effectively. Another explanation could be different use of instrumentation over time (Rikitake, 1968), and because of this the training data has different characteristics, and therefore the models trained on this data do not generalize well to unseen data. These results partly contradict the study of Nicolis et al. (2021) which found that LSTM was able to predict larger magnitudes better than smaller magnitudes, due the decreased number of events to process, which is contradictory with the general consensus that more data results in better performance. Possibly their model's capacity was to small.

The broader implications for society are widespread. Since earthquakes cannot be accurately predicted with presented models, resulting in both high true positives and true negatives, it is essential for policymakers to focus on disaster preparedness, building resilient infrastructure, and planning effective evacuation strategies in all earthquake-prone regions (Greenberg et al., 2007; Kaveh et al., 2020). As a result, this also has substantial economic implications on insurance, healthcare and construction

costs (Cavallo & Noy, 2009; Greenberg et al., 2007). The most expensive earthquake in U.S. history for the insurance industry occurred in California in 1994 and cost around $15.3 billion dollars according to Aon. Apart from these costs, predicting (or lack thereof) of earthquakes can have profound psychological impacts on people, including inducing panic, anxiety, phobic fears, feelings of vulnerability, guilt, tendencies towards isolation and withdrawal, depression, anger, frustration, interpersonal and marital problems, Post-Traumatic Stress Disorder (PTSD) (Fahrudin, 2012). This highlights the need for effective communication strategies and psychological support systems.

Future research could further investigate different models, the potential of incorporating other relevant features, such as geological, geophysical, or geomagnetic data, to improve model performance (Cicerone et al., 2009) and overall gather more data, e.g. from mobile phones (Kong et al., 2016; Minson et al., 2015).

6 CONCLUSION

This study compared linear regression, LSTM, and CNN-LSTM in predicting destructive earthquakes in Japan, highlighting their potential and limitations. The study demonstrates that while LSTM and CNN-LSTM models show promise in predicting earthquakes when quantitatively evaluating commonly used metrics, their practical value remains limited. Further research is needed to improve their performance and applicability.

Based on the findings of this study, future work could focus on exploring other deep learning architectures and techniques to address the issue of similar consecutive monthly predictions and imbalance, and focus on predicting on larger areas, with smaller sub-regions, on lower timeframes and with higher magnitudes. Additionally, gathering more data for longer periods of time with more sensitive and standardised instruments, and incorporating other relevant features, such as geological, geophysical, geomagnetic, GPS, and data from mobile accelerometers may improve model performance. Finally, it may be valuable to investigate the potential of ensemble methods more, combining the strengths of multiple models, to further enhance earthquake prediction accuracy, reliability, and practicality.

7 DATA SOURCE/CODE/ETHICS/TECHNOLOGY STATEMENT

The two datasets have been acquired from the USGS (US Geological Survey) and NGDC/WDS (National Geophysical Data Center and World Data Service) through an API (Application Programming Interface). The obtained data is publicly accessible. Work on this thesis did not involve

collecting data from human participants or animals. The original owner of the data and code used in this thesis retains ownership of the data and code during and after the completion of this thesis. All the figures belong to the author. The thesis code can be accessed through the GitHub repository following the link https://github.com/TheTrueSquirrel/thesis_spring_2023. In terms of writing, the author used assistance with the language of the paper. Generative language models (chatGPT and GPT-4 from OpenAI, 2021) were used to improve the author's original content, for paraphrasing, spell checking, grammar and assist in writing Python and LaTeXcode. In addition, Grammarly is used for spell checking.

REFERENCES

Allen, R. (2017). Quake warnings, seismic culture. *Science*, *358*, 1111–1111. https://doi.org/10.1126/science.aar4640

Allen, R., Cochran, E., Huggins, T., Miles, S., & Otegui, D. (2018). Lessons from mexico's earthquake early warning system. *Eos*, *99*. https://doi.org/10.1029/2018EO105095

Allen, R. M., & Melgar, D. (2019). Earthquake early warning: Advances, scientific challenges, and societal needs. *Annual Review of Earth and Planetary Sciences*, *47*(1), 361–388. https://doi.org/10.1146/annurev-earth-053018-060457

Asencio–Cortés, G., Morales–Esteban, A., Shang, X., & Martınez–Álvarez, F. (2018). Earthquake prediction in california using regression algorithms and cloud-based big data infrastructure. *Computers & Geosciences*, *115*, 198–210.

Asim, K., Martínez-Álvarez, F., Basit, A., & Iqbal, T. (2017). Earthquake magnitude prediction in hindukush region using machine learning techniques. *Natural Hazards*, *85*, 471–486. https://doi.org/10.1007/s11069-016-2579-3

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Carlson, J. M., Langer, J. S., & Shaw, B. E. (1994). Dynamics of earthquake faults. *Reviews of Modern Physics*, *66*(2), 657.

Cavallo, E. A., & Noy, I. (2009). The economics of natural disasters: A survey.

Chollet, F., et al. (2015). Keras.

Cicerone, R. D., Ebel, J. E., & Britton, J. (2009). A systematic compilation of earthquake precursors. *Tectonophysics*, *476*(3-4), 371–396.

Coburn, A., & Spence, R. (2003). *Earthquake protection*. John Wiley & Sons.

Cooijmans, T., Ballas, N., Laurent, C., Gülçehre, Ç., & Courville, A. (2016). Recurrent batch normalization. *arXiv preprint arXiv:1603.09025*.

Demiriz, A., Bennett, K. P., & Shawe-Taylor, J. (2002). Linear programming boosting via column generation. *Machine Learning*, *46*, 225–254.

Fahrudin, A. (2012). Psychosocial reaction and trauma after a natural disaster: The role of coping behavior. *Asian Social Work and Policy Review*, *6*(3), 192–202.

Feng, B., & Fox, G. C. (2021). Spatiotemporal pattern mining for nowcasting extreme earthquakes in southern california. *2021 IEEE 17th International Conference on eScience (eScience)*, 99–107.

Fox, G. C., Rundle, J. B., Donnellan, A., & Feng, B. (2022). Earthquake nowcasting with deep learning. *GeoHazards*, *3*(2), 199–226. https://doi.org/10.3390/geohazards3020011

Galkina, A., & Grafeeva, N. (2019). Machine learning methods for earthquake prediction: A survey. *Proceedings of the Fourth Conference on Software Engineering and Information Management (SEIM-2019), Saint Petersburg, Russia*, *13*, 25.

Ge, L., Wu, K., Zeng, Y., Chang, F., Wang, Y., & Li, S. (2021). Multi-scale spatiotemporal graph convolution network for air quality prediction. *Applied Intelligence*, *51*, 3491–3505.

Greenberg, M. R., Lahr, M., & Mantell, N. (2007). Understanding the economic costs and benefits of catastrophes and their aftermath: A review and suggestions for the us federal government. *Risk Analysis: An International Journal*, *27*(1), 83–96.

Gutenberg, B., & Richter, C. (1955). Magnitude and energy of earthquakes. *Nature*, *176*(4486), 795–795.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hayakawa, M., Yamauchi, H., Ohtani, N., Ohta, M., Tosa, S., Asano, T., Schekotov, A., Izutsu, J., Potirakis, S. M., Eftaxias, K., et al. (2016). On the precursory abnormal animal behavior and electromagnetic effects for the kobe earthquake (m˜ 6) on april 12, 2013. *Open Journal of Earthquake Research*, *5*(03), 165.

Hays, W. W. (1990). Perspectives on the international decade for natural disaster reduction. *Earthquake spectra*, *6*(1), 125–145.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Jipan, H., Wang, X., Zhao, Y., Chen, X., & Han, X. (2018). Large earthquake magnitude prediction in taiwan based on deep learning neural network. *Neural Network World*, *28*, 149–160. https://doi.org/10.14311/NNW.2018.28.009

Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. In *Advances in psychology* (pp. 471–495). Elsevier.

Kannan, S. (2014). Innovative mathematical model for earthquake prediction. *Engineering Failure Analysis*, *41*, 89–95.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 1725–1732.

Kaveh, A., Javadi, S., & Moghanni, R. M. (2020). Emergency management systems after disastrous earthquakes using optimization methods: A comprehensive review. *Advances in Engineering Software*, *149*, 102885.

Kong, Q., Allen, R., Schreier, L., & Kwon, Y.-W. (2016). Myshake: A smartphone seismic network for earthquake early warning and beyond. *Science Advances*, *2*, e1501055–e1501055. https://doi.org/10.1126/sciadv.1501055

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, *37*(4), 1748–1764.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, ... Xiaoqiang Zheng. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems [Software available from tensorflow.org]. https://www.tensorflow.org/

Met Office. (2010 - 2015). *Cartopy: A cartographic python library with a matplotlib interface*. Exeter, Devon. https://scitools.org.uk/cartopy

Mieler, M., & Mitrani-Reiser, J. (2018). Review of the state of the art in assessing earthquake-induced loss of functionality in buildings. *Journal of Structural Engineering*, *144*(3), 04017218.

Mignan, A., & Broccardo, M. (2019). One neuron versus deep learning in aftershock prediction. nature 574: E1–e3.

Minson, S., Brooks, B., Glennie, C., Murray, J., Langbein, J., Owen, S., Heaton, T., Iannucci, B., & Hauser, D. (2015). Crowdsourced earthquake early warning. *Science Advances*, *1*. https://doi.org/10.1126/sciadv.1500036

Mousavi, S., Sheng, Y., Weiqiang, Z., & Beroza, G. (2019). Stanford earthquake dataset (stead): A global data set of seismic signals for ai. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2019.2947848

National Geophysical Data Center & World Data Service (NGDC/WDS). (2023). Ncei/wds global significant earthquake database [data retrieved from NOAA National Centers for Environmental].

Nicolis, O., Plaza, F., & Salas, R. (2021). Prediction of intensity and location of seismic events using deep learning [Towards Spatial Data Science]. *Spatial Statistics*, *42*, 100442. https://doi.org/https://doi.org/10.1016/j.spasta.2020.100442

Ogata, Y. (1988). Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical association*, *83*(401), 9–27.

Ogata, Y. (1998). Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, *50*, 379–402.

OpenAI. (2021). Chatgpt: A large language model.

pandas development team, T. (2020). *Pandas-dev/pandas: Pandas* (Version latest). Zenodo. https://doi.org/10.5281/zenodo.3509134

Pardo, R. (1992). *Design, testing, and optimization of trading systems* (Vol. 2). John Wiley & Sons.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Perez, E., & Thompson, P. (1994). Natural hazards: Causes and effects: Lesson 2—earthquakes. *Prehospital and Disaster Medicine*, *9*(4), 260–272.

Pretto, E., Safar, P., Group, D. R. S., et al. (1993). 298 disaster reanimatology potentials revealed by interviews of survivors of five major earthquakes. *Prehospital and Disaster Medicine*, *8*(S3), S139–S139.

Rikitake, T. (1968). Earthquake prediction. *Earth-Science Reviews*, *4*, 245–282.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation* (tech. rep.). California Univ San Diego La Jolla Inst for Cognitive Science.

Rundle, J. B., Stein, S., Donnellan, A., Turcotte, D. L., Klein, W., & Saylor, C. (2021). The complex dynamics of earthquake fault systems: New approaches to forecasting and nowcasting of earthquakes. *Reports on progress in physics*, *84*(7), 076801.

Sobolev, G. A. (2015). Methodology, results, and problems of forecasting earthquakes. *Herald of the Russian Academy of Sciences*, *85*(2), 107–111.

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Waheed, U., Afify, A., Fehler, M., & Fulcher, B. (2020). Winning with simple learning models: Detecting earthquakes in groningen, the netherlands. *EAGE 2020 Annual Conference & Exhibition Online*, *2020*(1), 1–5.

Wang, Q., Guo, Y., Yu, L., & Li, P. (2020). Earthquake prediction based on spatio-temporal data mining: An lstm network approach. *IEEE Transactions on Emerging Topics in Computing*, *8*(1), 148–158. https://doi.org/10.1109/TETC.2017.2699169

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

Table 3: Performance metrics on test set, calculated for 10 iterations. Lookback-windows (6, 12, 72 months). The following static parameters were used: frequency aggregation (**day**), sub-region size in degrees (**5x5**), prediction window (**1 month**), and cutoff magnitude (**M** $\geq$ **4.5**).

Model	6M			12M			72M		
	P	R	F1	P	R	F1	P	R	F1
Multivariate LSTM	0.751	0.875	0.808	0.759	0.873	0.812	0.777	0.889	0.829
	0.757	0.870	0.810	0.770	0.859	0.812	0.770	0.898	0.830
	0.761	0.868	0.811	0.763	0.871	0.813	0.774	0.874	0.821
	0.755	0.870	0.808	0.757	0.878	0.813	0.775	0.849	0.811
	0.762	0.868	0.811	0.759	0.872	0.812	0.777	0.877	0.824
	0.757	0.872	0.810	0.764	0.870	0.814	0.787	0.886	0.834
	0.762	0.866	0.811	0.759	0.876	0.813	0.774	0.886	0.826
	0.764	0.867	0.812	0.753	0.884	0.813	0.765	0.883	0.820
	0.755	0.870	0.808	0.763	0.868	0.812	0.785	0.855	0.819
	0.759	0.868	0.810	0.772	0.862	0.815	0.766	0.877	0.818
CNN-LSTM	0.759	0.867	0.809	0.765	0.870	0.814	0.792	0.834	0.813
	0.748	0.885	0.811	0.770	0.862	0.814	0.788	0.871	0.827
	0.762	0.867	0.811	0.759	0.877	0.814	0.788	0.834	0.810
	0.742	0.892	0.810	0.741	0.905	0.815	0.772	0.886	0.825
	0.751	0.883	0.812	0.769	0.863	0.813	0.768	0.895	0.827
	0.766	0.860	0.811	0.754	0.884	0.814	0.788	0.855	0.820
	0.766	0.861	0.811	0.763	0.869	0.813	0.787	0.874	0.828
	0.743	0.898	0.813	0.755	0.884	0.815	0.784	0.849	0.815
	0.743	0.888	0.809	0.748	0.896	0.816	0.796	0.865	0.829
	0.745	0.884	0.809	0.766	0.870	0.814	0.769	0.889	0.825

Note: P = Precision, R = Recall, and F1 = F1-Score.

Table 4: Performance metrics on test set, calculated for 10 iterations. Prediction-window of 1 month and sub-regions scale of 5×5 degrees. The following static parameters were used: frequency aggregation (**day**), lookback-window (**72 months**), and cutoff magnitude (**M** $\geq$ **6**).

Model	1M		
	P	R	F1
Multivariate LSTM	0.032	0.333	0.059
	0.029	0.222	0.051
	0.016	0.111	0.027
	0.028	0.333	0.051
	0.024	0.222	0.044
	0.052	0.444	0.093
	0.013	0.111	0.023
	0.025	0.222	0.045
	0.024	0.111	0.040
	0.026	0.222	0.047
CNN-LSTM	0.041	0.667	0.077
	0.030	0.444	0.056
	0.017	0.333	0.032
	0.041	0.556	0.076
	0.018	0.111	0.031
	0.053	0.667	0.098
	0.027	0.444	0.051
	0.035	0.667	0.066
	0.011	0.333	0.022
	0.033	0.556	0.063

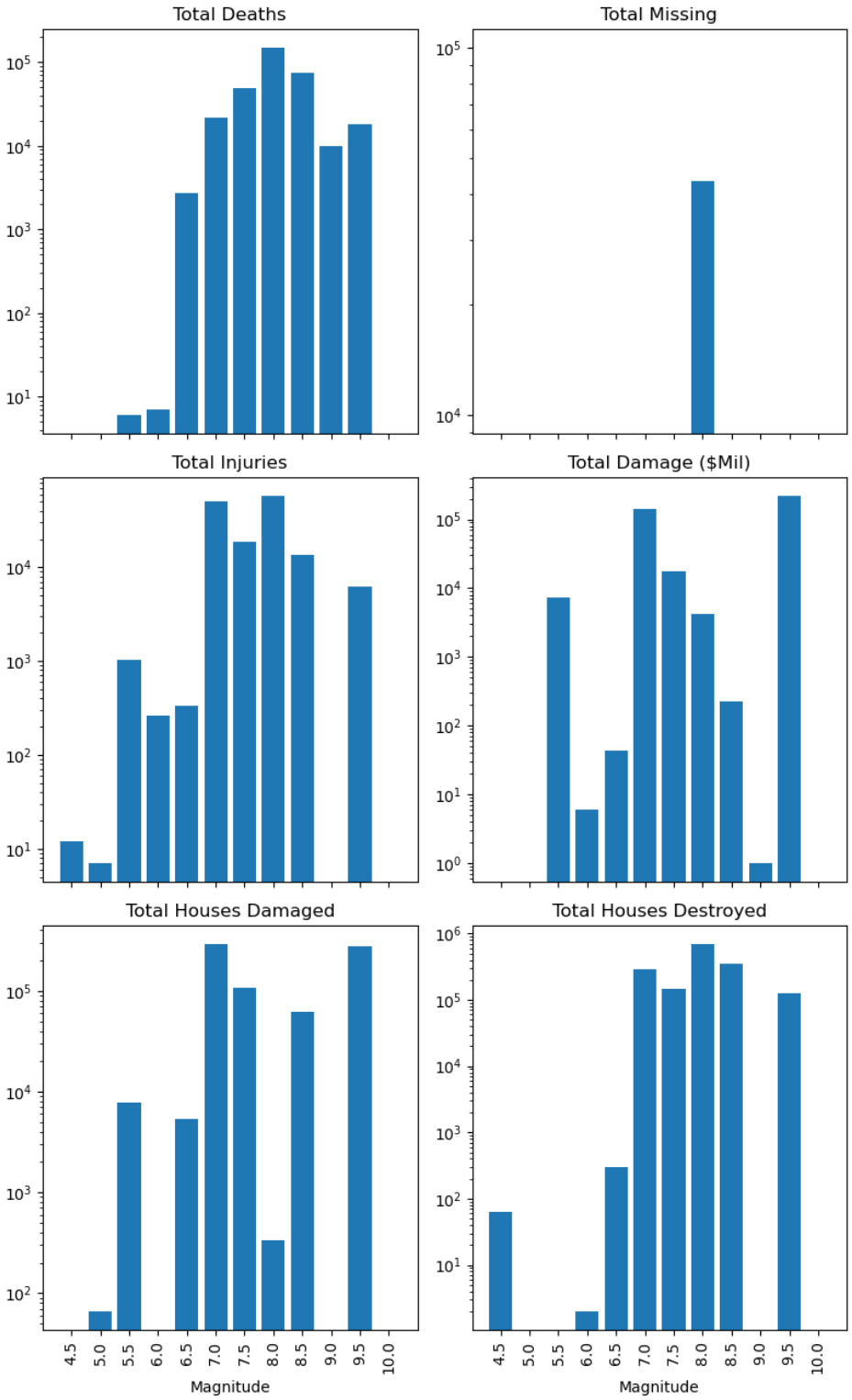


Figure 14: Socio-economic Consequences of Destructive Earthquakes in Japan Between 887 and 2022

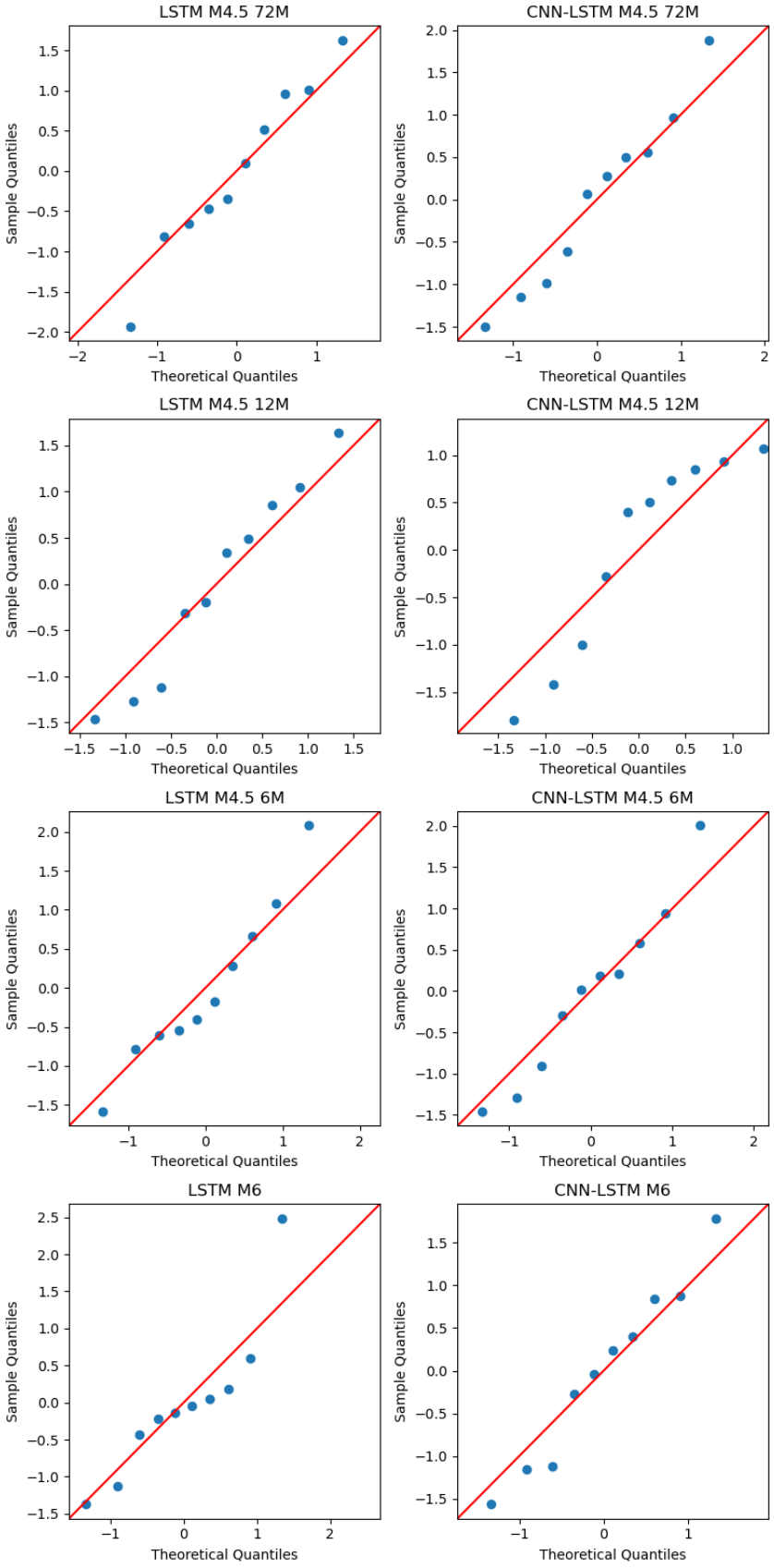


Figure 15: Q-Q plots of LSTM and CNN-LSTM using a cutoff magnitude of $M.45$ and lookback-windows of 72, 12, and 6 months (top six graphs). Bottom two graphs represent LSTM and CNN-LSTM using a cutoff magnitude of $M6$ and lookback-window of 72 months.