Artificial intelligence to detect and forecast earthquakes

VEDA LYE SIM ONG

Artificial Intelligence to Detect and Forecast Earthquakes

Veda Ong

A Thesis presented for the degree of
Master of Science by Research

Department of Earth Sciences
Durham University
United Kingdom

November 2020

Preliminary

Earthquake prediction has been a long standing goal in seismology. Despite significant effort, no statistically rigorous application exists involving the use of precursory phenomena to forecast large earthquakes. As a result, reported cases of precursors can be attributed to random noise or chance. The ability to robustly identify precursory signals and accurately attribute them to large moment magnitude (Mw) earthquakes could significantly improve our hazard preparedness. This report analyses the raw seismic signal prior to 31 Mw $\geq$ 6 earthquakes in the Japan region using deep neural networks. The neural network successfully detected short-term changes in the seismic signal correlated to the investigated earthquakes. This raises interesting questions for future research. The first is whether these 'precursors' can assist in reliably forecasting the timing, location and Mw of an impending earthquake. The second is the origin of the precursors, for example, what process generates them, and whether they can provide clues on the mechanics of fault slip during the earthquake cycle.

Artificial Intelligence to Detect and Forecast Earthquakes

Veda Ong

Abstract: Precursors to large earthquakes have been widely but not systematically identified. The ability of deep neural networks to solve complex tasks that involve generalisations makes them highly suited to earthquake and precursor detection. Large Mw earthquakes and associated tsunamis can have a huge economic and social impact. Detecting precursors could significantly improve seismic hazard preparedness, particularly if precursors can assist, within a more general probabilistic forecasting framework, in reducing the uncertainty interval on expected earthquakes' timing, location and Mw. Additionally, artificial intelligence has recently been used to improve the detection and location of smaller earthquakes, assisting in the completion and automation of seismic catalogues.

This paper is the first to present a deep learning-based solution for detecting and identifying short-term changes in the raw seismic signal, correlated to earthquake occurrence. Deep neural networks (DNNs) were employed to investigate the background seismic signal prior to 31 Mw $\geq$ 6 earthquakes in the Japan region. Instantaneous, precursor-related features (features correlated to the investigated earthquakes) were detected as opposed to predicting future values based on previously observed values in the case of time series forecasting. The network achieved a 98% train accuracy and a 96% test accuracy classifying noise unrelated to Mw $\geq$ 6 earthquakes from signal immediately prior to the investigated earthquakes. Additionally, the precursor-related features became increasingly systematic (more frequently detected prior to the investigated earthquakes) with earthquake proximity. Discriminative features appeared most dominant over a frequency range of $\sim$ 0.1-0.9 Hz, coinciding with microseismic noise and recent observations of broadband slow earthquake signal (Masuda et al. 2020). In particular, frequencies of $\sim$ 0.16 and $\sim$ 0.21 Hz provided significant precursor-related information.

Deep learning successfully detected features of the seismic data correlated to earthquake occurrence. Developing a better understanding of the origin of the precursor-related features and their reliability is the next step towards establishing an earthquake forecasting system.

Declaration

The work in this thesis is based on research carried out in the Department of Earth Sciences at Durham University. No part of this thesis has been submitted elsewhere for any degree or qualification.

The code used in this thesis was developed in Python by the author using specialised AI libraries Keras and Tensorflow.

Acknowledgements

I would like to sincerely thank Professor Stefan Nielsen and Dr Stefano Giani for being patient, time giving and wonderful supervisors. I am truly fortunate to have had the opportunity to work with you both and I am extremely grateful for all of your help and encouragement.

I would also like to thank my parents for providing continuous advice and support and Joshua Ellis for never failing to help me when I needed it and for always being optimistic and reassuring.

Contents

List of Figures

Chapter 1

Introduction

Earthquake prediction has been a long-standing goal in seismology. Currently, a reliable method for short-term (minutes-weeks) earthquake prediction does not exist. The ability to identify systematic earthquake precursors could have a significant effect on hazard preparedness and subsequently reduce the impact of highly destructive earthquakes.

Probabilistic forecasting of earthquakes is extremely limited and pre-earthquake processes are poorly understood. The epidemic-type aftershock sequence (ETAS) model is currently the most popular model to describe seismicity in a region (Ogata 1988). The model presents the idea that large earthquakes trigger more numerous aftershocks which are accompanied by background earthquakes. This method does not rely on empirical observations of precursors but provides a forecast based on a likelihood calculation (Kattamanchi et al. 2017). Precursory phenomena, such as short term earthquake clustering attributed to increased foreshock activity (Tamaribuchi et al. 2018) and changes in seismic velocity (Brenguier et al. 2008, Chen et al. 2010) have been observed prior to some large earthquakes, however, these observations frequently lead to false predictions and are therefore considered unreliable. Diagnostic precursors specific to a small Mw and location range could provide significant information for earthquake forecasting if identified systematically prior to earthquakes occurring within those ranges. Precursors not constrained to a Mw or location range could be more reliable when used in combination with other information such as typical earthquake recurrence in the investigated region.

A variety of approaches have been applied to earthquake forecasting. Earlier methods have involved forecasting earthquakes based on their recurrence intervals and statistical patterns (Nishenko & Buland 1987). Other, more popular approaches, rely on empirical observations of precursory changes (Turcotte 1991, Lomnitz 1994, Scholz 2002, Allen & Kanamori 2003). The empirical approach considers a variety of observations, ranging from precursory seismic activity (Brenguier et al. 2008) and electromagnetic fluctuations (Kamiyama et al. 2016) to chemical emissions (Prayogo et al. 2015) and anomalous animal behaviour (Hayakawa et al. 2016, Yamauchi et al. 2014). Although frequently occurring prior to some large earthquakes, these methods have not provided reliable, short-term forecasts on a consistent basis (Holliday et al. 2005). For example, it is understood that many large, natural earthquakes are preceded by slow slip and foreshock sequences (Passelègue et al. 2017). There is, however, difficulty in using slow slip events and earthquake swarms (foreshocks) to predict natural earthquakes as they do not occur systematically prior to large earthquakes and therefore cannot be considered for short-term prediction (Dublanchet 2018). Currently, the majority of reported earthquake precursors are non-seismological.

More recently, laboratory (lab) experiments have shown systematic changes prior to lab earthquakes. Earthquake precursors are thought to arise when faults reach critical stress conditions preceding shear failure (Rouet-Leduc et al. 2017). Prior to fault failure, lab earthquakes have shown an increase in small shear failures, each of which emit impulsive acoustic emissions (Rouet-Leduc et al. 2017). Systematic changes in elastic wave speed and acoustic transmissivity have also been observed prior to lab fault failure (Shreedharan et al. 2020, Scuderi et al. 2016). More recently, machine learning, a field used to analyse the statistical characteristics of large quantities of data, has been used to investigate changes in the acoustic signal emitted from lab fault zones (Rouet-Leduc et al. 2017). This technique enabled highly accurate prediction of lab earthquakes by identifying a signal emitted from the fault zone that was previously thought to be low amplitude noise. Although providing concrete evidence of precursors, results obtained from lab experiments cannot capture the entirety of the complex physical processes that occur during a natural earthquake.

The same machine learning method was applied to seismic data from the Cascadia sub-

duction zone (Rouet-Leduc et al. 2019). By posing the problem as a regression between the statistical characteristics of the continuous seismic data and the surface GPS displacement rate, the study showed that the Cascadia megathrust continuously emits a tremor-like signal with statistical characteristics that reflect the displacement rate on the fault. Although providing real-time access to the physical state of the slowly slipping portion of the megathrust, this method has not been applied to fast earthquake prediction.

Systematic precursors to fast earthquakes are yet to be identified (Mignan & Broccardo 2019). Difficulty in identifying natural precursors arises partly from the fact that without knowing the location of an impending earthquake, efforts cannot be focused towards detecting changes in the properties within and surrounding the fault zone prior to failure (Scuderi et al. 2016). Furthermore, the Mw of precursors is likely to be much smaller than the Mw of the event and thus precursors will often go unrecorded or unidentified by seismometers (Rouet-Leduc et al. 2017). Additionally, precursors may often be masked by other earthquakes or earthquake swarms which are characterised by entirely different statistical properties (Ishibashi 1988). There is hope that the significant increase in station density and sensitivity over the last 15 years will lead to advances in earthquake forecasting and precursor detection (Rouet-Leduc et al. 2017).

Deep learning is a subset of machine learning used to extract high level features from raw data. Multiple hidden layers of highly interconnected neurons, analogous to neurons in a biological brain, form deep neural networks (DNNs) which have shown superior performance in discovering complex patterns within very large datasets (Noh et al. 2015). A major advantage of using DNNs, is that, generally, there is no need for feature extraction or any significant pre-processing. As a result, large quantities of data can be used directly to train the neural network. A type of artificial neural network (ANN) known as a convolutional neural network (CNN) has shown excellent performance in various visual recognition problems such as image classification, object detection and semantic segmentation (Zhao et al. 2019). The task of recognising changes within a waveform time-series is very similar to that of recognising objects in 2D images. A significant advantage of using a CNN for seismic precursor detection is that CNNs can detect features of any scale (Zhao et al. 2017). As a

result, even a very small change in the signal close to an earthquake could be detected.

CNNs have frequently been applied to earthquake detection, generating improved earthquake catalogues by efficiently analysing large quantities of seismic data (Nguyen et al. 2017, Perol et al. 2018, Mousavi et al. 2019). However, research into the potential of CNNs and complex neural network architectures to improve earthquake predictability is extremely limited. Huang et al. (2018) utilised a simple CNN to investigate the seismic data prior to earthquakes in Taiwan. Taiwanese seismicity maps were transformed into 2D images by encoding earthquake Mw as brightness. A classification-based approach was employed to detect differences within seismicity maps up to 30 days prior to large earthquakes with a $(Mw) > 6$ and seismicity maps up to 30 days prior to small earthquakes ($Mw < 6$). Their algorithm led to an R-score of 0.303 where an R-score of 0 is the result of an entirely random prediction and an R-score of 1 is an entirely successful prediction. An R-score of 0.303 suggests that the CNN captured some precursory seismicity patterns, however, no further investigation was conducted into the patterns which led to this classification result. In addition, these results were not considered for probabilistic forecasting of earthquakes.

Encouraged by success in classification problems, researchers have also frequently applied CNNs to solve structured prediction problems. A commonly used example of this is semantic segmentation which outputs a prediction for each pixel in an image. The task of segmenting time series data is similar to image segmentation whereby a prediction is made for each time sample instead of per pixel. One approach, achieving a high accuracy and recall rate, involved the use of semantic segmentation for the detection of P and S phases in seismic data (Zhu & Beroza 2019).

Although CNNs are commonly used on 2D images (Huang et al. 2018), this research will investigate precursors based solely on features of the raw seismic signal. To our knowledge, no current research exists involving the use of neural networks to detect precursors in lab and real earthquake settings through analysis of raw time series data. Instead of using decision trees to detect changes in the statistical features of the signal as in (Rouet-Leduc et al. 2017), neural networks can be used to detect systematic, pattern-based changes in raw time series. As a result, this investigation will determine whether precursors can be detected without

any substantial data pre-processing. Lab data is obtained in a controlled environment and is therefore considerably less complex than real earthquake data. Initial analyses will involve the simplest task of detecting earthquakes in lab data. This will provide a good preliminary investigation for enabling a greater understanding of the application of CNNs (typically applied to 2D datasets) on time series. The techniques used here can then be tailored to the more complex tasks of detecting precursory phenomena in lab and real earthquake data.

Lab studies have shown systematic precursors to earthquake-like, frictional failure, however, whether such changes can be detected in real earthquake settings and used to forecast failure remains unanswered (Scuderi et al. 2016). Due to the widespread availability and abundance of seismic data, by identifying systematic precursors within seismic data as opposed to, for example, within electromagnetic emissions, this method could be applied to investigate precursors in other seismically active regions where seismic data is readily available. In addition, instead of considering long-term changes such as decreases in seismic wave speed and increased foreshock activity which do not systematically occur prior to large earthquakes, this project will focus on short-term fluctuations and attempt to detect patterns within the seismic data that occur over a smaller time-frame (minutes to hours) prior to large earthquakes. Focusing on a smaller time-frame and training a complex CNN for classification may more robustly enable the detection of previously undiscovered patterns in seismic signals. Finally, due to the success of semantic segmentation on time series data (Zhu & Beroza 2019), the technique will be applied to precursor detection. No prior work has used semantic segmentation to investigate earthquake precursors.

1.1 Thesis Aims

The aims of this thesis work are summarised as follows:

1. To enable the detection and forecasting of laboratory earthquakes (Chapter 3).
2. To detect and potentially identify systematic changes in seismic data that could be attributed to large (Mw $\geq$ 6) earthquake precursors and to test the potential of the algorithm for probabilistic earthquake forecasting (Chapter 4).

Chapter 2

Theory

2.1 Deep Neural Networks

All neural networks consist of an input layer, one or several hidden layers (layers located between the input and output of the network) and an output layer (Figure 2.1). A deep neural network (DNN) comprises a collection of neurons organised in a sequence of multiple layers. An individual neuron is a mathematical function that models the functioning of a conceptual biological neuron (Figure 2.2).

Within the field of machine learning, there are two main types of tasks: supervised, and unsupervised. In supervised learning tasks, each input to the neural network is associated with a label or ground truth which indicates the class of that input. The goal of supervised learning is to learn a function that, given a sample of data and desired outputs, best approximates the relationship between inputs and outputs (Donalek 2011). Unsupervised learning, on the other hand, does not have labelled outputs, so its goal is to infer the natural structure present within a set of data points (Zhao & Liu 2007). This thesis investigates a supervised learning approach (Figure 2.1).

The input layer of a neural network is responsible for receiving the inputs and transferring this information to the subsequent layers. It does not apply any operations on the input values. In a fully connected neural network, each hidden layer comprises several neurons where each neuron is connected to all neurons in the adjacent layers (Figure 2.1). It should be

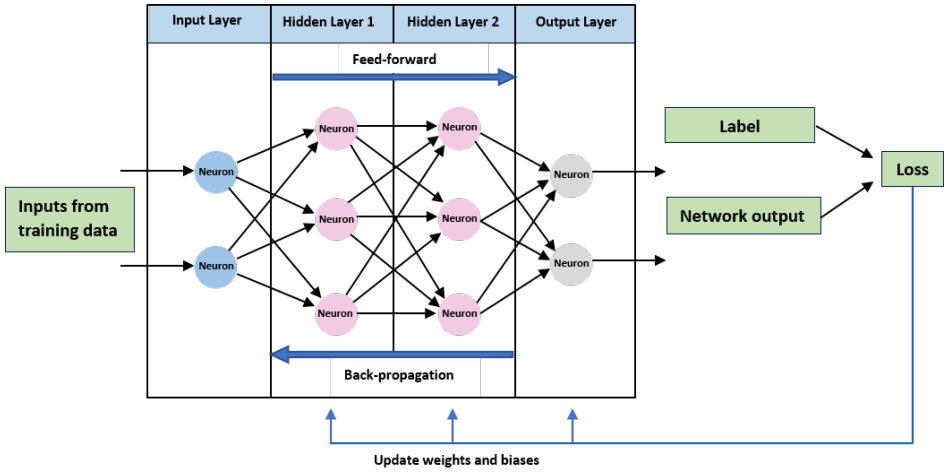

Figure 2.1: Simple DNN with 2 fully connected layers | Arrows between neurons represent their weighted connections. In the feed-forward stage, the input layer passes the inputs to the neurons in the first hidden layer. Within each neuron, the weighted sum of the inputs is calculated, a bias is added, and this value is passed through a non-linear activation function before leaving the neuron (Figure 2.2). Each neuron is connected to all neurons in the adjacent layers, representing a fully connected network. The output is formed in the output layer with the number of neurons representing the number of classes (2 in this network). The network output (prediction) is compared to the label or ground truth associated with the input. The loss (difference between the label and the prediction) is used to update the weights and biases in the learnable layers of the network.

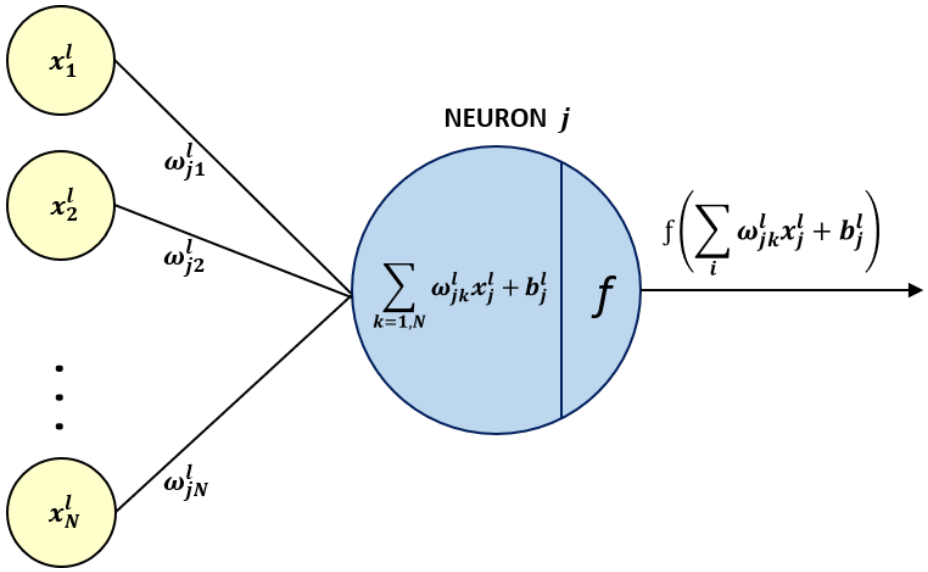

Figure 2.2: Operations within a single neuron | Each neuron in some layer l receives a set of activations (x_N^l) as input from neurons in the previous $(l-1)^{th}$ layer. Each input is associated with a weight (w_{jk}^l) which is the weight from the k^{th} neuron in the $(l-1)^{th}$ layer to the j^{th} neuron in the l^{th} layer, and a bias (b_j^l) which is the bias of the j^{th} neuron in the l^{th} layer. The weights and biases are updated during the learning process. In each iteration (forward pass), each neuron calculates a weighted average of the inputs and adds a bias. This result is passed through a non-linear activation function (f).

noted that individual neurons within a single layer are entirely independent of other neurons from the same layer and do not share connections with them.

Neurons receive as input the activations or outputs of the neurons from the previous layer. Each neuron performs a calculation to map its input to its output (Figure 2.2). Each output-input link between two individual neurons is associated with a weight which represents the strength of the connection between units. Within each neuron, the weighted sum of the inputs is calculated and a bias is added to the result. The bias is a constant value or vector whose function is to shift the result to the positive or negative side (Goodfellow et al. 2016). The weights and biases are the learnable parameters of the network and are randomised prior to training.

In a given layer 'l' (layer index is skipped here for simplicity), the calculation in Figure 2.2 for several 'j' neurons and 'N' inputs can be expressed by matrix multiplication as:

$$f\left(\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,N} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{j,0} & w_{j,1} & \cdots & w_{j,N} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_N \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_j \end{bmatrix}\right) \tag{2.1}$$

The weighted sum plus the bias are fed into an activation function that applies a non-linearity and normalises the input (Figure 2.2 and Figure 2.3). The activation function is responsible for transforming the summed, weighted input from the neuron into the activation of the neuron or output for that input. The neurons of the network jointly implement a complex nonlinear mapping from the input to the output (Wang et al. 2018). The output layer produces an output of the desired length. For example, a 2-class classification will have 2 output neurons.

A commonly used activation function is the rectified linear unit (ReLU) activation (Figure 2.3), introduced by (Hahnloser et al. 2000) which maps the output of a layer by the function shown in Equation 2.2

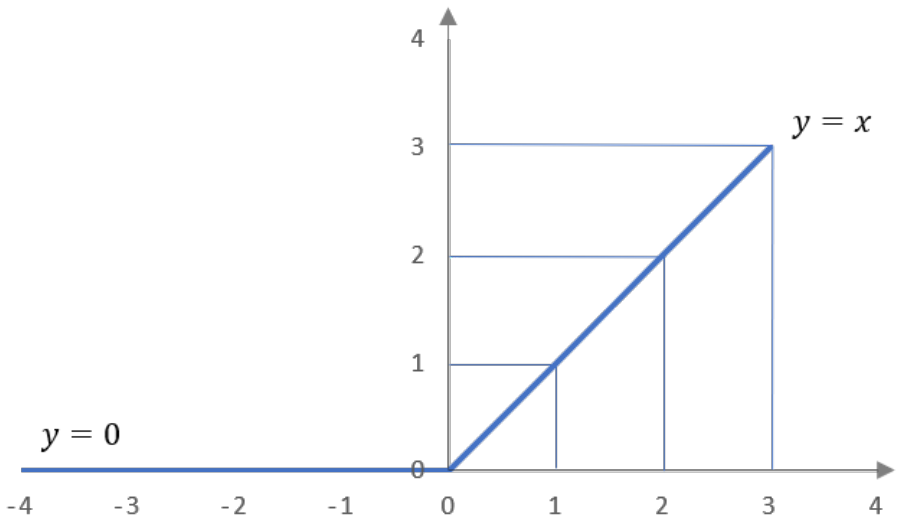

Figure 2.3: Visual representation of the ReLU activation function.

$$f(x) = \left\{ \begin{array}{ll} 0, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{array} \right\} \tag{2.2}$$

The ReLU activation function is a simple calculation that returns the value provided as input, or the value 0 if the input is 0 or less. The function is linear for values greater than zero, yet it is a non-linear function as negative values are always output as zero (Figure 2.3).

The non-linear activations within the neurons allow for complex input-output relations to be learnt through an optimisation process (Noh et al. 2015). The non-linear mapping is learned from the data by adapting the weights (w) and biases (b) in the network using error backpropagation, a concept further discussed below.

Prior to training the network, the data is split into train, test and validation datasets. Supervised learning is a type of learning where the dataset has been labelled, i.e. the expected output for each input already exists. The network is trained on the training dataset. The test dataset is used to validate the learned weights and biases during training to see how well the network performs on unseen data with each iteration. The validation dataset is used to test generalisation of the model after training and is commonly used to compare the performance of different models (networks).

Forward propagation is the process of feeding input values (x) from the training dataset to the neural network and obtaining an output or prediction ($\hat{y}$). The predicted output is

compared to the actual output or ground truth (y) and a cost function is calculated (Figure 2.1). A loss function computes the error for a single training example while the cost function is the average of the loss functions for all the training examples. The goal of the algorithm is to minimise the cost function. An example of a simple cost function (C) is the sum of the squared differences. With N training examples, this cost function is shown in Equation 2.3. The squared differences increase the error distance, thus making the poor predictions more pronounced compared to the successful predictions (Shang et al. 2016). Within Equation 2.3, $\mathbf{y}$ is a vector of true labels ($\mathbf{y}$ = [target(x), target(x)... target(x_N)]) and $\hat{\mathbf{y}}$ is a vector of network predictions.

$$C(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (\hat{y}_i))^2 \tag{2.3}$$

The learning process is aimed at minimising the cost function. This is implemented by gradient descent, an iterative algorithm used to find the minimum value of a differentiable function (Mousavi et al. 2019). As the cost function is continuous and differentiable, it comprises a continuous landscape of highs and lows or maxima and minima respectively. Gradient descent computes the gradient of the cost function with respect to the weights and biases for the entire training dataset (Equation 2.4). To consider the impact of each trainable parameter on the final prediction, partial derivatives of the cost function with respect to each weight and bias are calculated and saved in a gradient vector (∇) that has as many dimensions (n) as weights and biases (Equation 2.4). This gradient is used to calculate new weights and biases from current weights (w) and biases (b) by indicating the direction to move to update the parameters (Equation 2.5 and 2.6).

Combining the weights and biases into the learnable parameters of the network (x_1, x_2, $\cdots$,x_n), the gradient can be calculated as

$$-\nabla C(x_1, x_2, \cdots, x_n) = \begin{bmatrix} \frac{\partial C}{\partial x_1} \\ \frac{\partial C}{\partial x_2} \\ \vdots \\ \frac{\partial C}{\partial x_n} \end{bmatrix} \tag{2.4}$$

This gradient is used to update the weights and biases. Each weight and bias is changed by a certain amount.

$$w^l_{jk} = w^l_{jk} - \eta \frac{\partial C}{\partial w^l_{jk}} \tag{2.5}$$

$$b^l_j = b^l_j - \eta \frac{\partial C}{\partial b^l_j} \tag{2.6}$$

Parameters are updated in the direction of the gradients with the learning rate (η), determining how big of an update is performed at each iteration. The lower the learning rate, the more precise the convergence but the slower the learning (Ross, Meier & Hauksson 2018).

The weights and biases of the network are updated after each forward pass of data through the network. During backpropagation, gradient descent is used to update the parameters within each trainable layer of the neural network. The parameters of the network are empirically optimised with large amounts of data, such that a given input leads to an output that is as close as possible to the desired output or ground truth.

These hierarchically organised, non-linear mapping functions are typically used for tasks such as classification and regressions. They enable the development of robust, generalised representations of extremely large datasets (Mousavi et al. 2019). As opposed to using explicit template matching to search for patterns in time series, these algorithms are able to detect the general characteristics that individual examples of that class share. This non-explicit nature of neural networks enables them to reliably detect patterns without ever having seen an exact or even similar example (Ross, Meier, Hauksson & Heaton 2018).

2.2 Convolutional Neural Networks

The convolutional neural network (CNN) is a learnable feature extraction system that works together with a fully connected network for classification and regression tasks (Figure 2.4). The CNN can be considered a feature extraction system that distils the relevant information from the input. This information is commonly passed to one or several fully connected layers which use the extracted information to produce an output.

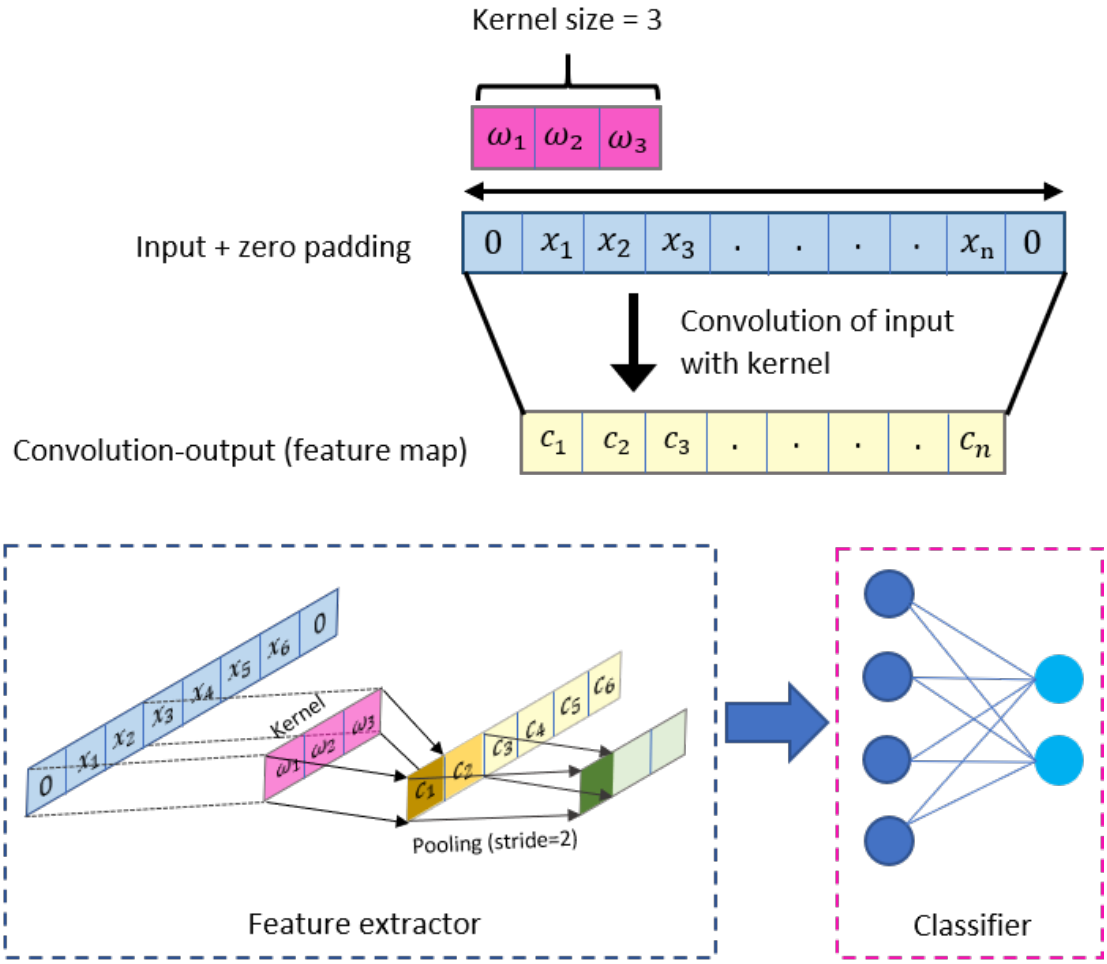

Figure 2.4: Basic structure of a classifier based on a CNN | The CNN is the feature extractor and the classifier uses the output from the feature extractor to assign a class label to each input. During a convolution operation, a convolutional kernel (vector of weights) is convolved with the input to generate a feature map. Several kernels are normally applied to the input, however, for simplicity of illustration, only a single kernel is applied in this example. The input is frequently padded with zeros in order to retain information at the boarders and preserve the input length. The feature extractor may include several convolutional layers, each followed by a pooling layer which strategically downsamples each feature map. Pooling involves selecting a pooling operation, similar to a kernel or filter, which is applied separately to each feature map to create a new set of the same number of pooled feature maps. The length of the pooling operation is smaller than the length of the feature map (length 2 in this diagram). The stride dictates how much the size of the feature map is reduced. A stride of 2 in this example indicates that the pooling layer will reduce the size of each feature map by a factor of 2. When several kernels are applied in a convolutional layer, the feature maps are stacked to form a single feature map before being input to the next convolutional layer. After the final convolution operation, this stacked feature map is input to the classifier. The classifier usually consists of one or several fully connected layers. In this diagram, the classifier consists of a single fully connected layer and an output layer. The number of output neurons in the output layer depends on the number of classes (2 in this example). Note, this is a one dimensional CNN, typically used for time series analysis.

Each convolutional layer contains several learnable filters (or kernels), each of which comprise a vector of weights (Figure 2.4). In the first or shallowest convolutional layer of the network, each filter is convolved with the input, producing an activation map that indicates the responses of that filter at every spatial position (Equation 2.7). The activation maps store information on the location of features within the input and how well those features correspond to the filters (Zhu et al. 2019). In simpler terms, they summarise the features detected in the input. The number of filters applied at any one convolutional layer is equivalent to the number of activation (feature) maps produced at that layer. These output feature maps along with a bias are passed through an activation function and stacked to form a single output feature map as input to the next convolutional layer (Figure 2.4).

The general expression of discrete convolution in 1D is:

$$g(x) = [w * f](x) = \sum_{dx=-a}^{a} w(dx) f(x+dx) \tag{2.7}$$

Where $g(x)$ is the output of convolution of the filter or kernel (w) with the input (f), x and dx denote the index of the discrete position in the series and $-a$ and a are the first and last indexes, respectively, of the convolutional kernel. Equation 2.7 refers to the operation in a convolutional layer. In comparison, Equation 2.1 indicates the operation in a fully connected layer. The convolution operation is applied to identify patterns of interest anywhere within the data. The coefficients of the filters are initialised randomly and are optimised, along with biases, during the training process.

Additional layers which often occur within a convolutional network are pooling and normalisation layers. After convolution, pooling layers are used to down sample the convolution output so that subsequent layers learn attributes of a rescaled representation of the data (Figure 2.4). In Figure 2.4, the pooling layer maps two activations on the feature map to a single activation. This helps to recognise variants of the same features with different sizes. Another key concept of a pooling layer is to provide translational invariance.

Batch normalisation layers normalise the activations of a convolutional layer during training. Batch normalisation is applied to speed up the convergence of the network and help improve generalisation. A typical convolutional block may consist of a convolutional layer

followed by an activation layer, a pooling layer and a batch normalisation layer. One or several of these blocks form a convolutional neural network. As the depth of the convolutional network increases, convolutional layers extract higher level (finer scale and more complex) features (Zhu & Beroza 2019).

The receptive field is an important concept to consider when designing a CNN architecture (Figure 2.5). It is defined as the region in the input space that a particular CNN's feature is observing or is affected by (Meier et al. 2019). Unlike FCNNs where the value of each neuron depends on the entire input to the network, a unit (activation of a feature map) in a CNN only depends on a region of the input, defined as the receptive field. Since any region of the input outside the receptive field of an activation does not affect the value of that activation, it is necessary to carefully control the receptive field to ensure that it covers the relevant input region (Rojas et al. 2019). The receptive field can be increased by stacking more convolutional layers to make the network deeper or using subsampling/pooling operations.

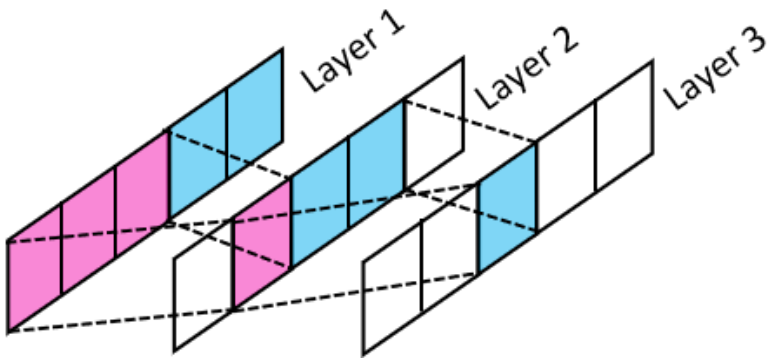

Figure 2.5: The receptive field of each convolutional layer with a kernel of length 3 | The pink area marks the receptive field of one activation in layer 2 and the pink and blue area marks the receptive field of one activation in layer 3.

CNNs have shown excellent performance in various visual recognition problems such as image classification, object detection and semantic segmentation. The task of detecting precursors in a waveform time-series is very similar to that of recognising objects in 2D images (earthquake seismograms can be viewed as 1-dimensional images with 3 components) (Meier et al. 2019). CNNs are extremely powerful in performing vision-based tasks, making them well suited for detecting patterns that may signify an impending earthquake. A major

advantage of using a CNN for precursor detection is that CNNs can detect features of any scale or target objects of any scale (Mignan & Broccardo 2020). So, with enough layers in the network, even if there is a very small change in the signal leading up to an earthquake, the CNN may be able to detect it. CNNs excel at performing pattern recognition that is invariant with respect to translation, scaling and other distortions, a weakness of DNNs and other machine learning methods (Renna et al. 2019).

2.3 Semantic Segmentation

Semantic segmentation is formulated to produce a prediction for each data point that is input to the neural network. A popular approach is for the network to follow an encoder/decoder structure (Figure 2.6). In this scenario, the network consists of two parts: a convolutional network (encoder) and a deconvolutional network (decoder).

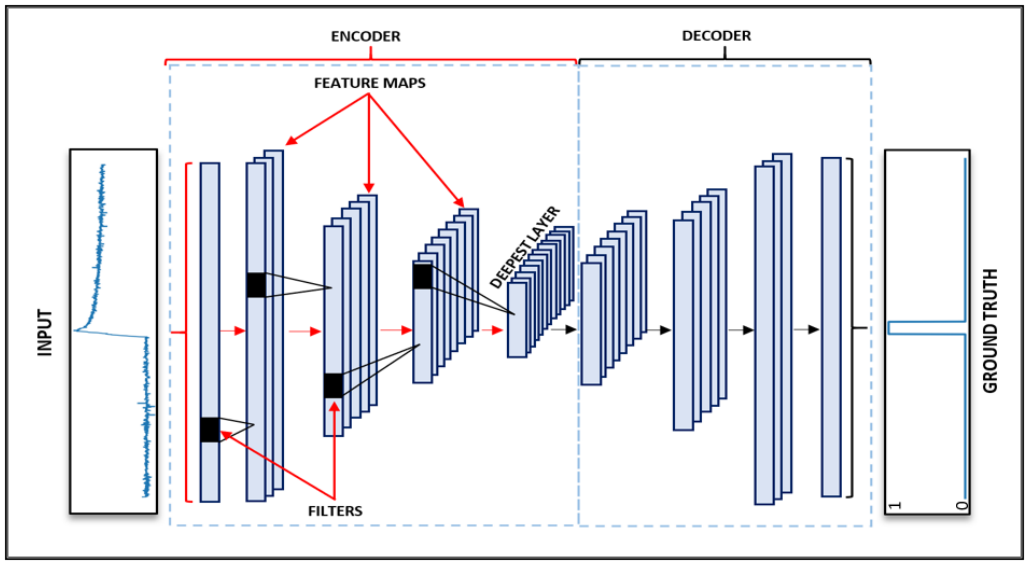

Figure 2.6: Encoder-decoder neural network architecture for semantic segmentation | The input in this figure is a small section of the lab data containing a single slip event. The ground truth is a plot of the labels for each time sample in the input (0 = noise, 1 = earthquake).

The convolutional network corresponds to the feature extractor and transforms the input to a multidimensional feature representation. The deconvolutional network maps these low resolution feature maps to full input resolution for point-wise classification (Wang et al. 2018). The final output of the network is a probability map of the same dimensions as the

input, indicating the probability that each data point belongs to one of the predefined classes. The probability map is compared to the ground truth, a cost is calculated where the cost is the average sum of the losses for each prediction and this value is used to update the weights and biases. The decoder of the network is key to a sample-wise detection as it traces the input locations with strong activations back to image space (Long et al. 2015). It effectively reconstructs the detailed structure of the time series in finer resolutions, seeking to localise properties of the time series into its classes.

In dense prediction tasks such as semantic segmentation, it is critical for each output pixel to have a large receptive field, such that no important information is left out when making the prediction (Chen et al. 2018). As earlier discussed, the receptive field can be increased by increasing network depth and by subsampling the input. Downsampling occurs in the encoder of the network to develop lower resolution feature maps which are highly efficient at discriminating between classes. The feature representations are upsampled in the decoder.

Pooling in convolutional networks is designed to filter noisy activations in a lower layer by abstracting activations in a receptive field with a single representative value (Zhao et al. 2017). This improves classification tasks by retaining only robust activations in upper layers, however, spatial information is lost during the process which may be critical for precise localisation required for semantic segmentation. To resolve this issue, unpooling layers in the decoder perform the reverse operations of the pooling in the encoder. This is achieved by recording the locations of maximum activations selected during pooling operations and using this information to return each activation back to its original location (Badrinarayanan et al. 2017). Deconvolutional or transpose convolutional layers densify the sparse activations obtained by unpooling by applying convolution-like operations with multiple learned filters. The output of the deconvolution and unpooling is an enlarged and dense activation map.

During training, the learned filters in deconvolutional layers aim to reconstruct the shape of objects in an image or patterns within a time series (Chen et al. 2018). Hence, similar to the convolutional encoder, a hierarchical structure of deconvolutional layers is utilised to capture different levels of shape information. Filters in lower layers may identify overall shapes in the time series while class-specific, fine details are encoded in filters in higher

or deeper layers. In this way, the network considers class-specific shape information when making a prediction.

If the input in Figure 2.6 was processed by a CNN for classification, the output would be a single prediction (i.e. the whole input would be classified as earthquake signal or as noise signal). This prediction gives no indication of where the earthquake occurs. In the case of segmentation, the network is optimised to fit the ground truth. As a result, segmentation produces a much more detailed view of what is in the input. Additionally, classification tasks rely heavily on a significant amount of data in each class. This requirement is not typically fulfilled when analysing earthquake data.

Semantic segmentation is a key application in image processing and the computer vision domain. Although infrequently applied to time series, semantic segmentation has shown recent success in detecting seismic phases within time series data (Zhu & Beroza 2019).

Chapter 3

Laboratory Methods and Results

3.1 Introduction

To our knowledge, precursors in lab data have not been investigated with the use of neural networks. Unlike other machine learning techniques such as decision trees (Rouet-Leduc et al. 2017), neural networks enable the analysis of raw time series. As a result, this chapter investigates the potential of neural networks to detect precursors in raw lab data. A semantic segmentation algorithm is used to analyse the data with the aim of detecting individual slip events and their precursors. The simpler task of earthquake detection was carried out prior to precursor detection and was used as a preliminary investigation to develop a segmentation network that performed well on time series data (as opposed to the 2D data typically input to a semantic segmentation algorithm). In this chapter, we analyse strain gauge data obtained during a triaxial loading experiment performed on a pre-cut granite sample (Figure 3.1). Strain gauge data were investigated as it provided a good measure of the elastic and anelastic deformation induced by the applied axial stress. Additionally, strain gauge data has been effective at observing the nucleation process (preparatory phase prior to an earthquake) (Buijze et al. 2020) which may be significant to the investigation of precursors.

3.2 Experimental Set Up

A triaxial loading cell was used to impose a constant strain rate of 1 μm/s on the pre-cut sample (Figure 3.1). A triaxial experiment was conducted as it enabled the testing of a sample under a high confining pressure and axial stress, simulating the conditions that occur during a natural earthquake. The confining pressure ($\sigma_2 = \sigma_3$) was fixed at 50 MPa and the sample was loaded axially (σ_1) until stick-slip events started to develop on the pre-cut fault.

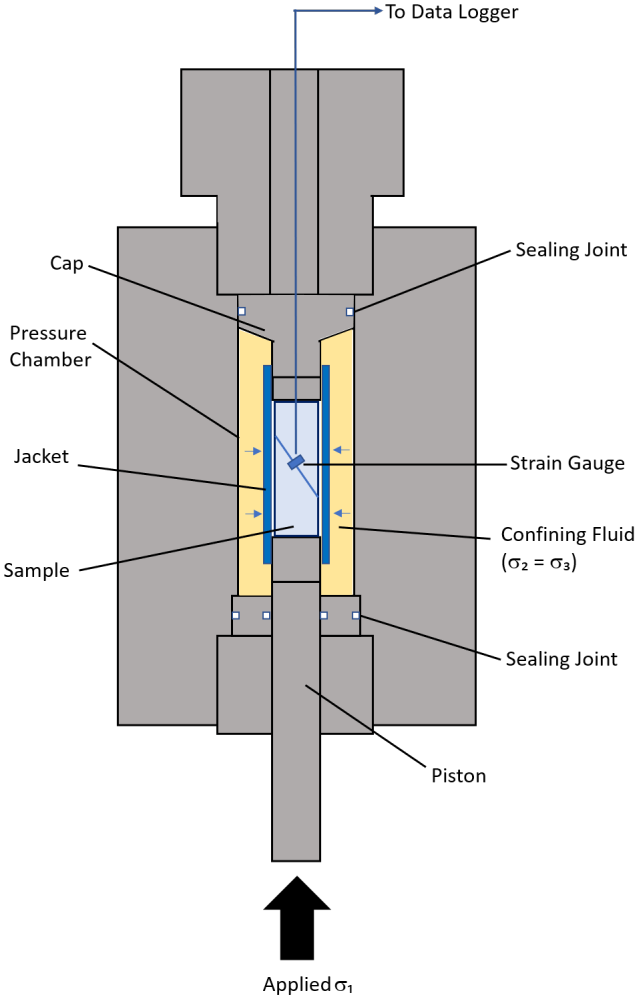

Figure 3.1: Experimental setup | Diagram of the triaxial cell. The triaxial cell hosts the sample and allows a high confinement pressure and vertical load to be applied. The cell structure (grey shaded areas) are built in high strength stainless steel and can sustain the maximum confinement pressure with negligible deformation. A neoprene jacket isolates the sample from the pressure chamber, which contains silicon oil. Four strain gauges were positioned across a pre-cut fault on the sample, however, only one is shown here for simplicity. The piston applied the axial stress (σ_1) used to load the sample. The confining stress ($\sigma_2 = \sigma_3$) was applied by pressurising the cell fluid surrounding the sample.

3.3 Sample Preparation

The sample consisted of a cylinder of Westerly granite with a diameter of 20 mm and a length of 50 mm. The cylinder was pre-cut diagonally across the sample to create a weak fault interface. Once assembled, the sample was insulated from the confining oil medium by a neoprene jacket (Figure 3.1). Four strain gauges were positioned across the fault. The strain gauge signal, corresponding to a deformation, was continuously recorded at a sampling rate of 10 MHz. This high frequency sampling enabled any dynamic stress-strain changes to be monitored and recorded. Strain gauge signals provide good estimates of the sample elastic constants and the dynamic evolution of the differential stress ($\sigma_1 - \sigma_3$) during dynamic rupture propagation (Passelègue et al. 2017).

3.4 Strain Gauge Dataset

The configuration used in this study was such that when the axial stress was increased, both the normal (σ_n) and tangential stress (τ) acting on the fault increased. When the state of stress reached a critical value corresponding to the peak stress of the fault, τ_c, instabilities occurred, leading to a macroscopic friction drop (τ_c/σ_n). The sample appeared to deform in a series of "stick-slip" events (Scholz 2002). The fault remained essentially locked except for sudden episodes of slip that are thought to be representative of earthquakes (Brace & Byerlee 1966). Seven of these fast slip events were recorded over the duration of the experiment (Figure 3.2). Some strain gauge sensors recorded the data to a higher signal noise ratio (SNR) than others.

3.5 Data Pre-processing

The only pre-processing applied to the strain gauge signal was decimation. Initially, decimation by a factor of 512 was applied to significantly reduce the computational power required to process the data. This was applied to prevent the quantity of data from limiting the complexity of the network architecture. When developing the network, a memory limit was

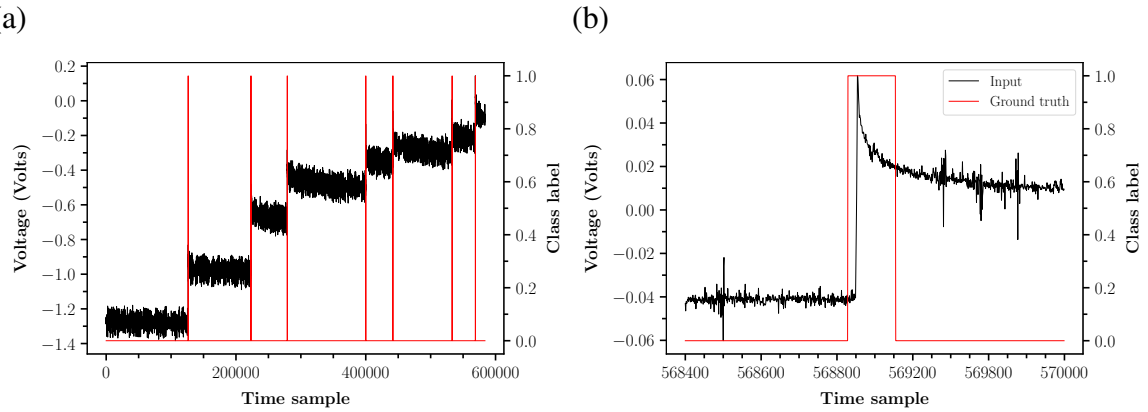

Figure 3.2: Recording from a single strain gauge sensor with a high SNR, decimated by a factor of 512 | (a) Whole dataset where 7 separate events were recorded. The ground truth is plotted with 'ones' representing earthquake signal and 'zeros' representing noise. (b) Single slip event and corresponding ground truth. A gradual increase in the voltage of the signal occurred immediately prior to the slip event in (b). This was also labelled as earthquake signal.

quickly encountered when analysing the data prior to decimation. Decimation by a factor of 512 was only applied to the task of earthquake detection for slip events in the high SNR strain gauge dataset. This decimation reduced the effective frequency to $\sim$ 19 kHz and may have introduced aliasing as the signal was not low-pass filtered prior to decimation. The decimation may have induced artefacts in the signal that filtering prior to decimation would not have, however, analysing the decimated data enabled an understanding of the robustness of the CNN to aliasing and other artefacts. Strain gauge signals preferentially capture lower frequency oscillations, therefore, in the case of a simple detection task, this decimation will unlikely have had a significant impact on the result.

The sample points that represented noise were manually labelled as 'zeros' and those that represented earthquake signal were labelled as 'ones' (Figure 3.2). The sample points representing the earthquake class were defined over a region containing the individual slip events as well the main recovery period post-slip (Figure 3.2b). Clear changes in the voltage of the signal immediately prior to the main slip events were also labelled as earthquake signal (Figure 3.2b). The result is a boxcar function with the same length as the input data (Figure 3.2a).

The ground truth and the whole time series were then split into train (first 75% of the data including the first 5 slip events), test (following 15% of the data containing the next

slip event) and validation (final 10% of the data containing the final slip event) datasets. This prevented any overlap between the datasets and enabled unbiased training and testing of the network. Although the number of different events was extremely small, this was not a significant issue due to the simplicity of this task. Additionally, data augmentation was applied to each dataset to increase the number of windows containing earthquake signal. This is explained in the following 2 paragraphs.

A sliding window approach was used to generate fixed length windows of the whole time series as input to the neural network. This method was selected in order to process the data and perform classification in batches (Liu et al. 2018). To account for the significant class imbalance (large quantity of noise windows compared to earthquake windows), simple data augmentation was applied. Data augmentation is a technique used to artificially create new training data from existing training data. Any applied data augmentation method must be chosen carefully and within the context of the training dataset and knowledge of the problem domain (Perez & Wang 2017). For example, data augmentation by horizontal flipping involves randomly selecting windows from the training dataset and inverting the y axis. This was not a sensible option in the context of time series as flipping the data horizontally would entirely change the features of the input and increase the chances of overfitting. Overfitting occurs when the network learns features of the training data that are different to those in the test and validation data. As a result of overfitting, the training accuracy increases whilst the test and validation accuracy decrease.

Instead of augmenting random windows, the amount of overlap between windows was adapted based on whether the windows contained noise signal only or whether they also contained earthquake signal. By increasing the amount of overlap from 0% (windows containing only noise) to 80% (windows containing earthquake signal), the class imbalance (ratio of noise-labelled time samples to earthquake-labelled time samples) was significantly reduced. Increasing the overlap is equivalent to applying several horizontal shifts to windows of the training data containing earthquake signal (Brownlee 2019). The method successfully increased the amount of training data in the earthquake class without making any changes to the raw data.

The width of the moving windows was 4096 samples. This length was chosen to enable the whole duration of each earthquake to be captured in several time windows. This may have improved the CNN's performance, enabling it to learn information from both before and after the event over the same window of data. In addition, 4096 is a power of 2 (2^{12}) therefore the input could be downsampled by a factor of 2, 12 times. This increased the number of down-sampling operations that could be applied to the input.

The data were prepared for training and testing by standardising each window separately. Standardisation involved rescaling the distribution of values such that the mean of the observed values was 0 and the standard deviation was 1. Standardising windows separately instead of standardising the whole time series before splitting it into windows encouraged the network to learn discriminative patterns within the time series as opposed to amplitude changes. Additionally, to prevent the network from learning anything trend related, the windows in the training dataset were shuffled before being input to the network.

3.6 Network Architecture

An encoder-decoder neural network such as that in Section 2.3 was designed to scan through continuous strain gauge data and classify each sample point in the input windows as one of two classes - 'noise' or 'earthquake'. The network produced two predictions or probabilities for each time sample. The probabilities were obtained by using a Softmax function which mapped the output of the network to a probability distribution over the predicted output classes. The Softmax function is a function that turns a vector of network outputs into a vector of probabilities that sum to 1. The Softmax function can be expressed as:

$$\text{Softmax}(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^{K} \exp(x_j)} \; for \; i = 1, ..., K \tag{3.1}$$

where x is the input vector and K is the number of classes. The argmax function (a function that returns indices of the maximum element of an array in a particular axis) then returned the maximum value of the two probabilities. This method was used to assign each time sample a label from one of the two classes.

The network architecture consisted of the VGG16 classifier (Simonyan & Zisserman 2014) combined with the U-Net encoder-decoder (Ronneberger et al. 2015) (Figure 3.3). Classical U-Net architectures are very popular for detailed segmentation i.e. medical images as they enable precise pixel-level localisation and high segmentation accuracy (Hesamian et al. 2019). Unlike many other segmentation networks, the U-Net has proved very useful for segmentation problems with limited amounts of data (Hesamian et al. 2019). Capable of learning from a relatively small training dataset and enabling precise segmentation, the U-Net was well suited to this task. A U-Net architecture typically consists of a contracting path to capture context (encoder) and a symmetrically expanding path to enable precise localisation (decoder) (Ronneberger et al. 2015).

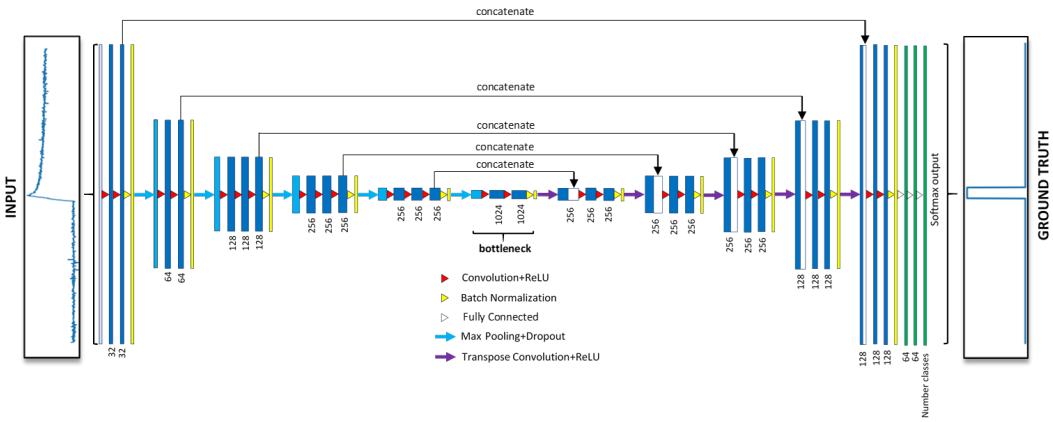

Figure 3.3: U-Net with VGG16 encoder | A window of the dataset is input to the network and the output prediction for each time sample is compared to the ground truth. The average loss is backpropagated through the network using gradient descent and used to update the learnable parameters of the network such as the filter coefficients and biases (Section 2.1). The dark blue layers represent the feature maps produced by convolution operations (Section 2.2). The length of each layer shows the relative feature map size (in time samples) and the number of feature maps produced by each convolution operation are indicated below each dark blue layer. Outputs from batch normalisation and fully connected layers are indicated by yellow and green layers respectively. Light blue layers are the result of applying max pooling to the convolution and batch normalised output (feature maps). Skip connections concatenate encoder layer outputs to their corresponding decoder layer. The number of neurons are shown below each fully connected layer.

The encoder architecture was inspired by VGG16, a network which has achieved great success in image classification (Simonyan & Zisserman 2014). Incorporating the VGG16 classifier in the U-Net increased network depth and complexity which led to an improve-

ment in the network's performance on this dataset. The encoder contained 13 convolutional layers, each followed by a ReLU activation function and dropout, a layer which improves generalisation by randomly excluding neurons during the training process. The feature maps were downsampled using 5 max pooling operations (Figure 3.3). Max pooling is an operation that calculates the maximum value in each specified patch of a feature map. The results of max pooling are down-sampled or pooled feature maps that highlight the most dominant feature in each patch. In VGG16, each max pooling operation halves the feature maps. To construct an encoder from the VGG classifier, the fully connected layers were removed and replaced by two convolutional layers that served as a bottleneck central part of the network, separating the encoder from the decoder (Figure 3.3).

Like the U-Net architecture, the expansive path or decoder of the network was symmetrical to the encoder and consisted of transposed convolutions, each of which doubled the size of the feature maps. In order to localise up-sampled features, the output of the transpose convolutions were concatenated with high-resolution features from the encoder via skip-connections (Ronneberger et al. 2015). Skip connections at each depth in the network directly concatenated the encoder layer output to its corresponding decoder layer. A convolution operation was applied to the output feature maps after each up-sampling operation to ensure the same number of features were present as in the symmetric encoder layer. This up-sampling procedure was repeated 5 times to pair up with 5 max pooling operations in the VGG16 encoder. Additionally, 2 fully connected layers were added to the end of the network. The U-net is a fully convolutional network (FCN) and does not contain any fully connected layers. In an FCN, the prediction of each output neuron will depend only on a subset of the input window. The addition of fully connected layers ensured that each output neuron utilised information from the whole input window when making a prediction. Whilst convolution is local, fully connected layers are global and increase the flow of information in the network. The output of the model was a segmentation map that indicated the prediction for each input time sample.

This network was adapted for use on 1D time series and trained from scratch using gradient descent with randomly initialised weights. In addition to the current network, batch

normalisation was added after convolution to increase stability and improve convergence of the network during training. Additionally, dropout was added to increase generalisation of the network. Hyperparameters such as the filter lengths and learning rate were optimised to improve the network's performance.

3.7 Network Optimisation for Earthquake Detection

The weights were initialised using the kernel initializer 'he_normal' which draws samples from a truncated normal distribution centred on 0. He_normal is suitable for deeper networks (Badrinarayanan et al. 2017) and produced good convergence of the network during training and the highest accuracy on the validation data.

The model was trained using the binary cross-entropy loss function and the RMSprop optimisation algorithm with a learning rate of 0.001, in batches of 64 windows, with a NVIDIA GeForce MX250 graphics processing unit (GPU). Cross entropy is a logarithmic loss function. A loss was generated for each sample point prediction. The loss function took the negative log of the probabilities for each sample point and computed the mean of the losses to obtain the final loss for the predicted probability distributions. In binary classification, cross-entropy for a single prediction can be calculated as:

$$Loss = -[y\log(p) + (1-y)\log(1-p)] \tag{3.2}$$

where log is the natural log, y is a binary indicator (0 or 1) or the true label and p is the predicted probability or model output. The binary cross-entropy loss function calculates the total loss by computing the average of the loss for the number of scalar values in the model output.

The batch size is a hyperparameter of gradient descent that controls the number of data windows to pass through the network before the model's internal parameters are updated. A batch size of 64 conveyed that 64 windows from the training dataset were passed to the network over a single iteration (one forwards and backwards pass through the network). This batch size allowed good convergence of the network during training and a learning rate of

0.001 produced the best performance (highest accuracy and lowest loss) on the validation dataset. To fine-tune the model weights, the learning rate was decreased by a factor of 10 when the validation loss failed to improve over 5 forward and backward passes of all training examples through the network (epochs). The minimum learning rate was set at 0.00001 to prevent the network from training at very low learning rates where updates to weights and biases become insignificant. The training process was programmed to terminate when the validation loss had not decreased in more than 15 epochs, preventing overfitting. With these parameters in place, the network converged quickly (within 10 epochs) and did not overfit to the training dataset. This was evident from observing that the test and validation loss and accuracy were similar to the training loss and accuracy. The network was trained to minimise the training loss, however, the final parameters or weights were stored when the test loss was at its lowest during training.

The best weights obtained a test accuracy of 99.95% and a test loss of 0.0015. The validation accuracy for this model was 99.5% and the loss was 0.023. These results indicated that the network was able to precisely segment the time series into two classes, even though only 7 individual slip events were used to train, test and validate the network (Figure 3.4 and Figure 3.5).

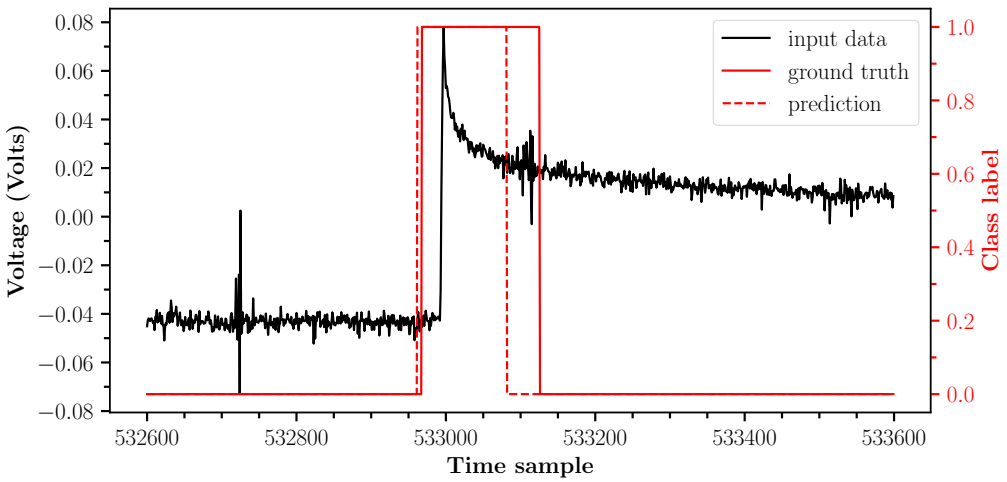

Figure 3.4: Section of a window in the test dataset containing the slip event in the test data | The corresponding ground truth and the network prediction is indicated.

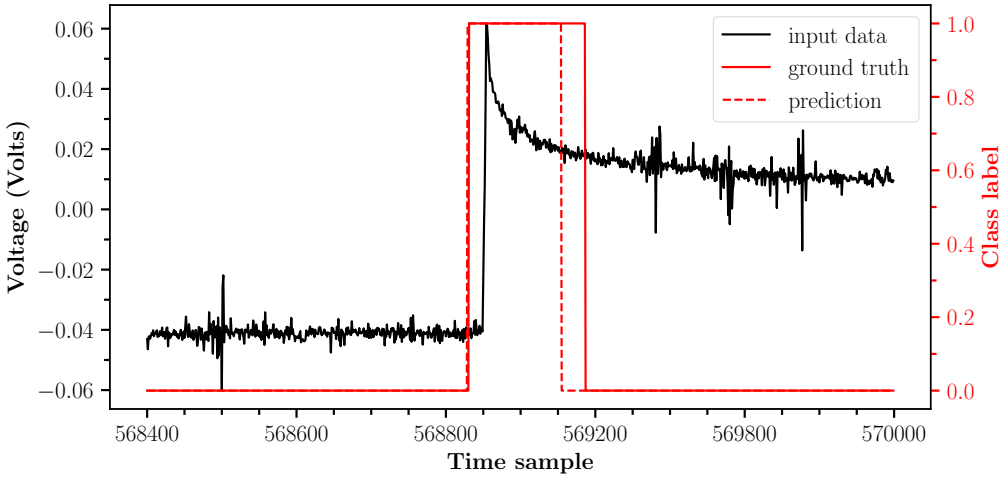

Figure 3.5: Section of a window in the validation dataset containing the slip event in the validation data | The corresponding ground truth and the network prediction is indicated.

3.8 Detection in Noisy Time Series

To determine how robust the network was at detecting earthquakes in much noisier time series, the same neural network was trained and tested on data from a low SNR strain gauge recording obtained during the same experiment (Figure 3.6). This sensor's recordings had a much lower SNR and, as a result, the individual slip events were significantly obscured by noise. Due to the increased difficulty of the task, the data were decimated by 256 instead of 512. A smaller decimation factor will have increased the level of detail and encouraged the network to learn more complex features and patterns specific to each class.

The noisier dataset was relabelled in accordance with the original (high SNR) dataset also decimated by a factor of 256 (Figure 3.7). The low SNR dataset and its corresponding ground truth are plotted in Figure 3.6.

In the high SNR dataset, the slip events were denoted by an increase in the signal voltage (Figure 3.7a). Differences in the orientations or positions of the strain gauges on the rock sample can alternatively result in a decrease in the signal during a slip event. This was observed in the low SNR dataset (Figure 3.7b), and could be explained by the fact that some sensors measured a contraction, whilst others measured an extension. Regardless, the event in Figure 3.7a is the same as that in Figure 3.7b.

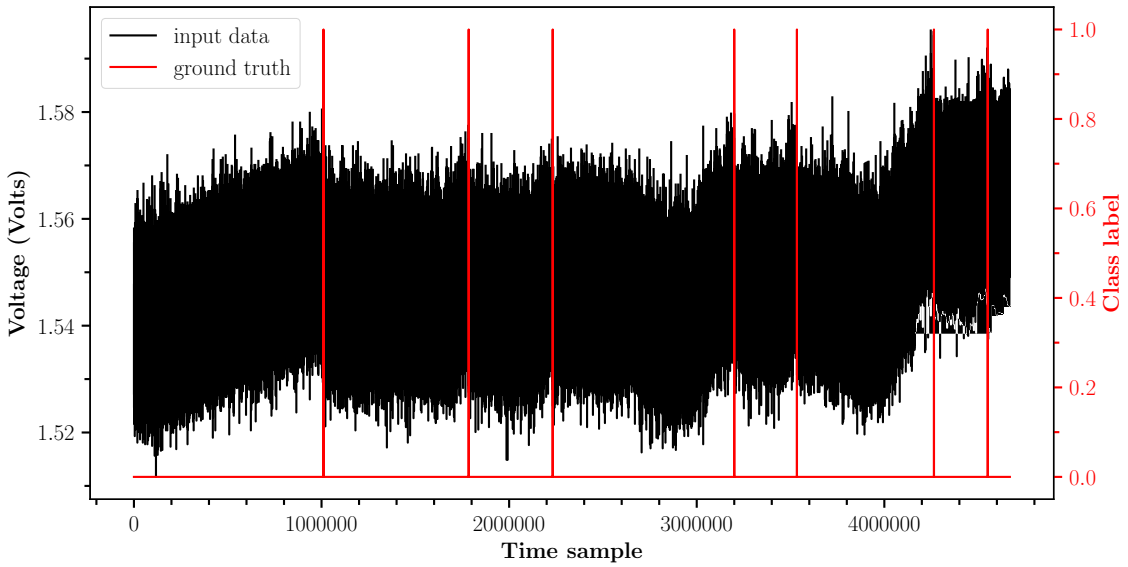

Figure 3.6: Low SNR time series dataset and corresponding ground truth | The same 7 events as in Figure 3.2a were recorded by this sensor.

(a) (b)

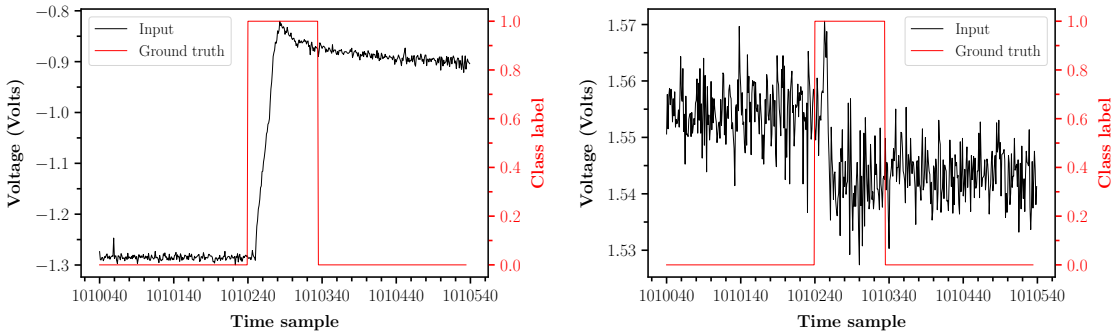

Figure 3.7: Method applied to label the low SNR dataset| (a) Single slip event from the high SNR dataset and corresponding ground truth. (b) Same slip event as in (a) from the low SNR dataset and corresponding ground truth. The slip event in (a) was used to label the event in (b).

The data were split into train, test and validation datasets and windows were generated with no overlap when they contained only noise signal and with 80% overlap when they contained earthquake signal. The events selected in each dataset were different to previously in Section 3.5. The validation dataset was selected to contain the lowest amplitude (most noise-obscured) slip event to evaluate the ability of the network to recognise more complex variants of the examples in the training data. Each window was standardised separately, and the network was re-trained from scratch on this low SNR dataset. The network achieved an

accuracy of 99.9% and a loss of 0.0019 on the test data and an accuracy of 99.8% and a loss of 0.006 on the validation data. The results are displayed in Figure 3.8 and Figure 3.9.

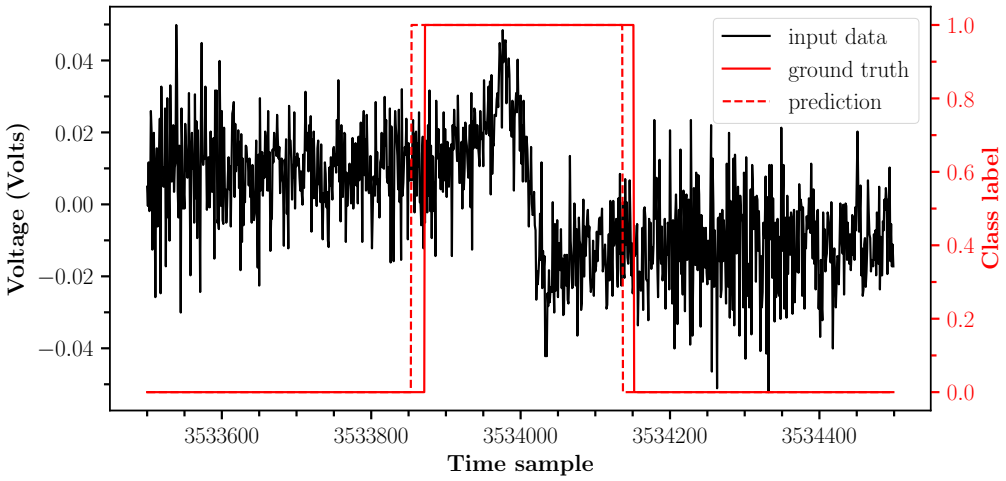

Figure 3.8: Window of the test dataset containing the test earthquake | The ground truth and the network prediction is plotted.

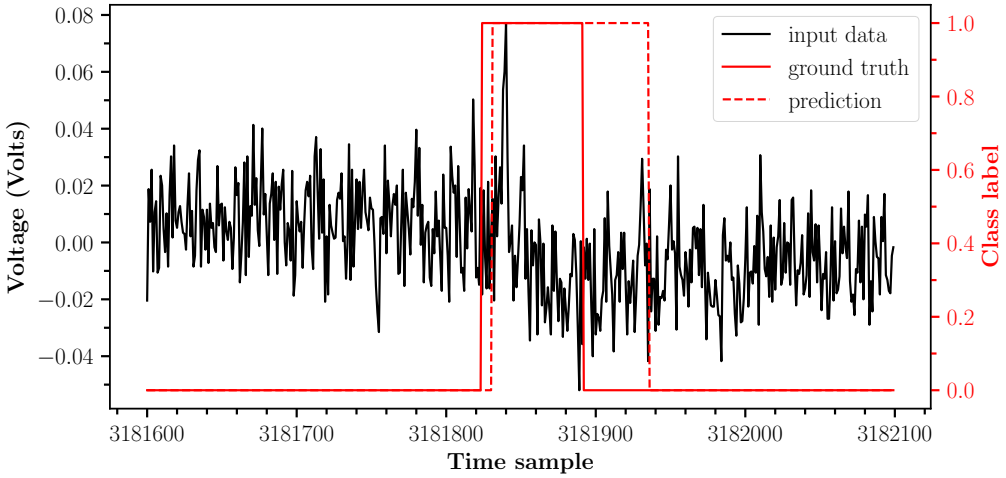

Figure 3.9: Window of the validation dataset containing the validation earthquake | The ground truth and the network prediction is plotted.

The results indicate precise segmentation of the low SNR time series into 'noise' and 'earthquake' classes. Even with the most noise-obscured slip event in the validation dataset, the network was still able to accurately localise the earthquake (Figure 3.9). The test and validation datasets comprised 15% and 10% of the whole dataset respectively, therefore, a large proportion of the overlapping windows in the test and validation data contained no

earthquake signal. The network did not detect earthquake signal in any of these windows, indicating that no false positive predictions occurred across the test and validation datasets.

These results indicate that the network was able to successfully extract discriminative features in the encoder of the network and accurately relocate significant activations back to input space for precise segmentation. The network performed extremely well on the test and validation datasets, indicating that the features learned are general and can be used to accurately segment unseen data. A high test and validation accuracy also indicate robustness of the network to noise as well as robustness to possible degradation of the input signal due to decimation.

3.9 Visualisation of Segmentation for Earthquake Detection

The end-to-end learning strategy of CNNs make their representations a black box meaning that it is difficult to understand the logic of their predictions. CNN representations were visualised both in intermediate network layers and the output layer using several inversion and gradient based methods. This included feature map visualisation, feature map inversion, saliency maps and filter visualisation. These visualisation techniques are typically performed on 2D convolutions and were adapted for application on time series data. All of the visualisation techniques, other than saliency maps, provided unclear and inconclusive results on time series data.

With image classification approaches, a natural question is if the model is truly identifying the location of the object in the image, or just using the surrounding context. Visualisation is important for investigating what features the network uses to make a prediction. The results should indicate that the network focused on features of the slip events when detecting an earthquake as opposed to the surrounding noise.

Saliency maps are intended to provide insight into what aspects of the input a CNN is using to make a prediction. Saliency maps plot the gradient of the predicted outcome from the model with respect to the input sample points. In other words, they visualise the derivatives

calculated using backpropagation. The magnitude of the derivatives on the saliency map determine which time samples need to be changed the least to affect the class score the most. Large values in saliency maps indicate time samples of high importance. The saliency map in Figure 3.10 indicates that the network learnt to detect the slip event as being different from the surrounding noise and not vice versa. The saliency at the location of the main slip event is larger in comparison to the saliencies before and after. The saliencies increased in the 225 time-samples prior to the labelled start of the slip event. This could indicate that time samples before the earthquake aided in its detection. The saliency map for a window containing the validation slip event in the low SNR dataset was similar to that in Figure 3.10. This result is shown in a later section (Section 3.14, Figure 3.16) and therefore was not included here.

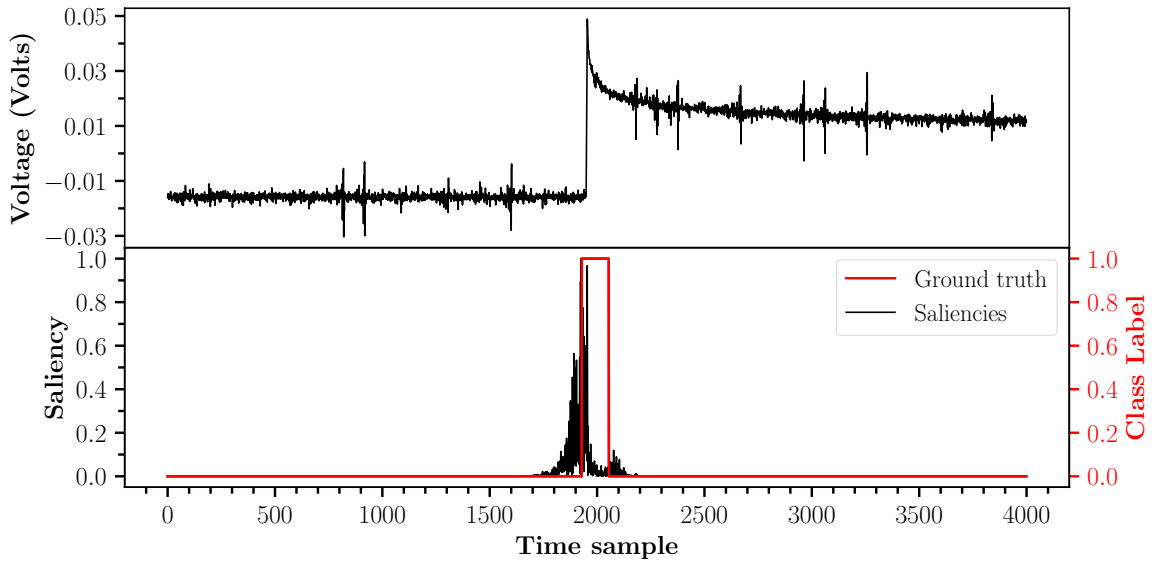

Figure 3.10: Example saliency result | Window of the validation data containing the slip event in the high SNR dataset (top) and its corresponding saliency map (bottom). The highest saliencies occurred at the location of the main slip event. The ground truth is plotted to indicate where the event was labelled.

3.10 Prediction

Precursors were investigated within the strain gauge signal recorded by the low SNR sensor. This dataset was selected as a high level of background noise is commonly observed in real seismic data (microseismic noise) and a higher noise level will have encouraged the network

to learn complex, precursor-related features, preventing any short-term, trend-based observations from influencing the features learnt. Additionally, the network achieved a higher validation accuracy on this dataset. It should be noted that during this investigation, the events selected in the train, test and validation datasets were random.

To adapt the earthquake detection task to be suited for prediction, changes were made to:

1. The raw data:
    - The data were decimated by 128 samples, 4 times less than originally. This was applied to increase the chance of detecting complex, precursor-related features in the signal.
2. The method of labelling the data:
    - The ground truth was generated by assigning each time sample in the data a label from one of 3 classes; 'precursors', 'impulsive earthquake signal' and 'noise'. 'Precursors' were not visually evident in the data and the signal labelled as a 'precursor' is justified in Section 3.11. The result was a boxcar function with 'zeros' where nothing occurred, 'ones' in the region prior to each earthquake and 'twos' during the main slip events.
3. The sliding window:
    - Moving windows with a length of 4096 time samples were used to scan the whole length of data. The overlap between windows containing precursors was increased such that the sliding window only moved by one time-sample in the region of the time series labelled as a 'precursor'. This significantly increased the number of windows containing precursor-labelled time samples, reducing the class imbalance (ratio of noise-labelled time samples to precursor-labelled time samples.
4. The network:
    - Introduced 2 Long Short Term Memory (LSTM) layers (Graves 2012) to the end of the network. An LSTM is a type of recurrent neural network that is capable of learning long-term order dependence in time series. The U-net is a fully convolutional network (FCN). Convolutional networks do not account for sequential

dependencies and, therefore, LSTM layers were used to interpret the features output from the original network across time steps. The addition of LSTM layers improved the predictive capability of the network.

- Network efficiency was increased by reducing the number of learnable parameters until the point at which the performance of the network decreased.

5. The method for evaluating the success of the network:
    - When testing the learned weights on the validation data, a prediction was considered successful if the network was able to consistently detect signal belonging to the class 'precursor' in the moving windows prior to an earthquake (before the earthquake was in the frame of view). The network was adapted to detect signal belonging to the class 'precursor' as early as possible prior to the event in the validation dataset. The network was developed to optimise the predictive capability on the event in the validation dataset as this provided the best generalisation.

3.11 Dataset Labelling for Prediction

The start of the individual slip events were labelled using the high SNR dataset also decimated by a factor of 128. To gain an understanding of when precursors might have occurred prior to the slip events, processes that govern rupture initiation on frictional interfaces were investigated. Several lab experiments and theoretical studies have suggested that earthquake faulting does not occur abruptly (Latour et al. 2013, McLaskey 2019, Guérin-Marthe et al. 2019, Ostapchuk & Morozova 2020). Instead, accelerating aseismic rupture growth within a nucleation zone precedes dynamic rupture propagation (McLaskey 2019). Earthquake nucleation has been observed prior to some crustal earthquakes. This precursory phase has been frequently accompanied by slow slip, identified from acceleration of GPS stations, and foreshock activity (Ruiz et al. 2017, Socquet et al. 2017, Tape et al. 2018). Lab experiments and simulations under a wide range of normal stresses have identified a nucleation phase which consists of slow propagation followed by a faster acceleration period, both of which are potentially aseismic (Figure 3.11). The acceleration is proceeded by dynamic rupture propagation. Understanding rupture nucleation is critical for the development of probabilistic

forecasting as it aids in determining when and under what conditions detectable precursory signals may be generated. Kaneko et al. (2016) identified three distinct phases of rupture evolution that are observed regardless of the applied normal stress: quasi-static propagation, acceleration, and dynamic rupture propagation (Figure 3.11).

Due to difficulty in identifying precursors based on visual characteristics of the signal, the start of the precursors was labelled in accordance with the transition between the slow slipping phase and the acceleration phase in Figure 3.11. The start of the slow slipping phase was hard to infer from previous lab experiments and apply to this dataset, therefore, this phase was not considered when labelling the data.

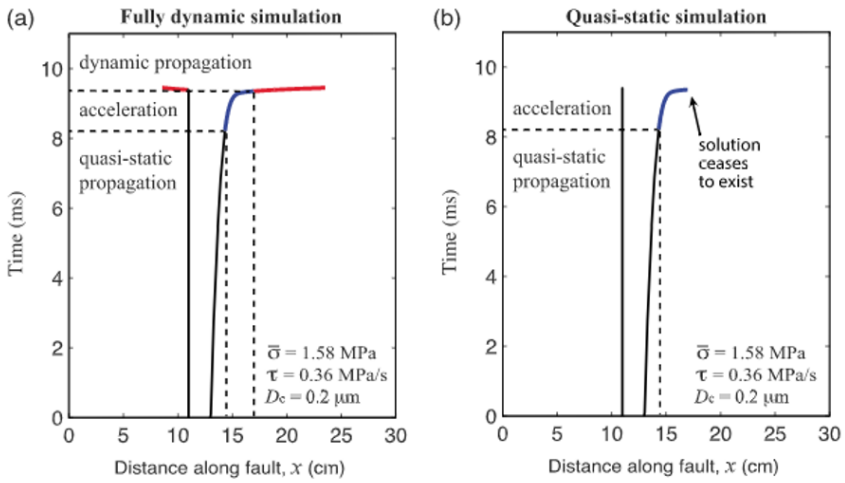

Figure 3.11: Examples of the nucleation phase frequently observed to occur prior to lab induced slip events (Latour et al. 2013) | This phase is indicated for (a) a fully dynamic simulation and (b) a quasi-static (slow slip) simulation. The acceleration phase (blue) was used as a guideline for labelling the sections of the strain gauge data thought to contain precursors.

In lab experiments, the acceleration phase has been identified to start around 1-2 ms prior to dynamic rupture propagation with the exact timing dependent on the normal stress and the loading rate (Kaneko et al. 2016). Based on these findings, the start of the region considered to contain precursors was specified 2 ms prior to the start of the earthquakes to reduce the chances of precursors being labelled as noise and ensure a maximum amount of data in the minority (precursor) class. As the data were decimated by a factor of 128, the data were labelled as a precursor from 156 samples prior to the start until the start of the slip events. The labelled events are indicated in Figure 3.12a and Figure 3.12b.

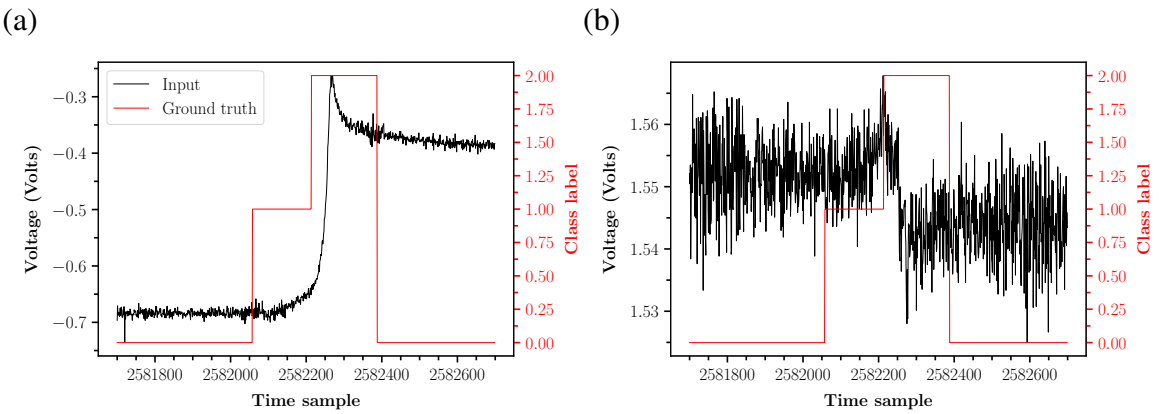

Figure 3.12: Method of labelling the data for earthquake prediction | (a) Slip event from the high SNR dataset with the corresponding ground truth. The dataset was labelled as a precursor (ones) 2 ms prior to the labelled start of the earthquake class (twos). (b) Same slip event as in (a) from the low SNR dataset with the corresponding ground truth. The section of the dataset containing earthquake signal was labelled using the low SNR data as previously in Section 3.8. A legend is plotted in (a), referring to both figures.

3.12 Results of Lab Earthquake Prediction

To determine the predictive capability of the network, the sliding window was moved along the whole validation dataset by a single time sample. This enabled a detailed understanding of the detections made by the network as the sliding window approached the earthquake. It also indicated whether the earthquake in the validation data had been predicted and detected or only detected. The network was considered successful at predicting the earthquake if it consistently detected signal belonging to class '1' or 'precursors' before any of the signal related to class '2' or 'earthquake' was in the view of the sliding window (Figure 3.13).

Figure 3.13a is the first window in the validation dataset where signal in the precursor class was detected by the network. When this window was input, the network classified the last few tens of samples in the window as precursors, see Figure 3.13b for a more detailed view. It is evident that the network detected precursor-related signal before earthquake-labelled signal was in the window input to the network. As a result, the network successfully predicted the earthquake in the validation dataset. The network was trained for 25 epochs. The best weights were saved and a validation accuracy of 99.4% was obtained with a loss of 0.04. No false positive predictions occurred. The network only detected signal belonging to the class 'precursors' in the windows immediately prior to the validation earthquake.

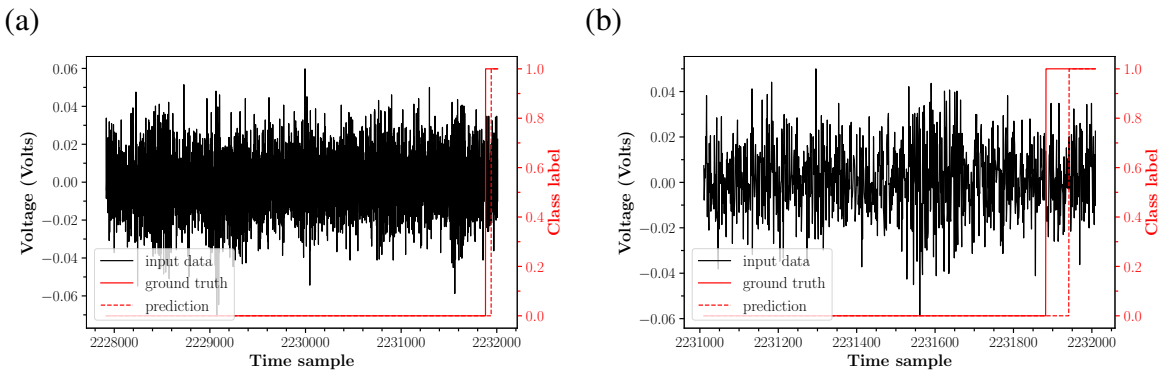

Figure 3.13: Network prediction | (a) First validation window where the network detected signal in the class 'precursors'. The ground truth and predictions are overlain. It is evident that the network detected signal in 'precursors' before the validation slip event was in the frame of view of the network. (b) Final 1024 time samples from the input window in (a), showing a more detailed view of the network prediction.

3.13 Visualisation by Occlusion

The final decision of the network on any input time sample is influenced only by points inside the receptive field of that sample (Section 2.2). All other points outside the receptive field do not contribute to the decision made by the network. As a result, if the receptive field of the network was not large enough to include the necessary information from the input window to make a prediction, the earthquake would not have been predicted. The receptive field of the network was increased by stacking many convolutional layers in the encoder of the network and by increasing the kernel size from 3 in the original VGG16 network to 7.

As the network was able to predict the event in the validation data, the receptive field must have been large enough to capture enough precursor-related information from the input. Various visualisation techniques can be used to gain an understanding of the sample points in the input that were significant for earthquake prediction. To visualise which sample points in Figure 3.13a were important for prediction, sample points were occluded (set to zero) cumulatively from the start of this window until the point at which the network was no longer able to make its original prediction, see stage 1 (Figure 3.14). Once this point was identified, the occlusion window was removed and sample points from the opposite end of the window were cumulatively occluded until the point at which the network could no longer make the original prediction, see stage 2 (Figure 3.14). The results of this experiment

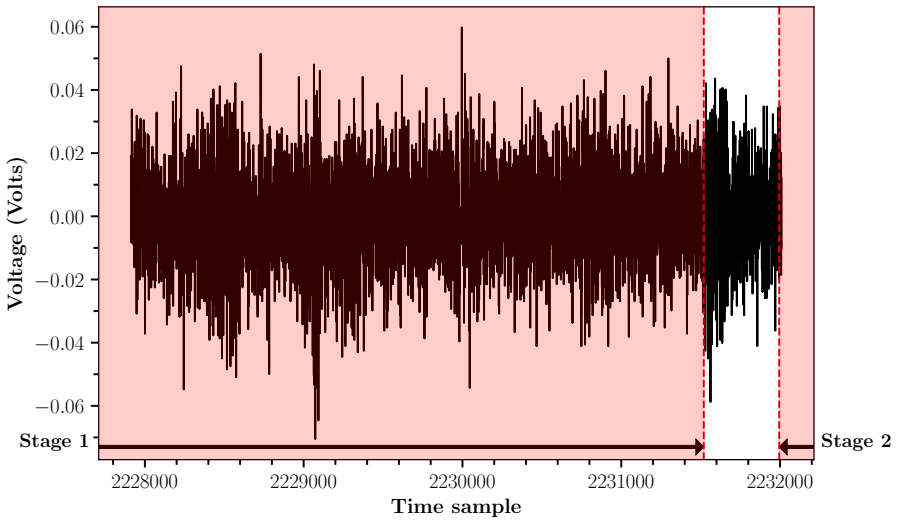

Figure 3.14: Result of the occlusion experiment | The section highlighted in red indicates the region of the input in Figure 3.13a that could be occluded separately for each stage such that the network produced its original prediction. This indicated that data within the red box did not contribute to the original prediction.

indicated a localised region which was required in the input for the network to make its original prediction (Figure 3.14). Figure 3.14 indicates how much of the window in Figure 3.13a could be set to zero (region shaded red), separately for each stage, before the network longer predicted the earthquake. This indicates that the unhighlighted region in Figure 3.14 was compulsory for prediction. Figures 3.15a and 3.15b show this region of importance in relation to the start of the earthquake in the validation data.

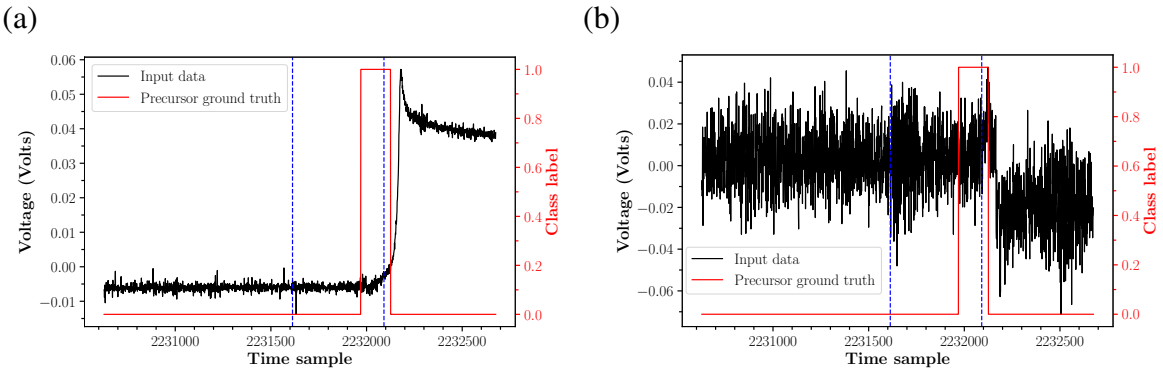

Figure 3.15: Results of the occlusion experiment | The section of the data between the two blue, dashed, vertical lines indicates the region of importance identified in Figure 3.14. This region is shown in relation to the start of the validation earthquake in (a) the high SNR dataset and (b) the low SNR dataset. The ground truth for the precursor class is indicated. It should be noted that the event in (a) is the same as that in (b).

3.14 Visualisation using Saliency Maps

Figure 3.16 shows the saliency map for a window of the validation data containing the whole validation earthquake. The dashed, blue lines outline a region prior to the earthquake where there is an increase in the saliency. The saliency map indicates that the network places importance on a short section of the data (within the blue lines in Figure 3.16 and 3.17) prior to the slip event.

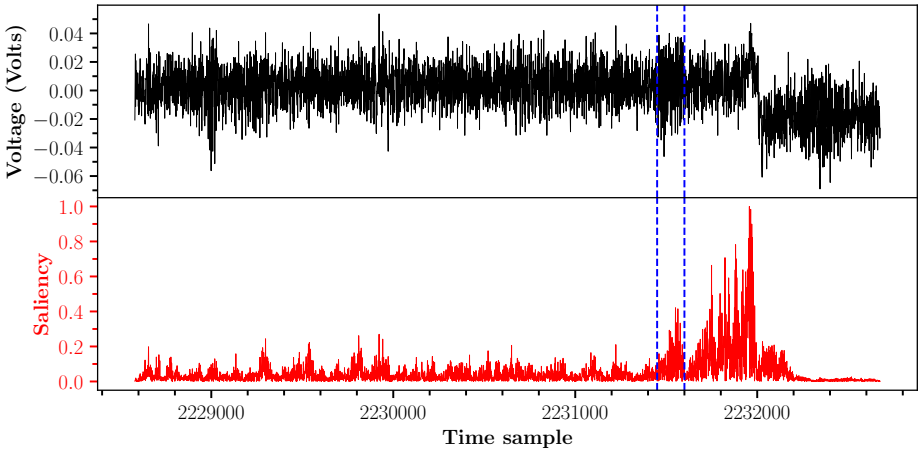

Figure 3.16: Saliency experiment result for prediction | Input window containing the validation earthquake (top) and corresponding saliency map (bottom) where high saliencies indicate a high importance for that sample point. Blue, dashed lines indicate a localised region of increased saliency prior to the main slip event.

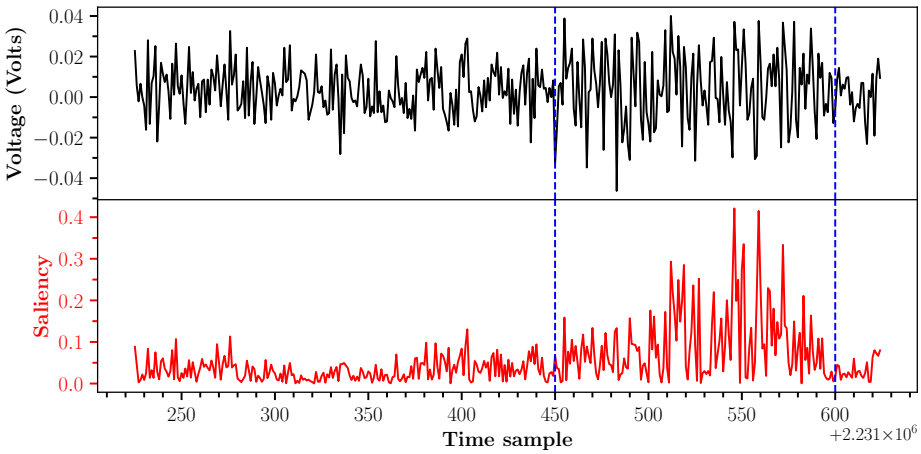

Figure 3.17: Detailed view of a region of increased saliency prior to the validation earthquake | Small section of the input (top) and of the saliency map (bottom) from Figure 3.16. The blue, dashed lines are at the same location as in Figure 3.16.

The saliency map highlights a region of the data prior to but not immediately before the earthquake which had a significant influence on its predictability. This region of importance coincides with the result from the occlusion experiment shown in Figure 3.15b. The first blue line in Figure 3.15b coincides with the first blue line in Figure 3.16 and Figure 3.17.

The results of this experiment indicated that for the slip event in the validation data, precursors occurred up to 514 time samples or 6.6 ms prior to the start of the event. It is unsure whether this coincided with the acceleration or slow slipping period of the nucleation phase in Figure 3.11. This is because the experiment was enclosed by a vessel and the position of the rupture front could not be observed. The result is more in accordance with observations from Nielsen et al. (2010) and Latour et al. (2013) where the acceleration phase is observed between 2 and 10 ms prior to the slip events.

Chapter 4

Investigating Precursors in Real Earthquake Data

4.1 Introduction

Despite significant effort, no statistically rigorous application exists involving the use of precursory phenomena to forecast large earthquakes. Real seismic data is analysed in this chapter, as opposed to lab data in Chapter 3, in order to determine whether systematic changes, attributed to precursors, can be detected and identified in the raw seismic signal prior to large (Mw $\geq$ 6) earthquakes. Additionally, the algorithm is tested to determine the potential of this method to probabilistically forecast earthquakes. A binary classification approach is used to classify windows of seismic data labelled as 'noise' from windows labelled as 'precursors'. A classification approach was applied due to a greater availability of data compared with previously in Chapter 3 and difficulty in accurately labelling the dataset (specifically the precursor class) for use in semantic segmentation. Instead of predicting the class of every time sample within each window input to the neural network as in Chapter 3, classification involved predicting the class of each window. The train, test and validation accuracy represented the number of correctly classified windows within the train, test and validation datasets, respectively. The 'earthquake' class in Chapter 3 is no longer included in this chapter as it was irrelevant to the investigation of precursors with a classification approach.

4.2 Geological Region Investigated

The Japan subduction zone has been well documented as a highly seismically active region due to its tectonic setting. It is located at the junction of four tectonic plates: The Pacific and Philippine oceanic plates and the Eurasian and North American continental plates (Figure 4.1).

As a result of this tectonic junction, Japan experiences around 400 Mw $>$ 0 earthquakes per day (McGuire et al. 2005). Additionally, earthquakes in Japan account for over 20% of all Mw 6 or greater earthquakes worldwide (Mogi 1981). Although the majority of earthquakes that occur do not have a significant impact, many have been highly destructive. The largest recorded earthquake was the 2011 Mw 9 Tohoku Earthquake, which ruptured the central section of the Japan Trench to a depth of approximately 50 km (Ozawa et al. 2012).

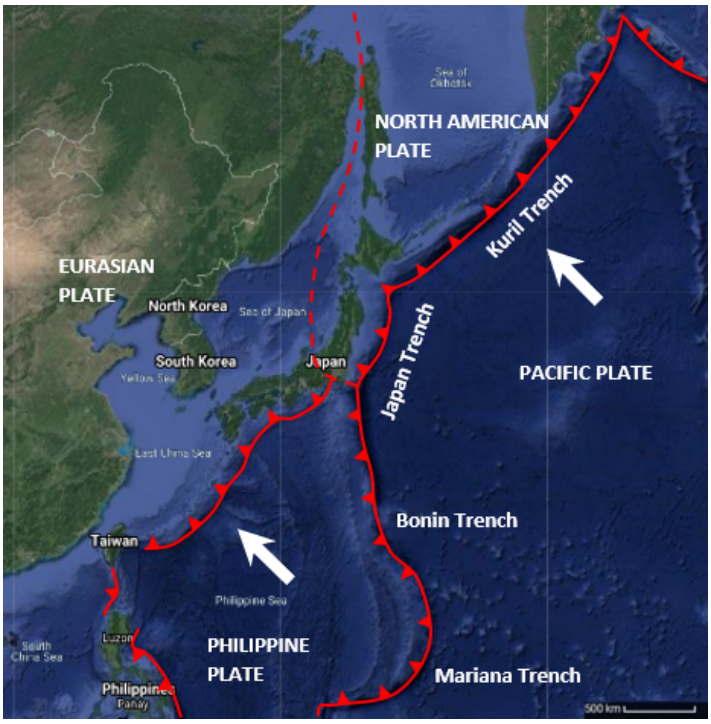

Figure 4.1: Tectonic plates and their boundaries surrounding Japan | The oceanic plates converge with the continental plates, generating several subduction zones.

Japan's National Project for Earthquake Prediction has been active since 1965 without success. An earthquake prediction should be short-term and based on observable physical phenomena or precursors (Uyeda 2013). The main reason for lack of success is failure to

detect reliable precursors (Uyeda 2013). Much of the financial resources and manpower of the project have been devoted to improving seismograph networks. Although reliable precursors are yet to be identified, a significant volume of well-recorded, earthquake-related seismic data exists in this region.

In addition to the dense seismic network and the high recurrence of large (Mw $>$ 6) earthquakes, aseismic slip with transient timescales of days to months has recently been observed in the Japan subduction zone using continuously monitored GPS arrays (McGuire et al. 2005). A continuously slipping subduction zone should increase the potential for precursors, however, it might also result in a significant number of foreshocks that could substantially overprint precursors in the seismic signal (McGuire et al. 2005).

4.3 Data Collection

4.3.1 Station of interest

The data were obtained from the Incorporated Research Institutions for Seismology (IRIS). A single station was selected from the Global Seismographic Network (GSN) and the data recorded by a single seismometer at this station were used to train and test the neural network. The GSN is a state-of-the-art digital seismic network comprised of 152 globally distributed stations. GSN instrumentation measures and records with high fidelity all seismic vibrations from high frequency, strong ground motions near an earthquake to the slowest global earth oscillations generated by extreme earthquakes (Davis et al. 2005). GSN stations attempt to obtain the best possible recording capability and provide the most reliable source of seismic data currently available.

A single GSN station was selected, IU MAJO, located in Matsushiro, central Japan, at a latitude of 36.546° and a longitude of 138.204° (Figure 4.2). The elevation of the station is 405 m and the data were obtained from a single instrument – Streckeisen STS-2 high-gain at a depth of 0 m and 40 Hz sampling frequency. To test generalisation of any discriminative features learnt during the training process to seismic data recorded by an instrument at a different GSN seismic station, a single event in the unseen (test/validation) dataset was

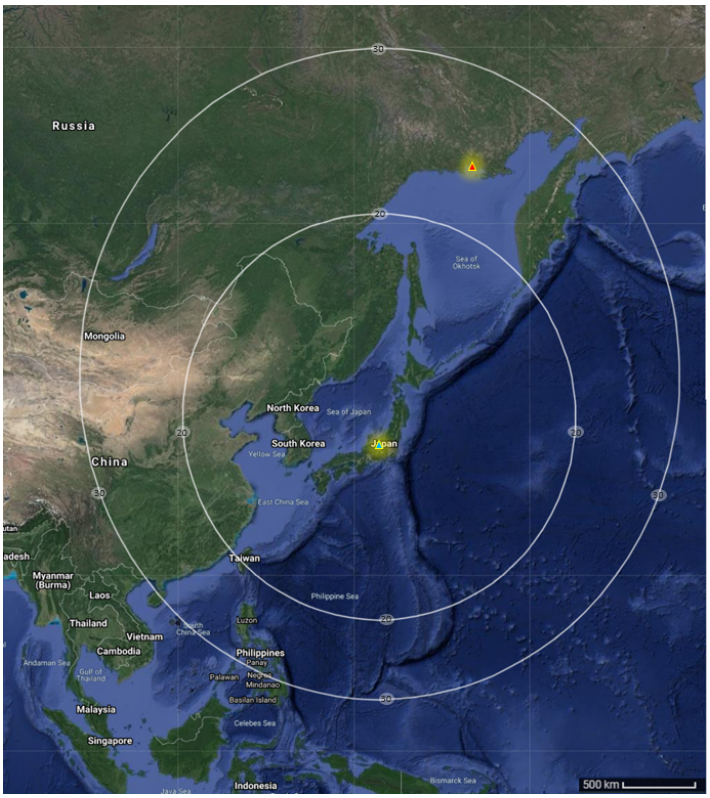

Figure 4.2: Region investigated | The location of the station of interest, IU MAJO, is indicated by a yellow and blue triangle. Station IU MA2 is indicated by a red and yellow triangle. A radius of 20° and 30° from the station of interest (IU MAJO) are indicated.

investigated from a different seismometer. The seismometer selected for this event was located at GSN station, IU MA2, at a latitude of 59.58° and a longitude of 150.77° (Figure 4.2). Similar to the seismometer of interest, the data were obtained from a Streckeisen STS-2 high-gain seismic instrument (40 Hz sampling frequency) at a depth of 2 m. Seismograms reflect the combined influence of the seismic source, the propagation path, the frequency response of the instrument and the ambient noise at the recording site (Bormann et al. 2009). Selecting the training data from a single instrument constrained the frequency response and surrounding noise, increasing the chances of detecting systematic precursors.

4.3.2 Magnitude range

Mw 6 or larger earthquakes were investigated. Mw 6 was set as the seismic event threshold because, firstly, it is considered more useful to forecast large earthquakes and, secondly, it was reasonable to expect that precursors would be insignificant and therefore harder to detect prior to small (Mw $<$ 6) earthquakes.

4.3.3 Time period

Instead of investigating precursors to all Mw $\geq$ 6 earthquakes recorded by the seismometer of interest, several constraints were applied to the events selected. The first constraint was the time-period over which earthquakes were selected. Earthquakes were only investigated if they occurred between March 2012 and February 2020.

The seismic data were originally downloaded for investigation in February 2020, so this was set as the upper limit. March 2012 was selected as the lower limit to reduce the influence of significant stress changes and afterslip from the March 2011 Mw 9 earthquake on the features learnt by the neural network during training and to improve generalisation of the algorithm to future earthquakes. The stress changes and afterslip following the March 2011 Mw 9 earthquake are described in the remainder of this subsection to justify the selected lower limit.

After a large earthquake, the resultant relaxation and transfer of stress along the fault causes aseismic afterslip that rapidly decays with time (Hu et al. 2016). Afterslip results in a series of smaller tremors which are considered aftershocks if they are associated with the fault that generated the main shock, or if they are on a different fault but within one full fault length from the earthquake's epicenter. Aftershocks generally decrease in Mw and frequency over time with a decay inversely proportional to the amount of time since the principal earthquake (Omori's law) (Utsu et al. 1995). Once the rate at which these tremors occur has declined to pre-earthquake levels, the sequence of aftershocks is thought to end (Peng et al. 2006). Stress changes may occur before, during and after an earthquake (Ozawa et al. 2012). This could result in significant changes to the characteristics of the seismic signal and any precursor-related features. As a result, stress changes associated with the 2011 Mw 9 Tohoku earthquake and the decay in the frequency of aftershocks were investigated.

The aim was to constrain the selected Mw $\geq$ 6 earthquakes to a period when there was a stable stress field and where earthquake frequency had reduced to pre-earthquake levels. These constraints were implemented to prevent the network from learning features in the seismic data specific to the earthquakes that occurred immediately after the Mw 9 Tohoku

earthquake when afterslip was most significant (Utsu et al. 1995). This could otherwise reduce generalisation of any detected precursors to earthquakes unassociated with significant afterslip from the Mw 9 earthquake. For example, investigating precursors to highly recurrent aftershocks could encourage the network to detect afterslip from one aftershock as the precursors to another, a phenomenon less likely to occur when earthquake recurrence has reduced to become stable. Aftershock recurrence decays with time according to Omori's Law (Utsu et al. 1995), therefore, discarding highly recurrent aftershocks from the investigation should have improved generalisation of precursors to future earthquakes where background seismicity is stable.

Finally, changes to the stress state along the fault before and after the Mw 9 earthquake could have altered characteristics of any precursors in the seismic signal. As a result, there was reason to limit the investigation to earthquakes that did not occur within the time frame associated with significant afterslip from the Mw 9 earthquake and to earthquakes that occurred after the Mw 9 earthquake (discarding those that occurred before).

Changes in stress before, during and after the 2011 Mw 9 Tohoku earthquake have previously been investigated (Becker et al. 2018). The mean horizontal normal stress was inferred from crustal earthquake moment tensors over time for a small, square region (1° by 1°) located above the Mw 9 fault slip area. The modification of crustal stress was not just limited to regions close to the rupture but was also seen regionally onshore in Honshu. The most significant modification to the stress field was found to occur at the time of the Mw 9 Tohoku earthquake. Comparing the stress anomalies, it was clear that the region between northern Honshu and the Japan trench, strongly compressive in the horizontal stress component, became extensional after the Mw 9 earthquake (Becker et al. 2018). The stress modification due to the Mw 9 earthquake was as expected from an understanding of the stress changes that occur during the megathrust cycle (Herman & Govers 2020). Additionally, there was indication of a short term-transient where a further increase of the horizontal stress occurred following the Mw 9 earthquake, until a plateau was reached after approximately 1 year. Although stress release is not homogeneous and will have affected neighbouring regions differently to the region investigated in (Becker et al. 2018), it may provide an indication of

when the stress state had returned to become stable following the Mw 9 earthquake.

The frequency of aftershocks was similarly investigated and a sudden, short-term increase of the seismicity rate was observed immediately after the Mw 9 earthquake (Toda 2019). This was followed by an approximately exponential decrease in the seismicity rate which is compatible with Omori's law of aftershock decay (Utsu et al. 1995). Roughly 1 year following the start of the Mw 9 earthquake, the rate had become stable and lower compared to the rate prior to the earthquake. A lower recurrence rate could be explained by the huge stress release that occurred during the Mw 9 Tohoku earthquake.

Changes in the seismicity rate coincided with changes in the stress state where both variables became stable $\sim$ 1 year after the Mw 9. To increase the chances of detecting systematic precursors and ensure generalisability of features learned to future earthquakes, the investigation was limited to earthquakes occurring after March 2012 (approximately 1 year after the Mw 9 earthquake).

4.3.4 Region

In addition to constraining the selected earthquakes to those occurring within a time interval and Mw range, only earthquakes relatively close to the station of interest were investigated. If any precursory changes in the seismicity exist, they would likely concern processes of small amplitude, quite close to the station of interest. Therefore, the area of investigation was limited to a region which was not larger than a few thousand km from the station. Investigating the quality of seismometer recordings at different distances from the station of interest, it became evident that the quality of the seismic data had degraded (amplitude of the earthquake signal decreased significantly) by $\sim$ 2500km from the station. As a result, earthquakes were investigated if their epicentre occurred within a radius of 20° (approximately 2220 km) from the station (Figure 4.2). This radius was an approximation and did not accurately reflect the distance from the station at which precursors or features related to precursors were no longer evident or significant enough to be detected in the seismic data.

4.3.5 Timing relative to other Mw $\geq$ 6 earthquakes

To reduce the influence of post seismic slip associated with other Mw $\geq$ 6 earthquakes on the seismic data investigated for precursors, Mw $\geq$ 6 earthquakes were only selected when no other Mw $\geq$ 6 earthquake located within a radius of 30° from the seismometer of interest occurred over the 48 hour period prior to the selected earthquakes. Due to the high recurrence of Mw $\geq$ 6 earthquakes in the Japan region, selecting a fairly short time period of 48 hours prevented a significant number of earthquakes from being removed from the investigation. 30° was selected instead of the original 20° to reduce influence from neighbouring earthquakes that did not occur within 20° from the station of interest. Although the duration of precursors and afterslip is unknown, based on the Omori's law of aftershock decay, it is likely that this constraint reduced the influence of other Mw $\geq$ 6 earthquakes on those investigated. Additionally, it should be stressed that majority of the investigated earthquakes did not occur within a few days from from another Mw $\geq$ 6 earthquake but within several days to weeks. This constraint increased the likelihood that the seismic data associated with each event contained minimal afterslip from other Mw $\geq$ 6 earthquakes.

4.3.6 Time period investigated prior to each earthquake

Ten hours of seismic data were investigated prior to each of the selected earthquakes. Ten hours was selected to focus on and investigate short term changes in the seismic data that could indicate an impending earthquake. Systematic precursors are yet to be identified in seismic data, therefore, there was no method of determining when precursors would be most evident prior to a Mw $\geq$ 6 earthquake. As a result, it was assumed that precursors would increase in significance with proximity to the earthquakes, as has previously been observed systematically prior to lab earthquakes (Johnson et al. 2013, Passelègue et al. 2017, Rouet-Leduc et al. 2017). Investigating a short time period prior to each earthquake encouraged the network to learn short-term changes. Additionally, as some of the earthquakes may have occurred only 48 hours from another Mw $\geq$ 6 earthquake, investigating a short period will have reduced the influence of other Mw $\geq$ 6 earthquakes on features learnt during training.

The seismic data consisted of 3 broadband channels (BHZ for vertical motion, BH1 aligned more than 5° from north and BH2 aligned more than 5° from east), each sampled at a frequency of 40 Hz.

4.3.7 Removal of events

The seismic data prior to each of the selected Mw $\geq$ 6 earthquakes were manually plotted and analysed. Some of the events were found to contain impulsive earthquake signal arising from smaller (Mw $<$ 6) earthquakes. Events containing impulsive earthquake signal above the noise level were discarded to encourage the network to analyse features of the background signal, removing the influence of earthquake waveforms (Rouet-Leduc et al. 2019).

In an ideal scenario, when investigating very short term (minutes-hours) earthquake precursors there would be no impulsive, high SNR earthquakes occurring in the time-period under investigation (in this scenario, over the 10 hours prior). The presence of highly impulsive earthquakes may alter the characteristics of the seismic data and affect the features learnt by the neural network during training (Ishibashi 1988). This could significantly impact or bias the results by encouraging the network to use earthquake signal as a strategy for classifying noise from precursors. This issue would be particularly significant when investigating very short-term precursors where the quantity of data input is very limited and therefore should be well representative of each class. Impulsive signal could prevent the network from learning underlying patterns in the seismic data related to precursors. In Japan there is a high recurrence rate of Mw 4-6 earthquakes. By removing events containing impulsive earthquakes, this issue should not have influenced the patterns learnt by the neural network, reducing bias in the strategy learnt by the network when forming a decision.

The remaining events contained only the background seismic signal (Figure 4.3a). Figure 4.3b shows an event containing impulsive signal above the noise level. In addition, poorly recorded events were removed (Figure 4.3c). As 10 hours of 40 Hz time series data were downloaded for each earthquake, the desired length of each channel was 1440000 time samples. On reading in each file, files were automatically discarded if any of the 3 channels did not have a length of 1440000 (Figure 4.3d). Having applied these constraints, 31 Mw $\geq$ 6

earthquakes remained to train and test the neural network (Figure 4.4, Table A.1 and Table A.2).

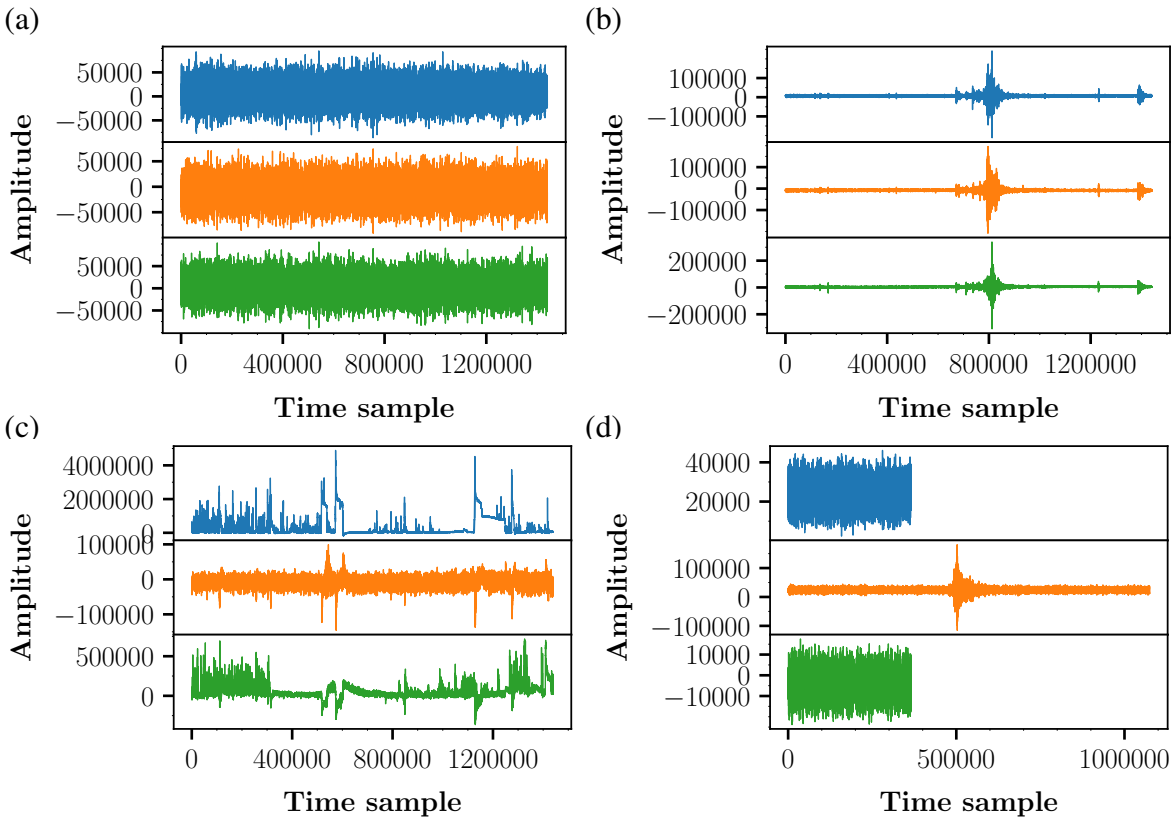

Figure 4.3: Examples of the 3 channels of seismic data (blue, orange and green plots) over the 10-hour period prior to the selected Mw $\geq$ 6 earthquakes (no standardisation) | (a) 10-hour period with no impulsive earthquake signal above the noise level. Events such as this were included in the investigation. (b) Event containing impulsive signal above the background noise level. This is an example of an event that was removed (c) 'Bad' data example – spikes unrelated to earthquakes. This event was removed (d) - Files with channels of varying lengths were also removed.

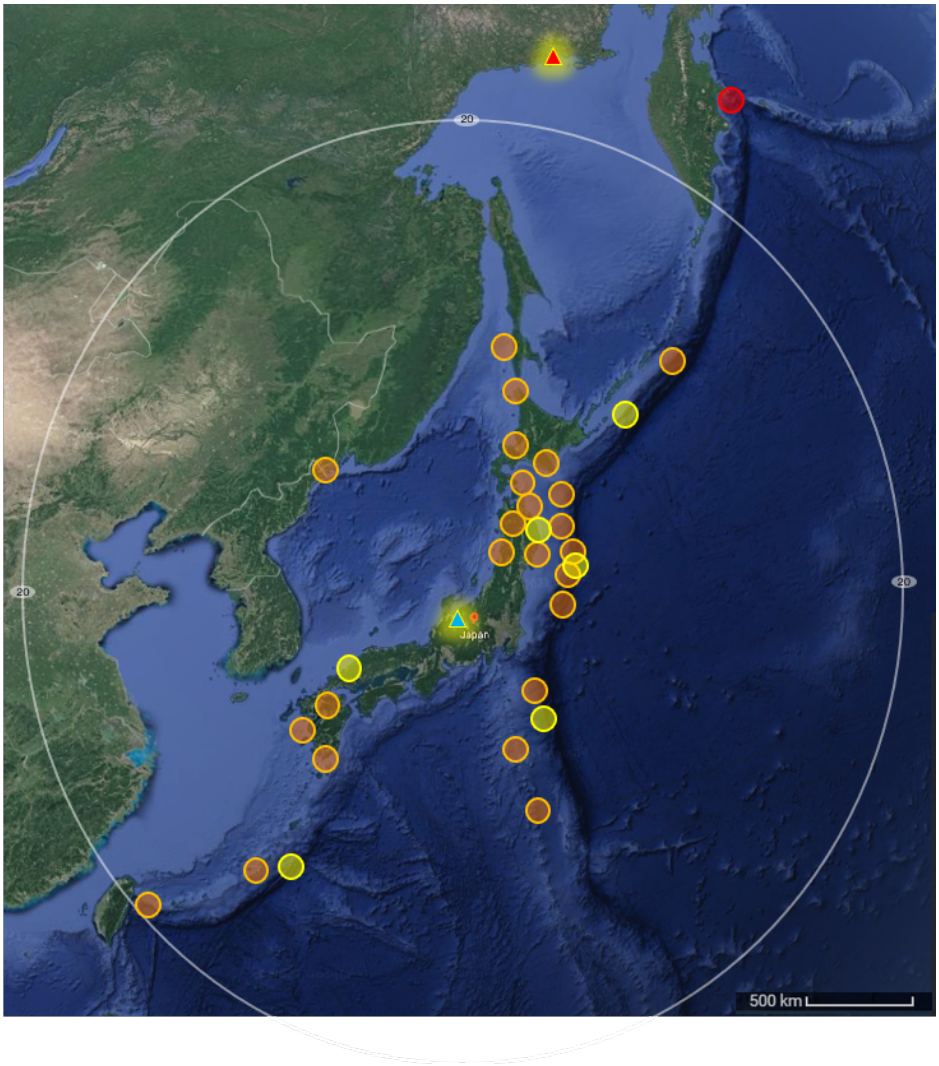

Figure 4.4: Mw $\geq$ 6 earthquakes investigated in the Japan region | Orange circles indicate the location of the epicentre for each Mw $\geq$ 6 earthquake in the training dataset. Yellow circles indicate the location of the epicentre for each Mw $\geq$ 6 earthquake in the unseen (test and validation) datasets. The red circle corresponds to the epicentre location for the single earthquake in the validation dataset that was recorded by a seismic instrument at a different seismic station (IU MA2). The location of station IU MA2 is indicated by a red and yellow triangle. All of the earthquakes other than the single event recorded by a seismometer at IU MA2 occurred within 20° of the station of interest, IU MAJO, whose location is indicated by a blue and yellow triangle. A circle with a radius of 20° from the station IU MAJO is included.

4.4 Data Preparation

The 31 earthquakes were randomly split into 24 earthquakes in the training data, 4 in the testing and 3 in the validation. The single event from station IU MA2 was investigated in the validation dataset, therefore, the validation data contained 2 events from IU MAJO.

The last 40000 samples (16.7 minutes) from each 10 hour file were labelled as 'precursors' and the first 40000 samples as 'noise'. A sliding window with an overlap of 650 samples was used to scan the labelled regions of the data for each earthquake. A window length of 16384 (2^{14}) samples or 6.83 minutes was found to produce the best result. To ensure that the noise and precursor class were entirely separate over the 10 hours investigated, the region of data assigned to each class was constrained to a small interval over the 10 hour period (i.e. 40000 samples). Small intervals in addition to a large window length resulted in very small dataset. An overlap between windows increased the number of windows in each class without having to increase the number of individual earthquakes. This method of augmentation was found to be very effective at improving generalisation and convergence of the network when investigating lab earthquake precursors. It should be noted that windows were generated separately for each event to remove the possibility that windows in the training data were from the same events as windows in the test and/or validation data.

Each window was standardised separately as opposed to standardising the whole 10 hours of seismic data. This prevented the network from detecting differences in amplitude and encourage the network to learn diagnostic patterns within the seismic signal. 1776 windows were used to train the network, 296 were used to test the network during training and 222 to validate the learned weights after training.

4.5 Results

4.5.1 Introduction

This section investigates the performance of several state-of-the-art neural networks before detailing the model that achieved the best performance on this dataset.

Selecting the same events in the training, testing and validation datasets and fixing other parameters unrelated to the network i.e. batch size, window length and optimiser ensured a fair comparison between the performance of different types of network. The validation dataset provided an unbiased evaluation of the model's fit to the training dataset and was used to compare the performance of different models. In addition, to make a fair comparison between networks, each network was trained 5 times and an average train, test and validation accuracy were recorded and used to compare models. An average was obtained as it provided a representative indication of the network's performance. As the weights and biases were randomly initialised and the input windows were shuffled prior to training, the networks may have trained slightly differently each time. A single measure of performance was therefore unreliable.

When training each of the neural networks, the weights and biases were saved when the test loss reached a minimum during training. It should be stressed that the network did not use the test loss to update the weights and biases during training. The network was trained to minimise the training loss, however, as the learned weights were validated on the test data after each epoch, the final or best weights were saved when the test loss reached a minimum during training.

Finally, it should be noted that other, unpublished networks were also investigated, however, these were not included in the report for simplicity.

4.5.2 Fully Convolutional Network (FCN)

The FCN in Wang et al. (2017), has shown significant success in time series classification, achieving better performance than other state-of-the-art approaches. This network was initially selected for this task due to its ability to generalise well and achieve exceptional performance on time series classification. Hyperparameters such as the learning rate, batch size and optimiser were tuned to obtain the best validation accuracy and loss on this dataset. The RMSprop optimiser (SN 2003) was selected with an initial learning rate of 0.0001. A batch size of 32 provided good network convergence and generalisation. The FCN achieved an average (over 5 training runs) of 54% train accuracy (loss = 0.68), 52% test accuracy (loss

= 0.69) and 50% validation accuracy (loss = 0.70). Of the windows correctly predicted, the prediction score or certainty for each window was very low (50-55%). The low train accuracy and high train loss indicated that the network was not able to learn many discriminative features separating noise windows from precursor windows in the training data. This was likely due to the simplicity of the network for the task it was given. As a result, the network was not able to effectively differentiate between the two classes in the train, test and validation datasets. It should be noted that the train accuracy did not improve above 60% at any point during training for any of the training runs. The main issue was likely that the the network was too simple to detect the complex patterns required to differentiate between the 2 classes. This was demonstrated by the low train accuracy.

4.5.3 Residual Network (ResNet)

The ResNet (He et al. 2016) is a deep residual network which involves the use of residual connections that skip one or more layers. These connections reduce the impact of the vanishing gradient problem (Glorot & Bengio 2010) by improving the flow of information through the network (Marquez et al. 2018). The ResNet 34 was applied to this dataset. This network included 34 convolutional layers (compared with 3 in the FCN), a Global Average Pooling layer or GAP layer which calculates the average output of each feature map in the previous layer and a fully connected layer. The result when using this network was an improved average training accuracy of 80% and a loss of 0.49. The average test and validation accuracies also increased to 71% (loss = 0.57) and 69% (loss = 0.55) respectively. This indicated that the network was able to learn some general characteristics. The network overfitted slightly to the training data and performed quite well on all 3 datasets.

4.5.4 Dilated Residual Network

The slightly overfitting nature of the ResNet was likely a result of the network depth. To further investigate the use of residual connections which successfully reduce the vanishing gradient problem, a dilated residual network was implemented (Yu et al. 2017), using the same skip connections as in the ResNet. In contrast to the ResNet, this model did not have a

substantial depth or complexity.

Convolutional networks for classification progressively reduce resolution of the input until the data is represented by very small feature maps which have lost a significant amount of spatial information (due to pooling operations). This loss of spatial acuity can limit classification accuracy (Wang et al. 2018). Dilation can be applied to alleviate this issue as it increases the resolution of the output feature maps without reducing the receptive field of individual neurons (Figure 4.5). By replacing pooling layers in traditional CNNs with dilated convolution, spatial resolution is preserved with depth in the network (Hamaguchi et al. 2018). Dilated convolution is convolution applied to the input with defined gaps. For example, a dilation rate of 1 is standard convolution where the dilation rate is the spacing between values in a kernel. A kernel of length 3 with a dilation rate of 2 will have the same field of view as a kernel of length 5 (Figure 4.5).

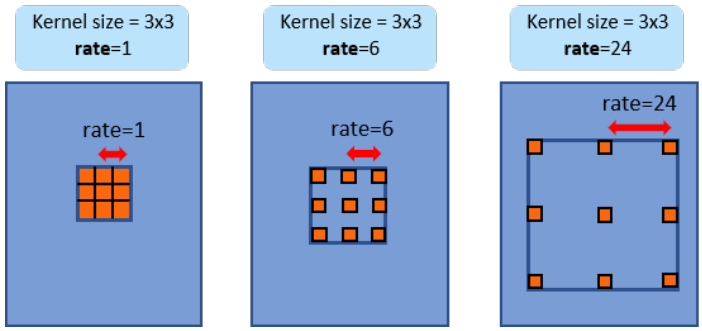

Figure 4.5: 2D dilated convolution with different rates, adapted from Xia et al. (2020).

The dilated residual network increased the training, testing and validation accuracy compared to the original FCN and the result of the training was similar to the testing and validation (73% train, 69% test, 65% validation). The only issue was that the accuracy for all three datasets was quite low. Additionally, the loss remained high with a training loss of 0.54, test loss of 0.62 and validation loss of 0.65. A high loss indicated that the network was unsure on its predictions.

4.5.5 LSTM-FCN

Long-short-term-memory (LSTM) recurrent neural networks (Gers et al. 1999) possess the ability to learn temporal dependencies in sequences. An LSTM-FCN has been proposed for

time series classification (Karim et al. 2017). In the LSTM-FCN, the fully convolutional block is augmented by an LSTM block followed by dropout. The fully convolutional block is the same as the original FCN. Simultaneously, the time series input is passed into the LSTM block which contains an LSTM layer followed by a dropout layer. The output of the convolutional block is concatenated to the output from the LSTM block and passed onto a softmax classification layer. The addition of the LSTM block resulted in a significantly longer training time and the classification result did not improve on the performance of the FCN.

4.5.6 LSTM

Using solely LSTM layers, the network took a noticeably longer time to train and performed worst of the networks tried. This indicated that convolutional layers were necessary for the classification task.

4.5.7 Additional Methods

In addition to optimising the hyperparameters to reduce overfitting and improve network performance on all 3 datasets, other techniques were investigated. Some of these included:

1. Each channel being input separately such that features were learnt from each channel individually before being fed into a multilayer perceptron (MLP) for classification (Zheng et al. 2014).
2. The filter or kernel length restricts the scale of features that can be learnt in a convolutional layer. This restriction is most significant in the first convolutional layer of the network where the receptive field is smallest. Using a variable kernel size in the first convolutional layer of the network meant that several different kernels with varying sizes were convolved with the input. The output of each convolution with different filter lengths were concatenated before being transferred through the rest of the network. This enabled features of different scales to be learnt in the first convolutional layer of the network (Cui et al. 2016).

Additional techniques did not improve network performance. The investigated networks produced quite poor results with average test and validation accuracies ranging from 50-71%.

4.5.8 Final Model

Since none of the investigated state-of-the-art networks performed extremely well, features from these networks were included in the FCN to try to improve its performance. Gradually increasing the complexity of a simple network and experimenting with different techniques proved to be the best method for generating a network that performed well on this task.

Changes Made to the FCN

1. A layer was added to the existing convolutional blocks (Figure 4.6). Originally, each convolutional block in the FCN consisted of a convolutional layer followed by a batch normalisation layer and a ReLU activation layer. Max-pooling was added to each block after the convolution operation. All but one of the convolutional blocks contained max pooling with a stride of 1. A stride of 1 indicated that the max pooling operation did not change the dimensions of the feature map. By applying max-pooling with a stride of 1, the operation concentrated the strong activations from the convolution output (feature map) and discarded the weak ones whilst maintaining the original dimensions of the output. This prevented loss of information which occurs when the dimension of the output is reduced, for example, if a stride > 1 was selected. A stride of 2 was selected in a single convolutional block as this improved the performance on the test and validation data (Figure 4.7).
2. The number of convolutional blocks was increased from 3 to 7 and, in doing so, the number of filters was increased, reaching a maximum of 256 filters in the final 2 convolutional blocks of the network (Figure 4.7).
3. Dilation (Figure 4.5) was added to all but the first 2 convolutional blocks and the dilation rate was increased with depth in the network. Dilated convolutions with a dilation rate greater than 1 produce gridding artifacts which is where adjacent units in the output are computed from separate sets of units in the input and thus have entirely different

receptive fields (Yu et al. 2017). To overcome this issue, hybrid dilated convolution was implemented, involving a dilation rate that increases and decreases in a sawtooth pattern (Wang et al. 2018). This performed slightly worse than the originally selected dilation rates which increased continuously with depth in the network. The addition of dilation did not make a significant improvement, possibly because the receptive field of the network was already large enough to contain the required information from the input to make its decision.

4. A dropout layer with a dropout rate of 0.2 was added after the fully connected layer to regularise the network (Hatami et al. 2018). This slightly improved its generalisation to the test data.
5. The kernel initialiser was changed from the default to 'random_normal' which uses a normal distribution to initialise the weights.
6. A random layer was applied directly to the input and improved the performance of the network on the validation data by an average of 10% over 5 training runs (Lee et al. 2019). The random layer is a convolutional layer which was added between the input and the rest of the neural network (Figure 4.7 and Figure 4.8). This convolutional layer was randomly initialised each epoch and its weights were normalised such that it did not significantly change the input. When convolved with the inputs, the random layer produced several slightly different versions of the inputs with each epoch. This worked similar to a data augmentation technique as it generated a larger variety of the training data. The random layer aided in generalising the network as it increased the number of visually different inputs which helped the model learn more general features or features that were consistently observed in the randomly augmented inputs. This prevented overfitting.

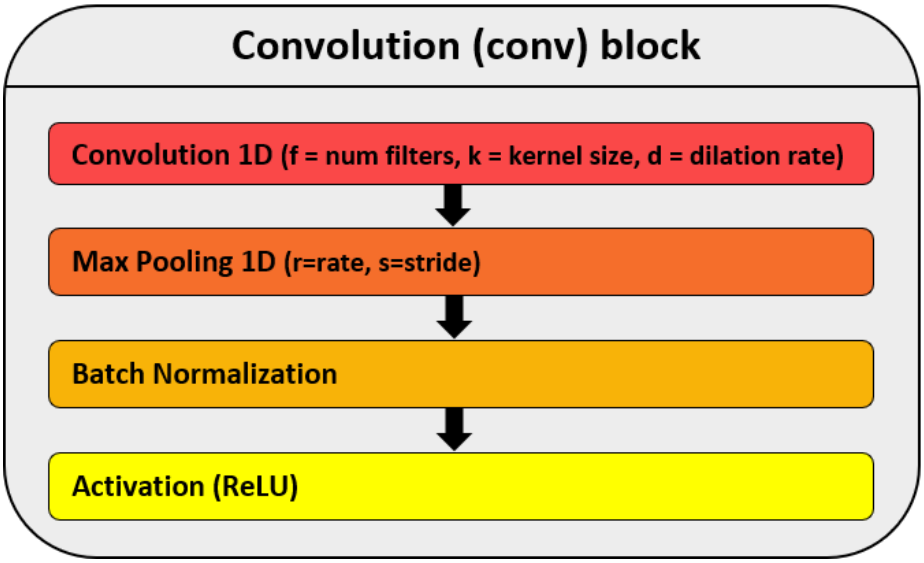

Figure 4.6: Convolution block containing 4 layers | Hyperparameters for each layer in the convolutional block are indicated.

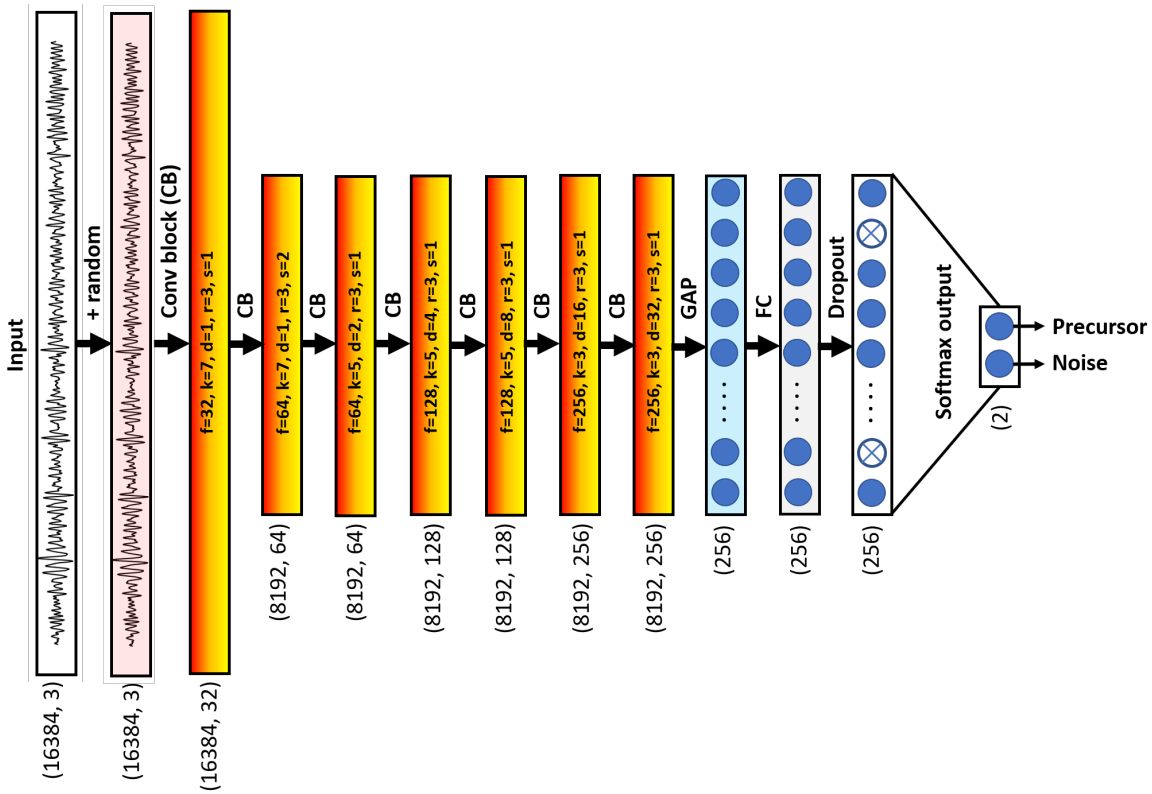

Figure 4.7: Network architecture | The input shape is indicated for each layer in the network. In the input and augmented input (Figure 4.8), this is (length of input, number of channels). In the convolutional blocks this is (length of input, number of feature maps). Finally, in the fully connected layers, this is (number of neurons). The GAP layer minimised overfitting by reducing the total number of parameters in the network. For an understanding of the parameters specified in the convolutional blocks (Figure 4.6).

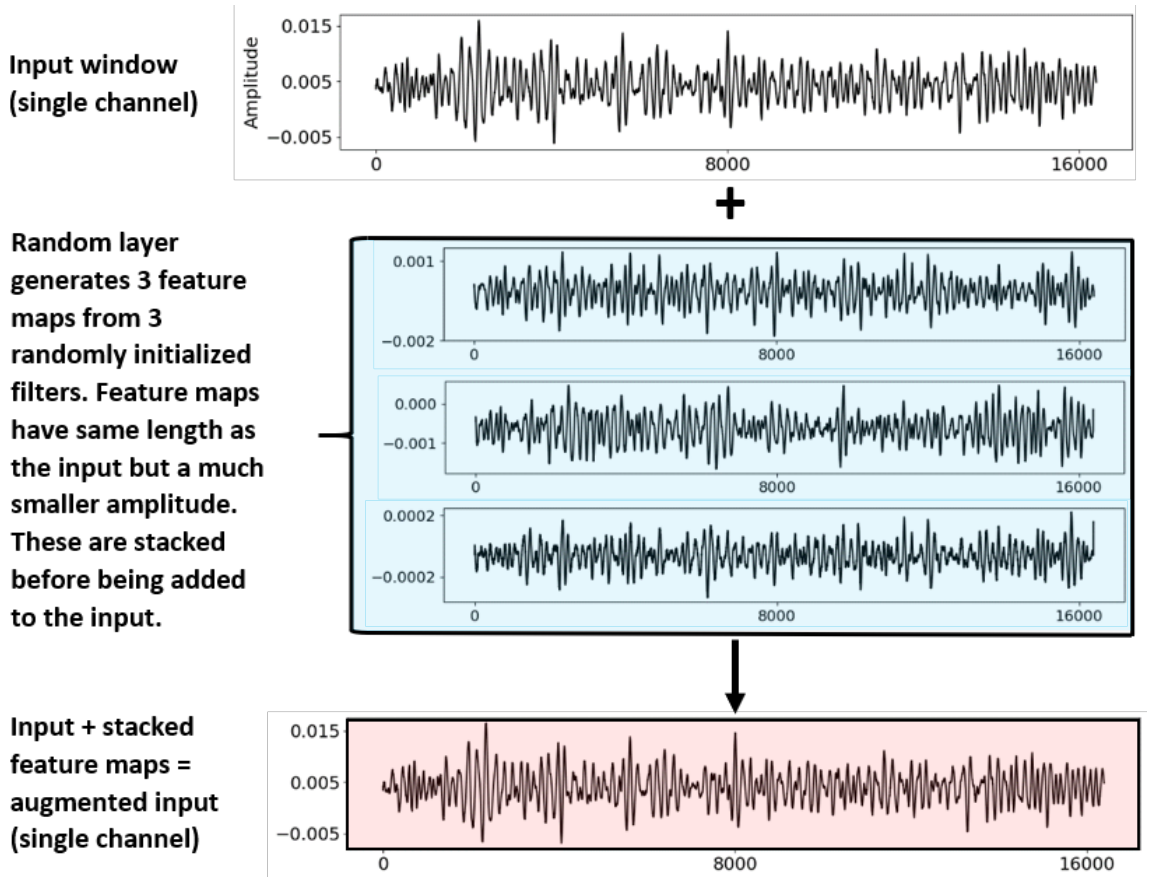

Figure 4.8: Visual explanation of how the random layer was used to augment an input window (length 16384 time samples) for a single channel of data.

Application of the random layer to the input generated an augmented version of the input. Weights in the random layer, representing the filter coefficients were randomly initialised from a normal distribution at the start of each epoch. As a result, the weights and biases in the random layer were not learned or updated during training. Three filters, each with a length of 3, were specified in the random layer and convolved with each channel in the input window, generating 3 feature maps per channel (Figure 4.8). The 3 feature maps were automatically stacked before being added to each channel in the input window. When comparing the augmented input to the original input, it was evident that the main features in the signal appeared to be unchanged and differences were most evident when investigating the finer details (Figure 4.8).

The network was trained using the binary cross-entropy loss function and the RMSprop optimisation algorithm with a learning rate of 0.0001, in batches of 32 windows with a NVIDIA GeForce RTX 2080 Ti GPU. The learning rate was reduced by a factor of 0.1 if

the loss of the test dataset did not improve over 5 consecutive epochs. A minimum learning rate of 0.00001 was selected such that the learning rate did not reduce to a value less than 0.00001. This prevented the network from training for significantly longer than was required. Training was stopped once the test loss no longer improved for 20 consecutive epochs. The network trained for 70 epochs. The weights and biases in the convolutional layers were initialised from a random normal distribution at the start of every training run. Unlike the learnable parameters in the network, the weights and biases in the random layer were randomly reinitialised every epoch.

The final network achieved an average training accuracy of 97%, an average test accuracy of 88% and an average validation accuracy of 85%. The high average accuracies indicated that the network was able to effectively learn the training data and translate the features learnt to the test and validation data. The training loss was 0.35, the test loss 0.48 and the validation loss 0.50.

Validation data is commonly used to compare network architectures. To create a test dataset that contained a larger number of earthquakes, more representative of the training data, the test and validation datasets were combined to form the test data. Retraining and testing the network on the new test dataset, the model achieved a 99% average training accuracy (loss 0.35) and an 87% average test accuracy (loss 0.50). The best weights achieved a train accuracy of 97% and a test accuracy of 89%. Figure 4.9a and Figure 4.9b indicate the confusion matrices for the best saved weights. The confusion matrix can be seen as an indicator of how reliable the algorithm is on the investigated data. It indicates the relative accuracy of the network in terms of four possible scenarios: 1. accurate prediction of an event (bottom right), 2. failure to predict an event (bottom left), 3. false prediction of an event (top right), 4. accurate prediction of no event (top left).

Due to the restrictions applied in Section 4.3 and the small fraction of data analysed over the 8 year period investigated, the confusion matrix may not provide an accurate indication of how the algorithm would perform on continuous data from 2012-2020.

(a) (b)

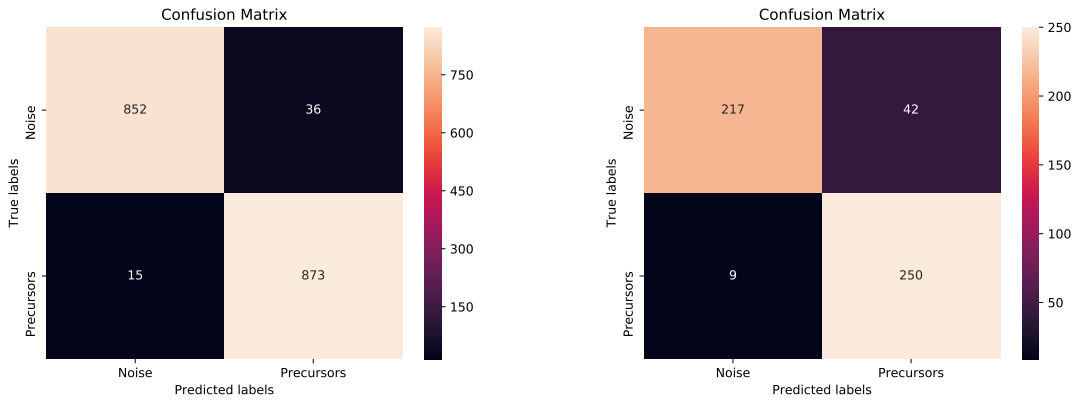

Figure 4.9: Confusion matrices indicating the performance of the classification model by summarising the prediction results on the train and test datasets | (a) Confusion matrix obtained when validating the best weights (89% test accuracy) on (a) the training dataset and (b) the test dataset. The confusion matrix indicates the relative accuracy of the network in terms of four possible scenarios: 1. accurate prediction of an event (bottom right), 2. failure to predict an event (bottom left), 3. false prediction of an event (top right), 4. accurate prediction of no event (top left). The numbers in each box indicate the number of windows classified in each scenario.

4.6 Visualisation

The task of identifying diagnostic features or patterns in the data that were used to discriminate precursors from noise was much more complex than visualising features of the input for earthquake detection in Section 3.9. Saliency maps no longer proved very useful as there were no localised regions of the input with high saliencies. Other gradient based methods such as feature map inversion and filter visualisation where the outputs are generally harder to interpret were not useful for visualising features of importance. Additionally, there were 3 channels to investigate as opposed to one. Each of the 7 earthquakes in the test dataset were investigated to determine whether a systematic difference could be identified between noise and precursor windows. Earthquakes in the test data were investigated to analyse the general features learnt by the network during training.

The correlation matrix was plotted for the whole 10 hours prior to each earthquake and for individual windows of data. The results were very consistent for all of the investigated seismic data. Figure 4.10 shows the correlation matrix for a randomly selected event from the test dataset.

The correlation matrix was a 3 by 3 matrix (a result of having 3 channels or variables) with a variance of 1 for each variable. For the example in Figure 4.10, the correlation between channel 1 and channel 0 was -0.14. This small correlation indicated a weak relationship between the channels implying that they were poorly related. The correlation was also negative such that as one increased, the other decreased. Channel 0 and channel 2 had a very weak, positive correlation of 0.03. Lastly, channel 1 and channel 2 had a very weak, negative correlation of -0.086. It was evident that for the data investigated, all three channels were very poorly correlated. A weak correlation between channels indicated that each channel provided very different information. As a result, all 3 channels had to be investigated.

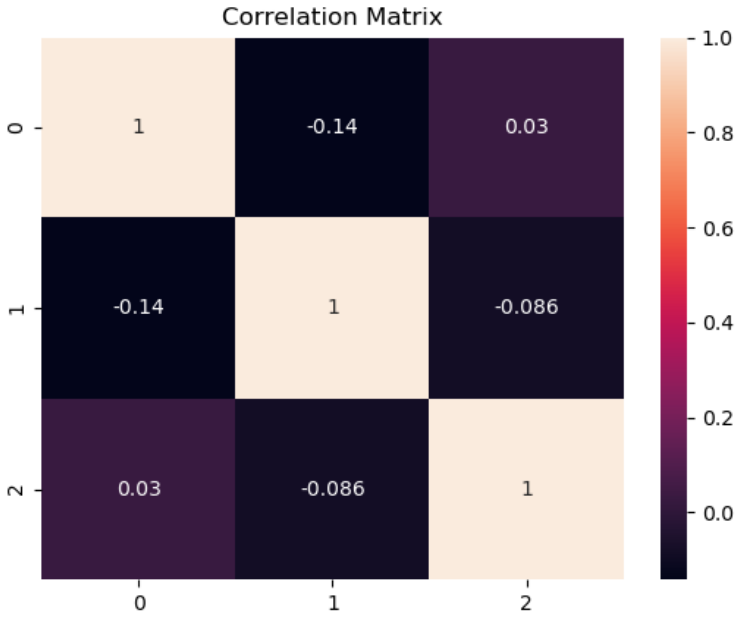

Figure 4.10: Correlation matrix for a 10-hour example in the test data.

4.7 Visualisation by Frequency Analysis

For each earthquake, 37 overlapping noise and 37 overlapping precursor windows were generated. To analyse the frequency content, frequency-amplitude spectra were obtained for the noise and precursor windows in the test data with the highest certainty/prediction score to their respective classes. The highest certainty noise and precursor windows for each of the

7 events in the test dataset were plotted (Figure 4.11). The prediction scores for each test window were obtained by validating the best weights (89% test accuracy) on windows in the test data. It should be noted that when validating weights, no ground truth exists. The results indicate the network's prediction on the input windows. Windows with the highest certainties were investigated as any discriminative features between precursor-labelled and noise-labelled windows should have been most evident. Additionally, the events in the test (unseen) dataset were investigated as opposed to those in the training dataset to visualise general instead of training-specific features which do not influence the predictive capability of the network. It should be noted that when visually comparing the noise and precursor windows from each test event (same colour plots in Figure 4.11) and between events, no systematic or significant differences were evident. This indicated that the network was required to differentiate between noise and precursor windows. As well as the certainty for each window, the R-score for each event is indicated in Figure 4.11 where:

$$R-score = \frac{TP}{TP+FN} - \frac{FP}{TN+FP} \tag{4.1}$$

TP is the number of precursor windows classified correctly, FN is the number of precursor windows classified incorrectly, TN is the number of noise windows classified correctly and FP is the number of noise windows classified incorrectly.

The R-score indicated the predictive power of the network - an R-score of 0 indicated that the network was unable to detect any discriminative features in the seismic signal separating precursors from noise and an R-score of 1 indicated that the network produced an entirely successful prediction. The R-score was calculated separately for each individual earthquake in the test data (Figure 4.11). The R-score of the network on the whole test dataset in Section 4.5 was 0.803. Interestingly, the R-score for the only earthquake recorded by the seismometer at station IU MA2 was 0.95 (Event 7 in Figure 4.11). This indicated that the network was able to effectively translate features learnt from the seismic data in the training dataset (data obtained by a single seismometer) to the seismic signal recorded by a seismometer at a different seismic station. This demonstrated that the discriminative features detected in Section 4.5 were also present within the seismic signal recorded by a seismometer at a distance of $\sim$

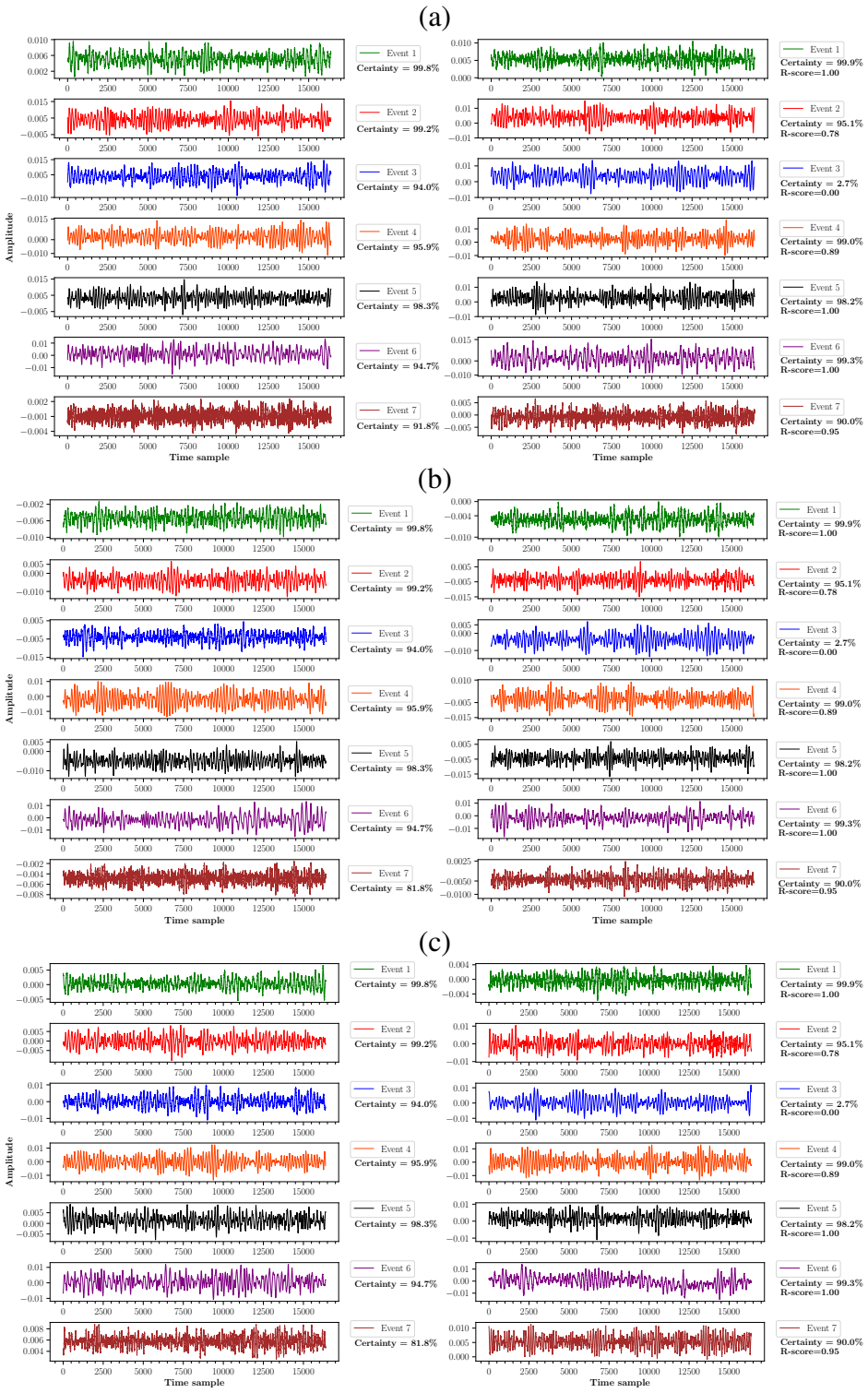

Figure 4.11: Precursor and noise windows with the highest prediction scores for each earthquake in the test data | For each earthquake, 37 overlapping windows of noise and 37 overlapping windows of precursors were generated. Each row of plots shows the precursor (left) and noise (right) window with the highest prediction score from each event in the test data. Note that the prediction score relates to the class of the window for example the certainty of a precursor window is the certainty of that window to the precursor class. The noise and precursor window from the same event were plotted using the same colour. The certainty for each window and the R-score for each event are indicated. For clarity, channel 0 (a), channel 1 (b) and channel 2 (c) were plotted separately.

2713 km from the seismometer of interest and that the learnt features were not specific to the seismic signal recorded by the seismometer of interest. A single event in the test data had an R-score of 0, see Event 3 in Figure 4.11. For this event, all noise and precursor windows were classified with a high certainty as a precursor.

The Fast Fourier Transform was computed for each window in Figure 4.11. The Fast Fourier Transform was used to quantify the input signal's frequency and phase content and each channel was investigated separately. The Fourier transform was applied to try to analyse differences in the frequency content between the two classes and identify frequency bands that may have been of importance to the network during classification. For each window, the constituent frequencies and their amplitudes were plotted.

Analysing the frequency spectra, the majority of the data constituted signal between 0.1 and 1 Hz, the frequency range of microseismic noise (Masuda et al. 2020) (Figure 4.12). Comparing the frequency spectra for the noise and precursor windows in Figure 4.11 (channels 1 and 2), no obvious differences were evident between the noise and precursor windows. The frequency spectra obtained from channel 0 showed a small anomaly between $\sim$ 2.6 and $\sim$ 3.0 Hz, commonly observed in the noise windows and absent in the precursor window (Figure 4.12). This anomaly was observed in 4 out of the 7 noise windows in Figure 4.11 and occurred at the same frequency range in each window (Figure 4.12b). This feature was the most obvious dissimilarity between the precursor and noise windows. Band-passing the input noise window between 2.7 and 2.9 Hz accentuated the anomalous waveform (Figure 4.13a).

The frequency range and the localised nature of the anomaly in the input suggested that the high frequency waveform originated from low amplitude, fast earthquake signal below the background noise level. Comparing the R-scores for each event (Figure 4.11a) with the frequency-amplitude spectrum for the selected noise window from each event (Figure 4.12b), it it became evident that the only event with an R-score of 0 (Event 3, Figure 4.11) did not contain the frequency anomaly identified in noise windows (Figure 4.12). In addition, it can also be concluded that the network must also rely on other features in the data to make its decision. This was demonstrated by the fact that the network achieved a high R-score for

events 6 and 7 (Figure 4.11) even though the frequency anomaly was absent in their noise windows (Figure 4.12b).

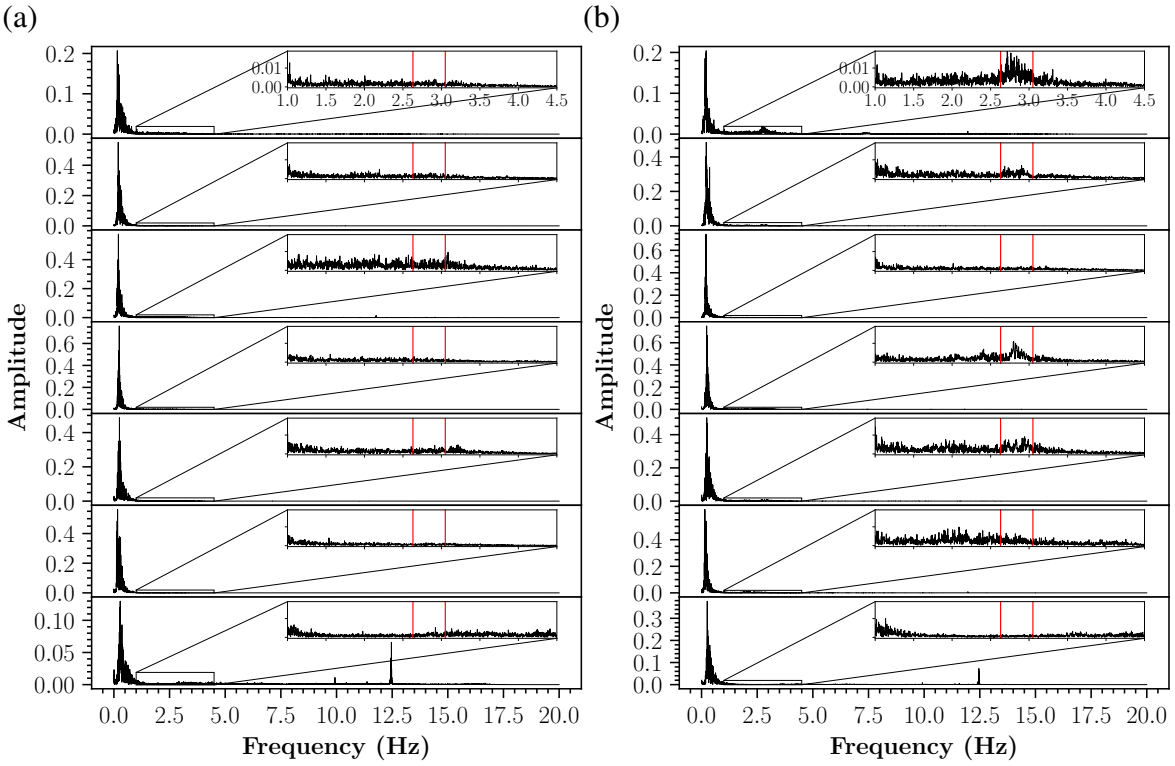

Figure 4.12: Frequency amplitude spectra for the (a) precursor and (b) noise windows from Figure 4.11a | The spectra are not labelled with Event 1-7 but this information can be obtained from Figure 4.11a. Red, vertical lines are plotted at frequencies of 2.6 Hz and 3.0 Hz and indicate the location of the frequency anomaly observed in 4/7 noise windows between $\sim$ 2.6 and $\sim$ 3.0 Hz.

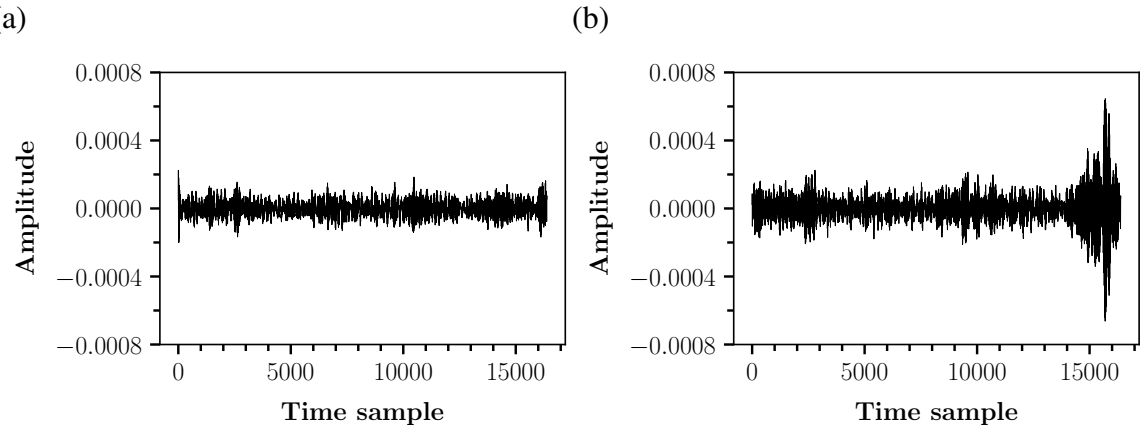

Figure 4.13: Investigating the frequency anomaly observed in channel 0 between $\sim$ 2.6 and $\sim$ 3.0 Hz (a) precursor window and (b) noise window from 'Event 5' in Figure 4.11a band-pass filtered between 2.7 and 2.9 Hz.

To gain an understanding of the significance of different frequencies in the test data, a low pass filter with a variable cutoff frequency was applied separately to each window in the whole test dataset. The best saved weights obtained in Section 4.5 (89% test accuracy) were validated on the low pass filtered test dataset and the loss and accuracy were recorded. An 8^{th} order (roll off = -48 dB /octave) low pass filter was applied (Figure 4.14). An 8^{th} order filter was selected as higher order filters introduced anomalous results at the extremes of the frequencies investigated. Selecting the highest order filter possible ensured the most rapid attenuation of frequencies above the cutoff frequency. Starting at a cutoff frequency of 20 Hz (the maximum frequency in the input data), the cutoff frequency was reduced in intervals of 0.1 Hz until only frequencies below 0.1 Hz remained in the test data. Each time the cutoff frequency was reduced, the best weights obtained in Section 4.5 were validated on the whole frequency filtered test dataset and the loss and accuracy were recorded. The change in accuracy and loss with cutoff frequency was plotted in Figure 4.15, together with the frequency-amplitude spectrum in Figure 4.12a. The results provided an indication of the frequencies in the seismic signal that were significant for discriminating noise from precursors.

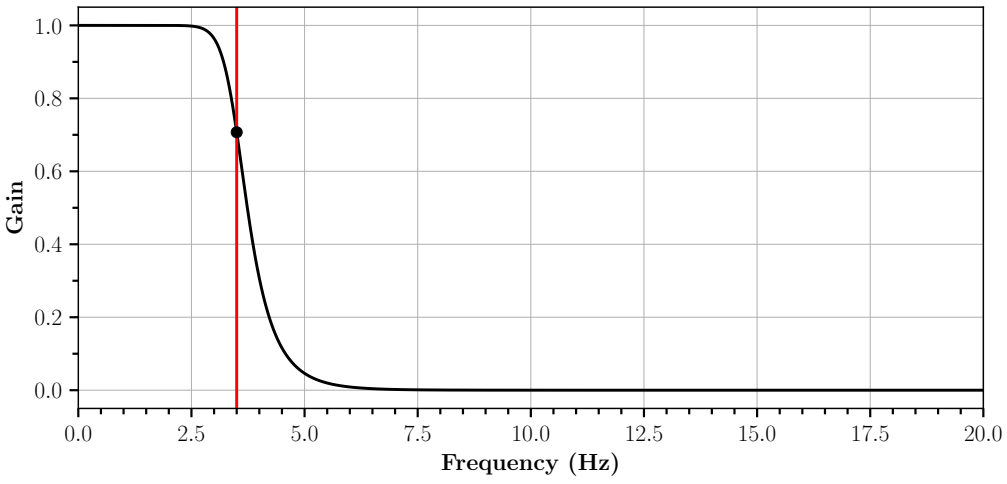

Figure 4.14: Frequency response of the low pass filter with a cutoff frequency of 3.5 Hz.

From Figure 4.15, it was evident that the accuracy on the test data started to descend at a cut off frequency of 3.5 Hz. From a cutoff of 2.6 Hz to 3.5 Hz, the accuracy of the network on the test data decreased by 0.024 (2.4%). The decrease in the accuracy of the test data was

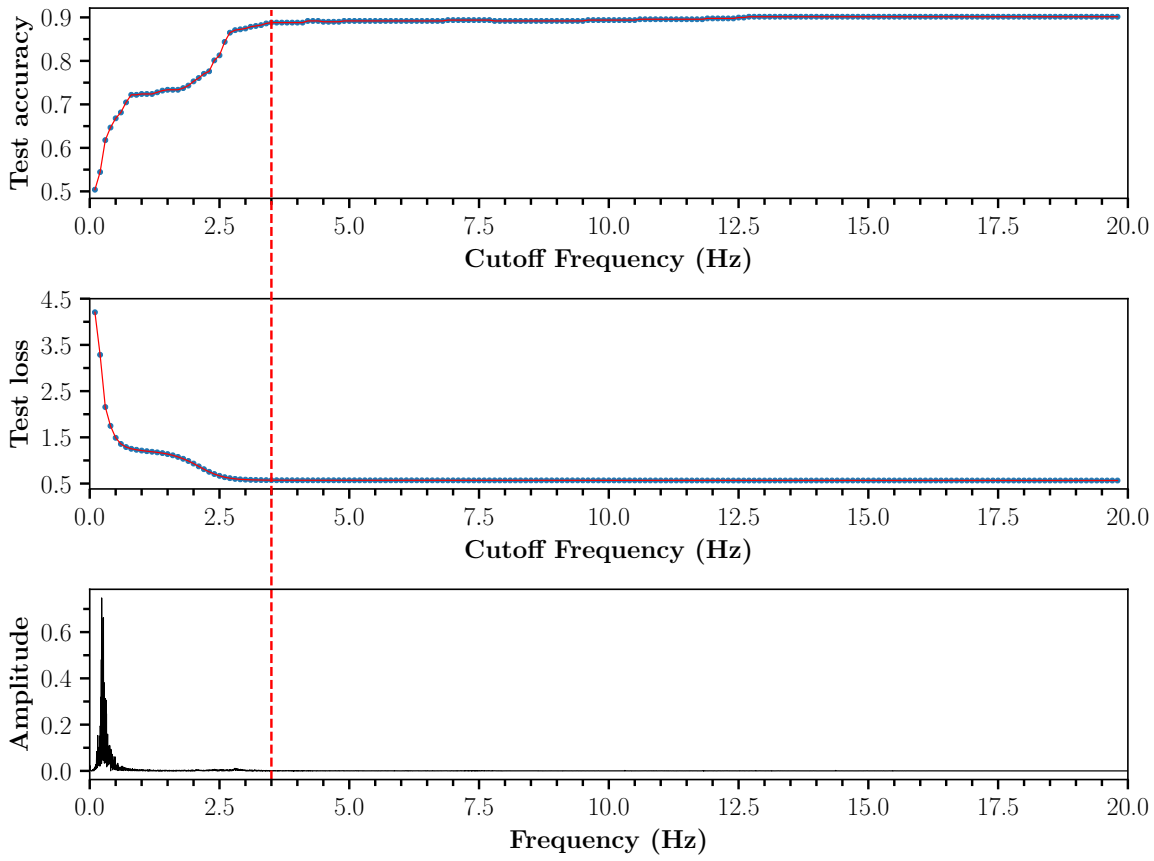

Figure 4.15: Changes in the test accuracy and test loss when applying the low pass filter in Figure 4.14 to the test dataset with a variable cutoff frequency | The frequency amplitude spectrum for a noise window containing the frequency anomaly in channel 0, same as in Figure 4.12, is plotted for comparison. The red, dashed, vertical line indicates the cutoff frequency at which the test accuracy started to decrease.

insignificant over the frequency range that coincided with the frequency anomaly observed in Figure 4.12b. This indicated that the frequency anomaly was not a highly discriminative feature and that the network was not very reliant on this frequency band when classifying noise from precursors.

From 1.8 Hz to 2.6 Hz cutoff frequency, the decrease in accuracy was more significant (0.124 or 12.4%). Over this frequency band, there was no obvious difference in the frequency-amplitude spectrum between precursor and noise windows.

The accuracy was fairly stable between 0.9 Hz and 1.8 Hz. From 0.1 to 0.9 Hz, the accuracy decreased by 0.218 (21.8%). This was clearly the sharpest decrease in the test accuracy, indicating that the low frequencies (0.1-0.9 Hz) in the signal were most significant for discriminating noise from precursors. At a cut off frequency of 0.1 Hz, the accuracy of

the network on the test data reached 0.5. As a result, frequencies smaller than 0.1 Hz did not need to be investigated. This was because, once the accuracy on the test data decreased to 0.5, no discriminative features remained in the input - the R-score became zero. An occlusion experiment was used to further the investigation of precursors (Section 4.8). Although similar, this occlusion experiment was not the same as that in Section 3.13.

4.8 Visualisation by Occlusion

For this occlusion experiment, a series of precursor windows in the test dataset with a high certainty gradient were investigated (Figure 4.16). A high certainty gradient occurred when there was a significant change in the classification certainty from one window to the next.

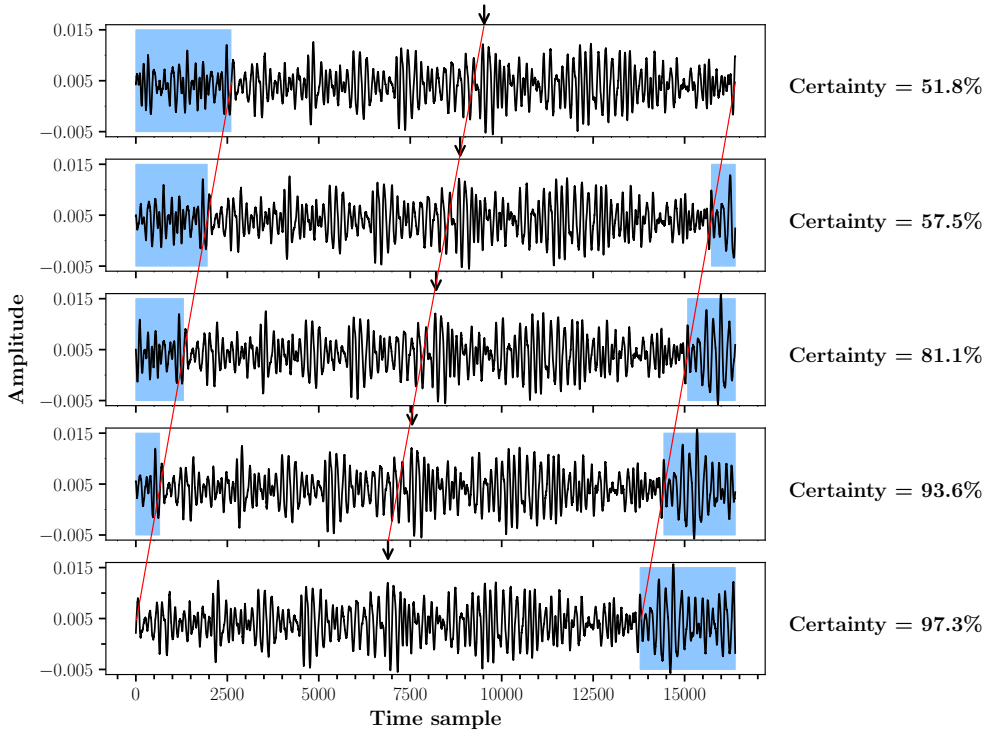

Figure 4.16: Sequential precursor windows (channel 0 only for simplicity) | The windows overlap by the sample stride of 650 time steps which might be clearer by noticing that the region not shaded in blue is the same in each plot. The arrows are plotted at the same location on each window and indicate the direction of the moving window along the length of the input. The certainty or prediction score of the network when classifying each window (all 3 channels) as a precursor is indicated. These precursor windows were obtained from Event 2 in Figure 4.11.

A large increase in certainty to the precursor class between successive windows will have coincided with an addition of features specific to the precursor class (precursor-related features) to the input or the removal of features associated with the noise class (noise-related features) from the input. As the overlap between windows was 650 samples, smaller than the window length, a change in the certainty from one window to the next could have been caused by the removal of 650 data points at the start of the window or the addition of 650 samples to the end of the window.

An occlusion experiment provided useful information regarding regions of significance in the input. This aided in identifying whether the addition and/or the removal of information from the input was important to the network when forming a decision and other regions of the input significant to the precursor class. It should be stressed that the meaning of the term 'precursor-related' varies dependent on the section in which it is defined. Throughout this and the following section, the term 'precursor-related features' refers to features specific to the labelled 'precursor' class and not to real earthquake precursors.

Occlusion sensitivity is a simple technique for understanding what features in the input were most important for classification. During occlusion, the aim was to systematically quantify the relative importance of different regions of the input to the classification result. The occlusion mask was a short length of zeros that moved along the input at a fixed stride. For each position of the occlusion mask along the input, the prediction score to the precursor class was determined by validating the best weights obtained in Section 4.5 on the occluded input. The prediction scores were plotted against the start of the occlusion mask along the input (Figure 4.17). All 3 channels were investigated or occluded at once and therefore all 3 channels contributed to the prediction scores in Figure 4.17.

The window with the greatest increase in certainty relative to the previous window was investigated (Figure 4.16). This corresponded to the window with a certainty of 81.1% (an increase of 23.6% from the previous window). The result of the occlusion experiment on this window is shown in Figure 4.17.

Different occlusion window lengths and sample strides were investigated. To localise the region or regions of the input significant to the precursor class, the length of the occlusion

mask was increased until the point at which the prediction score of the occluded window dropped below 50% in the occlusion output (Figure 4.17). A prediction score below 50% indicated that the network no longer classified the occluded input as a precursor, thereby highlighting on regions of the input crucial for precursor classification. The occlusion output first decreased below 50% when the occlusion mask was 400 samples long. To localise regions of significance, the mask length was not increased from 400 samples. Increasing the sample stride decreased the level of detail in the output. Keeping the sample stride and occlusion window length small enabled investigation of localised regions of the input significant to the network when classifying windows as precursors.

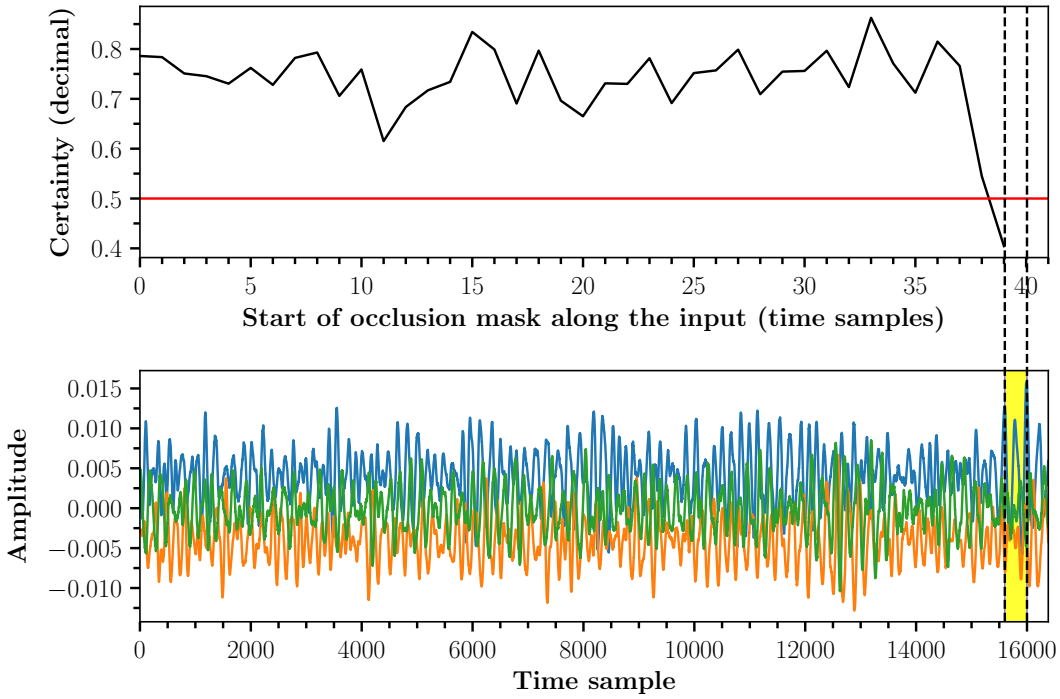

Figure 4.17: Occlusion sensitivity output and precursor window investigated | Occlusion sensitivity output (top) for the precursor window investigated (bottom). A mask length of 400 and a stride of 400 were selected. A horizontal, red line is plotted at an occlusion output of 0.5. Prediction scores below this line indicate regions of the input containing significant, precursor-related information. All 3 channels of the input window are plotted and the section of the input with high importance (region corresponding to an occlusion output < 0.5) is outlined by dashed, vertical lines marking its start and end and highlighted in yellow. With a window length of 16384 and a mask length and stride of 400, the last 384 time samples were not evaluated in this example. These 384 time samples were evaluated in Figure 4.18 when applying the same mask length with different sample strides.

A horizontal, red line was plotted in Figure 4.17 at a prediction score of 0.5 (50% certainty). When the occlusion output dropped below 50%, it indicated that the network predicted the occluded input as noise. It was evident that when the data points between 15600 and 16000 were removed, the network predicted the input as noise instead of as a precursor (Figure 4.17). Essentially, removal of these specific data points reduced the certainty of the window to the precursor class by 40.7%, enough to entirely change the classification of the input. The length of the occlusion mask (400 samples) was very small in comparison to the length of the input window (16384 samples), indicating that precursors may have been quite localised in the input. It is unsure whether this localised region was only significant when combined with other areas of the input or whether it provided significant precursor-related information on its own. This experiment demonstrated that the addition of information to the end of the window as opposed to removal of information from the start increased the prediction score of the window to the precursor class.

To investigate regions of significance to the precursor class in greater detail, smaller sample strides were selected. Setting the occlusion mask length at 400, the stride was reduced by factors of 2 from its original length of 400. The occlusion outputs for different sample strides are plotted in Figure 4.18.

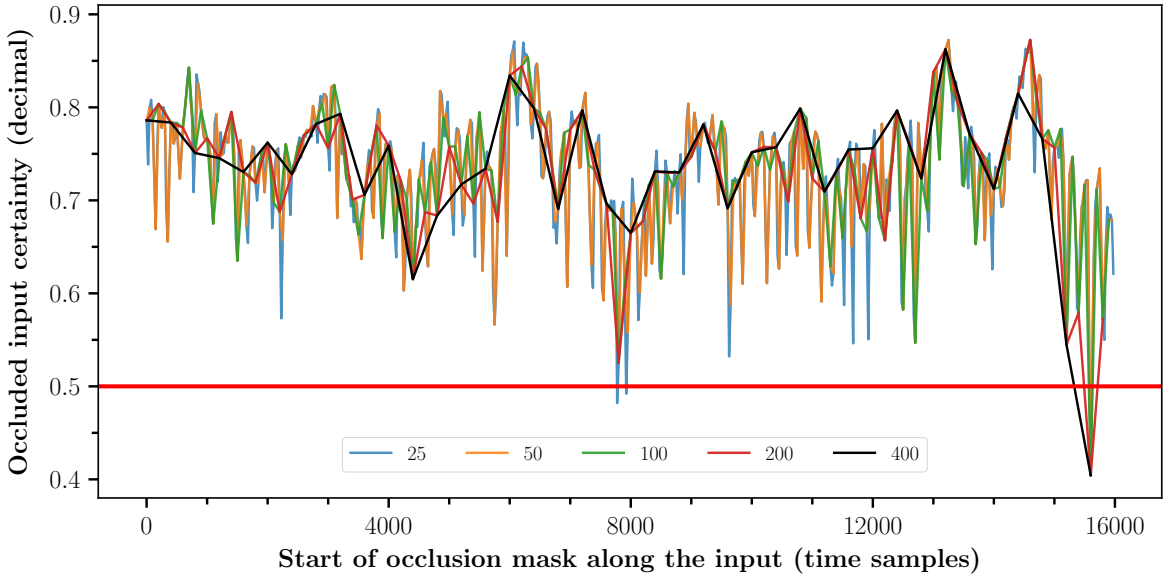

Figure 4.18: Occlusion outputs with a mask length of 400 and different sample strides indicated by different coloured outputs.

A higher level of detail in the occlusion output indicated that the region of the input from 15600-16000 samples provided the most significant precursor-related information. This was the only region of the input where the occlusion output fell much below the 50% certainty mark. This region of interest was highly specific i.e. moving the occluded interval by 25 samples in either direction, the output of the occlusion did not drop below 50% (Figure 4.18).

When increasing the level of detail in the occlusion output, it became evident that another region of the input decreased the classification certainty below 50%. This occurred from 7775-8175 samples and 7925-8325 samples, however, with much less significance (Figure 4.18).

During the occlusion experiment, all 3 channels were occluded before being input to the network. To further the investigation, each channel was investigated separately by occluding a single channel at a time and monitoring the occlusion output. Increasing the sample stride, it remained evident that the signal between 15600 and 16000 time samples contained the most significant precursor-related information. As a result, the original sample stride of 400 was selected for ease of comparison with the result in Figure 4.17. The occlusion outputs for each occluded channel were compared (Figure 4.19).

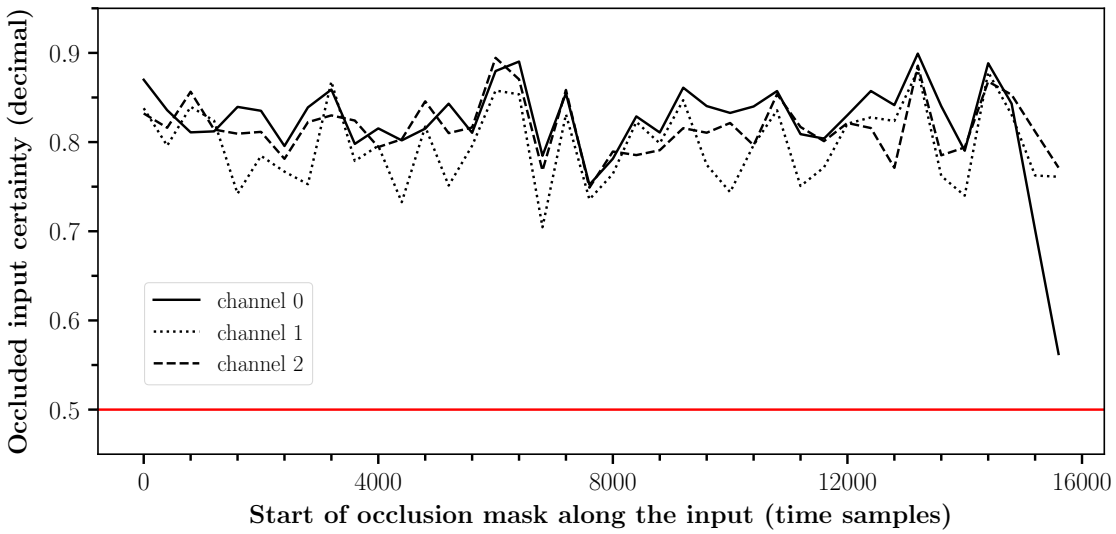

Figure 4.19: Occlusion sensitivity output when each channel was occluded separately | The 50% certainty mark is indicated with a red, horizontal line.

From Figure 4.19, it was evident that the occlusion output did not drop below 0.5 unless all 3 channels were occluded. This indicated that the network used patterns between channels such as similarities or differences as well as channel specific patterns. As a result, all 3 channels contributed to the network predictions. However, when comparing the occlusion outputs in Figure 4.19 for each channel, it became clear that channel 0 had a greater contribution to the network's decision. Occluding the interval from 15600-16000 in channel 0 was enough to reduce the certainty to the precursor class from 81.1% (without any occlusion) to 56.2% certainty, almost surpassing the 50% mark (Figure 4.19). The other 2 channels did not significantly reduce the prediction score over any occluded interval. Channel 0, therefore, provided precursor-related information that channels 1 and 2 did not.

Investigating channel 0 and the region of the input where the certainty dropped significantly below 50% in Figure 4.17, it was evident that there were two spikes, almost 400 samples apart that had to both be occluded for the certainty to drop below 50% (Figure 4.20).

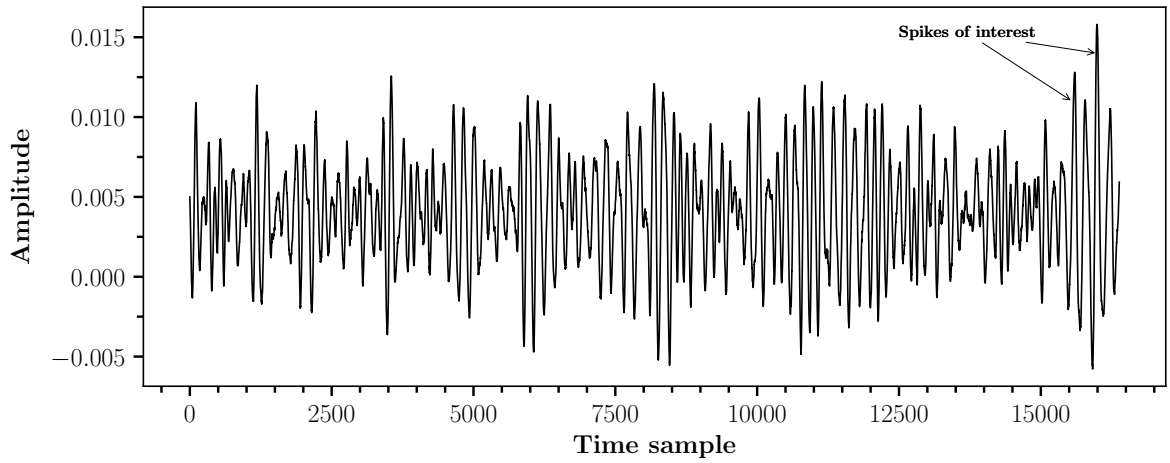

Figure 4.20: Channel 0 of the precursor-labelled window under investigation | When the region of the input containing the two spikes of interest was occluded (set to zero) in all 3 channels, the window was predicted as noise instead of precursors.

These visualisation techniques constrained the frequency range over which precursor-related features were most dominant. From these investigations, it was clear that the discriminative features separating noise windows from precursor windows were largely unrelated to high frequency earthquake signal. Instead, discriminative features coincided with the frequency range of microseismic noise and low frequency earthquakes (Masuda et al. 2020).

4.9 Precursor Frequency Analysis

The frequency content of the precursor window in Figure 4.20 was investigated to determine whether the 'spikes of interest' coincided with a frequency anomaly. The frequency content of the sequential precursor windows in Figure 4.16 were investigated using the Short Time Fourier Transform (STFT) (Figure 4.21). As opposed to indicating the frequency and phase content for a whole window, the STFT can be used as a method of quantifying the change of a nonstationary signal's frequency and phase content over time. The STFT of a signal is calculated by sliding an analysis window over the signal and calculating the discrete Fourier transform of the windowed data. The window moves over the signal at a fixed stride. A window length of 1000 was selected with a stride of 500. The STFT of the sequential precursor windows in Figure 4.16 are shown in Figure 4.21.

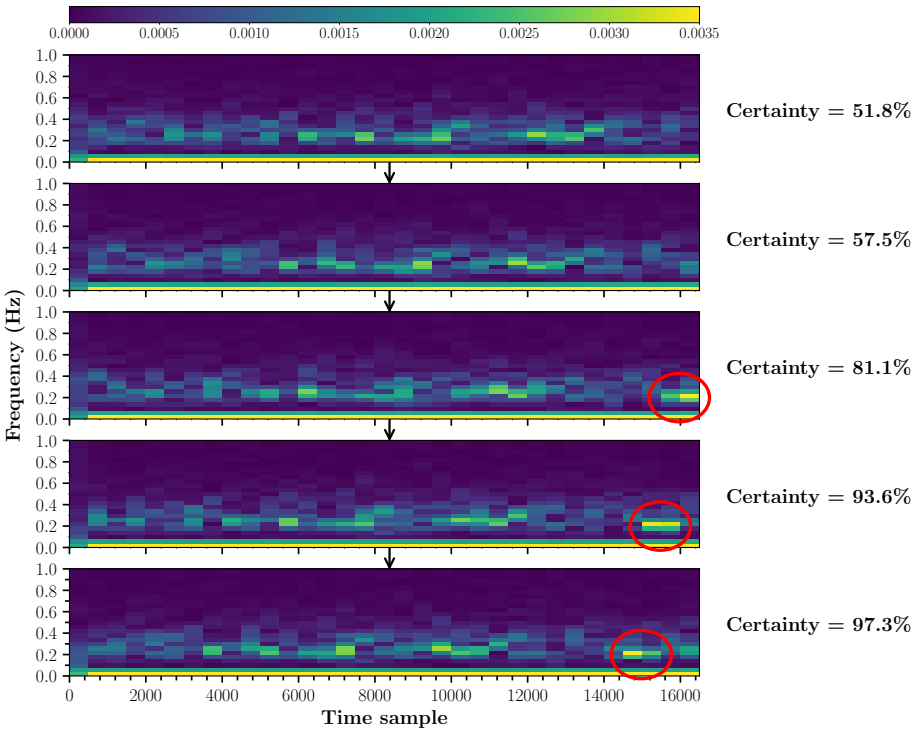

Figure 4.21: Short Time Fourier Transform for the sequential precursor windows (channel 0 only) in Figure 4.16 | The STFT was calculated by sliding an analysis window of length 1000, stride 500 over each of the sequential windows and obtaining the discrete Fourier transform of the windowed data. The certainty or prediction score of the network when classifying each window (all 3 channels) as a precursor is indicated. The red circles highlight a localised region of increased amplitude of frequencies 0.16Hz and 0.2Hz.

Comparing Figure 4.21 with the results obtained in Section 4.8, the sudden increase in certainty (57.5%-81.1%) between successive precursor windows could be attributed to an increase in the amplitude of frequencies 0.16 Hz and 0.2 Hz in channel 0. This amplitude increase appeared to be localised in the input, occurring over only one or two Fourier analysis windows (a section of the input with a length of 1000-1500 time samples). The location of the anomaly coincided extremely well with the region of interest identified in Section 4.8, suggesting that the precursor-related features within this region of interest were associated with the high amplitude, low frequency anomalies identified in Figure 4.21.

The frequency anomalies at 0.16 Hz and 0.2 Hz in Figure 4.21 can be considered significant to the precursor class for a single earthquake in the test dataset. To determine the importance of these frequency anomalies in distinguishing noise from precursors prior to all of the investigated earthquakes, the frequency-amplitude spectra for noise and precursor windows were obtained separately for each event in the train and test datasets (all 3 channels). For each dataset (train and test), the cumulative sum of the frequency responses for all events and their 3 channels were calculated separately for noise-labelled and precursor-labelled data and plotted on the same figure for ease of comparison (Figure 4.22).

(a) (b)

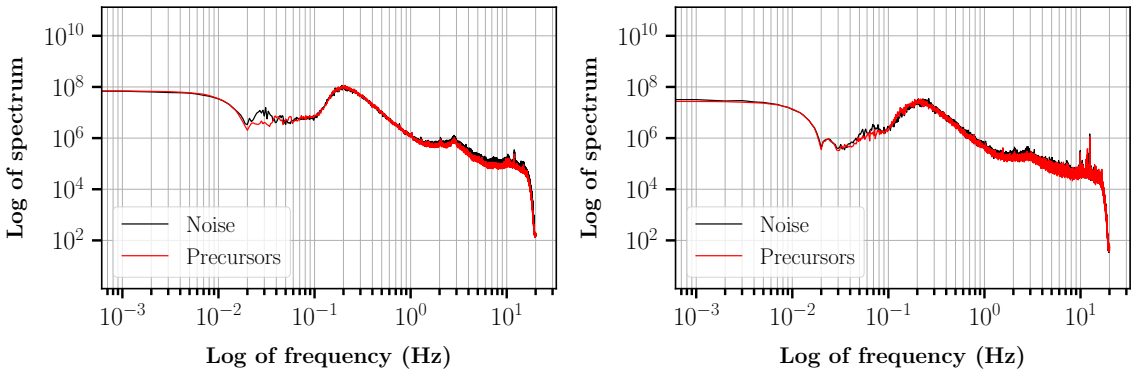

Figure 4.22: Logarithmic cumulative frequency spectra for windows labelled as noise and windows labelled as precursors | The spectra are plotted separately for (a) the train and (b) the test datasets.

No obvious differences were evident when comparing the cumulative frequency responses for noise and precursor data in the training and test datasets. Although some small differences between noise and precursor data occurred in the very low frequencies of the train-

ing dataset ($\sim$ 0.02 Hz - 0.04 Hz), these did not occur in the test data (Figure 4.22). Any non-systematic differences (differences not evident in both datasets) would unlikely have contributed to the classification result.

To further investigate any systematic and significant frequency differences between noise and precursor data in the train and test datasets, the relative percentage difference between the cumulative noise and precursor frequency responses were calculated for all 3 channels in the train and test datasets. The relative percentage difference was obtained by computing the difference between the cumulative precursor and noise spectra and normalising by the cumulative noise spectrum. The results for both the train and test datasets are shown in Figure 4.23 where the dots are the results and the curves are smoothed versions of the results.

The smoothed versions were obtained using Savitzgy-Golay smoothing with a width of 0.062 Hz (Press & Teukolsky 1990). Differences between the cumulative precursor and noise frequency responses were evident both in the test and train datasets. The frequencies at which significant and systematic differences occurred are indicated by dashed, vertical lines (Figure 4.23). These differences were evident at approximately 0.16 Hz and 0.21 Hz,

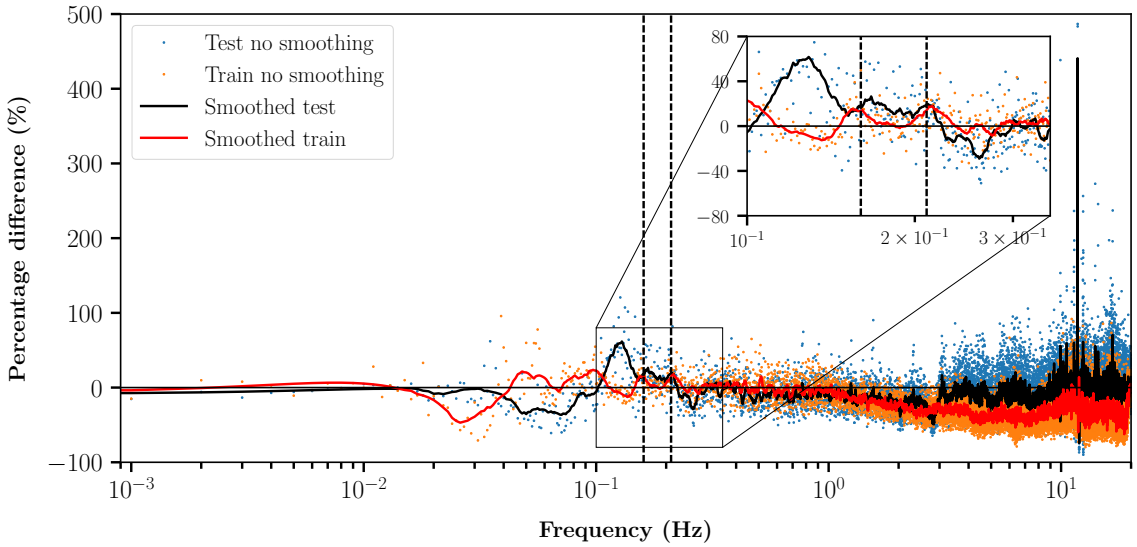

Figure 4.23: Relative percentage difference between the cumulative frequency spectra in Figure 4.22 for precursor-labelled and noise-labelled data | The test and train results are plotted on the same graph for ease of comparison. The dashed, vertical lines are plotted at frequencies 0.16 Hz and 0.21 Hz coinciding with significant amplitude differences between precursor and noise data in both the train and test datasets (peaks in the smoothed plots). A horizontal, black line is plotted at a 0% difference.

indicated by peaks in both of the smoothed plots.

The results obtained in Figure 4.21 and 4.23 coincided well, indicating two low frequencies that provided information for discriminating precursor windows from noise windows in the train and test data. From these investigations, it can be concluded that frequencies of $\sim$ 0.16 Hz and $\sim$ 0.21 Hz were significant during classification. The huge spike in amplitude at $\sim$ 12 Hz in the smoothed test plot in Figure 4.23 was irrelevant to the classification result. This can be concluded from Figure 4.15 which demonstrates that frequencies above 3.5 Hz did not significantly affect the prediction score on the test dataset.

4.10 Investigating Changes in the Significance of Precursor-Related Features with Earthquake Proximity

In Section 4.5, the first 16.7 minutes (40000 samples) of the 10 hour period of seismic data prior to each of the investigated earthquakes was labelled as noise and the final 16.7 minutes was labelled as precursors. When analysing this data, the network achieved a training accuracy of 97% and a test accuracy of 89%. A high train and test accuracy indicated that the network learnt general features of the training data, translatable to the test data.

To determine whether the network was able to discriminate noise (interval 1 in Figure 4.24) from other intervals over the 10 hour period, the performance of the network was investigated when the original noise data (interval 1) was classified from data in intervals 1-36 which were labelled as precursors (Figure 4.24). Windows of data were generated in each interval the same as previously in Section 4.5. This investigation enabled an understanding of the intervals that systematically contained features discriminative of the data in interval 1. In this context, systematic features refers to features significantly evident in both the train to the test datasets.

The network was retrained on the original noise data (interval 1) and separately on each interval from 1-36 which were labelled as precursors. The network was trained 5 times, separately for each interval, and the average train and test accuracies were calculated. The average test accuracy and the standard deviation were plotted against the start of the corre-

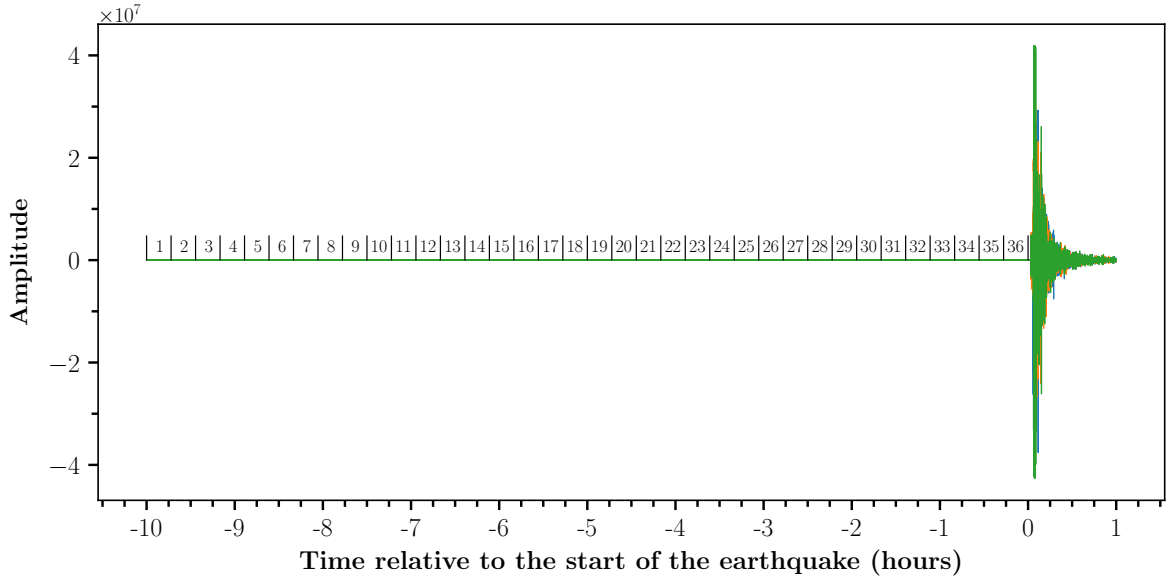

Figure 4.24: Example Mw $\geq$ 6 earthquake from the test dataset | 16.7 minute intervals over the 10 hours before the start of the earthquake are indicated by black, vertical lines where the length of time between each line is 16.7 minutes. Each interval is labelled with a value from 1-36 for ease of reference. The Mw $\geq$ 6 event occurs immediately after interval 36.

sponding 16.7-minute interval investigated (Figure 4.25). The test accuracy was investigated because it measured the ability of the network to discriminate noise from precursors on unseen data, providing a reliable evaluation of the network's performance. As the network had not experienced any generalisation issues on this dataset, changes in the test accuracy indicated how well the training dataset represented the test dataset and therefore how effectively the network was able to translate features learned during training to the test data. It should be stressed that during this experiment, data in the noise class remained fixed. As a result, the average test accuracies in Figure 4.25 reflected overall changes that occurred within the intervals labelled as precursors.

From Figure 4.25, it was evident that as the interval of the data labelled as a precursor approached the start of the earthquakes, there was a gradual increase in the test accuracy starting 3.3 hours prior (interval 25). Over the 3.3 hours prior to the earthquakes, the average test accuracy increased from 55% to 87%, a significant increase occurring over a short timespan. Considering that 55% test accuracy indicates an almost random prediction, significant changes must have occurred within the intervals labelled as a precursor over the 3.3 hours prior to the start of the investigated earthquakes. As the network was retrained on signal in

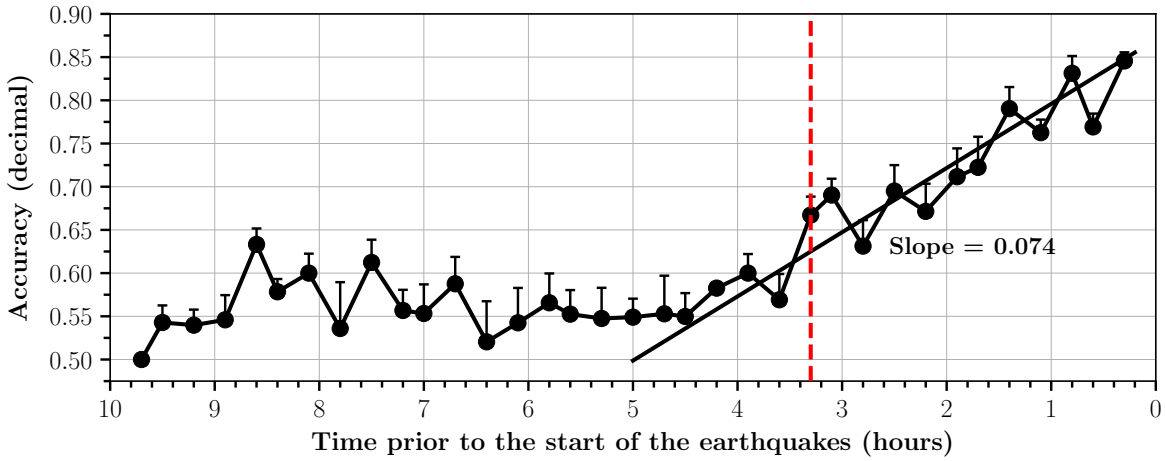

Figure 4.25: Graph showing changes in the average test accuracy when classifying signal in interval 1 labelled as noise from signal in intervals 1-36 labelled as precursors (Figure 4.24). | The average test accuracy is plotted against the start of the corresponding 16.7-minute interval investigated (labelled as a precursor). Standard deviation error bars were plotted on one side of the data points to improve clarity. The red, dashed, vertical line indicates the start of an increase in the test accuracy.

each interval, the low test accuracies in intervals 1-25 do not indicate that features discriminative of noise data in interval 1 did not exist significantly within these intervals prior to the earthquakes in the test data. A low test accuracy could have been attributed to the fact that discriminative features did not occur significantly in these intervals prior to earthquakes in the training data and therefore were not learnt during training. Intervals close in time to the noise interval may have been too similar and therefore difficult to discriminate. To conclude, this figure demonstrates that features discriminative of data in the noise class were more systematically observed prior to earthquakes in the train and test datasets with earthquake proximity. The results do not reflect changes that occurred prior to individual earthquakes.

An additional observation is that there was a subtle and gradual decrease in the standard deviation over the final 3.3 hours (interval 25-36). A small standard deviation indicated that the network learnt very similar features with each training run.

In Section 4.5, data selected 10 hours prior to the investigated earthquakes (interval 1) was labelled as noise. As the data in the noise class was associated with each of the Mw $\geq$ 6 earthquakes, the high test and train accuracy obtained in Section 4.5 (89% test accuracy) could be attributed to windows in the noise class containing precursor-related features (features discriminative of noise unrelated to Mw $\geq$ 6 earthquakes) more significantly than

windows labelled as precursors or vice versa. Note the change in the definition of 'precursor-related'. For example, the results in Section 4.9 provided an indication of two frequencies that were higher amplitude in the precursor windows. Assuming this feature is correlated to the investigated Mw $\geq$ 6 earthquakes, it is uncertain whether this feature is more closely associated with noise unrelated to Mw $\geq$ 6 earthquakes or with potential precursors to Mw $\geq$ 6 earthquakes.

Assuming that the network detected features correlated to the earthquakes, to determine whether these features were more or less significant with earthquake proximity, noise windows were selected from data unassociated with Mw $\geq$ 6 earthquakes. The newly selected noise windows were trained and tested against windows obtained from each interval in Figure 4.24 which were labelled as precursors.

4.10.1 Utilising Noise Unrelated to Mw $\geq$ 6 Earthquakes to Investigate Changes in the Significance of Precursor-Related Features with Earthquake Proximity

The 10-hour period prior to the Mw $\geq$ 6 earthquakes was further investigated by selecting noise data unrelated to any Mw $\geq$ 6 earthquake occurring within 30° from the seismometer of interest. Using the same window length and overlap as previously, 1147 noise windows were obtained from 31 randomly selected, 16.7-minute intervals occurring between 2012 and 2020. 31 earthquakes were investigated in Section 4.5, therefore the same number of noise windows were generated as precursor windows. The noise intervals were selected during time periods when no Mw $\geq$ 6 earthquakes occurred within 30° from the seismometer of interest and within 48 hours of the 16.7-minute noise intervals selected. In accordance with the constraints applied in Section 4.3, noise intervals containing impulsive earthquake signal above the noise level were not selected and 48 hours was assumed enough time to significantly reduce the influence of Mw $\geq$ 6 earthquakes on the seismic data. The same number of noise windows were generated as precursor windows. Precursor windows were obtained from each interval of seismic data in Figure 4.24. Each interval was investigated

separately by retraining and testing the network to obtain an average test accuracy over 5 training runs. The data in the noise class remained fixed throughout the experiment. The result is shown in Figure 4.26.

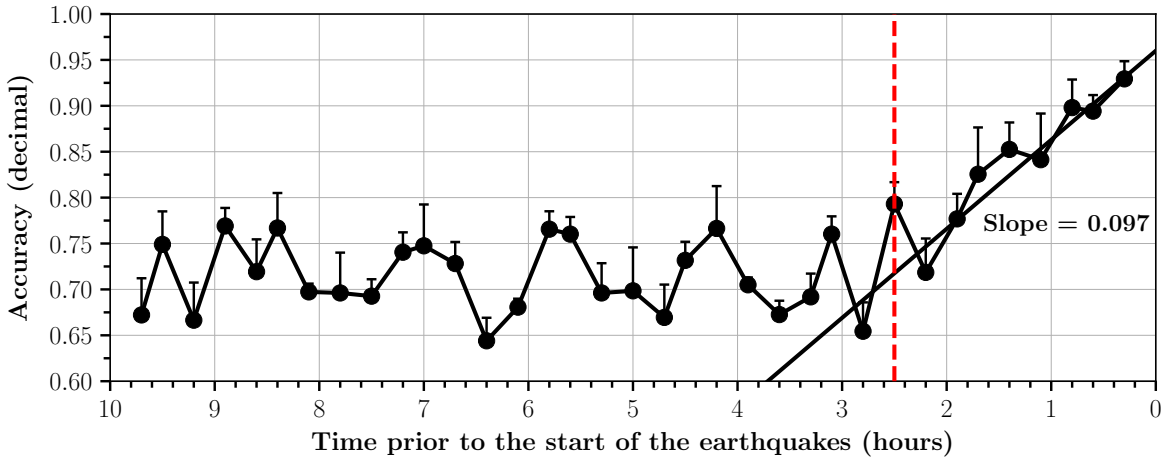

Figure 4.26: Graph showing changes in the average test accuracy when classifying signal unassociated with Mw $\geq$ 6 earthquakes (labelled as noise) from signal in intervals 1-36 (labelled as precursors) (Figure 4.24). | The average test accuracy is plotted against the start of the corresponding 16.7-minute interval investigated. Standard deviation error bars were plotted on one side of the data points to improve clarity. The red, dashed, vertical line indicated the start of the increase in the test accuracy.

Figure 4.26 indicated that there was a gradual increase in the test classification accuracy starting 2.5 hours prior to the start of the earthquakes. Similar to the results in Section 4.10, the highest test accuracy occurred when interval 36 was labelled as a precursor. This indicated that data in interval 36 contained features discriminative of noise unrelated to Mw $\geq$ 6 earthquakes most significantly. These results confirmed that precursor-related features were most dominant in the intervals immediately prior to the start of the earthquakes, becoming more systematically observed with earthquake proximity and, finally, becoming most significant in the interval immediately prior to the investigated earthquakes.

Overall, the test accuracy in Figure 4.26 was higher than in Figure 4.25. In Figure 4.26, the average test accuracy when classifying noise from the seismic data in interval 36 was 93%, 5% higher than previously (Figure 4.25). This result indicated that there was more dissimilarity between noise unrelated to Mw $\geq$ 6 earthquakes and data in interval 36 compared with data 10 hours prior to the Mw $\geq$ 6 earthquakes (interval 1) and data in interval 36. Additionally, in Figure 4.26, the test accuracy remained mostly between 65% and 75%,

whereas, in Figure 4.25, the test accuracy primarily fluctuated between 50% and 60% before increasing in the 3.3 hours prior to the start of the earthquakes. This indicated there was some difference between seismic data unrelated to Mw $\geq$ 6 earthquakes and data 10 hours before the Mw $\geq$ 6 earthquakes, suggesting that precursor-related features existed 10 hours before the earthquakes, however, less systematically than immediately prior.

4.10.2 Understanding the Significance of Precursor-Related Features Detected in Interval 36 with Earthquake Proximity

The best algorithm achieved a training accuracy of 98% and a test accuracy of 96% (Figure 4.27). These weights were achieved when classifying windows of seismic data unrelated to Mw $\geq$ 6 earthquakes from windows of seismic data in interval 36 in Figure 4.24. This result was interesting because features discriminative of noise were most systematically detected when interval 36 was investigated (labelled) as a precursor.

(a) (b)

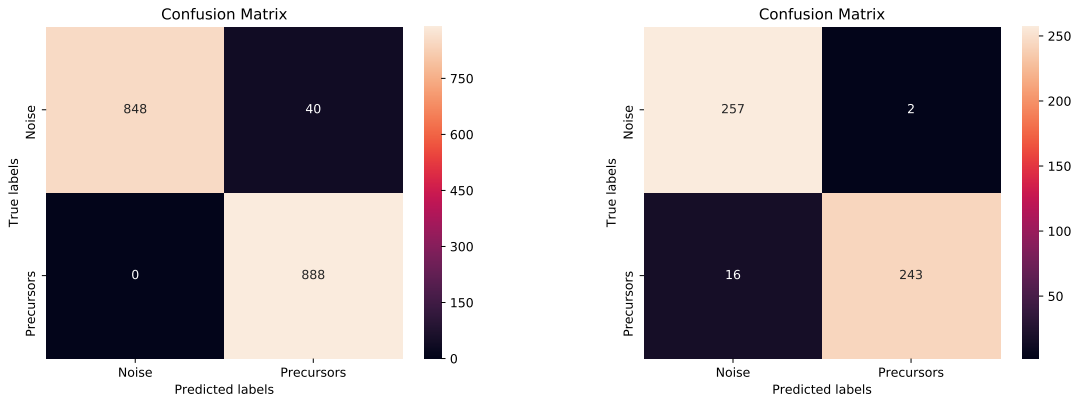

Figure 4.27: Confusion matrices indicating the performance of the classification model by summarising the prediction results on the train and test datasets | (a) Confusion matrix when validating the best weights obtained in Section 4.10.1 (96% test accuracy) on (a) the training dataset and (b) the test dataset. The confusion matrix indicates the relative accuracy of the network in terms of four possible scenarios: 1. accurate prediction of an event (bottom right), 2. failure to predict an event (bottom left), 3. false prediction of an event (top right), 4. accurate prediction of no event (top left). The numbers in each box indicate the number of windows classified in each scenario.

To determine whether the precursor-related features detected in interval 36 were translatable to other intervals over the 10-hour period investigated prior to each earthquake, the best weights (96% test accuracy) were validated on each 16.7-minute interval prior to the 31 investigated earthquakes (Figure 4.28).

When validating weights, no ground truth exists. The results indicate the network's prediction on windows of data generated over each interval. To evaluate the significance of the precursor-related features, all windows in each interval were labelled as a precursor and the noise class was discarded from the investigation. Simply put, windows generated at each interval in Figure 4.24 for all 31 earthquakes were input separately (for each interval) to the neural network. The network used the weights obtained from the best algorithm (96% test accuracy) to predict the class of each window. The result is a fraction of the number of windows predicted as a precursor for each interval over the total number of windows generated at each interval.

From the results shown in Figure 4.25 and Figure 4.26, it was evident that precursor-related features were most significant in interval 36. The presence of precursor-related features detected in interval 36 were investigated with time prior to the start of the earthquakes

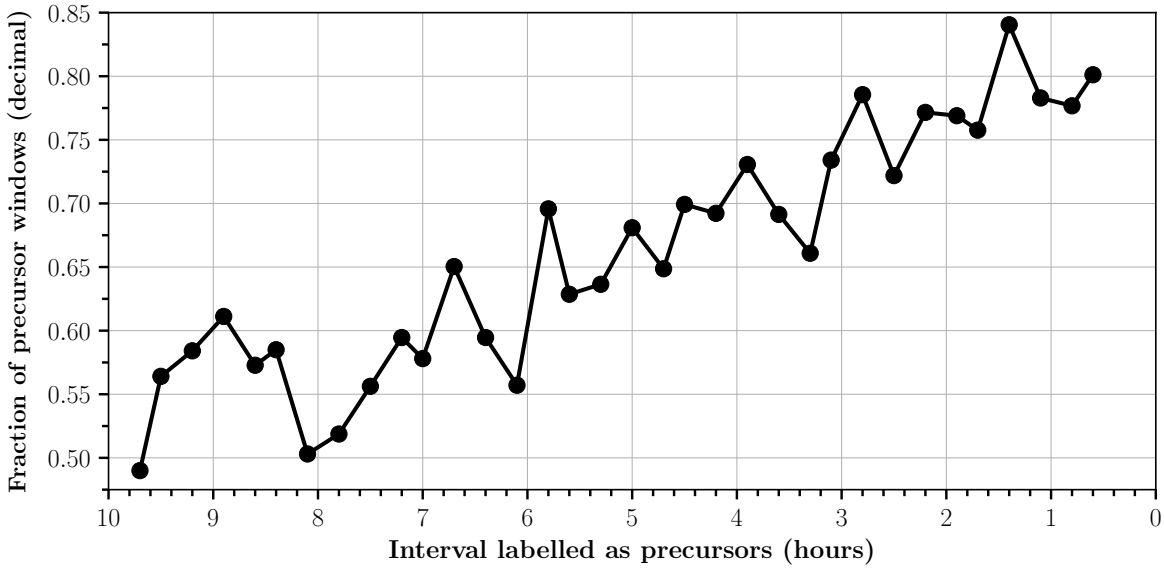

Figure 4.28: Best saved weights (96% test accuracy) validated on each interval in Figure 4.24 (except interval 36) labelled as a precursor prior to the investigated earthquakes | Each data point indicates the number of windows classed as a precursor in a single interval as a fraction of the total number of windows generated for each interval. This fraction is plotted against the start of the corresponding 16.7-minute interval investigated.

(Figure 4.28). The results of this experiment determined the significance of the features detected in interval 36 to each interval over the 10 hour period prior to the investigated earthquakes. The best weights were obtained by discriminating noise unrelated to Mw ≥ 6 earthquakes from seismic signal in interval 36. As a result, the fraction of windows classified as a precursor in each interval in Figure 4.28 indicated the proportion of windows that contained features more representative of precursor-related signal in interval 36 than of noise unrelated to Mw ≥ 6 earthquakes.

An increase was observed in the fraction of windows classified as a precursor with earthquake proximity. This demonstrated that, overall, precursor-related features in interval 36 were detected more systematically with earthquake proximity. All 31 earthquakes were investigated as a single dataset, therefore, this result does not indicate that precursor-related features became more frequent with proximity to each individual event. Interestingly, the majority of the windows in other intervals were classified as precursors, indicating that the precursor-related features detected in interval 36 were translatable to other intervals over the 10-hours investigated and were not specific to interval 36.

Another important observation was that 46% of the windows from interval 1 were classed as a precursor. This indicated that seismic data in interval 1 was not entirely representative of data in the noise class, otherwise, 0% of the windows would have been classed as a precursor (100% classed as noise). Comparing the results from Figure 4.28 and Figure 4.25, it can be concluded that the precursor-related features detected in interval 36 still existed 10 hours prior to some earthquakes.

4.11 Probabilistic Considerations

The algorithm used to detect these precursor-related features lacks robustness in the context of probabilistic forecasting. This lack of robustness can be attributed to:

1. The data selected in Section 4.3 to train and test the neural network. Several restrictions were applied in Section 4.3 to the data investigated for precursors. The restrictions were applied to reduce the variability of the data and encourage analysis of the

background signal, increasing the potential for the network to detect systematic precursors. These restrictions increased the likelihood of obtaining a false prediction in the context of earthquake forecasting as precursors were detected in the absence of impulsive earthquake signal and interference from other large earthquakes.

2. In Section 4.3, 31 Mw $\geq$ 6 earthquakes were selected over a continuous 8-year timeframe from 2012-2020. The total investigated time period was around 8 years, however, only a fraction of the 8 years was analysed. Investigating a very small time period increased the difficulty in quantifying the significance of the precursor-related features detected in Section 4.5 and Section 4.10.1 for earthquake forecasting. For example, it is poorly understood how frequently the precursor-related features occur unrelated to a Mw $\geq$ 6 earthquake. The risk of obtaining a false positive prediction is difficult to evaluate with the current dataset. This is mainly because the noise dataset in Section 4.10.1 spans a very short and selective time period and is therefore poorly representative of all data unrelated to Mw $\geq$ 6 earthquakes. As a result, the confusion matrix in Figure 4.27b does not provide an accurate indication of the likelihood of obtaining a false prediction.

Whilst the confusion matrix in Figure 4.27b can indicate the performance of the network, it cannot be used to determine whether there is a relationship between the detected precursor-related features and the investigated earthquakes. To evaluate the statistical significance of the discriminative features detected in the precursor class in Section 4.10.1, the null hypothesis was evaluated. In this context, the null hypothesis (H_0) states that the features detected in interval 36 are unrelated to the Mw $\geq$ 6 earthquakes investigated. The other interpretation is the alternative hypothesis (H_1) which states that there is a relationship between the detected features and the investigated earthquakes. When testing the null hypothesis, a probability called the p-value is calculated which determines the likelihood that the relationship exists if H_0 were true. In null hypothesis testing, if the p-value is less than a criterion (α), the result is considered statistically significant and H_0 is rejected in favour of H_1. Generally, a p-value < 0.05 is statistically significant (Ghasemi & Zahediasl 2012).

The best weights obtained in Section 4.10.1 classified randomly selected 16.7-minute

periods of noise unrelated to Mw $\geq$ 6 earthquakes from data in the 16.7-minute interval immediately prior to the start of the investigated Mw $\geq$ 6 earthquakes. When applying these weights in the context of probabilistic forecasting, non-overlapping windows with length 16384 time samples would be input to the neural network in real time and a prediction for each window would be obtained. Instead of forecasting an earthquake based on the single occurrence of a window predicted as a precursor, it may be more valuable to consider the certainty of the network in its prediction and/or the frequency of windows classified as a precursor over a predefined time interval. This may reduce the possibility of obtaining a false-positive prediction.

If we consider a scenario in which a threshold is used to determine whether or not an earthquake will occur based on observations from 5 hours of data, setting the threshold at 70% (i.e. at least 70% of non-overlapping windows over a period of 5 hours has to be classified as a precursor to forecast an earthquake), 22 out of the 31 investigated earthquakes would have been forecast when analysing the 5 hours of data prior to the start of interval 36. As a result, 9 earthquakes did not have more than or equal to 70% of windows classified as a precursor over the 5 hour period selected and were therefore not forecast. This threshold was arbitrarily chosen as an example.

For this scenario, let:

N_p = Number of earthquakes correctly forecast = 22

T_n = Total time period classified as noise = Length of selected period (5 hours) x number of earthquakes not forecast (9) = 45 hours

T_p = Total time period classified as a precursor = Length of selected period (5 hours) x number of earthquakes correctly forecast (22) = 110 hours

The probability of 22 events being forecast by chance equates to the probability that 22 'positive' forecasts all fell randomly within the time periods belonging to T_p, but only once in a given 5 hour time span. Such a probability can be computed as:

$$P = \prod_{i=1}^{N_p} \frac{T_p - 5(i-1)}{T_p + T_n} = 4e^{-11} \tag{4.2}$$

As P $<$ 0.05, the null hypothesis can be rejected and the relationship between the de-

tected precursory features and the earthquakes investigated can be concluded as statistically significant. This indicated that the precursory features detected by the network in interval 36 were correlated to earthquake occurrence. Given that large time intervals between earthquakes have not been analysed, it is difficult to evaluate whether similar signals could occur during aseismic intervals and the frequency of these false positive detections. This prevents an accurate estimate of the likelihood of obtaining a false positive prediction.

As the precursory features were identified as statistically significant, the term 'precursor' will be used from here on as a loose term to describe features correlated with the investigated earthquakes. It should be stressed that this result does not indicate that the precursor-related features are correlated to all Mw $\geq$ 6 earthquakes in this region.

4.12 Investigating Precursors in Continuous Seismic Data

Having determined where the detected precursors were most systematically observed prior to the investigated earthquakes, the neural network was trained and tested on continuous seismic data from 2015 to 2020 to see whether the network was still able to detect the precursors when no constraints were applied to the data used to train and test the network. Seismic data were obtained from the seismometer of interest at station IU MAJO and only 5 years of continuous data was investigated due to network related limitations (Section 5.4.2).

Non-overlapping noise windows were generated throughout the 5 years of continuous seismic data investigated. This excluded a 72 hour period prior to Mw $\geq$ 6 earthquakes occurring within 30° from the seismometer of interest. Figure 4.28 suggested that precursors still occurred up to 10 hours prior to the investigated earthquakes, therefore, preventing the 72 hours of seismic data prior to each Mw $\geq$ 6 earthquake from being investigated as noise reduced the likelihood of precursors being included in noise windows. Precursor windows were generated over the 5-hour period prior to each Mw $\geq$ 6 earthquake located within a radius of 20° from the station. 5 hours prior to each earthquake was selected as this period was found to contain precursors most systematically (Figure 4.25 and Figure 4.28).

Due to the substantial amount of noise windows relative to precursor windows, there

was a huge class imbalance. An issue with such a large class imbalance was that there were too few examples of the minority (precursor) class for the model to effectively learn the decision boundary. To overcome this issue, firstly, the noise data were randomly under-sampled (Prusa et al. 2015). 12000 noise windows with no overlap were randomly selected over the 5-year period investigated and sorted in order of their occurrence. Data labelled as a precursor was downloaded separately for each Mw $\geq$ 6 earthquake which were also sorted in order of their occurrence.

In addition to randomly under-sampling data in the noise class, a synthetic minority oversampling technique or SMOTE was implemented to increase the number of precursor windows in the training data (Arslan et al. 2019). Different to duplicating examples from the minority class in the training dataset, which would not provide any additional information to the model, SMOTE synthesised new examples from the precursor (minority) class, augmenting the training data. SMOTE augmented the training data by selecting examples that were close in feature space to the existing precursor windows. In more detail, a random window from the precursor class was chosen, k of the nearest neighbours (nearest precursor windows in feature space) were identified where k was typically 5, one of these nearest neighbours was randomly selected and a synthetic example was created at a random point between the two examples in feature space (Inoue 2018). New examples of precursor windows were generated from existing windows using the SMOTE technique. By implementing SMOTE and random undersampling, the same number of precursor windows were generated as noise windows in the training dataset.

The network was trained on the first 75% of noise windows which were sorted in order of occurrence and precursor windows from the first 48 out of the 60 Mw $\geq$ 6 earthquakes that occurred over the time period investigated and within 20° of the seismometer of interest. The test dataset contained the following 15% of noise windows and precursors from the next 9 earthquakes. The validation dataset contained the final 10% of the noise windows and precursors from the last 6 earthquakes. As a result of the train-test-validation split, each dataset approximately investigated a different time period. A window length of 8192 was selected due to frequent memory issues when using the original window length (16384 samples),

even after significantly reducing the batch size. The network was trained on 19200 windows, tested on 6000 windows and validated on 3600 windows and the ratio of noise to precursors in the training data was 0.5, a result of applying SMOTE to the training data. SMOTE was not applied to the test and validation datasets, however, the ratio of noise windows to precursor windows was 0.5 in the test and validation data as a result of using an overlap of 1024 samples (window length/8) when generating test and validation precursor windows.

The network produced an average (over 5 runs) accuracy on the training data of 93.4%, test accuracy of 72.7% and validation accuracy of 52.9%. The training, testing and validation losses were 0.56, 0.65 and 0.78, respectively. The high test and validation loss indicated that the network was not extremely confident in its predictions and/or that it confidently predicted windows as the incorrect class. A low validation accuracy demonstrated that the network was unable to effectively translate features learnt during training to the validation dataset. In a probabilistic sense, 72.7% test accuracy is quite significant. It indicates that the network was able to generalise features learnt during training to the test data. Improving the validation accuracy is key towards developing a forecasting system. Techniques investigated to improve generalisation are described below (see the following subsection).

Since the network had not previously experienced issues with overfitting, a low validation accuracy could be explained by the fact that the train and validation data investigated a different time-period and therefore the training data were not entirely representative of the validation data. Randomising the windows assigned to the train, test and validation datasets would reduce the possibility of the training data not being representative of the test and validation however, would prevent an understanding of how well features learnt from one time period can be translated to data from a different time period.

Alternatively, issues with generalisation could have resulted from precursors frequently being overprinted by impulsive earthquake signal. As a result, the network was unable to learn the same features that had been detected in Section 4.5 where precursors were investigated in the absence of significant impulsive earthquake signal. A method for removing impulsive earthquake signal could be implemented to investigate the importance of this factor for generalisation.

Another potential reason for poor performance on the validation data was the greater quantity and variety of noise data analysed in this investigation compared with previously in Section 4.10.1. This may have resulted in the precursory signal detected in Section 4.10.1 occurring more frequently in the data labelled as noise, preventing the same precursors from contributing to the classification result.

Finally, halving the window length reduced the amount of information stored in each window. When reducing the window length in Section 4.5 to 8192 time samples, the test and train accuracies only decreased by a few percent. As a result, this factor is unlikely to have had a significant impact on the validation accuracy, particularly if the network was detecting the same features as previously (Section 4.5).

4.12.1 Other methods investigated to improve the performance of the network

1. Applied transfer learning, a method which involved storing information gained from one task and applying the stored information to a different but related task. The best weights (96% test accuracy), obtained in Section 4.10.1, were used as the initial weights for training the network on the current task. This was as opposed to using randomly initialised weights prior to training. This method was implemented to encourage the network to consider the previously identified precursors in Section 4.10.1 during training.
2. Experimented with different standardisation and normalisation. Current standardisation involved standardising all 3 channels together instead of separately. Standardising each channel on its own did not significantly change the result. Additionally, the data were normalised between different ranges (-1 and 1, 0 and 1).
3. A low validation accuracy could have resulted from generalisation issues on the new data. To try to improve the generalisation of the network, data were investigated from two additional GSN stations. The idea was to focus the network towards identifying discriminative, underlying patterns or features in the data. Noise and precursors were investigated from 3 separate, neighbouring seismic stations (IU MAJO, II ERM and

IU YSS). The same seismic data recorded by 3 different instruments at 3 nearby but separate locations should appear slightly different but potentially contain the same underlying patterns. The aim was to remove the influence of any bias in the network's decision and encourage the network to learn the precursor-related features that had previously been identified. This method was thought to potentially improve generalisation by introducing variability in the features within each class and encouraging the network to learn general characteristics.

4. Experimented with different hyperparameter values and additionally tried using genetic algorithms for optimisation. The purpose of the genetic algorithm was to optimise a set of hyperparameters (the parameters that could not be learned by the network during training). Genetic algorithms are global search methods, based on principles such as natural selection, mutation and crossover (Harik et al. 1999). The genetic algorithm worked by firstly creating a population of randomly generated values. The algorithm scored each value based on some goal and selected and bred the best values in the population, mutating some values randomly to attempt to discover better values. The values which scored poorly were discarded during the process (Harik et al. 1999). Several parameters were tuned using this method including the number of neurons per layer, the activation function, the dilation rate, the network optimiser, the learning rate, the kernel initializer and the kernel size.

None of the applied methods improved the network's performance on the train, test or validation data. The training accuracy did not improve above 95% with any of the methods and techniques tried. This indicated that, unlike on the dataset in Section 4.10.1 where the algorithm reached a training accuracy of 98%, the network was unable to learn some of the training data. Considering that the same network was used, this may indicate that some precursor windows contained features indicative of noise and vice versa. This could have resulted from some of the precursor windows being overprinted by impulsive earthquake signal or some of the noise windows containing the previously detected precursors.

As we were unable to improve the test and validation accuracy, a system for autonomous earthquake prediction on live data was not implemented. A 52.9% validation accuracy indi-

cated a significant number of false predictions. As a result, this system would not provide a reliable method for forecasting unseen earthquakes.

Chapter 5

Discussion

In this Chapter, the results are summarised and further examined. Several limitations of the methodology are detailed, and possible solutions provided. Finally, future work is described.

This thesis investigates the research question: Can precursors be detected in the raw signal prior to lab and crustal earthquakes using deep neural networks and to what extent can they be used for earthquake forecasting?

5.1 Summary of Results

In Chapter 3, a lab experiment was carried out using a triaxial press machine. A sample of granite with a pre-cut fault was loaded until stick-slip events started to occur. Four strain gauges were positioned across the fault and the strain gauge signal, corresponding to a deformation, was recorded at a sampling rate of 10 MHz. Each of the strain gauge recordings had a different SNR. Two of these recordings were investigated: one with a very high SNR and the other with a very low SNR. As a preliminary experiment, the time samples in each dataset were labelled either as 'earthquake' or as 'noise' and windows of the data were generated as inputs to the semantic segmentation network. In both the high and low SNR strain gauge recordings, the network detected the slip events in the test and validation datasets with a high accuracy and no false positive predictions occurred (Section 3.7 and 3.8).

In Section 3.10 an additional 'precursor' class was included when labelling the low SNR strain gauge dataset and a moving window approach utilising semantic segmentation enabled

precursor-related features in the strain gauge signal to be inferred prior to the main slip event in the validation data. The algorithm successfully detected the slip event before it was in the frame of view of the network and the inferred precursors were detected up to 6.6 ms prior to the start of the lab induced earthquake. The significance of this experiment suffered due to the lack of available data and, as a result, no conclusion could be made regarding whether this observation was systematic.

In Section 4.5, short-term changes were detected in the raw, background seismic signal prior to several Mw $\geq$ 6 earthquakes in the Japan region. Achieving a test accuracy of 89%, the application of deep neural networks enabled the detection of discriminative features separating noise-labelled data (10 hours prior to the investigated earthquakes) from precursor-labelled data (immediately prior to the investigated earthquakes). In section 4.7, the discriminative features were found to occur dominantly over a frequency range from 0.1 to 0.9 Hz, corresponding to the frequency range of microseismic noise (Masuda et al. 2020). Precursor-related features (features specific to the precursor-labelled data) were identified prior to a single event in the test data (channel 0 only) in Section 4.8 at frequencies of $\sim$ 0.16 and $\sim$ 0.20 Hz (Section 4.9). The frequencies of the features identified prior to a single event coincided well with the 2 frequencies that were significantly higher amplitude in the precursor class compared to the noise class for the events in the train and test datasets (Figure 4.23). This suggested that the precursor-related features identified in Section 4.9 contributed significantly to the decision process of the network.

In Section 4.10.1, the algorithm achieved a 96% test accuracy classifying randomly selected noise data (seismic data unassociated with Mw $\geq$ 6 earthquakes) and data immediately prior to the Mw $\geq$ 6 earthquakes investigated. Precursor-related features (features discriminative of noise unrelated to Mw $\geq$ 6 earthquakes) became increasingly significant with earthquake proximity and were found to be correlated with the investigated earthquakes (Section 4.11). The network was not extremely robust when detecting precursors in the form of a forecasting system where features learnt from one time-period were translated to different time periods in the testing and validation (Section 4.12). When developing the forecasting system, features were learnt in the presence of impulsive earthquake signal. This factor may

have affected the ability of the network to learn the same features that had previously been identified in Section 4.6 as the network was originally designed to detect precursors in the absence of impulsive earthquake signal.

5.2 Comparison with Related Work

Precursors in lab data have previously been identified by investigating changes in elastic properties (elastic wave speed and elastic wave amplitude) and acoustic emissions (AEs) related to slip rate. By tracking acoustic activity prior to stick-slip instabilities, an exponential acceleration of precursory slip was systematically observed (Johnson et al. 2013), (Passelègue et al. 2017), (Rouet-Leduc et al. 2017). Precursor AEs, analogous to seismic events in the earth, were found to begin late in the stick-slip cycle and were frequently associated with microshear failures (precursors associated with grain rearrangements within the shearing layer). Here, the accumulation of microscopic rearrangements in gouge material led to creep, with the frequency of rearrangements increasing dramatically as the main slip event approached (Johnson et al. 2013). Preceding the slip there was a rapid acceleration of AE and microslips (Johnson et al. 2013).

A systematic decrease in the elastic wave amplitude (Shreedharan et al. 2020) and elastic wave speed (Scuderi et al. 2016) has also been detected prior to fault failure, providing a clear precursor to lab earthquakes. The Mw of these observations is related to the amount of slip that occurred prior to each dynamic event (Passelègue et al. 2017). In all lab experiments, precursory changes were found to evolve continuously until the start of the lab earthquake. As a result of these observations, one might expect precursor-related patterns identified in lab and seismic data to increase in amplitude and/or frequency with earthquake proximity.

In contrast to previous lab observations (Johnson et al. 2013), (Passelègue et al. 2017), (Rouet-Leduc et al. 2017), the saliency map in Figure 3.16 and Figure 3.17 suggested that precursors detected in the strain gauge lab data did not continuously increase in significance with proximity to the slip event. Instead, precursors additionally occurred over a localised period prior to the main slip event. This could suggest that the strain gauge signal contained

precursor-related information different to that previously identified to occur systematically, prior to lab earthquakes. The discrepancy between precursors identified in the strain gauge signal and those observed in lab data could be a result of the type of data investigated or the methods used to obtain the data. Different to piezometric and seismic data, strain gauge data preferentially reveals lower frequencies. This is because strain in the vicinity of a rupture has a larger ratio of static to dynamic components than seismic motion.

Earthquake predictability in seismic data indicated a gradual increase in the frequency of windows classified as a precursor across the train and test datasets with earthquake proximity (Section 4.10.2). Earthquakes were investigated as a single dataset, therefore, these results do not indicate that precursor frequency increased with earthquake proximity prior to each event. It can be concluded, however, that precursors became increasingly systematic with earthquake proximity for the earthquakes investigated.

Although some similarities in earthquake predictability may be evident in lab and crustal settings, this does not indicate a common precursor origin. For example, the microshear failures observed to increase exponentially with earthquake proximity in (Johnson et al. 2013) would correspond to very small (Mw $<$ 3) earthquakes in seismic data. These tiny earthquakes would have a corner frequency greater than 10 Hz, well above the diagnostic frequency range identified in Section 4.7 and Section 4.9 (Sibson 1989). The precursors detected in Section 4.5 and Section 4.10.1 could be attributed to tremor emitted by low or very-low-frequency earthquakes that are observed in the $>$1 Hz and 0.01–0.10 Hz frequency band respectively. These phenomena are separated by large microseismic noise at 0.1–1.0 Hz (Masuda et al. 2020) which coincides with the dominant frequency range of the detected precursors (Figure 4.15). Recent observations of the seismic signal emitted from the shallow part of the Nankai subduction zone, Japan, have suggested that the signal in the microseismic noise frequency band is accompanied by low and very-low-frequency earthquakes (Masuda et al. 2020).

5.3 Significance of Key Results

These results are the first to indicate statistically significant, short-term precursors in the raw seismic signal prior to several large (Mw $\geq$ 6) earthquakes. In Section 4.5, a high train, test and validation accuracy and a low loss demonstrated that the discriminative features learnt during training were similarly detected prior to the earthquakes in the test and validation datasets. When training and testing a neural network, a high train, test and validation accuracy are only achieved if the network has learnt general, yet discriminative features that are consistently evident in all 3 (train, test and validation) datasets. As a result, it can be concluded that the short-term precursors were systematic for the earthquakes investigated. In addition to these features being detected systematically prior to unseen earthquakes recorded by the seismometer of interest, the discriminative features were also detected in the seismic signal from a different seismometer and seismic station. Testing the learned weights on seismic data prior to a randomly selected Mw $\geq$ 6 earthquake recorded by a different seismometer to the original seismometer of interest, the network achieved an R-score of 0.95. This indicated that the frequency response of the seismometer and surrounding noise did not influence the detection of precursors and that the network had not learnt features of the seismic signal specific to the seismometer of interest. These results suggest that the seismic data captured some signature of the fundamental physics of the earthquake preparatory phase for Mw $\geq$ 6 earthquakes in the Japan region. A downside of feature generalisation to different seismometers is that there is a greater level of difficulty in constraining the geographic region of an impending earthquake in the case of earthquake forecasting.

When visualising the detected precursors in Section 4.6, it became evident that frequencies of $\sim$ 0.16 and $\sim$ 0.2 Hz provided precursor-related information. Assuming that the detected precursors were dominantly associated with some low frequency tremor, there is difficulty in understanding the reliability of the precursors when applied to earthquake forecasting. An inadequate quantity of data were analysed due to memory-related limitations, therefore, there is little understanding of how frequently the detected tremor would lead to a false earthquake prediction when applying the best weights (96% test accuracy) to a

continuous stream of seismic data. Ultimately, the complexity associated with earthquake forecasting relies in robustly detecting precursors in seismic data. This can only be achieved if there is enough computational power to process large quantities of data.

5.4 Methodological Limitations

The results were dependent on several limitations and assumptions. These are explored, and justification of these assumptions are detailed.

5.4.1 Data Related Limitations

Quantity of data

Due to the occurrence of the Mw 9 Tohoku Earthquake in 2011 and the frequent, large Mw aftershocks, earthquakes were only investigated from March 2012 onwards (Hirose et al. 2011) (Section 4.3). This prevented investigation of the highly recurrent aftershocks. Earthquakes prior to the Tohoku earthquake were not investigated as precursor characteristics could have been quite different (a result of significant fault stress release associated with the Tohoku earthquake). Therefore, precursors prior to the Mw 9 earthquake may not have been representative of precursors occurring after. This would have had implications on the performance of the network.

In addition to constraining the selected earthquakes to those occurring within a time interval from 2012 to 2020, only earthquakes within 20° from the station of interest were investigated. Additionally, earthquakes were only selected if they occurred at least 48 hours after another Mw $\geq$ 6 earthquake occurring within a 30° radius from the seismometer of interest. This condition was also applied to the noise intervals selected in Section 4.10.1. It was assumed that 48 hours was long enough to significantly reduce the influence of afterslip from other large Mw $\geq$ 6 earthquakes. There is no certainty that this condition was upheld but it is likely that any influence from other large earthquakes was minimal for the investigated earthquakes and noise intervals.

Final constraints were applied to remove data which was poorly recorded and contained

impulsive earthquake signal above the noise level. These events were removed both to prevent bias in the classification and to encourage the network to learn features of the background seismic signal, unrelated to impulsive earthquakes. This reduced the number of earthquakes investigated but prevented investigation of unreliable precursors such as foreshocks. The generalisability of the results is limited by the constraints that were applied to enable systematic precursor identification. Accurate detection and subsequent removal of impulsive (Mw $<$ 6) earthquakes could provide an effective method for investigating a larger quantity of Mw $\geq$ 6 earthquake data, similar to that selected in Section 4.3.

The reliability of the detected precursors at predicting Mw $\geq$ 6 earthquakes was reduced by the lack of data investigated in the noise and precursor class in Section 4.5 and Section 4.10.1. Improving the reliability is key for developing a forecasting system. An earthquake prediction must define 3 elements: 1) the date and time, 2) the location, and 3) the Mw. Currently, due to the small sample size, none of these 3 elements can be reliably predicted from the precursors identified. This raises the question of a) can these precursors still be detected in the presence of impulsive earthquake signal, b) are they systematic to all future Mw $\geq$ 6 earthquakes, c) does the timing at which they occur prior the earthquakes vary significantly, d) how far from the station of interest can they still be detected and e) are the precursors specific to the Mw range investigated?

Dataset labelling

During the initial investigation of precursors in Section 4.5, 10 hours of seismic data were selected prior to each earthquake and the start of each file (first 16.7 minutes of data or 40000 samples) was labelled as noise and the end of each file (last 16.7 minutes of data or 40000 samples) as precursors. Although 10 hours was simply an estimate of where data in the noise class occurred, investigating a short time-period enabled short-term changes to be identified. Additionally, due to the high recurrence of Mw $\geq$ 6 earthquakes in the Japan region, there was difficulty in investigating long, uninterrupted time periods prior to the earthquakes. No scientific justification could be made as to where precursors occurred, therefore, it was assumed that they would be most prominent immediately before each earthquake, as is often

the case when investigating precursors in lab data (Johnson et al. 2013, Passelègue et al. 2017, Rouet-Leduc et al. 2017).

5.4.2 Network Related Limitations

An issue when training the neural network was the amount of data that could be used as input to the network during a training epoch. A neural network stores both its data and parameters. As a result, the depth and complexity of the network (associated with the network parameters) restricted the amount of data that could be input before reaching a memory error. This prevented the network from being trained on large lengths of data. For example, this issue would have significantly increased difficulty when investigating longer term (days-weeks) changes in the seismic signal related to precursors. When training a neural network for this task, it was evident that a high degree of complexity was key for detecting precursors (the performance of the network on the dataset increased with complexity). Therefore, there was a trade-off between the network complexity and the amount of input data. Other parameters which contributed to the computational cost were the batch size and window length. When increasing the quantity of the input data, the batch size and/or the window length were subsequently reduced.

Investigating different window lengths, it was evident that a smaller window length reduced the accuracy of the network on the train and test datasets and restricted the network's ability to learn features of the data related to precursors. By halving the window length from 16384 to 8192, the average train accuracy in Section 4.5 reduced from 99.5% to 97.0% and the average test accuracy from 87.5% to 83.6%. Using a much smaller window length of 64 time samples, the average train accuracy decreased to 55.2% and the average test accuracy to 50.4%.

Due to the significance of the window length on the performance of the neural network, the batch size was frequently reduced to compensate for a larger window length. Batch size was an important hyperparameter to tune as it affected generalisation and convergence of the network. Too large of a batch size led to poor generalisation, however, this could be somewhat controlled by increasing the learning rate accordingly. Smaller batch sizes resulted

in faster convergence; however, the downside of a smaller batch size was that the model was not guaranteed to converge to the global optima (Radiuk 2017). The network generally performed well when using smaller batch sizes (8,16,32), therefore, initially reducing the batch size to reduce the computational cost provided the best solution.

A trade off existed between the amount of input data, the complexity of the network, the window length, and the batch size. This trade off restricted the amount of data that could be input to the network whilst fixing the optimum parameters. The simplest solution would be using a GPU with larger memory. Alternatively, online learning could be implemented. DNNs are typically trained by backpropagation in a batch setting, requiring the entire training data to be made available prior to the learning task. Online learning represents a class of algorithms that learn to optimise predictive models from a sequence of data provided over time (Choy et al. 2006). In online learning, models update continuously as each data point arrives. The model is updated using only the newest data points and, therefore, the system does not need to store a large amount of data in memory. Compared with batch learning, systems using online learning can maintain a much smaller amount of data storage. Additionally, online learning may aid in developing a forecasting system as it adapts to better changes in the data and can gradually discount the importance of past data (Perozzi et al. 2014). This would overcome any issues associated with the training dataset not being entirely representative of the test and validation datasets and could subsequently improve performance of the model on future data.

5.5 Future Work

This work indicates the existence of short term precursors imprinted in the raw seismic signal prior to several Mw $\geq$ 6 earthquakes in the Japan region. The practical implication is to develop a forecasting system that can detect precursors to Mw $\geq$ 6 earthquakes in real time. As the implementation in Section 4.12 did not achieve good results on the validation data, it would be more useful to further the investigation of the precursors identified in Section 4.9 and the result obtained in Section 4.10.1.

The difficulty in validating the best weights (96% test accuracy) on a constant stream of seismic data is the lack of robustness. Earthquakes were investigated over an 8-year period, however, due to memory-related limitations only a small fraction of the 8 years was selected to train and test the neural network. As a result, it is unknown how often the detected precursors would occur unrelated to a Mw $\geq$ 6 earthquake. The probability of obtaining a false positive prediction is difficult to quantify using this dataset, limiting the suitability of the algorithm for real-world operational earthquake forecasting. For example, if the detected precursors were not highly specific to the Mw range of the earthquakes investigated, a Mw $<$ 6 earthquake may be predicted as a Mw $\geq$ 6 earthquake. This could result in unnecessary and costly evacuation or other damage mitigation action.

To enable a better understanding of the reliability of this algorithm, a third class could be included to investigate the seismic signal prior to smaller (Mw $<$ 6) earthquakes. This may provide an indication of whether the detected precursors are specific to a Mw range. Additionally, there is the issue of generalisation as the precursors were detected in the absence of impulsive seismic signal. Developing a method for accurately detecting and removing impulsive earthquake signal could improve generalisation and would enable the investigation of a larger selection of Mw $\geq$ 6 earthquakes. Additionally, as precursors were still detected 10 hours prior to some earthquakes (Figure 4.28), it may be useful to investigate a longer time period prior to each earthquake.

Overall, to improve the robustness of the algorithm, a much greater quantity of data is required to train, test and validate the neural network. This was not possible with the current GPU which did not provide enough computational power for this task. Alternative to increasing the memory size, the network could be trained and tested on a larger quantity of data through the application of online learning, previously discussed in Section 5.4.2. It would be key to understand more precisely the likelihood of obtaining false predictions such that the practical significance of the algorithm could be evaluated.

Chapter 6

Conclusion

The purpose of this research project was to investigate short term precursors to lab and crustal earthquakes in raw time series data. By harnessing the success of deep neural networks for pattern recognition, this study detected precursor-related features in both lab and real seismic data.

In the preliminary experiment (Section 3.8), the semantic segmentation algorithm detected the slip event in the validation data (most noise-obscured event in the whole strain gauge dataset) and did not produce any false positive detections. This indicated the robustness of the network to noise and potential aliasing from decimation and demonstrated that the segmentation algorithm was able to efficiently and accurately process the lab data for earthquake detection.

In Section 3.10, precursor-related features in lab data were detected up to 6.6 ms (514 time steps) prior to the validation earthquake. The precursor-related features were identified through a saliency and occlusion experiment which indicated that the network placed importance on a short section of the data prior to the slip event in the validation dataset (Section 3.13 and 3.14). The localised nature of the precursory features identified prior to the main slip event (Figure 3.16) may suggest that the precursors originated from impulsive and tremor-like signals, similar to those identified in (Rouet-Leduc et al. 2017). The segmentation network was only validated on a single lab earthquake due to the lack of slip events in the strain gauge dataset and, as a result, there is little understanding whether the detected

precursors are systematic. The significance of these results in the context of lab earthquake forecasting is uncertain and requires validation on a larger dataset (more slip events).

In the real earthquake setting (Chapter 4), the precursor-related features (features correlated to Mw $\geq$ 6 earthquake occurrence) became increasingly systematic across the train and test datasets with earthquake proximity. Visualising the seismic data, 2 low frequencies ($\sim$ 0.16 and $\sim$ 0.21 Hz) were found to contain significant precursor-related information. The results suggest that the underlying seismic signal in this geographic region is imprinted with information regarding the physical state of the Japan subduction zone. Interestingly, Mw $\geq$ 6 earthquakes were investigated over 2 neighbouring but separate subduction zones (Figure 4.4). This may suggest that the detected precursors are translatable to other subduction zones settings. It should be stressed that this study analysed seismic data in the vicinity to a major subduction plate boundary. As previously shown (Bouchon et al. 2016), interplate faulting reveals substantially higher precursory activity than intraplate faulting. As a result, these precursors may be poorly transferable to seismic activity away from plate boundaries.

The empirical findings in Chapter 4 provide a new insight into real earthquake precursors. In addition to detecting and identifying some precursor-related features, this research raises the question of whether these precursors could be applied to earthquake forecasting and the reliability of this implementation. The insights gained from this study may be of assistance in understanding where short-term precursors are most systematically observed in the seismic signal prior to Mw $\geq$ 6 earthquakes in the Japan region.

The scope of this study is limited by the inadequate quantity of data used to train and test the neural networks. In Chapter 4, the quantity of data was restricted by the GPU memory which increased difficulty in investigating large lengths of data with the current resources. Additionally, the investigation of earthquake precursors in the absence of impulsive earthquake signal limits the generalisability of the network. In spite of its limitations, the study certainly contributes to an understanding of short-term precursors to Mw $\geq$ 6 earthquakes in the Japan region.

To better understand the implications of these results, future studies could further investigate the origin of the precursors identified as well as their reliability for earthquake

forecasting. Future investigation of precursors within this geographic region is greatly encouraged, particularly through the use of deep neural networks which can provide a complex understanding of nonlinear dependencies in time series data.

Appendix A

Earthquake Information

Table A.1: Earthquakes in the training dataset sorted in order of occurrence

Time	Latitude (°)	Longitude (°)	Depth (*km*)	Catalog	MagType	Magnitude	Event Location	Station	Epicentral distance from station (°)
2019-06-18T13:22:19	38.6370	139.4804	12.0	NEIC PDE	Mww	6.4	NEAR WEST COAST OF HONSHU, JAPAN	MAJO	2.32
2019-04-11T08:18:21	40.4096	143.2985	18.0	NEIC PDE	Mww	6.0	OFF EAST COAST OF HONSHU	MAJO	5.55
2019-01-08T12:39:31	30.5926	131.0371	35.0	NEIC PDE	Mww	6.3	KYUSHU	MAJO	8.43
2018-09-05T18:07:59	42.6861	141.9294	35.0	NEIC PDE	Mww	6.6	HOKKAIDO	MAJO	6.78
2018-01-24T10:51:19	41.1034	142.4323	31.0	NEIC PDE	Mww	6.3	HOKKAIDO	MAJO	5.62
2017-11-09T07:42:11	32.5208	141.4380	12.0	NEIC PDE	Mww	6.0	SOUTHEAST OF HONSHU	MAJO	4.83
2017-10-06T07:59:32	37.5033	144.0201	9.0	NEIC PDE	Mww	6.2	OFF EAST COAST OF HONSHU	MAJO	4.74
2017-09-20T16:37:16	37.9814	144.6601	11.0	NEIC PDE	Mww	6.1	OFF EAST COAST OF HONSHU	MAJO	5.33
2017-09-07T17:26:49	27.7829	139.8041	451.0	NEIC PDE	Mww	6.1	BONIN ISLANDS	MAJO	8.87
2016-04-14T12:26:35	32.7880	130.7042	9.0	NEIC PDE	Mww	6.2	KYUSHU	MAJO	7.18
2016-01-14T03:25:33	41.9723	142.7810	46.0	NEIC PDE	mww	6.7	HOKKAIDO	MAJO	6.48
2016-01-11T17:08:03	44.4761	141.0867	238.8	NEIC PDE	mww	6.2	HOKKAIDO	MAJO	6.93
2015-05-12T21:12:58	38.9005	142.0217	39.3	ISC	MW	6.8	NEAR EAST COAST OF HONSHU	MAJO	3.83
2015-04-20T01:42:58	24.0574	122.4319	28.1	ISC	MW	6.4	TAIWAN REGION	MAJO	18.43
2015-02-20T04:25:23	39.8189	143.6157	13.3	ISC	MW	6.2	OFF EAST COAST OF HONSHU	MAJO	5.37
2014-11-09T14:38:15	46.9300	140.6300	10.0	ISC	mb	7.6	PRIMOR'YE	MAJO	10.54
2014-08-10T03:43:18	41.1340	142.2790	50.6	ISC	MW	6.1	HOKKAIDO	MAJO	5.60
2014-03-13T17:06:51	33.6222	131.8077	83.4	ISC	MW	6.3	KYUSHU	MAJO	5.99
2014-03-02T20:11:22	27.4238	127.3279	118.9	ISC	MW	6.5	RYUKYU ISLANDS	MAJO	12.96
2013-04-21T03:22:16	29.9644	138.9741	431.3	ISC	MW	6.1	SOUTHEAST OF HONSHU	MAJO	6.61
2013-04-05T13:00:02	42.7359	131.0640	571.3	ISC	MW	6.3	E. RUSSIA-N.E. CHINA BORDER REG.	MAJO	8.27
2012-12-07T08:18:23	37.8201	144.1594	35.3	ISC	MW	7.2	OFF EAST COAST OF HONSHU	MAJO	4.91
2012-07-08T11:33:05	45.4209	151.3906	37.7	ISC	MW	6.0	KURIL ISLANDS	MAJO	13.31
2012-05-23T15:02:27	41.3569	142.1267	64.1	ISC	MW	6.0	HOKKAIDO	MAJO	5.70

Table A.2: Earthquakes in the test dataset sorted in order of occurrence

Time	Latitude (°)	Longitude (°)	Depth (*km*)	Catalog	MagType	Magnitude	Event Location	Station	Epicentral distance from station (°)
2018-11-14T21:21:50	55.6324	162.0008	50.2	NEIC PDE	Mww	6.1	NEAR EAST COAST OF KAMCHATKA	MA2	7.18
2017-07-26T10:32:57	26.8975	130.1836	12.0	NEIC PDE	Mww	6.0	SOUTHEAST OF RYUKYU ISLANDS	MAJO	11.81
2016-11-11T21:42:59	38.4973	141.5658	42.4	NEIC PDE	mww	6.1	NEAR EAST COAST OF HONSHU	MAJO	3.30
2016-10-21T05:07:23	35.3676	133.8148	5.7	NEIC PDE	Mww	6.2	WESTERN HONSHU	MAJO	3.74
2016-09-20T16:21:16	30.5017	142.0478	9.0	NEIC PDE	mww	6.1	SOUTHEAST OF HONSHU	MAJO	6.84
2013-12-08T17:24:54	44.4691	149.1330	34.1	ISC	MW	6.1	KURIL ISLANDS	MAJO	11.46
2013-10-25T17:10:17	37.1457	144.7540	14.7	ISC	MW	7.1	OFF EAST COAST OF HONSHU	MAJO	5.27

Bibliography

Allen, R. M. & Kanamori, H. (2003), 'The potential for earthquake early warning in southern california', *Science* **300**(5620), 786–789.

Arslan, M., Guzel, M., Demirci, M. & Ozdemir, S. (2019), Smote and gaussian noise based sensor data augmentation, *in* '2019 4th International Conference on Computer Science and Engineering (UBMK)', IEEE, pp. 1–5.

Badrinarayanan, V., Kendall, A. & Cipolla, R. (2017), 'Segnet: A deep convolutional encoder-decoder architecture for image segmentation', *IEEE transactions on pattern analysis and machine intelligence* **39**(12), 2481–2495.

Becker, T. W., Hashima, A., Freed, A. M. & Sato, H. (2018), 'Stress change before and after the 2011 m9 tohoku-oki earthquake', *Earth and Planetary Science Letters* **504**, 174–184.

Bormann, P., Klinge, K. & Wendt, S. (2009), Data analysis and seismogram interpretation, *in* 'New manual of seismological observatory practice (NMSOP)', Deutsches GeoForschungsZentrum GFZ, pp. 1–102.

Bouchon, M., Marsan, D., Durand, V., Campillo, M., Perfettini, H., Madariaga, R. & Gardonio, B. (2016), 'Potential slab deformation and plunge prior to the tohoku, iquique and maule earthquakes', *Nature Geoscience* **9**(5), 380–383.

Brace, W. & Byerlee, J. (1966), 'Stick-slip as a mechanism for earthquakes', *Science* **153**(3739), 990–992.

Brenguier, F., Campillo, M., Hadziioannou, C., Shapiro, N. M., Nadeau, R. M. & Larose, E.

(2008), 'Postseismic relaxation along the san andreas fault at parkfield from continuous seismological observations', *science* **321**(5895), 1478–1481.

Brownlee, J. (2019), *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*, Machine Learning Mastery.

Buijze, L., Guo, Y., Niemeijer, A., Ma, S. & Spiers, C. (2020), 'Nucleation of stick-slip instability within a large-scale experimental fault: Effects of stress heterogeneities due to loading and gouge layer compaction', *Journal of Geophysical Research: Solid Earth* **125**(8), e2019JB018429.

Chen, J. H., Froment, B., Liu, Q. Y. & Campillo, M. (2010), 'Distribution of seismic wave speed changes associated with the 12 may 2008 mw 7.9 wenchuan earthquake', *Geophysical Research Letters* **37**(18).

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H. (2018), Encoder-decoder with atrous separable convolution for semantic image segmentation, *in* 'Proceedings of the European conference on computer vision (ECCV)', pp. 801–818.

Choy, M. C., Srinivasan, D. & Cheu, R. L. (2006), 'Neural networks for continuous online learning and control', *IEEE Transactions on Neural Networks* **17**(6), 1511–1531.

Cui, Z., Chen, W. & Chen, Y. (2016), 'Multi-scale convolutional neural networks for time series classification', *arXiv preprint arXiv:1603.06995* .

Davis, P., Ishii, M. & Masters, G. (2005), 'An assessment of the accuracy of gsn sensor response information', *Seismological Research Letters* **76**(6), 678–683.

Donalek, C. (2011), Supervised and unsupervised learning, *in* 'Astronomy Colloquia. USA', Vol. 27.

Dublanchet, P. (2018), 'The dynamics of earthquake precursors controlled by effective friction', *Geophysical Journal International* **212**(2), 853–871.

Gers, F. A., Schmidhuber, J. & Cummins, F. (1999), 'Learning to forget: Continual prediction with lstm'.

Ghasemi, A. & Zahediasl, S. (2012), 'Normality tests for statistical analysis: a guide for non-statisticians', *International journal of endocrinology and metabolism* **10**(2), 486.

Glorot, X. & Bengio, Y. (2010), Understanding the difficulty of training deep feedforward neural networks, *in* 'Proceedings of the thirteenth international conference on artificial intelligence and statistics', JMLR Workshop and Conference Proceedings, pp. 249–256.

Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. (2016), *Deep learning*, Vol. 1, MIT press Cambridge.

Graves, A. (2012), Long short-term memory, *in* 'Supervised sequence labelling with recurrent neural networks', Springer, pp. 37–45.

Guérin-Marthe, S., Nielsen, S., Bird, R., Giani, S. & Di Toro, G. (2019), 'Earthquake nucleation size: Evidence of loading rate dependence in laboratory faults', *Journal of Geophysical Research: Solid Earth* **124**(1), 689–708.

Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J. & Seung, H. S. (2000), 'Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit', *Nature* **405**(6789), 947–951.

Hamaguchi, R., Fujita, A., Nemoto, K., Imaizumi, T. & Hikosaka, S. (2018), Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery, *in* '2018 IEEE winter conference on applications of computer vision (WACV)', IEEE, pp. 1442–1450.

Harik, G. R., Lobo, F. G. & Goldberg, D. E. (1999), 'The compact genetic algorithm', *IEEE transactions on evolutionary computation* **3**(4), 287–297.

Hatami, N., Gavet, Y. & Debayle, J. (2018), Classification of time-series images using deep convolutional neural networks, *in* 'Tenth international conference on machine vision (ICMV 2017)', Vol. 10696, International Society for Optics and Photonics, p. 106960Y.

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.

He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep residual learning for image recognition, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 770–778.

Herman, M. W. & Govers, R. (2020), 'Stress evolution during the megathrust earthquake cycle and its role in triggering extensional deformation in subduction zones', *Earth and Planetary Science Letters* **544**, 116379.

Hesamian, M. H., Jia, W., He, X. & Kennedy, P. (2019), 'Deep learning techniques for medical image segmentation: Achievements and challenges', *Journal of digital imaging* **32**(4), 582–596.

Hirose, F., Miyaoka, K., Hayashimoto, N., Yamazaki, T. & Nakamura, M. (2011), 'Outline of the 2011 off the pacific coast of tohoku earthquake (m w 9.0)—seismicity: foreshocks, mainshock, aftershocks, and induced activity—', *Earth, planets and space* **63**(7), 513–518.

Holliday, J. R., Nanjo, K. Z., Tiampo, K. F., Rundle, J. B. & Turcotte, D. L. (2005), 'Earthquake forecasting and its verification', *arXiv preprint cond-mat/0508476* .

Hu, Y., Bürgmann, R., Uchida, N., Banerjee, P. & Freymueller, J. T. (2016), 'Stress-driven relaxation of heterogeneous upper mantle and time-dependent afterslip following the 2011 tohoku earthquake', *Journal of Geophysical Research: Solid Earth* **121**(1), 385–411.

Huang, J., Wang, X., Zhao, Y., Xin, C. & Xiang, H. (2018), 'Large earthquake magnitude prediction in taiwan based on deep learning neural network', *Neural Network World* **28**(2), 149–160.

Inoue, H. (2018), 'Data augmentation by pairing samples for images classification', *arXiv preprint arXiv:1801.02929* .

Ishibashi, K. (1988), 'Two categories of earthquake precursors, physical and tectonic, and their roles in intermediate-term earthquake prediction', *pure and applied geophysics* **126**(2-4), 687–700.

Johnson, P., Ferdowsi, B., Kaproth, B., Scuderi, M., Griffa, M., Carmeliet, J., Guyer, R., Le Bas, P.-Y., Trugman, D. & Marone, C. (2013), 'Acoustic emission and microslip precursors to stick-slip failure in sheared granular material', *Geophysical Research Letters* **40**(21), 5627–5631.

Kamiyama, M., Sugito, M., Kuse, M., Schekotov, A. & Hayakawa, M. (2016), 'On the precursors to the 2011 tohoku earthquake: crustal movements and electromagnetic signatures', *Geomatics, Natural Hazards and Risk* **7**(2), 471–492.

Kaneko, Y., Nielsen, S. B. & Carpenter, B. M. (2016), 'The onset of laboratory earthquakes explained by nucleating rupture on a rate-and-state fault', *Journal of Geophysical Research: Solid Earth* **121**(8), 6071–6091.

Karim, F., Majumdar, S., Darabi, H. & Chen, S. (2017), 'Lstm fully convolutional networks for time series classification', *IEEE access* **6**, 1662–1669.

Kattamanchi, S., Tiwari, R. K. & Ramesh, D. S. (2017), 'Non-stationary etas to model earthquake occurrences affected by episodic aseismic transients', *Earth, Planets and Space* **69**(1), 157.

Latour, S., Schubnel, A., Nielsen, S., Madariaga, R. & Vinciguerra, S. (2013), 'Characterization of nucleation during laboratory earthquakes', *Geophysical Research Letters* **40**(19), 5064–5069.

Lee, K., Lee, K., Shin, J. & Lee, H. (2019), 'Network randomization: A simple technique for generalization in deep reinforcement learning', *arXiv* pp. arXiv–1910.

Liu, C.-L., Hsaio, W.-H. & Tu, Y.-C. (2018), 'Time series classification with multivariate convolutional neural network', *IEEE Transactions on Industrial Electronics* **66**(6), 4788–4797.

Lomnitz, C. (1994), *Fundamentals of earthquake prediction*, John Wiley & Sons New York.

Long, J., Shelhamer, E. & Darrell, T. (2015), Fully convolutional networks for semantic segmentation, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 3431–3440.

Marquez, E. S., Hare, J. S. & Niranjan, M. (2018), 'Deep cascade learning', *IEEE transactions on neural networks and learning systems* **29**(11), 5475–5485.

Masuda, K., Ide, S., Ohta, K. & Matsuzawa, T. (2020), 'Bridging the gap between low-frequency and very-low-frequency earthquakes', *Earth, Planets and Space* **72**, 1–9.

McGuire, J. J., Boettcher, M. S. & Jordan, T. H. (2005), 'Foreshock sequences and short-term earthquake predictability on east pacific rise transform faults', *Nature* **434**(7032), 457–461.

McLaskey, G. C. (2019), 'Earthquake initiation from laboratory observations and implications for foreshocks', *Journal of Geophysical Research: Solid Earth* **124**(12), 12882–12904.

Meier, M.-A., Ross, Z. E., Ramachandran, A., Balakrishna, A., Nair, S., Kundzicz, P., Li, Z., Andrews, J., Hauksson, E. & Yue, Y. (2019), 'Reliable real-time seismic signal/noise discrimination with machine learning', *Journal of Geophysical Research: Solid Earth* **124**(1), 788–800.

Mignan, A. & Broccardo, M. (2019), 'Neural network applications in earthquake prediction (1994-2019): Meta-analytic insight on their limitations', *arXiv preprint arXiv:1910.01178* .

Mignan, A. & Broccardo, M. (2020), 'Neural network applications in earthquake prediction (1994–2019): Meta-analytic and statistical insights on their limitations', *Seismological Research Letters* .

Mogi, K. (1981), 'Seismicity in western japan and long-term earthquake forecasting', *Earthquake Prediction: An International Review* **4**, 43–51.

Mousavi, S. M., Zhu, W., Sheng, Y. & Beroza, G. C. (2019), 'Cred: A deep residual network of convolutional and recurrent units for earthquake signal detection', *Scientific reports* **9**(1), 1–14.

Nguyen, V. Q., Yang, H.-J., Kim, K. & Oh, A.-R. (2017), Real-time earthquake detection using convolutional neural network and social data, *in* '2017 IEEE Third International Conference on Multimedia Big Data (BigMM)', IEEE, pp. 154–157.

Nielsen, S., Taddeucci, J. & Vinciguerra, S. (2010), 'Experimental observation of stick-slip instability fronts', *Geophysical Journal International* **180**(2), 697–702.

Nishenko, S. P. & Buland, R. (1987), 'A generic recurrence interval distribution for earthquake forecasting', *Bulletin of the Seismological Society of America* **77**(4), 1382–1399.

Noh, H., Hong, S. & Han, B. (2015), Learning deconvolution network for semantic segmentation, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 1520–1528.

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

Ostapchuk, A. & Morozova, K. (2020), 'on the mechanism of laboratory earthquake nucleation highlighted by acoustic emission', *Scientific Reports* **10**(1), 1–8.

Ozawa, S., Nishimura, T., Munekane, H., Suito, H., Kobayashi, T., Tobita, M. & Imakiire, T. (2012), 'Preceding, coseismic, and postseismic slips of the 2011 tohoku earthquake, japan', *Journal of Geophysical Research: Solid Earth* **117**(B7).

Passelègue, F. X., Latour, S., Schubnel, A., Nielsen, S., Bhat, H. S. & Madariaga, R. (2017), 'Influence of fault strength on precursory processes during laboratory earthquakes', *Fault Zone Dynamic Processes: Evolution of Fault Properties During Seismic Rupture* **227**, 229.

Peng, Z., Vidale, J. E. & Houston, H. (2006), 'Anomalous early aftershock decay rate of the 2004 mw6. 0 parkfield, california, earthquake', *Geophysical Research Letters* **33**(17).

Perez, L. & Wang, J. (2017), 'The effectiveness of data augmentation in image classification using deep learning', *arXiv preprint arXiv:1712.04621* .

Perol, T., Gharbi, M. & Denolle, M. (2018), 'Convolutional neural network for earthquake detection and location', *Science Advances* **4**(2), e1700578.

Perozzi, B., Al-Rfou, R. & Skiena, S. (2014), Deepwalk: Online learning of social representations, *in* 'Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining', pp. 701–710.

Prayogo, A. S., Pakpahan, S. & Sunardi, B. (2015), 'Assesment of electromagnetic and radon concentration as earthquake precursors', *METHODS* **6**, 7.

Press, W. H. & Teukolsky, S. A. (1990), 'Savitzky-golay smoothing filters', *Computers in Physics* **4**(6), 669–672.

Prusa, J., Khoshgoftaar, T. M., Dittman, D. J. & Napolitano, A. (2015), Using random undersampling to alleviate class imbalance on tweet sentiment data, *in* '2015 IEEE international conference on information reuse and integration', IEEE, pp. 197–202.

Radiuk, P. M. (2017), 'Impact of training set batch size on the performance of convolutional neural networks for diverse datasets', *Information Technology and Management Science* **20**(1), 20–24.

Renna, F., Oliveira, J. & Coimbra, M. T. (2019), 'Deep convolutional neural networks for heart sound segmentation', *IEEE journal of biomedical and health informatics* **23**(6), 2435–2445.

Rojas, O., Otero, B., Alvarado, L., Mus, S. & Tous, R. T. (2019), 'Artificial neural networks as emerging tools for earthquake detection', *Computación y Sistemas* **23**(2), 335–350.

Ronneberger, O., Fischer, P. & Brox, T. (2015), U-net: Convolutional networks for biomedical image segmentation, *in* 'International Conference on Medical image computing and computer-assisted intervention', Springer, pp. 234–241.

Ross, Z. E., Meier, M.-A. & Hauksson, E. (2018), 'P wave arrival picking and first-motion polarity determination with deep learning', *Journal of Geophysical Research: Solid Earth* **123**(6), 5120–5129.

Ross, Z. E., Meier, M.-A., Hauksson, E. & Heaton, T. H. (2018), 'Generalized seismic phase detection with deep learning', *Bulletin of the Seismological Society of America* **108**(5A), 2894–2901.

Rouet-Leduc, B., Hulbert, C. & Johnson, P. A. (2019), 'Continuous chatter of the cascadia subduction zone revealed by machine learning', *Nature Geoscience* **12**(1), 75–79.

Rouet-Leduc, B., Hulbert, C., Lubbers, N., Barros, K., Humphreys, C. J. & Johnson, P. A. (2017), 'Machine learning predicts laboratory earthquakes', *Geophysical Research Letters* **44**(18), 9276–9282.

Ruiz, S., Aden-Antoniow, F., Baez, J., Otarola, C., Potin, B., del Campo, F., Poli, P., Flores, C., Satriano, C., Leyton, F. et al. (2017), 'Nucleation phase and dynamic inversion of the mw 6.9 valparaíso 2017 earthquake in central chile', *Geophysical Research Letters* **44**(20), 10–290.

Scholz, C. H. (2002), 'Earthquakes and faulting'.

Scuderi, M., Marone, C., Tinti, E., Di Stefano, G. & Collettini, C. (2016), 'Precursory changes in seismic velocity for the spectrum of earthquake failure modes', *Nature geoscience* **9**(9), 695–700.

Shang, W., Sohn, K., Almeida, D. & Lee, H. (2016), Understanding and improving convolutional neural networks via concatenated rectified linear units, *in* 'international conference on machine learning', pp. 2217–2225.

Shreedharan, S., Bolton, D. C., Rivière, J. & Marone, C. (2020), 'Preseismic fault creep and elastic wave amplitude precursors scale with lab earthquake magnitude for the continuum of tectonic failure modes', *Geophysical Research Letters* **47**(8), e2020GL086986.

Sibson, R. H. (1989), 'Earthquake faulting as a structural process', *Journal of structural geology* **11**(1-2), 1–14.

Simonyan, K. & Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition', *arXiv preprint arXiv:1409.1556* .

SN, S. (2003), 'Introduction to artificial neural networks'.

Socquet, A., Valdes, J. P., Jara, J., Cotton, F., Walpersdorf, A., Cotte, N., Specht, S., Ortega-Culaciati, F., Carrizo, D. & Norabuena, E. (2017), 'An 8 month slow slip event triggers progressive nucleation of the 2014 chile megathrust', *Geophysical Research Letters* **44**(9), 4046–4053.

Tamaribuchi, K., Yagi, Y., Enescu, B. & Hirano, S. (2018), 'Characteristics of foreshock activity inferred from the jma earthquake catalog', *Earth, Planets and Space* **70**(1), 90.

Tape, C., Holtkamp, S., Silwal, V., Hawthorne, J., Kaneko, Y., Ampuero, J. P., Ji, C., Ruppert, N., Smith, K. & West, M. E. (2018), 'Earthquake nucleation and fault slip complexity in the lower crust of central alaska', *Nature Geoscience* **11**(7), 536–541.

Toda, S. (2019), 'Damaging aftershock hits japan after 55 years, temblor'.

Turcotte, D. L. (1991), 'Earthquake prediction', *Annual review of earth and planetary sciences* **19**(1), 263–281.

Utsu, T., Ogata, Y. et al. (1995), 'The centenary of the omori formula for a decay law of aftershock activity', *Journal of Physics of the Earth* **43**(1), 1–33.

Uyeda, S. (2013), 'On earthquake prediction in japan', *Proceedings of the Japan Academy, Series B* **89**(9), 391–400.

Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X. & Cottrell, G. (2018), Understanding convolution for semantic segmentation, *in* '2018 IEEE winter conference on applications of computer vision (WACV)', IEEE, pp. 1451–1460.

Wang, Z., Yan, W. & Oates, T. (2017), Time series classification from scratch with deep neural networks: A strong baseline, *in* '2017 International joint conference on neural networks (IJCNN)', IEEE, pp. 1578–1585.

Xia, H., Sun, W., Song, S. & Mou, X. (2020), 'Md-net: Multi-scale dilated convolution network for ct images segmentation', *Neural Processing Letters* pp. 1–13.

Yamauchi, H., Uchiyama, H., Ohtani, N. & Ohta, M. (2014), 'Unusual animal behavior preceding the 2011 earthquake off the pacific coast of tohoku, japan: A way to predict the approach of large earthquakes', *Animals* **4**(2), 131–145.

Yu, F., Koltun, V. & Funkhouser, T. (2017), Dilated residual networks, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 472–480.

Zhao, H., Shi, J., Qi, X., Wang, X. & Jia, J. (2017), Pyramid scene parsing network, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 2881–2890.

Zhao, Z. & Liu, H. (2007), Spectral feature selection for supervised and unsupervised learning, *in* 'Proceedings of the 24th international conference on Machine learning', pp. 1151–1157.

Zhao, Z.-Q., Zheng, P., Xu, S.-t. & Wu, X. (2019), 'Object detection with deep learning: A review', *IEEE transactions on neural networks and learning systems* **30**(11), 3212–3232.

Zheng, Y., Liu, Q., Chen, E., Ge, Y. & Zhao, J. L. (2014), Time series classification using multi-channels deep convolutional neural networks, *in* 'International Conference on Web-Age Information Management', Springer, pp. 298–310.

Zhu, L., Peng, Z., McClellan, J., Li, C., Yao, D., Li, Z. & Fang, L. (2019), 'Deep learning for seismic phase detection and picking in the aftershock zone of 2008 mw7. 9 wenchuan earthquake', *Physics of the Earth and Planetary Interiors* **293**, 106261.

Zhu, W. & Beroza, G. C. (2019), 'Phasenet: a deep-neural-network-based seismic arrival-time picking method', *Geophysical Journal International* **216**(1), 261–273.