Supervised machine learning on a network scale: application to seismic event classification and detection

Andrew Reynen

Thesis presented to the Faculty of Graduate and Postdoctoral Studies in partial fulfilment of the requirements for the degree of Master of Science in Earth Sciences

Department of Earth and Environmental Sciences
Faculty of Science
University of Ottawa

Supervisor

Dr. Pascal Audet (Department of Earth and Environmental Sciences)

Abstract

A new method using a machine learning technique is applied to event classification and detection at seismic networks. This method is applicable to a variety of network sizes and settings. The algorithm makes use of a small catalogue of known observations across the entire network. Two attributes, the polarization and frequency content, are used as input to regression. These attributes are extracted at predicted arrival times for P and S waves using only an approximate velocity model, as attributes are calculated over large time spans. This method of waveform characterization is shown to be able to distinguish between blasts and earthquakes with 99 percent accuracy using a network of 13 stations located in Southern California. The combination of machine learning with generalized waveform features is further applied to event detection in Oklahoma, United States. The event detection algorithm makes use of a pair of unique seismic phases to locate events, with a precision directly related to the sampling rate of the generalized waveform features. Over a week of data from 30 stations in Oklahoma, United States are used to automatically detect 25 times more events than the catalogue of the local geological survey, with a false detection rate of less than 2 per cent. This method provides a highly confident way of detecting and locating events. Furthermore, a large number of seismic events can be automatically detected with low false alarm, allowing for a larger automatic event catalogue with a high degree of trust.

Contents

1 INTRODUCTION 1
1.1.1 Machine learning 1
1.2.1 Motivation 2
1.2.2 Brief history on seismic event detection 3
2 METHODOLOGY 7
2.1 Event classification 8
2.1.1 Feature selection and creation 8
2.1.2 Application of machine learning 16
2.1.3 Addition of the noise class 19
2.2 Event detection 20
2.2.1 Extension of event classification method to event detection 20
2.2.2 Addition of the reversed class 21
2.2.3 Time-series probability calculation 22
2.2.4 Association algorithm 25
3 EXPERIMENTAL RESULTS 28
3.1 Event classification 28
3.1.1 Chosen data set 28
3.1.2 Station and network accuracy 29
3.2 Event detection 30
3.2.1 Chosen data set 30
3.2.2 Detection summary in Oklahoma 32
3.2.3 Comparison of detections with the OGS 34
4 DISCUSSION 36
4.1 Event classification 36
4.1.1 Effects of adding a noise classification 36
4.1.2 Effects of PCA and potential for a better classifier 38
4.1.3 Features and their potential distance dependence 40
4.2 Event detection 41
4.2.1 Regression tuning and potential use of smaller events 41
4.2.2 Weight checking and effects of adding a reversed classification 42
4.2.3 Threshold and distance-weighting selection 43
4.2.4 Catalogue comparison 44
4.3 Next steps and future research 45
5 CONCLUSIONS 46
Acknowledgments 48

References 49
Supporting Information 53
Logistic Regression: A Brief Explanation 59

List of figures

2. Methodology
Figure 1: Processing workflow
Figure 2: Feature sampling
Figure 3: Degree of polarization example
Figure 4: Averaged horizontal channel F1 and F2 travel time curves for earthquakes
Figure 5: Example event probability versus P and S arrival times
Figure 6: Feature weights used in event detection
Figure 7: Example map view of stacked event probability
3. Experimental Results
Figure 8: Map of events used for event classification
Figure 9: Map of events used and detected for event detection
Figure 10: Comparison of earthquake catalogues between the OGS and this study
Figure 11: Estimated versus given magnitudes in the OGS earthquake catalogue
4. Discussion
Figure 12: Result of adding noise observations to the training dataset
Figure 13: Class separation view in top three principal component axes
Supporting Information
Figure S1: Averaged vertical channel F1 travel time curves for earthquakes
Figure S2: Averaged vertical channel F2 travel time curves for earthquakes
Figure S3: Averaged horizontal channel F1 travel time curves for explosions
Figure S4: Averaged horizontal channel F2 travel time curves for explosions
Figure S5: Averaged vertical channel F1 travel time curves for explosions
Figure S6: Averaged vertical channel F2 travel time curves for explosions

1 INTRODUCTION

1.1.1 Machine learning

Machine learning comprises a rapidly expanding and adapting suite of mathematical algorithms to explore and harness the potential of the large datasets which have become prominent in our modern day. These methods use a subset of available data (called a training data set) to generate a mathematical model which can then be used to predict the attributes of new or existing data. In the machine learning literature, these attributes are called “labels”, which can be either quantitative (i.e., continuous variables), or qualitative (i.e., discontinuous or categorical variables). The training data can be summarized by a set of “features” that can also be continuous or categorical variables. Machine learning can be split into two major types of analysis: unsupervised and supervised. In unsupervised learning the algorithms search to find patterns (labels or trends) in the training data, or extract conjunctive features which express the data more compactly (dimension reduction). Supervised machine learning requires the training dataset to be a set of paired values, input (independent variables, or features) together with its known output (dependent variable, or label); a model is generated using this training set, which converts the inputs as close as possible to their respective outputs. If both the features and labels are continuous variables, this particular problem is called “regression” (in the classical statistical sense); if the labels are categorical, this problem is called “classification”. There can be mixed problems as well. A much more thorough description of machine learning can be found in Witten

et al. 2016. We will use supervised machine learning to classify seismic events as blasts or earthquakes, based on a set of features, and to automatically detect and locate earthquakes based on these models.

1.2.1 Motivation

The creation of seismic catalogues has long been a tedious task for their subsequent use in a wide variety of applications, such as induced seismicity studies (Ellsworth 2013; Keranen *et al.* 2013), nuclear blast monitoring (Kværna *et al.* 2007), examination of fault rupture (Cardwell & Isacks 1978) and determining Earth's internal structure (Dziewonski & Anderson 1981). Catalogues that are complete to lower magnitudes, in particular in areas of low seismicity, are important for probabilistic seismic hazard analysis as larger catalogues help constrain the magnitude recurrence relationship (Felzer 2006). For any given network of stations, catalogue completeness depends on the capacity of the network to robustly detect weak signals caused by low-magnitude events from ambient seismic noise. Methods designed to detect events typically have a set of parameters that can be optimized to detect as small events as possible. The task of the seismologist is to find some metric that indicates the confidence of event occurrence, and set the parameter values such that there are an acceptable number of false positives. This balancing act is dependent on the chosen detection method as each can result in a different number of reported events (e.g. Gibbons & Ringdal 2006; Yoon *et al.* 2015), with acceptable levels of tolerance to noise.

Over the past few decades, the amount of seismic data available has increased dramatically. In order to process considerable amounts of data in a timely manner,

approaches combining automatic detection schemes with revisions by an analyst are commonly used to confirm the occurrence of seismic events. However, due to a lack of human resources, or high seismicity, organizations have been forced to only review events within smaller regions, or magnitude ranges (NCEDC 2013; Darold *et al.* 2015). This leaves automatically detected events that are left unreviewed, for which the confidence in their actual occurrence is entirely determined by a statistic such as the root-mean-squared pick residual or number of contributing stations. To improve the confidence in the occurrence of an event, as well as to provide even more complete catalogues, we seek better automatic detection algorithms that provide a trustworthy metric at lower magnitudes.

1.2.2 Brief history on seismic event detection

A wide variety of automatic detection algorithms have been applied to seismic data sets (Sharma *et al.* 2010). Each method provides their own measure of waveform characterization, or similarity to a set of reference waveforms. Some methods produce a set of arrival times, or phase picks, based on the triggering of a characteristic function. A commonly known characteristic function, short-term/long-term averaging (Allen 1978), is fully dependent on the signal-to-noise ratio, and is forced to set trigger thresholds that fail to flag true arrivals with a low signal-to-noise ratio. Advanced characteristic functions, based on statistical measures such as kurtosis or skewness (Saragiotis *et al.* 2002; Baillard *et al.* 2014), are capable of producing phase picks at a signal-to-noise ratio close to unity (Galiana-Merino *et al.* 2008). However, these measures fail to utilize the relative amplitudes of real seismic events at different frequency bands. Relative

frequency content is accounted for in autocorrelation (Brown *et al.* 2008; Bostock *et al.* 2012) and cross-correlation (Gibbons & Ringdal 2006; Shelly *et al.* 2007) methods, as they are based on waveform similarity to a set of pre-detected and located event templates. These correlation methods will only report picks if the waveform has a shape similar to the template, allowing unique waveforms to go undetected and limiting the utility of these methods. This constraint is reduced with the application of a more recent algorithm, FAST (Yoon *et al.* 2015), which has shown the capability to match generalized features between waveforms as opposed to the specific waveform shape. Other improvements have been made to negate the effect of noise in the desired frequency range (e.g. Madureira & Ruano 2009; Lomax *et al.* 2012; Poiata *et al.* 2016), by essentially splitting the waveforms into a number of frequency bands and reducing the likelihood that noise overlaps with signal.

Machine learning techniques have also been applied for phase picking (e.g. Wang & Teng 1995; Dai & MacBeth 1997; Zhao & Takano 1999; Gentili & Michelini 2006; Kaur *et al.* 2013), event classification (e.g. Vallejos & McKinnon 2013; Mousavi *et al.* 2016) and event detection (e.g. Beyreuther *et al.* 2012; Riggelsen & Ohrnberger 2014), making use of a wide range and combinations of waveform attributes. However, none of these studies utilize these methods on a network scale for event detection. Sets of picks are grouped based on their goodness of fit to a reference velocity model. These sets lack a unified sense of the probability of event occurrence, as multiple statistics (such as the number of observations, pick quality and residuals) provide their own measure of the event's quality. The lack of a single value representing an event's quality has been overcome by methods that make use of observations by all stations simultaneously (e.g. Baker *et al.* 2005; Grigoli *et al.*

2016; Poiata *et al.* 2016). These network-based detections provide an estimate on the location of the event, as well as a single statistic relating to event likelihood.

1.2.3 Research Intent

In this study, a method for event classification and detection is presented that reports a network wide statistic proportional to the event likelihood, as well as a preliminary event time and location. We seek features that show a high amount of separation of their distributions between groups of seismic signals (hereafter referred to as classes), as classification algorithms are efficient at placing boundaries between the different classes. The method makes use of a machine learning algorithm, which allows for the training of optimal functions that convert seismic attributes to event type probability. Waveform attributes already shown to be successful at discriminating between noise, earthquakes and explosions are utilized. The component attributes are normalized to reduce amplitude-dependent effects. Both the polarization and frequency content across multiple bands are extracted, using multiple time samples about distinct phases. As there is a high amount of flexibility for the input, this method has the potential for use in networks of different scales, and various seismic event classifications. In addition, this method does not require a robust velocity model. The technique utilizes a set of known sources that do not need to be at the same locations as events to be classified, but rather just in a region where the generalized waveform attributes report similar values. The classification and detection methods are applied to separate regions. Classification yields over 99 per cent accuracy in the discrimination between blasts and earthquakes, even when training and test events are spatially separated (section 3.1.2). The event detection method yields 25 times

more events than those reported by the local geological survey with a false detection rate of less than 2 per cent (section 3.2.2). The creation of the algorithm first requires that known event classes can be separated with confidence; this knowledge is then extrapolated to process time-series data. Thus, the event classification method is described first, followed by the event detection algorithm.

2 METHODOLOGY

The methodology is outlined in a schematic flow diagram shown in Fig. 1, and will be a guide on the matrix dimensions of the data for any given stage during processing. Event classification uses steps (a)–(e). Event detection further uses steps (f)–(h).

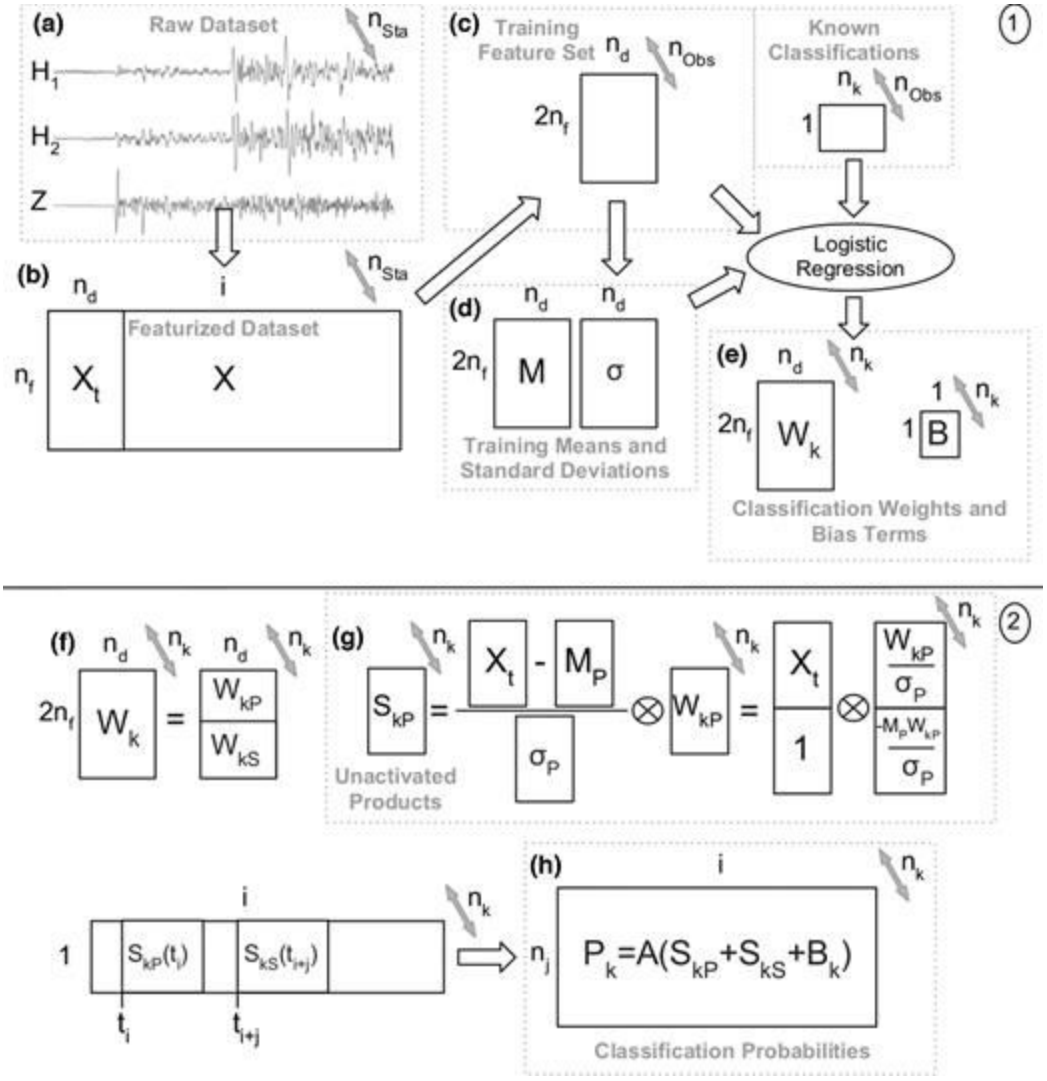


Figure 1: Overview of data processing workflow and related matrix dimensions. Rectangles outlined in black represent 2-D matrices. Variables listed to the left and above a matrix indicate the number of rows and columns. Grey arrow with attached variable above the right corner of a

matrix indicates that the matrix has an extra dimension with given length. The ⊗ symbol indicates the cross-correlation operator. The set of top panels labeled (1) show the workflow required for event classification (a)–(e); panels labeled (2) show additional processing steps necessary for event detection (f)–(h). Individual steps are: (a) raw data extracted from database; (b) raw data converted into downsampled feature data set; (c) selected windows of features to be used for training with logistic regression, with number of known classes; (d) mean and standard deviation of training data set; (e) output weights and bias terms for each classification; (f) splitting of the *P*- and *S*-phase weights to allow for calculation of event probability at any *P*–*S* delay; (g) normalized features multiplied by the output weights, for each classification and phase type; and (h) event probabilities for each classification, organized by *P*-phase arrival time and *P*–*S* delay time index.

2.1 Event classification

2.1.1 Feature selection and creation

This section describes the processing of time-series (steps (a)–(b) of Fig. 1) to produce inputs to the machine learning algorithm. In event classification, different event types or classes (e.g. earthquakes and blasts) can be distinguished based on a number of attributes and qualities of the time-series data: we will refer to these as features. The features that we select for classification are based on the degree of polarization (DOP) and the power spectra of three-component seismograms. We note that features for use in machine learning need not be predetermined; however, to limit the scope and length of this study, we select a number of features previously used by other researchers that successfully represent phase arrivals and their coda. In machine learning applications, an observation is a grouping of features, and the ordering of these features must be consistent between different observations. Here, an observation is a set of features calculated from the analysis of multiple overlapping windows at a single station within a network, for one particular event. Prior to any feature calculation, all time-series are pre-processed by first detrending

using a linear fit, followed by applying a bandpass filter with corner frequencies to be specified momentarily.

A requirement for features is that they have minimal amplitude dependence. This means that they can be equally compared across a broad magnitude range. As feature data are to be treated equally from any station, the features are designed such that they have minimal dependence on the exact shape of the waveforms. Three variables are used to define how often the features are sampled and how far in time they can be influenced by a nearby sample, both from the original time-series as well as the downsampled features. These variables are explained visually in Fig. 2. First, the interval length L_I gives the feature-sampling interval. Second, the window length L_W defines the length of time centred over a given feature sample where the feature is extracted from pre-processed data. Third, the averaging length L_A is the period prior to and including a given feature sample over which the features are normalized. As L_A will reduce the uniqueness of a given feature sample, it should be short enough to reduce the amount of overlap of features between different phase arrival signals, however it should also be long enough to generalize any given phase. L_I and L_W are chosen to reduce the number of final features, decrease processing time and reduce the size of the archived data, although $L_W \geq L_I$ as otherwise some of the original pre-processed data will not be used. Furthermore, L_W defines the longest resolvable period, so it must be large enough to capture the lower frequencies that are characteristic of a given class. The upper frequency limit should also be set to only include frequencies that assist in the discrimination between classes. In this study, the classes show significant energy up to the Nyquist frequency of the raw digitized seismograms. The bandpass filter applied during pre-processing uses

corners frequencies of 0.5 and 49.9 Hz, and the variables L_I, L_W and L_A are set at 1, 2 and 6 s, respectively. The two features selected are the DOP used by Kaur *et al.* (2013), as well as frequency-averaged spectrograms, based on the work by Rabin *et al.* (2016).

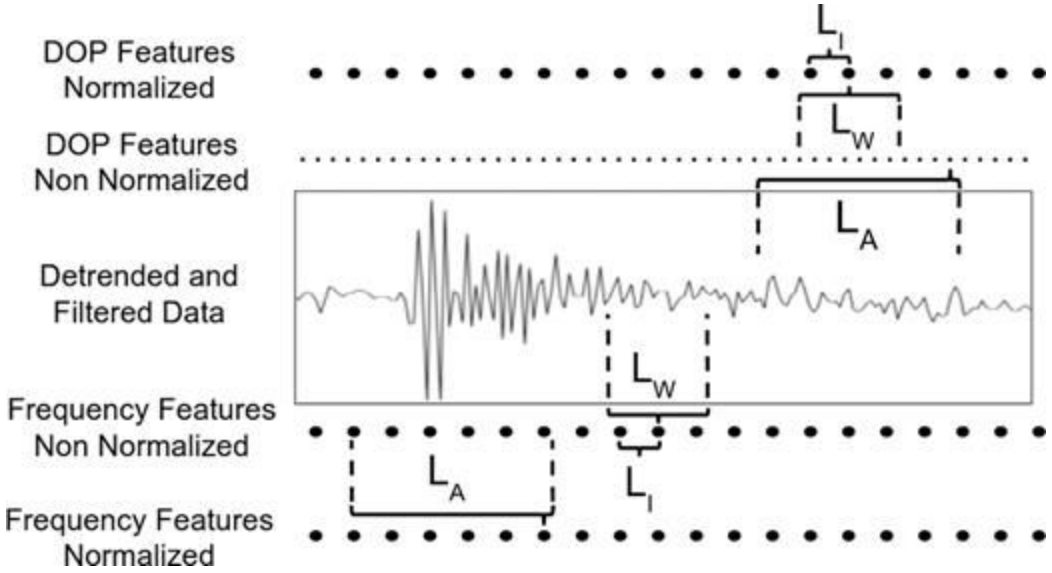


Figure 2: Visual representation of the three variables relating the lengths of time over which the features are sampled and normalized. The three variables are used in a different order for the estimation of the degree of polarization (top), and frequency content (bottom). L_I is the new sampling rate of the features; L_W represents a window centred over a given feature sample where it is calculated; L_A is a duration prior to, or around, a given feature sample over which it is calculated.

DOP is a measure of how much the ground is shaking in a given direction. If ground motion is only occurring along one direction, then DOP = 1. If the motion is occurring in all directions equally the DOP = 0. DOP is calculated using the equation:

$$DOP = \frac{(\lambda_1-\lambda_2)^2+(\lambda_1-\lambda_3)^2+(\lambda_2-\lambda_3)^2}{2(\lambda_1+\lambda_2+\lambda_3)^2} \quad (1)$$

where λ_J is the eigenvalue of the J^{th} principal component direction of a given station's three-component data. The DOP is calculated over the time-series in acceleration units, as opposed to velocity or displacement, as this has been tested here to produce the most apparent differences between different event classes. The DOP was found to be very sensitive to phase arrivals, therefore this feature was first sampled at a high rate of 50 Hz over a period of L_A, to capture early spikes. The DOP feature is then downsampled at the rate L_I and uses a window length of L_W, which spans across the initial DOP values calculated at 50 Hz. We extract three unique features using the DOP: the average DOP; the change of the average DOP relative to the previous sample and the maximum absolute change in DOP within each window. An example of these features is shown in Fig. 3.

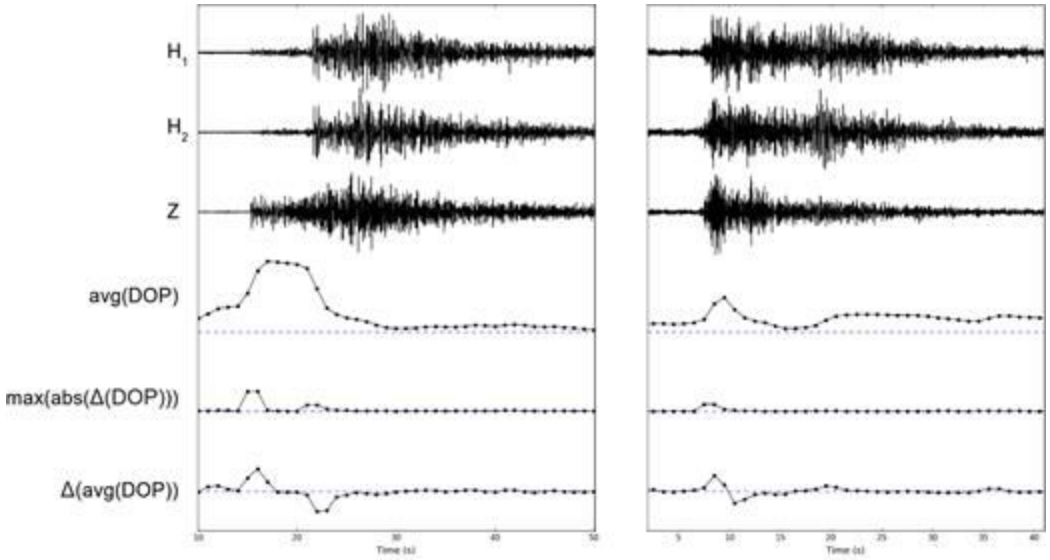


Figure 3: Samples traces of the degree of polarization (DOP) features calculated at a single seismic station. Left and right-hand panels show arrivals from an earthquake and explosion respectively, for all three components of motion. The lower three traces show, in descending order, the average (avg), maximum (max) absolute (abs) change (Δ) and change in the average of the DOP, with a sampling rate of L_I.

The second set of features makes use of the changing frequency content of signals containing seismic events. The frequency range of the spectrogram is limited by the Nyquist frequency, as well as the inverse of the window length L_W. First, the power spectral density, q, is calculated and sampled using L_W and L_I, respectively, from the pre-processed data. For each feature time sample, we sum q within H = 9 distinct frequency bands, yielding q_h, with h= 0–8. We choose non-overlapping frequency bands, starting at the lowest frequency and then increasing in near-half octaves. In this study, the boundaries of the frequency bands are (in Hz): 0.500, 0.833, 1.389, 2.314, 3.858, 6.430, 10.717, 17.861, 29.768 and 49.615. Each band is then averaged over time using L_A. The first frequency feature, F_1, is defined as:

$$F_1(i,h) = \frac{q_h(i)}{\sum_{i-ceil(L_A/s)}^{i} q_h(i)} \tag{2}$$

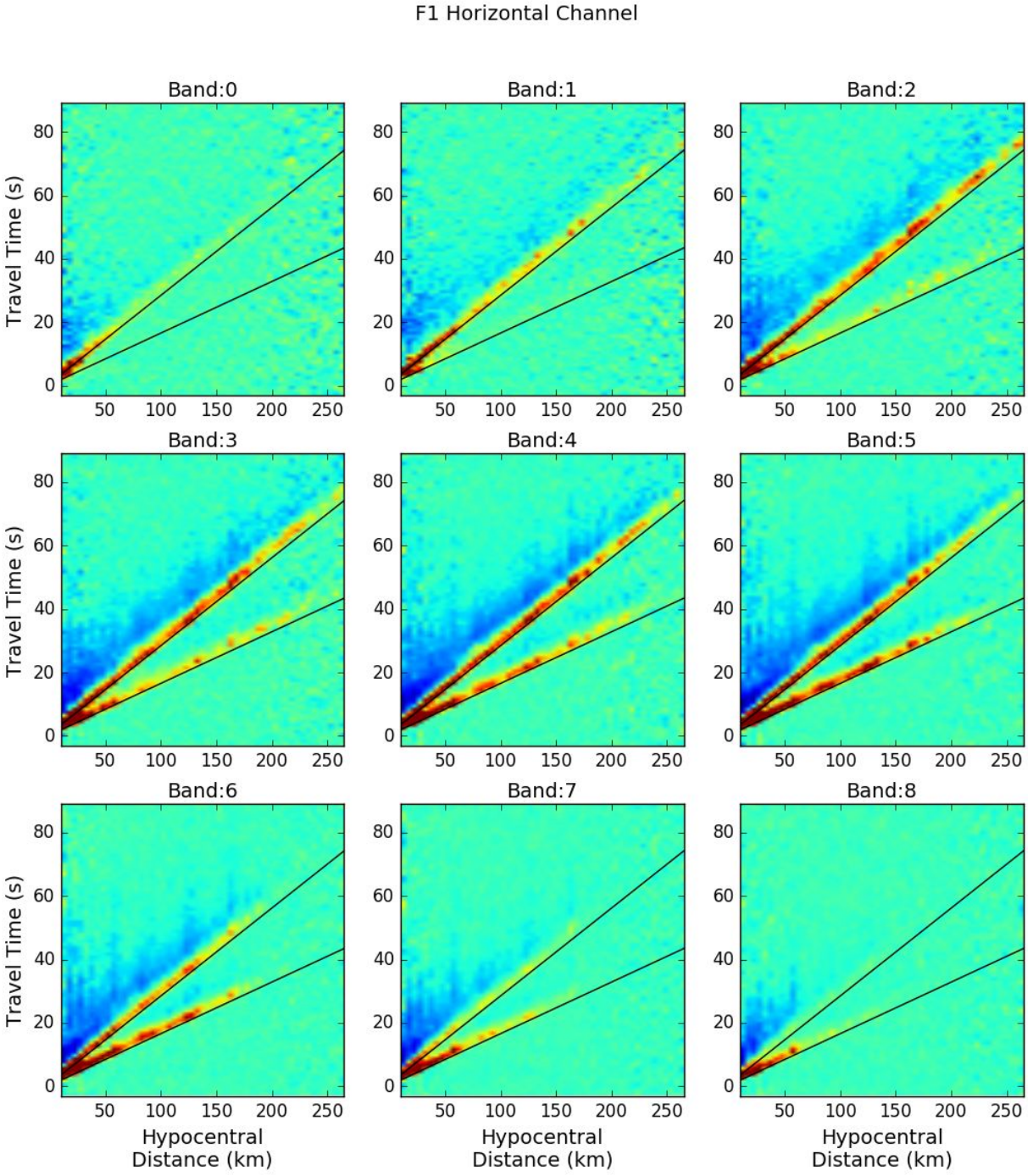
for each input channel, with a signal sampling rate of s, feature time index i, frequency band number h and where the ceil operator rounds a decimal number up to the nearest integer. The F_1 feature captures the relative power over time in each of the H bands separately. Additionally, a second set of frequency features is calculated. The second set is calculated by summing q within each band, all bands are then normalized by dividing by the sum of all bands at the given time sample. The second set is then time normalized by removing the average of each band's values L_A before the given sample. The second frequency feature F_2 is calculated in two steps:

$$Q_h(i) = \frac{q_h(i)}{\sum_{j=1}^{H} q_j(i)} \tag{3}$$

followed by

$$F_2(i,h) = Q_h(i) - \frac{1}{ceil(L_A/s)} \sum_{j=i-1-ceil(L_A/s)}^{i-1} Q_h(j) \quad (4)$$

As the F_2 feature is first normalized across the different bands prior to averaging (smoothing) across time, this feature captures the changes in the portion of power contained within the given band relative to all bands. As each feature is to be neither non-station nor backazimuth specific, the horizontal channel band values are averaged. F_1 and F_2 values on the horizontal channel for earthquakes are displayed in Fig. 4, additional figures showing the vertical channel, and for the explosion class are provided in Supporting Information (Figs S1–S6). The spectrogram results in 36 features, which is the product of the two types of normalization, two motion directions and nine frequency bands used. With the addition of the three DOP features, the number of unique features (n_f) used in this study is therefore 39.


F1 Horizontal Channel
Band:0
Band:1
Band:2
Band:3
Band:4
Band:5
Band:6
Band:7
Band:8
Travel Time (s)
Hypocentral Distance (km)
0
20
40
60
80
50
100
150
200
250

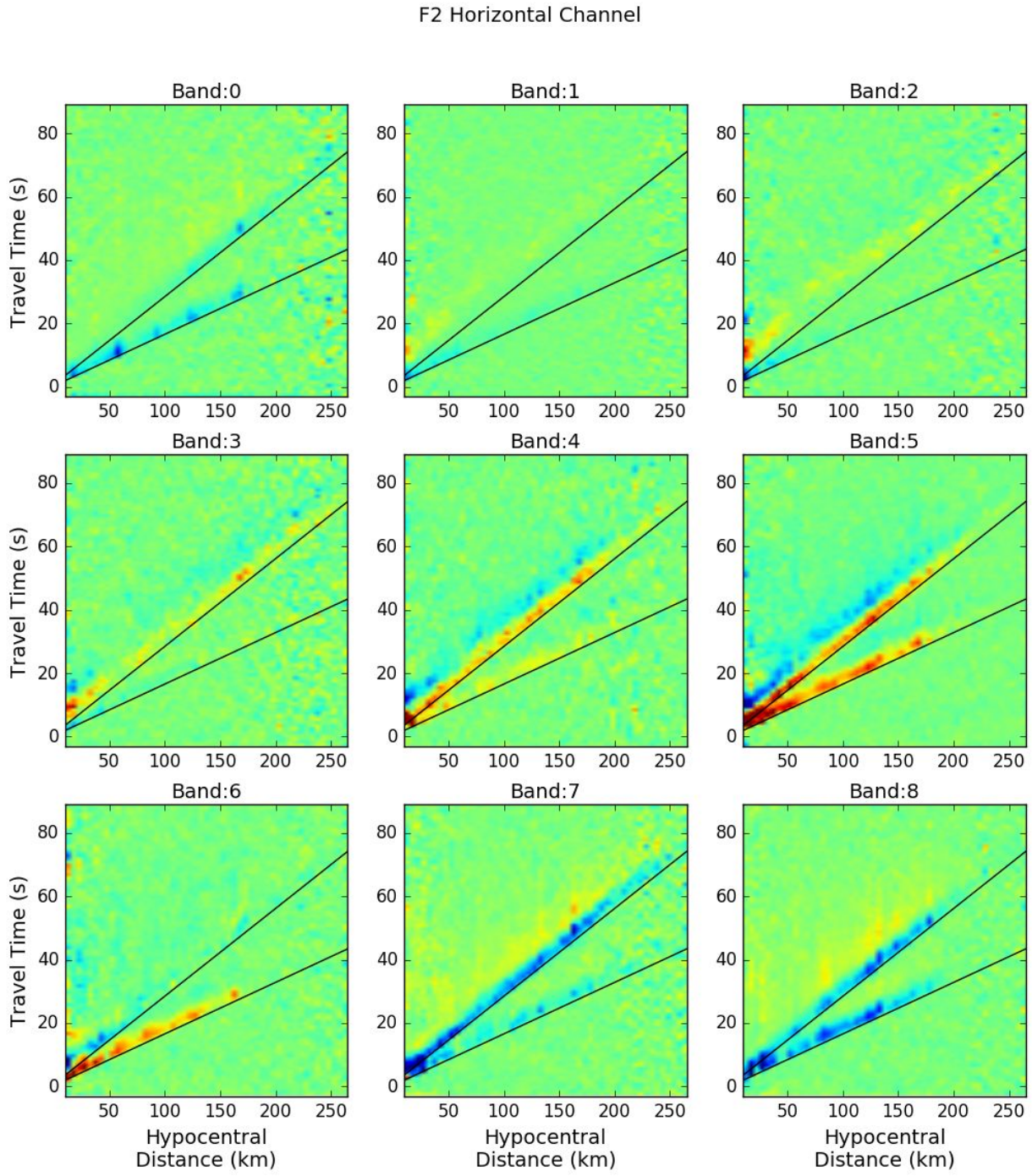


Figure 4: Averaged F_1 (top) and F_2 (bottom) values versus hypocentral distance, and time since origin estimated using the horizontal channels. Values come from all earthquakes and all stations used during event classification. Blue, green and red colours represent low, medium and high values, respectively. The lower and upper black lines indicate the approximated *P* and

S arrival times, respectively. Lower band numbers refer to lower frequencies that are contained within the band, defined using the boundaries: 0.500, 0.833, 1.389, 2.314, 3.858, 6.430, 10.717, 17.861, 29.768 and 49.615 Hz. The value of features for a given phase (*P* and *S*) are broadly consistent across the event-station distances observed relative to a given predicted phase arrival time. Such relations allow for these features to be used during training of weights regardless of event-station distance.

2.1.2 Application of machine learning

This section elaborates on how a machine learning technique can be used to derive an event classification probability, which includes steps (c)–(e) of Fig. 1. A given feature describes an attribute of the ground motion over L_W. Since L_W has a shorter length than a phase coda, multiple feature samples are required to express the coda. This multiplicatively increases the number of input features by the number of useful time samples n_d. As shown in Fig. 4, there are a large number of time samples where the features will largely be noise. As the sequence of features being input to the machine learning algorithm must consistently represent the same qualitative values in order, the temporal alignment of features is required. To align the features and reduce the feature space dimension, samples are selected starting one second before both the expected P and S arrival times, here continuing for a total of eight samples each (n_d = 8). One observation (one event recorded at one station) therefore results in $2n_f n_d$ = 624 features (Fig. 1b). The features relating to the previously known event catalogue with n_{Event} events consisting of explosions and earthquakes and recorded at n_{Sta} stations are extracted to form the known observations (Fig. 1c), where $n_{\text{Obs}} = n_{\text{Sta}} n_{\text{Event}}$.

The known catalogue is split into three separate groups: training, cross-validation and test sets. The training set typically includes the majority of the known observations, while the cross-validation and test sets take half of the

remaining portion. Each set ideally contains similar portions of each of the K known classes. The training set is used to determine the model output values, the cross-validation set assists to constrain the model input parameters, while the test set is used to estimate the error between the predicted output and true observations. In this study, observations are split on a per-event basis, in order to later evaluate the network-wide classification accuracy.

There is a wide variety of machine learning techniques that can be used to predict the classification of a new observation based on previous and known observations. Different techniques are better suited to particular data sets than others, and there is no specific criterion for choosing the best technique. A few techniques were tested briefly here; however, logistic regression (see supporting information, also explained in Hosmer *et al.* 2013) was ultimately chosen due to its computational speed, simplicity and ability to achieve accuracy values close to those obtained using more advanced techniques. Put simply, logistic regression finds the optimal multidimensional plane (hyperplane) for each class that best separates it from all other classes. Using logistic regression, one measure of the probability P_k that an observation is assigned to class k is given by:

$$P_k = softmax(z_k) = \frac{e^{z}k}{\sum_{j=1}^{n_k} e^{z}j} \tag{5}$$

with z_k equal to:

$$z_k = W_k \bullet X_t + B_k \tag{6}$$

where W_k and B_k are the weight array (i.e. parameters of the regression model) and bias array for class k, respectively and X_t holds the features from a given station

starting at time t. There are n_k bias values and weight arrays, one returned for each class k. Each weight array has the same dimensions as a single input observation. As there is only one weight value returned per feature, per class, this machine learning method allows the user to easily review the weights and confirm whether or not they hold physical meaning. Using the softmax activation function forces the probability of a given class to be normalized relative to all others. This assists in expressing an unsure probability, which occurs when a given observation's features lie in a feature space where two or more classes are commonly present. It is important to note here that the term probability refers to how likely an observation is contained in a class, given that the observation is in the span of the training observations. In other words, this probability assumes that the entire population of observations can be represented fully (including the proportion between classes) by the observations in the training set. This assumption is of course false for the data sets to be classified based on the regression model; nevertheless, it provides a rough estimate of the true probability. When solving for the weights that best match the observations to the known class, the features contained in the observations should be normalized to speed up the regression algorithm, and also to ensure that the weight values can be compared on the same scale. Here, each feature is normalized separately by removing its mean M and dividing by its standard deviation σ (Fig. 1d). The logistic regression method aims to reduce the difference between the predicted and the known probability of a given class, the latter being either 0 or 1. The function that governs this difference is referred to as the cost function, given by:

$$Cost = -\left(\frac{1}{n_{Obs}} \sum_{n=1}^{n_{Obs}} \sum_{k=1}^{K} ln(P_{kn}) Y_{kn}\right) + \beta \sum_{i=1}^{n_d} \sum_{j=1}^{n_f} \sum_{k=1}^{K} \left|w_{ijk}\right| \quad (7)$$

where n_{Obs} is the number of observations in the training set, P_{kn}, and Y_{kn} are the estimated and known probability that observation n is class κ, β is the regularization term and w_{ijk} is the i^{th} time aligned sample, j^{th} unique feature and k^{th} class of W. Gradient descent with a fixed learning rate is applied here to minimize the cost function. The model parameters (learning rate, regularization term and number of iterations) are determined through the use of a cross-validation set. To report a network-wide classification for an event, the probabilities from all stations are summed. Here, as we look to classify only between blasts and earthquakes, classification is dependent only on which real event class has the higher summed probability.

If the user has a limited training set, we suggest applying a method for dimension reduction as features use overlapping windows and are averaged using other nearby features, which creates redundant information. Reducing dimensionality has the potential to reduce the chance of overfitting. In this study, principal component analysis (PCA) is applied to capture 95 per cent of the original variance on each phase's features separately; this reduces feature size from 624 to 362 dimensions. PCA, however, reduces the ease of checking the physical meaning of weights. An additional option for dimension reduction is simply to remove a portion of features, as each feature (DOP, F_1 and F_2) provides its own insight into the waveform characteristics and may not be well suited for class discrimination in a particular case.

2.1.3 Addition of the noise class

At lower signal-to-noise ratios, the waveforms look less like any of the real classes resulting in probabilities more equally distributed among the classes. In order to effectively reweight these probabilities to assist with a network-wide statistic, an additional ‘noise’ class (i.e. ‘none of the above’ option) is added to the regression. Here, noise event times are manually selected when there is no resemblance of a real event occurring over a two-minute window. These noise events were assigned a number of random locations each. As all real events ultimately contain noise it is more beneficial that a real event be classified as noise, than noise be classified as real. This can be achieved by adding a larger portion of noise events than either blasts or earthquakes, thus biasing the regression algorithm to choose weights that correctly classify noise. Weights that correctly classify the majority of observations result in a low cost value; thus using a larger portion of noise events favours weights that classify noise as itself. These noise observations are added prior to regression (Fig. 1c). In this study, using between 10 per cent and 100 per cent more noise observations than any real class's observations allowed for optimal results.

2.2 Event detection

2.2.1 Extension of event classification method to event detection

The procedure for event detection tested here makes use of the majority of the components stated previously in the explanation of event classification. In the

previous section, the objective was to correctly classify an earthquake or explosion. Here, the objective is to confidently determine that a true seismic event has actually occurred. In order to detect events, class probability must be calculated continuously over time for each station and subsequently associated to declare events. Furthermore, only a single true seismic event class is required. That is, the user can group any selection of events, local, local and regional teleseismic etc., together to be the non-noise class.

A study period is selected, and the data are converted into the feature space using the same method as described in the event classification section. Noise events are generated by selecting random times and locations within the network, these times are manually confirmed to contain no easily visible events. The training set uses half of this catalogue, while the remaining half is used for determining model parameters.

2.2.2 Addition of the reversed class

During initial training, an issue occurred when the regression was too easy a problem for events with high signal-to-noise ratio, as the resulting *P* and *S* weights showed insignificant differences. The capability of knowing which phase is present is advantageous for later association of phases. We solve this ambiguity by defining a new class, termed the reversed class, which forces the regression to determine what is the difference between *P* and *S* coda. Three classes (real, reversed and noise) are selected for use in the detection algorithm. Here, the real class refers to any seismic event that occurs near the network of stations. The reversed class is a duplication of the real class, but swaps the features between the *P* and *S* arrival

windows. This class also has the beneficial side effect of ensuring that the *S* coda of an earlier event does not stack well with the *P* coda of a later event. The noise class represents all other background signals.

2.2.3 Time-series probability calculation

To use the computed weights for event detection, a class probability is calculated at each station with an interval of L_I. This operation can be represented as the 2-D cross-correlation, symbolized by $\otimes$ in Fig. 1(g), of the determined weights W_k with a moving window X_t for each class *k*. As the input observations prior to regression are normalized by removing the mean and standard deviation, the same normalization is applied to X_t at each time step (Fig. 1g). W has the dimensions [$2n_f$, n_d, n_k], and can be split into values relating only to a given class, and phase type, $W_{k,P}$ and $W_{k,S}$ with dimensions [n_f, n_d, n_k]. Probability is calculated in the same manner used during regression, which is:

$$P_k = softmax(S_{k,P} + S_{k,S} + B_k) \tag{8}$$

where $S_{k,P}$ and $S_{k,S}$ are the products of $W_{k,P}$ and $W_{k,S}$ with X_t at their respective time indices. Cross-correlation is computed on the *P* and *S* phases separately so that these two products can later be summed taking into account a specific *P*–*S* delay time. The combination of times for the selected *P* and *S* windows is represented here by the *P*-window index, and the number of feature time samples between the *P* and *S* windows. The sum of $S_{k,P}$ and $S_{k,S}$ are calculated for all time indices, and *P*–*S* delay indices. This sum is then sent to the softmax activation function (eq. 8) to gather the probability that an event has occurred with the given *P*-arrival window, and *P*–*S* delay time (Fig. 1h). An example of a station's reported earthquake probability is

shown in Fig. 5(a). A threshold T_{Sta} is applied to the resulting probabilities to extract paired times that are likely to be events (Fig. 5b). The threshold is based on the difference between the event probability P_E and noise probability P_N. Paired times that are later used satisfy the condition:

$$T_{Sta} \leq 0.5(P_E - P_N + 1) \tag{9}$$

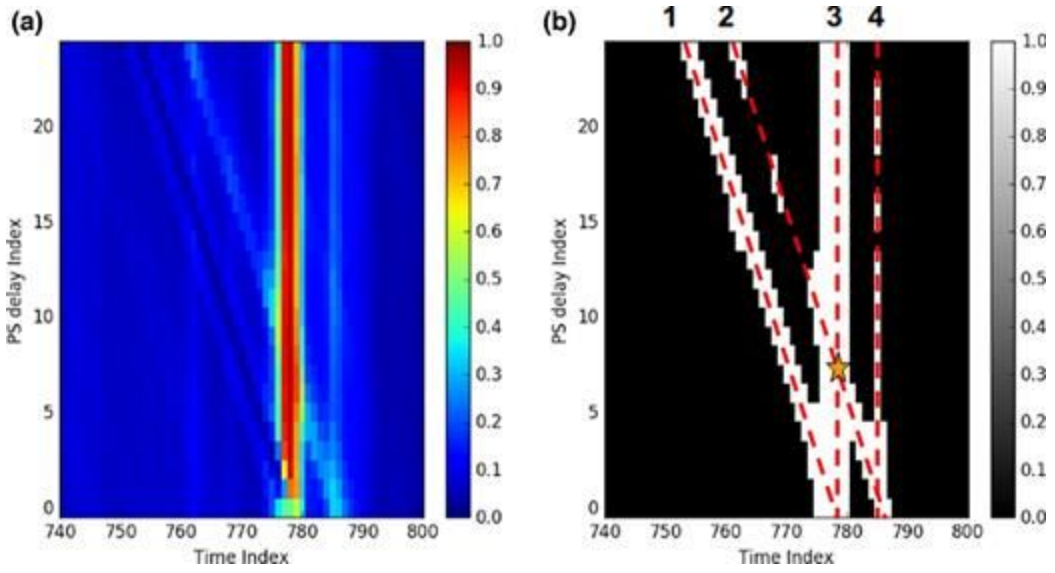


Figure 5: Event probability versus weight window positions observed at station STN23 for a magnitude 3.7 M_L occurring on 2015 September 9 at 03:42:49 UTC. *P*-phase arrival time is represented by the horizontal axis, while *S*-phase arrival is represented relative to the *P*-phase arrival time along the vertical axis. Panel (a) shows the probability that an event is occurring. Panel (b) demonstrates the binary thresholding of: (probability event − probability noise + 1)/2. The marked orange star position depicts an event origin; all values along the dashed red lines $\pm L_A$ are removed during future event detection iterations. These red lines in (b) indicate positions along which there will likely be a high probability with reference to the event origin. These are: (1) *S* weights producing a high probability, while positioned over the *P* arrival; (2) *S* weights giving high probability positioned over the *S* arrival; (3) *P* weights causing high probability positioned over the *P* arrival and (4) *P* weights producing high probability positioned over the *S* arrival.

Also, it is important to remember that

$$P_E + P_N + P_R = 1 \tag{10}$$

where P_R is the reversed classes probability. This is the result of using the softmax activation function. Noise probability is accounted for in this thresholding as weights for this classification may represent an unlikelihood that noise is occurring. As noise events are classified based on the fact that no real events are present, when coda occurs which is similar to an event, the probability of noise decreases. Thus, taking the difference between real and noise probabilities, and normalizing to be between 0 and 1, can give a more realistic sense of the true event probability. This unlikelihood is especially visible for the training set applied here, shown in Fig. 6. In this figure, negative weights are assigned to features that are large when aligned with a real event.

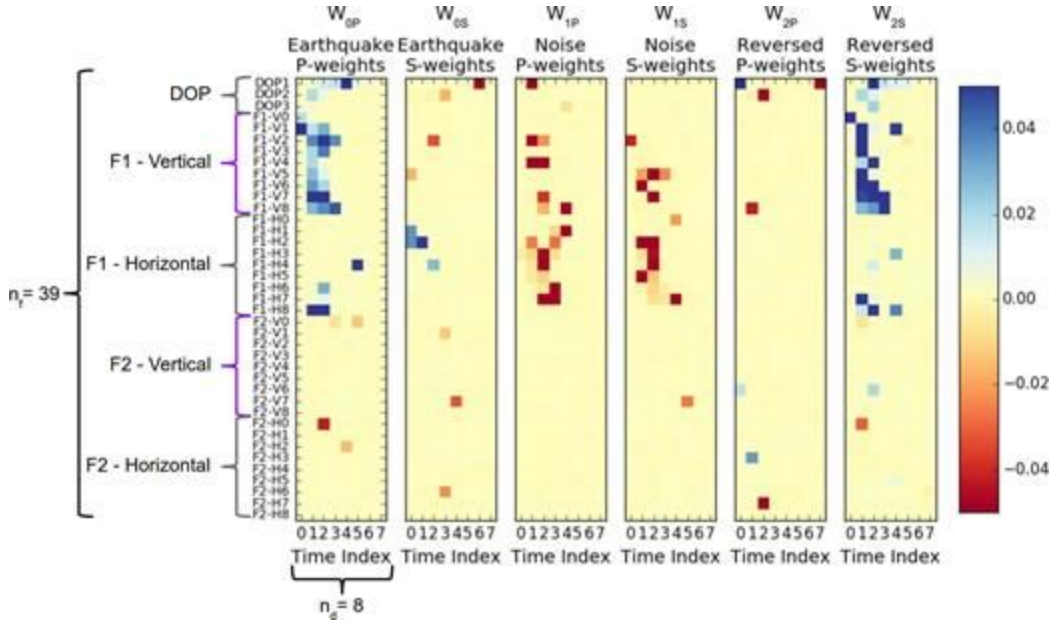


Figure 6: Weights applied during event detection for each class and phase type. For this set of output weights, they show that an earthquake is more likely if the *P* window is over a time where the F_1 content is larger than before. The noise weights indicate that noise is less likely if the F_1 content is larger. Here, the presence of an *S* phase is indicated by the negative F_1 horizontal channel weights in the noise class. A negative weight reduces the output probability and all classes probabilities sum to one, thus an earthquake is more likely when the noise probability decreases. Also, as expected, the reversed class shows near mirrored weights (reverse *P*

phase looks like earthquake *S* phase and vice versa), which reflects how the training data were generated. The average DOP (DOP1) also shows some use in identifying *P*-phase coda. F_2 features in this case do not play an important role in the discrimination between the classes. However, F_2 features have been seen to provide significant assistance when differentiating between multiple real (i.e. non-noise) classes.

2.2.4 Association algorithm

At this point, each station holds a thresholded (using T_{Sta}) list containing trios of: probability for each class, P-arrival time and $P-S$ delay time. In order to declare an event, these probabilities must be combined across the entire network, and then thresholded on an event basis. Although it is possible to apply another (or the same) machine learning technique to produce a single network statistic, the techniques (aware to the authors) require a fixed number of inputs. If the networks were to change size, or a particular station went offline, a new empirical formula to create the network-wide statistic would be required. Thus, a more typical stacking association algorithm is applied here: (1) the trios are first converted to shells with a probability value; (2) next the shells are stacked and (3) the maximum likelihood origin is extracted in an iterative process. Each of these steps is discussed separately below.

In step (1), all trios that pass the threshold are sent on to calculate its estimated origin time, and event-station distance. This calculation is done using the estimated P-arrival window time, the pair's associated $P-S$ delay time and the supplied velocity model. The probabilities are multiplied by a distance-weighting function that decreases with increasing distance. The origin time and event-station distance represent a half shell in 3-D space where the probability of an event has occurred at the given origin time. Shell radius values are discrete, as the potential

$P-S$ delay times are discrete. The shell thickness is determined from the velocity model and L_I such that no shell overlaps between each discrete hypocentral distance. Thus, for a given event origin time range with length L_I, no shell emanating from the same station overlaps. The thickness of the shell represents the association algorithm's inherent hypocentral distance error. Each shell's probability P_E is multiplied by a distance-weighting function.

In step (2), all shells with an origin time ±½ L_I are stacked (i.e. shell probabilities are summed) onto a 3-D grid. An example slice of this grid at the estimated focal depth is displayed in Fig. 7. In step (3), once the summed shell probabilities are obtained over the study period, the events are extracted in an iterative process. First, the gridpoint and origin time with maximum summed shell probability is extracted and declared as an event. Then, all shells related to that event are removed from the stacked grid. This is performed by removing any trio within L_A of the dashed red lines displayed in Fig. 5(b) to ensure that phase codas from the extracted event no longer contribute to the summed probability. The summed probability is then recalculated, and these two steps are repeated until the maximum event probability is lower than a user-defined summed probability threshold, T_{Sum}.

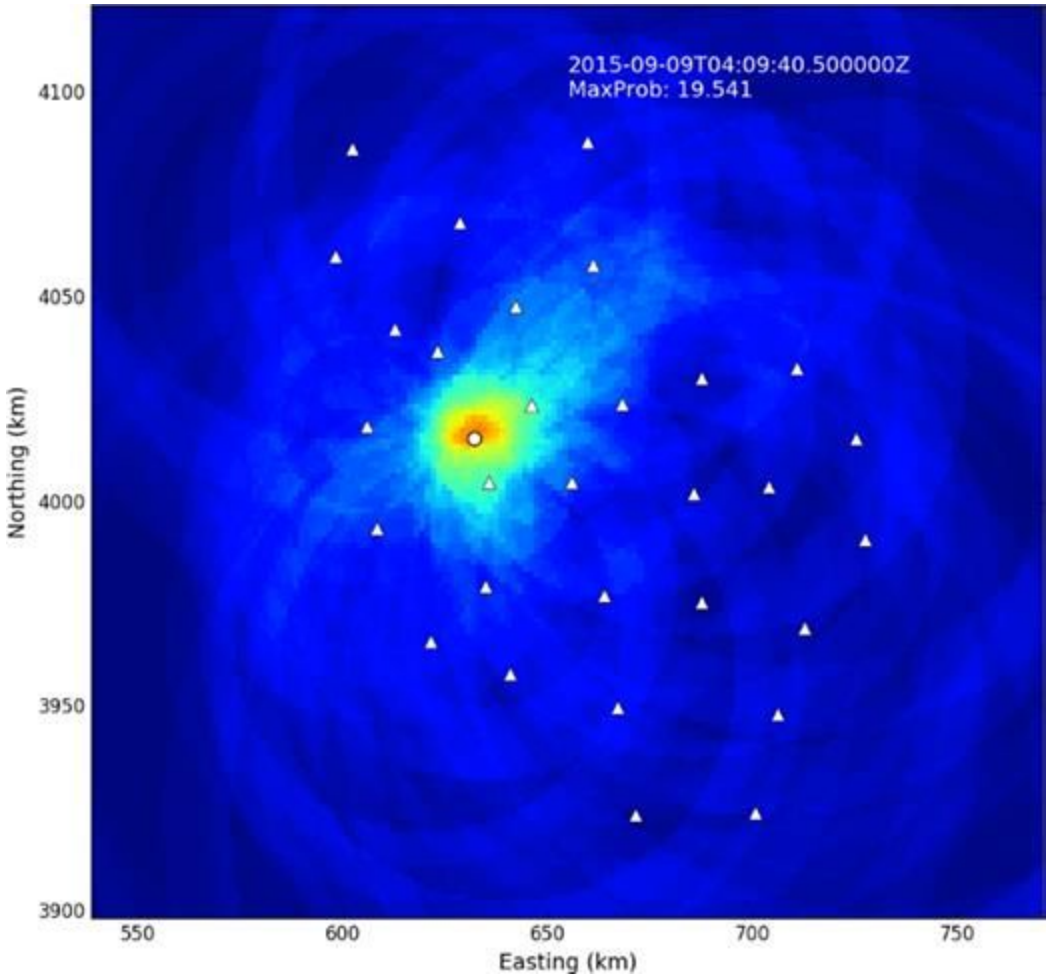


Figure 7: Horizontal slice at 11.5 km below sea level of the stacking grid used for event detection. White triangles represent station locations and the white circle shows the maximum probability location. Warmer colours show locations with a higher event probability.

3 EXPERIMENTAL RESULTS

3.1 Event classification

3.1.1 Chosen data set

To investigate the use of a machine learning algorithm on event classification, a data set was taken from the USGS catalogue in Fig. 8. The extracted catalogue consists of 335 explosions, and 350 earthquakes from 2015 January to 2016 September. Events were only taken within a small magnitude range, from 1.4 to 2.2 M_L, as recorded explosions are largely limited to this range in the study region. Narrowing the range for analysis also allowed us to compare earthquakes and explosions of similar amplitudes. Stations were chosen to: (1) cover the study area evenly; (2) be within 300 km of the furthest catalogue event; (3) have three-component recordings and (4) have a tolerable level of noise, such that picks could be placed manually at least some of the time on events in the extracted catalogue. A total of 13 stations (n_{Sta} = 13) were selected from the Caltech Regional Seismic Network, all of which have a sampling rate of 100 Hz. Including noise observations, n_{Obs} = 12 453.

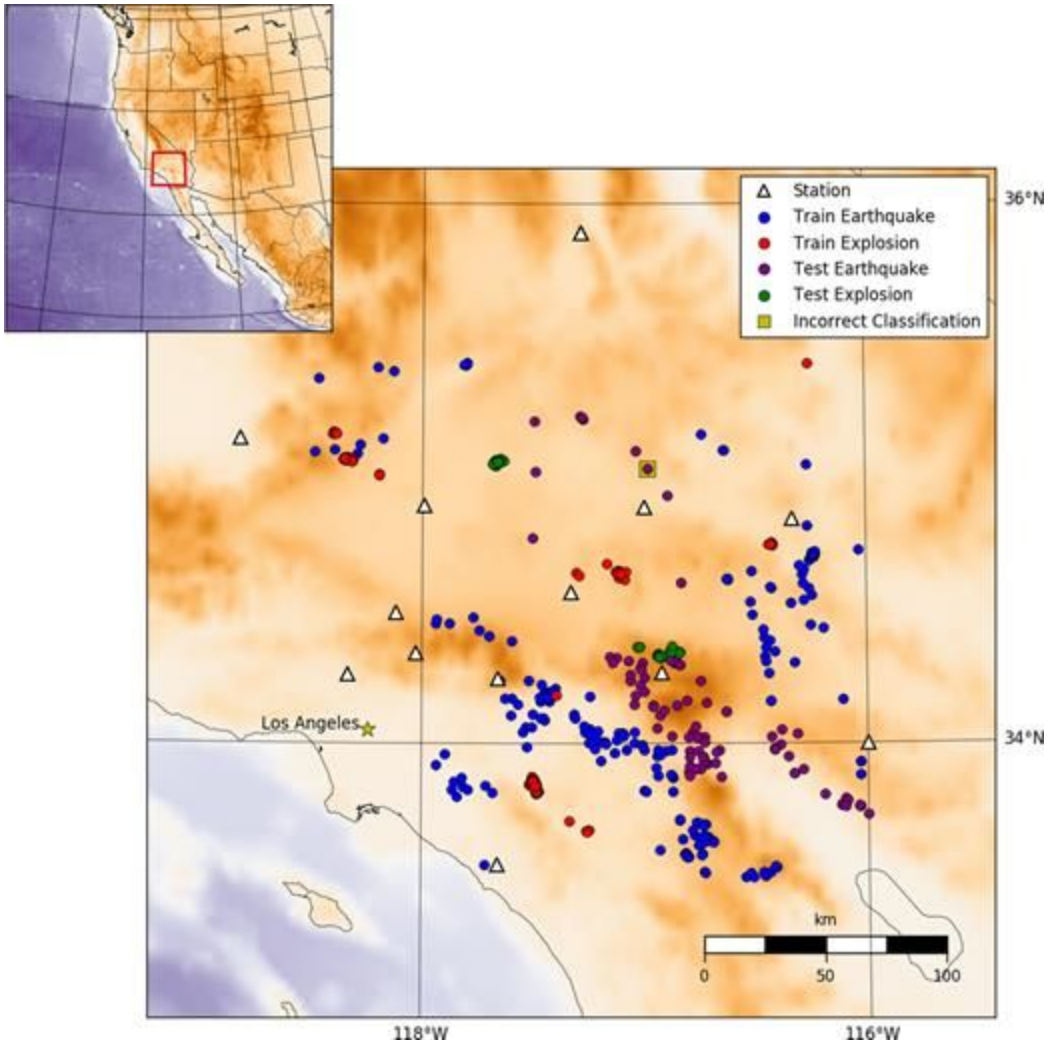


Figure 8: Map of events used during event classification, including the training and test sets extracted from the USGS catalogue for the period 2015 January–2016 September. In this second example, the training and test sets are chosen to avoid spatial overlap (the first test is described in Section 3.1.2). Magnitudes range between 1.4 and 2.2 M_L.

3.1.2 Station and network accuracy

The first experiment was split randomly on a per-event basis giving 60 per cent, 20 per cent and 20 per cent to the training, cross-validation and test sets, respectively. In a second experiment, a training and test set were separated spatially to reduce the chance of overfitting due to the large number of features; see Fig. 8.

This test used the same model parameters as those selected in the first. For blast and earthquake observations, the result is deemed correct if they show a higher probability for the appropriate class while ignoring the noise probability. Station accuracy is calculated based upon the portion of correct blast and earthquake predictions. Network accuracy takes into account the summed probability across all stations for a given event; again noise probabilities and noise events are ignored. The first experiment test set yielded a station accuracy of 84.1 per cent (1461 of 1737 observations), and a network accuracy of 99.3 per cent (136 of 137 events). As the second experiment used the same model parameters as the first, there was no need for a cross-validation set. The second experiment test set had a station accuracy of 85.3 per cent (2317 of 2716 observations) and a network accuracy of 99.5 per cent (213 of 214 events). Both test sets had one event that was misclassified, which was the same event in both cases.

3.2 Event detection

3.2.1 Chosen data set

In the detection method, we select a seismically active region that is different from the study region used in the classification section. Using different regions allows for a confirmation of the generalization of the classification method. The Nanometrics Research (NX) array located in Oklahoma, USA, was used as the first study region for event detection. The NX array consists of 30 three-component stations operating at 100 Hz, they are shown in Fig. 9. Weights from logistic regression are determined through the use of 200 earthquakes from the USGS

catalogue and 504 noise events. Earthquakes from the USGS catalogue were selected at random in year 2015, in a buffered region that includes areas approximately 100 km away from the nearest NX station.

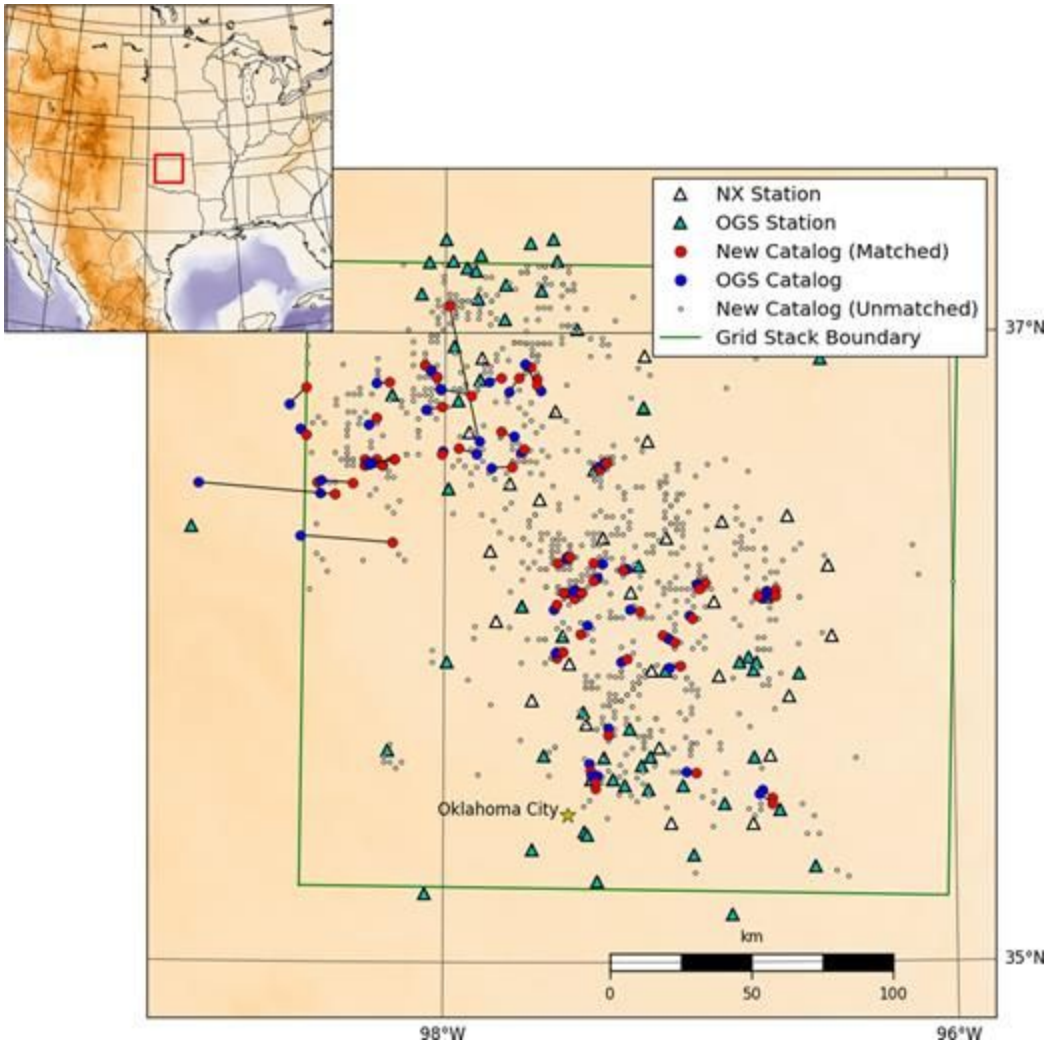


Figure 9: Comparison of epicentral locations for events detected in Oklahoma between 2015 September 7 up to and including September 13, which are contained in the OGS catalogue. Black lines connect the same events between catalogues. Stations are seen as part of the OGS network if the stations network code is in use by the OGS and the station has data available during the study period. OGS stations are shown for a visual comparison of the two networks coverage and density. Only new event locations with an estimated magnitude above the new catalogue's magnitude of completeness (M = 1.2) are plotted here. The green box outlines the

limit of the grid search area. Catalogue events that occur outside of this boundary are mislocated inside the boundary.

3.2.2 Detection summary in Oklahoma

The detection algorithm ran over a week of data, from 2015 September 7 up to and including September 13. There were 1954 events detected, using a station probability threshold T_{Sta} of 0.3, with an event summed probability threshold T_{Sum} of 1.75. The 1954 detections were visually inspected and are comprised of 1919 (98.21 percent) real events that occur locally, 34 local noise events and 1 teleseismic event. Seven of the 1919 real and local events are suspected blasts, based on their waveform shape. Of the 34 local noise events, 31 occur in two bursts on September 11 at 8 and 11 h UTC—these waveforms are weakly correlated between stations. No single event is reported more than once. The real events range in magnitude from 0.3 to 3.9 M_{L} (Fig. 10), these values come from an empirical peak ground velocity (PGV) magnitude relationship. An equation, similar to those as used in Akkar & Bommer (2010) as well as Yenier & Atkinson (2015), used for the relationship here is:

$$log(A) = c_0 + c_1 M - c_2 log(R) \tag{11}$$

where A is the PGV in ms^{-1}, M is the magnitude, R is the hypocentral distance in km and c_0, c_1 and c_2 are constants. The constants c_0, c_1 and c_2 were estimated to be −5.55, 0.93 and 1.32 respectively. Only PGV values from stations which are used for the detection of a given event are used. The median station magnitude is taken as the estimated event magnitude, a comparison of the estimated magnitude to those reported by the OGS is shown in Fig. 11. The catalogue generated here has a magnitude of completeness of 1.2 (Fig. 10).

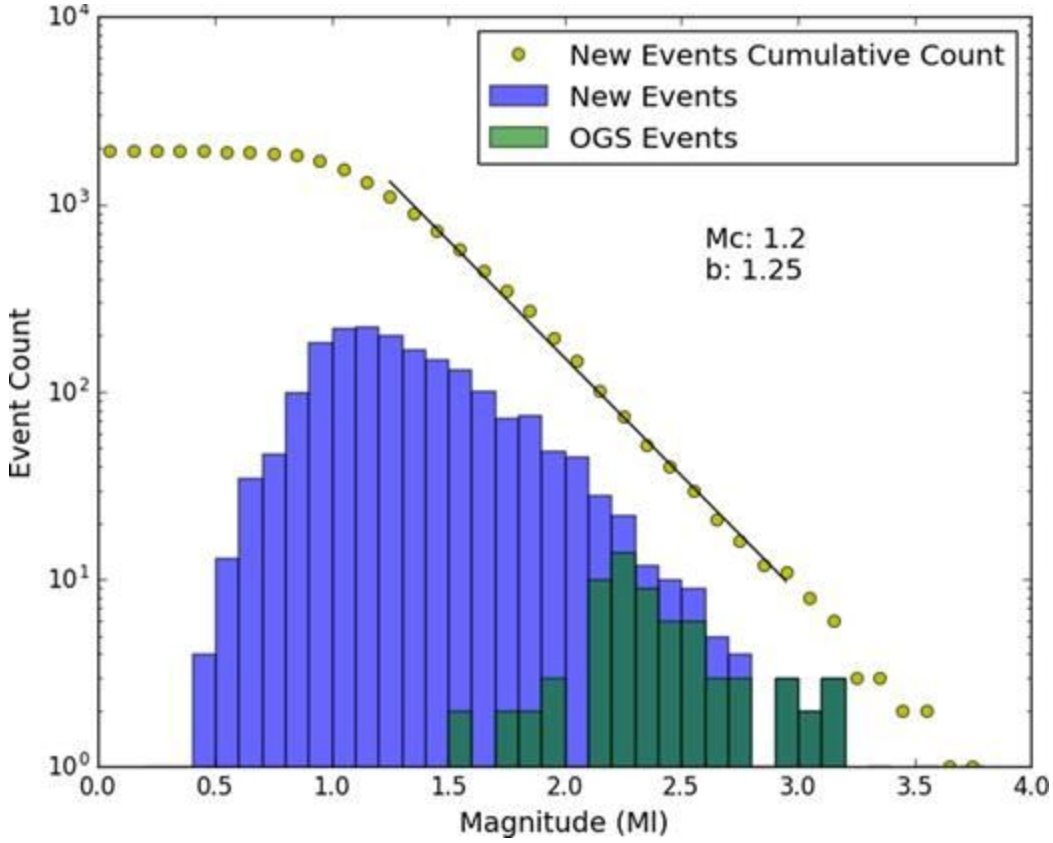


Figure 10: Event magnitude histograms for the event catalogue created in this study (New), and the catalogue provided by the OGS containing event origins from 2015 September 7 to September 13. The black line indicates the magnitudes over which the *b*-value was calculated.

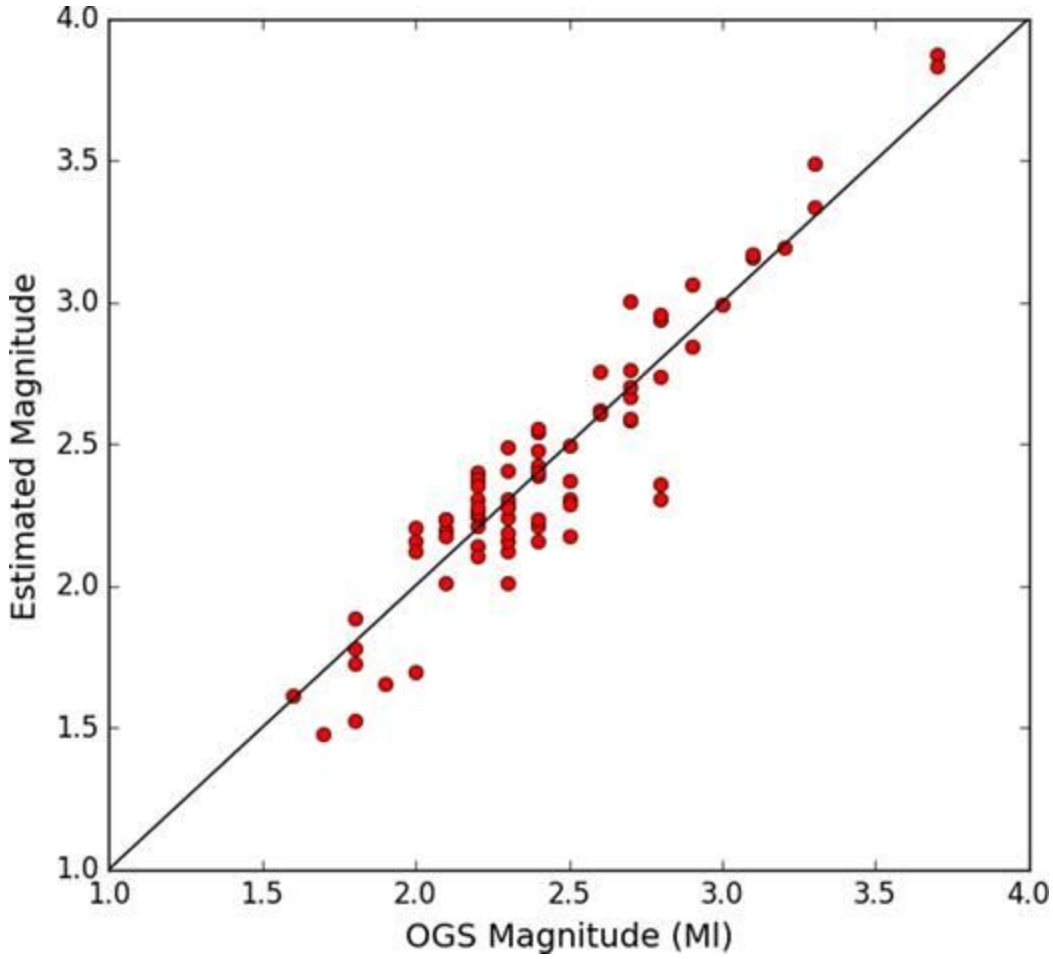


Figure 11: A comparison of the estimated magnitudes calculated using a PGV–M_L relationship to those as reported by the OGS. The black line provides a 1:1 line for reference.

3.2.3 Comparison of detections with the OGS

During the same time period of data as this new algorithm was run, the Oklahoma Geological Survey (OGS) reported a total of 77 events. Two of these events are reported twice, leaving 75 events. The OGS catalogue of events range in magnitude from 1.6 to 3.7 M_L with a magnitude of completeness of 2.3. All events in the OGS catalogue are also detected through use of the new method shown here. The ability for the new method to appropriately detect all previously catalogued events gives confidence into the ability for the method to generalize waveform

characteristics. A comparison of the newly generated catalogue, and the OGS catalogue shows that the event origin times and hypocentres agree reasonably well. Origin times and hypocentres differ by less than 3 s and 13 km for 73 of the 75 matched events, up to a maximum of 10 s and 50 km. The median epicentral and hypocentral difference between the catalogues is 3.3 and 6.7 km, respectively. An event epicentral location comparison is displayed in Fig. 9. In this region, the method shows to be a reliable first approximation of event origin time and location.

4 DISCUSSION

4.1 Event classification

The underlying aspiration of this study is to generalize event waveforms in a broad, yet unique sense so that machine learning tools can be used with high confidence in event classification and detection. Seismic recordings were shown to be fairly consistent across stations hundreds of kilometres apart when using attributes that utilize characteristics smoothed over time and are not dependent on the exact waveform shape. Such features allowed for the successful separation between event classes.

4.1.1 Effects of adding a noise classification

Adding the noise classification has the effect of increasing the confidence between the correct and incorrect class for events that have a low initial confidence. This transition between no noise and the addition of noise, and its effect on classification probability is displayed in Fig. 12. For events that originally have a high confidence indication, there is more likely to be a reduction in confidence when adding the noise classification. However, this is not a significant drawback as the difference in probability for the originally confident events is not enough to change the predicted class. As shown in Fig. 12, different amounts of noise have different distributions of changes in probability. Comparison between training of weights for event classification with different noise input shows that when moving from zero to one portion of noise, the majority of events increase in confidence. This effect is

boosted when moving from one portion to two portions of noise. However, when adding the third portion of noise, the confidence of the correct classification remains nearly constant. This indicates that there is an optimal amount of noise that should be added during training. As generating noise events is fairly trivial and requires little extra work, we recommend finding the optimal amount for each data set. The portion of noise selected can be added in as a model variable during regression to be determined with reference to the cross-validation set.

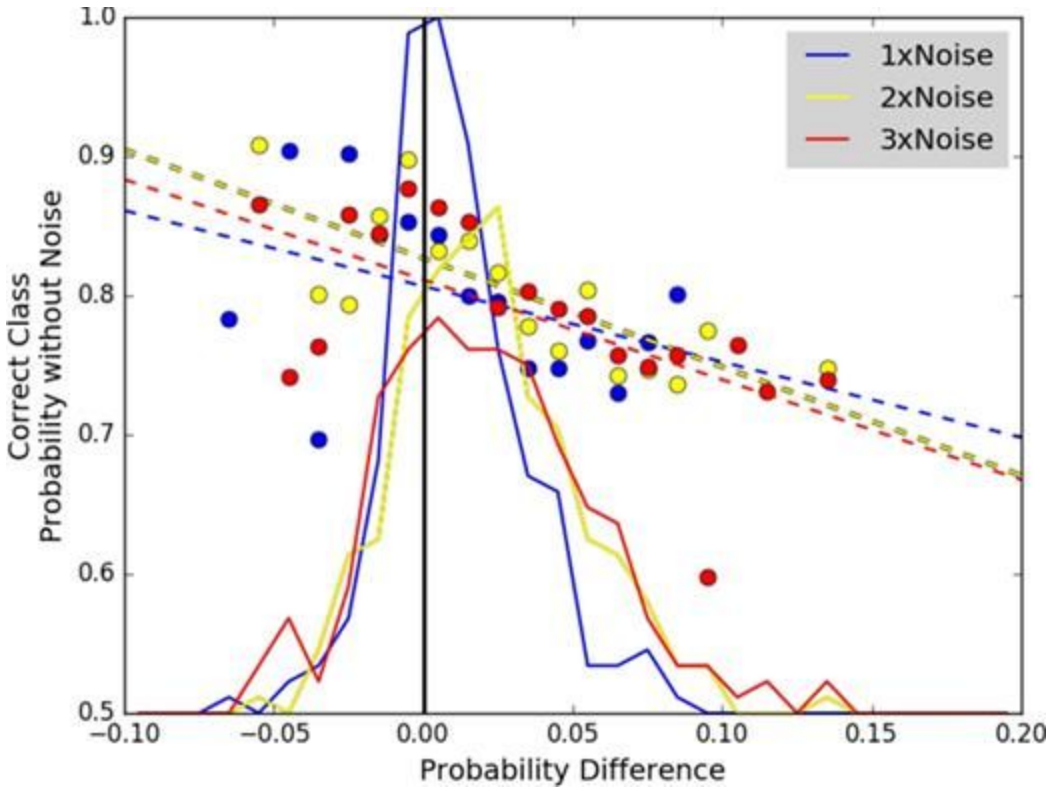


Figure 12: Comparison between different amounts of noise being added during classification training. Event probabilities corresponding to the correct class are examined here. Event probabilities are normalized first by dividing by the sum of all real event classification type probability. The horizontal axis represents the difference in the probability of the correct class; that is, the difference between training with no noise and training with a given amount of noise. Vertical axis shows the probability of the correct class, without any noise added during training. Each colour represents a different amount of noise in the training set. For example, '3 × Noise' has three times the amount of noise in the training set than '1 × Noise'. Data points indicate the

average initial probability within a given probability difference bin. Dashed coloured lines show the least-squares solution to the matching colours set of points. Solid coloured lines display the portion of events within a given difference bin, which are normalized to fit within the vertical axis. Here, the optimal amount of noise to include during training is the '2 × Noise' as this scenario shows the highest portion of events to the right of the zero probability difference line. Best-fit lines for all scenarios display similar trends. The trend indicates that events with lower confidence of the correct class without noise are assisted more by the addition of noise than events that originally had a high confidence of the correct class.

4.1.2 Effects of PCA and potential for a better classifier

PCA was applied during event classification, but it yielded only marginally larger station accuracy. However, reviewing the first few principal components provides some useful insights. Classification types are overlapping in PCA space, with the noise classification being the most clustered, as shown by Fig. 13. The figure shows that the optimal boundary between the clusters is non-linear and suggests that the use of a non-linear classifier is warranted. Preliminary results show though that output station accuracy using techniques with non-linear classifiers, as compared to logistic regression, is only slightly larger (less than 1 per cent station accuracy improvement with optimal model parameters). Although linear boundaries apply just a rough boundary between classes over a few dimensions, this approximation is close enough to more complex models, especially when considering the entire feature space. In this study,hyperplanes are used to segregate the classes, as this allows for easier checking of the physical meaning of output weights with minimal reduction in station accuracy. Checking the physical meaning allowed for rapid prototyping and thorough understanding of the method; however, the use of easily understood output weights should not be a limitation for future improvements of the method as other machine algorithms may be more appropriate given the user's amount and content of training data.

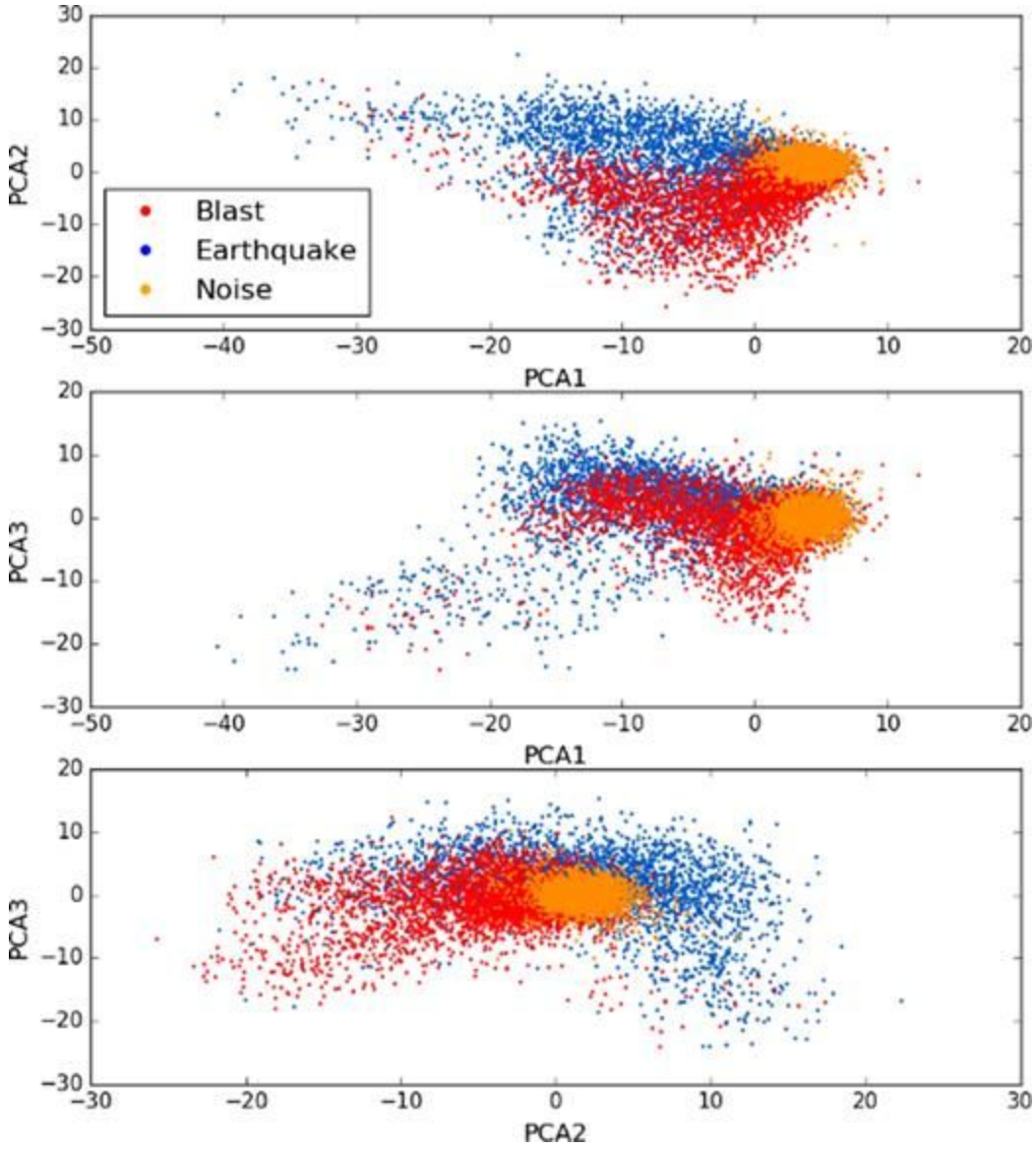


Figure 13: Plot of the reduced feature space after application of PCA during event classification. Only values along the first three axes (of the 362) that contain the most variability are displayed. Logistic regression determinesa hyperplane that best separate a given class from all other classes. Here, it can be seen that although a linear boundary can be used to separate classes with moderate success between any two axes, a non-linear boundary would be more suitable. However, when considering all possible pairs of axes, the linear boundaries are shown to provide comparable results to those of non-linear boundaries provided by more advanced machine learning techniques. Also, linear boundaries allow both for weights to hold a simpler physical meaning as well as a reduced number of weights, which reduces the potential for overfitting.

4.1.3 Features and their potential distance dependence

Waveforms decrease in amplitude with increasing distance from the source and higher frequency content decays faster due to anelastic attenuation. The expectation was that values of a given feature would change significantly as compared with the observation's event-station distance. This idea was confirmed when reviewing traveltime plots such as those shown in Fig. 4. In this figure, the first-order observation and most obvious fact is that the amplitude of the signal decreases with increasing distance. The second-order observation, which can be confirmed by reviewing the ratio of a band's amplitudes, is that the relative amplitude of bands changes with distance. Larger relative differences occur between bands that are more separated in the frequency domain. To be able to incorporate such relationships into the regression model, a non-linear combination of the features is required. Testing this using the event-station distance as a feature and a non-linear classifier revealed that the output weights were much more likely to classify as noise with increasing distance. Although this is true, the non-linear combination of features has the strong potential of removing useful information at large distances. However, an unwanted consequence of not using distance as a feature, while also using multiple windows about phase arrivals, occurs with observations that are taken close to the event hypocentre. At close distances, there is overlap between the *P* and *S* coda, which produces feature sets that are uncommon. This overlap is amplified by the averaging of features. This has the potential of pushing the observation towards an incorrect classification, as the feature vector does not conform to its typical orientation. If distance were used as a feature, the algorithm would have the potential to learn to treat observations with small

distances differently. This negative effect should be considered when choosing L_I, L_W, L_A and n_d knowing the portion of intended training samples at near event-station distances. Here, this effect had no significant impact on results, as only a small portion of the training set had observations at distances where the P- and S-coda window overlapped.

4.2 Event detection

4.2.1 Regression tuning and potential use of smaller events

Training of weights in this section made use of 100 earthquakes from the USGS catalogue and 252 noise events selected from random times in year 2015. As the signal-to-noise ratio of the input events is quite large, calculation of the weights can become quite arbitrary. To employ an analogy, if one were asked to rate books as either good or bad, while all of the good books had a green cover, and bad a red cover, the individual would soon realize that the colour of the cover is proof enough without reading the book. In this case, the algorithm can reach a very accurate solution using just a few of the input features. This analogy exposes the issue with regression when just a few features are needed to separate the classes, whereas in reality there are many more features that could have been leveraged, especially when reviewing a low signal-to-noise event. To compensate for this issue, weights were assigned a penalty that increased the cost function (eq. 7) and that scaled with the weights' absolute magnitude, that is, L1 regularization was applied during weight training.

4.2.2 Weight checking and effects of adding a reversed classification

Manual review of the weights as output by logistic regression is recommended when assessing their quality, as logistic regression provides a 1-to-1 ratio of weights to features. Resulting weights can be seen in Fig. 6. In this figure, the first column of features represents the values of a waveform attribute before the expected arrival. These weights should be significantly smaller in non-noise classifications as compared to the remaining columns. The remaining weights agree with expected ground motion behaviour caused by earthquakes, where typically the *P* coda shows higher amplitude on the vertical, while *S* coda is larger on the horizontal channels. The earthquake and reversed classes show nearly perfectly mirrored *P* and *S* weights, which is appropriate, as during training the input features for *P* and *S* are swapped. The F_1 features from the vertical channel are seen to scale positively with earthquake likelihood, while the F_1 features from the horizontal channel scale negatively with noise likelihood. In this portion of the study, the F_2 weights are shown to be insignificant, however, they remain here for a few reasons: (1) the F_2 weights are shown for completeness within the paper; (2) the abundance of training data and lack of worry on overfitting and most importantly (3) because the F_2 features have shown an ability to significantly increase the classification accuracy when using other data sets. This observation highlights another advantage to using machine learning, as selected features that do not yield an ability to differentiate classes are naturally down weighted. This allows users to utilize a number of

features that may assist in determining the class, at the expense of increased chance of overfitting.

4.2.3 Threshold and distance-weighting selection

Two thresholds, T_{Sta} and T_{Sum}, are required to decide if an event has occurred. T_{Sta} was subjectively set to 0.3, which was found to capture the majority of real event coda while discarding noise. Although a threshold for T_{Sta} of 0.5 would suggest that an event's coda is most likely in the window, a portion of the probability is captured by P_R, which reduces the maximum value possible for P_E + P_N. T_{Sum} was set to 1.75, below which noise events became more common. Near the threshold T_{Sum} of 1.75, it implies that two stations with high P_E could have stacked (0.88+), or three to four stations with a mid-range P_E (0.44–0.59) could have stacked.

The distance-weighting function reduces the chance that very local events, or bursts of noise with earthquake-like coda stack well across stations that are far apart. However, too little weight at larger event-station distances will cause events to be reported more than once when the majority of stations are at large distances away from the true event location. This duplication of reported events can occur when the correct origin is misplaced, thus only removing a portion of the shells contributing to the true origin, allowing the remaining shells to stack elsewhere. A station density-dependent weighting should also be considered when stations are at a distance less than the maximum shell thickness. The network used for this study provided a fairly evenly spaced set of stations. If the network were to have two identical stations stacked on top of each other, their contributing P_E values, although observing the exact same waveform, would contribute twice to the summed

probability. This unwanted effect would occur for any stations within a distance of the maximum shell width. Such stations should be downweighted to avoid supplying the same observation multiple times. In this study, stations were not close enough to consider station density-dependent weighting.

4.2.4 Catalogue comparison

The NX array and the stations that are available for use by the OGS during the study period give a similar station density (Fig. 9). However, the real event count detected in this study outnumbers those as reported by the OGS by a factor of 25. Events reported here are given with a high confidence level (98 per cent were real). The OGS catalogue documents events that have been manually reviewed by an analyst, and are thus limited by time constraints in such a seismically active area. Thus, the comparison of the catalogue output produced here with the OGS catalogue should not be seen as a direct comparison between automatic detection algorithms. However, institutions could report these lower magnitude events as automatic solutions, as the new method provides high confidence in the occurrence of events.
The resulting locations reported here make use of a highly simplified velocity model, using uniform velocities for both *P* and *S* waves. Also, the resolution of the shells used during this study is quite coarse (grid cell length is approximately 2 km). As the intent of this part of the study is to detect events, the factors relating a more accurate and precise location were not optimized. The locations could be improved by implementing a ray solver using a more realistic (3-D) velocity model, reducing the L_I parameter (which affects shell width), as well as decreasing the grid cell size. Given the low probability of false detections and the good match between detected

and catalogued events reported here (Fig. 9), an improved location algorithm could lead to fully automatic methods for both detecting and locating events with high confidence.

4.3 Next steps and future research

The method employed here has demonstrated promising results, however, there are a few notable additions that could be made for improvements. In particular, future studies could investigate the applicability of using of a semisynthetic training catalogue, thus allowing this technique to only require noise recorded in the region, and a rough velocity model. Also, the utilization of distance as a feature requires additional attention. As the relative values of waveform attributes change with distance, being able to capture such observations could prove beneficial for distinguishing event classes. Furthermore, during the training of weights in the Oklahoma data set, the regression algorithm quickly found a perfect solution while only applying a few of the many features. To overcome this issue, future applications could test using smaller magnitude events during training so that the defining characteristics across the entire feature space of a given class are utilized.

As a final remark, the approximated magnitude and associated summed event probability show a strong similarity to that between cross-correlation of a template waveform and filtered ground motion data. Such similarity suggests that the association tactic applied here could also be used to assist with cross-correlation results.

5 CONCLUSIONS

The objective of this paper is to synthesize seismic attributes, or features, use them in machine learning methods that have already been proven capable of distinguishing event waveforms on a per-station basis, and utilize them for a network-wide tool for event classification and detection. This method provides a way of generalizing attributes over a desired region allowing for recognition of event types at new and previous locations.

Separate regions were selected for tests of the method during event classification and detection. For event classification, we used 13 three-component stations to classify events of pre-determined locations in Southern California. High-classification accuracy between earthquakes and blasts of over 99 per cent is reported. Network classification accuracy was consistent using two different training sets. The first set split training events randomly across the study region; the second set split training and test events spatially. Thus, this classification method is insensitive to the locations of training events, as long as the generalized feature characteristics are comparable between the two locations.

A modified application of the classification method was presented for event detection. The second portion of the study used a network of 30 three-component stations. Two windows bounding unique seismic phases were used in the association algorithm. However, the use of these features is not restricted to events that only display two unique phases. Across a week's worth of data, 98 per cent of the 1954 events detected were real. In comparison to a public catalogue tailored for the region, which reported 75 events, the magnitude of completeness dropped by over

one magnitude unit. The event detection method provided thus gives a high accuracy in the occurrence of real seismic events, as well as approximate locations. With this acceptable false alarm rate, events could be automatically reported at a lower magnitude of completeness with confidence.

Acknowledgments

This work is supported by the Natural Sciences and Engineering Research Council (Canada) and the Ontario Ministry of Research and Innovation. The scripts used to process this data were written in Python, and used the Obspy (Krischer *et al.* 2015) and TensorFlow (Abadi *et al.* 2016) libraries. We thank the United States Geological Survey, Southern California Seismic Network and Oklahoma Geological Survey for the event catalogues. The facilities of the Incorporated Research Institutions for Seismology (IRIS) Data Management Center, Northern California Earthquake Data Center, as well as the Southern California Seismic Network, were used for access to waveforms, and related metadata used in this study. IRIS Data Services are funded through the Seismological Facilities for the Advancement of Geoscience and EarthScope (SAGE) Proposal of the National Science Foundation under Cooperative Agreement EAR-1261681. We would also like to thank Kit Chambers and Gordon Kubanek for their assistance in improving this paper, as well as Anthony Lomax and an anonymous reviewer for the constructive comments.

References

Abadi, M. et al., 2016. TensorFlow: a system for large-scale machine learning, in Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Savannah, Georgia, USA, Vol. 16, pp. 265–283.

Akkar, S. & Bommer, J.J., 2010. Empirical equations for the prediction of PGA, PGV, and spectral accelerations in Europe, the Mediterranean region, and the Middle East, Seismol. Res. Lett., 81(2), 195–206.

Allen, R.V., 1978. Automatic earthquake recognition and timing from single traces, Bull. seism. Soc. Am., 68(5), 1521–1532.

Baker, T., Granat, R. & Clayton, R.W., 2005. Real-time earthquake location using Kirchhoff reconstruction, Bull. seism. Soc. Am., 95(2), 699–707.

Baillard, C., Crawford, W.C., Ballu, V., Hibert, C. & Mangeney, A., 2014. An automatic kurtosis-based P- and S-phase picker designed for local seismic networks, Bull. seism. Soc. Am., 104(1), 394–409.

Beyreuther, M., Hammer, C., Wassermann, J., Ohrnberger, M. & Megies, T., 2012. Constructing a hidden Markov model based earthquake detector: application to induced seismicity, Geophys. J. Int., 189(1), 602–610.

Bostock, M.G., Royer, A.A., Hearn, E.H. & Peacock, S.M., 2012. Low frequency earthquakes below southern Vancouver Island, Geochem. Geophys. Geosyst., 13, Q11007, doi:10.1029/2012GC004391.

Brown, J.R., Beroza, G.C. & Shelly, D.R., 2008. An autocorrelation method to detect low frequency earthquakes within tremor, Geophys. Res. Lett., 35, L16305, doi:10.1029/2008GL034560.

Cardwell, R.K. & Isacks, B.L., 1978. Geometry of the subducted lithosphere beneath the Banda Sea in eastern Indonesia from seismicity and fault plane solutions, J. geophys. Res., 83(B6), 2825–2838.

Dai, H. & MacBeth, C., 1997. The application of back-propagation neural network to automatic picking seismic arrivals from single-component recordings, J. geophys. Res., 102(B7), 15 105–15 113.

Darold, A.P., Holland, A.A., Morris, J.K. & Gibson, A.R., 2015. Oklahoma earthquake summary report 2014. Oklahoma Geol. Surv. Open-File Rept. OF1-2015

Dziewonski, A.M. & Anderson, D.L., 1981. Preliminary reference Earth model, Phys. Earth planet. Inter., 25(4), 297–356.

Ellsworth, W.L., 2013. Injection-induced earthquakes, Science, 341(6142), 1225942, doi: 10.1126/science.1225942.

Felzer, K.R., 2006. Calculating the Gutenberg-Richter b value, AGU Fall Meeting Suppl., 87(52), Abstract S42C-08.

Galiana-Merino, J.J., Rosa-Herranz, J.L. & Parolai, S., 2008. Seismic phase picking using a kurtosis-based criterion in the stationary wavelet domain, IEEE Trans. Geosci. Remote Sens., 46(11), 3815–3826.

Gentili, S. & Michelini, A., 2006. Automatic picking of P and S phases using a neural tree, J. Seismol., 10(1), 39–63.

Gibbons, S.J. & Ringdal, F., 2006. The detection of low magnitude seismic events using array-based waveform correlation, Geophys. J. Int., 165(1), 149–166.

Grigoli, F., Cesca, S., Krieger, L., Kriegerowski, M., Gammaldi, S., Horalek, J. & Dahm, T., 2016. Automated microseismic event location using Master-Event Waveform Stacking, Sci. Rep., 6, doi:10.1038/srep25744.

Hosmer, D.W., Jr, Lemeshow, S. & Sturdivant, R.X., 2013. Applied Logistic Regression, Vol. 398, John Wiley & Sons.

Kaur, K., Wadhwa, M. & Park, E.K., 2013. Detection and identification of seismic P-waves using Artificial Neural Networks, in The 2013 International Joint Conference on Neural Networks (IJCNN, Dallas, TX, pp. 1–6, IEEE.

Keranen, K.M., Savage, H.M., Abers, G.A. & Cochran, E.S., 2013. Potentially induced earthquakes in Oklahoma, USA: links between wastewater injection and the 2011 Mw 5.7 earthquake sequence, Geology, 41(6), 699–702.

Kværna, T., Ringdal, F. & Baadshaug, U., 2007. North Korea's Nuclear Test: the capability for seismic monitoring of the North Korean test site, Seismol. Res. Lett., 78(5), 487–497. Event classification and detection with machine learning 1409

Krischer, L., Megies, T., Barsch, R., Beyreuther, M., Lecocq, T., Caudron, C. & Wassermann, J., 2015. ObsPy: a bridge for seismology into the scientific Python ecosystem,Comput. Sci. Discov., 8(1), 014003, doi:10.1088/1749- 4699/8/1/014003.

Lomax, A., Satriano, C. & Vassallo, M., 2012. Automatic picker developments and optimization: FilterPicker—a robust, broadband picker for real-time seismic monitoring and earthquake early warning, Seismol. Res. Lett., 83(3), 531–540.

Madureira, G. & Ruano, A.E., 2009. A neural network seismic detector, IFAC Proc., 42(19), 304–309.

Mousavi, S.M., Horton, S.P., Langston, C.A. & Samei, B., 2016. Seismic features and automatic discrimination of deep and shallow induced microearthquakes using neural network and logistic regression, Geophys. J. Int., 207(1), 29–46.

NCEDC Blog: News from the Northern California Earthquake Data Center, 2013. Northern California Earthquake Data Center. Available at: http://ncedc.org/blog/ncedcblog.php/ncss-event-review-threshold -change last accessed 1 December 2016.

Poiata, N., Satriano, C., Vilotte, J.P., Bernard, P. & Obara, K., 2016. Multiband array detection and location of seismic sources recorded by dense seismic networks, Geophys. J. Int., 205(3), 1548–1573.

Rabin, N., Bregman, Y., Lindenbaum, O., Ben-Horin, Y. & Averbuch, A., 2016. Earthquake-explosion discrimination using diffusion maps, Geophys. J. Int., 207(3), 1484–1492.

Riggelsen, C. & Ohrnberger, M., 2014. A machine learning approach for improving the detection capabilities at 3C seismic stations, Pure appl. Geophys., 171(3–5), 395–411.

Saragiotis, C.D., Hadjileontiadis, L.J. & Panas, S.M., 2002. PAI-S/K: a robust automatic seismic P phase arrival identification scheme, IEEE Trans. Geosci. Remote Sens., 40(6), 1395–1404.

Sharma, B.K., Kumar, A. & Murthy, V.M., 2010. Evaluation of seismic events detection algorithms, J. geol. Soc. India, 75(3), 533–538.

Shelly, D.R., Beroza, G.C. & Ide, S., 2007. Non-volcanic tremor and low frequency earthquake swarms, Nature, 446(7133), 305–307.

Vallejos, J.A. & McKinnon, S.D., 2013. Logistic regression and neural network classification of seismic records, Int. J. Rock Mech. Min. Sci., 62, 86–95.

Wang, J. & Teng, T.L., 1995. Artificial neural network-based seismic detector, Bull. seism. Soc. Am., 85(1), 308–319.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Yenier, E. & Atkinson, G.M., 2015. Regionally adjustable generic ground motion prediction equation based on equivalent point-source simulations: application to central and eastern North America, Bull. seism. Soc. Am., 105, 1989–2009.

Yoon, C.E., O'Reilly, O., Bergen, K.J. & Beroza, G.C., 2015. Earthquake detection through computationally efficient similarity search, Sci. Adv., 1(11), e1501057, doi:10.1126/sciadv.1501057.

Zhao, Y. & Takano, K., 1999. An artificial neural network approach for broadband seismic phase picking, Bull. seism. Soc. Am., 89(3), 670–680.

Supporting Information

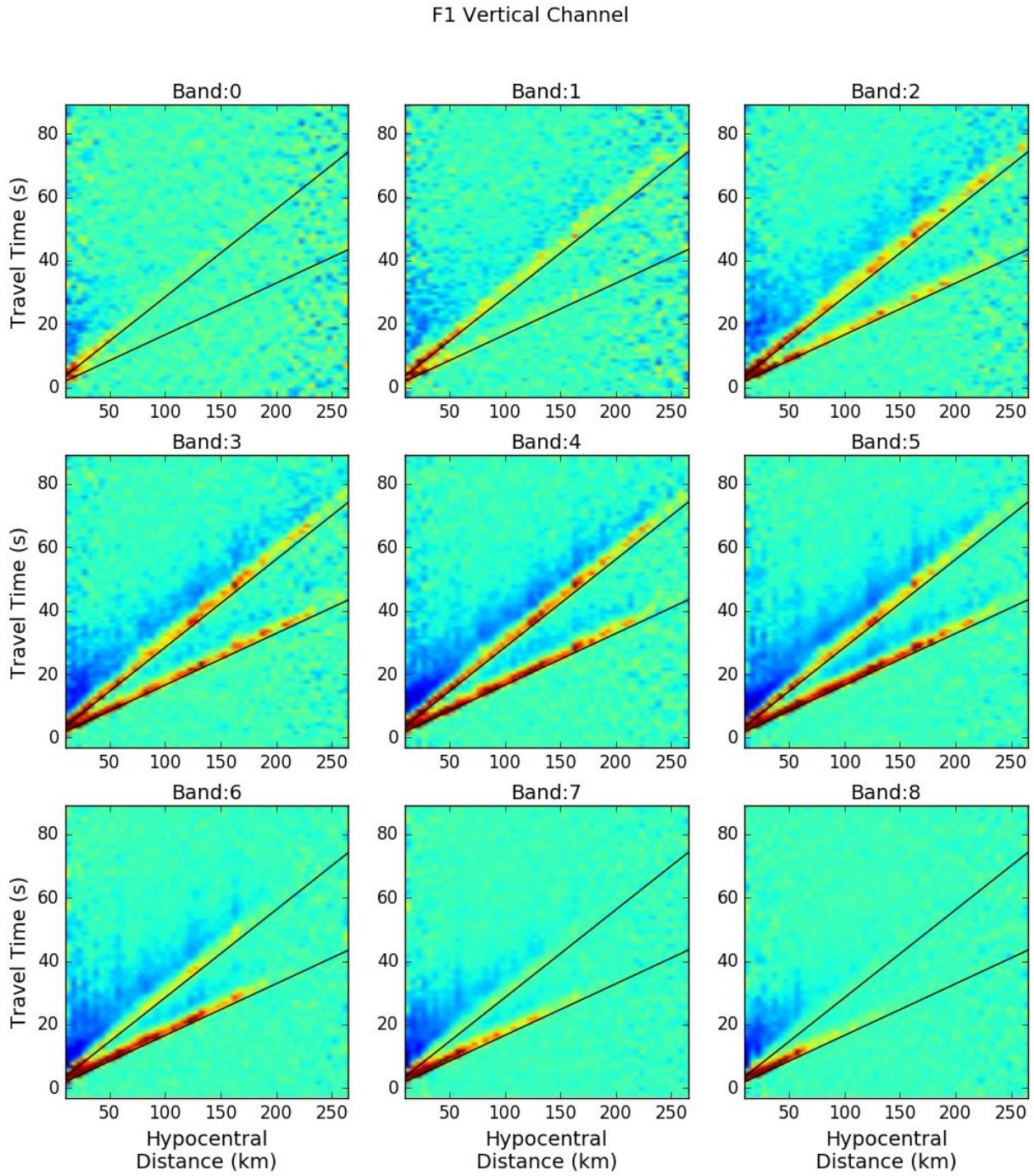


Figure S1: F_1 values extracted from vertical channels for earthquakes, applied during event classification. Figure format is the same as Fig. 4 (top).

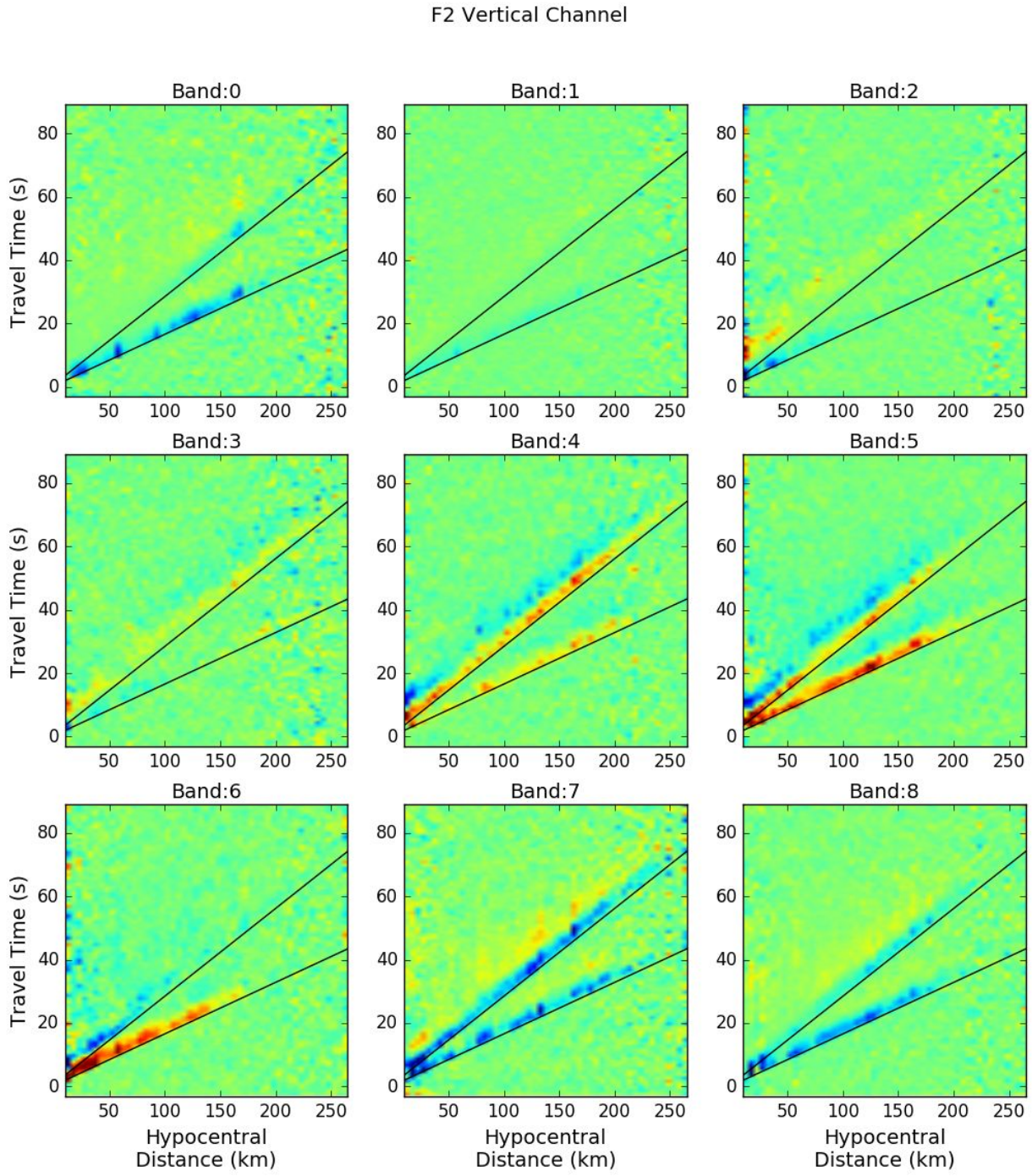


Figure S2: F_2 values extracted from vertical channels for earthquakes, applied during event classification. Figure format is the same as Fig. 4 (bottom).

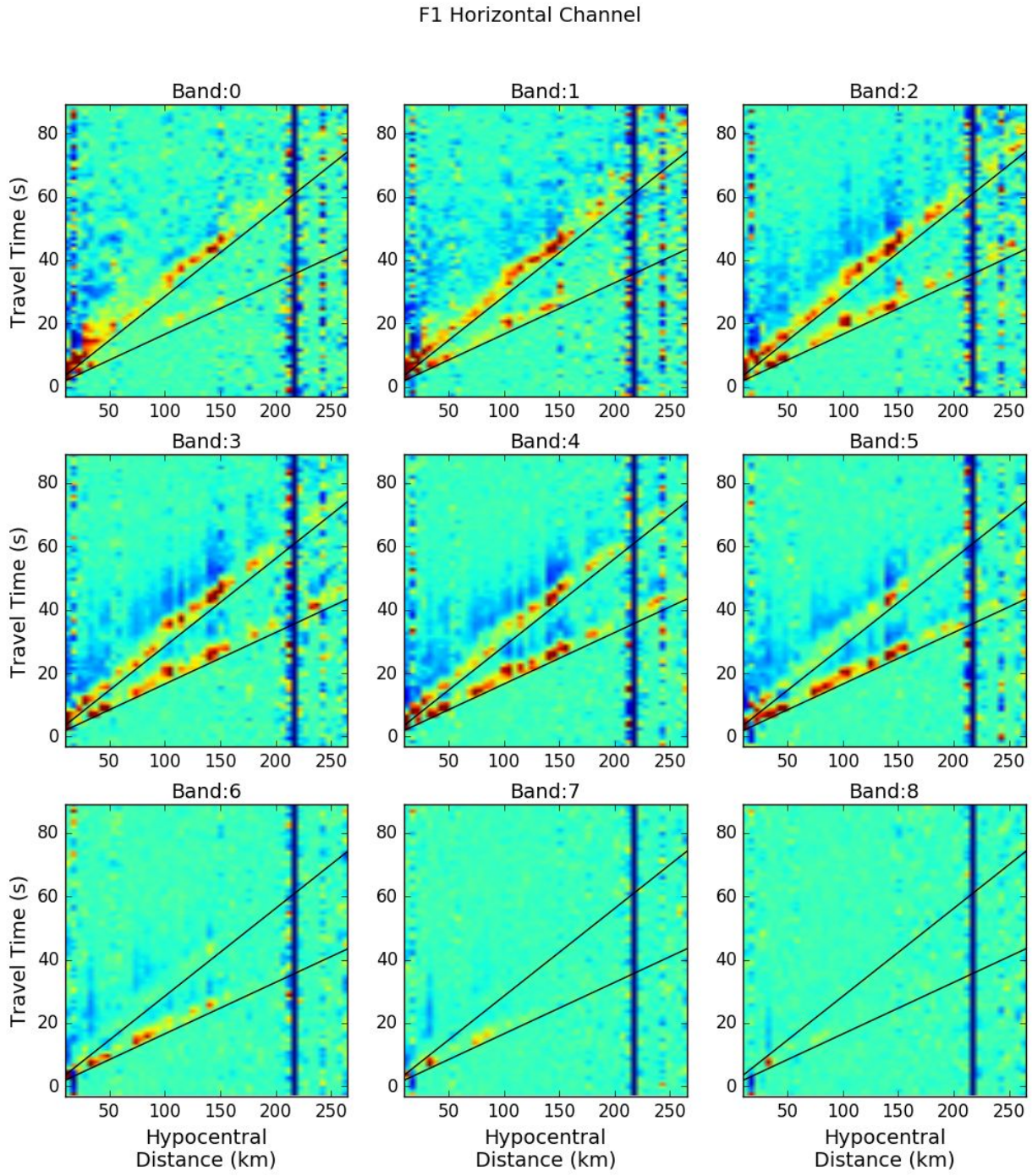


Figure S3: F_1 values extracted from horizontal channels for explosions, applied during event classification. Artefacts (seen as vertical stripes) occur due to a lack of observations at the given event-station distance. Figure format is the same as Fig. 4 (top).

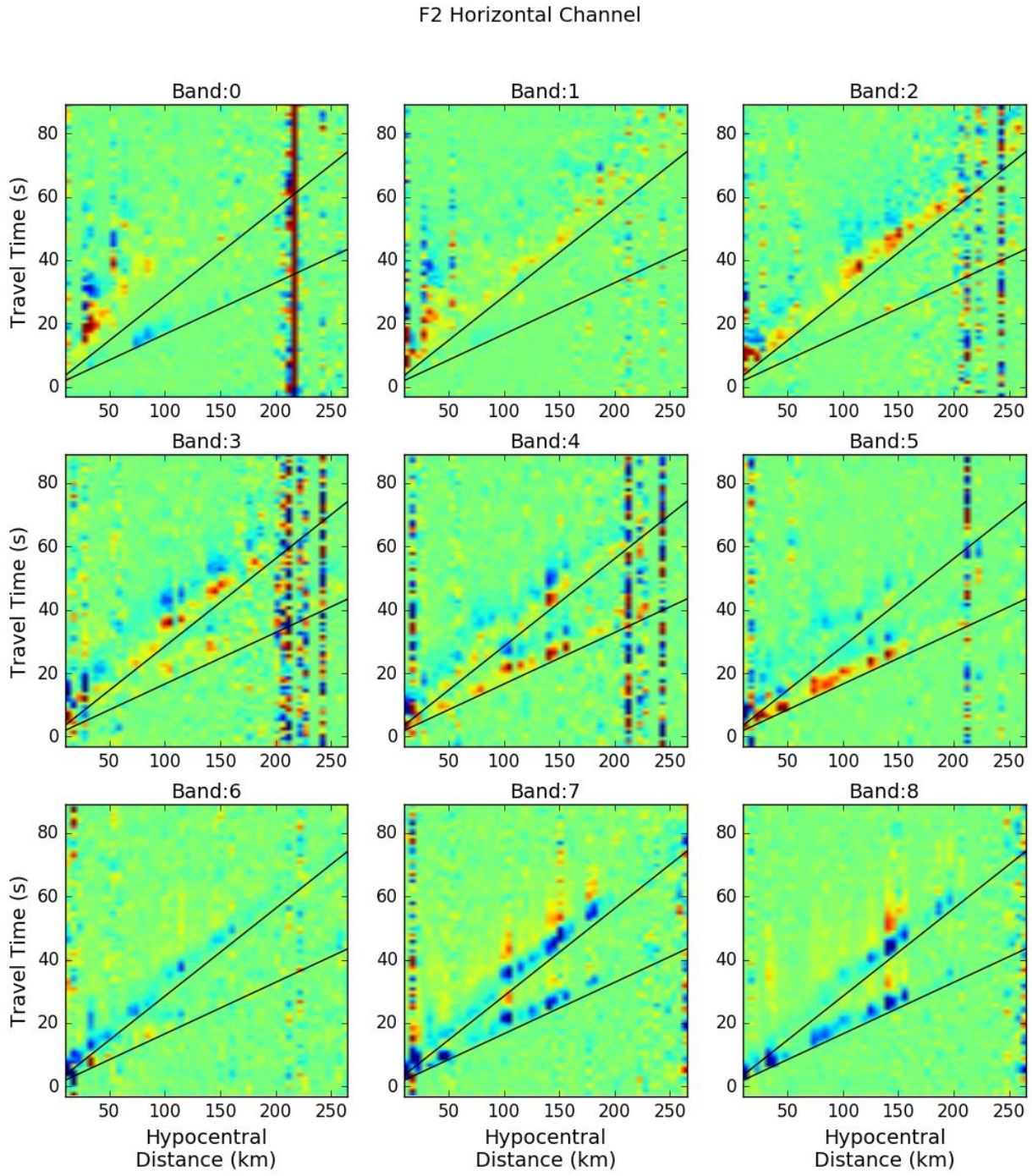


Figure S4: F_2 values extracted from horizontal channels for explosions, applied during event classification. Artefacts (seen as vertical stripes) occur due to a lack of observations at the given event-station distance. Figure format is the same as Fig. 4 (bottom).

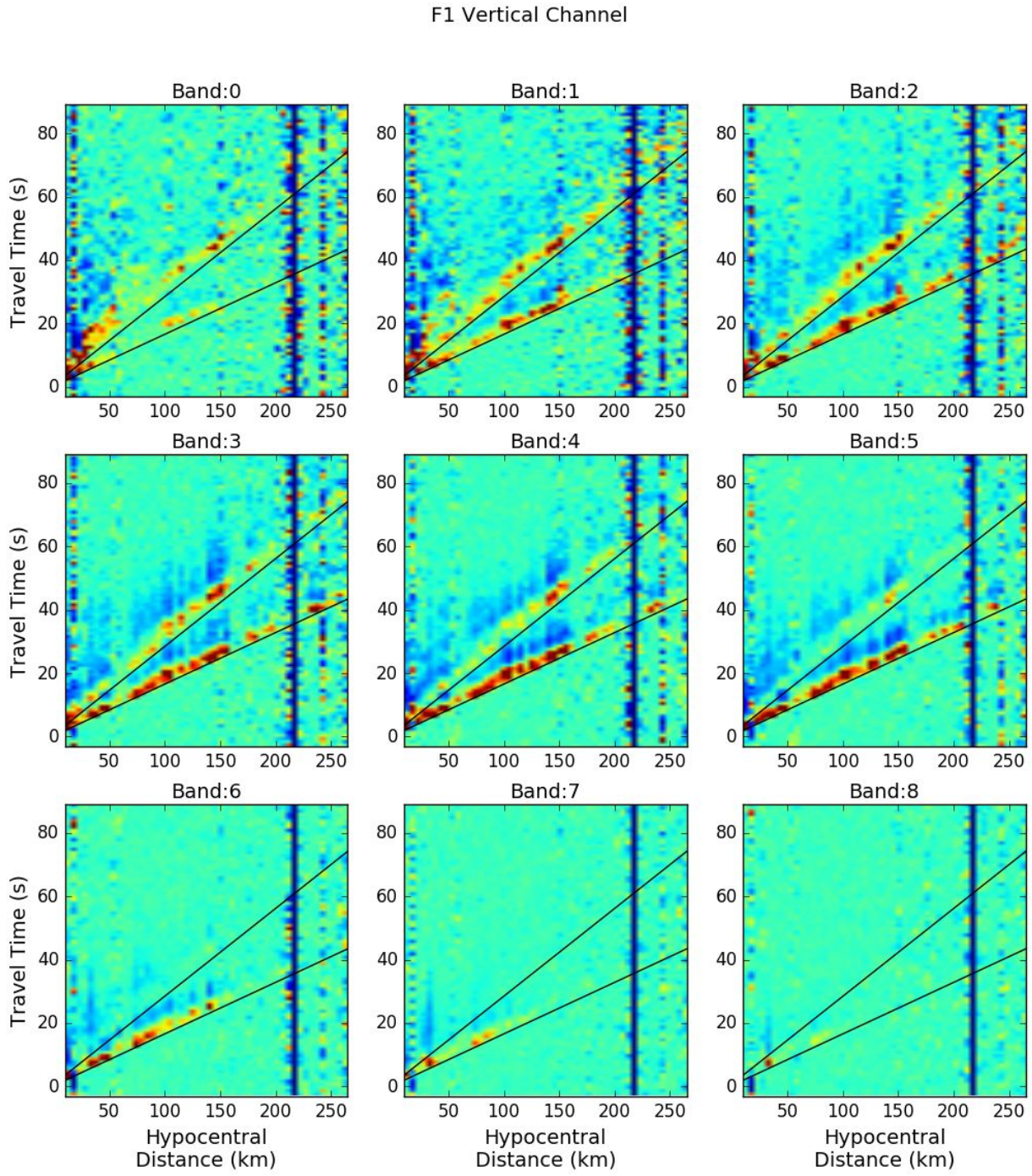


Figure S5: F_1 values extracted from vertical channels for explosions, applied during event classification. Artefacts (seen as vertical stripes) occur due to a lack of observations at the given event-station distance. Figure format is the same as Fig. 4 (top).

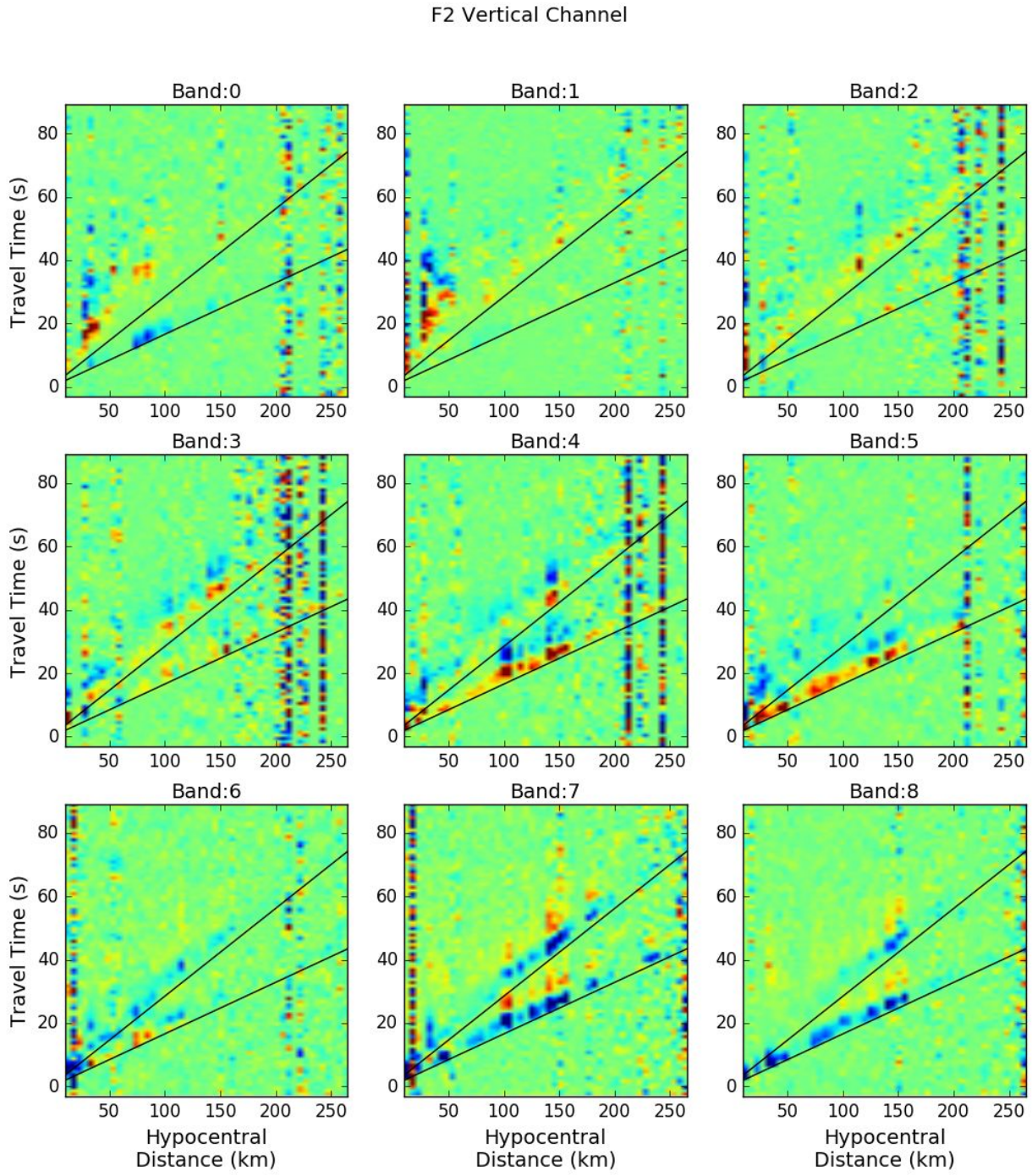


Figure S6: F_2 values extracted from vertical channels for explosions, applied during event classification. Artefacts (seen as vertical stripes) occur due to a lack of observations at the given event-station distance. Figure format is the same as Fig. 4 (bottom).

Logistic Regression: A Brief Explanation

Logistic regression is a method that searches for the optimal set of weights ($W \in \mathbb{R}^N$) and bias term ($B \in \mathbb{R}^1$) that best match a set of features ($in \in \mathbb{R}^N$) to one of two categories ($C \in 0, 1$). This takes the functional form:

$$\text{Activation Function}(W \cdot X + B) = C \qquad (12)$$

where the activation function serves to normalize the dot product $W \cdot X$ between 0 and 1. For an example where the features are in $\mathbb{R}^2$, the classes may be best split with the linear boundary shown in Fig. S7.

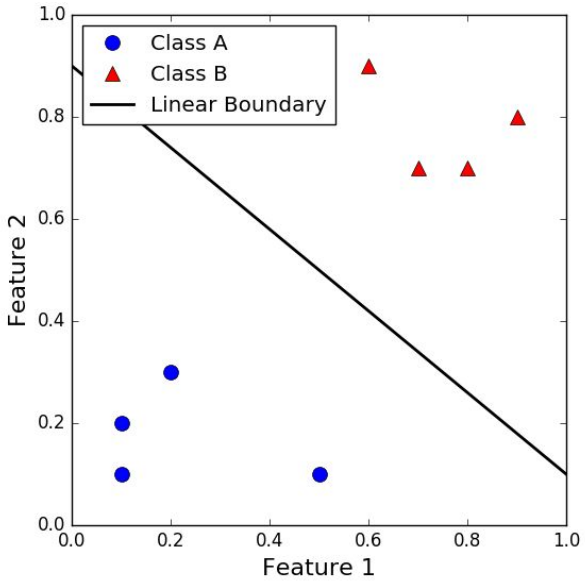


Figure S7: Example optimal boundary splitting classes using logistic regression.

Each of the points (observations) shown have an amplitude associated with it (relating to its class; either 0 or 1) which brings the point in/out of the page. Taking a cross section perpendicular to the boundary, we see how the model fits the observations (Fig. S8).

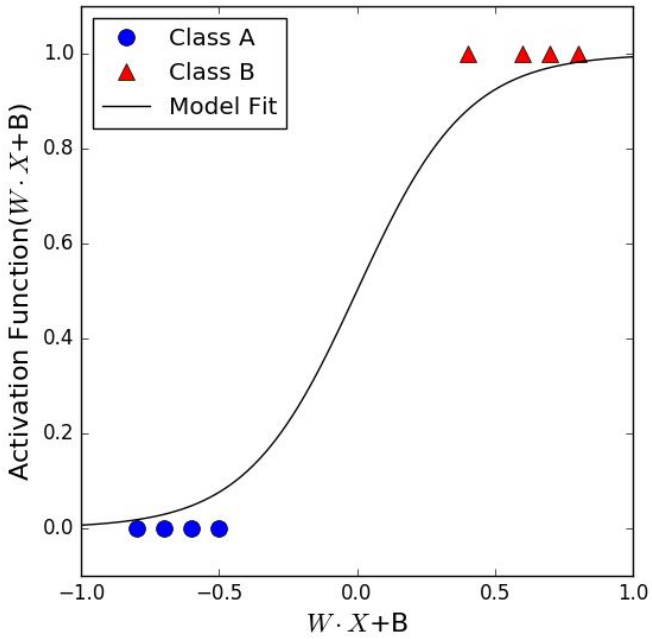


Figure S8: Example model fit to observations using logistic regression.

Logistic regression can be used to fit multiple classes by using a one-versus all method, where all of the amplitude values are set to 0 for all but the class in question which is given an amplitude of 1. A model is then determined for each class.