

Spring 2018

DISTINGUISHING EARTHQUAKES AND NOISE USING RANDOM FOREST ALGORITHM

Nishita Narvekar
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects

Part of the [Computer Sciences Commons](#)

Recommended Citation

Narvekar, Nishita, "DISTINGUISHING EARTHQUAKES AND NOISE USING RANDOM FOREST ALGORITHM" (2018).
Master's Projects. 606.

DOI: <https://doi.org/10.31979/etd.6kxb-c9tu>

https://scholarworks.sjsu.edu/etd_projects/606

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

**DISTINGUISHING EARTHQUAKES AND NOISE USING RANDOM FOREST
ALGORITHM**

A Project Report

Presented to

Department of Computer Science

San José State University

In Partial Fulfillment

Of the Requirements for the Class

CS 298

Author

Nishita Narvekar

May 2018

Advisor

Dr. Philip Heller

Acknowledgements

I would like to sincerely thank my project advisor, Dr. Philip Heller, for his continuous help, encouragement and time. I would also like to thank my committee members, Dr. Thomas Austin and Professor James Casaletto, for their feedback and suggestions. I would also like to thank my family and friends for their unwavering support.

Abstract

Earthquakes are a major cause of life and property destruction. It is known that earthquakes radiate energy in the form of surface and body seismic waves. P-wave and S-waves are types of body waves. Both waves can be detected and recorded at an earthquake station. These waves can be analyzed to detect earthquakes. Most of the earthquake prediction techniques today are a combination of geophysics and signal processing, which are relatively complex. Machine learning can be used to learn the behavior of seismic waves and help in early detection. Machine learning can also be employed to process massive amounts of raw seismic data. The goal of this project is to distinguish between earthquakes and noise. Recordings of seismic waves from earthquake stations contain significant noise, for example from mining explosions or surface vibrations caused by vehicle traffic. It is necessary to distinguish between noise and actual earthquake signals. In this project machine learning classification techniques will be used for this purpose.

Keywords: Earthquake, Phase classification, Seismic waveform, Machine learning, Fast Fourier Transform, Support Vector Machine algorithm, Random Forest algorithm

Table of Contents

1. Background	8
1.1 Types of seismic waves	9
1.2 ObsPy	10
1.3 Feature extraction and data pre-processing	13
1.4 Machine learning algorithms.....	14
1.5 Data Preprocessing: Noise Removal	16
1.6 Challenges associated with distinguishing earthquakes and non-earthquakes.....	17
2. Data	19
2.1 Data Collection	19
2.2 Clipping the data.....	20
2.3 Data visualization.....	23
3. Methods	31
3.1 Experiment 1	31
3.2 Experiment 2	31
3.3 Experiment 3	32
3.4 Experiment 4	32
3.5 Experiment 5	33
3.6 Experiment 6	34
4. Results	35
4.1 Experiment 1	35
4.2 Experiment 2	35
4.3 Experiment 3	40
4.4 Experiment 4	41
4.5 Experiment 5	43
Experiment 6.....	45
4.6.....	45
5. Discussion	47
6. References	50

List of Figures

Figure 1 OBSPY read function output.....	10
Figure 2 OBSPY header file.....	11
Figure 3 OBSPY plot function	12
Figure 4 OBSPY slice function	12
Figure 5 OBSPY max function.....	12
Figure 6 Main shock, aftershock and foreshock in the same seismogram [36]	20
Figure 7 P-wave and S-wave arrival time for Northern California earthquake.....	21
Figure 8 Recursive STA/LTA plot from OBSPY	22
Figure 9 Earthquake plot a	23
Figure 10 Earthquake plot b	23
Figure 11 Earthquake plot c	24
Figure 12 Earthquake plot d	24
Figure 13 Noise plot a.....	24
Figure 14 Noise plot b.....	25
Figure 15 Noise plot c.....	25
Figure 16 Noise plot d.....	25
Figure 17 Sound Signal plot a	26
Figure 18 Sound signal plot b.....	26
Figure 19 Sound signal plot c.....	26
Figure 20 Sound signal plot d.....	27
Figure 21 Scatter plot of amplitudes of all sampling points a.....	28
Figure 22 Scatter plot of amplitudes of all sampling points b	29

Figure 23 Amplitudes of all sampling points	29
Figure 24 Experiment 1 Cross Validation results	35
Figure 25 Experiment 2 cross validation results	36
Figure 26 Experiment 2 precision-recall plot.....	36
Figure 27 Experiment 2 overfitted decision tree	38
Figure 28 Experiment 2 overfitted decision tree	38
Figure 29 Experiment 2 decision tree	39
Figure 30 Experiment 3 cross validation graph.....	40
Figure 31 Earthquake and non-earthquake lengths scatter plot	41
Figure 32 Experiment 4 cross validation results	42
Figure 33 Experiment 4 precision-recall curve	42
Figure 34 FFT of seismic noise	43
Figure 35 Experiment 5 cross-validation graph	44
Figure 36 Experiment 5 precision-recall curve	45
Figure 37 Experiment 6 cross-validation time series accuracy	46
Figure 38 Experiment 6 random forest output	46
Figure 39 Experiment 6 random forest output	46
Figure 40 Random Forest model result comparison.....	48
Figure 41 Confusion matrix experiment 4 vs experiment5.....	48

List of Tables

Table 1 Experiment 2 confusion matrix.....	37
Table 2 Experiment 2 GridSearchSV output.....	37
Table 3 Experiment 2 GridSearchCV output	40
Table 4 Experiment 3 confusion matrix.....	41
Table 5 Experiment 4 confusion matrix.....	43
Table 6 Experiment 5 confusion matrix.....	45

1. Background

Earthquakes have been a major cause of destruction all around the world. These natural calamities come with little or no warning at all. There was a total of 10,341 earthquakes all over the world in 2017 [9]. The United States Geological Survey (USGS) collects all the earthquake data. Machine learning can be used to identify earthquake events and extract hidden information. An early warning framework for detecting events like an earthquake can be built using machine learning

An earthquake is characterized by shaking of the ground because of the sudden release of energy from the earth's lithosphere. These ground motion waves can be recorded and be studied later. If we are able to predict earthquakes a few minutes in advance and send out alerts it could save many lives and mitigate the damage by the earthquakes. A few minutes of warning could enable people to take precautions such as get to a safe place, take cover or switch off electricity.

Earthquake early warning systems use earthquake science and sensor monitoring technology to warn people when an earthquake is going to hit their location. Since 2006, The United States Geological Survey has been working to develop such an early warning system, called ShakeAlert for United States [22]. ShakeAlert is due to start sending out earthquake warning messages to the public in 2018. ShakeAlert detects the P-wave and estimates the magnitude and location of the incoming earthquake [21].

Earthquake waves or seismic waves are recorded by an instrument known as a seismograph. A Seismograph consists of a heavy mass attached to a fixed base. When the seismic waves reach the Seismograph the fixed base moves, but the heavy mass remains stationary because of inertia. This motion of the base with respect to the heavy mass can be translated to ground

motion and recorded as seismic waves. Now the Seismographs are much more modern but the basic idea behind them remains the same [9].

A seismogram is a record of ground motion as a function of time. A seismogram records motion in three directions (BHZ, BHN, and BHE). Channel BHZ indicates that the signal was recorded perpendicular to the Earth's surface, BHN is along the Y-axis and BHE is along the X-axis. The motion recorded in a seismogram may result from an earthquake or an explosion. Seismograms records the tiniest of motions. These motions could be a heavy vehicle on the road, waves hitting a shore or explosions near a mine [17]. Our focus is to classify signals into earthquake and seismic noise [11].

Earthquake stations record continuously. A twenty-four hour long recording may consist of multiple earthquakes and seismic noise or just seismic noise. This project can also be useful while clipping seismic data. Machine learning classification algorithms will be used to distinguish between earthquakes and seismic noise. Machine-learning analysis of seismic data involves data preprocessing, training feature identification, and algorithm selection.

1.1 Types of seismic waves

There are two types of seismic waves: body waves and surface waves. Body waves move through the earth's crust whereas surface waves only move near the earth's surface like ripples in water. There are two kinds of body waves – P waves and S waves. P wave is also known as the compressional wave or the primary wave. It travels faster than the S wave. Hence the P wave is recorded first at the seismograph [4]. The motion of the P wave is similar to sound waves. It pushes and pulls the rocks it is traveling through, in the direction of travel. The P waves can travel through solids as well as liquids. The S wave is the secondary wave or the transversal wave. The S wave

travels slower than the P wave. The S wave only travels through solids. As the S wave only pass through solids, seismologists concluded that the earth's outer core is actually liquid [4]. The S wave moves the particles of the earth in an up and down motion. The S wave is more destructive than the P wave. As the seismic waves travel through the earth's volume away from the epicenter of the earthquake, the distance between the P wave and S wave increases. This time difference between the P and S wave will be the warning time that can be given to citizens [22].

1.2 ObsPy

Obspy is an open source python framework for manipulating and deriving results from seismic data [6]. Obspy can be used for reading or writing seismic files in various formats. Obspy has predefined functions for algorithms used in seismology like STA/LTA, trigger picker algorithms, bandpass filter etc. In this project Obspy was used slicing, reading waveforms and extracting features from the seismic waves. Obspy is a python library [6]. It is a very handy tool for seismic analysis. The following Obspy functions were used in this project:

```
tr = read('/Users/Nishita/Documents/cs298/data/newearthquake/IU.WVT.00.BH1.M.2014.110.193125.SAC')
print(tr)

1 Trace(s) in Stream:
IU.WVT.00.BH1 | 2014-04-20T19:31:25.019538Z - 2014-04-20T19:39:53.969538Z | 20.0 Hz, 10180 samples
```

Figure 1 OBSPY read function output

1. Reading the header file – The header file has some very useful information that can be used as features. A typical SAC header file has 30-40 fields which has information regarding the climatic conditions near the station, the rock properties and how the signal was recorded. Following header values were used as features in the project [28].
 - a. NPTS – Number of sampling points
 - b. DELTA – Increment between each sampling data point


```
tr.plot()
```

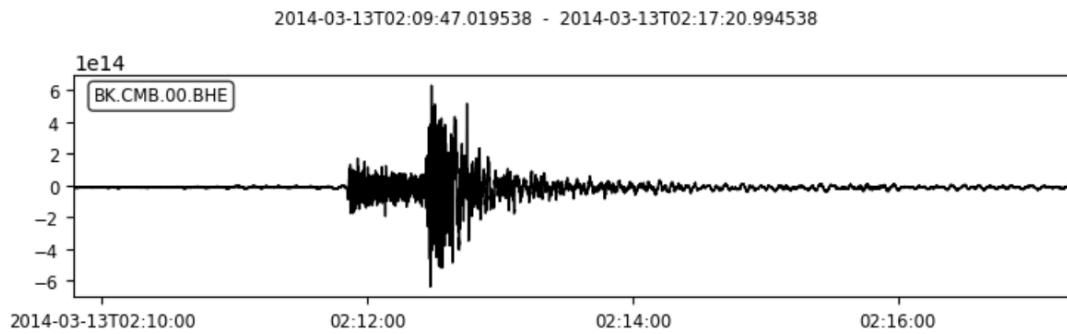


Figure 3 OBSPY plot function

3. Slice – Slice returns a new stream object that is cut to the given start and end time [30].

```
dt=tr[0].stats.starttime
```

```
st = tr.slice(dt, dt + 100)
```

```
st.plot()
```

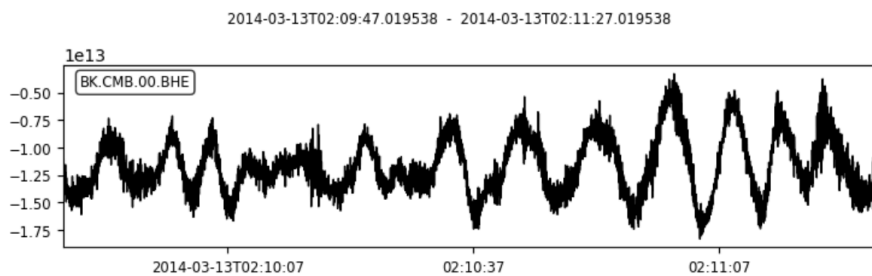


Figure 4 OBSPY slice function

4. Max – Max returns the absolute maximum amplitude in the trace [31]

```
tr.max()
```

```
[-146907.0]
```

Figure 5 OBSPY max function

1.3 Feature extraction and data pre-processing

Seismic data is in the form of waveforms. The waveform can be used as it is or a number of features can be extracted from these waveforms to train machine learning algorithm. A number of feature selection techniques were discussed in the papers.

In the paper by J. Ramirez et al [8] multi-scale analysis and spherical feature manifold is used for feature extraction. Multiscale polarization technique is used to analyze the signal at different frequency ranges. Singular value decomposition (SVD) is then applied to these signals. In spherical feature manifold, the multi-scale polarization feature extraction technique is applied to each data matrix. In this way, the coefficients of each wavelet are decomposed. These decomposed coefficients and the seismic data matrix obtained from the SVD process are used as features [8]. O. Lindenbaum et al [9] uses Fourier Transforms for feature extraction. G. Zhao et al. extracted time domain features, wavelet features and features directly calculated from the waveform. Some of the time domain features used were standard deviation, mean, median etc. Wavelet features included the energy ratios, Shannon entropy etc. Other features that were used in the study were the amplitude of the P-wave arrival, the maximum amplitude of the S-wave group etc [10]. Takase et al. used the Electromagnetic Wave(EM) data and decided the frame period to be one hour and extracted 24 frames per day. They extracted the cepstrum of the EM wave and passed it to the ML algorithm [11]. Cepstrum is the inverse Fourier Transform of the logarithm of the estimated spectrum of a signal [32]. Urata et al [12] also analyzes the electromagnetic waves. The EM waves are averaged over 6 seconds and 150 seconds and fed to a data logger. The data

logger provides the numerical and waveform data which is fed to the machine learning algorithm. Astuti et al [13] uses Singular Value Decomposition (SVD) for feature extraction. The signal is projected on the Eigen vectors of the Linear predictive coding (LPC) filter. The LPC coefficients are estimated using the Levinson–Durbin algorithm. The mathematical equation is then reduced to the SVD form and features are selected accordingly .

E. Ruano et al [14] uses the Power Spectrum Density (PSD) technique for feature extraction. They have also used band-pass filter and STA/LTA ratios for roughly detecting seismic events. N. Orlic et al [15] used ML to distinguish between earthquakes and explosions using genetic algorithm-based boosting approach. They use a genetic algorithm to determine optimal discriminating features. After multiple re-runs and boosting the features are selected. S. Gentili et al [16] also use filtering techniques to remove noisy signals at lower frequencies. They have designed P and S extractor modules for labeling P and S-wave arrival times. The extractor module outputs the onset time with an estimate of its accuracy. The rough picks for P and S waves are compared with the output of the neural network to determine the final P and S arrival times .

The papers have discussed many ways of feature extraction and data preprocessing. Both these steps are very important as we don't have ready-made features. The data that we have is in SAC binary format. Each file has a header section which has more information about the waveform. We can use some of the techniques discussed above for feature extraction and data preprocessing.

1.4 Machine learning algorithms

After the data is ready the next step is to identify a machine learning algorithm for classifying earthquakes and noise. Traditionally a combination of geophysics and statistical

methods were used to analyze earthquakes. But given the huge dataset of historical data, it makes much more sense to use machine learning algorithms [26].

J. Ramirez et al [8] use kernel ridge regression for classifying seismic phases into Pn-phase and Lg-phase with an accuracy of 71% on an average using cross-validation. O. Lindenbaum et al [9] use K-NN for classifying earthquakes and man-made explosions. They tested against 62 man-made explosions and were able to detect 85% of the explosions correctly. G. Zhao et al. [10] also classifies earthquakes and explosions. The paper compares three classification techniques Back-propagation neural networks, support vector machine and BP-AdaBoost classifier. They achieved very good results with BP- AdaBoost classifier, selecting BP-NN as a weak classifier. They discuss results by splitting their dataset into multiple variations. They also give their results in terms of precision, recall and accuracy.

S. Urata et al. [12] use HMMs for detection of anomalous signals. Parameter estimation for the HMM model is calculated by the Baum-Welch algorithm. The average precision is calculated using the Viterbi algorithm. The output of a trained HMM is a high acceptance probability for a normal pattern and a low acceptance pattern for an anomalous pattern. W. Astuti et al [13] proposed a machine learning technique to predict the magnitude, location and time for the next earthquake. They combined Singular Value Decomposition and Support Vector Machine.

A.E. Ruano et al. [14] proposed support vector machines for detecting earthquakes. The SVM model was tested on 226 earthquakes and 309 explosions. The SVM classifier incorrectly classified 71 events. The detection time was 110 seconds.

N. Orlic et al. [15] formulated a novel genetic algorithm for discriminating between earthquake and explosions. The proposed genetic algorithm automatically searches for discriminating features from the seismogram. A boosting method was used to improve the model's

performance for searching the best features. After the feature set is decided from the training dataset, the same feature set can be used on unknown data. The performance of this model was tested on limited data of 80 seismograms and achieved an accuracy of 85%.

S. Gentili et al. [16] propose an automatic P-wave and S-wave arrival time detection method based on neural networks. The network for both P-wave and S-wave were simple perceptrons, without children nodes. The classification results were interpreted as follows- the neuron which has the higher output of the network is called as the winning neuron.

Machine learning models can very effectively learn the seismic patterns of earthquakes. Once a model is trained it can be used to distinguish between earthquake and noise. Machine learning algorithms can be used for predicting the P-wave and S-wave arrival times as well.

1.5 Data Preprocessing: Noise Removal

The seismic signals recorded at the earthquake stations are often mixed with noise. It is necessary to remove noise before the data is fed to the machine learning algorithm as the model needs to understand the earthquake pattern without noise. The noise can be introduced by instruments, external disturbances such as explosions, mining activities etc. These noises can be removed by either identifying patterns of noise or passing the signal through a filter. Many noise removal techniques are discussed in the literature.

J. Ramirez et al. [8] rely on their windowing algorithm to eliminate maximum noise. On the other hand, O. Lindenbaum et al. [19] rely on bandpass filter to remove amplitudes which are above a certain threshold. Windowing techniques can be used to clip data which has an amplitude above a certain threshold and then pattern recognition can be used to identify the noise. Detrend filter can be used to remove a trend from the trace. Overall trends that are not similar to the data

can be removed with the help of the detrend filter. It is typically used to remove noise introduced by recording instruments. The bandpass filter can be used to filter out noise that is above or below a certain threshold. The filtering techniques are not a full proof way to remove noise, but they can be combined with other machine learning techniques to remove noise.

The literature discusses many ways of feature extraction, training a model and noise removal. Machine learning for earthquake prediction is a novel technique. Earthquakes are very unpredictable natural calamities. It is very difficult to predict an earthquake even a few minutes before it actually happens. If we are able to predict earthquakes much in advance it will avoid a lot of destruction. Machine learning seems to be a step in the right direction.

Predicting earthquakes will require historical data and a tailored machine learning approach. As seen in the literature a number of machine learning techniques can be applied to predict earthquakes, but the main question is whether it will perform well. The performance of any machine learning algorithm depends on the quality of training data, feature extraction methods and the machine learning algorithm. Traditionally statistical methods in geophysics were applied to seismic data to study earthquakes. In the age of artificial intelligence, it makes sense to process these large number of seismic signals using machine learning. I think the future of earthquake physics will rely heavily on machine learning to process massive amounts of raw seismic data.

1.6 Challenges associated with distinguishing earthquakes and non-earthquakes

There have been many studies in exploring earthquakes, their causes and why they occur. Traditionally a combination of geophysics and signal processing has been used for studying earthquakes. Using machine learning to identify earthquake patterns is relatively new and has not

been explored much. Dr. Daniel McNamara, a seismologist, said that seismology is a very complicated science and is still not well developed [24]. For example, Japan experiences 11% of the world's earthquakes and it has invested considerably to develop an early warning system for earthquake, in spite of all these efforts the 2011 Japan earthquake could not be predicted in advance [25].

Earthquakes and explosions both release energy, their signal pattern looks very similar [23]. Machine learning algorithms can be used to identify patterns of P-wave and S-wave [8][9][10][12][13][14]. Once trained the machine learning algorithm can be used to predict the P-wave and S-wave arrival time. One of the major problems is that the data all around the world for earthquakes is not in a single format. Moreover, the seismograms recorded on a particular station is dependent on the geographical conditions around it [41]. Such variations in data will not be helpful for training the algorithms. Also, there is a problem of false alarms which are events that are predicted as earthquakes but are due to some other reasons such as noise or explosions. A major problem that would need attention is distinguishing an earthquake from seismic noise.

An earthquake monitoring tool should be able to distinguish between noise and earthquake in real time. The tool should be able to identify the source of the signal. Such a distinction cannot be made manually. Machine learning algorithms can mine hidden information and distinguish an earthquake from noise. Machine learning can be used to build fully automatic tools [23]. The model can be improved by continuous training. Earthquake stations generate massive amounts of data. The data is too large for comprehensive analysis. Machine learning is ideal for such a scenario [26].

2. Data

2.1 Data Collection

The earthquake signals were collected from the Incorporated Research Institutions for Seismology (IRIS) [33]. One hundred and twenty US universities contribute and operate IRIS [34]. IRIS facilitates data acquisition, data management and distribution of seismological data. Generally, when you get the raw data it has both earthquakes and just plain recordings with some noise. It is imperative to extract the earthquake from the recording. There could be multiple earthquakes or aftershocks within 24 hrs. All these earthquakes have to be identified separately. IRIS Wilber 3 is web interface to look up seismic events and request data. It allows the user to select the type of event, its intensity and the neighboring earthquake stations. IRIS Wilber 3 is interactive and easy to use as its GUI is self-explanatory. Events can be selected in the following manner [33] –

1. Find seismic events using IRIS Wilber 3
2. Enter dates of the event you are interested in
3. Enter a range of magnitudes
4. Select location from the map
5. For requesting the data, you can specify the networks, channels, distance range, Azimuth Range, start time and end time with respect to S arrival time and P arrival time

For this project events with magnitude greater than 4 were selected. Data was collected from stations within a distance range of 8. If an earthquake station is closer to epicenter it will record

the maximum magnitude and intensity of the earthquake [35]. For the seismic noise training data, signals from earthquakes stations was collected from those days when there was no earthquake..

2.2 Clipping the data

Aftershocks occur within a few minutes of the actual earthquake. Clipping can be used to separate and identify different earthquakes. Clipping can also be used for separating earthquake from noise [36]. In the figure below, we can see the main shock, aftershock and foreshock in the same seismogram.

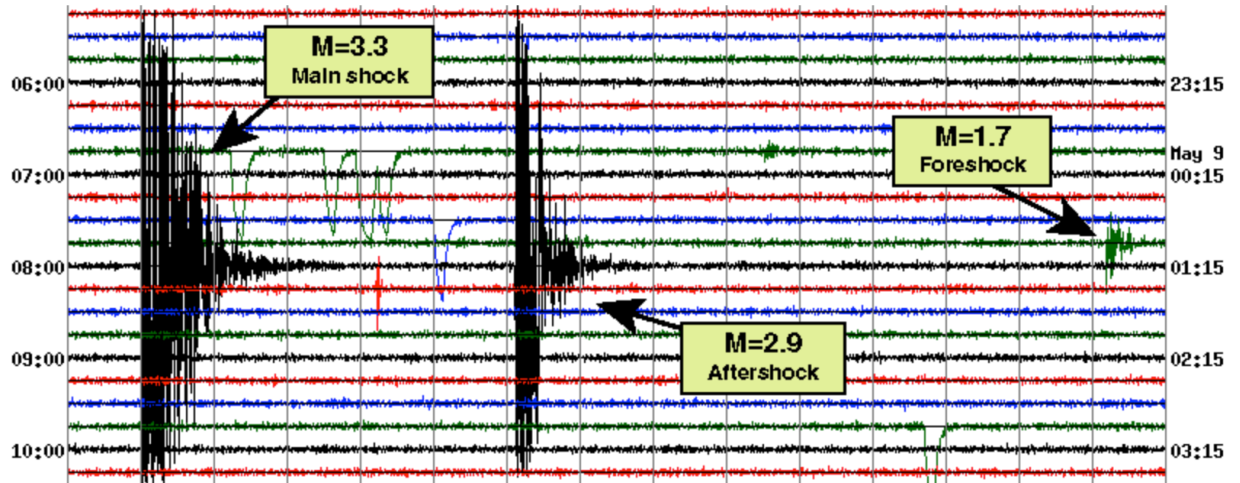


Figure 6 Main shock, aftershock and foreshock in the same seismogram [36]

There are many ways to sperate out different earthquakes. Windowing and clipping are one of the popular methods. Recursive STA/LTA is a popular algorithm for windowing tasks. The short time average/long time average (STA/LTA) is a triggering algorithm which is used to identify changes in the seismic amplitude. It is like an amplitude threshold algorithm but much better. Tuning and specifying the parameters of the STA/LTA algorithm is the difficult part [5]. The results of STA/LTA are not always perfect as seen in the experiment below.

An earthquake of magnitude 6.1 was recorded in Northern California on 08/24/2014, IRIS Wilber3 gives the P-wave and S-wave arrival time as seen in the figure below.

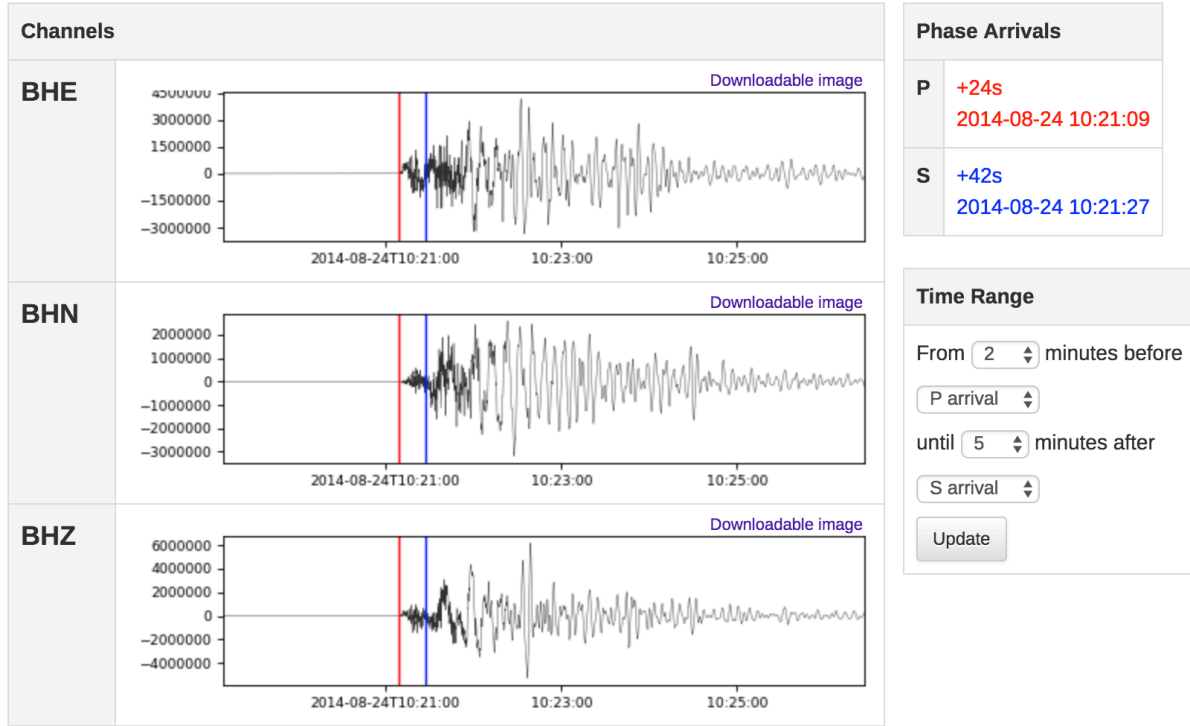


Figure 7 P-wave and S-wave arrival time for Northern California earthquake

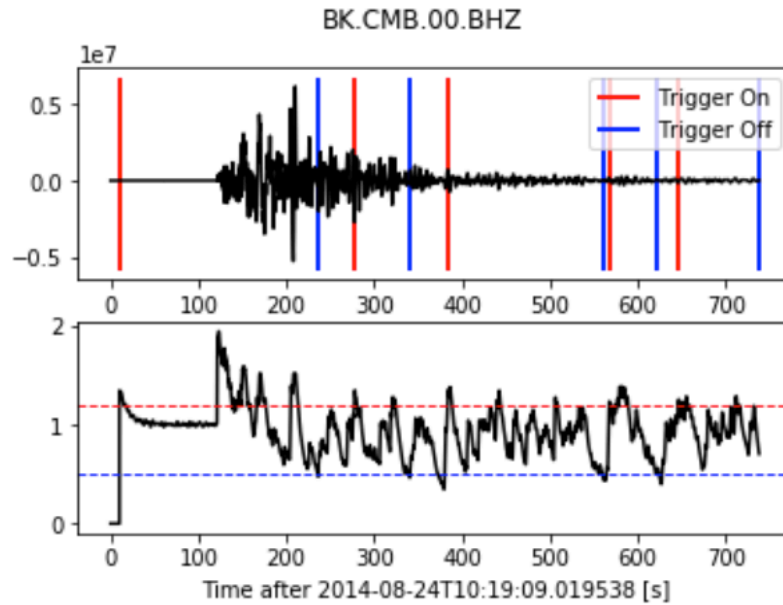


Figure 8 Recursive STA/LTA plot from OBSPY

But when Recursive STA/LTA from Obspy is used to calculate the P-wave and S-wave arrival times with default parameters it misses them completely. The problem here is not necessarily in the Recursive STA/LTA function but it could be faulty parameter tuning.

Another way to distinguishing earthquakes and non-earthquakes is training a machine learning algorithm to do this. This method is quicker and gives very good results provided the training dataset is good. In this project we will be implementing the latter approach. For the non-earthquake data, I have simply clipped data by randomly picking an earthquake file measuring its size and clipping non-earthquake data of the same size. This ensures that earthquake and non-earthquake files are of similar size.

2.3 Data visualization

The data used in this project is in SAC format. The SAC data can be plotted so that its contents can be inspected. Some examples of earthquake signals are plotted below.

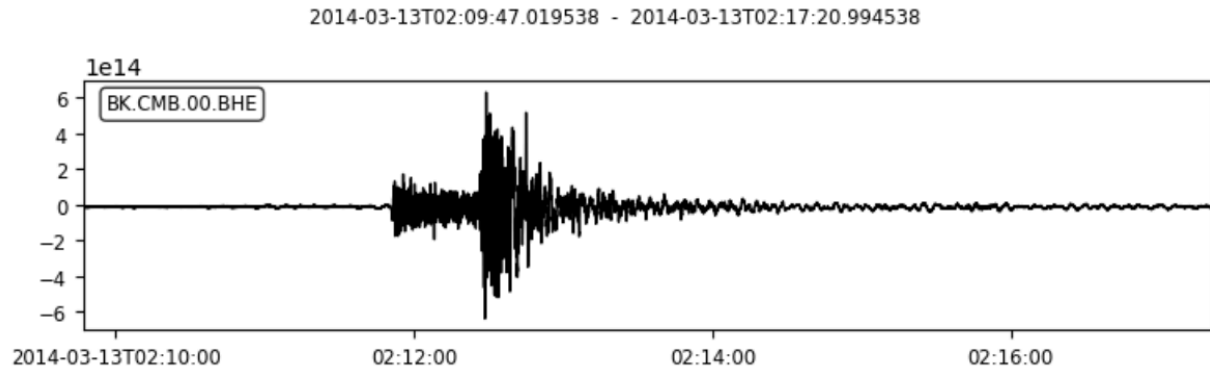


Figure 9 Earthquake plot a

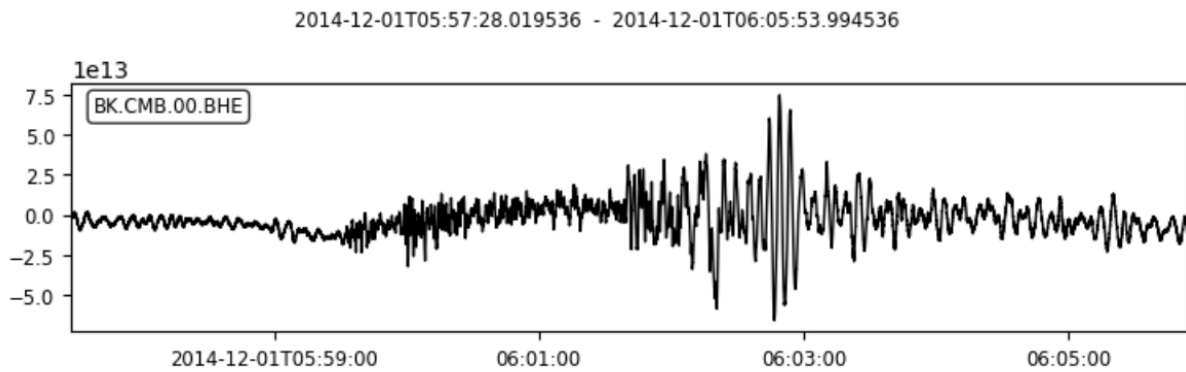


Figure 10 Earthquake plot b

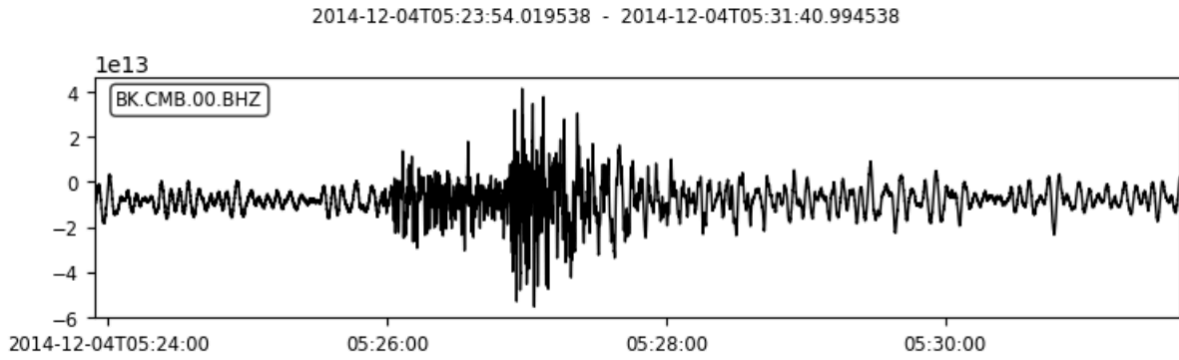


Figure 11 Earthquake plot c

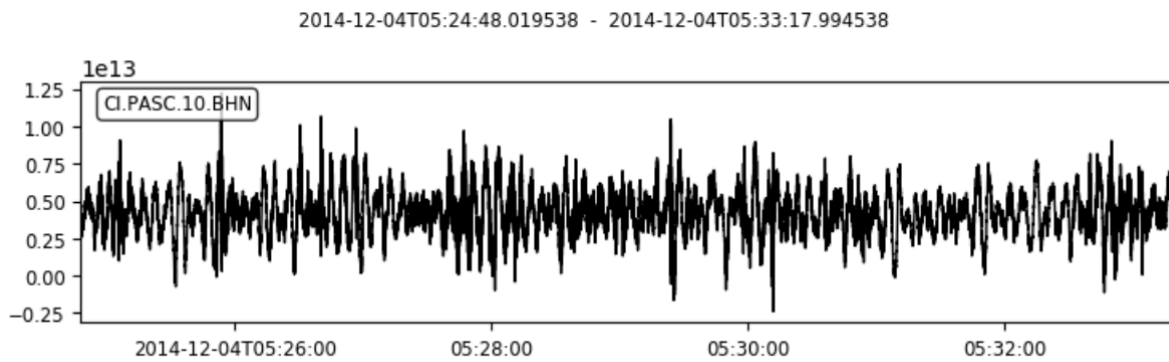


Figure 12 Earthquake plot d

Some examples of seismic noise is plotted below –

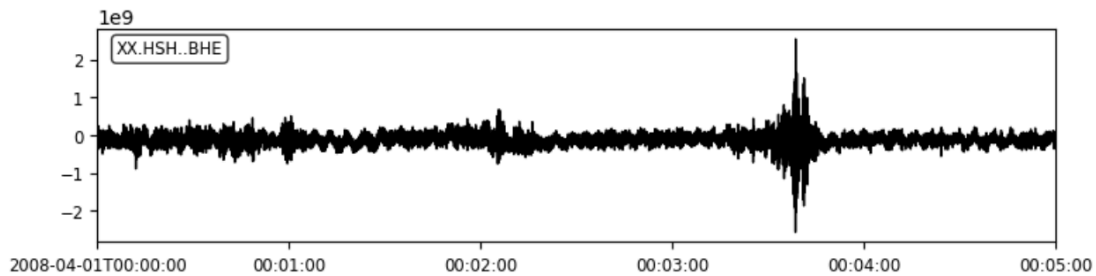


Figure 13 Noise plot a

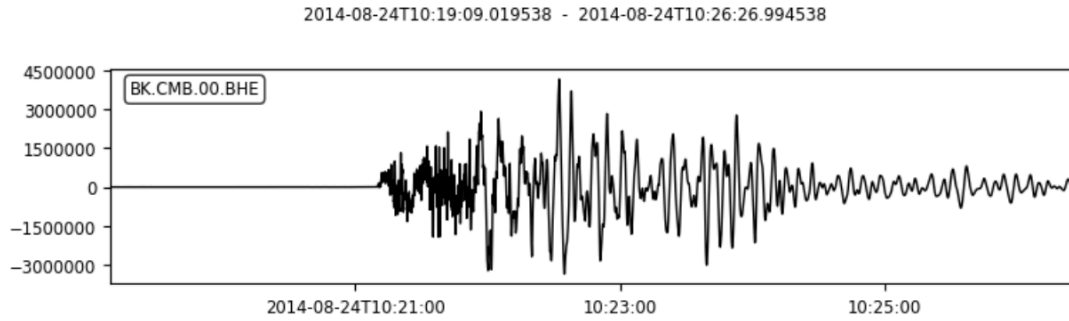


Figure 14 Noise plot b

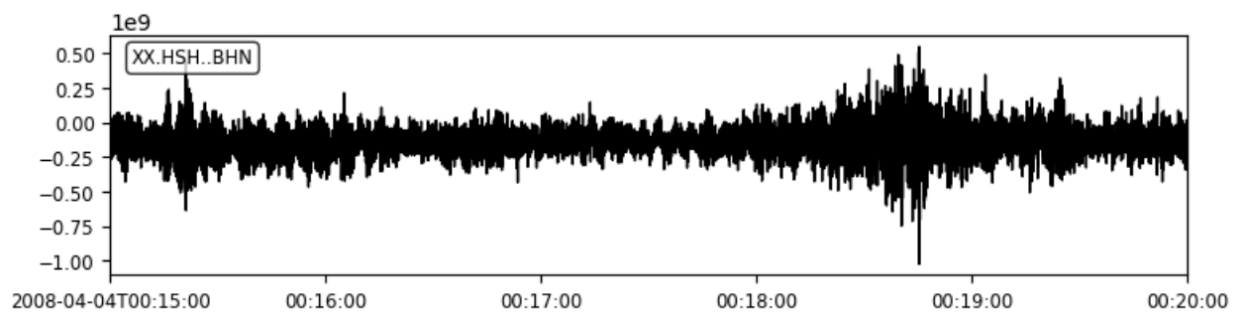


Figure 15 Noise plot c

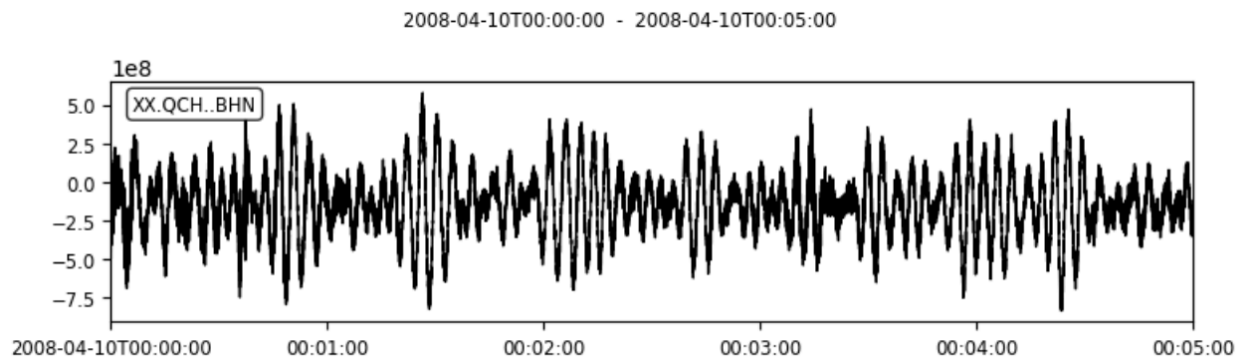


Figure 16 Noise plot d

In the last experiment of this project, we have also used sound signals to check the robustness of the model. The sound waves were used from the GTZAN Genre Collection dataset. Some

additional beatboxing sounds were also added to the dataset. A sound signal looks similar to earthquake signals. I plotted some of the sound signals from the dataset.

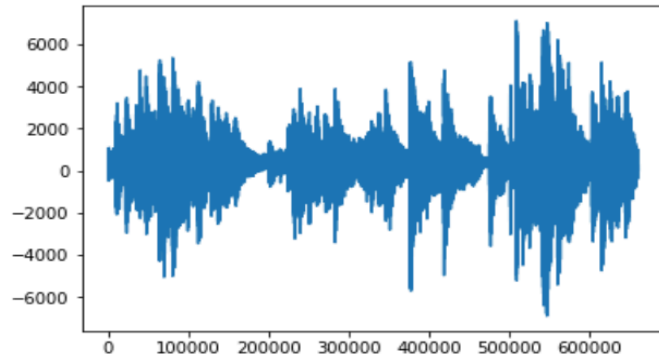


Figure 17 Sound Signal plot a

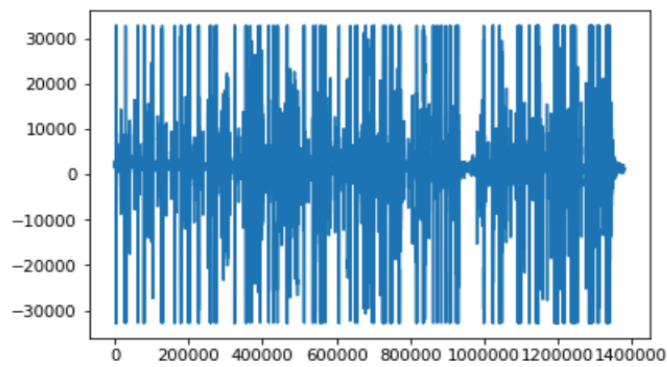


Figure 18 Sound signal plot b

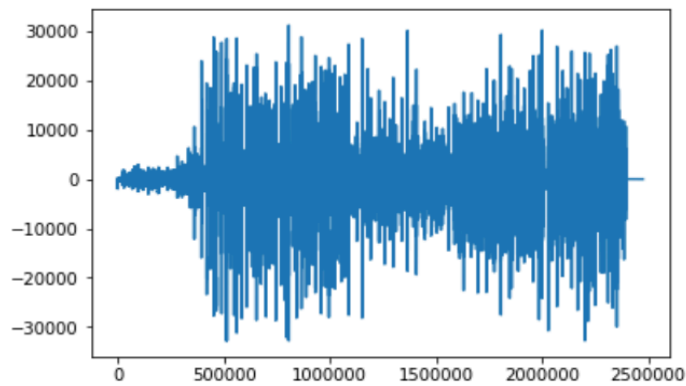


Figure 19 Sound signal plot c

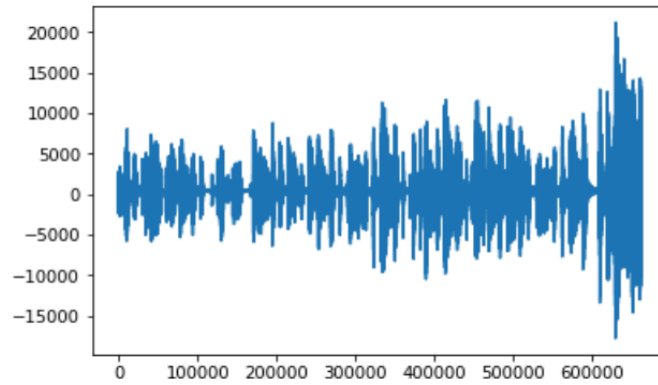


Figure 20 Sound signal plot d

In this project the amplitudes of all the sampling points is used as a feature for classifying earthquakes. If we use all the amplitudes we are essentially passing the entire signal pattern to the machine learning algorithm. A plot of a signal and its corresponding scatter plot of all its sampling

points amplitudes can be seen below. The signal plot and its amplitudes scatter plot are clearly similar.

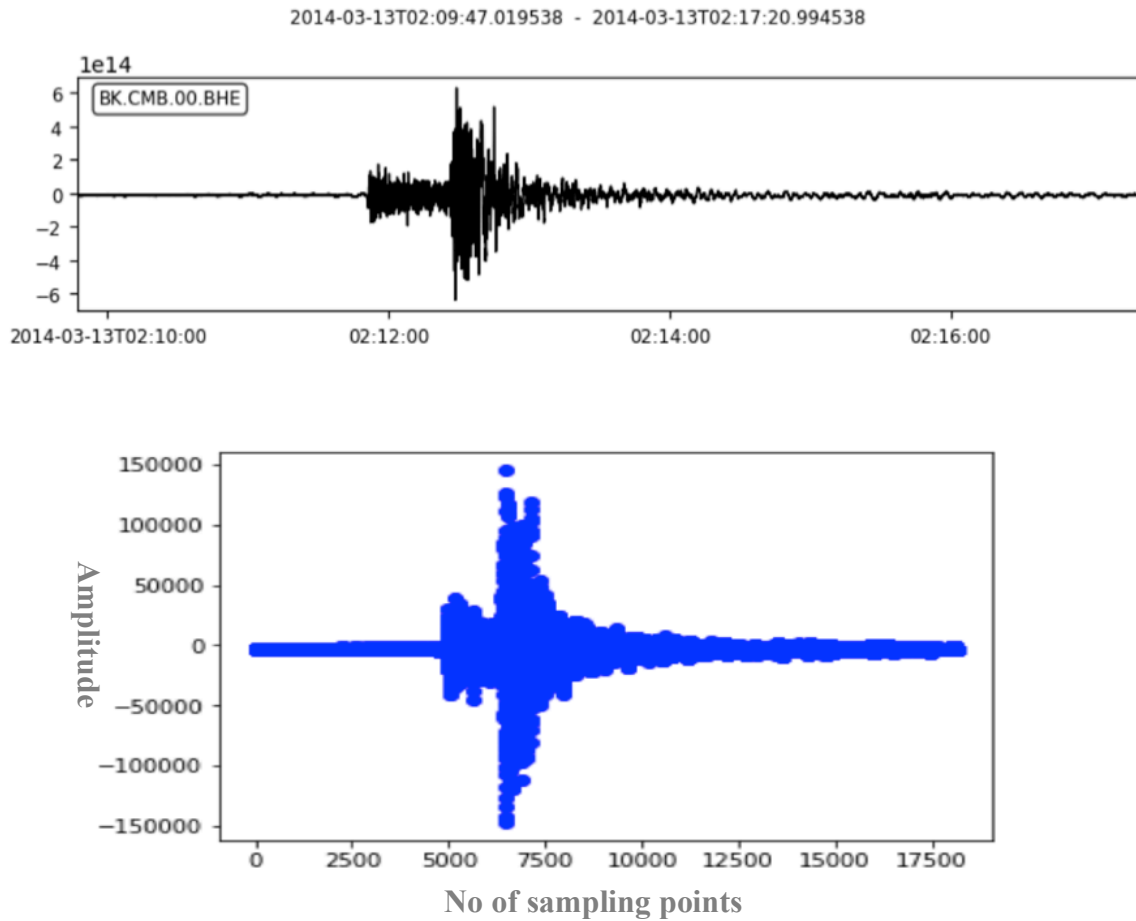


Figure 21 Scatter plot of amplitudes of all sampling points a

2014-12-01T05:57:28.019536 - 2014-12-01T06:05:53.994536

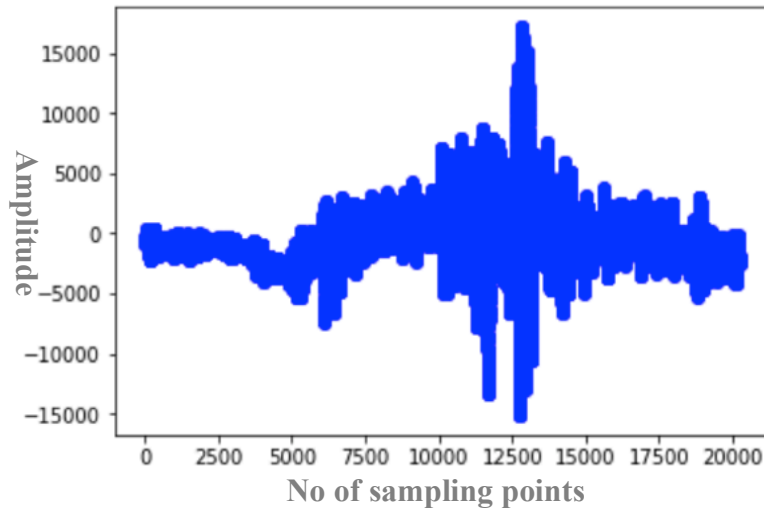
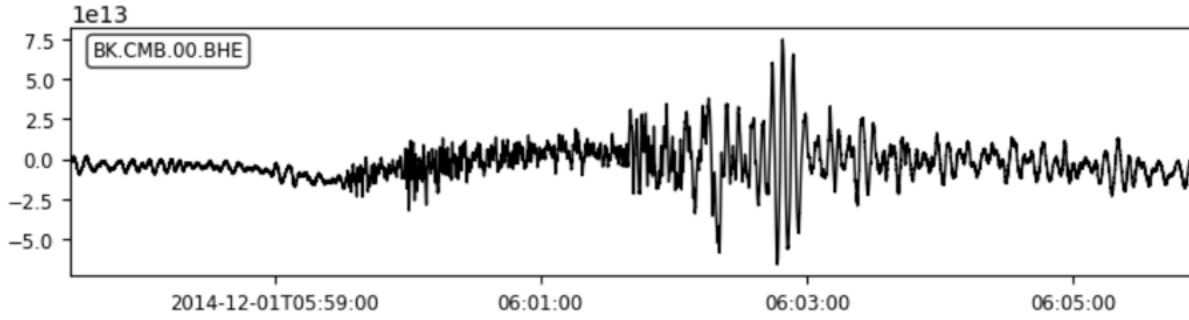


Figure 22 Scatter plot of amplitudes of all sampling points b

The array of amplitudes which was actually passed to the machine learning algorithm was in numerical format. The array ran up to approximately 80 pages

```
[-2.56000e+03, -3.16000e+03, -2.82800e+03, -2.83600e+03, -3.03200e+03, -2.76100e+03, -3.13200e+03,  
-3.09900e+03, -2.66000e+03, -3.28000e+03, -3.01100e+03, -2.82400e+03, -3.44100e+03, -2.93800e+03,  
-2.93400e+03, -3.28200e+03, -3.26200e+03, -3.25500e+03, -3.09300e+03, -3.26800e+03, -3.10100e+03,  
-3.14100e+03, -3.37500e+03, -3.25000e+03, -3.17000e+03, -3.14800e+03, -3.44800e+03, -3.40800e+03,  
-3.18100e+03, -3.34700e+03, -3.53600e+03, -3.25900e+03, -3.27300e+03, -3.49700e+03, -3.13400e+03,  
-3.44300e+03, -3.43900e+03, -3.13000e+03, -3.62000e+03, -3.26700e+03, -3.35000e+03, -3.51900e+03,  
-2.90500e+03, -3.55200e+03, -3.48400e+03, -3.02400e+03, -3.44900e+03, -3.27000e+03, -3.33500e+03,  
-3.34900e+03, -3.30500e+03, -3.49600e+03, -3.18400e+03, -3.30700e+03, -3.48500e+03, -3.10100e+03,  
-3.23400e+03, -3.34500e+03, -3.21100e+03, -3.49300e+03, -3.20400e+03, -3.10200e+03, -3.54300e+03,  
-3.09600e+03, -3.27800e+03, -3.42500e+03, -2.87700e+03, -3.50100e+03, -3.34000e+03, -2.94600e+03,  
-3.45200e+03, -3.26700e+03, -3.28500e+03, -3.25600e+03, -3.08500e+03, -3.54600e+03, -3.21600e+03,  
-2.91900e+03, -3.51200e+03, -3.10200e+03, -3.07800e+03, -3.52700e+03, -3.16200e+03, -3.36300e+03,  
-2.97000e+03, -2.94800e+03, -3.70300e+03, -3.01600e+03, -3.11400e+03, -3.38100e+03, -2.89300e+03,  
-3.32800e+03, -3.25500e+03, -3.10400e+03, -3.19900e+03, -3.06600e+03, -3.33800e+03, -3.19100e+03,  
-2.90700e+03, -3.26600e+03, -3.34300e+03, -2.96300e+03, -3.18200e+03, -3.25100e+03, -2.90600e+03,  
-3.27300e+03, -3.32400e+03, -2.92000e+03, -3.11400e+03, -3.19900e+03, -3.18500e+03, -3.20600e+03,  
-2.95300e+03, -3.16700e+03, -3.32500e+03, -2.95600e+03, -3.25100e+03, -3.31800e+03, -2.95800e+03,  
-3.22400e+03, -3.07000e+03, -3.15800e+03, -3.10700e+03, -3.07700e+03, -3.22200e+03, -2.92500e+03
```

Figure 23 Amplitudes of all sampling points

Combining all these data sources a number of experiments were performed which are described in detail in the following sections.

3. Methods

3.1 Experiment 1

In the first experiment, 100 earthquake and 100 non-earthquake files or noise files were picked. The files were read using the OBSPY read API and stored in a stream object. The start and end time of the signal was extracted from the stream object. The signal was then converted to ASCII format using OBSPY, start time and end time were parameters to the function used to convert the signal to ASCII format. The extracted signal was then vectorized using term frequency-inverse document frequency (TFIDF). A support vector machine was then trained using this vectorized earthquake and non-earthquake signals.

3.2 Experiment 2

In the second experiment, around 1600 signals were used to train the model. The dataset consisted of 791 earthquake signals and 783 non-earthquake or noise signals. In this experiment, the SAC header file values were also used to train the model. Along with the signal, features such as the start time of the signal, the end time of the signal, the channel name and increment between samples (delta) was used. In this experiment instead of extracting the entire signal in ASCII format, only the amplitudes of all the sampling points were used. All these features were combined, and a dataset was formed. This dataset was used to train SVM.

3.3 Experiment 3

In experiment 3 the same experimental setup of experiment 2 was used. Instead of SVM, random forest machine learning algorithm was used to train the model.

3.4 Experiment 4

In this experiment, the model was challenged with a tougher dataset. The non-earthquake files were manually clipped and were of the same size whereas the earthquake files were not modified and used as it is. In this experiment, changes were made to the algorithm to ensure that both types of files are of similar size and have the similar number of data points. When a non-earthquake file was picked, an earthquake file was also randomly picked. The length of the randomly picked earthquake file was measured and the non-earthquake file was clipped accordingly.

To make it more difficult for the model to distinguish between earthquakes and non-earthquakes random high amplitude spikes were added to non-earthquake data. A non-earthquake signal would be essentially flat if no noise is present in it. Introducing random spikes in the signal would make it difficult for the model to identify as not an earthquake. The following steps were implemented to introduce the above changes.

1. Randomly pick an earthquake file
 - a. Extract the amplitudes of all the sampling points in an array 'earthquake_array'
 - b. Measure the length of the array, let's call it earthquake_length.
 - c. Find the maximum amplitude from the earthquake_array.
 - d. Extract a total of 3000 sampling points from the earthquake_array. 1500 sampling points before the maximum amplitude and 1500 sampling points after the maximum amplitude.

- e. Now we have a slice of data with very high amplitude spikes in it.
2. Now proceed with the non-earthquake file
 - a. Extract the amplitudes of all the sampling points in an array 'noise_array'
 - b. Slice the array to earthquake_length
 - c. Pick a random point in the earthquake array
 - d. Append the high amplitude slice of data at this point.
3. Use features from the header file such as the channel name, number of sampling points, delta, start time of the signal and end time of the signal

3.5 Experiment 5

In this experiment, an attempt is made to improve the results of experiment 4 by performing transformations on existing features. The experiment setup is same as experiment 4, an additional feature was added by performing Fast Fourier Transform on the amplitudes of both earthquake and non-earthquake signals. Following steps were performed in this experiment –

1. Replicate step 1 to step 3 of experiment 4
2. Perform Fast Fourier Transform (FFT) of the earthquake and non-earthquake signal
3. Add the Fast Fourier transform as an additional feature along with all the extracted features in step 1.

3.6 Experiment 6

The aim of this experiment is to test if the model is able to distinguish between an earthquake and sound waves. Sound waves are very similar to earthquakes, some studies convert earthquake signals to sound for further processing and analysis [37]. In this experiment, I download 78 .wav sound files from the GTZAN Genre Collection dataset. As the sound waves do not have a header file with features similar to earthquakes, only the amplitudes of the sampling points will be considered as features. The amplitudes of both the sound waves, earthquake signals and non-earthquake signals are extracted. The aim of this experiment is to test if the model classifies the sound wave as non-earthquakes.

4. Results

4.1 Experiment 1

The model achieved a prediction accuracy of only 56%. The graph below shows the result of 5-fold cross-validation.

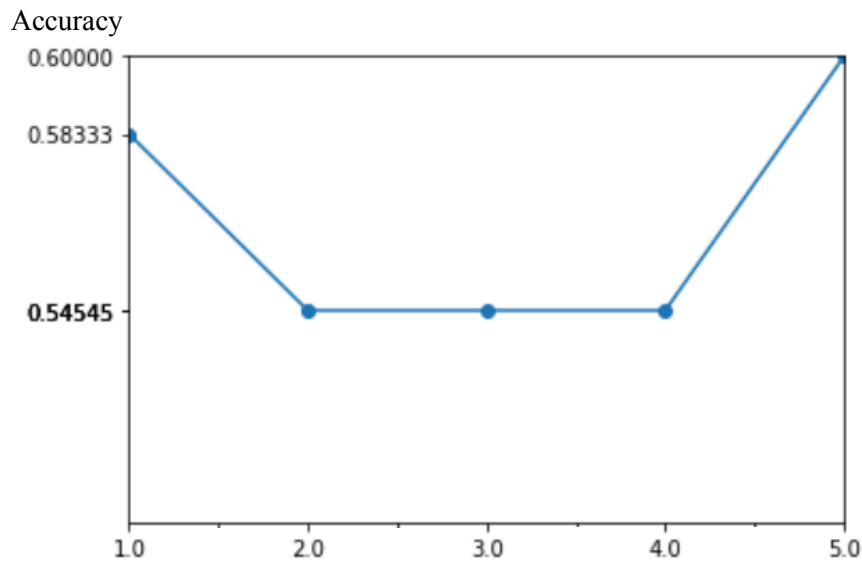


Figure 24 Experiment 1- 5 fold Cross Validation results

4.2 Experiment 2

The model achieved a prediction accuracy of only 56%. The graph below shows the result of 5-fold cross-validation.

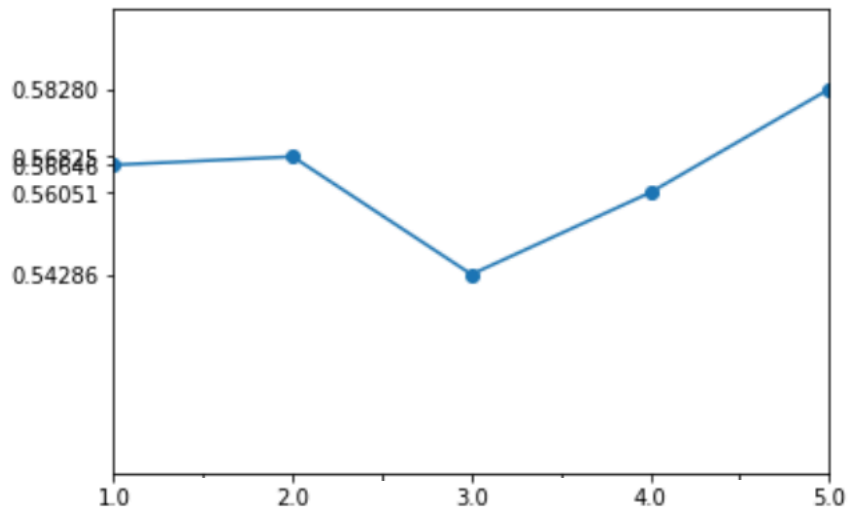


Figure 25 Experiment 2 cross validation results

The precision-recall curve for this experiment is as follows –

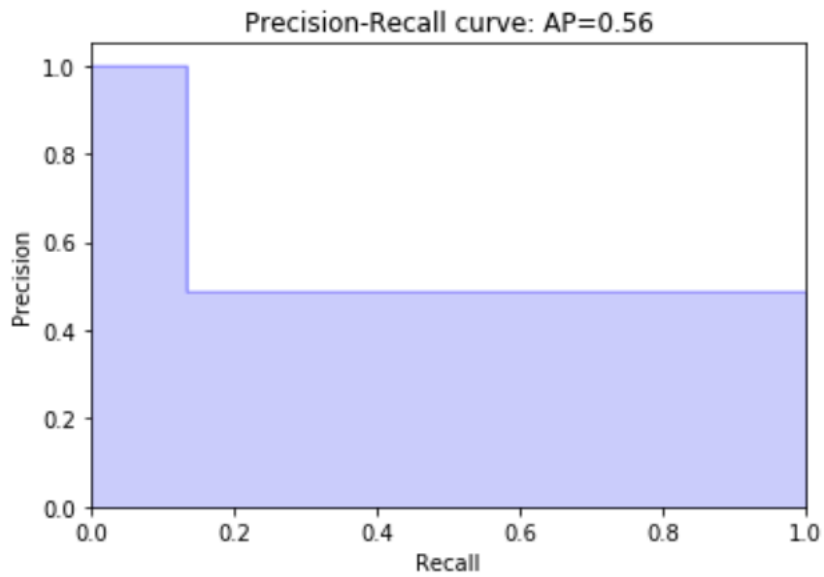


Figure 26 Experiment 2 precision-recall plot

Table 1 Experiment 2 confusion matrix

	Predicted No	Predicted Yes
Actual No	243	0
Actual Yes	199	31

GridSearchCV was used to apply an exhaustive search over specified SVM parameters.

GridSearchCV returned the following results.

Table 2 Experiment 2 GridSearchSV output

	mean_test_score	mean_train_score	param_C	param_kernel	rank_test_score
1	0.999365	0.999365	1	linear	1
2	0.508895	0.81226	1	rbf	3
3	0.999365	0.999365	10	linear	1
4	0.507624	0.791613	10	rbf	4

According to the results of GridSearchCV parameter C=1 with kernel= 'linear' and parameter C=10 with kernel='linear' should give good results. After training an SVM with these parameters I got the following results.

I got the following tree after training on decision trees classifier

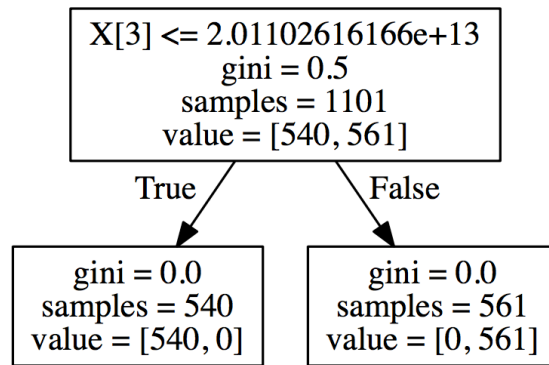


Figure 27 Experiment 2 overfitted decision tree

The tree is a perfect classifier, the separating factor being the end time. The same case applied to the start time as well. After removing the end time, the start time acted as the perfect separator.

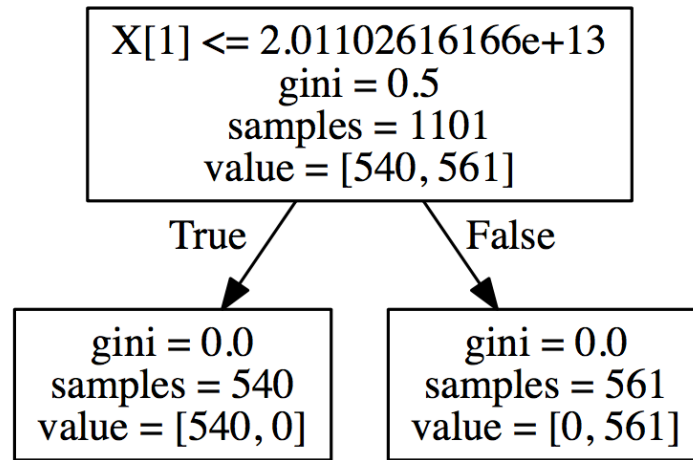


Figure 28 Experiment 2 overfitted decision tree

After removing the start and end time the decision tree classifier returned a multilevel tree, which meant it took a lot of the features under consideration.

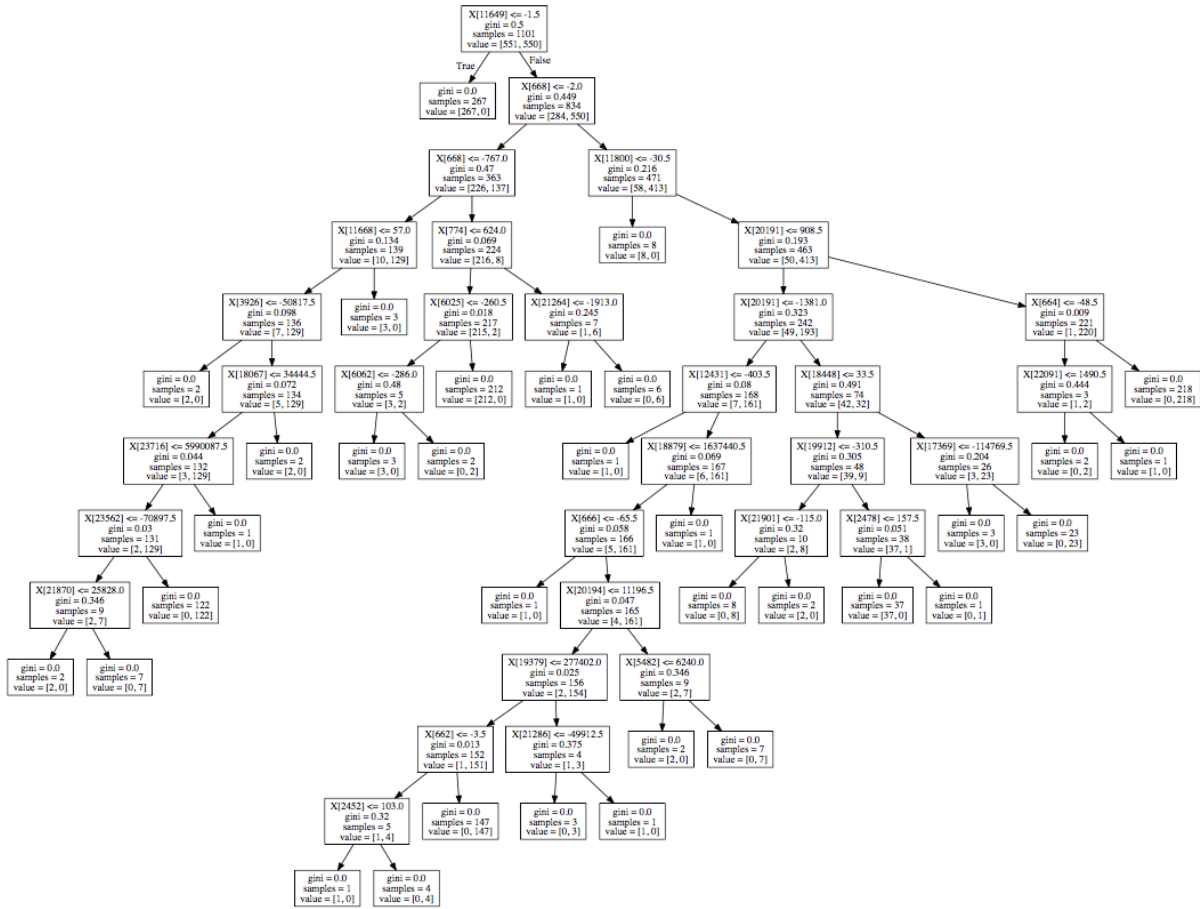


Figure 29 Experiment 2 decision tree

After dropping the start and end time from the features, I trained SVM with the parameters suggested by GridSearchCV. The new dataset with reduced features was again given the GridSearchCV to find the optimal parameters to train SVM. It returned the following results –

Table 3 Experiment 2 GridSearchCV output

	mean_test_score	mean_train_score	param_C	param_kernel	rank_test_score
1	0.523507	1	1	linear	2
2	0.550826	1	1	rbf	1
3	0.523507	1	10	linear	2
4	0.550826	1	10	rbf	1

4.3 Experiment 3

Random forest algorithm was used to train the model with the new dataset. The graph below shows the result of 5-fold cross-validation.

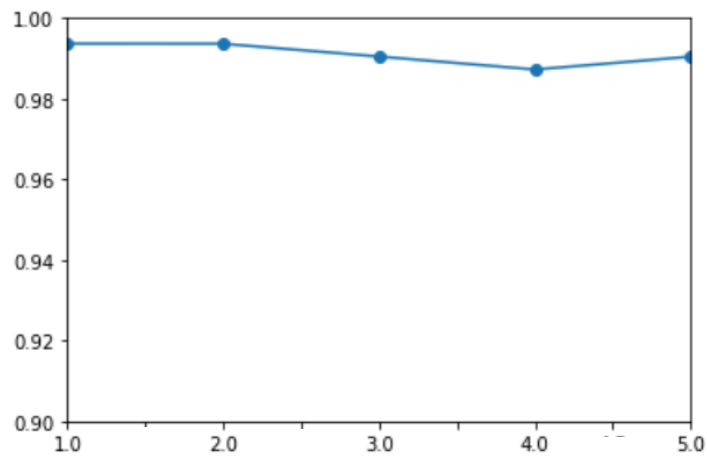


Figure 30 Experiment 3 cross validation graph

Table 4 Experiment 3 confusion matrix

	Predicted No	Predicted Yes
Actual No	222	3
Actual Yes	0	247

4.4 Experiment 4

In this experiment, the lengths of the non-earthquake signals were clipped to match the length of the earthquake signals. In the graph below the lengths of both the earthquakes and non-earthquakes are plotted. The blue dots are non-earthquakes whereas the red dots are earthquakes. As seen in the scatter plot below they are now a good mix.

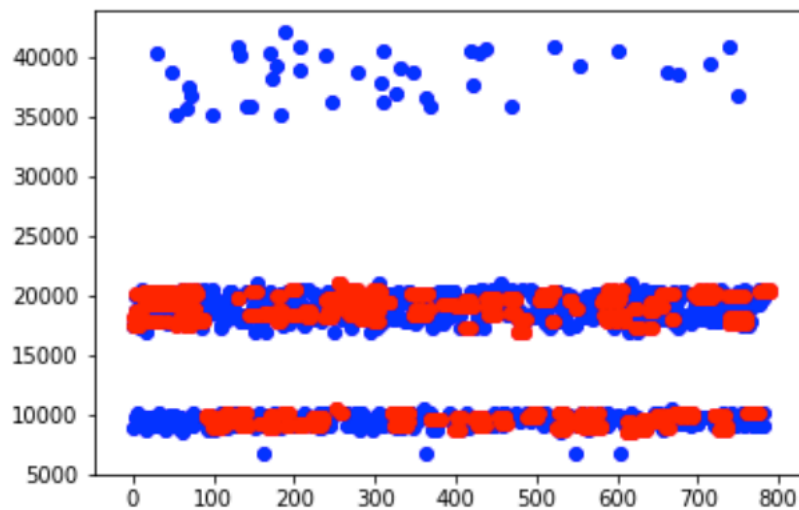


Figure 31 Earthquake and non-earthquake lengths scatter plot

After appending spikes of high amplitudes in the non-earthquake signals at random locations it was expected that the random forests model will find it harder to distinguish between earthquakes and non-earthquakes. As we had anticipated the prediction accuracy went down to 89% on an average. The graph below shows the result of 5-fold cross-validation of the tough dataset.

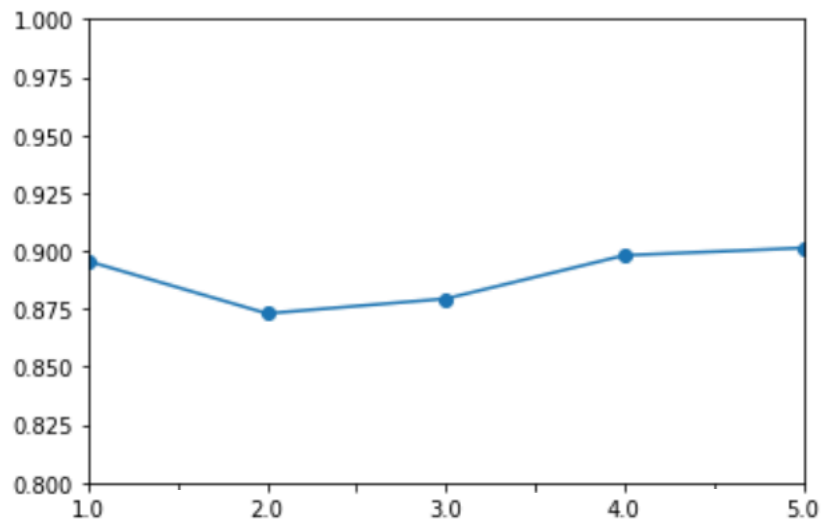


Figure 32 Experiment 4 cross validation results

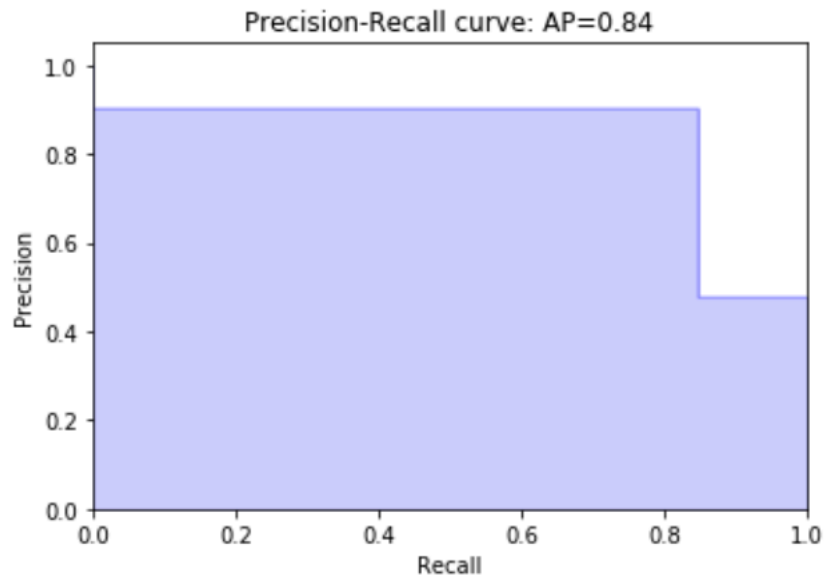


Figure 33 Experiment 4 precision-recall curve

Table 5 Experiment 4 confusion matrix

	Predicted No	Predicted Yes
Actual No	227	21
Actual Yes	34	191

4.5 Experiment 5

Fast Fourier transforms are widely used in the world of Seismology. FFT gave really good results even with the tough dataset. The figure below shows how the signal transforms after applying Fast Fourier transform.

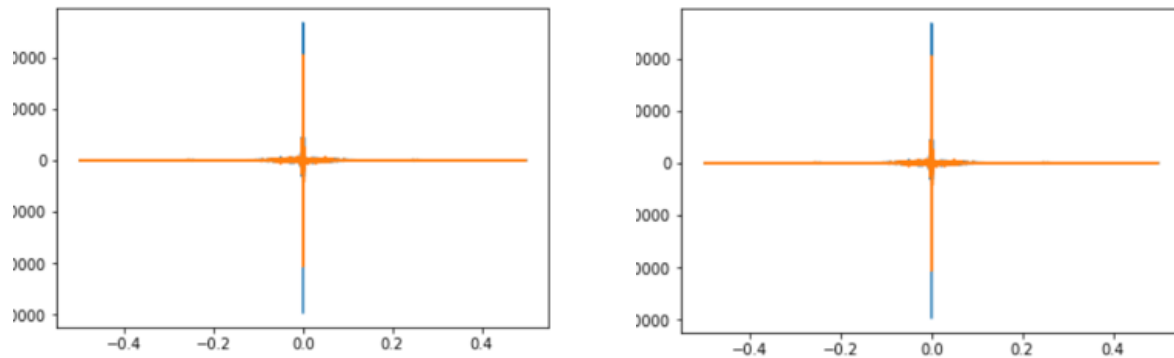


Figure 34 FFT of seismic noise

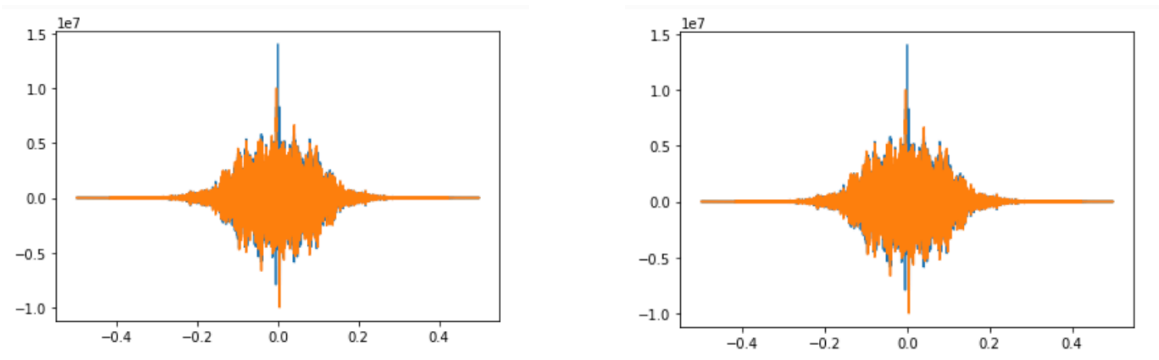


Figure 35 FFT of earthquake signals

The prediction accuracy increased to a little more than 95% with a random forest model. The graph below shows the result of 5-fold cross-validation of the tough dataset with FFT.

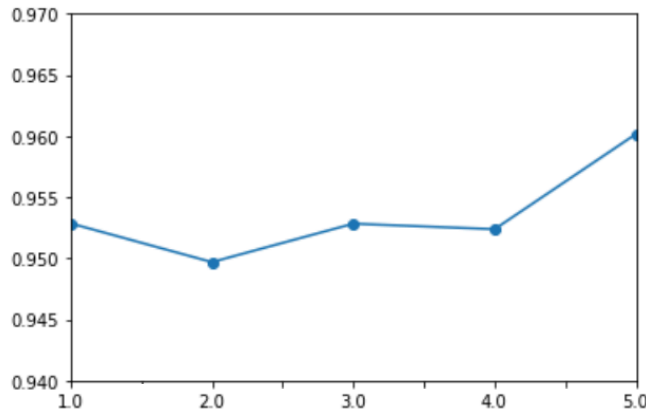


Figure 36 Experiment 5 cross-validation graph

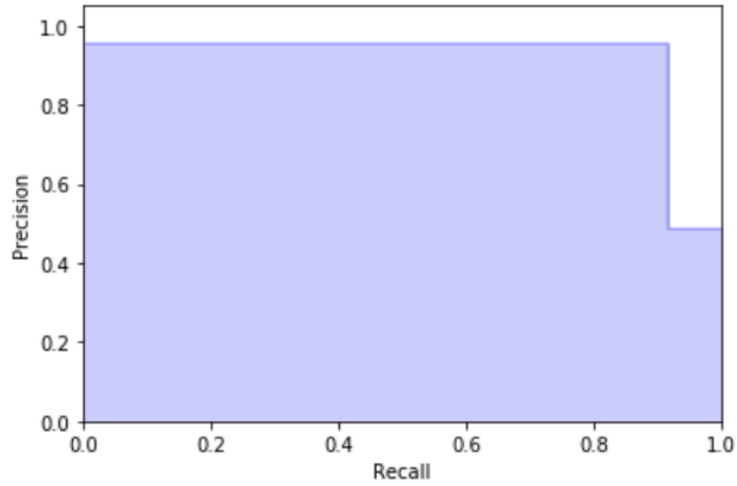


Figure 37 Experiment 5 precision-recall curve

Table 6 Experiment 5 confusion matrix

	Predicted No	Predicted Yes
Actual No	233	10
Actual Yes	19	211

4.6 Experiment 6

Training the algorithm only using time series features without sound we get an accuracy 92% on an average.

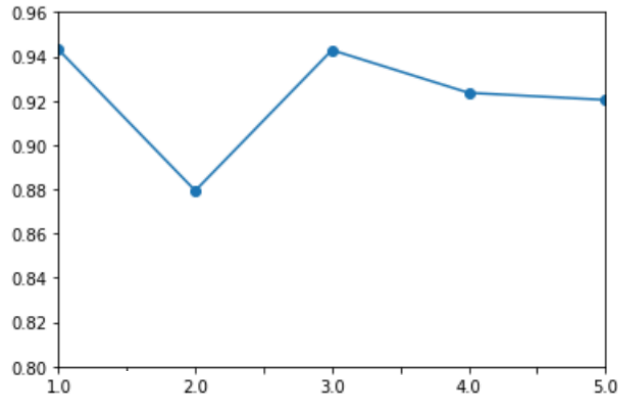


Figure 38 Experiment 6 cross-validation time series accuracy

The model trained with a dataset containing earthquakes and non-earthquakes with no sound waves and tested it against sound waves it classified only one sound wave as a non-earthquake.

```

predicted
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
      1])

```

Figure 39 Experiment 6 random forest output

Next, the random forest model was trained with a mix of earthquake, non-earthquake and 55 sound files which were marked as non-earthquake. I then tested my model with 24 sound files. It classified 21 out of 24 sound waves correctly as non-earthquakes

```

predicted
array([0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0])

```

Figure 40 Experiment 6 random forest output

5. Discussion

In experiment 1 the model gave a low prediction accuracy. It may be because only 200 signals were used to train the model. Experiment 2 gave similar results to experiment 1. Additional data and features did not help in training the model. SVM is a very versatile model, tuning parameters is very important while training SVM. GridSearchCV was used for an exhaustive search of a combination of parameters that would work best for SVM. GridSearchCV returned parameter $C=1$ with kernel= 'linear' and parameter $C=10$ with kernel='linear' as the best parameters. SVM gave a prediction accuracy of almost 100% with these parameters. As SVM returned almost perfect results, it could mean SVM was overfitting. In order to understand this further, I used decision trees algorithm on the data. Decision tree algorithm is not a black box algorithm, it is easy to see what is exactly going on. The decision tree algorithm returned a perfect classifier. The model was overfitting on the start time of the signal. After removing start time, the model overfitted on end time of the signal. After removing both the features, decision tree classifier returned a general classifier. The new dataset was used to train SVM with parameters $C=1$ with kernel= 'linear' as suggested by GridSearchCV. The model returned a prediction accuracy of approximately 50%. The new dataset was again given to GridSearchCV to find optimized parameters for the new dataset. All the combinations returned by GridSearchCV gave low test data accuracies. The mean train score was 1 but test scores were poor. It still means that the SVM model was overfitting. As decision tree was not overfitting anymore, I decided to try a more randomized decision tree which is random forests algorithm.

The random forest model performed well and achieved a prediction accuracy of 99% on an average. After testing the model with the tougher dataset, the prediction accuracy dropped as expected. Fast Fourier transform worked really well as seen in experiment 5. Fast Fourier

transform is widely used in seismology. The model performed better than the previous experiment. Thus, combining traditional analytical methods such as FFT with machine learning gave the best results so far.

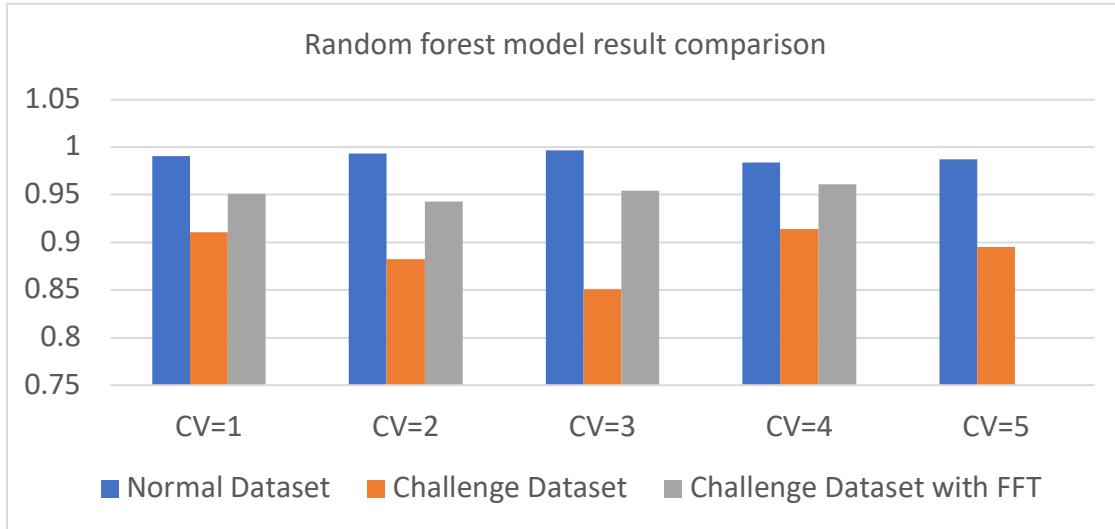


Figure 41 Random Forest model result comparison

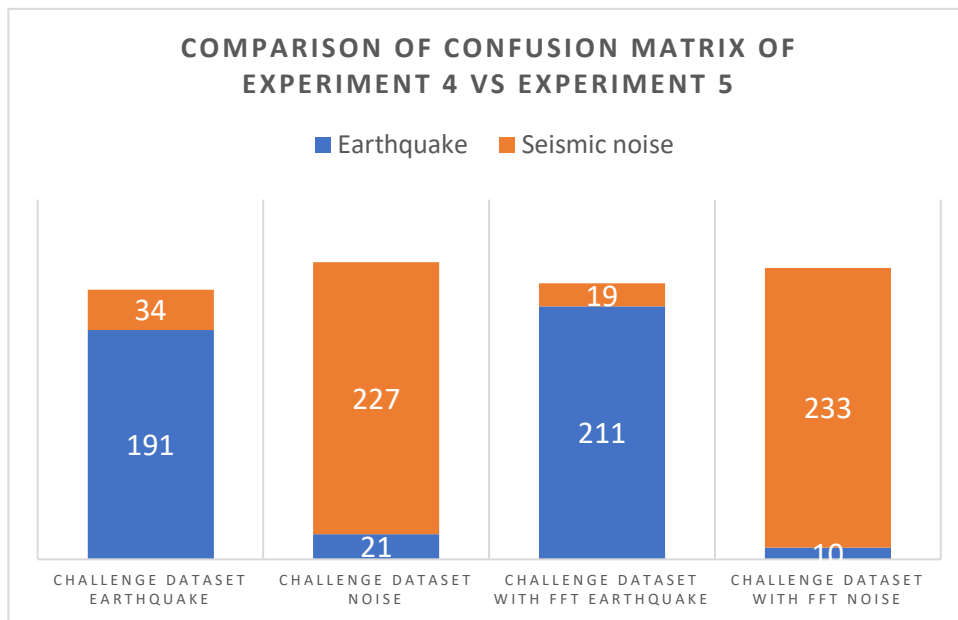


Figure 42 Confusion matrix experiment 4 vs experiment5

The random forest model which was trained on earthquakes and station noises was not able to classify sound waves as noise. When sound waves were included in training data, the model was able to classify sound waves as non-earthquakes.

Machine learning algorithms can be used effectively for distinguishing earthquakes and noise as seen in the above experiments. Support Vector Machine is known to work well with time series data [38]. However, the SVM model that was trained on earthquake data was weak. On the other hand, random forest model successfully classified approximately 95% of the signals. Random forest works well with overfitting [39] which was a problem with SVM. Appending high amplitude spikes to noise would model scenarios such as a mining blast near an earthquake station or a heavy truck passing on top of it. The random forest model with Fast Fourier transform was able to classify and distinguish these scenarios with an accuracy of 96% .

In this project, no distinction was made between different stations. The signals recorded at each station vary slightly. The noise in the signal depends on the rock composition, the instrument error and other external factors [41]. If a random forest model is trained separately for each station it may return better results. The next step in this project would be to predict the P wave arrival time and the S wave arrival time. We have already successfully distinguished the signals into earthquakes and non-earthquakes. The earthquake recordings can be further analyzed to predict P-wave arrival time and the S-wave arrival time. Depending on the location of the epicenter the proposed early warning model will be able to give an earthquake warning up to a few minutes.

6. References

1. Iris.edu. (n.d.). *How are earthquakes recorded? (Station)*. [online] Available at: https://www.iris.edu/gallery3/general/posters/exploring_earth/station [Accessed 12 Jan. 2018].
2. Earthquake.usgs.gov. (2018). *Earthquake Glossary*. [online] Available at: <https://earthquake.usgs.gov/learn/glossary/?term=seismograph> [Accessed 17 Nov. 2017].
3. Sms-tsunami-warning.com. (n.d.). *Seismograph & Seismometer*. [online] Available at: <http://www.sms-tsunami-warning.com/pages/seismograph#.WrsmRNPwYWI> [Accessed 16 Feb. 2018].
4. Endsley, K. (2007). *What Is Seismology and What Are Seismic Waves?*. [online] Geo.mtu.edu. Available at: <http://www.geo.mtu.edu/UPSeis/waves.html> [Accessed 29 Apr. 2018].
5. Trnkoczy, A. (1998). Understanding and setting STA/LTA Trigger algorithm parameters for the K2. [online] Available at: http://ftp://hazards.cr.usgs.gov/Eq_Effects/GeekPack/Procedures-Configs-Info/1_Dataloggers/K2-Altus/Sta-Lta.PDF [Accessed 29 Nov. 2017].
6. M. Beyreuther, R. Barsch, L. Krischer, T. Megies, Y. Behr and J. Wassermann (2010), "ObsPy: A Python Toolbox for Seismology", *SRL*, 81(3), 530-533, DOI: 10.1785/gssrl.81.3.530
7. Docs.scipy.org. (2018). *Discrete Fourier Transform (numpy.fft) – NumPy v1.14 Manual*. [online] Available at: <https://docs.scipy.org/doc/numpy/reference/routines.fft.html#module-numpy.fft> [Accessed 19 Mar. 2018].
8. J. Ramirez Jr and F. G. Meyer, "Machine Learning for Seismic Signal Processing: Phase Classification on a Manifold," *2011 10th International Conference on Machine Learning and Applications and Workshops*, Honolulu, HI, 2011, pp. 382-388.
9. O. Lindenbaum, N. Rabin, Y. Bregman and A. Averbuch, "Multi-channel fusion for seismic event detection and classification," *2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE)*, Eilat, 2016, pp. 1-5.

10. Gang Zhao, Hanming Huang and Xinxin Lu, "Discriminating earthquakes and explosion events by seismic signals basing on BP-AdaBoost classifier," *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, Chengdu, 2016, pp. 1965-1969.
11. Y. Takase, S. Sako and T. Kitamura, "Detection of anomalous environmental electromagnetic waves based on daily average using Hidden Markov Model," *2009 Sixth International Conference on Networked Sensing Systems (INSS)*, Pittsburgh, PA, 2009, pp. 1-1.
12. S. Urata, H. Yasukawa, A. Itai and I. Takumi, "A study on clustering for anomalous signal detections from electromagnetic wave data," *2012 IEEE International Geoscience and Remote Sensing Symposium*, Munich, 2012, pp. 6083-6086.
13. W. Astuti, R. Akmeliawati, W. Sediono and M. J. E. Salami, "Hybrid Technique Using Singular Value Decomposition (SVD) and Support Vector Machine (SVM) Approach for Earthquake Prediction," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 5, pp. 1719-1728, May 2014.
14. Ruano A.E., Madureira G., Barros O., Khosravani H.R., Ruano M.G., Ferreira P.M, "Seismic detection using support vector machines," in *Neurocomputing*, 135, pp. 273-283, 2014
15. N. Orlic, S. Loncaric, "Earthquake—explosion discrimination using genetic algorithm-based boosting approach," in *Computers & Geosciences*, ISSN: 0098-3004, Vol: 36, Issue: 2, Page: 179-185, 2010
16. S. Gentili, A. Michelini, "Automatic picking of P and S phases using a neural tree," in *Journal of Seismology*, 2006, Volume 10, Number 1, Page 39, January 2006
17. "List of earthquakes in 2017", *En.wikipedia.org*, 2017. [Online]. Available:https://en.wikipedia.org/wiki/List_of_earthquakes_in_2017. [Accessed: 19-Dec- 2017].
18. "List of earthquakes in 2017", *En.wikipedia.org*, 2017. [Online]. Available: https://en.wikipedia.org/wiki/List_of_earthquakes_in_2017. [Accessed: 19- Dec- 2017].
19. "Causes and Appearance of Noise in Seismic Data Volumes, by Satinder Chopra and Kurt J. Marfurt; #41476 (2014).", *Searchanddiscovery.com*, 2017. [Online]. Available:

- <http://www.searchanddiscovery.com/documents/2014/41476chopra/chopra.html>.
[Accessed: 19- Dec- 2017].
20. "obspsy.core.trace.Trace.detrend — ObsPy Documentation (1.1.0)", *Docs.obspsy.org*, 2017.
[Online]. Available:
<https://docs.obspsy.org/packages/autogen/obspsy.core.trace.Trace.detrend.html#obspsy.core.trace.Trace.detrend>. [Accessed: 19- Dec- 2017].
21. De Groot, R. (2018). *ShakeAlert | Earthquake Early Warning*. [online] Shakealert.org.
Available at: <https://www.shakealert.org/> [Accessed 21 Feb. 2018].
22. Earthquake.usgs.gov. (n.d.). *Earthquake Early Warning*. [online] Available at:
<https://earthquake.usgs.gov/research/earlywarning/overview.php> [Accessed 1 May 2018].
23. Vink, G. (1994). *Nuclear testing and nonproliferation*. Arlington, VA: IRIS Consortium.
24. Donovan, T. (2011). *Could Scientists Have Predicted Japan's Disaster?*. [online] HuffPost UK. Available at: https://www.huffingtonpost.com/2011/03/11/sendai-earthquake-2011-japan_n_834688.html [Accessed 1 May 2018].
25. Norio, O., Ye, T., Kajitani, Y. et al. *Int J Disaster Risk Sci* (2011) 2: 34.
<https://doi.org/10.1007/s13753-011-0004-9>
26. Bloor, R. (2017). *Why do Machine Learning on Big Data? | Skytree*. [online] Skytree.
Available at: <http://www.skytree.net/machine-learning/why-do-machine-learning-big-data/> [Accessed 3 May 2018].
27. <http://www.geo.mtu.edu/UPSeis/waves.html>
28. Ds.iris.edu. (2012). *SAC Data File Format*. [online] Available at:
https://ds.iris.edu/files/sac-manual/manual/file_format.html [Accessed 1 May 2018].
29. Docs.obspsy.org. (2016). *obspsy.core.stream.Stream.plot — ObsPy Documentation (1.1.0)*.
[online] Available at:
<https://docs.obspsy.org/packages/autogen/obspsy.core.stream.Stream.plot.html> [Accessed 11 Apr. 2018].
30. Docs.obspsy.org. (2016). *obspsy.core.stream.Stream.slice — ObsPy Documentation (1.1.0)*.
[online] Available at:
<https://docs.obspsy.org/packages/autogen/obspsy.core.stream.Stream.slice.html> [Accessed 11 Apr. 2018].

31. Docs.obsipy.org. (2016). *obsipy.core.trace.Trace.max* — *ObsPy Documentation (1.1.0)*. [online] Available at: <https://docs.obsipy.org/packages/autogen/obsipy.core.trace.Trace.max.html> [Accessed 10 Apr. 2018].
32. En.wikipedia.org. (2018). *Cepstrum*. [online] Available at: <https://en.wikipedia.org/wiki/Cepstrum> [Accessed 3 May 2018].
33. Ds.iris.edu. (2018). *IRIS: Wilber 3: Select Event*. [online] Available at: https://ds.iris.edu/wilber3/find_event [Accessed 3 May 2018].
34. Iris.edu. (2018). *IRIS*. [online] Available at: <https://www.iris.edu/hq/> [Accessed 2 May 2018].
35. Pubs.usgs.gov. (2016). *The Severity of an Earthquake*. [online] Available at: <https://pubs.usgs.gov/gip/earthq4/severitygip.html> [Accessed 7 Dec. 2017].
36. Earthquake.usgs.gov. (n.d.). *Examples*. [online] Available at: <https://earthquake.usgs.gov/monitoring/seismograms/examples.php> [Accessed 3 May 2018].
37. Holtzman, B., Candler, J., Turk, M., & Peter, D. (2014). Seismic sound lab: Sights, sounds and perception of the earth as an acoustic space. In *Sound, Music, and Motion - 10th International Symposium, CMMR 2013, Revised Selected Papers* (Vol. 8905, pp. 161-174). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 8905). Springer Verlag. DOI: 10.1007/978-3-319-12976-1_10
38. A. Kampouraki, G. Manis and C. Nikou, "Heartbeat Time Series Classification With Support Vector Machines," in *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 4, pp. 512-518, July 2009. doi: 10.1109/TITB.2008.2003323
39. En.wikipedia.org. (2017). *Random forest*. [online] Available at: https://en.wikipedia.org/wiki/Talk%3ARandom_forest#Overfitting [Accessed 1 May 2018].
40. Dost, B., Zednik, J., Havskov, J., Willemann, R. J., Bormann, P. (2012): Seismic Data Formats, Archival and Exchange. - In: Bormann, P. (Ed.), *New Manual of Seismological*

Observatory Practice 2 (NMSOP-2), Potsdam : Deutsches GeoForschungsZentrum GFZ,
pp. 1—19.

41. E. SHERIFF, R. (2006). Factors affecting seismic amplitude. *Geophysical Prospecting*. 23.
125 - 138. 10.1111/j.1365-2478.1975.tb00685.x.