***Seismic data analysis and classification based on Machine Learning***

A Project Report Presented to

The faculty of the Department of Electrical Engineering

San José State University

In Partial Fulfillment of the Requirements for the Degree

Master of Science

By

*Shaminderpal singh Sandhu*

*EE297B Section 1, Spring 2018*

*Shaminderpal.sandhu@sjsu.edu*

*Ali Saeidi Ashtiyani [010813036]*

*EE297B Section 1, Spring 2018*

*ali.saeidiashtiyani@sjsu.edu*

**Department of Electrical Engineering**

Charles W. Davidson College of Engineering

San José State University

San Jose, CA 95192-0084

***Abstract- Seismic events such as Earthquakes are very under-researched in global scientific domains but, with advancements in Machine Learning, classification and analysis of seismic data is becoming more feasible. This project focuses on developing techniques to effectively classify Seismic data into earthquake events and seismic noise (mining blasts, man-made noise etc.). The project not only focuses on developing deep learning models but, is also focused on collecting seismic data from credible repositories with proper labelling. An extensive research in seismology ensued proper knowledge of Earthquake waves, their propagation and behavior.*** ***Main motivation behind this project is to defy already accepted triggering (STA/LTA, Z Transform etc.) as well as Picking (AR picker, baer picker etc.) algorithms and formulate new techniques to extract P and S wave features. Problem with these algorithms is that different threshold values can produce entirely different triggers and accurate thresholds are hard to generalize for large number of events.***

# 

# **Acknowledgement**

This Research project would have never been possible without valuable guidance and advices of Dr. Birsen Sirkeci. She acted as a mentor and guided us through various phases of project starting from literature review to performance measurements. She worked hard alongside our team and pushed us harder to achieve goals. We would also like to thank her for wise suggestions and answering our doubts, whenever needed.

We would like to thank IRIS’s Data services for permission to collect data and replying to our queries in timely manner. We would like to extend our gratitude to our counterpart Team in computer science department of SJSU and Dr. Yezu Gao for answering our queries and providing data for cross-validation.

The Electrical Engineering department deserves special thanks for providing us with platform and resources to conduct this research work and assisting us with academic paperwork.

Contents

[**Acknowledgement** 3](#_Toc513889635)

[**I.** **Introduction** 5](#_Toc513889636)

[**II.** **Seismic Waves** 6](#_Toc513889637)

[**A.** **P-Waves** 6](#_Toc513889638)

[**B.** **S-Waves** 7](#_Toc513889640)

[**III.** **Seismic Data Types** 7](#_Toc513889641)

[**A.** **SEED Format** 7](#_Toc513889642)

[**B.** **miniSEED Format** 8](#_Toc513889643)

[C. **Data-less SEED** 8](#_Toc513889644)

[**D.** **SEED Format** 8](#_Toc513889645)

[**IV.** **Event Triggering** 8](#_Toc513889646)

[**A.** **STA/LTA** 9](#_Toc513889647)

[**B.** **Delayed and Recursive STA/LTA** 10](#_Toc513889649)

[**V.** **Data Collection** 10](#_Toc513889652)

[**A.** **Noise Collection** 12](#_Toc513889656)

[**B.** **Data Visualization** 12](#_Toc513889661)

[**VI.** **Data Processing** 13](#_Toc513889663)

[**A.** **Data Clipping** 13](#_Toc513889664)

[**B.** **JSON Creation** 14](#_Toc513889665)

[**C.** **Noise processing and its challenges** 15](#_Toc513889666)

[**VII.** **Results** 16](#_Toc513889667)

[**A.** **Model Architecture** 16](#_Toc513889668)

[**B.** **Trials and Performance** 17](#_Toc513889669)

[**VIII.** **Conclusion and Future** 18](#_Toc513889670)

[**IX.** **REFERENCES** 20](#_Toc513889671)

# **Introduction**

Earthquake refers to ‘shaking of earth’s surface’ with sudden release of energy. These shakes could have different magnitudes which pertains to intensity with which earth shakes. Even though the mankind has faced this phenomenon from the beginning of time, seismic events are very under-researched in global scientific domains but, with advancements in Machine Learning, classification and analysis of seismic data is becoming more feasible. This project focuses on developing techniques to effectively classify seismic data into earthquake events and seismic noise where seismic noise can be caused by mining blasts, constructions or manmade explosions.

Various seismic data collection centers are set up by various seismological institutions such as IRIS that collects seismic data 24 hours a day and 365 days a year at many different regions. These data collection centers are called seismic stations. This collected data is not free of errors and require some sort of processing before being used in a classification model. We do not require whole 24-hour data, as earthquake occurs only once or twice (may be more times) a day. This requires triggering algorithms to be used by seismic stations to segregate actual earthquake data from continuous stream.

Pre-processing data demands filtering data to remove any kind of unwanted noise from seismic waveforms. But, in order to check actual performance of classification model, input data should include proper balance of both earthquake events and no earthquake (Noise) events. This project uses various methods discussed below to make sure that No earthquake data is free of any low intensity events.

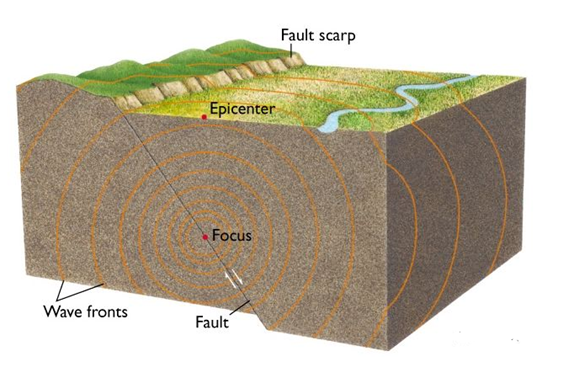


Figure - Earthquake Terminology

The earthquake is not a single event but consist of multiple small events combining together to form a destructive force. It consists of many energy waves where the most important elements are P-waves and S-waves. The data collection methodology that we use involves these waves and their time of arrival at the station. Arrival times can be extracted by using various Pick algorithms, which are not very credible as small change in threshold values of these algorithm can make non-earthquake event to be classified as earthquake event and finding these thresholds involve manual statistical calculations.

Main motive of this project was to defy these already accepted, less credible triggering (STA/LTA, Z Transform etc.) as well as picking (AR picker, baer picker etc.) algorithms [4] [5]. This was achieved by designing custom triggering and picking techniques which do not depend on manual thresholds and can extract arrival times as accurate as up to few nanoseconds. These techniques were incorporated in data collection phase and labels self-define P and S wave times by using start and end time of collected SAC files.

# **Seismic Waves**

Earthquakes are one type of seismic event while many other events such as mining blasts, underground tunnel digging as well as other man-made activities can cause earth’s surface to shiver. In this project, we will consider all other seismic events as noise except the earthquake. The body waves are emitted first in any earthquake. They attained their name from fact that body waves always travel below the surface of earth (just like inside body of earth). Their velocity is highest in seismic waves and they are first ones to arrive at any seismic station. These waves are further divided into 2 specific waves.

## **P-Waves**

First kind of body wave is known as primary wave or P-wave in short. The P-wave can move through solid rocks and any liquids in their path. P waves are fastest in terms of velocity of propagation and generally first to arrive at any station. It pushes and pulls the rock it moves through just like sound waves push and pull the air. It is analogues to hearing your windows rattling during loud thunders. P waves are also known as **compressional waves**. Particles subjected to a P wave move in the same direction as the wave i.e. the direction in which energy is traveling in as shown by Fig-2. [1]

## C:\Users\shaminder sandhu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.png

Figure 2- A P WAVE TRAVELS THROUGH A MEDIUM BY MEANS OF COMPRESSION AND DILATION. PARTICLES ARE REPRESENTED BY CUBES IN THIS MODEL. IMAGE ©2000-2006 LAWRENCE BRAILE

## **S-Waves**

Second type of body waves is known as secondary waves or S-waves. These waves are slower than P-waves (60% speed of p waves) and arrive after the p waves at a station. S waves move rock particles up and down, or side-to-side--perpendicular to the direction that the wave is traveling. S wave can only move through solid material and not liquid. [1] Fig-3 shows S-wave in action.

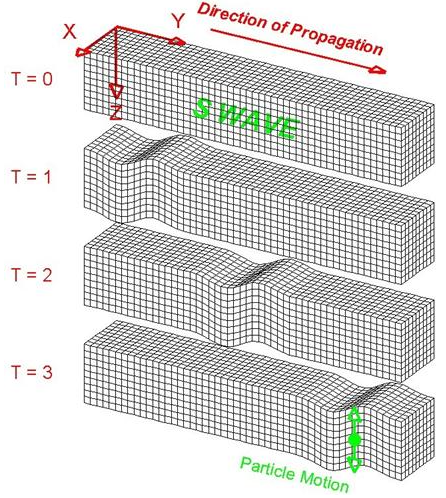


Figure 3 - AN S WAVE TRAVELS THROUGH A MEDIUM. PARTICLES ARE REPRESENTED BY CUBES IN THIS MODEL. IMAGE ©2000-2006 LAWRENCE BRAILE

# **Seismic Data Types**

Various seismology institutions around the world are dedicated to earthquake analysis and continuously collect seismic data throughout year. The data collecting centers consist of different seismic sensor networks called seismic stations. Seismic data is maintained by various international data centers such as IRIS DMC, GEOFON, ORFEUS DC, GEOSCOPE and other data centers of the International Federation of Digital Seismograph Networks (FDSN) [2].

These stations are maintained by different institutes and possibly collect seismic data in different formats but in order to achieve degree of correlation, most of data centers store data in popular seismic data formats common to all data repositories. These data seismic data centers can be contacted to obtain data for research purposes.

## **SEED Format**

The Standard for the Exchange of Earthquake Data (SEED) is an international standard format for the exchange of digital seismological data. SEED was designed for use by the earthquake research community, primarily for the exchange between institutions of unprocessed earth motion data [3]. A full SEED format consists of time series values along with metadata. Metadata of SEED consists of information such as P, S wave arrival time, sampling rate, station name, station location etc. Detailed information and characteristics can be read in SEED manual at [3].

SEED format is not designed for processing thus it should be first converted into other processing friendly formats such as SAC, before supplying to python or any other program for processing.

## **miniSEED Format**

MiniSEED is also known as data only part of SEED file. It means that it has only seismic waveform with very basic and necessary metadata (also called header or additional information part) like station name, channel code, start and end time of waveform. Metadata of miniSEED is generally supplied separately with this file. Blocks of data in miniSEED are known as ‘records’ and record lengths are multiple of 512 bytes. 512 bytes is length of real time data blocks with 4096 is length of archived data blocks. These records are combined together to create a miniSEED file. miniSEED is also storage friendly but not processing friendly [2] [3].

## **Data-less SEED**

As name of this file is self-evident, this type of file has only header information about a seismic waveform and no waveform data. Dateless SEED is always combined with miniSEED file for completion. It was created to facilitate storage as well as data corruption issues during transmission. In order to convert miniSEED into other processing friendly formats, dateless SEED is used to create headers, while it can also be used to cross validate data in full SEED files. It does not contain any time series value [3].

## **SEED Format**

Seismic analysis code (SAC) is a processing friendly file format. There are many processing friendly formats such as SEISAN, GSE, GSE2 etc., but we will be using SAC files in our project. SAC file consists of 2 parts- header and data part. Header consists of side information such as channel type, station name, location, P-wave start time, S-wave start time, no. of samples and sampling rate etc. The data part contains actual sampled version of seismic waveform which can be processed using common processing modules like Obspy in python. A SAC file provides near to complete package for effective seismic studies.

Each station collects same data from different directions and angles called channels. There are many different channels related to a seismic waveform, but we will be focusing on mainly three- BHE, BHZ and BHN.

This project relies on credible repository of IRIS DMC and uses IRIS’s Wilber3 tool to collect data. The samples collected are in SAC format and contain additional information in headers which can be used as extra features for classification models.

# **Event Triggering**

Triggering algorithms were developed to make sure that recording of seismic activity starts only when an event is triggered (occurs), thus keeping stations dormant for most of time. Real time trigger algorithms keep on monitoring incoming data and activates storage device on station when signal meets criteria set by algorithm (thresholds).

With technological advancement, we now have many storage solutions. Stations nowadays, record data continuously for seismic research purposes. Trigger algorithms are now used during processing of seismic data to distinguish noise from earthquake events, also called denoising of waveforms. These algorithms may result in false triggers due to noise spike exceeding threshold set by algorithm. There is no universal trigger algorithm but a good algorithm must create least number of false triggers and effectively differentiate between event and noise [4].

## **STA/LTA**

## Short time average/ long time average is most widely used trigger algorithm in seismology domains. The working of STA/LTA is quite simple. It processes signal in two moving time windows known as short time average window (STA) and long time average window (LTA). Duration of STA window (let’s say 3 sec) is very small as compared to LTA window (let’s say 24 sec). Average of absolute values of signal in these two windows is calculated and ratio STA/LTA is maintained and changes with each iteration. STA/LTA uses two different thresholds- trigger threshold (TT) and de trigger threshold (DT) [4].

Whenever ratio STA/LTA goes above TT, an event is declared, and trigger is said to be occurred. When ratio falls below DT, event is terminated [4].

Fig-4 represents working of STA/LTA algorithm.

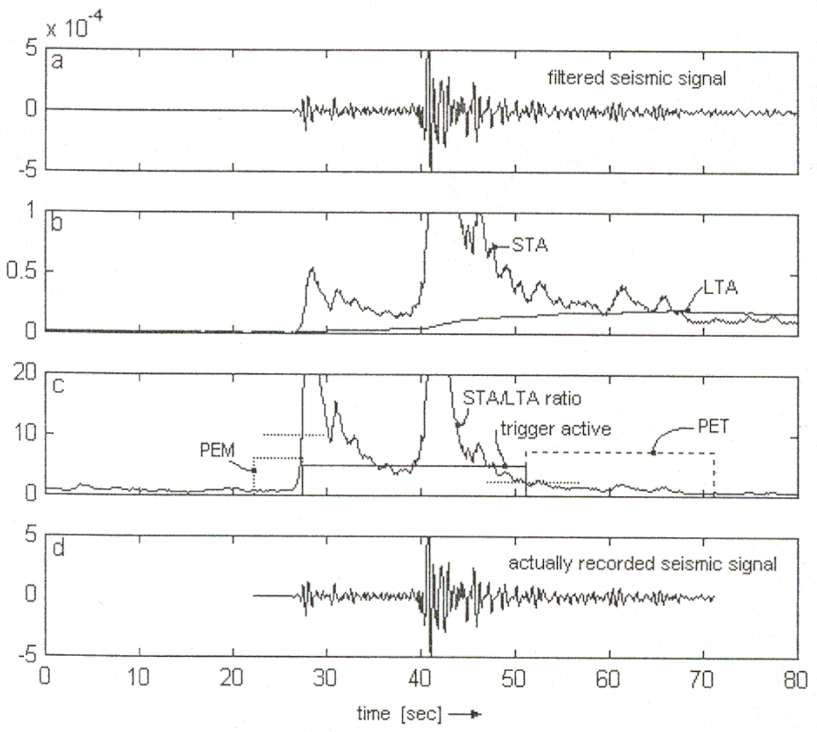


Figure 4-STA/LTA algorithm [4], 2 horizontal dotted lines represent thresholds

PEM and PET are called pre-event time and post event time which are also recorded along with event to avoid missing any subsidiary information [4].

STA/LTA window sizes and trigger thresholds are defined by user to adjust sensitivity of algorithm. The selection of these parameters is discussed in [4].

## **Delayed and Recursive STA/LTA**

## In standard STA/LTA, the LTA window starts 1 sample after STA window. For better statistical independence, windows can be delayed by a finite number of samples between them (say 100 samples). Delayed STA/LTA increases the memory usage as system has to keep track of not only window times but also difference between them with each iteration [5].

## The recursive STA/LTA is similar to the standard STA/LTA except that for each successive time step, a fraction of the average data value, rather than a specific data point value is removed [6].

C:\Users\shaminder sandhu\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.png

Fig-5 shows working of recursive STA/LTA on sample waveform.

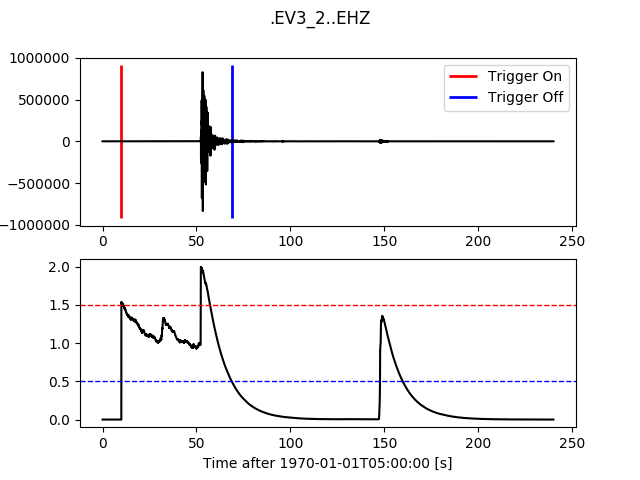


Figure 5-implementing recursive STA/LTA in python

# **Data Collection**

# The data collection is major factor in this project as new techniques were developed to eradicate usage of triggering and picking algorithms. Seismic waveform data is collected in SAC format which is machine learning friendly as many python libraries are available to process SAC waveform. This project makes use of obspy module to visualize, read, write and filter seismic waveforms.

# Iris DMC’s wilber3 tool is used to collect seismic events as well as noises (No earthquakes). The wilber3 repository considers earth as flat model (1-D) while collecting data from International Federation of Digital Seismograph Network (FDSN) stations. Data is collected in California and neighbor regions. Different institutions have their own registered stations approved by FDSN. The data is collected from stations registered to networks **AZ (Anza Regional Network), BK (Berkeley Digital Seismic Network), CI (Southern California Regional Network)** and **NN (Western Great Basin/ Eastern Sierra Nevada)**. All of these networks exist in California thus, making sure that occurring event is close to ample number of stations

# The data is collected around the co-ordinates **N = 44.83, W = -136.33, E = -110.05, S = 31.11**. There are total of 65 events in 2016 around this location. All of these events have magnitude equal to or greater than 4.0 on Richter scale. As number of stations is already restricted by networks, 75 (±1) stations collect each of 65 events. Each station collects 3 SAC files for an event pertaining to channel BHE, BHZ and BHN thus making total number of collected SAC files equal to **65 x 75 x 3= 14625.** Figure shows 65 events collected in California and adjacent regions.

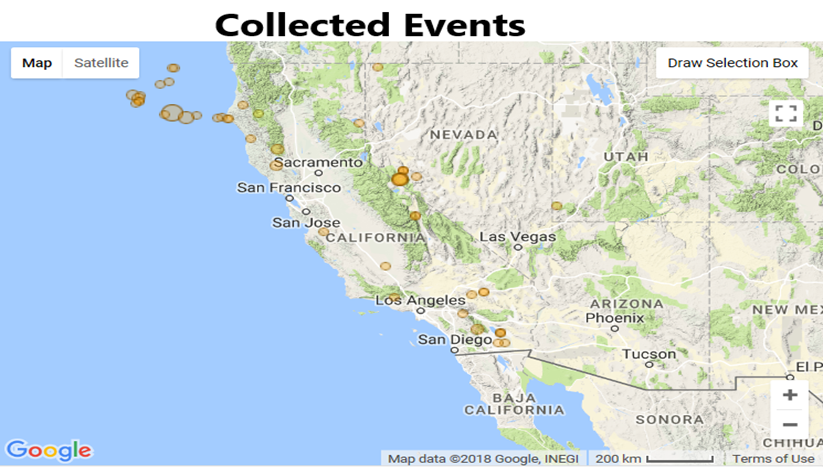


Figure 6- Collected Events in California Region ©Wilber 3

In order to extract P and S wave arrival times as well as other important features from collected waveforms, 3 different waveforms were collected for each event per station. These waveforms were collected as:

P wave= 1 min before p to 1 min after p arrival.

S wave= 1 min before s to 1 min after s arrival.

Event= 1 min before event to 2 min after event.

Event file contains whole earthquake data, which will be used for classification while P and S wave files can be used to extract P and S arrival times by just adding 1 minute to start time of SAC file. Similarly, Event time (Time at which earthquake happens) can be extracted by adding 1 minute to start time of event SAC file. These times can become additional features to be used in classification model and to ultimately predict P and S arrival times in real time data to predict occurrence of an earthquake.

## **Noise Collection**

## In any good classification model, samples of events (Earthquake) and no events (Noise or very low amplitude earthquakes) must be fairly equal, or at least be in good proportion [13]. In order to achieve this, we collected noise samples for each of 65 events as well. It is fairly assumed that before earthquake happens, the station just collects noise data. This noise data might be very low amplitude earthquakes (≤ 1.0), man-made underground noise, mining blasts or bomb testing etc. so, noise is collected as:

## Noise= 5 min before event to 0 min after event.

## While processing data one theory was that, as Iris only concentrates on collecting earthquake events, noise collected might contain some high amplitude samples related to some other event that IRIS was not considering at the moment. This can reduce performance of model and can give totally unexpected results.

## In order to be on safe side, we obtained alternate noise data recorded continuously at 16 stations in Sichuan region in China. These samples were collected in absence of any earthquake and thus contain only pure noise samples. As it is continuous stream of data, each SAC file is 24 Hour long and has sampling rate different than IRIS data. This data can be resampled and used to cross-validate final results.

## **Data Visualization**

## The data collected is in SAC format, which stores complete waveform of an earthquake signal sampled as amplitudes of that waveform. Fig-7 shows plot of a collected waveform in SAC format.

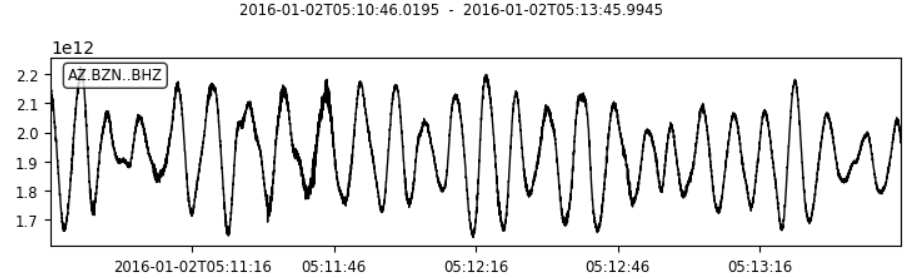


Figure 7-Plot of one earthquake out of collected data.

The SAC format also stores other valuable information along with waveform, called header. Header portion of SAC contains side information about station as well as event itself. Fig-8 shows header information of one of the collected events.

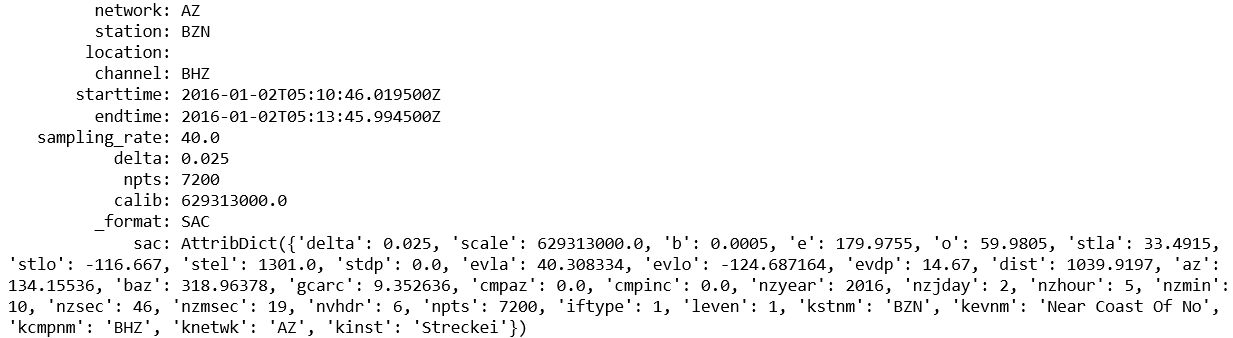


Figure 8- Header portion of a collected event

Header part includes information such as station name, station location, network to which it is registered, start and end time of SAC file as well as location of epicenter as well. Some of these attributes can be used as additional features to be used in classification model. Feature selection is a tricky as some features such as locations as well as channel names can help model to be more generalized while other features can decrease performance.

# **Data Processing**

## **Data Clipping**

The data collected (Events) is between 3-4 minutes of length, depending on duration for which an earthquake lasts. IRIS samples data at 40 samples/sec and a 3-minute-long waveform produces 7200 samples. If each sample of waveform is considered a feature it would mean that machine learning model has to process 7200 features with each feature having thousands of data points. Processing this amount of data is not impossible but requires rather long time and processing power.

In order to tackle this, it is more convenient to clip full waveform into smaller time windows cut in consecutive order from original longer waveform. It is also possible to shift focus from one window to another and find out that around which time effect of earthquake was strongest (Probably close to arrival of S wave). The event data was clipped into 3 slices consecutively, each of duration 10 seconds and starting from event start time (earthquake occurrence). Each of slices has 400 samples (40 samples/sec). This means that each SAC file pertaining to event per station produces 3 new SAC files (Slice1, Slice2, Slice3) as shown in Figure-9.

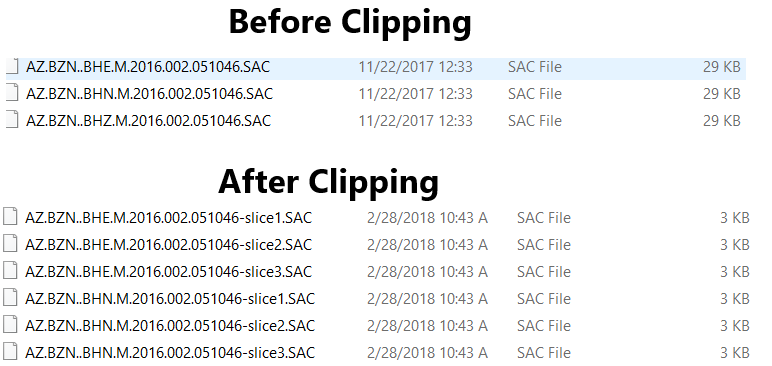


Figure 9- clipping SAC files

The names and headers of clipped waveforms were kept same in order to extract and use side information as additional features later on.

## **JSON Creation**

Next step in data processing is to convert raw waveform data into machine learning friendly format such as CSV or JSON so, that it can be easily fed into model. JSON file can be easily written and read without processing data again from beginning. All SAC files are sampled at 40 samples/sec (same as IRIS’s sampling) to convert into numeric amplitudes of waveform as shown in Figure-10.

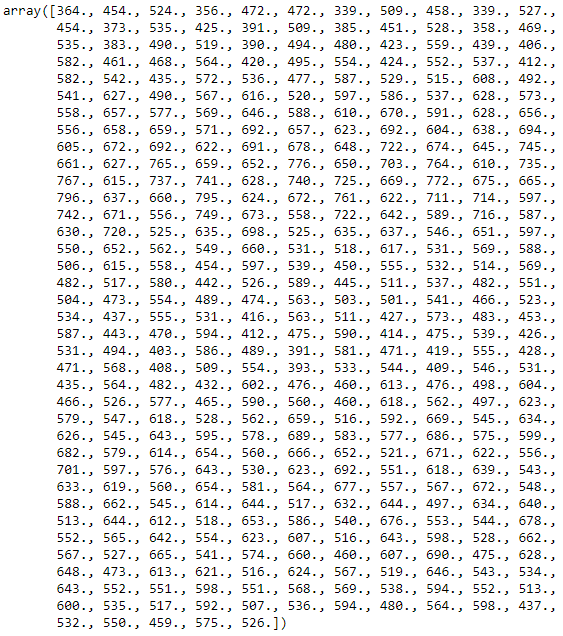


Figure 10- Amplitude values of 10 sec window

We have total **14625** SAC files, which when divided into BHE, BHN and BHZ channels comes out to be **4806** events, after deleting some stations with incomplete data. After converting into arrays of sampled waveforms, our algorithm divides data into 3 channels (BHZ, BHE, BHN) as well as their corresponding 10 second windows (Slice1, slice2, slice3). This data is put into a dataframe and appended with **8000** noise samples at end. Noise is put at end of dataframe so that it can be easily replaced with alternate noise samples if required. Ratio of event to noise samples in original data is 4806:8000.

The processed data is stored as JSON file and contains a label called **is\_EQ** which is 0 for non-earthquake data and 1 for earthquake event data as shown in figure-11.

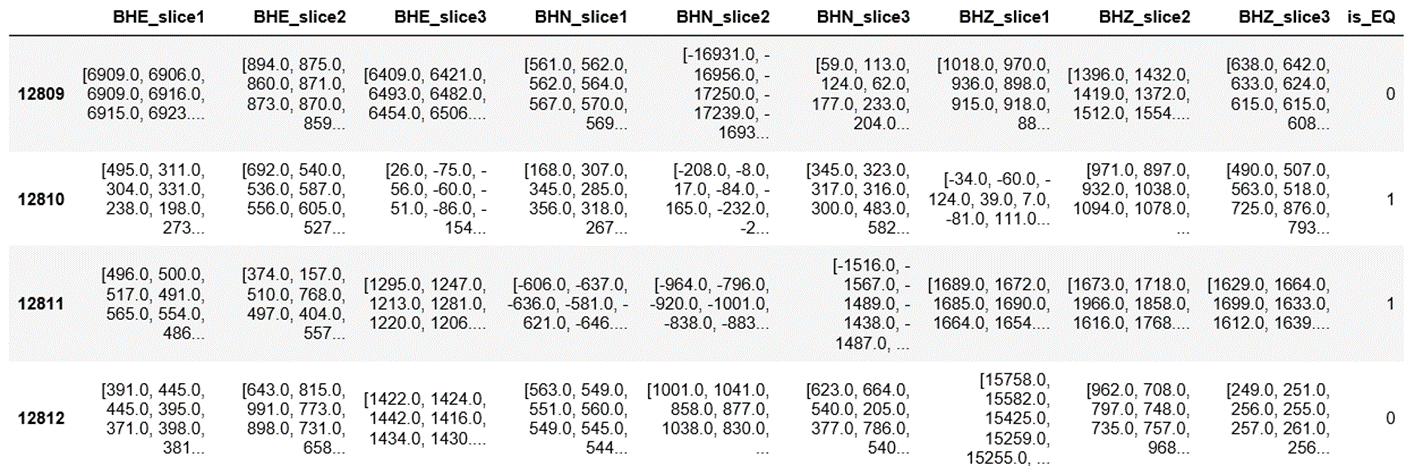


Figure 11- JSON file containing events and noise data

## **Noise processing and its challenges**

The noise collected is processed similarly to rest of data with only exception being that number of slices were chosen to be 5 instead of 3, for events. It means using 50 seconds of collected noise waveform. It is done to increase number of noise samples to 8000 which is nearly double than event samples (4806). The reason to use nearly double noise samples is that collected noise waveforms had some abnormalities in them. These abnormalities were detected as data with original noise was used to train initial 1-D classifier and performance of model came out to be worst than expected. When noise data was replaced with pure noises (from Chinese stations), the performance of same model increased substantially.

The problem with noise samples was that some of samples had very high amplitudes (even higher than Events), which was causing models to wrongly classify them as events and hence decreasing performance measures. In order to eradicate these rogue noises from data, scaling techniques were modified as per need. The noise samples were scaled using standard scaling techniques of scikit learn and scaling coefficients were saved. These scaling coefficients were used to detect noise samples having high amplitudes, comparable to earthquake events. These noises were then deleted from data using custom written algorithm. After processing original 8000 noise samples, the algorithm deleted approximately half of the samples and leaving data with event to noise ratio of **4806:4000**, which is perfect proportion to be used in classifier model.

The other solution to this problem was to use pure noise (Chinese station) data instead of collected noises. This approach shows very good performance with all the models but it is more like picking oranges from apples and is far away from real life scenarios. The scaling approach is analogous to picking rotten apples from batch of apples and is very similar to real time data, which is handled by most of seismic stations around the world.

# **Results**

## **Model Architecture**

The processed data was used to train many 1-D standard classifiers such as Linear Regression, Random forest and Support Vector Machines etc. but major goal of project was to develop a high-performance Convolution Neural Network (CNN). It is due to fact that all these classifiers accept only 1-D input so, only one channel data can be used for training and testing. In order to achieve high performance all three channels should be used as they collect data from different angles and hence probability of error diminishes greatly.

The designed CNN has 7 convolution layers where, each convolution layer performs 1-D convolution of input data. Each layer uses RELU activation function with stride length of 2. Kernel size is kept as 3. The convolution layers are followed by single Dense layer which uses sigmoid activation to predict probability of given input to be earthquake event (1) or non-earthquake event (0). The probability close to 1 will result in event being classified as earthquake and vice versa for noise sample. Figures-12, 13 represents CNN architecture and layer information.

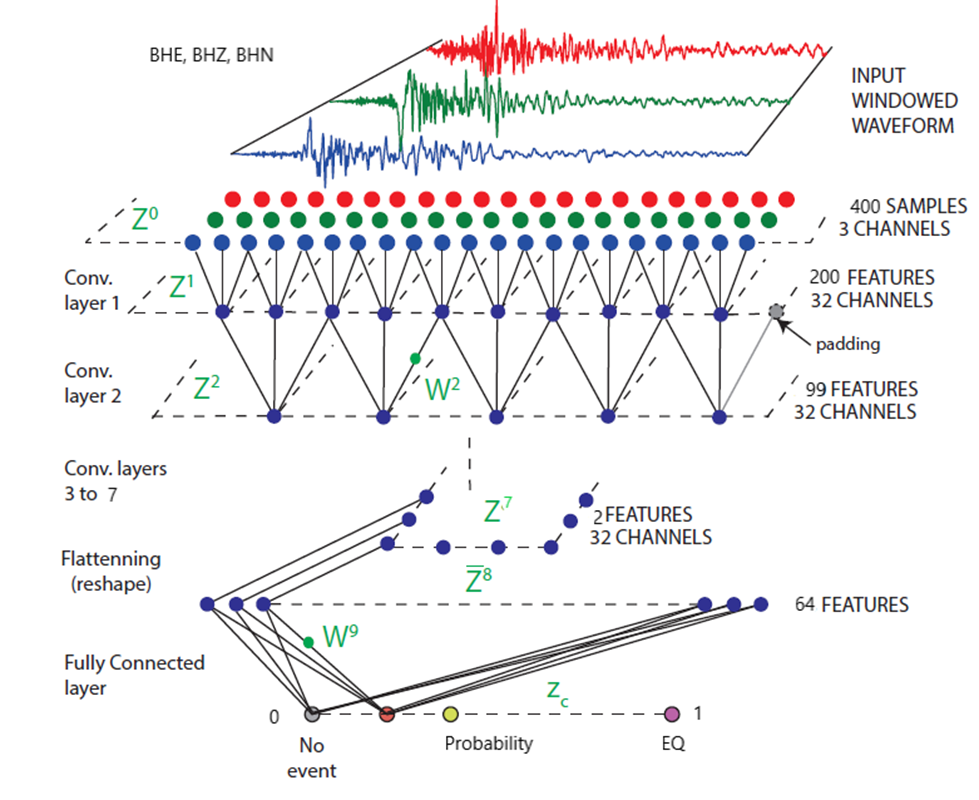


Figure 12- CNN Architecture [12]

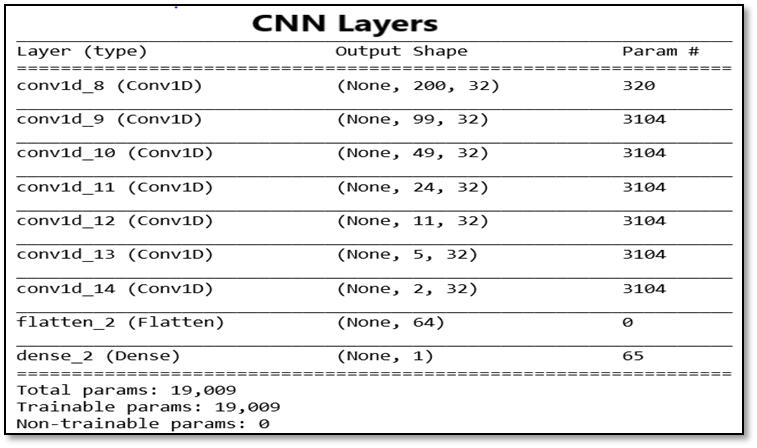


Figure 13- CNN layer information

The input to CNN is 3-D shaped tensor which contains all three channels (BHE, BHZ, BHN) stacked in parallel. Data is reshaped and stacked to give parallel channel input to CNN. Using all three channels is one of the reasons for high accuracy of CNN.

## **Trials and Performance**

The CNN described above uses early stopping criteria based on validation loss to avoid overfitting and saves best parameter weights to be used for test data or handling different data sets in future without requiring to re-train model. Train to test data ratio is 75:25 which means, 6604 samples were used for training and 2202 samples for testing performance. The training set is further divided into 75:25 as train and validation data.

As problem posed is binary classification in nature, it is advised to use multiple performance metrics to evaluate performance od model. It also helps to understand CNN better as model is probabilistic in nature. We used four different metrics to evaluate performance: Binary Accuracy, Precision, Recall and F measure score. The latter three were custom written to be used with Keras CNN. The loss function used in CNN is Binary Cross entropy which also helps with early stopping mechanism.

This project and its efficiency is mainly dependent on credibility and proper processing of data. In order to demonstrate different data scenarios, we used various combinations of data as well as models to find out best performing model.

As 1-D classifiers accept one channel input, the BHZ channel shows best performance as compared to BHE and BHN. It is in accordance with theory as BHZ channel captures strongest effect of any seismic event. The performance of various models is shown in Table-1.

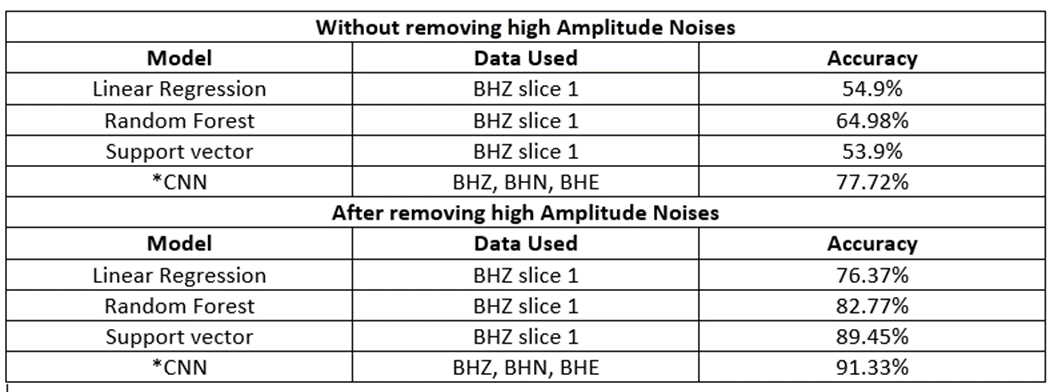


Table 1- Model Performance Comparison

The high amplitude noises in original data play major role in deciding performance of any model as it is clear from Table 1. The accuracy shoots up exponentially after removing high amplitude noises from data. The accuracy shown in Table 1 is binary accuracy for all the models for comparison. Detailed metrics for CNN are displayed in Figure.

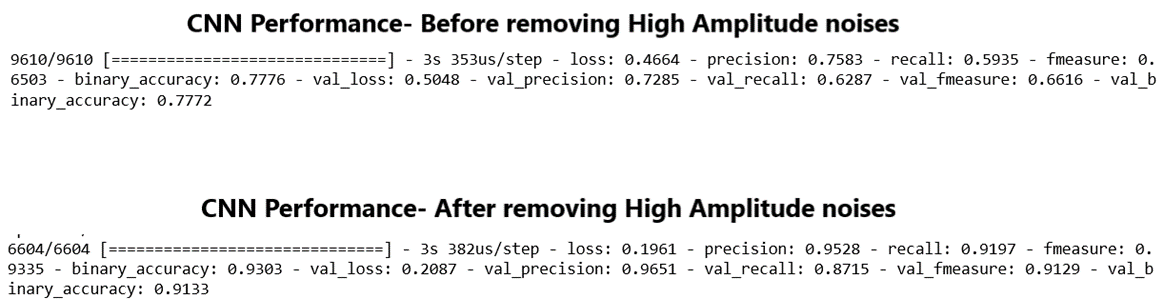


Figure 14-CNN Performance

# **Conclusion and Future**

The CNN built for classification remains state of art and excels in almost every performance metric as illustrated in results. Adding Pooling layers have additional positive effect on performance but it reduces number of features extracted for next layers. It can be tackled by reducing number of convolution layers and remain topic for future research. The different approach for triggering and picking methods used in this project reduces dependency of research work on error prone algorithms as thresholds in trigger/pick algorithms are hard to maintain.

The effective collection of data and noise processing remains interesting research topics to pursue for future. The performance majorly depends on effective removal of high amplitude noises. The algorithm used to remove rogue noise demands a threshold value for scaling coefficients which is calculated after complex statistical computations. This algorithm can be further improved by automating threshold selection for different data sets. Algorithm also shows better performance if co-relation between collected events is reduced.

Research work can be further modified to generalize better for seismic events by including additional features such as ground water well level as well as environmental temperature when earthquake occurs. These two features show minor/major change in case of earthquake event whereas amount of change depends on numerous factors such as epicenter location as well as magnitude of event. There is ample evidence that seismic events also effect temporal magnetic field of earth. Inclusion of these features require further research and studies to find out co-relation between amount of change and seismic event. It remains interesting topic to pursue for future researches.

# **REFERENCES**

[1]. “Seismology introduction for budding seismologists”, Michigan Technological University, 2007.

[2]. “Seismic waveform data retrieval”, Joachim Saul, GFZ German Research Centre for Geosciences, December 2014, DOI: 10.2312/GFZ. NMSOP-2\_IS\_8.3.

[3]. “SEED Reference Manual Version 2.4”, Standard for the Exchange of Earthquake Data, International Federation of Digital Seismograph Networks Incorporated, August, 2012.

[4]. “Understanding and parameter setting of STA/LTA trigger algorithm” Amadej Trnkoczy (formerly Kinemetrics SA), September, 1999.

[5]. “A comparison of select trigger algorithms for automated global seismic phase and event detection”, Richard C Aster, Colorado State University, Article in Bulletin of the Seismological Society of America, February 1998.

[6]. “Evaluation of Seismic Events Detection Algorithms”, B. K. SHARMA1, AMOD KUMAR1 and V. M. MURTHY, Central Scientific Instruments Organization, Chandigarh Kurukshetra University, Kurukshetra.

[7]. “Trigger/Picker Tutorial”, ObsPy Documentation (1.1.0)

[8]. “Malaysia Monitored for Earthquakes by 25 Sensors”, Martin Carvalho.

[9]. “Machine Learning for Seismic Signal Processing: Seismic Phase Classification on Manifold”, Juan Ramirez Jr. and Francois G. Meyer.

[10]. “The discrete wavelet transform: Wedding the `a trous and Mallat algorithms”, M.J. Shensa.

[11]. “Multi-Channel Fusion for Seismic Event Detection and Classification”, Ofir Lindenbaum, Neta Rabin, Yuri Bregman, ICSEE International Conference on the Science of Electrical Engineering,2016

[12] “Convolutional Neural Network for Earthquake Detection and Location”, Thibaut Perola, Michael Gharbib, Marine A. Denollec, 7 Feb 2017

[13] “Hundreds of earthquakes per day: The 2014 Guthrie”, M. Benz, N. D. McMahon, R. C. Aster, D. E. McNamara, D. B. Harris, Oklahoma, earthquake sequence, Seismological Research Letters 86 (2015) 1318{1325.