MATAM Group 2 Summer Research Report

Halil Ibrahim Kanpak, Mustafa Meric Kasap, Kubilay Payci

September 2024

Abstract

This report presents the research activities of group 2 (Halil Ibrahim Kanpak, Mustafa Meric Kasap, Kubilay Payci). Our work is focused on forecast of earthquakes in a given space and time based on the catalogues reported. This document provides an overview of the project, methodologies employed, results obtained, and future work. Here is the link for our codebase :
https://github.com/Mustafa-Meric-Kasap/AnalysisOfEarthquakeCatalogues.git

1 Introduction

The summer research project conducted by MATAM Group 2 aimed to forecast potential earthquakes based on historical earthquake records from a catalog. The team, comprising Halil Ibrahim Kanpak, Mustafa Meric Kasap, and Kubilay Payci, explored the application of machine learning, statistical methods, and deep learning techniques for data classification and forecasting. This report summarizes the collective efforts and outcomes of our research.

Our objective was to predict the characteristics of potential earthquakes given the historical earthquake data for a specific region, leveraging a multidisciplinary approach that integrated statistics, machine learning, and geology.

During the summer, the team engaged in various activities, including the preprocessing and enhancement of datasets, application of multiple machine learning algorithms, hyperparameter tuning, and evaluation of the performance of machine learning models.

2 Prior Work

3 Method

We employed a diverse array of statistical methods to understand and forecast the data. Specifically, we utilized clustering techniques to partition the dataset into meaningful subsets, representing groups of related earthquakes. We then applied a metric function to project the dataset features into a one-dimensional space, which facilitated the training of basic machine learning models. Additionally, we optimized the metric function through hyperparameter tuning to better capture the underlying patterns within the data. Finally, we trained deep learning models using the data subsets derived from the clustering algorithms.

3.1 Data sets

We have used 3 data sets consisting of:

1. KOERI BOUN data set restricted to Turkey (lat: 36-42, lon: 26-45) from 1900 until today
2. KOERI BOUN data set on global scale off all times
3. AFAD earthquake data set restricted to Turkey (lat: 36-42, lon: 26-45) from 1900 until today

4. PRF (Pacific Ring of Fire) data set is data set restricted to Pacific Ring of Fire fault zones which known for its frequent high magnitude earthquakes.

Firstly we have trained and validate our models on all three of data sets. After conducting various experiments we decided on working on the first data set of KOERI on Turkey as it contains the most precise records for us to train our data. One important topic on choose of data sets is the term "Magnitude Completeness". Magnitude Completeness refers to the homogeneity of earthquake data. With the advancement of technology records of small earthquakes becomes more common leading to a false observation of increment of occurrence of those small earthquakes. This false observation leads models to a perform poorly and more importantly perform on a false basis.
To prevent this we have also employed another function to "homogenize" an arbitrary earthquake catalouge. How this function works is that it takes the minimum occurence of all the magnitudes on bins of time intervals and takes out earthquakes randomly from other bins. This way magnitude completeness is achieved. One may choose to pick on a rule instead of randomly taking out earthquakes from bins of time intervals to deterministically create data sets but we have choose to take earthquakes out randomly to create dataset out of biases.

3.1.1 Clustering

Clustering is another integral part of our project. One of the main problems occurring in training is that the ML models. We have used 3 clustering algorithms:

- **Space-Time Clustering:** This approach picks an earthquake and for a given radius, time interval picks earthquakes from the catalogue that are within the radius from the chosen earthquake event in specified time interval.
- **K-NN Clustering:** This approach uses the K-NN algorithm on features of earthquakes in catalogue. In this setting we have picked various different features to conduct the K-NN algorithm which is included as another hyper parameter for our models.
- **Spectral Clustering:** Just as similar to K-NN algorithm we also have employed spectral clustering to enhance the neighboring effect and select earthquakes that are incident with one another on geographical features. What we aimed on using spectral clustering is that we have observed that it performs better on selecting the earthquakes that are related to one another respect to fault lines which is a physical reality that K-NN misses.

What we aim in clustering algorithms throughout the project is that, instead of considering an entire data set of earthquakes wholly dividing it into earthquakes that are related

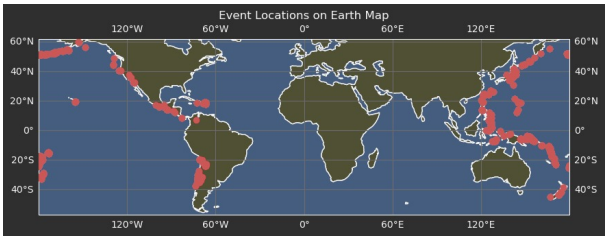

Figure 1: Pacific Ring of Fire earthquakes Labelled as "risky" that we have obtained with KOERI dataset.

to one another helps for us to understand the earthquakes relations and train the ML models effectively.

3.1.2 Earthquake Batches

Using our clustering algorithms and building on our motivation related to earthquake clusters, we aim to create a dataset with a more convenient structure derived from earthquake catalogs. We first homogenized our data with respect to magnitude completeness as a starting point. We then selected earthquakes with magnitudes above a certain threshold, labeling them as "risky," and another set with lower magnitudes, labeling them as "safe." This approach was intended to create true and false labels for our classification algorithms, with a magnitude threshold of 5.5.

After selecting earthquakes from the homogenized datasets, we identified the last s earthquakes that occurred closest to the selected event within each cluster, forming a time series of earthquakes. The structure of our dataset now consists of s earthquakes selected based on a clustering algorithm, with labels either "safe" or "risky" according to the initial chosen earthquakes. We also considered the potential conflict of aftershocks among earthquakes; thus, when creating a series of earthquakes, we excluded aftershocks before forming the series.

We primarily formed our earthquake series datasets from the KOERI dataset for Turkey and the Pacific Ring of Fire. This choice was made to understand the physics of earthquake incidents in Turkey. However, due to the relatively small number of earthquakes in Turkey, we also included data from the Pacific Ring of Fire to obtain a more homogeneous dataset with respect to magnitudes and to test model performance on different earthquakes.

It is important to note that during the dataset formation process, we opted to store only the differences between consecutive earthquakes rather than all the information from each event. This approach was chosen to improve memory efficiency.

3.2 Metric Function

We have employed a "metric function" for using its image as a feature space for our ML models. How this metric function works is that it takes an earthquake catalogue data set. Preprocess all the features accordingly on our prior knowledge of features (converting time data to timestamps, change the scale of magnitudes etc.) and do a weight sum of all the features with predetermined weights giving the result of this sum as "score".

Reasons why we use a metric function instead of dimensionality reduction algorithms or using data raw on ML models is that:

- Metric function allows us to modify the scale of features freely independent of our models. As an example magnitude of an earthquake grows on a logarithmic scale whereas it's time of event is on a time scale and it's location of event is obtained using global coordinates. Using this metric function it is possible to accord them all on a linear scale and treat accordingly, normalizing and weighting afterwards regarding to the results.
- Using this scaling among the features we are also able to do a weighted sum of them, giving us a "score" of that earthquake for us to use it in our classification of earthquakes algorithms on simpler ML models.
- Using data raw on ML models works poorly as it approaches the problem independent of our prior knowledge on earthquake catalogues. This models perform poorly on forecasting event places of earthquakes, magnitudes and extracting the relation among the earthquakes. This is because ML models statistically approaching the situation predicting the mostly occurred earthquakes in average of event places to lower the loss function. Using various other loss functions or techniques such as resampling earthquakes (which is also needs to be justified as it allows model to predict higher magnitude earthquakes without any ground of knowledge.) for training.

3.2.1 Hyperparameter Tuning

We have utilized optimization search algorithms to find optimal weights for our metric function. Which made models able to extract the determinant information of the features of earthquakes. We have used bayesian and grid search to find best weighting parameters

3.3 Models

3.3.1 ML algorithms

We have used and tried various ML algorithms to find the optimal. After considering problem with batches introduced we focused on algorithms that includes a memory and extracts the serial information from the data. This has been achieved with broad range of ML algorithms ranging Convolutional Neural Networks (CNN) and LSTM's. Most important problem to be addressed in our training is the problem of overfitting the frequent data. This is important as the main aim is to detect the earthquakes that are bigger in magnitude. To resolve this we have used various resampling techniques and avoid using data augmentation. This way of approach is the indeterminacy and the existence of the fact physics involved in the earthquakes. Data augmentation was creating unrealistic earthquake scenerios which one may not be able to determine the sanity. This is why we have used resampling. One important thing here is the fact that overfitting a particular earthquakes physics.

3.3.2 Model selection

We have used literature and optuna framework to find best models and parameters for our data sets that we have created out of catalogues.

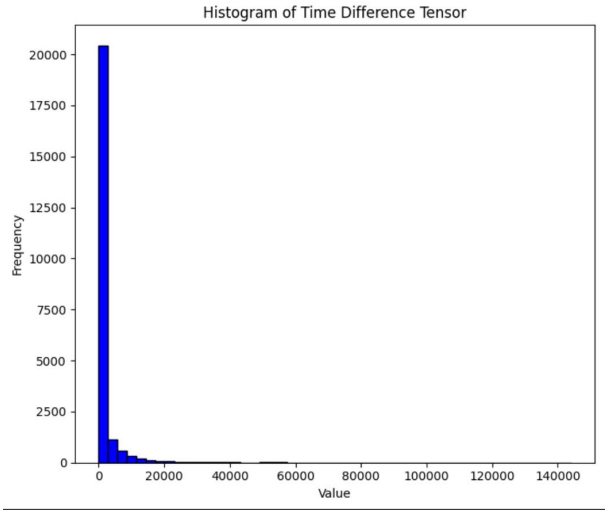

Figure 2: Reaction time of our earthquake series. Reaction time refers to the time gap among last earthquake in series and labelled earthquakes difference in time.

4 Experimental evaluation

We conducted our experiments using the NVIDIA Tesla T4 GPU provided by Google Colab, which features 16 GB of GDDR6 memory. The system was equipped with an Intel Xeon processor and 12.7 GB of RAM, allowing for efficient training and evaluation of deep learning models. For the sake of simplicity, we have chosen experiments that best describe our improvements and models that provide the best results given the specifications. With all ML models ranging from a broad family from random forests, simple linear regression models to graph neural networks that we have experimented we also have evaluated our data sets as they are the integral component of this work. One other important thing in our approach is that with our data set of earthquake series how do we react in time. That is after a sequence what is the expected time of earthquake event. We also have evaluated our data sets regarding to this ,with outliers also occurring in this case too, we found out that sequences alert in average of 3 days for earthquakes labelled as "risky" in magnitude.

Model	**Dataset**	**Epochs**	**Loss**	**Validation Loss**	**ROC-AUC**	**Notes**
LSTM	KOERI	10	0.0014	0.0135	0.9987	Overfitted on frequent data, used earthquake batches
LSTM	KOERI	10	0.0007	0.0031	1.0000	Worked better on imbalanced data with oversampling
GRU	KOERI	10	0.0015	0.0019	0.9998	MLP based GRU model
CNN	KOERI	100	0	0.0308	0.5598	Did not use oversampling
CNN	KOERI	100	1.9829	1.6605	0.9729	model's performance has deteriorated with epochs (Adam)
CNN	KOERI	100	0.0039	0.0024	1.0000	Used dilation layers to extract information of earthquake series
CNN	KOERI	100	0.1671	0.1902	0.9950	Worked best with Adagrad optimizer
Random Forest	KOERI	-	-	0	1.0000	Used oversampling with naive rf algorithm
Random Forest	KOERI	-	-	0.0020	1.0000	Used oversampling, restricted maximum depth of trees
LSTM	PRF	10	0	0.0011	0.997	Used homogeneous data respect to labels
LSTM	PRF	10	0.0552	0.0322	0.9997	Used binary cross entropy to better extract classification data
GRU	PRF	10	0.0264	0.0107	1.0000	MLP based GRU model on homogeneous data
CNN	PRF	100	0.0018	0.0039	1.0000	Used CNN on homogeneous data
CNN	PRF	100	0.5586	0.5577	0.9213	Experiment with Adagrad optimizer yet performed poorly
CNN	PRF	100	0.0022	0.0090	1.0000	Used dilation layers
Random Forest	PRF	-	-	0.1177	0.9310	Restriction on depth
Random Forest	PRF	-	-	0.0957	0.9605	

Table 1:
KOERI dataset refers to earthquake data from Turkey, collected between 1990 and 2024.
PRF dataset refers to Pacific Ring of Fire region between 1993 and 2024 (again collected with KOERI's catalogue).
For further specifications on models please check models notebook file on repository.

5 Conclusion

As a result we have managed to develop an "Earthquake Detection and Alerting" system thanks to Machine Learning and Statistics approach with patterns and information identified in earthquake catalouges. Main challange was to base and relate findings of statistics with physics of the geology and create meaningful deductions and criticize results of Machine Learning models properly. Hence we have decided the best model in practice would be an alerting system on an arbitrary fault zone. What we propose is not a particular model but an comprehensive approach for such an detection and alerting system. Now using our system one may pick an arbitrary fault zone and conduct our earthquake series approach with a chosen clustering method of ours. Form a data set out of that earthquake catalogue. Finally conduct the appropriate machine learning approach using the empirical knowledge we introduce in table in section 4. With that one is able to obtain a model that is able to detect the earthquakes of a given region and is able to be updated with newly given earthquakes.