Earthquake Prediction using XAI

Rishabh Jain

IIT Delhi
Research Intern
National University of Singapore

Guide: Vishal Srivastava

National University of Singapore

Summary Report

3rd July 2024

National University of Singapore

Outline

1. VAN Method for EQ Prediction
2. Models on our Data
3. Next Steps

VAN Method for EQ Prediction

Novelty: -VAN team (Athens) first to establish a relations between *seismic electric signals* and Earthquakes. -Claimed they were able to predict earthquakes of magnitude larger than 5, with an uncertainty of 0.7 units of magnitude, within a radius of 100 km, and in time window ranging from several hours to a few weeks.

Methodology: -Introduced the concept of *natural time,* a time series analysis technique which puts weight on a process based on the ordering of events.

-Two terms characterize each event, the "natural time" X, and the energy Q_k. X is defined as k/N, where k is an integer (the k-th event) and N is the total number of events in the time sequence of data.

They introduced a critical term κ, the "variance in natural time" :

$$\kappa = \sum_{k=1}^{N} p_k (\chi_k)^2 - \Big(\sum_{k=1}^{N} p_k \chi_k\Big)^2$$

where $\chi_k = k/N$ and $p_k = \frac{Q_k}{\sum_{n=1}^{N} Q_n}$

Observation: The specific value of κ **= 0.070** has been empirically determined by the VAN team as a critical threshold. At this value, earthquake is claimed to be imminent after a second analysis.

Continuation

Results: The VAN team claim that out of seven mainshocks with magnitude Mw>=6.0 from 2001 through 2010 in the region of latitude N 36° to N 41° and longitude E 19° to E 27°, all but one could be classified with relevant SES activity identified and reported in advance through natural time analysis.

Criticism: -The results have been questionable in the past due to a high number of false positives as well as many earthquakes being missed.

-Signals from non-seismic activities interference severely disturb the model's prediction

-Statistical validity: The empirical threshold $\kappa = 0.070$ is seen by some as arbitrary, lacking a solid theoretical foundation.

-The effectiveness may vary with geographical regions.

Positives: -The VAN method has shown a considerably higher success rate in predicting significant earthquakes. For instance, the 1995 predictions for earthquakes with magnitudes of M6, M6.6, and M6.5 were successful, with the M6.6 earthquake occurring in an area where no large earthquakes had been recorded for 1000 years

- Hence, many countries including Japan, China, Russia, etc. are using the VAN method to its best effect are looking to build on it. This highlights the fact that seismic electric signals may actually be a critical parameter for earthquake prediction.

Models on our Data - 1D CNN & LSTM

-Employed Keras tuner to optimise hyper-parameters, however did not see significant improvement in outcome.
-The Precision and Recall are extremely poor for class 1 (earthquake), while for class 0, it is much more reasonable. The obvious reason for this is that the data is skewed towards non-earthquake events (class 0).
-A very similar outcome can be seen for the LSTM model on the next page.

1D CNN

```
model = Sequential([
    Conv1D(96, kernel_size=3, activation='relu', input_shape=(500, 1)),
    MaxPooling1D(pool_size=2),
    Dropout(0.2),
    Conv1D(32, kernel_size=3, activation='relu'),
    MaxPooling1D(pool_size=2),
    Dropout(0.2),
    Flatten(),
    Dense(64, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])


model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

model.summary()
```

```
119/119 [==============================] - 2s 13ms/step
              precision    recall  f1-score   support

     class 0       0.80      0.99      0.88      3010
     class 1       0.37      0.02      0.04       784

    accuracy                           0.79      3794
   macro avg       0.58      0.51      0.46      3794
weighted avg       0.71      0.79      0.71      3794
```

Continuation

LSTM

```
[10] model_L = Sequential()
     model_L.add(LSTM(64, input_shape=(500, 1), return_sequences=True))
     model_L.add(Dropout(0.2))
     model_L.add(LSTM(64, return_sequences=False))
     model_L.add(Dropout(0.2))
     model_L.add(Dense(1, activation='sigmoid'))


     model_L.compile(optimizer='adam',
                   loss='binary_crossentropy',
                   metrics=['accuracy'])
```

```
119/119 [==============================] - 24s 182ms/step
              precision    recall  f1-score   support

     class 0       0.79      1.00      0.88      3010
     class 1       0.00      0.00      0.00       784

    accuracy                           0.79      3794
   macro avg       0.40      0.50      0.44      3794
weighted avg       0.63      0.79      0.70      3794
```

Next Steps

-Use Weights and Biases tool to improve the model
-Try out transform based model for better result
-Discuss with Prof. for better approaches