# Short Time Earthquake Prediction

Ardavan Sassani
Georgia State University, asassani1@gsu.edu

***Abstract* - Machine learning techniques and artificial intelligence algorithms have been utilized by several studies to improve the accuracy of forecasting earthquake events. Due to the complexity of the earthquake dynamics, it is difficult to model and simulate by using mathematical equations. The main problem is predicting the time, location, and magnitude simultaneously. In this study, we tried to simplify the problem by using a grid system to convert geospatial data into a vector of cells, map continuous magnitude into classes and limit the time of prediction to a time window of seven days. Hence, we changed it to a simple classification model that tries to predict the probability of an event with a specific class for the next seven days for each region cell. This study used a new technique to convert the geospatial data to a temporal feature tensor to feed the models. At the modeling stage, we tried four machine learning algorithms (SVM, Random Forest, LightGBM, KNN). The results showed that Random Forest has the best evaluation performance among the others, with F1_score values of 0.93 and 0.94 for positive and negative predictions, respectively.**

*Index Terms* –Earthquake regional forecasts, Machine learning, Lookback Period,

## INTRODUCTION

Earthquake predictions are crucial for hazard and risk assessment, well-informed risk management choices, and Early Warning Systems (EWS) emergency action. Regarding timing, earthquake forecasts may be divided into two broad categories: long-term forecasts (made months or years in advance) and short-term predictions (hours or days in advance). Due to the complexity of earthquake phenomena, earthquake prediction is an involute subject. The main problem of earthquake forecasting is its complexity of magnitude, location, and time of the event [1]. All these variables are continuous, and it is impossible to predict them all at once. This project aims to solve this problem by utilizing a set of techniques to simplify variables and convert them into a categorical classification problem.

Machine learning algorithms have been extensively used in several geological applications. Seven classes of ML methods have found the most use to date in four topic areas in earthquake engineering. The seven classes of ML methods are ANN, support vector machine (SVM), response surface model (RSM), logistic regression (LR), decision tree (DT) and random forest (RF), hybrid methods that couple two or more soft computing algorithms, and all other methods (e.g. evolutionary computing (EC) and genetic expression programming (GEP)) that are not significant in a number of applications [2]. We can convert this problem to binary classification and use conventional ML algorithms to predict the chance of having an event in a specific area [3] [4]. This method assumes that F. Corby suggests that the complex motion recorded by geodesists at subduction zones might be diagnostic of earthquake imminence [5]. Y. Kagan used only past earthquake data to estimate future earthquake rate density (probability per unit area, time, and magnitude) on a 0.1-degree grid for a region including California and Nevada. Their long-term forecast is not explicitly time-dependent but can be updated at any time to incorporate information from recent earthquakes [6].

## DATA AND METHOD

### Data collection and analysis

All data are collected from USGS [7] public data repository from 1902 to 2022 for a specific region in southern California (see Figure I). This dataset must be requested via a Rest API with some limitations for the number of records at each query. It needs to be done with a customized script and integrate the results in one local data warehouse. The same technique must be used to maintain the data and create queries for the model in the future. The format of this data is GeoJson and contains 28 attributes which are summarized in


FIGURE I
THE STUDY AREA IN CALIFORNIA.

The raw dataset contains 901,968 records, and we selected five attributes Latitude, Longitude, Depth, Magnitude, and Time. The remaining attributes have no relationship to our study, because some have been used to calculate the center of the event and the accuracy of detecting the magnitude and location of occurred earthquakes. The other attributes like source, url, and id are administrative objects and do not affect the prediction target. Figure II Shows an overview of the raw dataset. One observation is the shape of the histogram, which is the bell shape. In the normal condition, we expect the right skewed magnitude curve. One reason is the increased precision and number of seismograph stations over time. The older stations could detect the events with magnitude values greater than some thresholds, which improved over time. We have found the same pattern in several related studies [8] [4]. To find the best period, we plot the count of recorded events against the time and observe a trend inside the data over time. Figure III Shows the count of events has increasing rate after the 1970s, which explains the bell shape of the curve. Based on this observation, we filter out events older than 1970 to have more consistent data. Also, we can see the different values between the three classes in terms of events frequency over time. Events with higher magnitude have more counts when we go back in time, and the events with lower magnitude were not detected at the same time, which can be evidence of our reason for explaining the bell shape of the distribution.

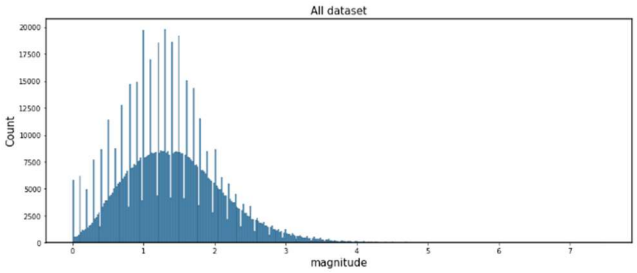



FIGURE II
RECALL VALUES VS. MEAN OF MAGNITUDE OF THE SAMPLE CELL FOR RANDOM FOREST AND LIGHTGBM ALGORITHMS

## Data cleaning and outliers

The preliminary report of data is summarized in Table I. The only attribute with missing or out-of-range value is depth. To solve this problem, we use a weighted average of depths of events in the same cell block. The final report of data is shown in Table III.

TABLE I
PRELIMINARY REPORT OF THE DATASET

| Attribute | Data type | Missing Rate |
|---|---|---|
| datetime | datetime64[ns] | |
| longitude | float64 | 0 |
| latitude | float64 | 0 |
| depth | float64 | -0.00016 |
| magnitude | float64 | 0 |

TABLE II
USGS GEOJSON DATA ATTRIBUTES

| | Attribute | Range | Description |
|---|---|---|---|
| 1 | alert | green, yellow, orange, red | The alert level from the PAGER earthquake impact scale. |
| 2 | cdi | [0.0, 10.0] | The maximum reported intensity for the event. |
| 3 | code | | An identifying code assigned by the corresponding source for the event. |
| 4 | detail | | Link to GeoJSON detail feed from a GeoJSON summary feed. |
| 5 | dmin | [0.4, 7.1] | Horizontal distance from the epicenter to the nearest station. |
| 6 | felt | [44, 843] | The total number of felt reports submitted to the DYFI? system. |
| 7 | gap | [0.0, 180.0] | The largest azimuthal gap between azimuthally adjacent stations (in degrees). |
| 8 | ids | | A comma-separated list of event ids that are associated to an event. |
| 9 | mag | [-1.0, 10.0] | The magnitude for the event. |
| 10 | magType | Md, Ml, Ms, Mw, Me, Mi, Mb, MLg | The method or algorithm used to calculate the preferred magnitude for the event. |
| 11 | mmi | [0.0, 10.0] | The maximum estimated instrumental intensity for the event. |
| 12 | net | ak, at, ci, hv, ld, mb, nc, nm, nn, pr, pt, se, us, uu, uw | The ID of a data contributor. Identifies the network considered to be the preferred source of information for this event. |
| 13 | nst | | The total number of seismic stations used to determine earthquake location. |
| 14 | place | | Textual description of named geographic region near to the event. |
| 15 | rms | [0.13,1.39] | The root-mean-square (RMS) travel time residual, in sec, using all weights. |
| 16 | sig | [0, 1000] | A number describing how significant the event is. Larger numbers indicate a more significant event. |
| 17 | sources | ,us,nc,ci, | A comma-separated list of network contributors. |
| 18 | status | automatic, reviewed, deleted | Indicates whether the event has been reviewed by a human. |
| 19 | time | | Time when the event occurred. Times are reported in milliseconds. |
| 20 | title | | The title of the feed. |
| 21 | tsunami | 0,1 | This flag is set to 1 for large events in oceanic regions and 0 otherwise. |
| 22 | type | earthquake, quarry | Type of seismic event. |
| 23 | types | | A comma-separated list of product types associated to this event. |
| 24 | tz | [-1200, +1200] | Time zone offset from UTC in minutes at the event epicenter. |
| 25 | updated | | Time when the event was most recently updated. |
| 26 | url | | Link to USGS Event Page for event. |
| 27 | longitude | [-120.30, -114.30] | Decimal degrees longitude. Negative values for western longitudes. |
| 28 | latitude | [32.0, 37.5] | Decimal degrees latitude. Negative values for southern latitudes. |

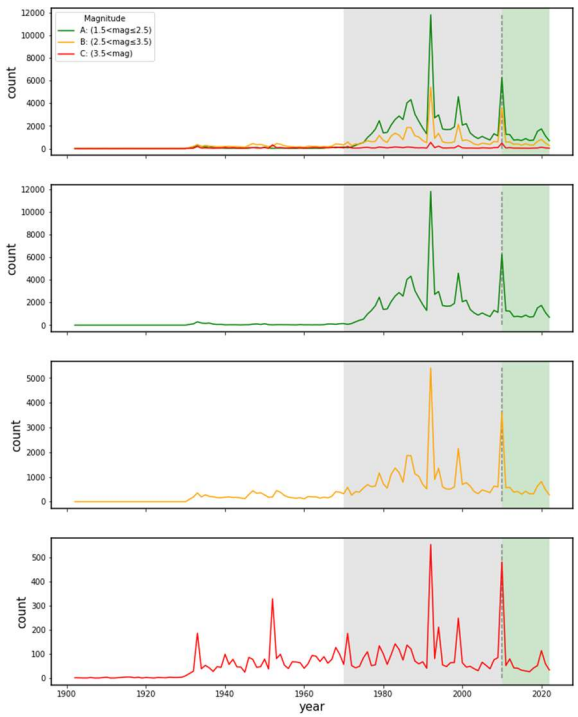



FIGURE III
DATA DISTRIBUTION AMONG DATE FOR EACH CATEGORY

TABLE III
DESCRIPTION OF DATASET

| | longitude | latitude | depth | magnitude |
|---|---|---|---|---|
| count | 362850 | 362850 | 362850 | 362850 |
| mean | -117.042 | 34.336611 | 5.86 | 2.08 |
| std | 1.208 | 1.330851 | 4.88 | 0.52 |
| min | -120.3 | 32 | 2.83 | 1.5 |
| 25% | -117.852 | 33.262 | 2.39 | 1.7 |
| 50% | -116.829 | 34.179 | 5.37 | 1.9 |
| 75% | -116.233 | 35.064 | 8.57 | 2.3 |
| max | -114.301 | 37.5 | 146.9 | 7.5 |

***Discretization***

To have more consistent data, we filtered out all events with a magnitude value less than 1.5 from the dataset. We classified them into three categories A, B, and C. Details of the binning and quantity of each group are shown in Table IV. Figure III Shows the distribution of filtered data and categorized datasets.

TABLE IV
BINNING AND QUANTITY OF EACH CLASS OF EVENT

| Range | Category | Portion |
|---|---|---|
| 2 – 2.5 | A | 0.58 |
| 2.5 – 3.5 | B | 0.32 |
| 3.5 < | C | 0.05 |

To have better control of data points, we created a grid to discretize the location of events to track and find the relationships between neighbor areas for each cell. The grid size is 100x100, and the actual cell area is approximately 10kmx10 km. It is larger and smaller on the south and northern boundaries, respectively (see Figure V).

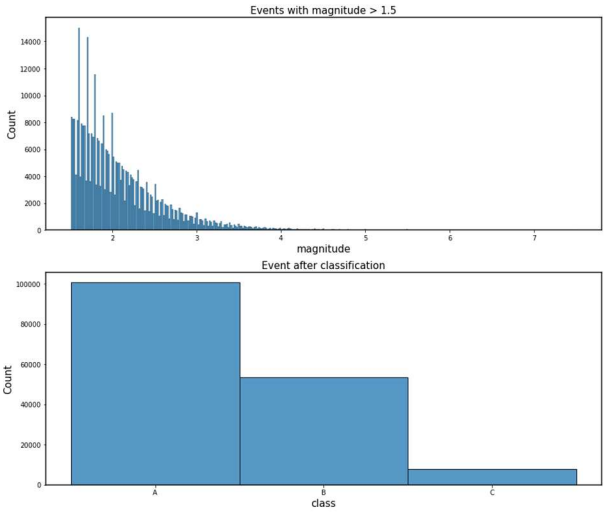



FIGURE IV
DISTRIBUTION OF DATA AFTER FILTERING AND CATEGORIZATION

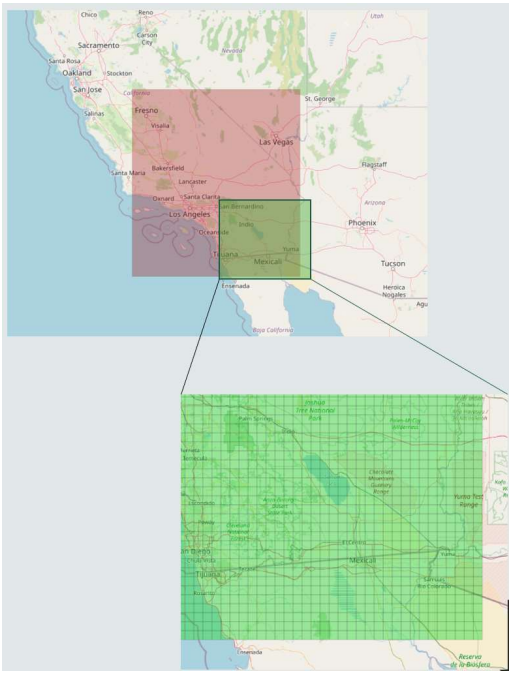


FIGURE V
A GRID SYSTEM TO AGGREGATE DATA IN INDIVIDUAL CELLS

We also grouped data by week and assigned the week of the year to each aggregated data point. This helped us narrow our search space (forecasting domain) to one week ahead. At this point, we have a dataset with a shape of 347,837x5, which contains mode magnitude class, averaged magnitude value, average depth, time, and cell indexes (10,000 unique cells from 100x100 grid). Then we transformed the dataset by adding cell ids as a new index and having a dataset of shape 2809x10000. Each row contains events for all cells on a specific date. We summarized historical data to have an overview (see Figure VI). To create this graph, we calculate

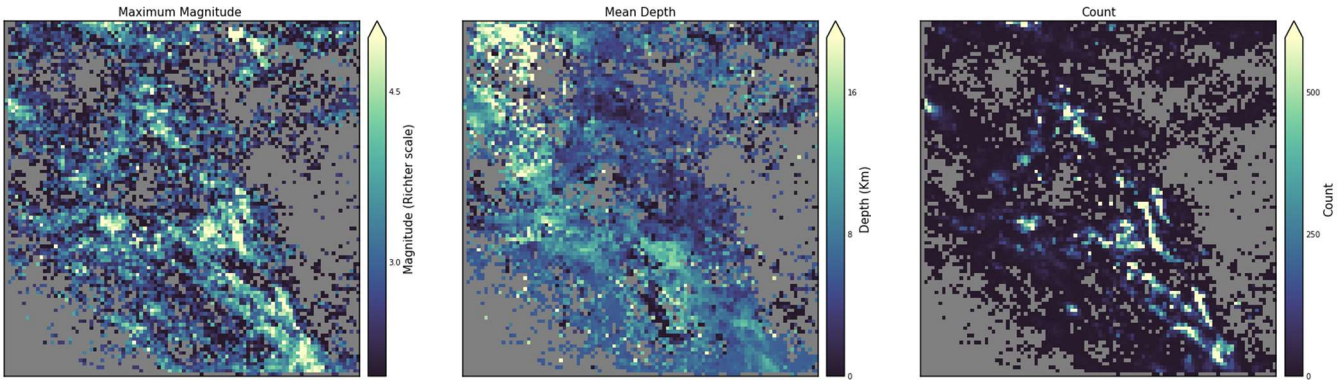



FIGURE VI
SUMMARY OF HISTORICAL DATA BASED ON MAXIMUM MAGNITUDE, MEAN DEPTH, AND THE COUNT OF EVENTS

The average depth, the maximum level of recorded magnitude, and the count of all events in each cell for the entire dates. Also, we will use the cells with the most frequent events to select and tune our final model.

**Feature extraction**

Our main assumption was that there are relations between the probability of a future event and historical data (a week before) for each cell and its neighbors. To find these relations, we implemented a system to get neighbors for each cell by a given radius. A radius equal to one, means that the eight neighbors of the cell, and a radius equal to three is 24 neighbors by three cell distance (see Figure VII). Then we used this neighbor list to select the aggregated values for magnitude and depth for each region. We used mean, max, and count functions for depth, magnitude value, and event occurrence. This method gave us an informative feature set that we were able to tune by changing the radius value. The final value that we ended up with was four. To label the dataset, we used a minimum threshold to assign a Boolean value based on the magnitude of the target cell. It means that we have no event for any target cell at that time if the magnitude value is less than the threshold and vice versa. Then we shifted back these target values to one unit (one week). At this point, we have a tensor with the shape of 10,000x2809x13, which are cell_id x date x (12)features + label.

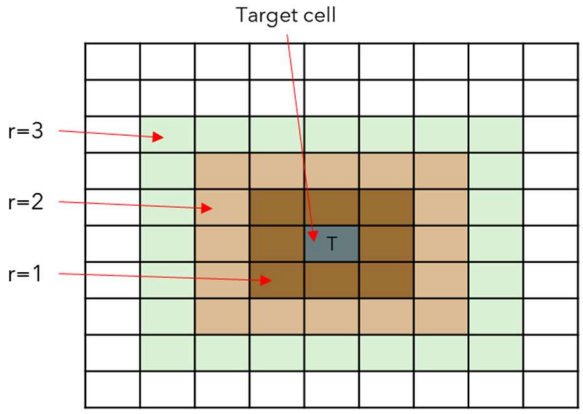
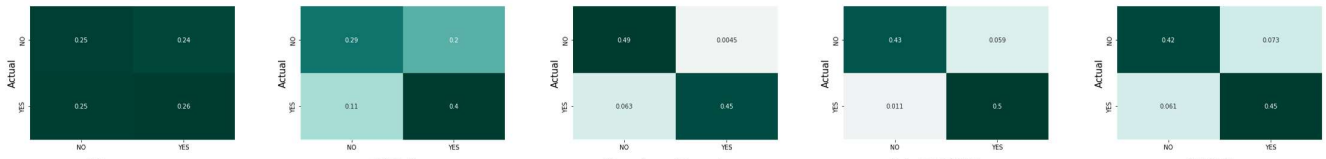



FIGURE VII
NEIGHBORS BASED ON THEIR DISTANCES (RADIUS) TO THE TARGET CELL

**Train and Test splitting**

We divided the dataset into training and testing with 70% and 30%, respectively. Then we used 30% of the training section to tune the models. This dataset was highly imbalanced, so we tried SMOTE method to make train and test datasets with a balanced distribution. Figure IX illustrates the results before and after implementing the stratification technique.




Dummy SVM Random Forest LightGBM KNN

FIGURE VIII
CONFUSION MATRICES FOR PREDICTIONS OF ALL MODELS

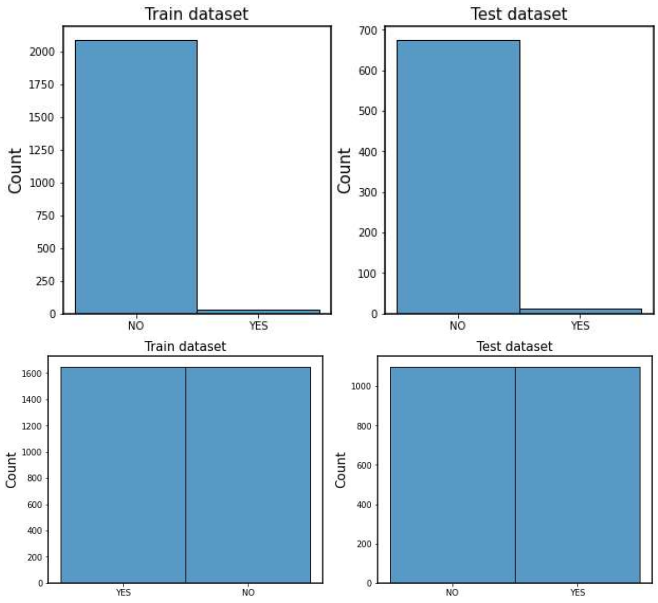



FIGURE IX
DISTRIBUTION OF LABELED TARGETS BEFORE AND AFTER SMOTE

### Machine learning models

We tuned, trained, and tested five machine learning models using ten topmost frequent cells with the highest value of events count. The evaluation metrics that we used are f1_score, Jaccard, precision, and recall for both true and false labels. Results are summarized in Table V. Also, the confusion matrixes for all models are shown in Figure VIII.

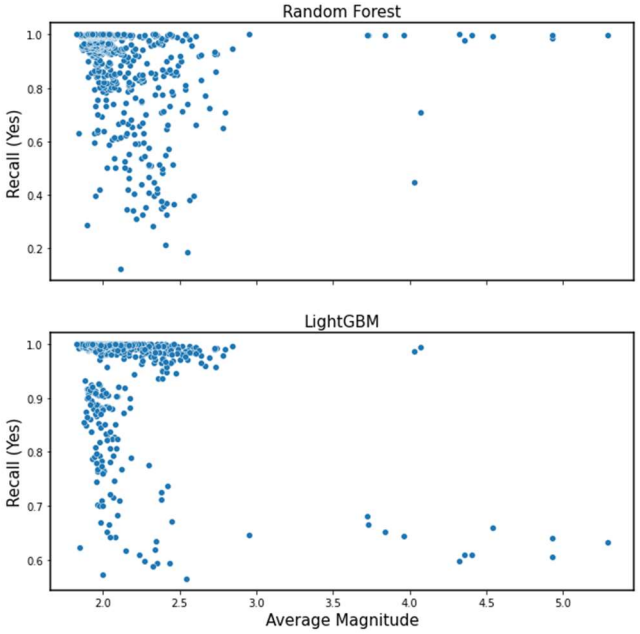



FIGURE X
RECALL VALUES VS. MEAN OF MAGNITUDE OF THE SAMPLE CELL FOR RANDOM FOREST AND LIGHTGBM ALGORITHMS

TABLE V
THE RESULT AFTER TRAINING CANDIDATE MODELS WITH THE TEN TOPMOST FREQUENT CELLS DATA

| Evaluation metric | | Dummy | SVM | R.F. | LGBM | KNN |
|---|---|---|---|---|---|---|
| F1_score | Yes | 0.5 | 0.72 | **0.93** | 0.93 | 0.87 |
| | No | 0.5 | 0.65 | **0.94** | 0.93 | 0.87 |
| Jaccard | Yes | 0.34 | 0.57 | 0.87 | **0.88** | 0.77 |
| | No | 0.34 | 0.49 | **0.88** | 0.87 | 0.76 |
| Precision | Yes | 0.5 | 0.67 | **0.99** | 0.89 | 0.86 |
| | No | 0.5 | 0.73 | 0.89 | **0.97** | 0.87 |
| Recall | Yes | 0.5 | 0.79 | 0.88 | **0.98** | 0.88 |
| | No | 0.5 | 0.59 | **0.99** | 0.89 | 0.85 |

According to preliminary results, we had two options with the highest evaluation performance; Random Forest and LightGBM. After another step of the investigation, we have found that The Random Forest model has missed events with low values of magnitude (see Figure X), which is not essential if we ignore them in real life because, in trade between precision and recall, we put more attention on events with a higher magnitude which Random Forest output covers the majority of them. Thus, we used the Random Forest algorithm for the rest of the cells.

## RESULTS

We run the tuned Random Forest algorithm for all 10,000 cells in dataset. For each cell, we create a list of neighbors with radius size of one, two, three and four, which provide us a set of {8, 16, 24, 32} neighbors lists, respectively. We use these lists to aggregate data as we explained before. We added a condition to filter feature list based on their size. We only considered the cells with a magnitude value of 2.5 or above. After running the model for all remaining cells, we get 660 cells with enough data to forecast the probability of next week's earthquake. Then we get the mean of evaluation metrics for each cell with at least one prediction. In Figure XI each pixel represents the mean value of each evaluation metric in two labels positive and negative separately. Overall performance of the model is summarized in Table VI.

TABLE VI
FINAL EVALUATION VALUES

| Evaluation Metrics | Value |
|---|---|
| Yes_f1 | 0.909 |
| Yes_jacard | 0.851 |
| Yes_recall | 0.895 |
| Yes_precision | 0.954 |
| No_f1 | 0.917 |
| No_jacard | 0.862 |
| No_recall | 0.940 |
| No_precision | 0.921 |

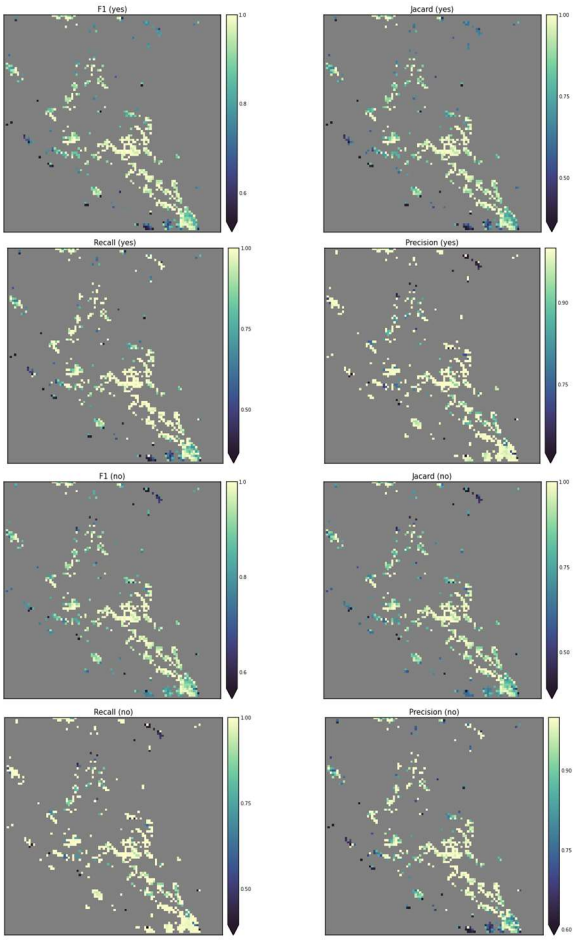



FIGURE XI
AVERAGE VALUES OF EVALUATION METRICS FOR EACH CELL WITH POSITIVE AND NEGATIVE LABELS

## CONCLUSION

Forecasting an earthquake in advance has uncountable benefits, from saving individual lives and properties to reducing recovery time. Based on the nature of earthquake dynamics, having a robust and reliable model that receives new data and predicts the probability of a future event's exact time, location, and magnitude simultaneously is not easy. Using some simplification techniques can help us to overcome this problem. In this study, we utilized a simple system to extract and create new features from adjacent location attributes and feed the models with them. With this method, we could predict earthquake events seven days ahead for each block in southern California's specific region. This model can hit positive events with a recall value of 0.88 by 0.99 accuracy of positive precision, which can be used for EWS feeding data. This model can be improved in many aspects. We only considered the time window of seven days, which can be changed to any time period. Still, the issue is that we lose available data to train the model by decreasing the window size. By increasing this value, we lose the valence of forecasting. For example, it is worthless to forecast the probability of an earthquake event by a window size of six months. The other parameter is how we find the related neighbors. Currently, it is a simple squared shape of adjacent cells, but it can be converted to other types of distances by considering the soil and other tectonic attributes. It can be automatically updated weekly in response to the evolving nature of the region's tectonics.

## REFERENCES

[1] Keilis-Borok, Vladimir, "EARTHQUAKE PREDICTION:State-of-the-Art and Emerging Possibilities," *Earth Planet. Sci.,* vol. 30, pp. 1-33, 2002.

[2] Yazhou Xie, M.EERI, Majid Ebad Sichani, "The promise of implementing machine learning in earthquake engineering: A state-of-the-art review," *Earthquake Spectra,* vol. 36, pp. 1769-1801, 2020.

[3] Karimzadeh, S., Matsuoka, M., Kuang, J., "Karimzadeh, S., Matsuoka, M., Kuang, J., & Ge, L. (2019). Spatial prediction of aftershocks triggered by a major earthquake: A binary machine learning perspective," *ISPRS International Journal of Geo-Information,* vol. 8, p. 462, 2019.

[4] Asim, K. M., Martínez-Álvarez, "Earthquake magnitude prediction in Hindukush region using machine learning techniques," *Natural Hazards,* vol. 85, pp. 471-486, 2017.

[5] Corbi, F., Sandri, L., Bedford, "Machine learning can predict the timing and size of analog earthquakes," *Geophysical Research Letters,* vol. 46, pp. 1303-1311, 2019.

[6] Kagan, Y.Y., Jackson, D.D., "Short- and Long-Term Earthquake Forecasts for California and Nevada," *Pure Appl. Geophys,* vol. 167, pp. 685-692, 2010.

[7] "USGS," [Online]. Available: https://earthquake.usgs.gov/.

[8] Anushka Joshi, Chalavadi Vishnu, C Krishna Mohan, "Early detection of earthquake magnitude based on stacked ensemble model," *Journal of Asian Earth Sciences,* vol. 8, 2022.

## AUTHOR INFORMATION

**Ardavan Sassani,** Graduate research assistant in Disaster Informatic Computational Epidemiology (DICE), Department of Computer Science, Georgia State University.