USGS Earthquake Hazards Program, Final Report: Deep Learning Based Approach to Integrate MyShake's Trigger Data with ShakeAlert for Faster and Robust EEW Alerts (Award No. G20AP00058, May 2020 through April 2021)

Author: Qingkai Kong

Affiliation:

Previously at Berkeley Seismology Lab, UC Berkeley, till December 2020

Present: Lawrence Livermore National Laboratory (started January 2021)

Address: 7000 East Ave, Livermore, CA 94550

Email: kongqk@berkeley.edu or kong11@llnl.gov

Final Technical Reports shall include these components:

Acknowledgement:

Research supported by the U.S. Geological Survey (USGS), Department of the Interior, under USGS award number G20AP00058. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the opinions or policies of the U.S. Geological Survey. Mention of trade names or commercial products does not constitute their endorsement by the U.S. Geological Survey. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract Number DE-AC52-07NA27344. This is LLNL Contribution Number LLNL-TR-830065.

Abstract

Earthquake Early Warning (EEW) systems depend on the dense seismic network to make fast and accurate earthquake detections to issue useful early alerts. The recent development and implementation of the ShakeAlert system is relying on the west coast seismic network to detect and estimate the earthquake parameters in real-time. While working on making improvements on the current system, one potential approach is to include more seismic recordings from various sources, such as the low-cost smartphone seismic network, such as the MyShake network, or the Google Android smartphones, since these smartphone seismic networks have much more portable stations (phones) to potentially provide useful data to the system. This report summarizes the initial exploration of using a deep learning approach to combine the data from both traditional seismic stations and the smartphone data in California. Due to the heterogeneity nature of the data, we aggregate data from the phones (using simulation data), as well as that from traditional seismic stations to grid cells. By generating real-time triggering ratio grid cell maps, the designed deep learning algorithm can process the data from multiple sources and detect the earthquake faster than only using that from a traditional seismic network.

1. Introduction

Earthquake early warning (EEW) systems are a race with earthquakes (Gasparini, Manfredi, and Zschau 2007), the faster we can detect the earthquake, the more warning time we can provide for the people who are likely to be affected (Allen and Melgar 2019). With the recent launch of the ShakeAlert public roll out (Kohler et al. 2020), more people can be protected from nearby earthquakes. One way to improve the detection speed and accuracy of the EES system is to increase the number of seismic stations and cover the region better. A low cost fashion can be a potential solution, which uses the sensors within commercial devices, such as smartphones, as portable seismic stations (Allen, Kong, and Martin-Short 2019; Bossu et al. 2018; Lee et al. 2019; Finazzi 2016; Minson et al. 2015). MyShake is a project developed at Berkeley Seismology Lab that uses the sensors inside the daily used smartphones to detect earthquakes in a crowdsourcing way, which can potentially increase the number of sensors in a place within a very short time frame (Qingkai Kong et al. 2016). The idea behind is to turn the smartphones into portable seismic sensors once the MyShake application is downloaded. The app monitors the phone's accelerometer when it is in steady state (not moving for 5 min), and a standard STA/LTA algorithm decides whether the phone is moved by some motion. Once the phone movement is detected, an artificial neural network algorithm running on the phone will check if the motion is due to earthquake or human activities. If the motion is due to an earthquake, then a message with the timestamp, latitude, longitude and acceleration amplitude with some of the phone metadata will be immediately sent to the server for further analysis. These triggers arriving at the server can be further analyzed. Due to the earthquake nature, triggers from the smartphones can show clustering patterns both in time and space, which can be used for detection. Furthermore, time series data of 5-min also uploaded when the phone connected to WiFi and power, this provides more data for research analysis. The initial data from the MyShake observations show promising results, earthquake early warning, routine network operations, array processing, building health monitoring and so on can potentially achievable (Qingkai Kong, Martin-Short, and Allen 2020; Qingkai Kong, Allen, and Schreier 2016; Q. Kong, Patel, and Inbal 2019; Qingkai Kong, Martin-Short, and Allen 2019; Inbal et al. 2019; Qingkai Kong et al. 2018). In 2020, built on top of MyShake experiences, Google launched the Android Earthquake Alerting System (https://blog.google/products/android/earthquake-detection-and-alerts/), which is aiming to set up a global earthquake early warning system using all the Android phones they have. This will be a dense global seismic network for the purpose of earthquake early warning, and potential can do much more.

The opportunity to scale up the seismic network not only on the hardware side to use thousands of phones, but also in recent development of machine learning algorithms. Machine learning algorithms have the capability to extract useful features for specific tasks from large amounts of data, especially the recent development of the deep learning algorithm, which can automatically extract features (Goodfellow, Bengio, and Courville 2016; LeCun, Bengio, and Hinton 2015; Voulodimos et al. 2018). There are existing efforts using machine learning algorithms in EEW (Li et al. 2018; Meier et al. 2019; Fauvel et al. 2020; Apriani, Wijaya, and Daryono 2021; Saad, Hafez, and Soliman 2021) which show promising results.

In this report, we summarize the initial exploration of combining the data from seismic stations as well as the smartphone seismic network using a machine learning approach. There are mainly two parts for this report: first, in order to explore how we can integrate the data from the two networks, we conduct analysis to understand what benefits we can have to combine the two networks using simulations, especially on whether we can have faster alerts using the two networks. We tried both 0.1% population density as phone contributors (more realistic for MyShake users) as well as the 10% population, which is more along the lines of Google's Android phones can be integrated. Results from this analysis show the expectation of how much overall faster the combined system can achieve and where the events occur affect the system detection time. Second, we show the initial results from the designed deep learning based approach that can handle different data sources motivated by how human beings are viewing the patterns. By using the simulation data as the training data, the designed model can recognize the earthquake occurring patterns.

2. Data

Data used in this study are mainly from two sources (1) EPIC data from ShakeAlert (2) Simulated smartphone data using the designed simulation platform.

2.1 EPIC data

With help from Ivan Henson from Berkeley Seismology Lab, we got the seismic station trigger data from the algorithm EPIC in the ShakeAlert system. Data are downloaded from 2016-02 to 2020-08 for the associated events that are larger than M3.5 in the USGS earthquake catalog. In total, there were 375 events in CA during this period. This dataset includes the following data fields for each of the earthquakes:

- Associated earthquake event id from USGS
- Seismic station trigger time
- Seismic station trigger location (latitude and longitude)

2.2 Smartphone simulated data

The main data used in this study are generated from the simulation platform, which can generate phone triggers including latitude, longitude, trigger time, trigger PGA for earthquakes and random events. The following lists the data used for the two applications.

2.2.1 Data used for understanding the improvement of detection speed if phone data are added

Data used in this analysis contains both simulated data for the 0.1% and 10% population contributing the phone data. We use the same set of 375 events in 2016-02 to 2020-08 to generate the simulated events within 300 km from the epicenter. Simulated data contains:

- Trigger time
- Trigger location (latitude and longitude)
- Trigger amplitude
- Due to earthquake (P or S) or random trigger

2.2.2 Data used for training the deep learning model

To train and test the deep learning model, we generate the following simulation data

- Center at (38, -122), with 0.1 steps in both directions for latitude and longitude to generate grid cells. In total, there are 30 bins for latitude and longitude (about 330 by 330 km) to generate cells.
- Magnitude range: 3.5 to 7.6 with 0.5 magnitude step

- Earthquake time during the day: 00:00:00, 06:00:00, 12:00:00, 18:00:00 (at different time, there are different percentage of the phones available, with daytime, less phones available for detection)

In total, we generated 8100 earthquake scenarios and 1,425,817 non-earthquake (images not containing earthquake triggers) and 492,570 earthquake images, with 1,212,069 training samples, 327,546 validation samples, and 378,772 test data.

3. Method

3.1 Simulation platform

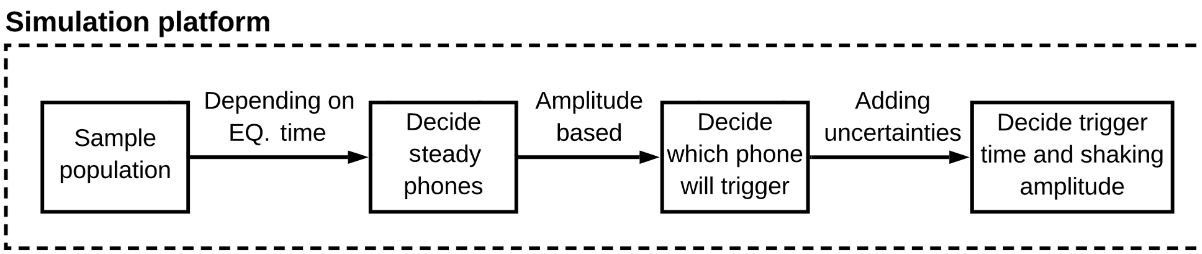

Figure 1. The overview of the structure of the simulation platform, adopted from (Qingkai Kong, Martin-Short, and Allen 2020).

The simulation platform is built on top of the MyShake observations that can generate synthetic trigger times and locations in a region based on the population. The steps of generating smartphone triggers are showing in Figure 1, starting with defining a percentage of population can contributing the data in one region, the platform will decide the phone distribution based on the earthquake occurring time and location (because different percentage of phones are ready to monitor the earthquakes depending on the time of the earthquakes) and a triggering probability based on a attenuation relationship. Based on these triggering probabilities and attenuation relationship, the platform will provide the time and distribution of the phones that would trigger during this simulated earthquake, for more details, please refer to (Qingkai Kong, Martin-Short, and Allen 2020).

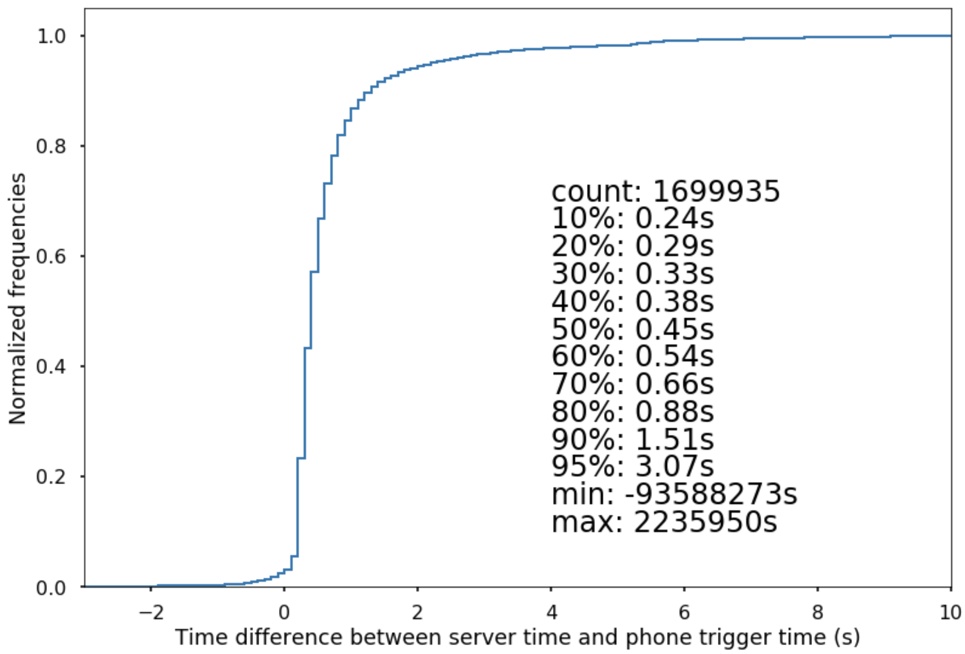

Figure 2. The distribution of the time difference in seconds between server time and phone trigger time from MyShake network data (2019-10-17 to 2020-03-01) that is used for approximating the network latencies in the simulation platform.

In this study, we also add one more feature into the simulation platform - the network latencies. Figure 2 shows the distribution of the time difference in seconds between the trigger arrival time on server and phone local trigger time. This distribution is used to provide the basis to draw the time latency potentially within the smartphone network in the simulation platform, so that it is more accurate in the estimation of the detection time using the smartphone data.

Data from this simulation platform are both used in the understanding of the improvement of detention speed if phone data is added and also the design of a deep learning approach to combine both data sources shown in the preliminary results section. An example of both the MyShake observed triggers as well as the simulated triggers with the EPIC triggers are shown in Figure 3. As for the MyShake observations, as this is a small earthquake M3.81, we can see that the triggers actually happened during the passage of both P and S waves while in the simulated case, most of the triggers are along the S wave passage. This is due to the conservative relationship we used in the simulation platform. But overall, the simulation platform captures the main characteristics of the patterns.

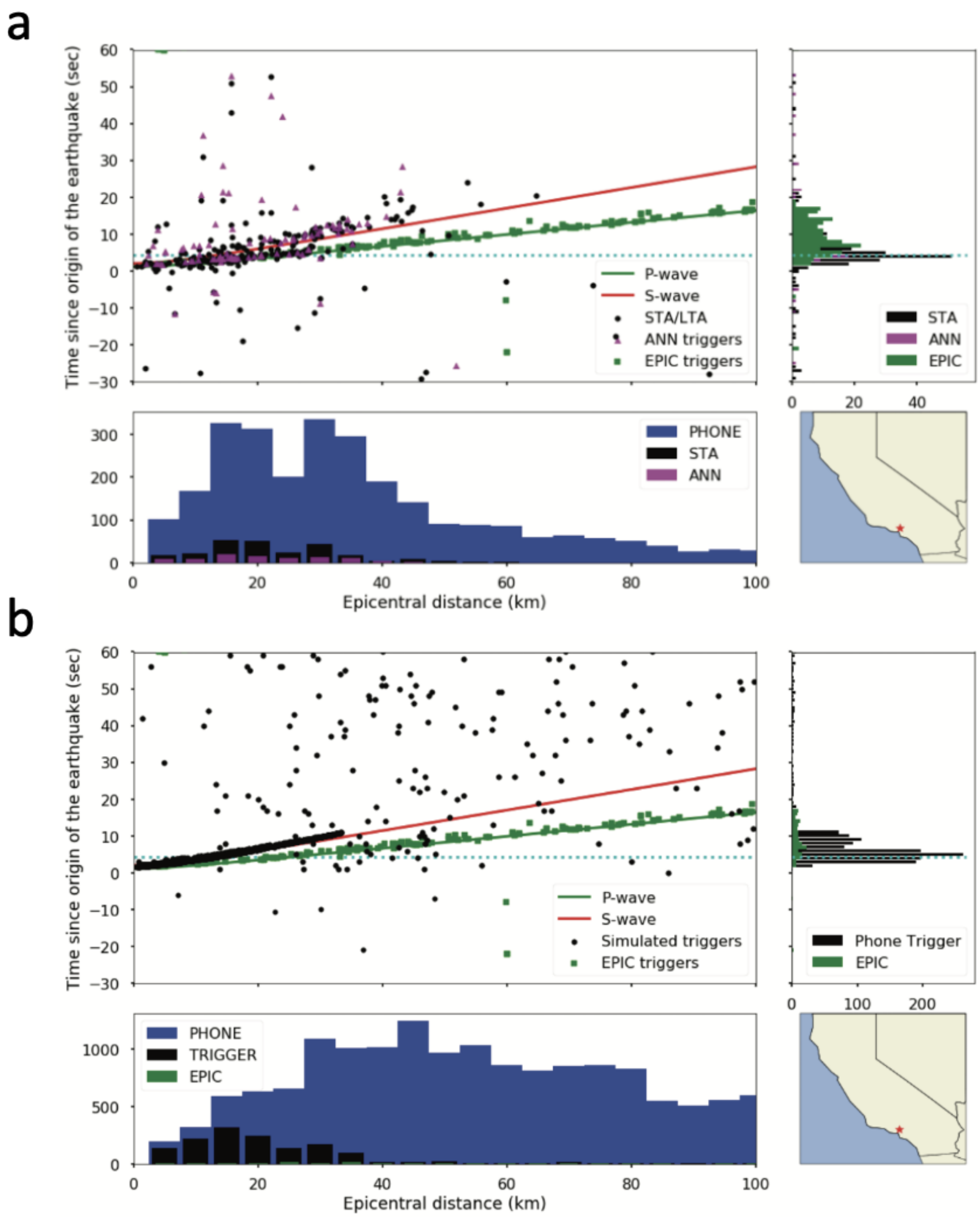

Figure 3. Trigger examples from real MyShake observations as well as the simulated triggers for M3.81 earthquakes near San Fernando, CA on 2020-07-30T13:48:19. (a) MyShake real observations. (b) Simulated smartphone triggers. Each figure has 4 panels, top left: Trigger distances v.s. time, green and red lines are theoretical P and S arrivals. Top right: trigger histogram over 1s time bins. Bottom left: trigger histogram over 5 km distance bins. Bottom right: map shows the event location.

3.2 Deep Learning approach for combining seismic station data and the phone data

Since the data from the traditional seismic stations and the smartphone sensor network have different qualities, and the data from the seismic stations are of much higher quality, we need a way to combine them. The difference between the patterns of the triggers from earthquakes and non-earthquakes sources is that those from the earthquakes are usually showing coherence moveout from the epicenter due to the travel of the seismic waves, while those from the non-earthquake sources are mostly random in nature. If we look at the distribution of the triggers from the phones or the seismic in time sequence, we can see the coherent triggers moving out for the earthquake events. This inspired the idea of the designed method to try to capture these trigger patterns using the deep learning approach, since they are really good for working with images and movies, which are just sequences of images (Goodfellow, Bengio, and Courville 2016; LeCun, Bengio, and Hinton 2015; Voulodimos et al. 2018; Schmidhuber 2015).

In order to take advantage of deep learning's capabilities to process images, we turn the problem into image-based detection. In order to generate the images that capture the trigger patterns, we first divide the spatial regions into 0.1 by 0.1 degree cells, and use the triggering ratio (number of triggers divide the number of phones) within each cell as the value to indicate the detection of the shaking. Depending on the confidence level of the traditional stations, we can set the station trigger in the cell with a high ratio, such as 0.6 or 0.8. Figure 4 shows the trigger patterns from the grid cells before and 5s, 10s, and 20s after the earthquake, we can clearly see the patterns of the cells light up. The ratios in each cell are calculated by using a 20s length window, with 0.1s step. Note, due to the imbalanced nature of the data (there are more random images than earthquake images), we need to use a class weights to panelize more if the model makes mistake on the earthquake samples.

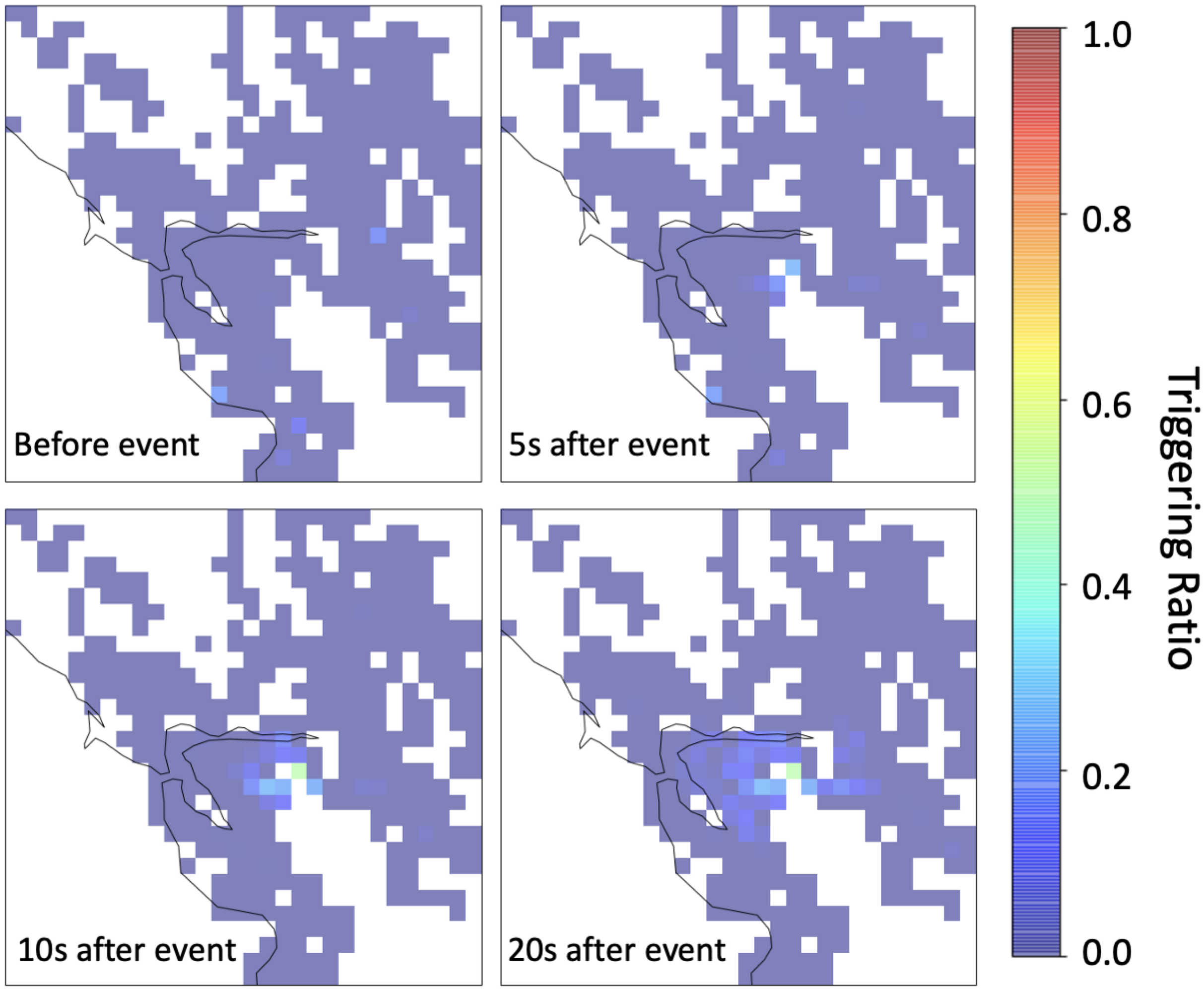

Figure 4. The trigger cell images at different times, i.e. before the event, 5, 10, and 20s after the event. Each cell is 0.1 by 0.1 degree, and the color of the cell shows the triggering ratio. Places without cells are no phones. These images are centered at (38, -122) with 30 bins for latitude and longitude respectively (about 330 by 330 km).

3.2.1 Baseline Convolutional Neural Network

To give a reference model performance as a baseline, we use the most common structure, a simple convolutional neural network (CNN) with 3 hidden layers. Table 1 shows the structure of the model and the total number of parameters are 121,409. The input images each have dimensions of 30 by 30, and can be multiple channels. We test using 1, 3 and 5 consecutive images as the input (corresponding to 1, 3 and 5 channels), and the best result is using 3 images. The standard CNN model extracts the features from the input images, and passes it to deeper layers. With the deeper layers, finer features will be extracted with more kernels. A binary_crossentropy loss function and adam optimizer are used to train the model. Please see the results in section 4.

Layer	Kernel size	Output shape	# of Parameters
Conv2D	(3, 3)	(None, 28, 28, 32)	320
MaxPooling2D	(2, 2)	(None, 14, 14, 32)	0
Conv2D	(3, 3)	(None, 12, 12, 64)	18496
MaxPooling2D	(2, 2)	(None, 6, 6, 64)	0
Conv2D	(3, 3)	(None, 4, 4, 64)	36928
Flatten	/	(None, 1024)	0
Dense	/	(None, 64)	65600
Dense	/	(None, 1)	65

Table 1. Structure of the Convolutional Neural Network. The input dimension is 30 by 30 pixels.

3.2.2 Convolutional-LSTM model

Since the above baseline model is a standard CNN model which is good at dealing with individual image and extract features, the problem we are facing also has the temporal information that provides important features. Therefore, a more sophisticated model which has the capability to capture this temporal evolution can be tested. Because Recurrent Neural Network (RNN) is good at addressing the temporal contexture information in the data, we choose a Convolutional-LSTM (ConvLSTM) as an improved model. ConvLSTM is a type of RNN for Statio-Temporal estimation that has convolutional structures in both the input-to-state and state-to-state transitions (Shi et al. 2015).

Table 2 shows the structure of this ConvLSTM model, we only used one ConvLSTM layer at this point, and the structure still has room to be improved on such as by experimenting more layers. We use categorical_crossentropy as the loss function, and use the adam optimizer to train the model.

Layer	Kernel size or dropout ratio	Output shape	# of Parameters
ConvLSTM2D	(2, 2)	(None, 29, 29, 32)	17024

Dropout	0.5	(None, 29, 29, 32)	0
Flatten	/	(None, 26912)	0
Dense	/	(None, 50)	1345650
Dense	/	(None, 2)	102

Table 2. The structure of the model with the ConvLSTM layer.

4. Preliminary Results

4.1 Understanding the improvement of detection speed if phone data are added

We first conduct analysis on combining both the smartphone triggers with the current ShakeAlert triggers, to understand the improvement of the system detection time. Using the designed simulation platform, we can generate the simulated phone triggers for the earthquakes that are first detected by the EPIC algorithm. In total, 375 events are first detected by the EPIC algorithm in CA during 2016-02 to 2020-08.

4.1.1 Assuming 0.1% of the population contributing phone data

We start the analysis by randomly sampling 0.1% of the California population as the potential smartphone sensors that are complementary to the ShakeAlert seismic stations.

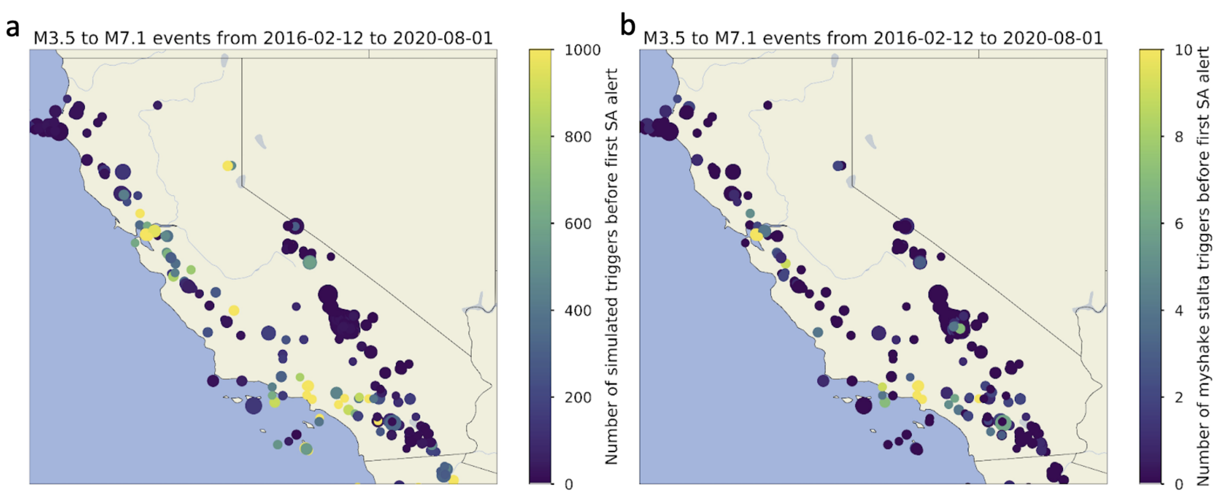

Figure 5. Number of smartphone triggers before the first SA alert for different events, sizes of the circles represent the magnitude of the earthquakes. (a) 0.1% of the population are assumed as smartphone sensors. (b) the real number of smartphone triggers from MyShake network. Note: for real MyShake triggers, most of the downloads in CA are after Oct 2019, this is one of the reasons there are few/no real MyShake triggers before the ShakeAalert first alert.

The number of smartphone triggers before the first ShakeAlert alert for simulated and real data is shown in Figure 5. From the left figure, we can see where we expect many triggers from the phones can potentially help improve the speed of the detection. Since the EPIC algorithm is relying on the first 4 triggers to declare a new event, we also analyze the time of when the first 4 triggers arrive at the server using traditional seismic stations alone or combining with the phones, and show the results below.

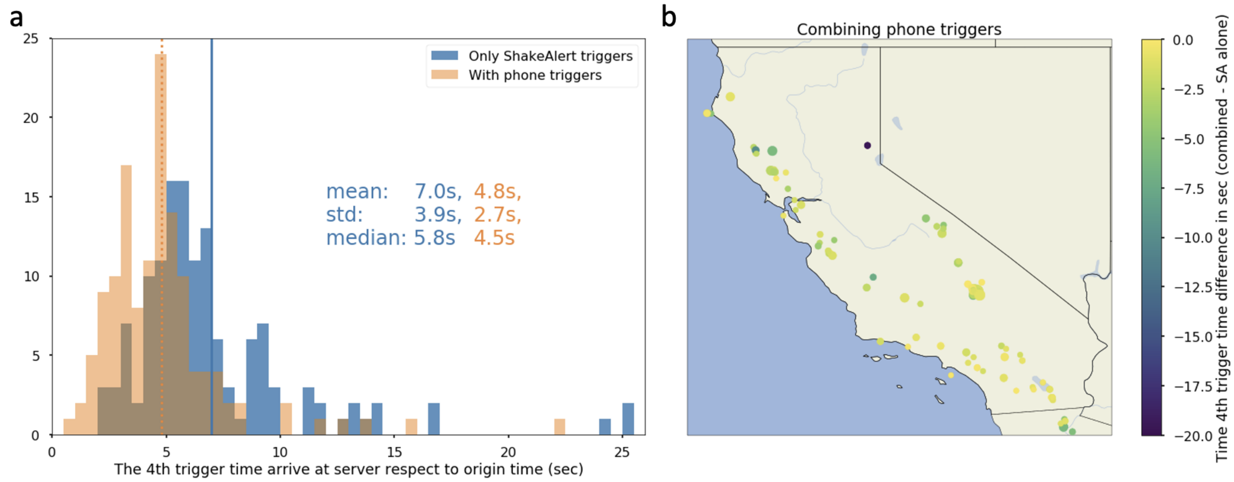

Figure 6. Results with 0.1% of the population that contributes data to the smartphone seismic network. (a) The histogram shows the time of the first 4 triggers arriving at the server with/without phone triggers. Blue bars are triggers from only ShakeAlert, while orange bars are triggers from both networks. The vertical lines are the corresponding mean values. (b) The spatial distribution of these events with color-coded how many seconds can be faster after combining the triggers from the phones. Negative numbers means faster.

Figure 6 shows the results with 0.1% of the population that contributes their data for the smartphone seismic network. Figure 6a shows that, with this amount of phone contributors, 134 (35.7%) events can have potential faster alerts by combining the phone sensors, about 2.2s faster (7 - 4.8s) for the mean value. Figure 6b shows where we can see big improvements (darker color dots). From it, we can see that places with more population but sparse seismic stations see the biggest improvements.

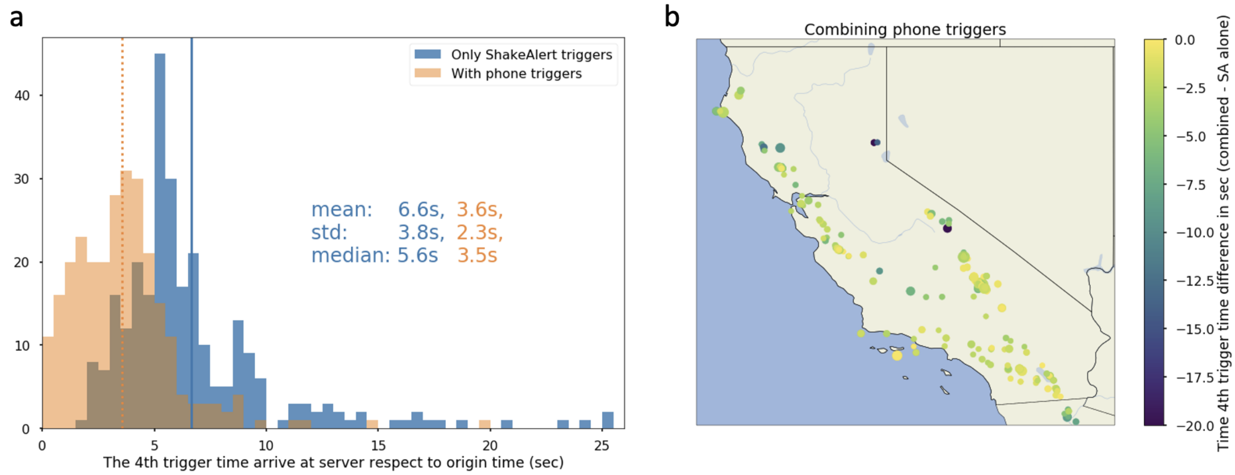

Figure 7. Results with 10% of the population that contributes data to the smartphone seismic network. (a) The histogram shows the time of the first 4 triggers arriving at the server with/without phone triggers. Blue bars are triggers from only ShakeAlert, while orange bars are triggers from both networks. The vertical lines are the corresponding mean values. (b) The spatial distribution of these events with color-coded how many seconds can be faster after combining the triggers from the phones. Negative numbers means faster.

4.1.1 Assuming 10% of the population contributing phone data

As of the launch of the recent Google Earthquake Alerting system (August 2020), there are many more phones that can potentially contribute data. As shown in the data section, we estimate about 10% of the population that can contribute their data to the system. Figure 7 shows the results with 10% of the population that contributes their data for the smartphone seismic network. Figure 7a shows that, with 10% phone contributors, 270 (72.0%) events can have potential faster alerts by combining the phone sensors, about 3.0s faster (6.6 - 3.6s) for the mean value. The Figure 7b shows where we can see big improvements (darker color dots). From it, we can see that places with more population but sparse seismic stations see the biggest improvements.

Figure 8 shows the locations of the events that have no improvements (which means all the first 4 triggers are from traditional seismic stations). We can see most of these events occurred at places where we usually have sparse populations or off the coast. For these events, it is better still relying on the traditional seismic network for the detection.

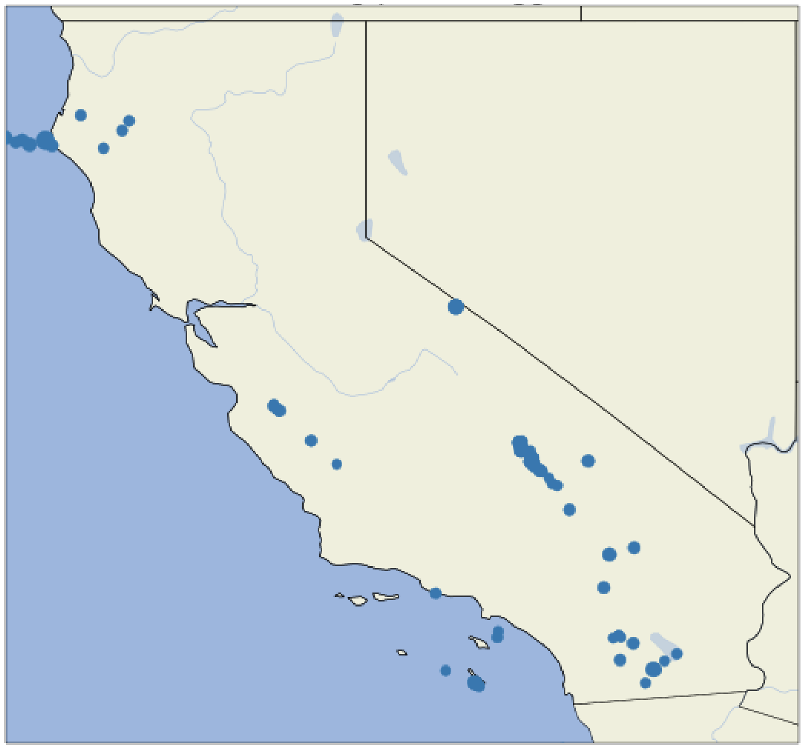

Figure 8. Locations of the events (blue dots) that have no improvements when combining the phone triggers.

4.2 The detection with deep learning model

In this section, we will show the initial results from the baseline model as well as the ConvLSTM based model. The dataset contains 67 real earthquake events in this region. The current MyShake DBSCAN method detects 36 events, while the baseline CNN detects 28 events and the ConvLSTM model detects 32 events. Note, the following results are the ones only from the phone data without adding in the ShakeAlert triggers. Note that, all the phone triggers used in this section are generated by the simulation platform with 0.1% of the population assumed as contributors.

4.2.1 Results from the baseline CNN model

Results from the baseline CNN model are summarized in Figure 9. Since the loss almost flattened, we stopped training at 30 epoch, and chose the lowest loss model as our model to test (Figure 9a). Figure 9b shows the detected events compared with the current MyShake DBSCAN method, 28 events are detected by the baseline CNN model while the DBSCAN detects 36. We can see there are many events not detected due to the low sensor coverage at these places. Panels c and d in Figure 9 show the detection time difference when the baseline CNN model compares with the current MyShake DBSCAN algorithm running on the same data as well as the alert time from the ShakeAlert system using the real data.

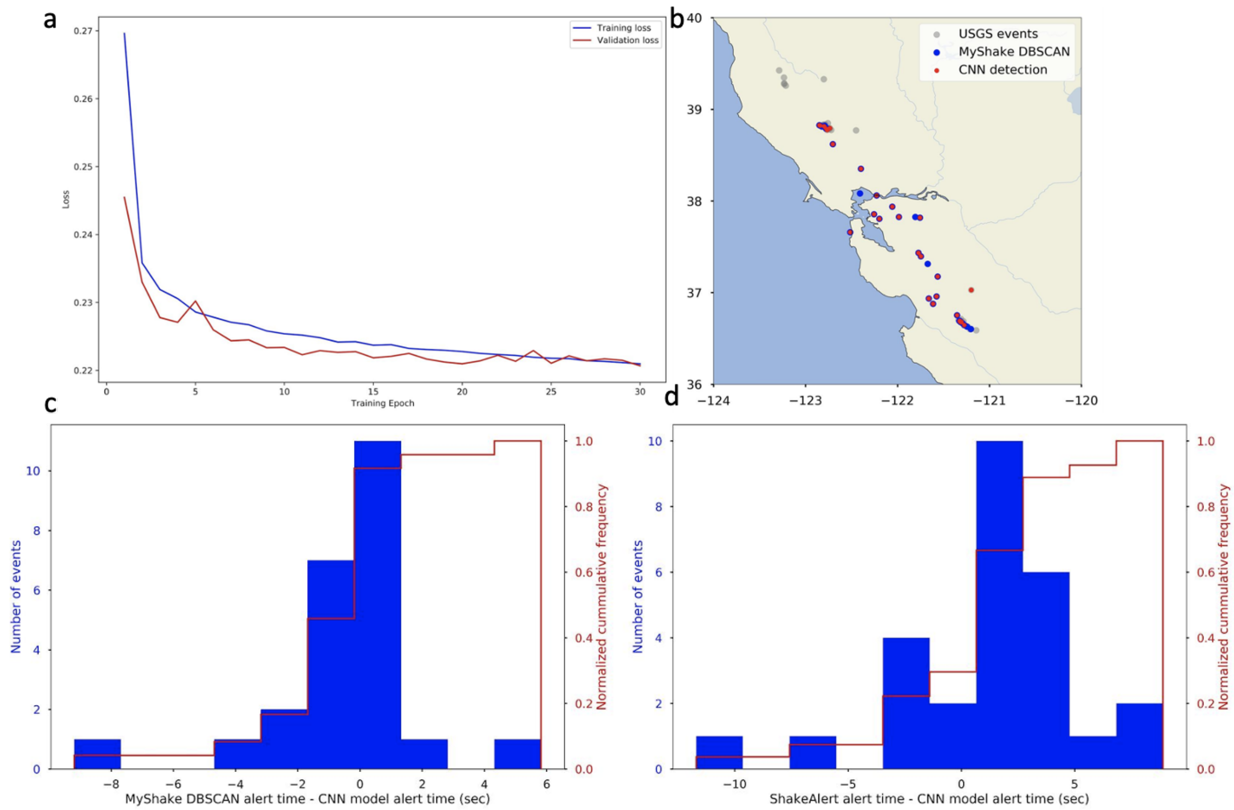

Figure 10. Summary of the baseline CNN performance. (a) training curve showing the training and validation curve. (b) The baseline model detected events compared with the current MyShake DBSCAN method. (c) the histogram of the detection time difference between MyShake DBSCAN method and the baseline CNN model, the mean, median and standard deviation are -0.42s, -0.12s and 2.51s. (d) The histogram of the detection time difference between ShakeAlert and the baseline CNN model, the mean, median and standard deviation are 0.67s, 0.8s and 3.29s.

4.2.2 Results from the ConvLSTM model

Figure 11 shows the equivalent performance summary for the ConvLSTM model on the 64 real events with the best model at epoch 47. Overall, this model detects 32 events comparing the baseline 28 events. It also shows slightly faster detection time than the baseline CNN model.

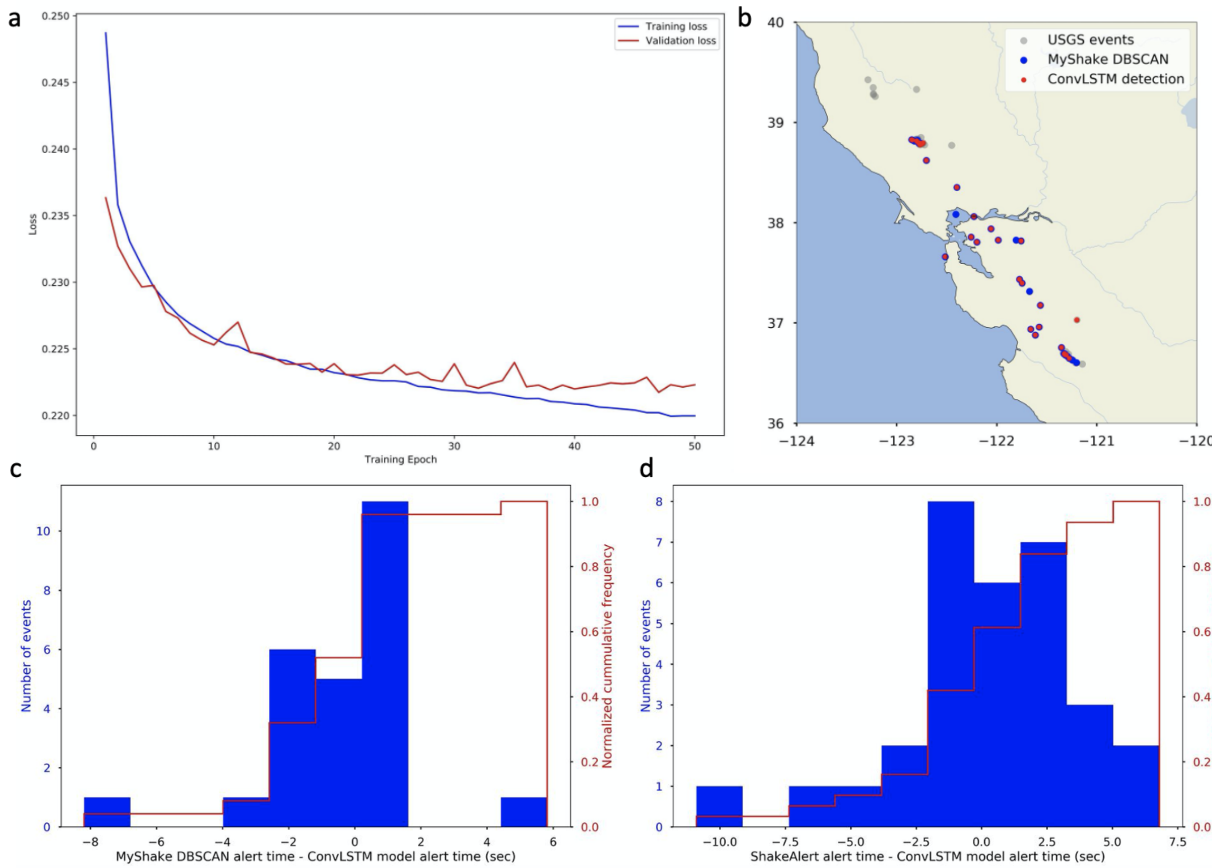

Figure 11. Summary of the ConvLSTM performance. (a) training curve showing the training and validation curve. (b) The ConvLSTM model detected events compared with the current MyShake DBSCAN method. (c) the histogram of the detection time difference between MyShake DBSCAN method and the ConvLSTM model, the mean, median and standard deviation are -0.39s, 0.11s and 2.30s. (d) The histogram of the detection time difference between ShakeAlert and the ConvLSTM model, the mean, median and standard deviation are 0.18s, 0.5s and 3.48s.

Figure 12 gives four examples from different magnitude earthquakes with simulated phone triggers. We can clearly see that when there are more triggers, the patterns are easily recognized by the model, while cases with fewer phone triggers are not easy to detect for the models.

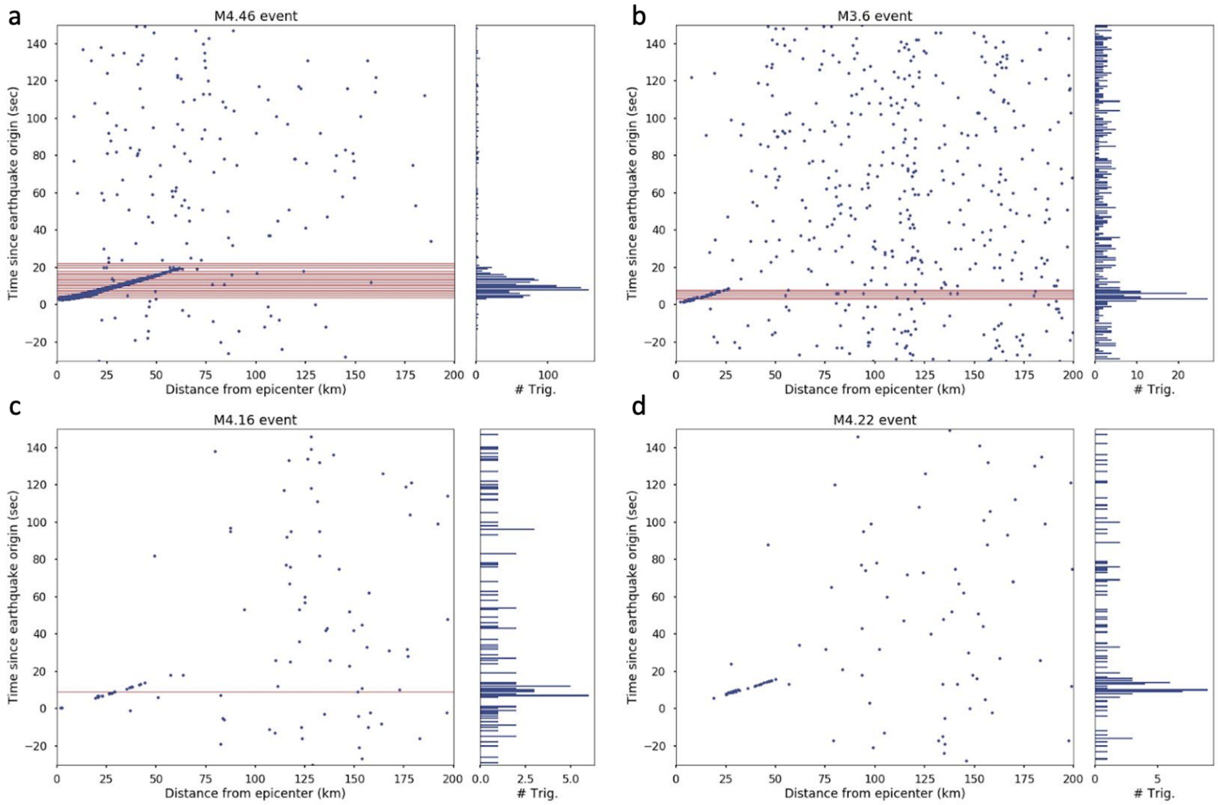

Figure 12. Detection Examples for 4 different events. The left panel in each figure shows the trigger distances v.s. Time since origin, each dot is a trigger. The red lines are the time when the model detects there is an event. The right panel is the histogram of the triggers in 1s time bins. (a) USGS event id nc73291880. (b) USGS event id nc72768191. (c) USGS event id nc72766046. (d) USGS event id nc72663506.

5. Conclusions and discussions

This study is only the first step in the effort to combine the smartphone data with the seismic stations to improve the performance of the ShakeAlert system. We first use the simulation data to understand how fast we can improve the detection time by combining phone data with the ShakeAlert system in California. With 0.1% of the population contributing the phone data, we can see 134 events (35.7%) can be potentially faster, with a mean value of 2.2s faster than only using seismic stations. If we can increase the phone contributors to 10% of the population, we will see 270 events (72.0%) can be detected faster than only using the seismic stations. Furthermore, we can also see that the events that occurred close to population centers benefit more, which is more important to increase the warning time for these areas. We also compared two deep learning models using a relatively small dataset for prototype purposes, one is a standard CNN model that extracts spatial features from 3 consecutive images, and the other is the one with a ConvLSTM layer, which can extract both spatial and temporal features. We found the ConvLSTM model works better than the standard CNN model.

Due to the limitation of the time, there is still much work needed to make this method really reliable and accurate. Here we list out the main ones that need to be carried out.

- Optimize the model. The current tests only show the method is working, and to get the best results, finding the best combination of the hyperparameters will be important. Such as quality control parameters (number of phones is needed in the cells), number of images used in the ConvLSTM model, size of the grid cells etc.
- Dealing with abnormal cases: the current model trained using the simulation platform, which only generates random triggers. But in nature, some cases may generate a large number of triggers within a short time in a region, such as big thunder storms (lightning). Potential solution is to add in more training that triggers the cells at different triggering levels at different times, so that the model can learn roughly the seismic wave speed related trigger patterns.
- Adding in other earthquake parameters in the estimation. Currently, this model only detects the earthquake occurrence, without estimating the magnitude, location of the earthquake. My thinking for the location and magnitude of the earthquake, after the detection of the earthquake using the ConvLSTM, we can add a chain deep learning model with the same input of the trigger images (as used in the training ConvLSTM) as well as a grid cell image with median/mean amplitude in each cell. The output of this model will be the latitude, longitude and magnitude of the earthquake.
- Initially, we can set each seismic station trigger as ratio 1 in the cell, so that these high quality seismic stations play an important role in the detection. But tuning this as a parameter maybe provide better results.

6. Project materials

Codes for the simulation platform and the deep learning model training can be found in github repo: https://github.com/qingkaikong/USGS_G20AP00058

Data used in this study can be found in this repository: www.shorturl.at/knNRY

7. Acknowledgement

U.S. Geological Survey (USGS), Department of the Interior, under USGS award number G20AP00058. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the opinions or policies of the U.S. Geological Survey. Mention of trade names or commercial products does not constitute their endorsement by the U.S. Geological Survey. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract Number DE-AC52-07NA27344. This is LLNL Contribution Number LLNL-TR-830065. We also thank the previous and current MyShake team members: Roman Baumgaertner, Garner Lee, Arno Puder, Louis Schreier, Stephen

Allen, Stephen Thompson, Jennifer Strauss, Kaylin Rochford, Akie Mejia, Doug Neuhauser, Stephane Zuzlewski, Asaf Inbal, Sarina Patel and Jennifer Taggart for keeping this Project running and growing. All the analysis of this project is done in Python. We thank all the MyShake users who contribute to the project!

8. References

Allen, Richard M., Qingkai Kong, and Robert Martin-Short. 2019. “The MyShake Platform: A Global Vision for Earthquake Early Warning.” *Pure and Applied Geophysics*, October. https://doi.org/10.1007/s00024-019-02337-7.

Allen, Richard M., and Diego Melgar. 2019. “Earthquake Early Warning: Advances, Scientific Challenges, and Societal Needs.” *Annual Review of Earth and Planetary Sciences*, May. https://doi.org/10.1146/annurev-earth-053018-060457.

Apriani, M., S. K. Wijaya, and Daryono. 2021. “Earthquake Magnitude Estimation Based on Machine Learning: Application to Earthquake Early Warning System.” *Journal of Physics. Conference Series* 1951 (1): 012057.

Bossu, Rémy, Frédéric Roussel, Laure Fallou, Matthieu Landès, Robert Steed, Gilles Mazet-Roux, Aurélien Dupont, Laurent Frobert, and Laura Petersen. 2018. “LastQuake: From Rapid Information to Global Seismic Risk Reduction.” *International Journal of Disaster Risk Reduction* 28 (June): 32–42.

Fauvel, Kevin, Daniel Balouek-Thomert, Diego Melgar, Pedro Silva, Anthony Simonet, Gabriel Antoniu, Alexandru Costan, et al. 2020. “A Distributed Multi-Sensor Machine Learning Approach to Earthquake Early Warning.” *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (01): 403–11.

Finazzi, Francesco. 2016. “The Earthquake Network Project: Toward a Crowdsourced Smartphone‐based Earthquake Early Warning System.” *Bulletin of the Seismological Society of America* 106 (3): 1088–99.

Gasparini, Paolo, Gaetano Manfredi, and Jochen Zschau. 2007. *Earthquake Early Warning Systems*. Springer Science & Business Media.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Inbal, Asaf, Qingkai Kong, William Savran, and Richard M. Allen. 2019. “On the Feasibility of Using the Dense MyShake Smartphone Array for Earthquake Location.” *Seismological Research Letters*. https://doi.org/10.1785/0220180349.

Kohler, Monica D., Deborah E. Smith, Jennifer Andrews, Angela I. Chung, Renate Hartog, Ivan Henson, Douglas D. Given, Robert de Groot, and Stephen Guiwits. 2020. “Earthquake Early Warning ShakeAlert 2.0: Public Rollout.” *Seismological Research Letters*. https://doi.org/10.1785/0220190245.

Kong, Qingkai, Richard M. Allen, Monica D. Kohler, Thomas H. Heaton, and Julian Bunn. 2018. “Structural Health Monitoring of Buildings Using Smartphone Sensors.” *Seismological Research Letters* 89 (2A): 594–602.

Kong, Qingkai, Richard M. Allen, and Louis Schreier. 2016. “MyShake: Initial Observations from a Global Smartphone Seismic Network.” *Geophysical Research Letters* 43 (18): 9588–94.

Kong, Qingkai, Richard M. Allen, Louis Schreier, and Young-Woo Kwon. 2016. “MyShake: A Smartphone Seismic Network for Earthquake Early Warning and beyond.” *Science Advances* 2 (2): e1501055.

Kong, Qingkai, Robert Martin-Short, and Richard M. Allen. 2019. “Towards Global Earthquake Early Warning with the MyShake Smartphone Seismic Network Part 1 -- Detection Algorithm and Simulation Platform.” *arXiv [physics.geo-Ph]*. arXiv. http://arxiv.org/abs/1909.08136.

Kong, Qingkai, Robert Martin‐Short, and Richard M. Allen. 2020. “Toward Global Earthquake

Early Warning with the MyShake Smartphone Seismic Network, Part 1: Simulation Platform and Detection Algorithm.” *Seismological Research Letters* 91 (4): 2206–17.
Kong, Q., S. Patel, and A. Inbal. 2019. “Assessing the Sensitivity and Accuracy of the MyShake Smartphone Seismic Network to Detect and Characterize Earthquakes.” *Seismological Research Letters*. https://pubs.geoscienceworld.org/ssa/srl/article-abstract/90/5/1937/573153.
LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. “Deep Learning.” *Nature* 521 (7553): 436–44.
Lee, J., J. Kim, S. Choi, and Y. Kwon. 2019. “A Smart Device Using Low-Cost Sensors to Detect Earthquakes.” In *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 1–4.
Li, Zefeng, Men-Andrin Meier, Egill Hauksson, Zhongwen Zhan, and Jennifer Andrews. 2018. “Machine Learning Seismic Wave Discrimination: Application to Earthquake Early Warning.” *Geophysical Research Letters* 45 (10): 4773–79.
Meier, Men-andrin, Zachary E. Ross, Anshul Ramachandran, Ashwin Balakrishna, Suraj Nair, Peter Kundzicz, Zefeng Li, Jennifer Andrews, Egill Hauksson, and Yisong Yue. 2019. “Reliable Real-Time Seismic Signal/Noise Discrimination With Machine Learning.” *Journal of Geophysical Research, [Solid Earth]* 124 (1): 788–800.
Minson, Sarah E., Benjamin A. Brooks, Craig L. Glennie, Jessica R. Murray, John O. Langbein, Susan E. Owen, Thomas H. Heaton, Robert A. Iannucci, and Darren L. Hauser. 2015. “Crowdsourced Earthquake Early Warning.” *Science Advances* 1 (3): e1500036.
Saad, Omar M., Ali G. Hafez, and M. Sami Soliman. 2021. “Deep Learning Approach for Earthquake Parameters Classification in Earthquake Early Warning System.” *IEEE Geoscience and Remote Sensing Letters* 18 (7): 1293–97.
Schmidhuber, Jürgen. 2015. “Deep Learning in Neural Networks: An Overview.” *Neural Networks: The Official Journal of the International Neural Network Society* 61 (January): 85–117.
Shi, Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-Chun Woo. 2015. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting.” In *Advances in Neural Information Processing Systems*, edited by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf.
Voulodimos, Athanasios, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. 2018. “Deep Learning for Computer Vision: A Brief Review.” *Computational Intelligence and Neuroscience* 2018 (February): 7068349.