Machine Learning Aspects of the MyShake Global Smartphone Seismic Network

by Qingkai Kong, Asaf Inbal, Richard M. Allen, Qin Lv, and Arno Puder

ABSTRACT

This article gives an overview of machine learning (ML) applications in MyShake—a crowdsourcing global smartphone seismic network. Algorithms from classification, regression, and clustering are used in the MyShake system to address various problems, such as artificial neural network (ANN) and convolutional neural network (CNN) to distinguish earthquake motions, spatial–temporal clustering using density-based spatial clustering of applications with noise (DBSCAN) to detect earthquakes from phone aggregated information, and random forest regression to learn from existing physics-based relationships. Beyond existing efforts, this article also presents a vision of the role of ML in some new directions and challenges. Using MyShake as an example, this article demonstrates the promising combination of ML and seismology.

INTRODUCTION

MyShake is a new global smartphone-based seismic network that relies on crowdsourcing (Kong, Allen, Schreier, *et al.*, 2016). Since its public release in 2016, MyShake has covered six continents with about 296,000 downloads globally. Currently, 40,000 active phones carry the MyShake app, with about 6000 devices contributing data to our server on a daily basis. Data collected by these smartphones enabled new applications. For example, Kong, Allen, and Schreier (2016) observed that *P*-wave amplitudes exceed the noise level on phones located as far as 100 km from the epicenter of **M** 5.2 earthquakes, and that the amplitudes of seismic signals recorded on smartphones are similar to those recorded by high-quality seismometers. Therefore, earthquake parameters (magnitude, location, and origin time) could be estimated from the MyShake waveforms with reasonable accuracy (Q. Kong *et al.*, unpublished manuscript, see Data and Resources). A. Inbal *et al.* (unpublished manuscript, see Data and Resources) show that the spatial distribution of smartphone noise amplitude within the Los Angeles basin is correlated with anthropogenic sources such as major traffic highways, the airport, and the Long Beach seaport. The dense spatiotemporal coverage of the MyShake array paves the road for new applications such as building health monitoring and earthquake detection via array backprojection (Kong, Allen, *et al.*, 2018; A. Inbal *et al.*, unpublished manuscript, see Data and Resources).

Harvesting and analyzing seismic data from phones require complex tasks that could benefit from machine learning (ML). The first part of this article presents an overview of various ML-based applications implemented within the MyShake network. In the second part, we discuss various problems related to crowdsourcing of noisy seismic data, and present potential ML-based approaches for addressing them.

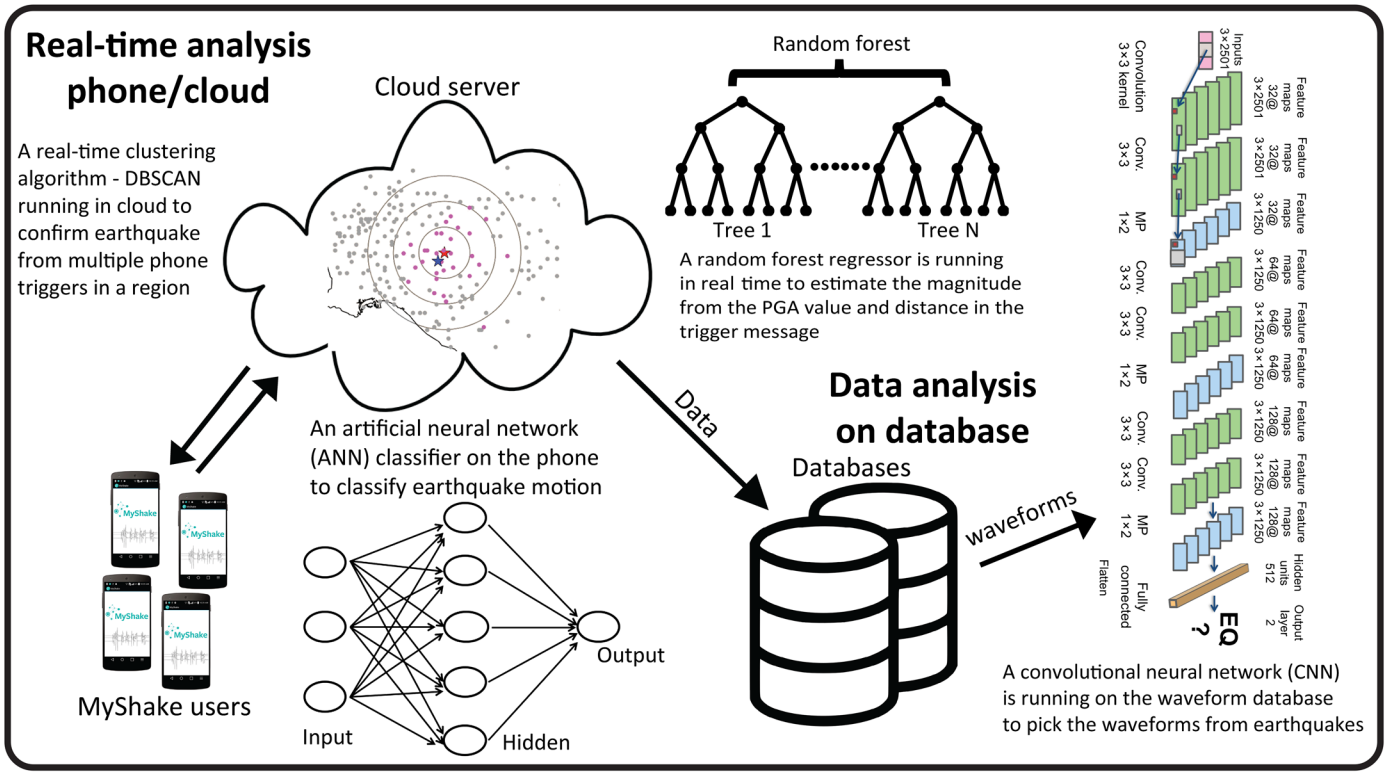
CURRENT APPLICATIONS

We start by briefly describing MyShake: A system designed to detect earthquakes in near-real time using sensors built into smartphones. In the current implementation, each phone reports the detection of earthquake-like motion to the cloud server with a short-trigger message including timestamp, location, and amplitude. Earthquake parameters are automatically determined based on in-cloud aggregation of many earthquake triggers. Ground motions recorded around the trigger time are also stored in the database for further analysis. The system architecture is described in Kong *et al.* (2015). Four existing ML algorithms are running in the MyShake system shown in Figure 1, which are summarized in the following two subsections.

Real-Time Analysis on Phone and Cloud (ANN, DBSCAN, and Random Forest)

Ground motion exceeding the phone noise level can be excited by a natural or anthropogenic source. To distinguish earthquake-like motion from daily human activities on a single phone, we formulate this problem as a binary classification problem. Both human activity data (from volunteers) and earthquake data (shake table and simulation data) were collected to form the training datasets. Different frequency domain, time domain, and statistical features were extracted from 2 s window three-component waveforms to capture the difference between those motions. To reduce the chance of overfitting and computation cost, only three best features were selected from 18 different features using a greedy forward feature selection method (Kuhn and Johnson, 2013). Besides classification accuracy, ease of implementation and computational cost are also considered as the feature selection criteria, due to the limited resources on phones. To reduce class bias, we used the K-means clustering algorithm (Hartigan and Wong, 1979) to downsample the majority class. We tested different ML algorithms and found that with carefully selected features, the various algorithms performed similarly well. Because the artificial neural network (ANN) algorithm can be more easily implemented on the phones, it was selected for real-

 doi: 10.1785/0220180309

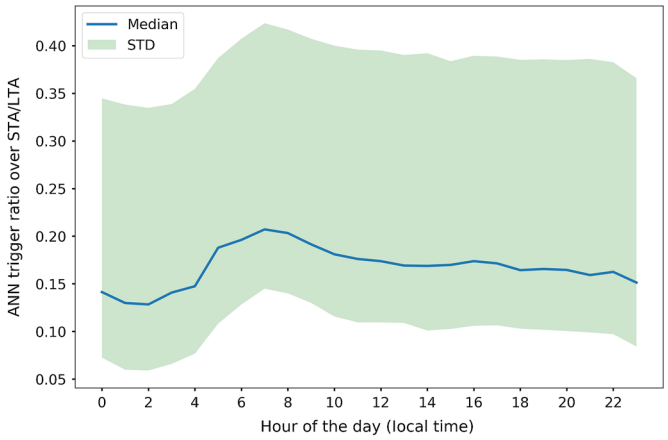


▲ **Figure 1.** Sketch overview of the MyShake system and the machine learning (ML) algorithms that are currently used or under testing in the system both in real time and offline modes. DBSCAN, density-based spatial clustering of applications with noise; PGA, peak ground acceleration. The color version of this figure is available only in the electronic edition.

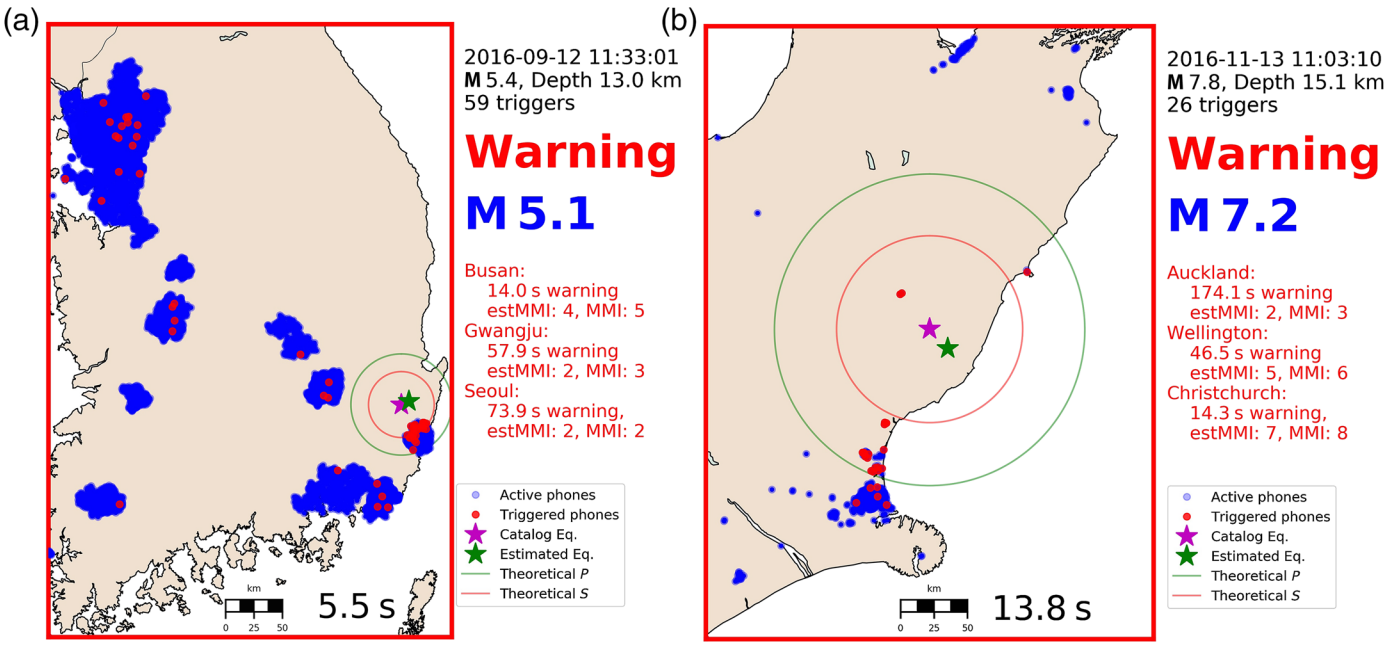
time applications. Details of the ANN design could be found in Kong, Allen, Schreier, *et al.* (2016).

The final ANN structure is shown in Figure 1, which consists of three input features for the input layer, a five-neuron hidden layer, and a binary output layer. This simple structure allows the ANN detector on the MyShake phones to capture most characteristics common to signals generated by earthquakes and human activities as shown in figure 4c,d in Kong, Allen, Schreier, *et al.* (2016). The current MyShake app needs to be stationary first, which is determined by a simple short-term average/long-term average (STA/LTA) algorithm. Once the STA/LTA triggers, it activates the ANN algorithm to distinguish the movements. Following the public release of MyShake, we found that the false positive rate changes with time, as shown in Figure 2. The STA/LTA algorithm on the phone triggers whenever the phone moves, whereas the ANN algorithm only triggers when the movement is similar to that of earthquakes. The ratio between these two indicates the average ANN false detection rate in the real world. Specifically, from 10 p.m. to 5 a.m., the ratio is mostly below 10%; whereas from 6 a.m. to 9 p.m., the trigger rate is between 10% and 20%. The majority of these ANN triggers are not due to earthquakes.

The second-level detection algorithm runs on the cloud server to collectively confirm an earthquake by considering clusters of users' triggers both in time and space. The current algorithm under testing is density-based spatial clustering



▲ **Figure 2.** Ratio of MyShake artificial neural network (ANN) triggers to short-term average/long-term average (STA/LTA) triggers. Data used here are from 1 July 2017 to 1 July 2018 in the San Francisco Bay area of California, with a total of 4853 unique users with 3,498,239 STA/LTA triggers and 399,903 ANN triggers. It shows the percentage of human triggers that passed the ANN check and were classified as earthquake-like motion for each hour of the day during this period. The line is the median value and the shaded area is the standard deviation. The color version of this figure is available only in the electronic edition.



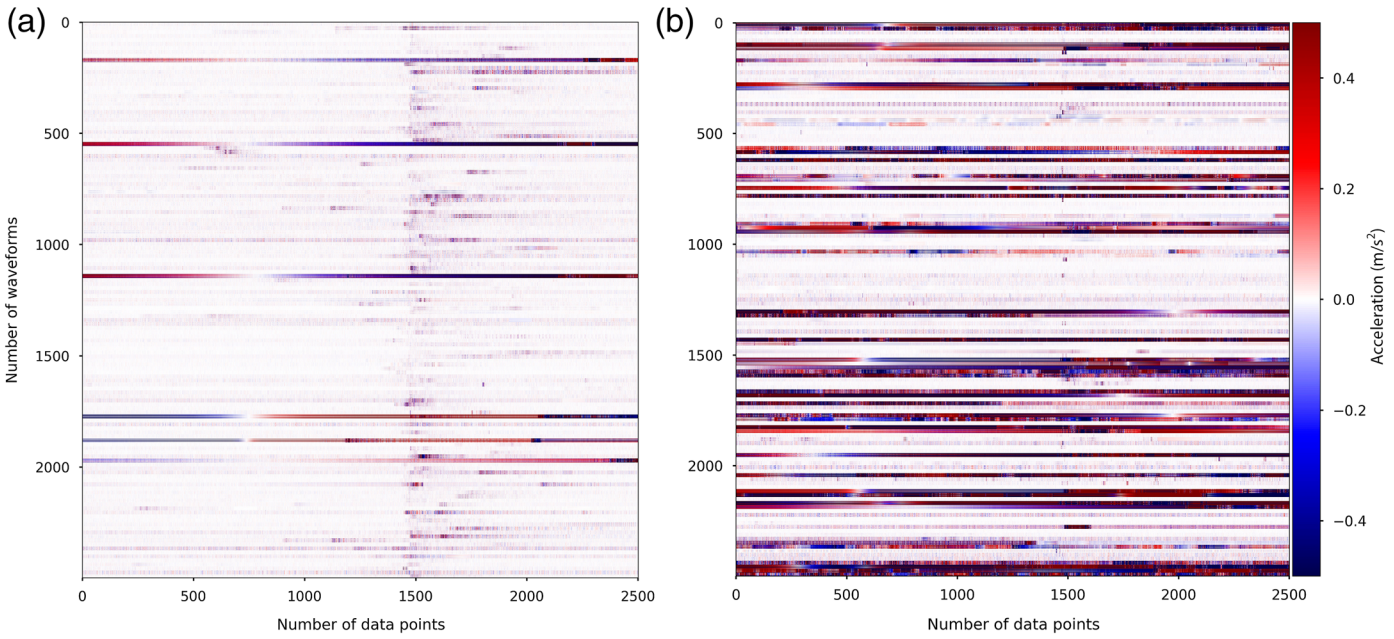
▲ **Figure 3.** (a) The **M** 5.4 South Korea event on 12 September 2016 11:32:55.770 (UTC) and (b) the **M** 7.8 Kaikōura earthquake in New Zealand on 13 November 2016 11:02:56.346 (UTC). The figures show the time our algorithm detected the earthquakes in simulations of these events, 5.5 and 13.8 s after the origin of the earthquakes, respectively. The magenta star is the catalog location whereas the green star is the estimated location using triggers from the phones. Blue dots are the active phones sampled from the population (0.001% of the total population), and red dots are the triggers. Red dots outside of the *P* (green circle) and *S* (red circle) waves are the noise triggers based on the observations from the MyShake network. The blue magnitude on the right of each figure is the estimated magnitude by the random forest regressor. Warning times, estimated, and true modified Mercalli intensities (MMIs) are shown for three cities. The color version of this figure is available only in the electronic edition.

of applications with noise (DBSCAN, Ester *et al.*, 1996). The algorithm has two parameters: epsilon and min_samples. The advantage of the algorithm is that there is no need to specify the number of clusters, and it can automatically find all the clusters that satisfy the requirement. The algorithmic steps are: (1) for each point in the dataset, we draw an n-dimensional sphere of radius epsilon around the point (assuming we have n-dimensional data). (2) If the number of points inside the sphere is larger than min_samples, we set the center of the sphere as a cluster, and all the points within the sphere belong to this cluster. (3) Loop through all the points within the sphere with the above two steps, and expand the cluster whenever it satisfies the two rules. (4) Points that do not belong to any cluster are ignored or treat them as outliers. To improve efficiency, regular grid cells of size 10 × 10 km are used to aggregate neighboring triggers occurring within a 20 s sliding window. We currently set epsilon to 200 km and min_samples to two grid cells. Once a cluster is formed (i.e., an earthquake is confirmed), the system activates a grid search on the triggers within the cluster to find the best earthquake location and origin time. The algorithm continues associating new triggers to the initial cluster until there are no new triggers. Once the epicenter is determined, a trained random forest regressor (Breiman, 2001) (1,000,000 randomly generated peak ground acceleration values at various distances for different magnitudes were used as the training data) is used to estimate the magnitude of the earthquake based on the attenuation relationship from Cua (2005).

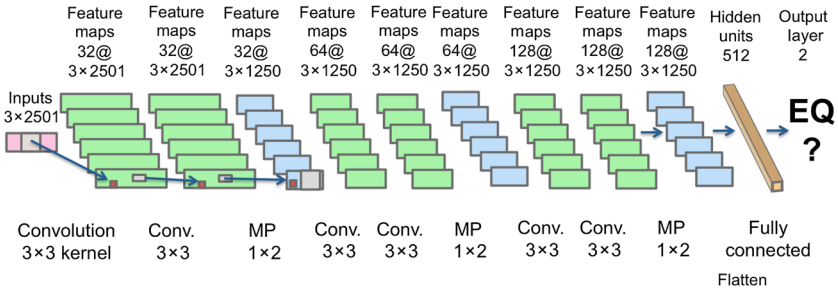
We are currently working on a MyShake trigger simulation platform, which builds on top of the MyShake observations to test the detection algorithm at various locations globally. Two examples of running the network detection algorithm are shown in Figure 3.

Data Analysis on the Waveform Database (CNN)

The ANN algorithm running on the phones uses only a 2s window of the waveform due to the real-time needs of earthquake detection. However, the classification procedure can be significantly improved by using the longer waveform data on the server (non-real time). Inspired by the work of Perol *et al.* (2018), a convolutional neural network (CNN) is trained by forming images using three-component waveforms to take advantage of CNNs' significant image processing power (LeCun *et al.*, 2010). Because the exact horizontal orientation of each phone is usually unknown, we permuted the three components of the waveforms to form images using the combination of (x, y, z), (x, z, y), (y, x, z), (y, z, x), (z, x, y), (z, y, x). Specifically, we added the three-component waveforms to an image with one color channel that results in a 3 by 2501 array (2501 data points sampled at 25 Hz with 60 s of pretrigger and 40 s after



▲ **Figure 4.** Images fed into the convolutional neural network (CNN). (a) Earthquake waveforms and (b) noise waveforms. Each figure has 2501 data points on the *x* axis and 2499 waveforms on the *y* axis, color coded by amplitude. Every three waveforms form an image that is fed into the CNN. We plot 2499 waveforms vertically for visualization purposes. The color version of this figure is available only in the electronic edition.



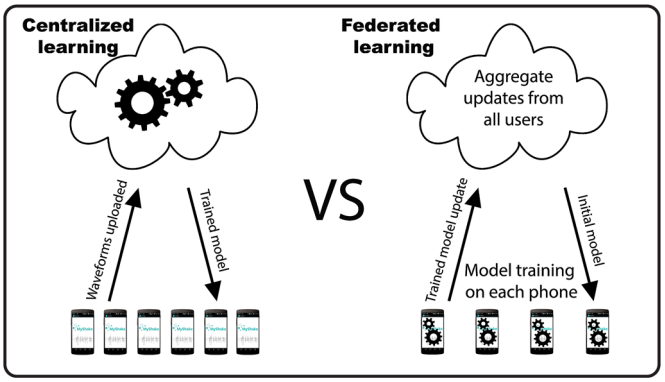
▲ **Figure 5.** The current CNN structure. Input data are images with 3 by 2501 dimensions. 32@3 × 2501 means that 32 feature maps are applied to the 3 by 2501 image. Conv. 3 × 3 means that a 3 by 3 kernel is used to calculate the feature maps. MP 1 × 2 means that max-pooling (1 by 2) is applied on the feature maps. The last layer is a flattened fully connected layer with 512 hidden units. The color version of this figure is available only in the electronic edition.

trigger waveforms). To increase the size of the training datasets and achieve better generalization, we augmented the data in the following ways: (1) adding different levels of random noises (a Gaussian noise with zero mean and standard deviation ranging from 0 to 0.01*g*), (2) rotating the two horizontal components at 5° incremental steps, (3) randomly flipping the signs of the three acceleration components, and (4) randomly shifting the signal by up to 2 s. These augmentations are based on the idea of the phones having different noise levels, arbitrary orientations, flipped and triggers at a different time. Altogether 1.5 million records were used in training and testing (75% training). The noise distribution was randomly sampled from the MyShake nonearthquake triggers.

Figure 4 shows the formed images for earthquakes (Fig. 4a) and nonearthquakes (Fig. 4b). The earthquake waveforms clearly have different characteristics. Waveforms are only processed by removing the mean and trend before forming the images. After the preparation step, the images were fed into the CNN to determine which images indicate earthquakes. We started with a simple network structure and gradually added more layers. The final structure is shown in Figure 5, which contains six convolutional layers as well as three max-pooling layers, and there are multiple dropout operations to reduce overfitting. A fully connected layer at the end makes the final binary decision. The overall accuracy for the test dataset is 96.77%.

NEW DIRECTIONS AND CHALLENGES

The above ML applications show the effectiveness of finding nonlinear decision boundaries to make decisions (ANN, CNN), identifying spatial clustering and associations efficiently (DBSCAN), and learning the complex physics-based functions (random forest). Here, we discuss a few untested ideas for new directions and challenges that we will work on. This serves as an ML vision for the project.



▲ **Figure 6.** Two approaches for training customized models: centralized learning versus federated learning. The color version of this figure is available only in the electronic edition.

One Algorithm Cannot Fit All

Currently, the ANN algorithm running on users' phones is the same for all phones. But each user's behavior is different, male versus female, young versus old, and so on. Besides, the hardware qualities used in the phones have a very wide spectrum. Ideally, we would like to train a customized model for each person to capture these details or a shared model that fits groups of people. Two potential approaches are shown in Figure 6, which are (1) centralized learning: all users upload a few days' human activities to the central server and customized classifiers are trained at the server. The trained models are then pushed back to the phones. (2) Federated learning (Konečný *et al.*, 2016; McMahan *et al.*, 2016): the users download an initial model, and retrain the model locally on each phone. Only summaries of the changes are sent to the centralized server as a small focused update. These updates from each user can be aggregated to make improvements to the initial model to form a new shared model.

Currently, we are testing both approaches. A test version of an Android application has been created that is able to update the model from the server, and a TensorFlow framework (Abadi *et al.*, 2016) has been built on Android phones to allow for training the neural network on the phones.

Dynamic Networks

Unlike traditional seismic networks where the locations of the sensors are fixed, the MyShake network can change all the time due to the movement of the users. Spatially, the sensors can move from city to city. Temporally, each hour during the day, the number of sensors which are stationary (best for detecting earthquakes) may vary. Figure 7 shows the spatial and temporal dynamic nature of the network.

Figure 7a shows the spatial distribution of the MyShake users. We can clearly see the uneven spatial distribution of the users, which can cause the network to perform better at places where more phones are available. In addition, the configuration of the network is changing. For example, Figure 7b shows the percentage of phones that are best for detecting earthquakes (i.e., steady for more than 30 min) during each hour of the day. We see that the network has the best detection capability from midnight to 6 a.m., with over 70% of the phones being steady, whereas during the day (10 a.m. to 8 p.m.), only about 20% of the phones are in steady positions to record good waveforms.

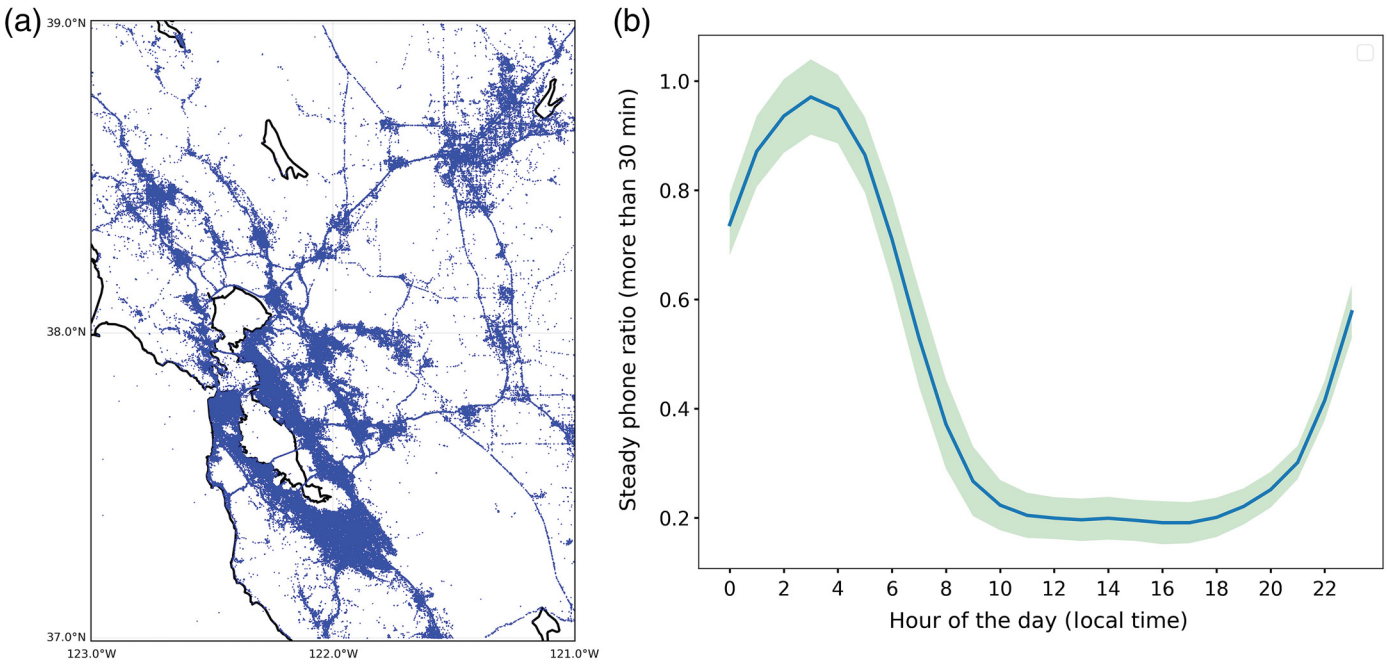
In summary, such spatial and temporal dynamics require an adaptive detection algorithm that could change its parameters accordingly. One promising approach is to apply an ML algorithm that learns the mapping function between the dynamic configuration of the network and the detection parameters so that the detection algorithm can quickly adapt to various situations in the real world.

Spoofing the System

Real-time earthquake early warning could potentially save lives and reduce economic losses (Strauss and Allen, 2016). But false alarms caused by spoofing attacks could generate panic and economic losses as well. Attacks can occur at any layer but for the context of this article we assume that the backend infrastructure is secured via traditional means (e.g., firewalls). Therefore, it is important to understand and address potential spoofing of earthquake triggers that may occur in the real world. Specifically, there can be three different types of spoofing activities and potential risks or vulnerabilities of the system against such spoofed earthquake triggers: (1) mimicking earthquake-like movements on individual smartphones to generate false triggers and trick the ANN algorithm into thinking an earthquake is occurring, (2) injecting false triggers into the system from one or multiple independent users/phones, and (3) injecting false triggers into the system at coordinated time and locations from multiple colluding users/phones. Identifying these potential risks and solving them is critical. Some potential solutions that go beyond traditional means to secure the system include adversarial machine learning, which is the study of effective ML techniques against an adversarial opponent (Huang *et al.*, 2011; Tygar, 2011). Instead of trying to make a better model, the first step is to break the trained model by thoroughly understanding the input data, feature extraction, training, and the learning algorithm, and test various cases that could break the ML algorithms used in the system. In addition, data-driven approaches can be coupled with physics-driven approaches. We could incorporate the physical model of how earthquake waves propagate and utilize the patterns behind it to add additional validation checks to capture spoofing attacks.

A Generic Sensor Collection and Fusion Platform

We hope the MyShake system/platform is just one step in the seismology community to include more low-cost consumer sensors. Various Internet of things devices, such as the accelerometers on cars, voice assistants, sensors at smart homes, Raspberry Pi type sensors, drone videos, closed-circuit television (CCTV) cameras at home and cities are capable of recording the environment and have built-in communication units to pass data to servers. Based on the MyShake experience, we hope in the future we could start to host other types of sensor data and generalize the workflow pipeline to deal with various types



▲ **Figure 7.** Spatial and temporal dynamics of the MyShake network. (a) The footprint of the MyShake users in the San Francisco Bay Area, California, U.S.A., and the dots are user locations reported in the heartbeat messages (modified from Kong, Inbal, *et al.*, 2018). (b) The percentage of phones that is steady for more than 30 min during each hour of the day. The line is the average percentage, whereas the shaded area is the standard deviation. MyShake user data from 1 July 2017 to 1 July 2018. The color version of this figure is available only in the electronic edition.

of data on this platform. Data fusion using ML could provide an effective solution to take advantage of information from various data sources (Torra, 2003; Ni-Bin Chang, 2018). There is high potential in combining the various datasets to extract extra information using ML algorithms. For example, a straightforward approach in feature-level fusion is to extract features individually from each data source and feed them into a unified ML algorithm to complete the tasks.

CONCLUSION

In this overview article of MyShake's ML aspects, we presented some existing efforts that apply ML to this new type of seismic network to address various problems. Selected new challenges and directions are also discussed here in the hope to motivate more discussions on applying ML in Earth sciences, particularly in seismology. Some of these challenges require us to collaborate with other communities, such as computer science, statistics, and data science. At the same time, the problems in seismology and the data we are collecting really could drive the development of ML and data science in the future, and the MyShake seismic network is just one of these examples in our field.

DATA AND RESOURCES

Data recorded by MyShake are currently archived at Berkeley Seismological Laboratory and are constrained by the privacy policy of MyShake (see http://myshake.berkeley.edu/privacy-policy/index.html, last accessed November 2018). For information about access to the data for research purposes contact rallen@berkeley.edu. The unpublished manuscript by A. Inbal, Q. Kong, W. Savran, and R. M. Allen, "Toward microseismic imaging with the dense MyShake smartphone array" and Q. Kong, A. Inbal, S. Patel, R. M. Allen, and L. Schreier, "MyShake: Detecting and characterizing earthquakes with a global smartphone seismic network."

ACKNOWLEDGMENTS

MyShake is a joint collaboration between the Berkeley Seismology Laboratory and Deutsche Telecom Silicone Valley Innovation Center. The Gordon and Betty Moore Foundation funded this project through Grant Number GBMF5230 to University of California, Berkeley (UC Berkeley). The authors thank the MyShake team members: Roman Baumgaertner, Garner Lee, Louis Schreier, Stephen Allen, Stephen Thompson, Akie Mejia, Jennifer Strauss, Kaylin Rochford, Doug Neuhauser, Stephane Zuzlewski, Sarina Patel, and Jennifer Taggart for keeping this project running and growing. The authors also thank all the MyShake users who contribute to the project.

REFERENCES

Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.* (2016). Tensorflow: A system for large-scale machine learning, *Proc. of the 12th USENIX Sympo-*

sium on Operating Systems Design and Implementation (OSDI'16), 265–283.
Breiman, L. (2001). Random forests, *Machine Learn.* **45,** no. 1, 5–32.
Cua, G. B. (2005). Creating the virtual seismologist: Developments in ground motion characterization and seismic early warning—Caltech, *Ph. D. Thesis*, available at https://thesis.library.caltech.edu/572/ (last accessed November 2018).
Ester, M., H. P. Kriegel, J. Sander, and X. Xu (1996). A density-based algorithm for discovering clusters in large spatial databases with noise, *Proc. of the Second International Conference on Knowledge Discovery and Data Mining*, 226–231.
Hartigan, J. A., and M. A. Wong (1979). Algorithm AS 136: A K-means clustering algorithm, *J. Roy. Stat. Soc. Ser. C Appl. Stat.* **28,** no. 1, 100, doi: 10.2307/2346830.
Huang, L., A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar (2011). Adversarial machine learning, *Proc. of the 4th ACM workshop on Security and artificial intelligence—AISec'11*, ACM Press, Chicago, Illinois, USA, 21 October 2011, doi: 10.1145/2046684.2046692.
Konečný, J., H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon (2016). Federated learning: Strategies for improving communication efficiency, available at http://arxiv.org/abs/1610.05492 (last accessed November 2018).
Kong, Q., R. M. Allen, M. D. Kohler, T. H. Heaton, and J. Bunn (2018). Structural health monitoring of buildings using smartphone sensors, *Seismol. Res. Lett.* **89,** no. 2A, 594–602, doi: 10.1785/0220170111.
Kong, Q., R. M. Allen, and L. Schreier (2016). MyShake: Initial observations from a global smartphone seismic network, *Geophys. Res. Lett.* **43,** no. 18, 9588–9594, doi: 10.1002/2016gl070955.
Kong, Q., R. M. Allen, L. Schreier, and Y.-W. Kwon (2016). MyShake: A smartphone seismic network for earthquake early warning and beyond, *Sci. Adv.* **2,** no. 2, e1501055, doi: 10.1126/sciadv.1501055.
Kong, Q., A. Inbal, R. Allen, and J. Strauss (2018). MyShake: Building a global smartphone earthquake early-warning system, *SEG Technical Program Expanded Abstracts 2018, Society of Exploration Geophysicists*, 4867–4871, doi: 10.1190/segam2018-2996624.1.
Kong, Q., Y.-W. Kwony, L. Schreierz, S. Allen, R. Allen, and J. Strauss (2015). Smartphone-based networks for earthquake detection, *2015 15th International Conference on Innovations for Community Services (I4CS)*, IEEE, Nuremberg, Germany, 8–10 July 2015, doi: 10.1109/I4CS.2015.7294490.
Kuhn, M., and K. Johnson (2013). An introduction to feature selection, in *Applied Predictive Modeling*, Springer, New York, New York, 487–519.
LeCun, Y., K. Kavukcuoglu, and C. Farabet (2010). Convolutional networks and applications in vision, *Proc. of 2010 IEEE International Symposium on Circuits and Systems*, IEEE, Paris, France, 30 May–2 June 2010, doi: 10.1109/ISCAS.2010.5537907.
McMahan, H. B., E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas (2016). Communication-efficient learning of deep networks from decentralized data, Vol. 54, available at http://arxiv.org/abs/1602.05629 (last accessed November 2018).
Ni-Bin Chang, K. B. (2018). *Multisensor Data Fusion and Machine Learning for Environmental Remote Sensing*, CRC Press, Boca Raton, Florida.
Perol, T., M. Gharbi, and M. Denolle (2018). Convolutional neural network for earthquake detection and location, *Sci. Adv.* **4,** no. 2, e1700578, doi: 10.1126/sciadv.1700578.
Strauss, J. A., and R. M. Allen (2016). Benefits and costs of earthquake early warning, *Seismol. Res. Lett.* **87,** no. 3, 765–772, doi: 10.1785/0220150149.
Torra, V. (2003). Trends in information fusion in data mining, in *Information Fusion in Data Mining*, Springer Berlin Heidelberg, Heidelberg, Germany, 1–6.
Tygar, J. D. (2011). Adversarial machine learning, *IEEE Internet Comput.* **15,** no. 5, 4–6, doi: 10.1109/mic.2011.112.

Qingkai Kong
Berkeley Seismological Laboratory
University of California, Berkeley
209 McCone Hall
Berkeley, California 94720 U.S.A.
kongqk@berkeley.edu

Asaf Inbal
Department of Geophysics
Tel Aviv University
Ramat-Aviv, Tel-Aviv 69978
Israel

Richard M. Allen
Berkeley Seismological Laboratory
University of California, Berkeley
279 McCone Hall
Berkeley, California 94720 U.S.A.

Qin Lv
Department of Computer Science
University of Colorado Boulder
430 UCB
Boulder, Colorado 80309 U.S.A.

Arno Puder
Computer Science Department
San Francisco State University
1600 Holloway Avenue
San Francisco, California 19132 U.S.A.

Published Online 5 December 2018