Earthquake Early Warning with Graph Learning

Tim Clements[1], Elizabeth Cochran[2], Annemarie Baltay[1], Sarah Minson[1], and Clara Yoon[2]
[1]U.S. Geological Survey, Earthquake Science Center, Moffett Field, CA, USA
[2]U.S. Geological Survey, Earthquake Science Center, Pasadena, CA, USA
tclements@usgs.gov

The goal of Earthquake Early Warning (EEW) is to characterize when and where strong ground motion is expected as soon as possible after an earthquake starts. The EEW systems currently in use across the world, such as ShakeAlert or the Japan Meteorological Agency's EEW system, either rapidly estimate earthquake source properties (e.g., location and magnitude) then predict shaking using ground motion attenuation relations, or forward predict ground motion as it travels across a seismic network. We introduce the Graph Prediction of Earthquake Shaking (GRAPES) EEW algorithm, which characterizes EEW as a spatio-temporal graph learning problem. Here the nodes of the graph are seismometers, the edges of the graph are connections between neighboring seismometers, the inputs at each node in the graph are 4-second windows of real-time seismic waveforms, and the prediction target for the graph is the peak ground acceleration at each station (node) in the network over the next 40 seconds. Our GRAPES model predicts future shaking using a set of convolutional, fully connected, and graph neural network layers. We train our model in an end-to-end fashion on a dataset of earthquake and ambient noise recorded by the KiK-net and K-NET strong motion networks in Japan. We test and compare GRAPES' ground motion prediction accuracy and warning-time performance to ShakeAlert's during the 2019 M7.1 Ridgecrest, CA earthquake. We show that GRAPES predicts ground motion up to 5 seconds faster and with up to 0.8 MMI units less error, on average, than ShakeAlert. We attribute GRAPES prediction timeliness and accuracy to its ability to predict ground motion without estimating a magnitude and its tolerance of unexpected temporary increases in data latency.