

Estimation of Site Amplification from Geotechnical Array Data Using Neural Networks

Daniel Roten^{*1} and Kim B. Olsen¹

ABSTRACT

We use deep learning to predict surface-to-borehole Fourier amplification functions (AFs) from discretized shear-wave velocity profiles. Specifically, we train a fully connected neural network and a convolutional neural network using mean AFs observed at ~ 600 KiK-net vertical array sites. Compared with predictions based on theoretical SH 1D amplifications, the neural network (NN) results in up to 50% reduction of the mean squared log error between predictions and observations at sites not used for training. In the future, NNs may lead to a purely data-driven prediction of site response that is independent of proxies or simplifying assumptions.

KEY POINTS

- We train deep neural networks to predict observed surface-to-borehole amplification functions.
- We achieve a 50% reduction in prediction error with respect to theoretical 1D SH amplifications.
- Deep learning may lead to empirical predictions of site responses without relying on proxies or assumptions.

INTRODUCTION

The densification of seismic networks, such as the California Strong Motion Instrument Program strong-motion network in California and the KiK-net observatory in Japan, have vastly increased the number of earthquake records available for strong motion research. In addition, the deployment of borehole accelerometers at many locations has resulted in a large volume of vertical array data, which has contributed to a better understanding of linear and nonlinear site response during strong shaking (e.g., Thompson *et al.*, 2010; Bonilla *et al.*, 2011; Roten *et al.*, 2013, 2014). However, despite the increased amount of data, the standard deviations of intensity measures in ground motion prediction equations (GMPEs) have barely decreased over the past four decades (Douglas, 2003; Strasser *et al.*, 2009). Standard deviations in GMPEs remain high because empirical methods use very simple models to approximate highly complex wave propagation phenomena (Bommer and Abrahamson, 2006). Site conditions in most GMPEs are typically reduced to the average velocity in the top 30 m, V_{S30} and in some cases basement depth (e.g., the depth to a constant shear-wave velocity of 1 km/s, Z_1) (Abrahamson *et al.*, 2014). Similarly, ground motions recorded on vertical arrays have demonstrated the shortcomings of current site response prediction techniques, in particular, the assumption of a laterally constant medium

(Thompson *et al.*, 2009, 2010, 2012). Three-dimensional simulations with sophisticated structural models and nonlinear wave propagation codes are needed to study the response of such sites (e.g., Gatti *et al.*, 2018; Hu *et al.*, 2020). Although such case studies may shed light on the wave propagation effects behind the site response observed at a particular location, it is not clear how this approach can be generalized to sites for which no sophisticated 3D velocity models are available. Clearly, new methods that harness the sheer volume of strong-motion data (including data acquired on vertical arrays) are needed to reduce standard deviations of intensity measure predictions.

Although seismology has always been a data intensive field, enormous amounts of data are currently being collected in a broad spectrum of fields ranging from technology to finance to healthcare. Combined with increasingly powerful computers, the availability of these very large datasets has been driving progress in machine learning (ML) techniques, in particular, deep learning applications, which thrive under large amounts of data.

An exciting aspect of deep neural networks (NNs) is their ability to detect patterns in the input data that allows them to make sense of labeled output data. In contrast to shallow learning algorithms, deep NNs are less dependent on feature engineering, that is, the process of transforming input data into features from which the output can be derived using a simple mathematical expression. In site response prediction, one could think of proxies such as V_{S30} or Z_1 as engineered features needed to carry out regression analysis for calibration of traditional GMPEs (i.e., a

1. Department of Geological Sciences, San Diego State University, San Diego, California, U.S.A., <https://orcid.org/0000-0002-3078-485X> (KBO)

*Corresponding author: droten@sdsu.edu

Cite this article as Roten, D., and K. B. Olsen (2021). Estimation of Site Amplification from Geotechnical Array Data Using Neural Networks, *Bull. Seismol. Soc. Am.* **XX**, 1–11, doi: [10.1785/0120200346](https://doi.org/10.1785/0120200346)

© Seismological Society of America

shallow learning method). A deep learning algorithm would not depend on such engineered features and could process the entire velocity information available for a site without resorting to simplifications that discard valuable data. The idea is that the network will identify new features from the provided velocity profile that will guide a more accurate site response prediction compared with proxies such as V_{S30} or Z_1 .

NNs have been proposed for prediction of site response in previous studies (e.g., [Derras et al., 2012, 2014](#); [Withers et al., 2020](#)). However, to our knowledge, all previous ML models for site response rely on the same proxies typically used in classical empirical models. For example, the NN trained by [Derras et al. \(2014\)](#) and [Withers et al. \(2020\)](#) used the V_{S30} to characterize site conditions, while [Derras et al. \(2012\)](#) used the V_{S30} along with the fundamental resonance frequency f_0 .

In this study, we propose training a deep NN to learn the observed mean site response at KiK-net sites based on the entire soil stratigraphy. The goal is to develop a method that predicts site response based entirely on observed vertical array data, without relying on proxies (such as V_{S30}) or simplifying assumptions (such as 1D or a vertically incident wave field) made in traditional site response assessment.

We have arranged the content of our article as follows. In the [Deep Learning and Design of NNs](#) section, we provide a quick overview of deep learning and elaborate on the design of the NNs. In the [Data Preparation](#) section, we describe the data preparation methods used in the calculation of transfer functions from KiK-net strong-motion sites. Results of site response prediction using the deep learning are presented in the [NN Training and Prediction Results](#) section.

DEEP LEARNING AND DESIGN OF NNS

Artificial neural networks (ANNs) are modeled after biological NNs found in animal brains and consist of a collection of artificial neurons interacting with each other. ANNs are typically organized in layers, and every ANN consists of an input layer accepting the input parameters and an output layer that produces the desired prediction.

Overview of NNs

Deep NNs feature at least one but typically several hidden layers located between the input and output layers. In a fully connected ANN, also called multilayer perceptron (MLP), each artificial neuron in each layer is connected to every other neuron of the previous and next layers (Fig. 1). Therefore, each neuron receives an input signal from every neuron in the previous layer and redirects a modified signal to every neuron in the next layer. The input function $z(\mathbf{x})$ into a neuron consists of a weighted sum of the inputs $\mathbf{x} = (x_1, x_2, x_3, \dots, x_m)$ from the m individual neurons of the previous layer:

$$z(\mathbf{x}) = \mathbf{w}\mathbf{x} + b,$$

in which \mathbf{w} is a vector describing the weights of each neural connection. The offset b is also called bias. If the current layer has n nodes and the previous layer has m nodes, there are $n \times m$ connections from the current to the previous layer, and the weight matrix connecting the two nodes has shape $n \times m$. In addition, there are n offsets that must be trained. The weights \mathbf{w} and offset b are trainable parameters.

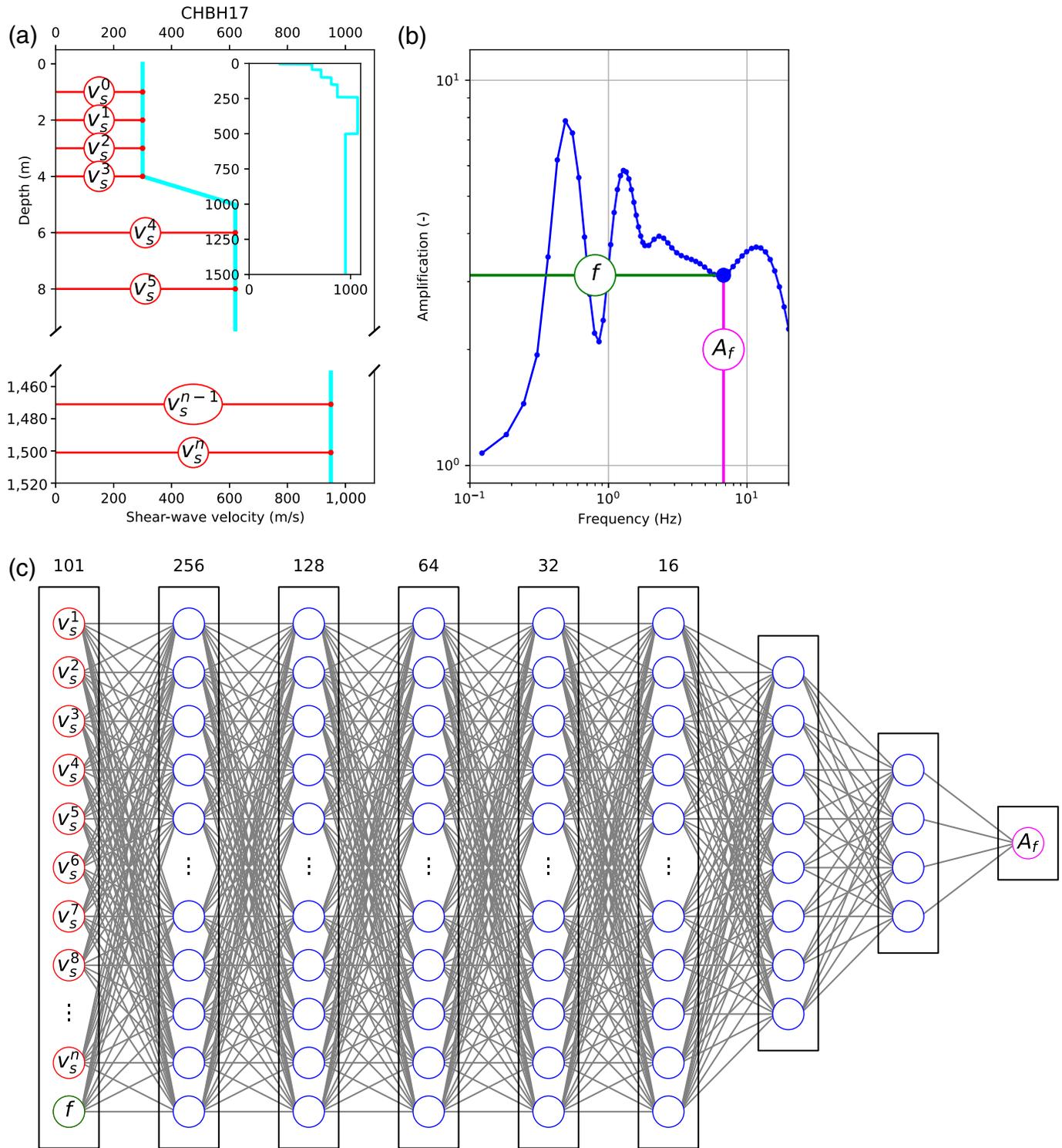
Activation functions allow the ANN to learn nonlinear functions. Without activation functions, the total output of the ANN would represent a linear function regardless of the depth of the networks (i.e., regardless of the number of hidden layers). Typical choices of activation functions include sigmoids, rectified linear units (ReLU), or hyperbolic tangent functions (tanh) ([Goodfellow et al., 2016](#)). If the ANN is used for regression, the output node uses a linear activation, allowing the ANN to output any real number.

The weights and biases are optimized by training the ANN. The goal of training is to minimize the loss function, which quantifies the difference between the desired output provided in the training data and the network's actual output. Forward propagation in a feedforward ANN refers to the computation of the network's output value based on the chosen input and the ANN's current weights and biases, with information flowing from the input to the output layer. This order is reversed during backpropagation, for which the gradient of the loss function with respect to the ANN's weight is computed based on the input and desired output of one or several training examples. Training consists of minimizing the loss by performing gradient descent on the loss function.

Because there are many trainable parameters in an ANN and the number of training examples is often limited, deep NNs are prone to overfitting (e.g., [Goodfellow et al., 2016](#)). An overfitted model will perform very well on the input set but will generalize poorly to the test set, with low misfit error on the training set but high error on the test set (i.e., the model exhibits high variance). Overfitting also affects inversion problems encountered within different domains of seismology, such as seismic tomography (e.g., [Nolet, 2008](#)). A common technique to reduce variance (overfitting) in such scenarios is to add L1 or L2 regularization ([Hastie et al., 2009](#)), which penalizes large weights and thereby reduces the number of free parameters in the model. Although this type of regularization can also be applied to deep NNs, it is more common to reduce variance using a technique called dropout ([Srivastava et al., 2014](#)). In dropout regularization, a predefined fraction of neurons is randomly eliminated during each training iteration. This prevents the network from relying on a single feature and allows it to generalize better to data it has not encountered during training.

MLP architecture

In the fully connected ANN design (also called MLP) used in this study, the input layer expects the shear-wave velocities



extracted at 100 predefined depths from the soil profile (Fig. 1a,c); the sampling interval gradually increased from 1 m near the surface to 30 m at 1500 m depth. The properties of the last layer were projected onto the remaining depth intervals at shallower sites. The desired frequency of site amplification was also provided to the ANN algorithm and represented the last value in the input layer. The output layer consisted of a single neuron with the site amplification value at the specified

Figure 1. Prototype of artificial neural network (ANN) for prediction of simulated transfer functions. (a) Shear-wave velocities (V_s , red nodes) were discretized at $n = 100$ depths and (c) fed into the input layer along with the frequency f of amplification (green node). Hidden layers in (c) are shown by blue neurons. Where not all nodes are shown, the true number of nodes is given at the top of the layer. (b) The output node contains the amplification A_f at the specified frequency. The color version of this figure is available only in the electronic edition.

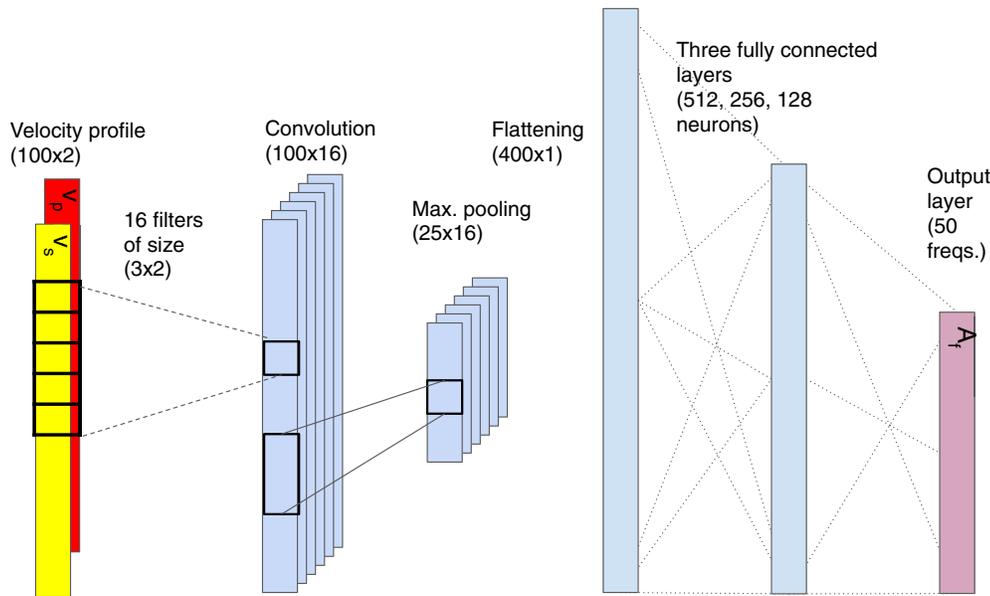


Figure 2. Schematic of convolutional neural network (CNN) design for prediction of site response. For simplicity, only one of the three fully connected layers between the flattened layer and the output layer is shown. The output layers specifies the amplification function (AF) at 50 frequencies (many-to-many design). The color version of this figure is available only in the electronic edition.

frequency (Fig. 1b,c). Our design chosen for the fully connected NN (MLP, Fig. 1c) used a many-to-one layout, accepting many inputs but producing just a single output value. That is, the ANN only predicted the amplification at one frequency at a time. One could also design a similar network using a many-to-many configuration and predict the amplification at several frequencies at the same time. We experimented with both many-to-one and many-to-many designs and found that the many-to-one configuration was superior to the many-to-many configuration for the MLP. However, a many-to-many design was adopted for the convolutional neural network (CNN) described subsequently.

Our MLP used seven hidden layers, and the number of neurons decreased gradually from 256 nodes in the first hidden layer toward the single-node output layer. Following DeVries *et al.* (2017), the activation function assigned to the hidden layers alternated between hyperbolic tanh and ReLUs, and a linear activation function was applied at the output layer (Fig. 1).

CNN architecture

In a convolutional layer, nodes are not directly connected to nodes in the next layer. Instead, the data in the layer are convolved using a series of filters. The dimensions of the convolutional layer's output depend on the type of convolution (overlap and stepping size) and the number of filters, with each filter creating a new representation of the input data. However, as most convolutional layers use many filters, the output is typically large and downsampled in a pooling layer following the

convolutional layer. A set of convolutional and pooling layers may be followed by another set of convolutional and pooling layers, or the data are flattened and directed into a fully connected layer.

CNNs are especially effective for image recognition or classification problems because they are able to extract information from the spatial arrangement of the pixels. Although predefined filters have long been used in image processing, the effectiveness of CNN derives from the network's ability to optimize the filters depending on the training data. In other words, the filter parameters are optimized during backpropagation such that the features extracted by the different filters are effective at carrying out the CNN's task.

In our case, we applied a CNN to take advantage of the spatial information in the velocity profile. We used V_S and the P -wave velocity V_P as different image channels, analogous to the red, green, and blue channels used in image recognition. In contrast to image recognition, in which the input image is 3D (two spatial dimensions plus three channels), our input was only 2D (V_S and V_P at different depths, Fig. 2). We did not use densities because they were not provided for KiK-net profiles.

In our CNN design, we used a single convolutional layer with 16 filters of dimensions 5×2 (Fig. 2) right after the input layer (dimension 100×2 , with V_S and V_P at 100 predefined depths). The output of the convolutional layer consisted of 100×16 values, which we reduced to 25×16 values using a pooling layer. The output of the pooling layer was flattened and fed into a fully connected layer of 512 nodes. Two more hidden layers with 256 and 128 nodes followed. The output layer contained 50 nodes, which represented the desired amplification function (AF) at 50 predefined frequencies (Fig. 2). A ReLU function was used for activation right after the pooling layer, and we alternated between ReLU and tanh functions in the three fully connected layers. As in the MLP, a linear activation was used in the output layer. Dropout regularization was applied after each layer. Batch normalization was carried out before each activation to improve convergence (this was also done in the MLP). Both the MLP and the CNN were implemented with the Keras library for Python (Chollet, 2018) using the TensorFlow (Abadi *et al.*, 2015) backend.

DATA PREPARATION

KiK-net data

Although the KiK-net website provides the option to search for earthquake records based on different parameters, we found that the interface was not suitable for selecting and downloading the relatively large amount of records that we wanted to use for this project. Instead, we developed a script that downloads all of the earthquake records from the KiK-net website and stores them locally. The acceleration time series from all earthquakes recorded by KiK-net stations between January 1997 and August 2020 were downloaded from the Kyoshin website (see [Data and Resources](#)). Five parallel download sessions were executed to retrieve the records, which amounted to about 105 GB in total. A single event file was provided for each recorded earthquake. Event files consist of tar archives with acceleration time series in K-NET ASCII format for all KiK-net sites which recorded the event. Earthquake and station metadata were extracted from event files and stored in a local database. Three separate tables with station information (i.e., station code, location, elevation, and sensor depth), event information (event ID, date, magnitude, and hypocenter), and record information (event ID, station code, peak ground acceleration [PGA], and distance) were generated and stored in Python Pandas dataframes.

In this article, we focus on predicting the mean weak-motion amplification at a site. The prediction of event-specific amplification, including potential nonlinear or wave propagation effects, is left for future work.

To rule out a significant impact of nonlinearity on mean AFs, we only used records with surface PGAs below 200 cm/s^2 . We note that it is generally difficult to exclude nonlinearity based on a surface acceleration threshold because the onset nonlinearity is controlled by the level of strain. We therefore experimented with different thresholds and found that a value of 200 cm/s^2 does not result in altered AFs compared with threshold values of 50 and 100 cm/s^2 . The maximum PGA threshold of 200 cm/s^2 was selected to strike a balance between removing unwanted nonlinear responses and keeping large or nearby events with good signal-to-noise ratios.

Where available, we randomly picked 20 events with PGAs within 50 to 200 cm/s^2 . If less than 20 events with $50 \text{ cm/s}^2 < \text{PGA} < 200 \text{ cm/s}^2$ were available, we selected the 20 events with the highest PGA. The number of 20 events per site was chosen because all except four sites (KNMH18, FKOHO2, SOYH3, and AICH23) recorded more than 20 events until August 2020, and no site recorded less than 10 events.

The next step in the KiK-net data processing workflow consisted of the extraction of time-series data from the selected observations. Surface-to-borehole transfer functions were computed for both horizontal components and smoothed using a Konno–Ohmachi filter (bandwidth $w = 10$), and the geometric mean of both horizontals was computed. We then interpolated the amplification at 50 frequencies of interest,

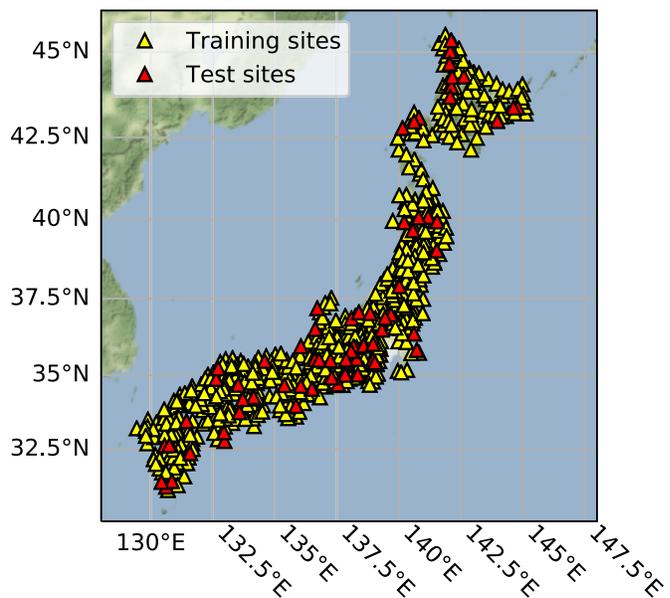


Figure 3. Locations of KiK-net sites assigned to training and test sets in this study. The color version of this figure is available only in the electronic edition.

which are logarithmically spaced between 0.3 and 20 Hz. This procedure was carried out for a total of 13,210 events. Computationally, the data preparation was expensive because two fast Fourier transforms and Konno–Ohmachi filtering operations were executed for each record. To accelerate the process, the web scraper and data processing workflow were deployed on the commodity cluster Rhea at the Oak Ridge Leadership Computation Facility (OLCF). We used the Apache Spark Engine to distribute the data processing on up to five nodes and 80 central processing unit cores. This approach resulted in a wall-clock time of less than two hours for the computation and smoothing of the AFs for all 13,210 records.

Training and test datasets were created as follows: first, we randomly selected 90% of the sites to contribute to the training set, and the remaining sites were assigned to the test set. Figure 3 shows the distribution of training and test sites among the KiK-net stations. We used the same selection of training and test sites for all different NN layouts and hyperparameter choices shown in this article, to allow for a one-by-one comparison of network performances. We created training and test datasets by iterating over all records pertaining to each given training and test site. In the many-to-many layout used in the CNN for the prediction of mean AFs, the training and test sets contained just one data point per site. In the many-to-one design of the fully connected network, one data point for training or testing was created for each site and frequency for the prediction of mean amplification. The number of datapoints per site equals the number of events at the site times the number of frequencies in the prediction of event-specific amplifications using the many-to-one NN layout.

The training and test sets for the prediction of event-specific amplifications using the many-to-one NN layout contain one data point for each frequency and each observation per site. The many-to-many design in the CNN requires just one data point per observation and site in the training and test sets.

NN TRAINING AND PREDICTION RESULTS

We trained the fully connected NN and convolutional NN with the mean AFs for 596 KiK-net sites assigned to the test set. Mini-batch gradient descent using the Adam optimizer (Kingma and Ba, 2014) was carried out to minimize the mean squared logarithmic error (MSLE) between observed and predicted theoretical AFs. We chose the MSLE as our loss function to incorporate the large range of amplifications observed between different sites and frequencies. A low MSLE is consistent with good visual agreements if AFs are plotted in logarithmic space as is conventionally done. We also report the mean absolute error (MAE) between predicted and observed AFs. The batch size was set to 2048 for the MLP and to 50 for the CNN. We trained the ANNs for 1000 epochs using the default learning rate of 10^{-3} .

Prediction of mean AFs using the MLP

Figure 4 shows the learning curve obtained during training of the fully connected neural network (MLP) with mean AFs. The training loss (MSLE) is reduced from an initial value of 1.36 to 0.011 (Table 1). Input features (i.e., V_S and frequency of amplification) were standardized by removing the mean and scaling to unit variance before training. To control the amount of overfitting to the training data, we used dropout regularization in the first five hidden layers. The dropout rate was adjusted to a value of 0.15 by trial and error. Lower values resulted in a higher validation error, whereas higher values increased the training error without further reducing the validation error. Figure 5 compares observed and predicted mean AFs for nine randomly selected training sites. The low training error is reflected in the good match between observed and predicted mean AFs.

We used the trained MLP to predict mean site amplifications at the 66 test sites and obtained an MSLE of 0.128 (Table 1). Figure 6 compares the observed and predicted mean amplifications at nine randomly selected test sites. The MSLE at the displayed test sites ranges from about 0.040 for sites at which the predicted AF is close to the observations (e.g., ISKH04, MIEH06, and KOCH13) to values above 0.150 for sites at which the predicted AF does not reproduce the observation well (e.g., YMNH14, SRCH01, and YMTH07). However, predicting the site response from a soil profile is generally difficult due to multidimensional effects, modeling inaccuracies, uncertainties in soil property estimates, and effects of the downgoing wave (e.g., Bonilla *et al.*, 2002; Thompson *et al.*, 2009, 2012; Zhu *et al.*, 2020).

Although the test MSLE provides a quantitative estimate of the residuals between predictions and observations, the

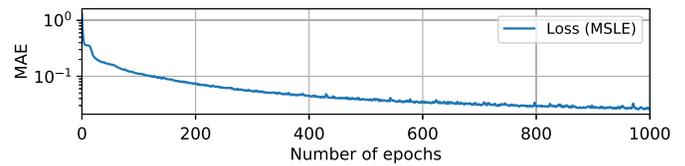


Figure 4. Learning curve with mean absolute training and validation errors during Adam optimization of the fully connected neural network (multilayer perceptron [MLP]) for prediction of mean site response. The color version of this figure is available only in the electronic edition.

significance of this error cannot be assessed without comparison with a baseline solution. In ML, simple statistical models (such as a linear regression), human judgement (e.g., Ng, 2018), or published state-of-the-art models (e.g., Wang, Walters, and Yu, 2020; Wang, Kashinath, *et al.*, 2020) are typically used as a baseline. Regardless of the absolute value of the test error, an ML model is considered useful if it outperforms a reasonable or well-established baseline. For the presented case of predicting surface-to-borehole AFs from the full soil profile, several alternate solutions may be considered a baseline, for example, empirical site amplification models based on the V_{S30} (e.g., Seyhan and Stewart, 2014) or the theoretical 1D SH AF (SH1D). Alternatively, a custom baseline may be derived by fitting a shallow ML model to our training data.

Here, we use the theoretical SH1D AF for a vertically incident plane wave as a baseline. The SH1D amplification represents an obvious choice for the baseline because it is computed from the same input features used to train the MLP (the full vertical shear-wave velocity profile). Moreover, the SH1D method allows for computation of the surface-to-borehole AF that we want to predict, whereas empirical models typically predict the amplification with respect to a reference site. Therefore, the SH1D amplification represents the tool that engineering seismologists would likely pick to solve the given problem.

Of course, more sophisticated baselines could be constructed, for example, by accounting for multidimensional effects using 3D wave propagation simulations (e.g., Thompson *et al.*, 2009; Hu *et al.*, 2021); however, such methods rely on additional information that is not available for most KiK-net sites.

We computed the SH1D amplification assuming a horizontally layered structure and a vertically incident plane wave. Densities and quality factors were derived from the shear-wave velocity profiles using an empirical relation (Brocher, 2006). Theoretical AFs were smoothed in the same way as observed mean AFs. The MSLE between theoretical and observed AFs is listed for each site in Figure 6 (as well as Fig. 5 for reference, although we note that it makes little sense to compare training losses with theoretical predictions). With the exceptions of sites GIFH25 and SRCH01, the theoretical model results in a larger prediction error than the NN.

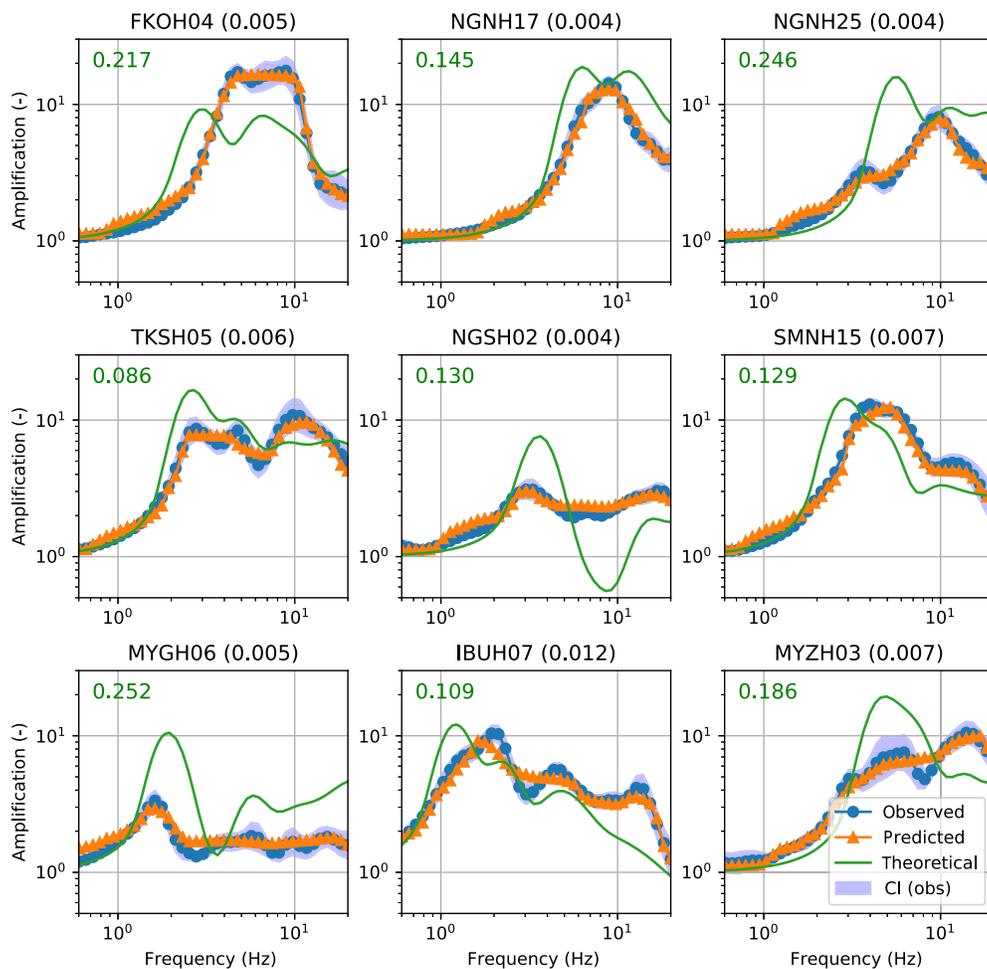


Figure 5. Comparison between observed mean AFs (blue) and AFs predicted by the MLP (orange) for nine randomly selected training sites. Solid green lines show theoretical 1D site AFs. Numbers in brackets next to the site name give the training loss (mean squared logarithmic error [MSLE]) for the site. Green numbers in the upper left corner show the baseline loss (based on the theoretical SH1D AF) for the site. The color version of this figure is available only in the electronic edition.

The MSLE of 0.22 between theoretical and observed AFs for the 66 test sites (Table 1) is used as a baseline to assess the NN's performance, and it shows that the NN is generally predicting mean AFs more accurately than the theoretical model.

TABLE 1
Errors between Observed and Predicted Amplifications*

Model	MSLE (Train)	MAE (Train)	MSLE (Test)	MAE (Test)
Baseline			0.216	2.005
MLP	0.011	0.451	0.128	1.543
CNN	0.021	0.595	0.104	1.307

*Baseline, theoretical SH1D amplification; CNN, convolutional neural network; MAE, mean absolute error; MLP, multilayer perceptron; MSLE, mean squared logarithmic error (loss).

Prediction of mean AFs using the CNN

The convolutional CNN was trained using observed mean site amplifications for the same 596 training sites as for the MLP. As with the MLP, we adjusted dropout rates for the CNN by trial and error to minimize the trade-off between high-model bias (in case of poor performance for training sites) and high variance (in case of overfitting). For the CNN, we tuned dropout rates to different values for each layer. A dropout rate of 0.5 was used after the pooling layer (Fig. 2), and dropout rates of 0.3, 0.15, and 0.10 were used for the three subsequent, fully connected hidden layers, respectively.

The CNN was trained for 2000 epochs using a batch size of 50 sites. The loss was reduced from an initial value of 1.45 to a minimum of 0.043. We used the trained model to predict AFs for training and test sites and obtained MSLE values of 0.021 and 0.104, respectively (Table 1). The MSLE for predicted AFs at training sites (0.021, Table 1) is lower than the lowest minimum value obtained during optimization (0.043). This discrepancy is caused by dropout regularization.

Dropout regularization randomly eliminates neurons during training, resulting in a relatively high training error. However, all neurons are enabled during prediction (dropout rate is set to zero), which results in a lower prediction error than training error.

Both MSLEs and MAEs at test sites are lower for the CNN than for the MLP. Moreover, the CNN achieves a test MSLE that is 50% lower than the baseline (Table 1). Figure 7 compares observed, predicted, and theoretical (baseline) AFs for the same nine test sites as shown for the MLP (Fig. 6). The CNN results in more accurate predictions, especially for sites YMNH14 (MSLE reduction from 0.146 to 0.028), MIEH08 (0.043–0.020), and GIFH25 (0.064–0.030).

A notable difference between the observations and the SH1D baseline concerns the peak at the fundamental frequency, in which the baseline frequently predicts much higher

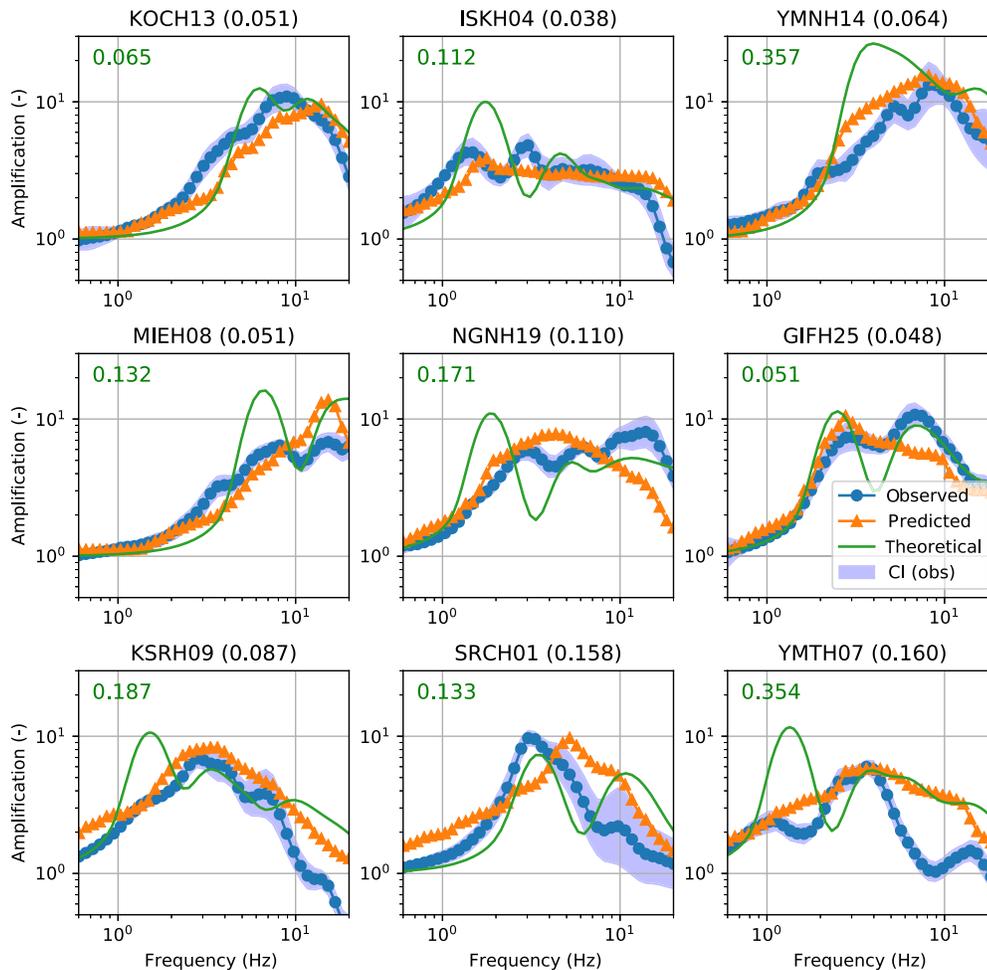


Figure 6. The same as Figure 5 but showing mean and predicted mean amplifications for nine randomly selected test sites. The color version of this figure is available only in the electronic edition.

amplitudes than observed, for example, at sites IKSH04, MIEH08, NGNH19, KSRH09, and YMTH07 (Figs. 6 and 7). This discrepancy between theoretical and observed surface-to-borehole amplification has been discussed in previous studies and attributed to the effects of the downgoing wave, lateral heterogeneities, or the presence of strong velocity contrasts below the downhole site (e.g., Bonilla, 2001; Thompson *et al.*, 2009; Zhu *et al.*, 2020; Hu *et al.*, 2021). Because the NNs are not bound by any theoretical assumptions, they are less affected by this problem and predict the peak of the amplifications based on the training data.

Figure 8a shows the distribution of the MSLE obtained by the baseline, MLP, and CNN at the 66 test sites. The CNN clearly represents an improvement with respect to both the MLP and baseline. For example, the CNN distribution peaks at an MSLE of 0.05, with a median MSLE of 0.073. Baseline (SH1D) errors are more uniformly distributed with a median MSLE of 0.181. Compared with the MLP, the CNN achieves an MSLE below 0.075 for more sites and errors above 0.225 for

fewer sites. We also computed the change in MSLE between the two NN designs and the baseline for each site and plotted the distribution of the change of error (Fig. 8b). With respect to the baseline, the NNs results in an improvement at most sites; the CNN also outperforms the MLP in terms of number of sites at which an improvement is observed.

We calculated the geometric mean of the test MSLE over all 66 test sites at each of the 50 discrete frequencies (Fig. 9). Prediction errors for the MLP and CNN tend to be highest at frequencies between 5 and 10 Hz, for which the geometric mean of the error is ~ 10 times larger than at 1 Hz. Baseline errors are highest between 2 and 10 Hz; they are above the ANN errors throughout the frequency band, but are below the MLP for frequencies below 0.5 Hz (Fig. 9a). In particular, the CNN performs better than the baseline in the frequency range between 2 and 10 Hz (Fig. 9b).

SUMMARY AND OUTLOOK

We have calculated mean observed AFs for 662 KiK-net vertical arrays. 90% of the sites were assigned as training sites, with the remaining 10% withheld as test sites. An MLP and a CNN were trained to predict the observed AFs from a discretized representation of the velocity profiles.

Both NN designs converged to a solution with minimal loss and accurately reproduced the observed AFs at the training sites. Although the quality of the prediction at the test sites varied, both the MLP and the CNN outperformed predictions based on the theoretical SH1D site response in terms of MSLE between observed and predicted AFs. Predictions made by the CNN resulted in an MSLE that was 50% lower than the SH1D baseline and 25% lower than the predictions by the MLP. Proper regularization and fine-tuning of the dropout rate was found to be essential to obtaining good predictions at test sites not used for training.

These results show that artificial NNs have the potential to take advantage of the full velocity profile information for more

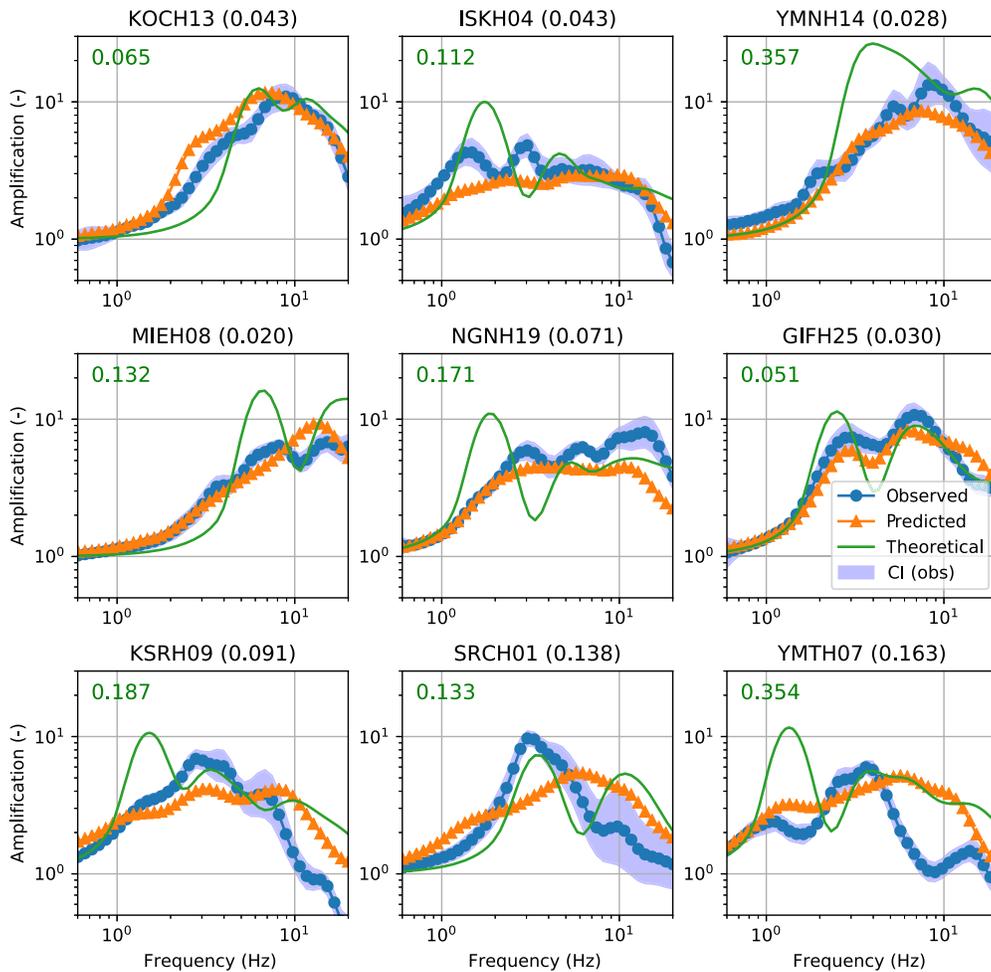


Figure 7. The same as Figure 6 but showing predictions by the CNN at the same test sites. The color version of this figure is available only in the electronic edition.

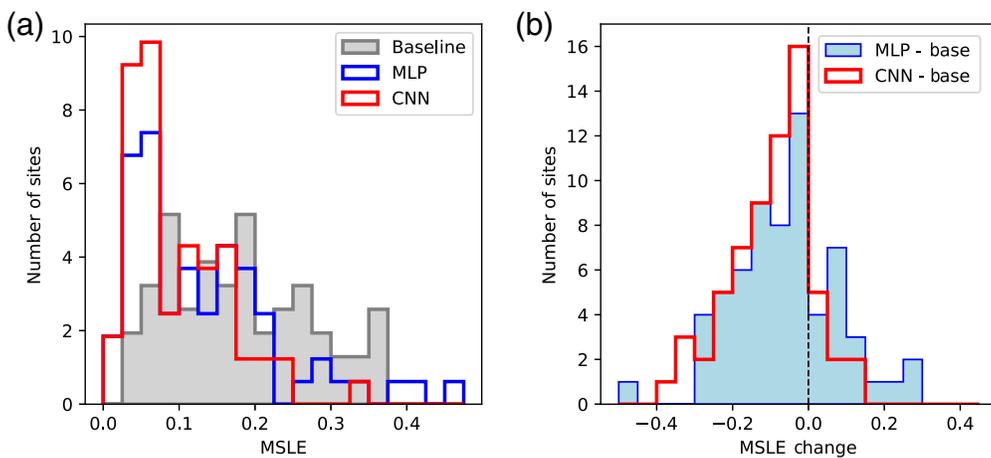


Figure 8. (a) Distribution of site-specific test MSLE obtained from baseline, MLP, and CNN results. (b) Distribution of change in MSLE achieved using MLP and CNN with respect to the baseline (Base). The color version of this figure is available only in the electronic edition.

accurate predictions of observed AFs. Although a simple SH1D AF does not represent the state-of-the-art for site-specific seismic-hazard analysis in engineering seismology, it serves as a useful benchmark that demonstrates the limits of commonly made assumptions (in particular, a horizontal 1D layered structure and a vertically incident plane wave). An NN that learns to predict AFs purely from data is not bound by such assumptions, and we have demonstrated the level of improved accuracy with respect to the baseline that can be learned from data.

In future work, we will also evaluate the accuracy of the NN-predicted AFs against empirical site AFs, which are typically based on the V_{S30} . Although our efforts in this article were focused on the prediction of mean AFs, future work should address the prediction of event-specific amplification. Here, the effects of incident wavefield characteristics, scattering, and nonlinearity would be captured by feeding the network with earthquake magnitude, hypocentral distance, and input signal metrics (e.g., PGA, spectral accelerations at different frequencies, and duration). In the case of a CNN, this requires a slightly more complicated design with mixed data inputs. We also recommend for future work exploring the use of information on the multidimensionality of a site structure.

DATA AND RESOURCES

Acceleration time series from KiK-net strong-motion stations (National Research Institute for Earth Science and Disaster Resilience [NIED], 2019) were

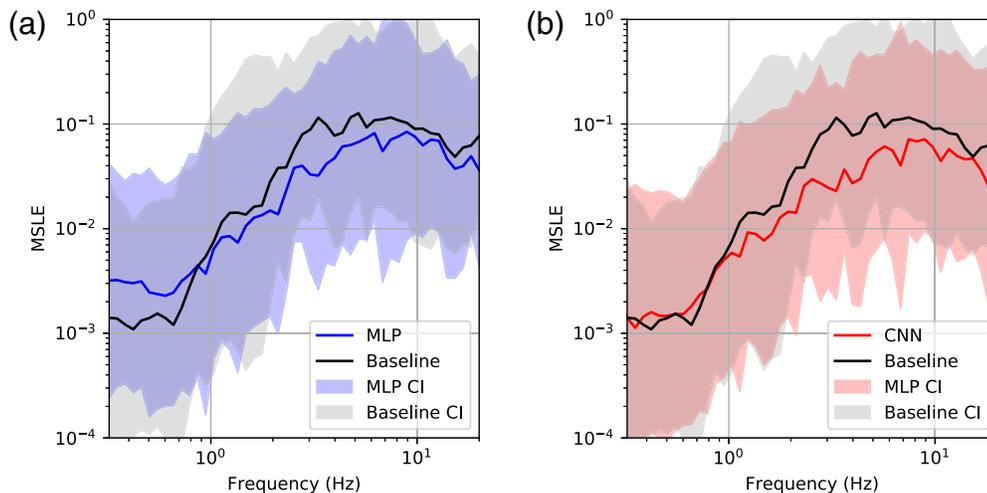


Figure 9. Geometric mean of MSLE as a function of frequency for all test sites. (a) MLP versus baseline. (b) CNN versus baseline. CI, confidence interval. The color version of this figure is available only in the electronic edition.

downloaded from <https://www.kyoshin.bosai.go.jp/>. The data scraper for KiK-net records was programmed in Python using the Requests library <https://requests.readthedocs.io/en/master/>. The Apache Spark Unified Analytics Engine for Big Data is available at <https://spark.apache.org/>. KiK-net datafiles were read using the ObsPy Python Framework for Seismology (Beyreuther *et al.*, 2010, docs.obspy.org). The Pandas Data Analysis Library for Python is available at <https://pandas.pydata.org/>. The Keras deep learning library (Chollet, 2018) can be obtained at <https://keras.io>. We used the Tensorflow backend for Keras (<https://tensorflow.org>). The StandardScaler from the scikit-learn library (Pedregosa *et al.*, 2011) was used to normalize input data for the multi-layer perceptron (MLP) (<https://scikit-learn.org/>). All websites were last accessed in October 2020.

DECLARATION OF COMPETING INTERESTS

The authors declare no competing interests.

ACKNOWLEDGMENTS

This project is supported by the California Department of Conservation, California Geological Survey, Strong Motion Instrumentation Program, Contract Number 1019-014. The authors are grateful to the National Research Institute for Earth Science and Disaster Resilience (NIED) for providing vertical array strong-motion data. Data preprocessing were carried out on the Oak Ridge Leadership Computational Facility (OLCF) cluster Rhea under an Innovative and Novel Computational Impact on Theory and Experiment (INCITE) research allocation. The authors thank the guest editors and two anonymous reviewers for their comments, which helped to improve the article.

REFERENCES

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.* (2015). TensorFlow: Large-scale machine learning on heterogeneous systems, available at <https://tensorflow.org> (last accessed March 2021).

Abrahamson, N. A., W. J. Silva, and R. Kamai (2014). Summary of the ASK14 ground motion relation for active crustal regions, *Earthq. Spectra* **30**, no. 3, 1025–1055.

Beyreuther, M., R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann (2010). ObsPy: A Python toolbox for seismology, *Seismol. Res. Lett.* **81**, no. 3, 530–533.

Bommer, J. J., and N. A. Abrahamson (2006). Why do modern probabilistic seismic-hazard analyses often lead to increased hazard estimates? *Bull. Seismol. Soc. Am.* **96**, no. 6, 1967–1977.

Bonilla, L. F. (2001). *NOAH: User's Manual*, Institute for Crustal Studies, University of California, Santa Barbara, California.

Bonilla, L. F., C. Gelis, and J. Regnier (2011). The challenge of non-linear site response: Field data observations and numerical simulations, *4th IASPEI/IAEE International Symposium: Effects of Surface Geology on Seismic Motion*, 23–26 August 2011, University of Santa Barbara, Santa Barbara, California.

Bonilla, L. F., J. H. Steidl, J.-C. Gariel, and R. J. Archuleta (2002). Borehole response studies at the Garner Valley downhole array, southern California, *Bull. Seismol. Soc. Am.* **92**, no. 8, 3165–3179.

Brocher, T. (2006). Key elements of regional seismic velocity models for ground motion simulations, *Proc. of Int. Workshop on Long-Period Ground Motion Simulations and Velocity Structures*, 14–15 November 2006, Tokyo, Japan.

Chollet, F. (2018). Deep learning with Python and Keras: The practical guide from the developer of the Keras library, MITP-Verlags GmbH & Co. KG, Bonn, Germany, 361 pp.

Derras, B., P.-Y. Bard, and F. Cotton (2014). Towards fully data driven ground-motion prediction models for Europe, *Bull. Seismol. Soc. Am.* **12**, no. 1, 495–516.

Derras, B., P.-Y. Bard, F. Cotton, and A. Bekkouche (2012). Adapting the neural network approach to PGA prediction: An example based on the KiK-net data, *Bull. Seismol. Soc. Am.* **102**, no. 4, 1446–1461.

DeVries, P. M. R., T. B. Thompson, and B. J. Meade (2017). Enabling large-scale viscoelastic calculations via neural network acceleration, *Geophys. Res. Lett.* **44**, no. 6, 2662–2669.

Douglas, J. (2003). Earthquake ground motion estimation using strong-motion records: A review of equations for the estimation of peak ground acceleration and response spectral ordinates, *Earth Sci. Rev.* **61**, nos. 1/2, 43–104.

Gatti, F., F. Lopez-Caballero, D. Clouteau, and R. Paolucci (2018). On the effect of the 3-D regional geology on the seismic design of critical structures: The case of the Kashiwazaki-Kariwa Nuclear power plant, *Geophys. J. Int.* **213**, no. 2, 1073–1092.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*, Vol. 1, MIT Press, Cambridge, Massachusetts, 777 pp.

- Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Science & Business Media, Berlin, Germany, 764 pp.
- Hu, Z., D. Roten, K. B. Olsen, and S. M. Day (2020). Modeling of empirical transfer functions with 3D velocity structure, *Bull. Seismol. Soc. Am.*, doi: [10.1785/0120200214](https://doi.org/10.1785/0120200214).
- Hu, Z., D. Roten, K. B. Olsen, and S. M. Day (2021). Modeling of empirical transfer functions with 3D velocity structure, *Bull. Seismol. Soc. Am.* doi: [10.1785/0120200214](https://doi.org/10.1785/0120200214).
- Kingma, D. P., and J. Ba (2014). Adam: A method for stochastic optimization, available at <http://arxiv.org/abs/1412.6980> (last accessed March 2020).
- National Research Institute for Earth Science and Disaster Resilience (NIED) (2019). National Research Institute for Earth Science and Disaster Resilience K-NET, KiK-net, doi: [10.17598/NIED.0004](https://doi.org/10.17598/NIED.0004).
- Ng, A. (2018). Machine learning yearning, available at <https://www.deeplearning.ai/programs/> (last accessed March 2020).
- Nolet, G. (2008). *A Breviary of Seismic Tomography*, Cambridge University Press, Cambridge, United Kingdom, 360 pp.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. (2011). Scikit-learn: Machine learning in Python, *J. Mach. Learn. Res.* **12**, 2825–2830.
- Roten, D., D. Fäh, and L. F. Bonilla (2013). High-frequency ground motion amplification during the 2011 Tohoku earthquake explained by soil dilatancy, *Geophys. J. Int.* **193**, no. 2, 898–904.
- Roten, D., D. Fäh, and L. F. Bonilla (2014). Quantification of cyclic mobility parameters in liquefiable soils from inversion of vertical array records, *Bull. Seismol. Soc. Am.* **104**, no. 6, 3115–3138.
- Seyhan, E., and J. P. Stewart (2014). Semi-empirical nonlinear site amplification from NGA-West2 data and simulations, *Earthq. Spectra* **30**, no. 3, 1241–1256.
- Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: A simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.* **15**, no. 1, 1929–1958.
- Strasser, F. O., N. A. Abrahamson, and J. J. Bommer (2009). Sigma: Issues, insights, and challenges, *Seismol. Res. Lett.* **80**, no. 1, 40–56.
- Thompson, E. M., L. G. Baise, R. E. Kayen, and B. B. Guzina (2009). Impediments to predicting site response: Seismic property estimation and modeling simplifications, *Bull. Seismol. Soc. Am.* **99**, no. 5, 2927–2949.
- Thompson, E. M., L. G. Baise, Y. Tanaka, and R. E. Kayen (2012). A taxonomy of site response complexity, *Soil Dynam. Earthq. Eng.* **41**, 32–43.
- Thompson, E. M., Y. Tanaka, L. G. Baise, and R. E. Kayen (2010). Three-dimensional site response at KiK-net downhole arrays, *7th International Conf. on Urban Earthquake Engineering & 5th International Conf. on Earthquake Engineering*, 3–5.
- Wang, R., K. Kashinath, M. Mustafa, A. Albert, and R. Yu (2020). Towards physics-informed deep learning for turbulent flow prediction, *Proc. of the 26th ACM SIGKDD International Conf. on Knowledge Discovery & Data Mining*, 1457–1466.
- Wang, R., R. Walters, and R. Yu (2020). Incorporating symmetry into deep dynamics models for improved generalization, available at <http://arxiv.org/abs/2002.03061> (last accessed March 2020).
- Withers, K. B., M. P. Moschetti, and E. M. Thompson (2020). A machine learning approach to developing ground motion models from simulated ground motions, *Geophys. Res. Lett.* **47**, no. 6, doi: [10.1029/2019GL086690](https://doi.org/10.1029/2019GL086690).
- Zhu, C., M. Pilz, and F. Cotton (2020). Evaluation of a novel application of earthquake HVSR in site-specific amplification estimation, *Soil Dynam. Earthq. Eng.* **139**, 106301, doi: [10.1016/j.soildyn.2020.106301](https://doi.org/10.1016/j.soildyn.2020.106301).

Manuscript received 31 October 2020

Published online 23 March 2021