Geophysical Journal International

Geophys. J. Int. (2022) **229,** 704–718
Advance Access publication 2021 December 01
GJI Seismology

https://doi.org/10.1093/gji/ggab488

Transfer learning: improving neural network based prediction of earthquake ground shaking for an area with insufficient training data

Dario Jozinović [1,2] Anthony Lomax,[3] Ivan Štajduhar[4] and Alberto Michelini[1]

[1]*Istituto Nazionale di Geofisica e Vulcanologia, Via di Vigna Murata* 605, 00143 *Rome, Italy. E-mail: djozinovi@gmail.com*
[2]*Department of Science, Università degli Studi Roma Tre, Via Ostiense* 159, 00154 *Rome, Italy*
[3]*ALomax Scientific,* 320 *Chemin des Indes,* 06370 *Mouans-Sartoux, France*
[4]*Department of Computer Engineering, Faculty of Engineering, University of Rijeka,* 51000 *Rijeka, Croatia*

Accepted 2021 November 29. Received 2021 November 25; in original form 2021 May 12

SUMMARY

In a recent study, we showed that convolutional neural networks (CNNs) applied to network seismic traces can be used for rapid prediction of earthquake peak ground motion intensity measures (IMs) at distant stations using only recordings from stations near the epicentre. The predictions are made without any previous knowledge concerning the earthquake location and magnitude. This approach differs significantly from the standard procedure adopted by earthquake early warning systems that rely on location and magnitude information. In the previous study, we used 10 s, raw, multistation (39 stations) waveforms for the 2016 earthquake sequence in central Italy for 915 $M \geq 3.0$ events (CI data set). The CI data set has a large number of spatially concentrated earthquakes and a dense network of stations. In this work, we applied the same CNN model to an area of central western Italy. In our initial application of the technique, we used a data set consisting of 266 $M \geq 3.0$ earthquakes recorded by 39 stations. We found that the CNN model trained using this smaller-sized data set performed worse compared to the results presented in the previously published study. To counter the lack of data, we explored the adoption of 'transfer learning' (TL) methodologies using two approaches: first, by using a pre-trained model built on the CI data set and, next, by using a pre-trained model built on a different (seismological) problem that has a larger data set available for training. We show that the use of TL improves the results in terms of outliers, bias and variability of the residuals between predicted and true IM values. We also demonstrate that adding knowledge of station relative positions as an additional layer in the neural network improves the results. The improvements achieved through the experiments were demonstrated by the reduction of the number of outliers by 5 per cent, the residuals R median by 39 per cent and their standard deviation by 11 per cent.

Key words: Europe; Waveform inversion; Neural networks, fuzzy logic; Time-series analysis; Earthquake early warning; Earthquake ground motions.

1 INTRODUCTION

Having information about earthquake generated ground motions in the shortest time possible (Minson *et al*. 2018) is of great importance for earthquake monitoring. For a timescale of 5–10 min after the earthquake, the ShakeMap software (Wald *et al*. 1999) provides maps of ground shaking [in terms of peak ground acceleration (PGA), peak ground velocity (PGV) and spectral acceleration (SA) at 0.3, 1 and 3 s periods and macroseismic intensity], which can be used by disaster risk managers for rapid assessment of the earthquake impact. On a shorter timescale (few seconds), earthquake early warning systems (EEWSs) have been developed in the seismological community (see the reviews of Satriano *et al*. 2011; Cremen & Galasso 2020). These systems, developed as regional (e.g. Kohler *et al*. 2018) or on-site EEWS (e.g. Spallarossa *et al*. 2019), seek to detect and characterize earthquakes rapidly and provide warnings to points, or areas, not yet impacted by ground shaking.

In a recent study, Jozinović *et al*. (2020, J2020, hereinafter) used a machine learning (ML) approach to predict the ground shaking intensity at a pre-defined set of seismic stations within a given seismic area, as quickly as possible. The inputs to the ML model were 7, 10 or 15 s long waveforms (the length of the waveform window was investigated within the study) from a pre-defined and fixed set of seismic stations with all traces starting at the earthquake origin time for simplicity. The outputs of the ML model were the intensity

measures (IMs; i.e. PGA, PGV and SA at 0.3, 1 and 3 s periods) on the selected stations. This configuration entails that the strongest shaking would be recorded on the traces of the stations closest to the epicentre, while the model would give predictions for the stations farther from the epicentre. Clearly, the balance between the recorded and predicted IMs will depend upon the source–receiver acquisition geometry and the selected traces' window length. One notable feature of the model is that it does not require any information about the earthquake parameters (magnitude, location, etc.) and uses just the multistation waveforms pattern to give the predictions. The best compromise between the accuracy and timeliness was found when 10 s windows were used. In J2020, it was also found that the IMs prediction accuracy was similar to the accuracy attainable using the ground motion prediction equations (GMPEs) by Bindi *et al*. (2011) which require, however, an earthquake location and magnitude as input. In addition, it was found that the ML model was able to predict with useful accuracy the IMs at the stations which had no input data available (and were replaced with a window of zeros). In this study, we use the model developed in J2020 and apply it to the area of central-western Italy.

Increasing usage of ML in seismology has led to the development of ML based rapid earthquake characterization algorithms and rapid peak ground motion prediction algorithms. Some of the developed algorithms deal with rapid seismic wave discrimination (e.g. Li *et al*. 2018) or rapid earthquake characterization (e.g. Böse *et al*. 2012; Hsu *et al*. 2016; Ochoa *et al*. 2018; Saad *et al*. 2020; van den Ende & Ampuero 2020; Münchmeyer *et al*. 2021; Zhang *et al*. 2021). Böse *et al*. (2008) have approached the problem of rapid earthquake characterization for EEWS using multistation waveforms to extract a series of chosen parameters and use them as inputs for a feedforward neural network. Kong *et al*. (2016) developed an ML algorithm that detects earthquakes on smartphone accelerometers and uses the information from the triggered smartphones to estimate the earthquake location. Otake *et al*. (2020) used a recurrent neural network that adopted waveforms from 4 stations as input, to predict the shaking intensity at one target location. The study of Münchmeyer *et al*. (2020) deals with the EEWS problem with a technique called TEAM, which takes multistation seismic waveforms as input and predicts the PGA, which is similar to the technique developed in J2020 and used in this paper. The main difference between the two algorithms derives from the use of different neural network model types. TEAM uses a combination of a transformer and a convolutional neural network (CNN) whereas in J2020 we rely exclusively on a CNN. The advantage of using TEAM comes from the possibility to use any set of up to 25 stations as the input to the model, which provides flexibility in applying the network to different areas without the need to retrain the model. In contrast, the J2020 approach always uses the same, structured, set of stations as the input to the model. The advantage of this approach is that the CNN learns the specifics of ground motion at every station (the number of stations and their order is fixed—a specific station is always at the exact same place in the input waveform) and the whole pattern amongst the stations but it needs to be retrained in order to be applied to a different area with a different set of stations. This would not be a limit for the J2020 model for application to a dense set of stations, like those in J2020, with a large enough training data set. However, training it for other areas with a sparse network of stations that has a smaller-sized data set of earthquake waveforms available for training, would likely lead to poorer results compared to J2020. In this study, we try to overcome this problem by testing the use of transfer learning (TL) for model training (Bozinovski 2020; Pan & Yang 2009 for a review), which is further described in Section 3.1.

TL uses two ML models in the following way: the first, already trained, model is used for the initialization of the weights of the second model. More precisely, TL consists of taking a model pre-trained on a source data set (usually larger) or a source task (e.g. single-station magnitude determination) and using it (or its parts) for training a model on a target data set or for a target task (e.g. multistation IM prediction), where the source and target data sets, or tasks, are sampled from different underlying distributions. In doing this, it is expected that the first distribution is similar enough to the second, most importantly in the input data sets, so that its use for TL will likely improve model performance compared to a model having its weights randomly initialized. Note that TL is effectively the replication of the natural mental processes that occur in all species of exploiting previous knowledge for new learning.

There has been some application of TL to seismology (e.g. Titos *et al*. 2019; Chai *et al*. 2020; Johnson *et al*. 2021; Otović *et al*. 2021) although its use is not yet widespread for applications of ML in seismology. In the TEAM algorithm, Münchmeyer *et al*. (2020) explored the use of TL to improve the PGA prediction for large events. In this study, we explore the use of TL for improving our IM prediction algorithm. We use the pre-trained model from the J2020 study, with the same source task as the task used in this study (which means that we use the same model architecture as the one in J2020), but with a different source data set. We also explore TL from a different source task (magnitude determination from single station waveforms), trained on a different source data set.

2 DATA

The input data from central western Italy (hereinafter denominated the CW data set) consist of three-component waveforms of 256 earthquakes with magnitude $2.9 \leq M \leq 5.1$ (Fig. 1a), from 39 stations. The earthquakes occurred between 2013 January 1 and 2017 November 20. The earthquake depths range from 3.3 to 64.7 km (Fig. 1b). The stations and the earthquakes are located in the area bounded by latitude [41.13°, 46.13°] and longitude [8.5°, 13.1°] (Fig. 2), with epicentral distances ranging from 10 to 498 km (Fig. 1c). The area overlaps slightly with the area of central Italy used in J2020 from which the pre-trained model architecture is taken. To avoid possible data leakage from the pre-trained model, events that were used in that study were excluded from our data set. We chose the stations having the largest number of events recorded while making sure that there is an acceptable spatial distribution of the stations to cover more area with stations that are close to earthquake epicentres to simulate possible early warning use (more details about the stations in Supporting Information Table S1). The stations belong to the IV, GU and MN networks.

The three component, 3C, waveform data were downloaded using the INGV FDSN web services for HN∗ (acceleration) and HH∗ and EH∗ (velocity) channels, where $* \in$ [E, N, Z]. The data were processed to remove the instrument response, velocity data were differentiated to acceleration and, if necessary, the data were resampled to 100 Hz. For $M < 4$ earthquakes, the HH and EH channels were used after differentiation and for earthquakes with $M \geq 4.0$ the HN channels were used. For certain stations and for some earthquakes, the waveform data were completely missing and we chose to replace them with zeros, as in J2020. This data selection and processing follows the criteria outlined in J2020.

The *target variables* consisted of the IMs associated with each recording: PGA, PGV and SA at 0.3, 1 and 3 s periods, which

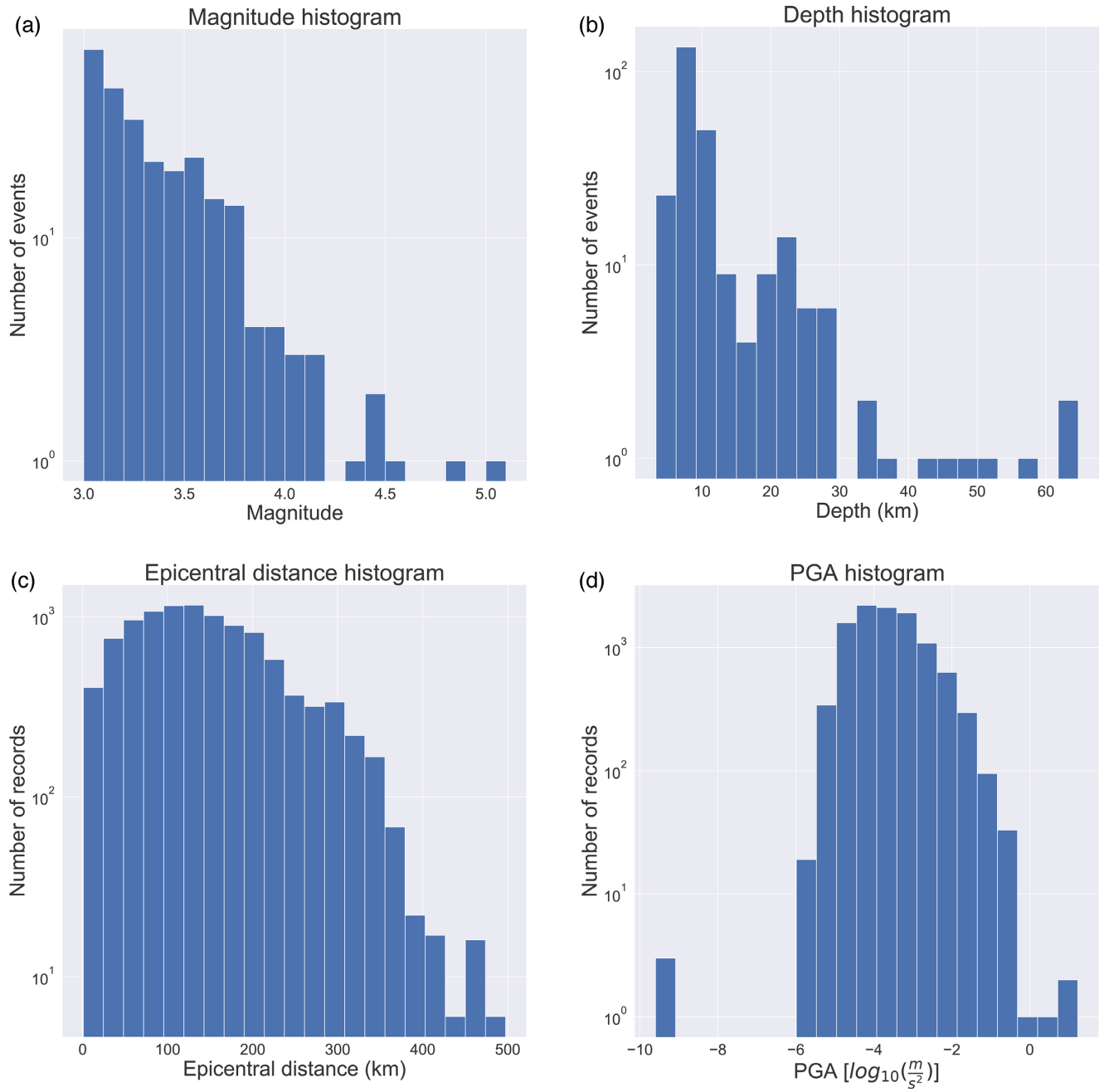

Figure 1. Histograms of the selected 256 earthquakes: (a) earthquake-magnitude distribution, (b) earthquake-depth distribution, (c) epicentral distance distribution and (d) distribution of PGA [$\log_{10}$(m s^{-2})]

were extracted from windows that started 5 s before the predicted *P*-arrival and ended 10 s after the predicted *S*-arrival (please note that this window size is not related to the window size of the CNN input used in the experiments and mentioned in the later chapters of the study). For stations that had no data, the IMs were calculated using the USGS ShakeMap software using the latest configuration for Italy (Michelini *et al*. 2020) to ensure no missing output data (target variables). Since we are using a fixed number of stations for the input, we must always provide an array of fixed size (*cf.* J2020). However, sometimes input data is missing, in this case we use an array of zeros to fill the missing data (Garcia-Laencina *et al*. 2010), which is a natural way to employ the station dropout technique (Kriegerowski *et al*. 2019). We have a fixed set of outputs of the model for which we need to provide training and validation target data. Although using the ShakeMap approximations introduces further assumptions into the model, we consider it the best way to fill the missing data because we want our model to provide approximations for a site even when the input data are missing (which could be important for e.g. EEW purposes). This resulted in the following composition of target values: 91.4 per cent were observed, while 8.6 per cent were calculated using ShakeMap. We have calculated the first *P* arrival times on the stations from the theoretical travel times using the Java TauP Toolkit by Crotwell *et al*. (1999) that calculates theoretical travel times and paths as implemented in the Obspy Python library (Krischer *et al*. 2015) and the *ak135* velocity model (Kennett *et al*. 1995). We performed a visual check of

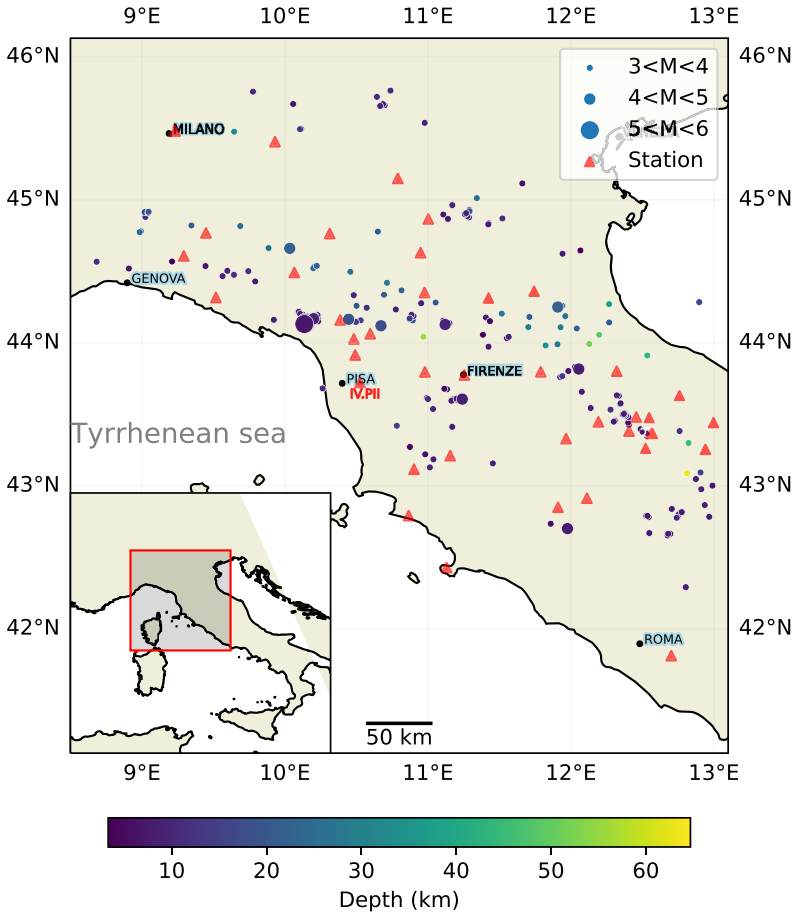

Figure 2. Spatial distribution of the 256 earthquakes (dots—scaled according to earthquake magnitude and coloured according to earthquake depth) together with the 39 stations (red triangles) selected for this study. Station IV.PII, used in the later analysis, is labelled also with a red text.

several waveforms and it was found that the theoretical travel times calculated were satisfactory to the purpose of this study.

We use two other data sets for pre-training the models for TL. The first data set is the J2020 central Italy data set (CI hereinafter) consisting of 915 earthquakes recorded by 39 stations with the same sampling rate and target labels as provided in this study. CI has essentially the same structure as the data set from this study except for a different set of recording stations.

The other data set used consists of a globally distributed set of local earthquake waveforms STEAD (Mousavi *et al*. 2019). This data set is much larger than CI, and it provides 3C single station earthquake waveforms, that is the data set structure is different from the data set of this study. In particular, the waveforms provided by STEAD are not recordings from a fixed set of stations. This essentially means that we cannot use STEAD to pre-train a CNN model that has the same, multistation, architecture for IM prediction that we use in this study although they can be used in the first layers of our CNN model as explained below. STEAD has a sampling rate of 100 Hz and the amplitude units of data counts. The maximum epicentral distance is 350 km. We used only the earthquake waveforms with magnitude $M \geq 3$ (the criterion also used for preparing the data set for this study) providing 106 245 waveforms from STEAD. Detailed information about STEAD can be found in the respective article.

We also checked other available data sets, such as LEN-DB (Magrini *et al*. 2020). However, we chose STEAD because it is a global data set having the data sampled using the same sampling rate as the CI and CW data (100 Hz).

3 METHOD AND TRAINING

The CNN model adopts the architecture proposed by J2020, using the Keras Python library (Chollet *et al*. 2015). Input to the model is a combination of all the waveform data (all 39 stations) for a given earthquake, for an input array size (39, 1000, 3), where 39 is the number of stations, 1000 is the number of samples (with a sampling rate of 100 samples per second and a 10 s window) and 3 is the number of components. The ordering of the stations is always preserved. The waveform data for each earthquake starts 3 s before the estimated *P* arrival time at the station nearest to the epicentre, or at earthquake origin time if the estimated *P* arrival is less than 3 s after origin time (for consistency with J2020 where all the arrivals at the closest station were maximally 3 s after origin time). The data are normalized by the input maximum (i.e. the largest amplitude observed across all stations within the input time window), and this maximum is saved as a normalization value which is later inserted into the network. The model outputs are arrays of size (39,5), where 39 is the number of stations, and 5 is the number of predicted IMs per station (i.e. PGA, PGV and SA at 0.3, 1 and 3 s periods). We applied the base-10 logarithm to all the IMs (i.e. $\log_{10}$IM). Relative to J2020, we made a small change to the architecture—we moved the dropout to the flattened layer, instead of the layer that was combining

the metadata and the flattened layer (Fig. 3a). We did this because in one of the tests we add additional 117 (39×3) constant data (the station distances and azimuths to a reference station and the $v_{S,30}$ at the stations) to the metadata layer, and in this way, we ensure that the constant metadata inserted are always present, throughout the model-training process.

3.1 Transfer learning

Next, we describe the concept of TL. The outcome of ML optimization techniques involving uncertainty, such as gradient descent applied to a non-convex cost function, often depends on the initial values assigned to network weights (e.g. neural network parameter values). Weight initialization is usually done by random sampling with augmentation, e.g. He or Glorot (Bengio *et al*. 2017). Depending on the outcome of random initialization, the training process is likely to converge to a local optimum which might or might not be close to the global one (the one implying good generalization properties of the model). This is also heavily influenced by the model topology and also the set optimization characteristics (i.e. the hyperparameter values). This problem is further accentuated when using more-complex-domain data sets, and/or when using smaller-sized data sets. This effect is to some extent mitigated by the depth of the neural network—using a deeper network is usually better—albeit not entirely. Therefore, choosing a good initialization of network weights is likely to result in a better performing model, and is also likely to give faster training, that is in fewer epochs because of faster convergence. TL can be used to enforce a more likely 'good initialization' of network weights, such that will likely result in a better performing model. More formally, a model M1 learned from data set D1 can be co-adapted (i.e. specialized) to data set D2, by using the useful features (i.e. reusable network weights) from M1 to initialize the appropriate features of M2, and then training M2 on D2. By useful features, we denote those network weights that are used for feature extraction in deep learning. In a CNN model, these are usually the weights tied to convolutional filters. The remaining network weights values in M2 are, again, initialized randomly. The process of training M2 on D2 is usually done analogously to the training of M1 on D1, however ordinarily by using smaller learning rates for the transferred weights. By doing this, we prevent erratic changes in the feature extraction layers influenced by the uncertainty in the classification layers, especially during early model training, that is in the beginning epochs. In practice, it is not always clear if the tasks M1 and M2 or data sets D1 and D2 are similar enough to each other to warrant improvement when using TL over training the M2 with random weights initialization. In this study, we have explored the use of TL to answer these questions. Furthermore, Otović *et al*. (2021) showed that TL for applications of CNNs on time-series data can be useful even in the cases where D1 and D2 are from different domains of time-series (e.g. medicine and seismology), which makes us more confident that TL is a viable strategy for improving the results on small data sets in seismological applications of ML. This processing is somewhat analogous to the addition of prior information in geophysical inversion. Examples include the non-linear inversion when locating earthquakes where the initial location is assigned next to the first recording station, or when performing 3-D tomographic seismic velocity inversions where the best fitting 1-D model is used as the starting model of the non-linear iterative procedure.

3.2 Experiments and training

To investigate the effect of TL on model performance when the training data are insufficient, we perform a series of experiments. They are illustrated in Fig. 4 and are explained in more detail below. First, we train the model from the very beginning using only the available CW data (first experiment—No TL). Next, we experiment with TL by using a model pre-trained on the same type of problem (CI data set; second experiment CI → CW). In addition, we also experiment with TL from a pre-trained model on a different seismological problem (magnitude characterization; third experiment, STEAD → CI) to improve the CI model. We then use the improved CI model as a pre-trained model for TL on CW data (fourth experiment, STEAD → CI → CW). After that, we test the addition of station ($v_{S,30}$) and interstation information (distances and azimuths) as additional inputs to the CNN model (fifth experiment, CI → CW + additional data). In the last test, we change the output of the CNN to predict the IMs only at one station (sixth experiment). Specific modelling choices were made by trial-and-error while observing model performance on the validation subset.

To train the CNN with random weights initialization (first experiment—No TL), the weights are initialized using the *Glorot* uniform initializer (Glorot & Bengio 2010).

For TL from a pre-trained model, the initial model weights are taken directly from parts of the CNN model from J2020 (second experiment, CI → CW). More specifically, only the weights of the first two convolutional layers are used, while the remaining layers are initialized using the Glorot uniform initializer. This has been done since the first two layers in this architecture extract seismogram features irrespective of the station specifics (geographical pattern, soil, topography, etc.), while the subsequent layers combine the extracted features from individual stations exploiting the network station pattern.

The data set used in J2020 is relatively small, having only 915 earthquakes. Therefore, we use another database of earthquake waveforms, STEAD, which provides a large number of training data to improve the feature extractors (i.e. the first two convolutional layers) and exploit TL. To this end, we constructed a CNN model for magnitude determination from single-station earthquake waveforms (Fig. 3b), designing an architecture in which the filters from the first two layers are easily transferable for our task. Then we used the model pre-trained on STEAD to train the model for central Italy IM prediction (STEAD → CI, i.e. improving the results achieved in the J2020 study). After the validation loss stops improving, all the layers are then fine-tuned, that is set as trainable with a small learning rate of 10^{-5}, and the training is continued. The CI CNN model trained using TL was then used (i.e. TL applied again) as a pre-trained model for training with the CW data set (fourth, STEAD → CI → CW, experiment).

In the next stage of experimentation (fifth experiment, CI → CW + additional data), we added the information about the stations (interstation distances and azimuths, $v_{S,30}$ of the stations) to the metadata layer (Fig. 3a). The interstation distances and azimuths chosen are the distances and azimuths of every station from an arbitrarily chosen reference station (ASQU), where we used 0 for the values of the reference station, producing an input of shape 39×3.

In the last part of the experiment, we use only a single station for output (Fig. 3a), station PII, to see how much our algorithm could be improved when the target task was 'simplified' (sixth experiment). Separately, we also experimented with the adoption of shorter time windows.

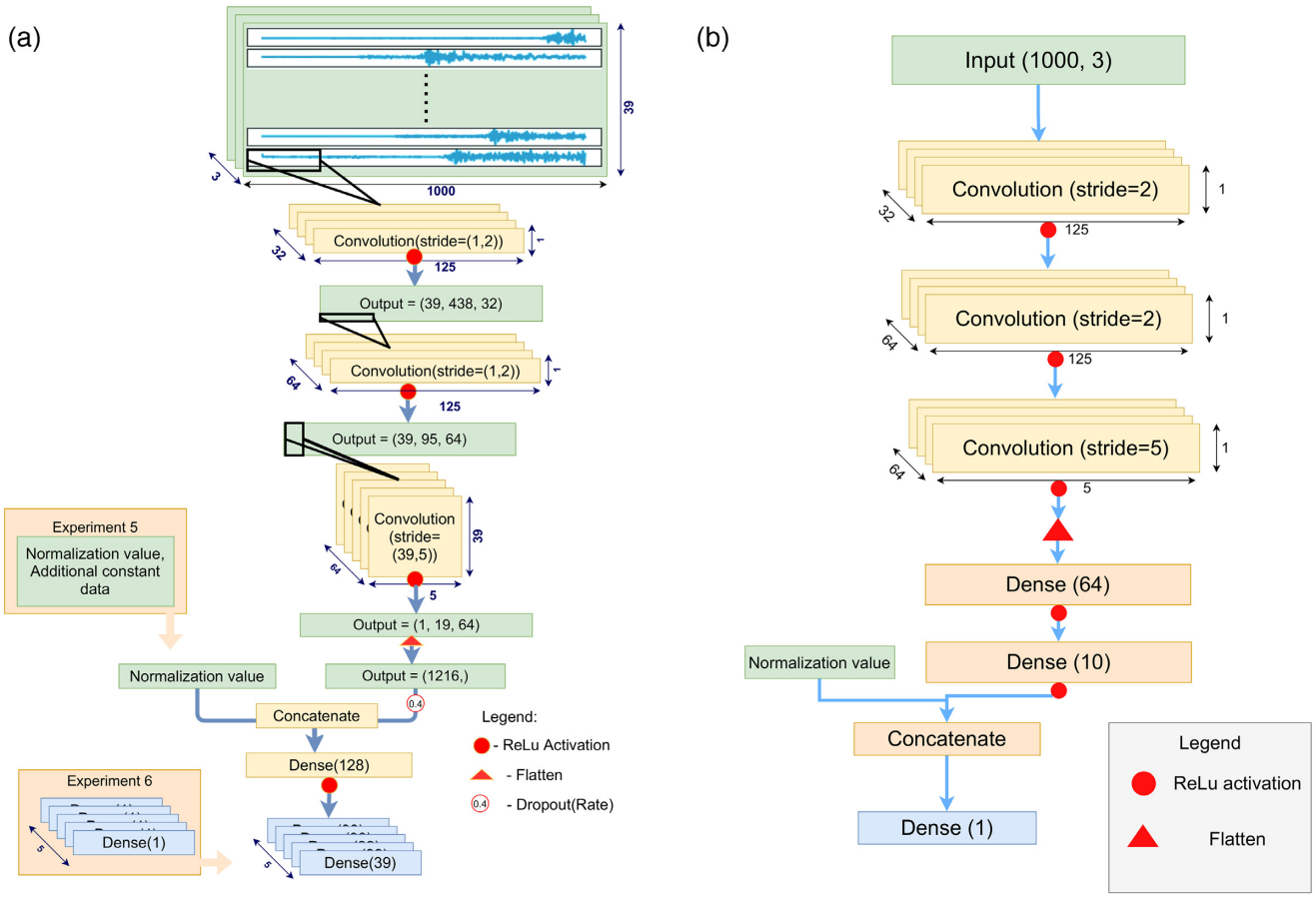

Figure 3. The architecture of the CNN models used. (a) Model for IM prediction. Boxes shaded yellow represent filter banks and operators, boxes shaded green represent inputs and the boxes shaded blue represent the final output. Orange boxes represent modifications of the model during the experimentation stage, with the title of the boxes denoting the experiment at hand. (b) Model for magnitude determination. Boxes shaded green represent inputs, shaded yellow represent convolutional layers, shaded orange represent fully connected layers, and box shaded in blue represents the output.

The data of the CW data set had been split randomly into training (80 per cent) and test (20 per cent) subsets. For evaluation of the performance of the CNN model, we use fivefold cross-validation, which splits the randomly permuted data set into five equally sized disjoint subsets, and uses each of them as the test set to the model trained on the remaining four subsets. We have done all the modifications in the model on only one subset, and, when finally defined, we run the training on all the subsets. We then join the predictions on the test subsets from all five folds. To test the statistical significance of the difference in results, we calculated the squared errors of the different experiments and then used the Wilcoxon signed-rank test to check if the differences between them are statistically significant (Wilcoxon 1992). The analysis of the results is presented in the following section.

The batch size used for optimization was 8, and the mean squared error (MSE) loss function was used for model optimization. Hyperparameter values for model optimization were based on J2020. We have also used an early stopping procedure, which terminates the training if the validation loss did not improve for 25 epochs. Together with the early stopping, we also reduce the learning rate by a factor of 0.2 if the validation loss did not improve for 8 epochs, with the minimum possible learning rate being 10^{-5}. We used an Nvidia 1050 4GB graphics card for training, which took 13 and 5 min on average, for training without TL and when TL approach was used, respectively.

4 RESULTS

Here, we present the results of all the 6 experiments. For each experiment, the CNN model results from all fivefold cross-validation test sets are averaged. The mean squared errors were: 1.69 for the no TL experiment, 1.41 for the CI → CW experiment, 1.42 for the STEAD → CI → CW experiment and 1.28 for the CI → CW + additional data experiment. Residuals $R = \log_{10}(\mathrm{IM}_{\mathrm{obs}}/\mathrm{IM}_{\mathrm{predicted}})$ are calculated and the outlier records with residuals (of any IM) $|R|>1$ are replaced with 1. We replaced the outliers with 1 to limit the effect of seriously erroneous predictions, which have here been defined as those predictions that are at least an order of magnitude (i.e. $|R| > 1$) different from the observed values. Finally, we calculated the mean, median and standard deviation of the residuals (hereinafter referred to as results). All the experimental setups are illustrated in Fig. 4. In Tables 1–5, we present the results for the experiments on CW data separated by IM, and in Table 6 we report the number of outliers. In Fig. 5, we present the results using box-plots for each IM. For experiments 1, 2, 4 and 5 we report the results on all the data here, and the results are split into residuals on observed target labels and ShakeMap-derived target labels in the Supporting Information. For the other experiments we report here the results separated into observed target labels and ShakeMap derive target labels.

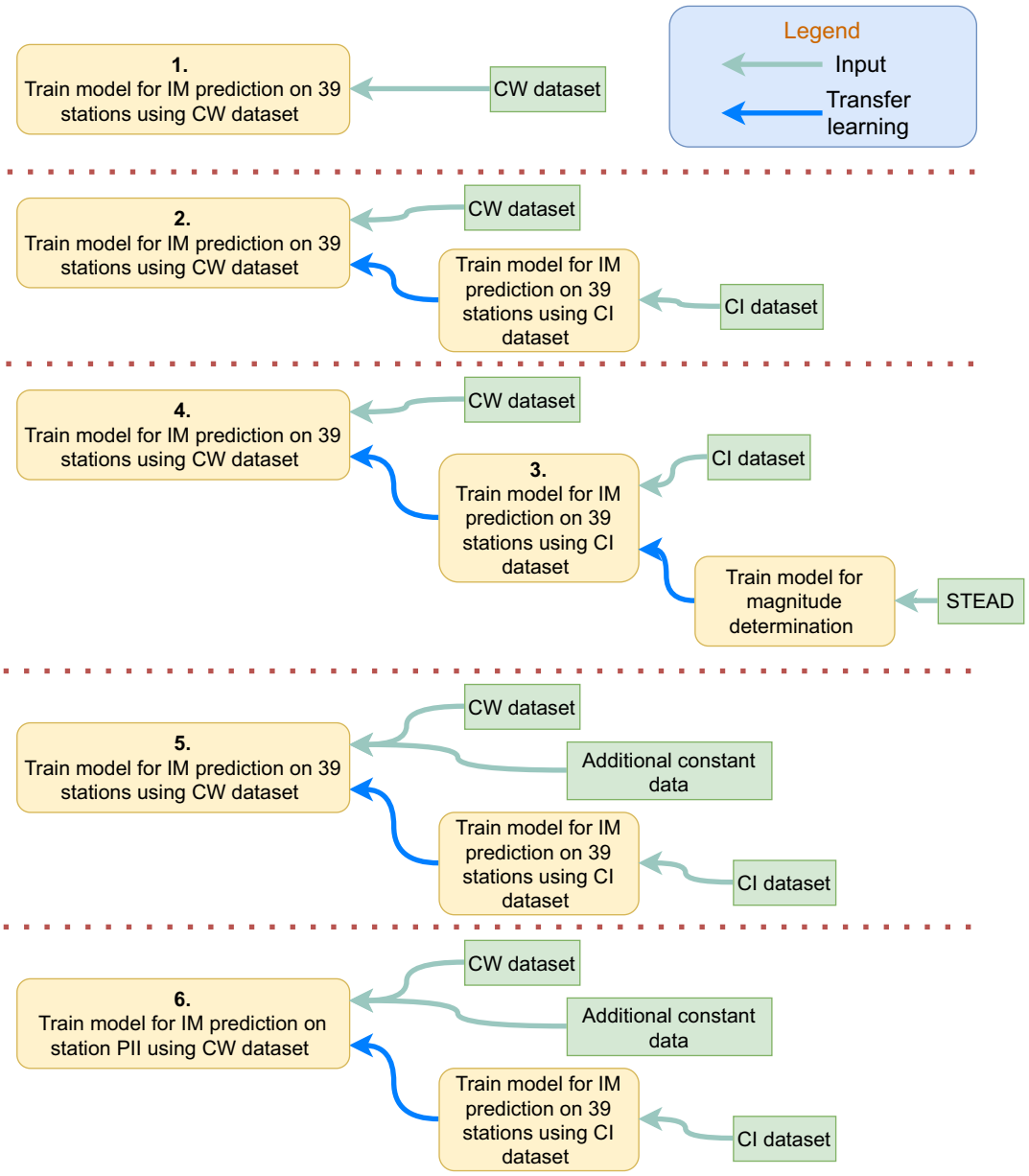

Figure 4. Diagram of the tests performed. The yellow boxes denote the trained CNN models, with the number in the box marking the number of the test. Input data are represented by the green boxes. Arrows represent input (green) and TL (blue).

Table 1. PGA residual R statistics for the four experiments on CW data set.

Experiment	Median	Mean	STD
No TL	−0.061	−0.029	0.508
CI → CW	−0.048	−0.019	0.474
STEAD → CI → CW	−0.039	−0.019	0.479
STEAD →CI → CW + add. data	−0.037	−0.013	0.449

Table 2. PGV residual R statistics for the four experiments on CW data set.

Experiment	Median	Mean	STD
No TL	−0.081	−0.04	0.514
CI → CW	−0.066	−0.031	0.467
STEAD → CI → CW	−0.05	−0.022	0.470
STEAD →CI → CW + add. data	−0.047	−0.021	0.444

Table 3. PSA 0.3 s residual R statistics for the four experiments on CW data set.

Experiment	Median	Mean	STD
No TL	−0.055	−0.022	0.508
CI → CW	−0.047	−0.014	0.480
STEAD → CI → CW	−0.027	−0.008	0.482
STEAD →CI → CW + add. data	−0.033	−0.004	0.455

4.1 Training with random weights initialization

The results of the CNN model trained with random weights initialization (using the Glorot uniform initializer) are presented here. The results are shown in Tables 1–6 and Fig. 5. The results separated for the data with observed targets and with ShakeMap calculated targets are available in the Supporting Information Section C.1.

Table 4. PSA 1 s residual R statistics for the four experiments on CW data set.

Experiment	Median	Mean	STD
No TL	−0.058	−0.023	0.507
CI → CW	−0.045	−0.01	0.475
STEAD → CI → CW	−0.032	−0.008	0.474
STEAD →CI → CW + add. data	−0.034	−0.008	0.459

Table 5. PSA 3 s residual R statistics for the four experiments on CW data set.

Experiment	Median	Mean	STD
No TL	−0.07	−0.035	0.511
CI → CW	−0.052	−0.028	0.463
STEAD → CI → CW	−0.036	−0.014	0.461
STEAD →CI → CW + add. data	−0.046	−0.018	0.451

Table 6. Number of outliers $|R| > 1$ for the four experiments on CW data set.

Experiment	Number of outliers (per cent)
No TL	15.68
CI → CW	12.04
STEAD → CI → CW	12.56
STEAD → CI → CW + add. data	10.55

4.2 Transfer learning from a model pre-trained on central Italy data set

The results of the CNN model trained using TL from the pre-trained CI CNN model are shown in Tables 1–6 and Fig. 5. The results separated for the data with observed targets and with ShakeMap calculated targets are available in the Supporting Information Section C.2.

When the results are compared with those of the previous section, we can see that improvement comes in the reduction of the number of outliers and a decrease in standard deviation of the residuals. There was a statistically significant improvement for all the five IMs (p-value < 0.05). We achieved the best results when the weights of the first two convolutional layers are used from the pre-trained model with their learning rates set to zero, while the remaining layers are initialized using the Glorot uniform initializer. We have also tried to train the CNN model using TL while leaving all layers to be fully trainable; however, this deteriorated the performance, suggesting that the feature extractors learned in the first two layers of the pre-trained CNN model are useful for the training on this data set. This means that the first two layers in our architecture, that is those that analyse single station waveforms are transferable between models used for training on different areas, but should be trained on large, high-quality data sets.

4.3 Improving the CI model by TL from a different task

The results for CI using a pre-trained model for magnitude determination from STEAD data are presented here. We report the data split into residuals on observed target labels and ShakeMap-derived target labels for an easier comparison with the J2020 study results. The results have been calculated as in previous chapters, except for the outliers which were discarded from the calculations (instead of being set to have their absolute value to 1) to be consistent with the J2020 study on CI data. Removal of the outliers resulted in 96.45 per cent data kept for observed IM targets and 93.84 per cent for ShakeMap generated IM targets. Results are shown in Table 7. We also report the results from the J2020 trained on CI data in Table 8. In J2020, large residual values $|R|>1$ were also removed resulting in 92 per cent for the observed IMs and 87.49 per cent of the data kept for the ShakeMap predictions.

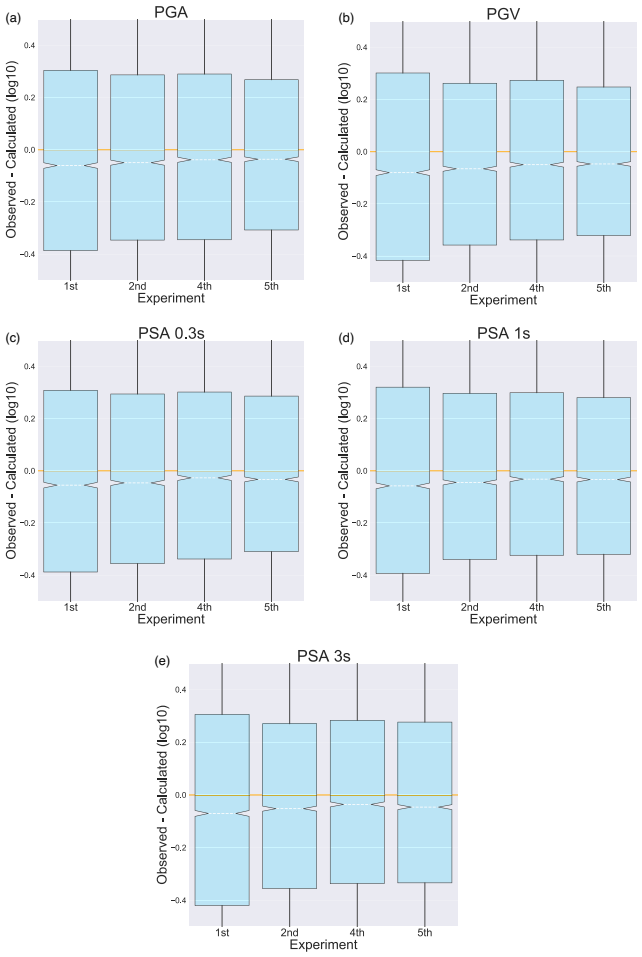

Figure 5. Boxplots for the residuals $\log_{10}\mathrm{IM}_{\mathrm{observed}}$—$\log_{10}\mathrm{IM}_{\mathrm{predicted}}$ of the CNN training on the CW data set from Sections 4.1 (No TL experiment), 4.2 (CI → CW experiment), 4.4 (STEAD → CI → CW experiment) and 4.5 (CI → CW + additional data experiment) for different IMs: (a) PGA, (b) PGV, (c) PSA 0.3 s, (d) PSA 1 s and (e) PSA 3 s. The boxes show the interquartile range, the white dashed line shows the median and the orange line shows zero.

Compared to the previous study J2020 (Table 8), we achieved an improvement by using transfer learning in reducing the number of outliers, the standard deviation of the residuals and their bias (i.e. the difference of $|R|$ from 0). We also found that the results are slightly better when using the STEAD pre-trained model. The results not separated into the data with observed targets and with ShakeMap calculated targets are available in Supporting Information Section C.3.

The best results were achieved when the first layer learning rate was set to 0 and the weights of the second layer were initialized with the pre-trained weights but left trainable. We achieved further improvement in the results by fine-tuning the weights of all the layers with a learning rate of 10^{-5}.

Table 7. IMs' residual *R* statistics for the STEAD → CI experiment: the CNN predictions on the CI data set using TL from the pre-trained STEAD model. The results are reported for observed IMs (for the stations having recorded data) and the ShakeMap predictions (for the stations that had no recorded data).

IM	Observed median	Observed mean	Observed STD	ShakeMap median	ShakeMap mean	ShakeMap STD
PGA	−0.011	−0.005	0.276	−0.010	−0.025	0.331
PGV	−0.017	−0.005	0.252	−0.046	−0.040	0.317
SA(0.3)	−0.005	−0.0003	0.28	−0.011	−0.011	0.334
SA(1.0)	−0.007	0	0.266	−0.058	−0.041	0.319
SA(3.0)	−0.013	0	0.287	−0.091	−0.075	0.352

Table 8. IMs' residual statistics for the CNN predictions for the observed IMs (for the stations having recorded data), ShakeMap predictions (for the stations that had no recorded data) and the predictions of GMPE by Bindi et al. from the study Jozinović *et al*. (2020).

IM	Observed median	Observed mean	Observed STD	ShakeMap median	ShakeMap mean	ShakeMap STD	GMPE median	GMPE mean	GMPE STD
PGA	0.038	0.035	0.346	0.059	0.038	0.372	0.013	0.017	0.352
PGV	0.036	0.034	0.338	0.043	0.041	0.380	− 0.174	− 0.151	0.33
SA(0.3)	0.031	0.031	0.34	0.056	0.046	0.37	− 0.284	− 0.252	0.359
SA(1.0)	0.029	0.034	0.338	0.001	0.017	0.374	− 0.207	− 0.198	0.303
SA(3.0)	0.019	0.027	0.374	− 0.037	− 0.012	0.404	0.026	0.083	0.368

4.4 Using the newly trained CI model for TL

Since the results obtained for the CI experiment of J2020 have been improved compared to the previous study (Section 4.3), we tried to use the improved pre-trained model trained on CI data to improve the TL for the CW data set. The results are shown in Tables 1–6 and Fig. 5. The results separated for the data with observed targets and with ShakeMap calculated targets are available in Supporting Information Section C.4. When comparing the results with the results in Section 4.2 (pre-trained CI model trained with random weights initialization) we can see that the results were comparable overall. There was no statistically significant difference for all the five IMs (all the p-values were > 0.05) when comparing the results with those in Section 4.2. We also calculated the results using the STEAD pre-trained model directly on CW data and found similar results (results available in Supporting Information Section C.5).

4.5 Adding additional knowledge

The results with interstation distances and azimuths as additional metadata added to the model are reported here. The results are shown in Tables 1–6 and Fig. 5. The results separated for the data with observed targets and with ShakeMap calculated targets are available in Supporting Information Section C.6. In Supporting Information Section C.7, we show the results of the model with added interstation distances and azimuths if no transfer learning was used.

When comparing with the results of the previous sections on training the model on the CW data set (Sections 4.1, 4.2 and 4.4) we see an improvement in the smaller number of outliers and reduced standard deviation. There was a statistically significant improvement for all the 5 IMs (p-value < 0.05).

The best results were achieved after scaling the distance (expressed in km) by dividing by the maximum distance (246 km), and with azimuths specified as sine and cosine of the azimuths. The addition of $v_{S,30}$ did not improve the results.

4.6 Results for station PII

We have extracted the residuals between the true values and the CNN model predictions trained using TL described in Section 4.5 for the station PII (part of the original set of 39 stations). Station PII was chosen as an example of usage of our algorithm for EEW purposes, where the warnings received at station PII could be used for a point of interest nearby (e.g. an industrial plant, a high risk monument, etc.). Using the same criteria for outliers as before, the mean, median and standard deviations of the residuals were calculated and are shown in Tables 9–14 (experiment marked as *Original* in the tables). We report the results for the observed targets and ShakeMap predicted targets separately and we note that the station PII had missing input waveforms for 42.9 per cent of earthquakes. This is likely the reason why the results for the data with waveforms present in the input show a deterioration of the results in the number of outliers and standard deviation compared to the data with input waveforms not present for the station PII. They also show slightly lower (absolute) bias, with the bias having opposite signs.

4.7 Using station PII as the only model target

The previous results in Section 4.6 are showing the results for station PII when the targets were the IMs at the 39 stations. In this section, we report the results when we use only the station PII as the model target. Using the same criteria for outliers as before, the mean, median and standard deviations of the residuals were calculated and are shown in Tables 9–14 (experiment marked as *10 s PII target* in the tables). Compared to Section 4.6, where 39 stations were used as a target, we see an improvement in all the metrics for the observed target values. For the target labels derived from ShakeMap predictions, the results for the standard deviation are similar in some IMs, whereas the bias and the number of outliers are always reduced.

We also test how reducing the waveform window length affects the results by using a window length of 7 s (in all the previous sections the waveforms had a window length of 10 s). Using the same

Table 9. PGA residual *R* statistics for the four experiments on station PII.

Experiment	Median (Observed)	Mean (Observed)	STD (Observed)	Median (ShakeMap)	Mean (ShakeMap)	STD (ShakeMap)
Original	0.171	0.208	0.455	−0.256	−0.222	0.44
10 s PII target	0.116	0.089	0.419	−0.117	−0.117	0.42
7 s PII target	0.105	0.114	0.463	−0.122	−0.118	0.446
GMPE	0.419	0.411	0.281	–	–	–

Table 10. PGV residual *R* statistics for the four experiments on station PII.

Experiment	Median (Observed)	Mean (Observed)	STD (Observed)	Median (ShakeMap)	Mean (ShakeMap)	STD (ShakeMap)
Original	0.133	0.136	0.417	−0.187	−0.199	0.415
10 s PII target	0.073	0.055	0.379	−0.137	−0.089	0.391
7 s PII target	0.082	0.082	0.453	−0.079	−0.082	0.417
GMPE	0.162	0.147	0.269	–	–	–

Table 11. PSA 0.3 s residual *R* statistics for the four experiments on station PII

Experiment	Median (Observed)	Mean (Observed)	STD (Observed)	Median (ShakeMap)	Mean (ShakeMap)	STD (ShakeMap)
Original	0.256	0.238	0.425	−0.301	−0.254	0.449
10 s PII target	0.125	0.123	0.416	−0.21	−0.153	0.441
7 s PII target	0.162	0.161	0.454	−0.191	−0.141	0.447
GMPE	0.353	0.331	0.268	–	–	–

Table 12. PSA 1.0 s residual *R* statistics for the four experiments on station PII.

Experiment	Median (Observed)	Mean (Observed)	STD (Observed)	Median (ShakeMap)	Mean (ShakeMap)	STD (ShakeMap)
Original	0.096	0.112	0.439	−0.17	−0.118	0.441
10 s PII target	0.046	0.035	0.411	−0.107	−0.043	0.427
7 s PII target	0.024	0.04	0.452	−0.041	−0.054	0.456
GMPE	0.001	−0.017	0.337	–	–	–

Table 13. PSA 3.0 s residual *R* statistics for the four experiments on station PII.

Experiment	Median (Observed)	Mean (Observed)	STD (Observed)	Median (ShakeMap)	Mean (ShakeMap)	STD (ShakeMap)
Original	0.022	0.057	0.387	−0.126	−0.064	0.463
10 s PII target	0.014	0.013	0.37	−0.035	−0.016	0.415
7 s PII target	−0.002	0.033	0.411	−0.032	−0.14	0.454
GMPE	0.117	0.122	0.365	–	–	–

criteria for outliers as before the mean, median and standard deviations of the residuals were calculated and are shown in Tables 9–14 (experiment marked as *7 s PII target* in the tables).

We also calculate the predictions for the station PII using the GMPE of Bindi *et al*. (2011), for which we used the final earthquake location and magnitude. We used the station magnitude (station PII) to calculate the GMPE predictions if it was available through INGV web services. The predictions were calculated only for the earthquakes for which the station PII had recorded data in Tables 9–14). The residuals *R* were calculated as for the CNN predictions. The median, mean and standard deviation of *R* are reported in Tables 9–14 (experiment marked as *GMPE* in the tables). By comparing the

Table 14. Number of outliers $|R| > 1$ for the four experiments on station PII.

Experiment	Number of outliers (Observed) (per cent)	Number of outliers (ShakeMap) (per cent)
Original	8.55	11.4
10 s PII target	7.24	7
7 s PII target	9.2	10.5
GMPE	12.82	–

results in Tables 9–14, we can see that the GMPE of Bindi *et al*. (2011) performs better on station PII in terms of variability, while CNN performs better in terms of bias of the results and the number of outliers.

5 DISCUSSION

In the study, we have shown that the introduction of TL in our CNN model can compensate for the lack of data and improve the results for rapid prediction of earthquake IMs, We have found that using a pre-trained CNN model, trained on CI in J2020, improved the results obtained using only the CW data set (Section 4.2, Fig. 5). Improvements (mean of the improvements of the five IMs) include: reducing the number of outliers by 4 per cent, the residuals median by 21 per cent and their standard deviation by 7 per cent. These results suggest that TL is a viable technique to improve model performance on small-sized data sets. The convolutional filters in the first two layers of the CNN model (i.e. those used for TL) are the same for all the stations, which means that they have to be general enough to be able to extract features from the inputs regardless of which station they are currently analysing. The third layer, which looks at cross station information (i.e. the station pattern of the ground motion and station-specific site amplifications due to soil type, topography, etc.), did not improve the results when included in TL for our problem. This was expected since the convolutional filters used in the third layer are of height 39 and span the specific geometrical/geographical pattern of the recording stations. Therefore, inserting a sequence of recording traces for TL unrelated to that target problem will worsen the results, unless the same geographical pattern is used and the stations reflect the same characteristics in terms of local site amplifications—a highly unlikely situation in practice. Given that our first two layers use single station filters, we could use the first two layers of the pre-trained model regardless of the number of stations used to create the input data. This can be useful if we want to train our model for an area that has a different number of stations available.

We also show in Section 4.3 that parts of the CNN model, which was trained for a different problem (single station magnitude determination), can be used for TL on our CNN model, with an improvement on the results of the J2020 study in the number of outliers, the standard deviation of the residuals and their bias. Lower levels of improvement on the stations with missing data could be explained by the fact that the pre-trained model (i.e. trained on STEAD) did not have stations in which the waveform data were not present, which is instead the case for 7 per cent of the data in CI. It is, however, noteworthy that the CNN model was still able to compensate for the missing data even if the first two layers (pre-trained on STEAD) were not pre-trained with missing data. The STEAD waveforms were velocity waveforms in counts, that is different units as the waveforms used for CI and CW. However, as the first two layers of our model operate on normalized waveforms without absolute amplitude information, and the normalization constant is only inserted later in the model, the use of data in counts for pre-training the first two layers should not adversely affect the model performance.

STEAD consists of earthquake waveforms recorded on available stations up to a specified distance from the earthquake epicentre, in contrast to our use of a fixed set of stations with waveforms for a set of earthquakes for training our IM prediction model. This difference essentially means that we cannot use these data sets to pre-train a CNN model that has the same, multistation, architecture for IM prediction as we use in this study. However, this problem was circumvented by pre-training a single station and magnitude determination model instead, and then reusing its specific transferable layers for IM prediction, which is our target goal.

When using the pre-trained model trained on the same problem (CI IMs prediction, Section 4.2), the best results were achieved when the learning rate of the first two layers was set to 0. When using the model trained on the different problem instead (magnitude determination, Section 4.3), they were achieved when the first layer learning rate was set to 0 and the second remained trainable. This suggests that the features extracted by the first layer of the magnitude determination model can be directly used for IM prediction, without the need for further parameter fine-tuning. Model performance can be improved, however, by fine-tuning the second layer parameters to the target domain (i.e. the IM prediction). Moreover, further improvement achieved by fine-tuning all the layers showed that even the weights of the first layer, which was used from the pre-trained model, could be further adapted to better fit the new domain. This was achieved by using a smaller learning rate for fine-tuning, whereas, on the other hand, using greater learning rates did not improve model performance. When we tried to fine-tune the first two layers using a TL model in the same domain, the results did not improve, and fine-tuning led to overfitting.

The comparison of the results in Sections 4.2 and 4.4 suggests that the weights of the first two pre-trained layers were already satisfactory for TL on our problem and that the training bottleneck was in the deeper layers of our model. The CNN model is learning the interstation relations (locations, distances) and the characteristics of the stations implicitly during the training, as no station information has been given to it. Explicitly providing the interstation distances and azimuths improved the results of the model (Section 4.5). This is not a form of TL, but it does follow a very similar philosophy in that we guide the model to obtain useful features that describe the data without needing to learn how to extract them from more implicit representations. Normalizing the interstation distances by the maximum of the distances improved the results, which follows the suggested normalization for neural network inputs (LeCun *et al*. 2012). The azimuth between the stations, Az, was initially provided in degrees and radians without any results improvement. The results were improved with the use of $\sin(Az)$ and $\cos(Az)$, which are effectively normalizing azimuth description to the range $[-1,1]$. This azimuth description also gives the neural network a more meaningful measure of the closeness of two angles and removes the possible ambiguity that could come from using the angles (e.g. the angles 360° and 0° are the same angle, but the numerical difference between them is 360). Giving only the sine or cosine would also confuse the model, as they are both non-injective functions; however, providing both allows for an accurate description of the angle. In contrast, the addition of $v_{S,30}$ did not improve the results of the CNN model. Overall, when we compare the results in Sections 4.5 and 4.1, we can see that the improvement from the use of TL comes in terms of reducing the number of outliers by 5 per cent, the residuals R median by 39 per cent and their standard deviation by 11 per cent. The main improvement comes from using the pre-trained models, as can be seen from a comparison of the results in Section 4.2 and Supporting Information Section C.7.

The data in our data set are not uniformly distributed: the larger values of IMs are underrepresented in the data set (Fig. 1a). This is a natural consequence of the Gutenberg-Richter power-law earthquake distribution. Biased data sets usually produce biased ML models. In Fig. 6, we plotted linear regression fits to the PGA predictions from our four experiments on CW data set, and a true = predicted line for comparison. We can see that models across all of our

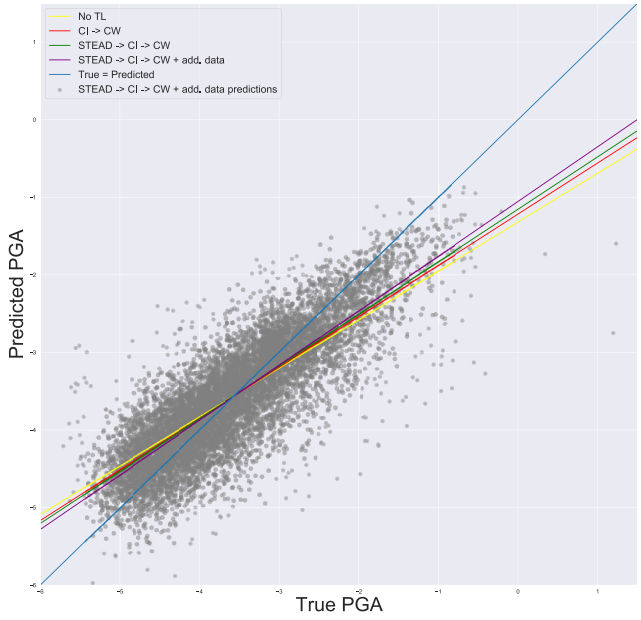

Figure 6. True versus predicted PGA scatter plot of the experiment STEAD $\rightarrow$CI $\rightarrow$ CW + additional data (grey points). The blue line represents the true = predicted values. The other lines represent a linear regression fitted to the PGA predictions of the four models: No TL model (yellow line), CI $\rightarrow$ CW model (red line), STEAD $\rightarrow$CI $\rightarrow$ CW model (green line) and STEAD $\rightarrow$CI $\rightarrow$ CW + additional data model (purple line).

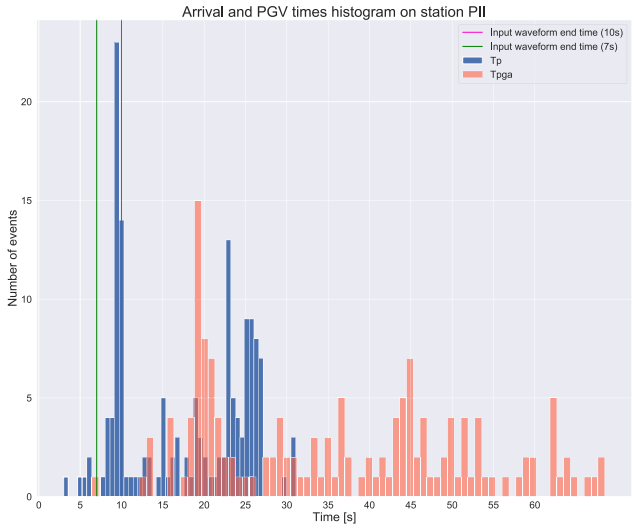

Figure 7. Histograms of waveform arrival and peak ground motion delay times. Blue bars show the difference between the *P* arrival time at the station PII and the input waveform start time T_p. Light orange bars (reddish when overlapping with the T_p blue bars) show differences between the recorded PGA time and the input waveform start time T_{pga}. Green and magenta vertical lines show the end of the 7 and 10 s input waveforms, respectively. The bars on the right of those lines show the number of events, with possible warning times before the *P* arrival or the PGA time on station PII. The total number of events shown for T_p is 266, while for the T_{pga} only the 152 events with recorded waveforms (i.e. recorded PGA time) are shown.

four experiments were biased: they were underpredicting large PGA values and overpredicting small PGA values. We can also see that the introduction of TL and additional data somewhat reduced the bias of our models. The improvement, however, was not large. We also tried to use sample weights (see e.g. Wu & Datta 2021) when training the model (described in detail in Supporting Information Section C.8), which did not reduce the model bias. Münchmeyer *et al*. (2020) used oversampling of large events during the pre-training and training phase, and by adding data of large magnitude earthquakes from other regions during training. Our CNN model can only accept data that always come from the same input stations, making it impossible to add data from other regions. We, however, tried to change the magnitude distribution in our pre-training data set: we used only earthquakes with magnitude > 4 from the STEAD data set (24 774 earthquakes) for training the magnitude determination model. We then trained the STEAD $\rightarrow$ CW + additional data model (with the first layer set to not being trainable) and found that the modification of the pre-training subset does not reduce the bias of our CNN model on large IM values.

We performed an example of the usefulness of our algorithm for EEW purposes at the station PII: we calculated the differences T_p between the first *P* arrival times on the station PII (IV network) and the start time of the input waveforms, and show them in Fig. 7 (blue histogram bars). In Fig. 7, we see that, for many earthquakes, the CNN model would be able to give an early warning in the vicinity of station PII, depending on the length of the input window used. This rate is greater if we consider that the peak ground motions originate from *S* and surface waves, which arrive later, giving some more warning time before the strongest shaking. Therefore, in Fig. 7 we also show the differences T_{pga} (light orange histogram bars) between the recorded PGA times at the station PII and the input waveform start times. These results do not account for the times needed for running the algorithm (very minor indeed) and for data transmission. It follows that the timing relation for T_p and T_{pga} shown in Fig. 7 support the use of our CNN model as an EEW system for the area around Pisa, as the large majority of T_{pga} occur well after the 10 s input waveform end time (pink line in Fig. 7) so there would be warnings before the strongest shaking for a large majority of earthquakes. Moreover, if we reduce the input waveform length (Section 4.7), we can see in Fig. 7 that these warnings for a large number of earthquakes could be given even before the first *P* arrival at PII (i.e. the large majority of T_p are after the 7 s input waveform end time (green line)). An example earthquake input waveforms are shown in Supporting Information Fig. S1, with an epicentral distance from station PII of 66 km, with $T_p = 9.42$ s and $T_{pga} = 18.82$ s.

The results shown in Section 4.6 suggest that the algorithm would give useful and timely predictions, for cases both with and without waveform data for station PII. Contrary to the results in the sections before, the CNN performs better on the data without the input waveforms present. The main difference in the case of station PII compared to the other stations used is the lower number of earthquakes for which the input waveforms are available (152 of 266 earthquakes). This means that it has less training data to learn how to predict the actual IMs recorded on the stations, and more data to learn how to predict from the ShakeMap derived values at those stations. It is also interesting to note that the CNN model is underpredicting the IMs for earthquakes for which the input waveforms exist, and overpredicting for earthquakes for which there are no input waveforms on the station PII, with the absolute values of mean and median smaller of the residuals *R* on the observed data. However, on the stations with missing data, the CNN results are being compared to ShakeMap predictions, which are calculated using a GMPE of Bindi *et al*. (2011) and geospatial interpolation of the observed values at the stations (Worden *et al*. 2020). We calculated

the IMs on the station PII by using the GMPE of Bindi *et al*. (2011), which uses the final earthquake location and magnitude, for the earthquakes for which observed IMs exist. We then calculated the residuals R and found that the GMPE is also underpredicting the IM values, with a higher median than the CNN model on the same data. This suggests that the overprediction of CNN for the ShakeMap predictions comes from the CNN results having a smaller bias on recorded IM values.

When using only the station PII as a target, we can see that the results of the CNN are improved. This is expected because we are reducing the number of IMs, which are unknown variables to the model, from 195 to 5. In practice, it means that the model layers can be focused exclusively on the information required for predicting the shaking at a single station (PII). This suggests that for achieving the best results for a specific site of interest, individual training should be done. However, the training data does not need to be changed, only the output location. It is also interesting to note that the results were mostly improved, compared to Section 4.6, for the earthquakes with recorded waveforms. Reducing the length of the window, expectedly, led to poorer model performance and larger uncertainties, although there is a trade-off between additional warning time achieved and loss in accuracy. An optimal ratio between accuracy and timeliness can be selected for every application individually. This trade-off also suggests that in a real-time application of the methodology progressively larger time windows could be employed after an earthquake occurs to obtain increasingly accurate predictions (e.g. the approaches of Münchmeyer *et al*. 2020 or Zhang *et al*. 2021).

6 CONCLUSIONS

We used a CNN model to predict IMs (PGA, PGV and PSA at 0.3, 1, 3 s) at 39 stations using 10 s long, multistation waveforms, that start 3 s before the first P-wave arrival. The training set has only 266 earthquakes (with $M > 3$) and TL was used to improve the IMs' prediction accuracy.

We show that TL is a powerful methodology to use in waveform data analysis for cases with insufficient training data. The TL can be done by using a pre-trained model trained on the same problem (IM prediction) or a different one (magnitude characterization)—with both cases improving the model accuracy. We also show that the inclusion of additional knowledge to the model (the interstation relations) can improve the training results.

We find with TL that the first two layers of the pre-trained model are the most important for our problem because they are used for feature extraction from single station inputs, therefore being the only layers that can be reasonably transferred, and that the learning rate of these layers can be set to 0. Therefore, when doing TL using our proposed model, only the third, cross-station layer needs to be retrained. This also implies that the adoption of 39 stations to construct the input waveforms for TL is arbitrary and can be varied according to the target problem, as we use TL only for the single-station feature extractors.

The experiment to predict the IMs at only one station (PII) showed that the simplification of the problem (i.e. reduced indeterminacy deriving from many fewer unknowns) led to performance improvements. We also show that in a theoretical application of our model as an EEW system for points of interest for the area in the vicinity of the station PII, our CNN model could be used to provide warnings for a large number of earthquakes of the selected data set.

7 DATA AND RESOURCES

Earthquake catalogue and waveform data have been downloaded through the INGV FDSN web services (http://terremoti.ingv.it/en/webservices_and_software; INGV Seismological Data Centre 2006; Emersito Working Group 2018). Waveforms have been downloaded and processed using python library Obspy (Beyreuther *et al*. 2010). IMs for the stations with no data have been calculated using USGS ShakeMap 4 (http://usgs.github.io/shakemap/sm4_index.html). The CNN model has been developed using the Keras Python Deep Learning library (Chollet *et al*. 2015). The models and the data of this paper are available on https://github.com/djozinovi/TLpredIM.

ACKNOWLEDGEMENTS

The research has been funded by SERA EU project (Seismology and Earthquake Engineering Research Infrastructure Alliance for Europe; contract no. 730900). It has also greatly benefited from a travel grant for a short-term scientific mission provided by G2NET, COST action CA17137. The authors would like to thank the operators of RSN (INGV) and the USGS ShakeMap team of developers. This work was also partially supported by the project INGV Pianeta Dinamico 2021 Tema 8 SOME (CUP D53J1900017001) funded by Italian Ministry of University and Research "Fondo finalizzato al rilancio degli investimenti delle amministrazioni centrali dello Stato e allo sviluppo del Paese, legge 145/2018".

AUTHOR CONTRIBUTION

The topic of this study was conceived by Alberto Michelini and Dario Jozinović, with contributions from Anthony Lomax and Ivan Štajduhar. Dario Jozinović prepared, downloaded and processed the earthquake data; Alberto Michelini prepared and processed the ShakeMap data. The CNN model has been set up and trained by Dario Jozinović with contribution from Ivan Štajduhar. The results have been analysed by Dario Jozinović with contribution by Alberto Michelini. The final manuscript has been written by Dario Jozinović and Alberto Michelini, with contributions by Anthony Lomax and Ivan Štajduhar.

REFERENCES

Bengio, Y., Goodfellow, I. & Courville, A., 2017. *Deep Learning,* Vol. 1, MIT Press.

Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y. & Wassermann, J., 2010. ObsPy: a Python toolbox for seismology, *Seismol. Res. Lett.,* **81**(3), 530–533.

Bindi, D., Pacor, F., Luzi, L., Puglia, R., Massa, M., Ameri, G. & Paolucci, R., 2011. Ground motion prediction equations derived from the Italian strong motion database, *Bull. Earthq. Eng.,* **9**(6), 1899–1920.

Böse, M., Wenzel, F. & Erdik, M., 2008. PreSEIS: a neural network-based approach to earthquake early warning for finite faults, *Bull. seism. Soc. Am.,* **98**(1), 366–382.

Böse, M., Heaton, T. & Hauksson, E., 2012. Rapid estimation of earthquake source and ground-motion parameters for earthquake early warning using data from a single three-component broadband or strong-motion sensor, *Bull. seism. Soc. Am.,* **102**(2), 738–750.

Bozinovski, S., 2020. Reminder of the first paper on transfer learning in neural networks, 1976, *Informatica,* **44**(3), 291–302.

Chai, C. *et al.*, 2020. Using a deep neural network and transfer learning to bridge scales for seismic phase picking, *Geophys. Res. Lett.,* **47**(16), e2020GL088651.

Chollet, F. *et al.*, 2015. Keras. https://keras.io. Last accessed: 11/2021

Cremen, G. & Galasso, C., 2020. Earthquake early warning: recent advances and perspectives, *Earth Sci. Rev.,* **205,** 103184, doi:10.1016/j.earscirev.2020.103184.

Crotwell, H. P., Owens, T. J. & Ritsema, J., 1999. The TauP Toolkit: flexible seismic travel-time and ray-path utilities, *Seismol. Res. Lett.,* **70**(2), 154–160.

EMERSITO Working Group, 2018. *Rete sismica del gruppo EMERSITO, sequenza sismica del 2016 in Italia Centrale,* Istituto Nazionale di Geofisica e Vulcanologia (INGV), doi:10.13127/SD/7TXEGDO5×8.

García-Laencina, P. J., Sancho-Gómez, J. L. & Figueiras-Vidal, A. R., 2010. Pattern classification with missing data: a review, *Neural Comput. Appl.,* **19**(2), 263–282.

Glorot, X. & Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks, in *Proceedings of the Thirteenth Int. Conf. Artificial Intelligence and Statistics,* Sardinia, Italy, pp. 249–256.

Hsu, T. Y., Wu, R. T. & Chang, K. C., 2016. Two novel approaches to reduce false alarm due to non-earthquake events for on-site earthquake early warning system, *Comput. Aided Civ. Infrastruct. Eng.,* **31**(7), 535–549.

INGV Seismological Data Centre, 2006. *Rete Sismica Nazionale (RSN),* Istituto Nazionale di Geofisica e Vulcanologia (INGV), doi:10.13127/SD/X0FXNH7QFY.

Johnson, S. W., Chambers, D. J., Boltz, M. S. & Koper, K. D., 2021. Application of a convolutional neural network for seismic phase picking of mining-induced seismicity, *Geophys. J. Int.,* **224**(1), 230–240.

Jozinović, D., Lomax, A., Štajduhar, I. & Michelini, A., 2020. Rapid prediction of earthquake ground shaking intensity using raw waveform data and a convolutional neural network, *Geophys. J. Int.,* **222**(2), 1379–1389.

Kennett, B. L., Engdahl, E. R. & Buland, R., 1995. Constraints on seismic velocities in the Earth from traveltimes, *Geophys. J. Int.,* **122**(1), 108–124.

Kohler, M. D., Cochran, E. S., Given, D., Guiwits, S., Neuhauser, D., Henson, I. & Schwarz, S., 2018. Earthquake early warning ShakeAlert system: west coast wide production prototype, *Seismol. Res. Lett.,* **89**(1), 99–107.

Kong, Q., Allen, R. M., Schreier, L. & Kwon, Y. W., 2016. MyShake: a smartphone seismic network for earthquake early warning and beyond, *Sci. Adv.,* **2**(2), e1501055.

Kriegerowski, M., Petersen, G. M., Vasyura-Bathke, H. & Ohrnberger, M., 2019. A deep convolutional neural network for localization of clustered earthquakes based on multistation full waveforms, *Seismol. Res. Lett.,* **90**(2A), 510–516.

Krischer, L., Megies, T., Barsch, R., Beyreuther, M., Lecocq, T., Caudron, C. & Wassermann, J., 2015. ObsPy: a bridge for seismology into the scientific Python ecosystem, *Comput. Sci. Discovery,* **8**(1), 014003.

LeCun, Y.A., Bottou, L., Orr, G.B. & Müller, K.R., 2012. Efficient BackProp, in *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science,* Vol. 7700, eds Montavon, G., Orr, G.B. & Müller, K.R., Springer.

Li, Z., Meier, M. A., Hauksson, E., Zhan, Z. & Andrews, J., 2018. Machine learning seismic wave discrimination: application to earthquake early warning, *Geophys. Res. Lett.,* **45**(10), 4773–4779.

Magrini, F., Jozinović, D., Cammarano, F., Michelini, A. & Boschi, L., 2020. Local earthquakes detection: a benchmark dataset of 3-component seismograms built on a global scale, *Artif. Intell. Geosci.,* **1,** 1–10.

Michelini, A., Faenza, L., Lanzano, G., Lauciani, V., Jozinović, D., Puglia, R. & Luzi, L., 2020. The new ShakeMap in Italy: progress and advances in the last 10 Yr, *Seismol. Res. Lett.,* **91**(1), 317–333.

Minson, S. E., Meier, M. A., Baltay, A. S., Hanks, T. C. & Cochran, E. S., 2018. The limits of earthquake early warning: timeliness of ground motion estimates, *Sci. Adv.,* **4**(3), eaaq0504.

Mousavi, S. M., Sheng, Y., Zhu, W. & Beroza, G. C., 2019. STanford EArthquake Dataset (STEAD): a global data set of seismic signals for AI, *IEEE Access,* **7,** 179 464–179 476.

Münchmeyer, J., Bindi, D., Leser, U. & Tilmann, F., 2020. The transformer earthquake alerting model: a new versatile approach to earthquake early warning, *Geophys. J. Int.,* doi:10.1093/gji/ggaa609.

Münchmeyer, J., Bindi, D., Leser, U. & Tilmann, F., 2021. Earthquake magnitude and location estimation from real time seismic waveforms with a transformer network, *Geophys. J. Int.,* ggab139. https://doi.org/10.1093/gji/ggab139.

Ochoa, L. H., Niño, L. F. & Vargas, C. A., 2018. Fast magnitude determination using a single seismological station record implementing machine learning techniques, *Geod. Geodyn.,* **9**(1), 34–41.

Otake, R., Kurima, J., Goto, H. & Sawada, S., 2020. Deep learning model for spatial interpolation of real-time seismic intensity, *Seismol. Soc. Am.,* **91**(6), 3433–3443.

Otović, E., Njirjak, M., Jozinović, D., Mauša, G., Michelini, A. & Štajduhar, I.: 2021 Intra-domain and cross-domain transfer learning for time series data - How transferable are the features?, *Knowledge-Based Systems* 107976 Elsevier

Pan, S. J. & Yang, Q., 2009. A survey on transfer learning, *IEEE Trans. Knowl. Data Eng.,* **22**(10), 1345–1359.

Saad, O. M., Hafez, A. G. & Soliman, M. S., 2020. Deep learning approach for earthquake parameters classification in earthquake early warning system, *IEEE Geosci. Remote Sens. Lett.,* **18,** 1293–1297. .

Satriano, C., Wu, Y. M., Zollo, A. & Kanamori, H., 2011. Earthquake early warning: concepts, methods and physical grounds, *Soil Dyn. Earthq. Eng.,* **31**(2), 106–118.

Spallarossa, D., Kotha, S. R., Picozzi, M., Barani, S. & Bindi, D., 2019. On-site earthquake early warning: a partially non-ergodic perspective from the site effects point of view, *Geophys. J. Int.,* **216**(2), 919–934.

Titos, M., Bueno, A., García, L., Benítez, C. & Segura, J. C., 2019. Classification of isolated volcano-seismic events based on inductive transfer learning, *IEEE Geosci. Remote Sens. Lett.,* **17**(5), 869–873.

van den Ende, M. P. & Ampuero, J. P., 2020. Automated seismic source characterization using deep graph neural networks, *Geophys. Res. Lett.,* **47**(17), e2020GL088690.

Wald, D. J., Quitoriano, V., Heaton, T. H., Kanamori, H., Scrivner, C. W. & Worden, C. B., 1999. TriNet "ShakeMaps": rapid generation of peak ground motion and intensity maps for earthquakes in southern California, *Earthq. Spectra,* **15**(3), 537–555.

Wilcoxon, F., 1992. Individual comparisons by ranking methods, in *Breakthroughs in Statistics,* pp. 196–202, eds Kotz, S. & Johnson, N.L., Springer.

Worden, C.B., Thompson, E.M., Hearne, M. & Wald, D.J., 2020. *ShakeMap Manual Online: Technical Manual, User's Guide, and Software Guide,* U. S. Geological Survey, http://usgs.github.io/shakemap/, doi:10.5066/F7D21VPQ.

Wu, D. J. & Datta, A., 2021. Continuous Weight Balancing, arXiv:2103.16591.

Zhang, X., Zhang, M. & Tian, X., 2021. Real-time earthquake early warning with deep learning: application to the 2016 M 6.0 Central Apennines, Italy earthquake, *Geophys. Res. Lett.,* **48**(5), 2020GL089394.

SUPPORTING INFORMATION

Supplementary data are available at *GJI* online.

Figure S1. Input example with the normalized traces for an $M = 4.4$ earthquake

Figure S2. Non-normalized input example for an $M = 4.4$ earthquake.

Figure S3. The CNN model results for $\log_{10}\text{PGA} > 2$ of the different sampling weights approaches used. The blue line represents the $\log_{10}\text{PGA}_{\text{true}} = \log_{10}\text{PGA}_{\text{predicted}}$ and the red, green and purple lines represent the regression lines for no weights model, model that used the maximum of the IM for the weights and the model that used the earthquake magnitude for the weights, respectively.

Table S1. The stations used in the study and some of their characteristics.

Table S2. IMs' residual statistics for the CNN predictions for the observed IMs (for the stations having recorded data) and for the ShakeMap predictions (for the stations that had no recorded data).

Table S3. IMs' residual statistics for the CNN predictions for the observed IMs (for the stations having recorded data) and the ShakeMap predictions (for the stations that had no recorded data).

Table S4. IMs' residual *R* statistics for the STEAD → CI experiment: the CNN predictions on the CI data set using TL from the pre-trained STEAD model.
Table S5. IMs' residual statistics for the CNN predictions for the observed IMs (for the stations having recorded data) and the ShakeMap predictions (for the stations that had no recorded data).
Table S6. IMs' residual *R* statistics for the STEAD → CW experiment: the CNN predictions on the CW data set using TL from the pre-trained STEAD model.
Table S7. IMs' residual statistics for the CNN predictions for the observed IMs (for the stations having recorded data) and the ShakeMap predictions (for the stations that had no recorded data).
Table S8. IMs' residual statistics for the CNN predictions.

Please note: Oxford University Press is not responsible for the content or functionality of any supporting materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the paper