17th World Conference on Earthquake Engineering, 17WCEE
Sendai, Japan - September 13th to 18th 2020

DEEP LEARNING TECHNIQUES FOR STRUCTURAL RESPONSE PREDICTION DURING STRONG GROUND MOTIONS

A. A. Torky[(1)], S. Ohno[(2)], T. Kashima[(3)]

[(1)] *Ph.D. Student, Tohoku University, ahmed.torky@dc.tohoku.ac.jp*
[(2)] *Associate Professor (Dr.Eng), International Research Institute of Disaster Science, Tohoku University, ohno@irides.tohoku.ac.jp*
[(3)] *Senior Research Engineer (Dr.Eng), Building Research Institute, Tsukuba, Japan, kashima@kenken.go.jp*

Abstract

In this paper, deep learning techniques are applied to predict a building's structural response to strong ground motions. Data from sensors near and inside structures measure accelerations during strong ground motions. The Building Research Institute (BRI) ANX building, an eight-story structure, has experienced major earthquakes since 1998. Sensors in the building provide and accumulate big data of historic events. Changes of the natural frequency of the ANX building from the big data is initially quantified. The time-series data of the historic events can be used to predict future response to future events using deep learning models rapidly. Although previous literature attempted similar time-series predictions, the building's multiple floors' accelerations were not predicted from base/ground acceleration. Accurate deep learning models could predict the relationship between ground/base motion and several floors' motions. Four different machine learning algorithms are implemented to predict multiple floor responses from ground accelerations.

The supervised AI algorithms used are: Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Levenberg–Marquardt Recurrent Neural Networks (LM-RNN). The neuron structures for all types of neural networks are explained. Big data management and filtering is necessary for importing accurate meaningful features to neural networks. A new proposed algorithm is presented for all learning models and training is performed with a new selection of hyperparameters. Firstly, the supervised learning techniques are trained with a window of **N**-past features. Subsequently, the models are deployed for inference on seismic events which the models have not been trained on. Parametric studies of windowing, down-sampling, batching, network structure, number of training epochs, and dropout rates are compared.

The different AI techniques' accuracies are compared. Although there are slight changes in the natural frequency of the BRI ANX building over time, the LSTM neural networks could still predict acceleration response of recent events accurately. Fourier Spectrum comparisons between the sensors' true acceleration signal and the neural network predicted acceleration signals are used as validation. Inference of trained models can be used to continuously predict multiple floor responses due to earthquakes.

Keywords: strong ground motion; structural acceleration response; time-series prediction; deep learning techniques; Fourier spectrum.

1. Introduction

Soft-computing techniques using artificial intelligence (AI) methods are now more reliable for engineering applications than ever before. This is possible by means of the current advancement of AI methods in recent years and by consistent data acquisition methods that have been active for the past few decades. When deployed in an insightful manner, supervised learning algorithms could now be employed to aid in structural engineering [1, 2]. It can also be used for rapid prediction earthquakes for early earthquake warning systems [3].

Machine learning is a branch of AI [4], and contains neural networks such as artificial neural networks (ANN). With the advancement of more robust parallel computing technologies, ANNs were architecturally modified to include more learning parameters and to accept more data, while minimizing efforts in feature extraction. Such networks perform deep feature extractions intrinsically, consequently termed deep neural networks (DNN). Some of the most notable DNNs that are used for time-series forecasting are recurrent neural networks (RNN) [5] and long short-term memory neural networks (LSTM) [6].

Structural dynamic analysis is usually performed using incremental numerical methods [7] for system identification. Incremental methods of high-fidelity structures could require finite element (FE) models of the structure that could include thousands or millions of finite elements. With previously recorded time-dependent loads, the FE model can be executed to forecast the dynamic reaction of a structure. An engineer could use this technique several times until the FE model converges to perform a realistic response. As the method requires several iterations and FE models consume computing power and time, it is not feasible to use such methods for real-time dynamic response prediction.

Whole structures' responses to seismic activity could be predicted using RNNs, as a means of response emulators that are accurate and less computationally expensive than FEM models. Real-time prediction is necessary for system identification and damage identification. Deep learning models can be trained for several hours or days, however, inference on reliable trained models can be performed at fractions of a second. There is a need for a more refined method for time-series prediction from acceleration data instead of error-prone computed displacement data. Previous research have used LSTMs [8] for response prediction of one floor from ground movement, however case studies presented were of short buildings. It is necessary to ascertain that LSTMs can capture nonlinear behaviors of structures, torsional movement, and complex mode shapes of medium to high-rise buildings. It is also essential to visualize the building's predicted dynamic behavior and frequencies from the predicted time-series' Fourier spectrum.

The organization of this paper continues with Section 2, which briefly describes the machine/deep learning algorithms that could be used for time-series prediction. Section 3 proposes the new algorithm for structural acceleration response prediction using the defined AI algorithms. Section 4 describes the BRI annex building case study and seismic events characteristics. The proposed algorithm is initially tested on the case study using four methods, and then optimized using the preferred technique in Section 5. The paper is then concluded with discussions on the outcomes and recommendations for future work.

2. ANN and DNN Theory

There are several machine learning techniques that excel at temporal sequence prediction, particularly for supervised learning tasks. Of which is the Levenberg–Marquardt algorithm based RNN (LM-RNN) as a machine learning method, and DNN, RNN, and LSTM as deep learning methods. Machine learning neural networks typically have weights and biases defined in neurons that are optimized through backpropagation. Deep learning methods have evolved to include new optimizers and activation functions [9]. They deal with big data better as feature selection is handled by the technique, while users focus on selecting useful data that represents their case. Deep learning models have the capability of understanding nonlinear relationships between features and targets. A simplified illustration of the structures of each algorithm is in Fig. 1. A brief description of each network's neuron is given in the subsections below.

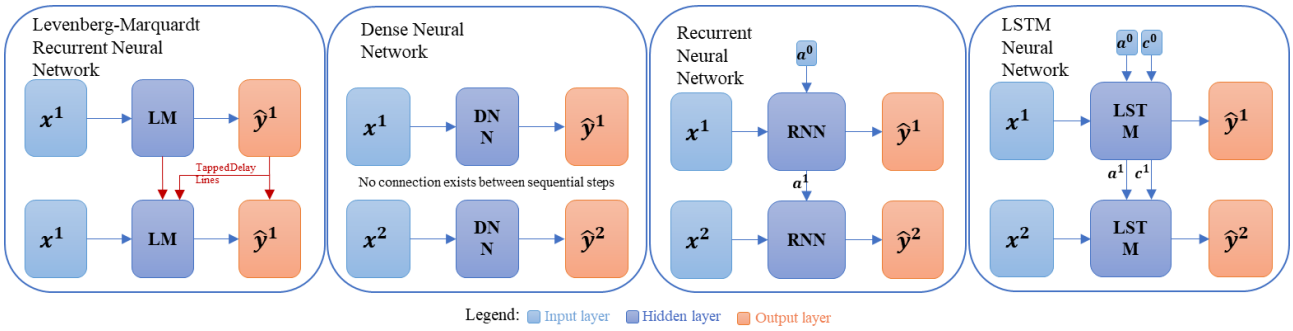


Fig. 1 – Simplified differences between networks. For simplicity, networks are illustrated in the case that they include just one hidden layer and two time-steps.

2.1. Levenberg–Marquardt based RNN (LM-RNN)

LM-RNN [10] is an RNN in continually sampled time trained to produce outputs that converge to targets through backpropagation with a second order optimization method called the Levenberg–Marquardt algorithm, which is a least-squares variant. LM-RNNs are considered as shallow networks, do not include the large variety of activation functions and optimizers available for deep learning techniques, and training takes a considerable amount of time as they are implemented on CPUs. Each neuron of the network could receive information from inputs, previous layers, and previous time steps. Receiving information from previous temporal sequences adds recurrent time-delayed connections called tapped delay lines. Equations of this network's neurons are:

$$\widetilde{IW}_i^t = W_{ix}x^t \tag{1}$$

$$\widetilde{LW}_i^{t-n} = W_{ix}a_i^{t-n} \tag{2}$$

$$a^t = \tanh(\widetilde{IW}^t + \widetilde{LW}_i^{t-n} + \sum\widetilde{LW}_j^{t-n} + b_i) \tag{3}$$

$$\hat{y}^t = a^t \tag{4}$$

Where x^t, W_{ix}, b_i, and a_i^t are inputs to a neuron, weights of the current neuron, bias of the current neuron, and output of the current neuron after applying an activation function tanh, respectively. $\widetilde{LW}_i^{t-n}$ and $\widetilde{LW}_j^{t-n}$ are the coefficients from previous n delayed steps of the current *i* neuron and subsequent *j* neurons. $\hat{y}^t$ is the output of the RNN at a temporal point.

2.2. Dense neural networks (DNN)

DNNs are feed-forward fully connected/dense networks that include weights and biases, as described by Eq. (5), optimized through backpropagation. Back-propagation includes optimizers such as stochastic gradient descent and Adam. Activation functions σ could be tanh or rectified linear unit (ReLU) and provide nonlinear characteristics to output $\hat{y}^t$. Typical DNNs contain high number of neurons in each layer with several layers. These optimizers and activation functions are available in RNNs and LSTMs as well. DNNs do not have connections with subsequent temporal steps.

$$\hat{y}^t = \sigma(W_{yx}x^t + b_y) \tag{5}$$

2.3. Recurrent neural networks (RNN)

RNNs are basically DNNs with temporal connections between their internal states. Eq. (6) displays that the output of a hidden layer a^t is a function of previous temporal steps' states in addition to the current step. The activation function is applied to the weighted output of the neuron cell before becoming $\hat{y}^t$ in Eq. (7). A problem that occurs often with RNNs is the vanishing gradient problem, which is relevant in seismic response as waveforms are dampened and lose energy. This makes RNNs suitable for harmonic and periodic time-series.

$$a^t = \tanh(W_{ax}x^t + W_{aa}x^{t-1} + b_a) \tag{6}$$

$$\hat{y}^t = \sigma(W_{ya}a^t + b_y) \tag{7}$$

2.4. Long short-term memory neural networks (LSTM)

LSTMs are advanced forms of RNNs with more gates to overcome the vanishing gradient problem, which are the forget gate in Eq. (8) and the update gate in Eq. (9). These gates allow LSTMs to forget temporary misleading influences from inputs and previous states, which are computed from Eq. (10) and Eq. (11), while maintaining and updating useful recurring information. The output gate in Eq. (12) could now be computed, whereas $\hat{y}^t$ is computed from the outcome of Eq. (13). Thus, LSTMs could maintain "memories" across thousands or millions of discrete time steps, which can be useful for subsequent earthquake events.

$$\Gamma_f^t = \sigma(W_f x^t + W_f a^{t-1} + b_f) \tag{8}$$

$$\Gamma_u^t = \sigma(W_u x^t + W_u a^{t-1} + b_u) \tag{9}$$

$$\tilde{c}^t = \tanh(W_c x^t + W_c a^{t-1} + b_c) \quad (10)$$

$$c^t = \Gamma_f^t \cdot c^{t-1} + \Gamma_u^t \cdot \tilde{c}^t \quad (11)$$

$$\Gamma_O^t = \sigma(W_O x^t + W_O a^{t-1} + b_O) \quad (12)$$

$$a^t = \Gamma_O^t \cdot tanh(c^t) \quad (13)$$

3. Sliding-Window Method for Structural Response Prediction

To be able to predict multiple responses of sensors' channels, it is recommended to attain an N sequence of ground accelerations first. This N sequence could be of accelerometers located at the ground near the building site or in the lowest basement of the building at the interface with the underlying soil. A "window" of the N sequential accelerations could provide AI models with enough features to explain the cyclic or seasonal trend in the input data, making it an easier task for the model to predict the relationship between an input window and target channels' accelerations on the building. Such a window could "slide" incrementally along multiple seismic ground motion time-series while training the supervised learning model, as illustrated in

Fig. 2. It is preferable to have multi-channel targets, as these outputs could give more confidence to the training process.

It is a challenge to choose the optimum "sliding window" size N, thus the window size could be considered as a hyperparameter that needs to be investigated during multiple iterations of training and testing models. In addition, it is common that accelerometer data is provided at a sampling frequency of 100Hz. High sampling frequencies will include excessive data and as window sizes remain short, it will be a challenge and time consuming for the model to gain knowledge on trends across multiple windows. Therefore, down-sampling is also added to the list of changeable hyperparameters during model training iterations.

Other hyperparameters that can be optimized are training data batches, number of epochs for training the model, number of hidden layers and their neurons, and the dropout rate of neurons in each layer. Severely increasing the number of batches during one training epoch could unnecessarily increase the computing time of backpropagation, whereas decreasing them to the minimum could yield unreasonable accuracy. The number of hidden layers with their neurons is also a difficult task to narrow down, however, dramatically increasing their numbers could result in excessive inference time.

The proposed "sliding window" technique with all the mentioned parameters are integrated into an algorithm, as demonstrated in Fig. 3, for training multiple iterations of models using the previously mentioned supervised learning techniques in Section 2. The brief flow of the algorithm is as follows:

1. Past earthquake event data are selected to satisfy the minimum PGA requirement, and specific accelerometers' data from these events are imported.
2. The seismic data time-steps length is unified between all events, then passed through a Butterworth bandwidth filter to eliminate noise. Ground acceleration channels are allocated as input data, while the multi-channel targets are set as outputs. Data is then split to training and testing data.
3. Hyperparameters are identified (e.g. window size, down-sample ratio, number of layers, etc…), then discrete ranges are set to iterate on them.
4. Complete training and testing processes are performed on each model iteration, including plots of predicted waveforms.
5. Iterations of training are looped on all combinations of the hyperparameters assigned.
6. For validation, predicted time-series and their Fourier magnitude spectrum are visualized against their true counterparts. Also, the mean squared error (MSE) of each model is calculated.

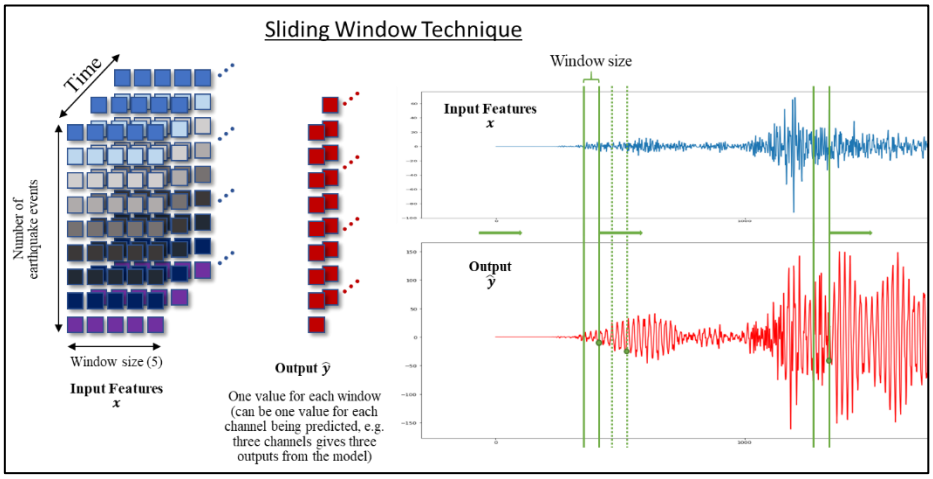


Fig. 2 – Sliding window technique for multi-quake input and multi-channel target.

Such an algorithm could be used for existing buildings that have accelerometers installed in them. This modeling method could also be used for soil-structure interaction seismic predictions of buildings on multilayered soil. The incremental stride or "slide" in this study is of just one incremental time-step between a window and the subsequent one. The algorithm is converted to a Python program.

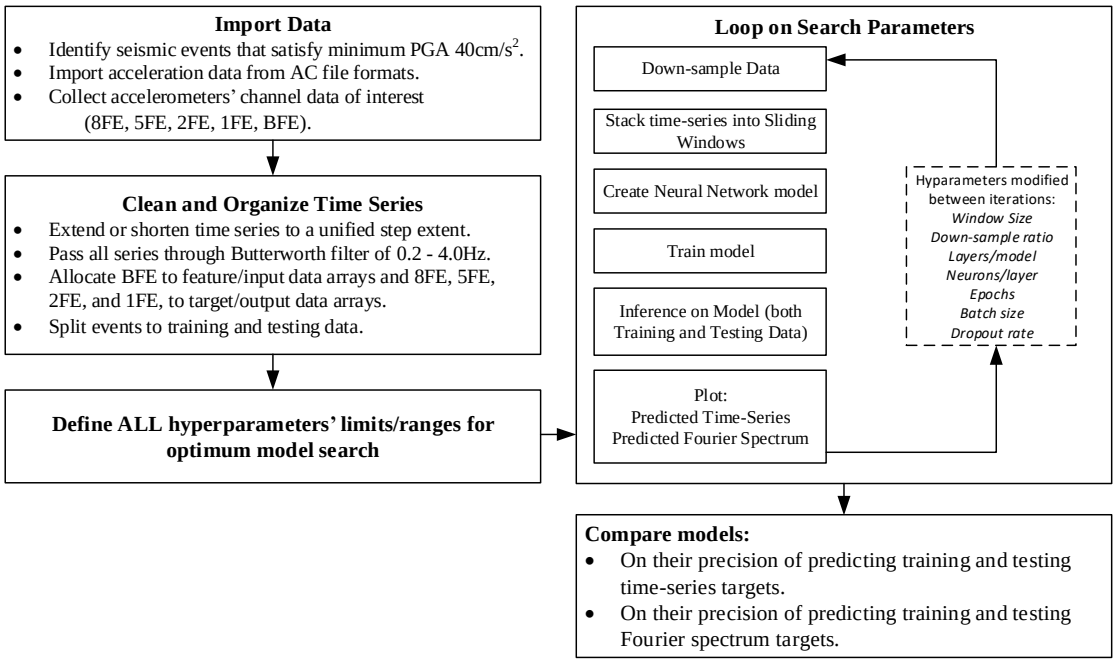


Fig. 3 – Algorithm for training supervised learning models with the sliding window technique.

4. Case Study – BRI Annex Building

The case study focuses on the intriguing medium-rise annex building of the Building Research Institute (BRI) of Japan. It is an eight-story steel reinforced concrete structure, with one basement below ground level. It is in Tsukuba, Japan and linked to the main BRI building with pathways. The main and annex structures are separated with expansion joints. More structural details and the soil profile are described in [11].

For strong motion observation, the BRI strong motion network includes accelerometers in both the main and annex buildings. There are eleven accelerometers in the annex building which have three channels each for translations in the North-South (NS) component, East-West (EW) component, and Up-Down (UP) component. The accelerometers' distribution is shown in Fig. 4. Time-series data have been accumulated and

stored from these accelerometers for more than 20 years. Of these accelerometers, five are placed at the east side of the building (BFE, 1FE, 2FE, 5FE, 8FE), with sampling frequency of 100Hz each. An interesting fact of the annex building is that torsional vibration and mode shapes are evident from accelerometers' results.

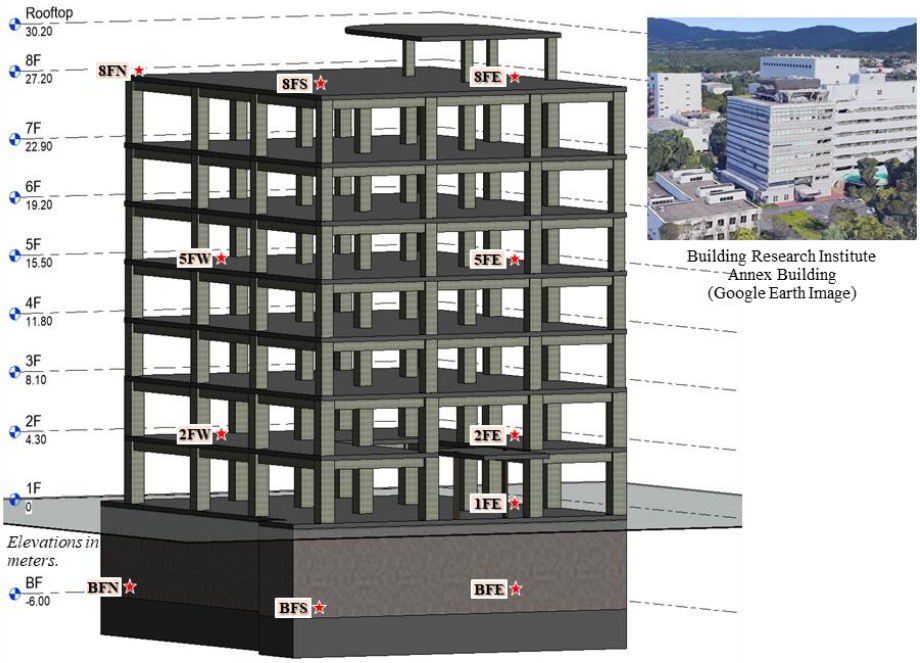


Fig. 4 – 3D distribution of the sensors on skeletal view of BRI Annex building.

Moderate to strong ground motion events that affected the annex building site were selected for this study. The database of all events is accessible from [12], where the time history of all the annex accelerometers can be requested in addition to other data from the BRI Network. The events that included at least maximum peak ground accelerations (PGA) of 40cm/s^2 could provide enough data for training deep learning time-series models. Thirty-one seismic events satisfied such a filter. These strong ground motions could be obtained from the BRI Network database by choosing the station name as "Annex and Main Bldg., Building Research Institute". Channel accelerations were subjected to Butterworth bandwidth filtering between 0.2Hz and 4.0Hz, as the natural frequencies of the structure are within that range and high frequencies were eliminated to reduce any noise in the readings. In general, more events that included PGA greater than 40 cm/s^2 occurred after the 2011 Tohoku-oki Earthquake than before that event.

To visualize the influence of the 2011 Tohoku-oki earthquake [13], Fast Fourier Transform was performed on recorded time-series. All Fourier spectra and spectral ratios were smoothened with a Parzen window of 0.2 Hz. To determine the first natural frequency of the building, the spectral ratios 8FE/BFE for the NS and EW directions are plotted in Fig. 5. The first natural frequency in both directions could be estimated at around 1.0 Hz. It is not easy to identify the spectrum amplitude peak of the torsional frequency, as previously identified in [11], however, a minor peak is noticeable at 1.8Hz. Spectral ratios 5FE/BFE are also displayed with amplitude peaks at 1.0 Hz. Phase changes are estimated between 1.5 Hz to 3.5 Hz, where notable phase turning appears only in spectral ratios of 5FE/BFE. This could be a higher order mode of vibration or second mode of vibration as accelerometer 5FE is placed at mid-height of the building.

In past research, the annex building's first natural frequencies have varied in time considerably [14]. To briefly demonstrate the change, dynamic characteristics were compared for six earthquake events. These included the Tohoku-oki event, two events prior and three subsequent events. It is evident from Fig. 6 that the first natural frequency of the building has slightly shifted to a lower frequency; Table 1 quantifies the values. The decrease in natural frequency was further investigated in [14], however, for this study it is interesting to explore if the annex building could be represented by a trained model from a multitude of different events.

Table 1 – Fourier Spectral Ratio (t=0-167s, Parzen Window 0.2Hz). Peaks between 0.101Hz and 4.999Hz.

Date of record	Ratio	Peak Amplitude Ratio	Peak Freq. (Hz)
11/22/2016 5:59	180-8FE/180-BFE	7.944	0.952 Hz
12/7/2012 17:18	180-8FE/180-BFE	10.38	0.937 Hz
4/16/2011 11:19	180-8FE/180-BFE	9.661	0.922 Hz
3/11/2011 14:46	**180-8FE/180-BFE**	**7.954**	**1.053 Hz**
8/20/2008 15:13	180-8FE/180-BFE	6.673	1.691 Hz
10/6/2004 23:40	180-8FE/180-BFE	13.14	1.379 Hz

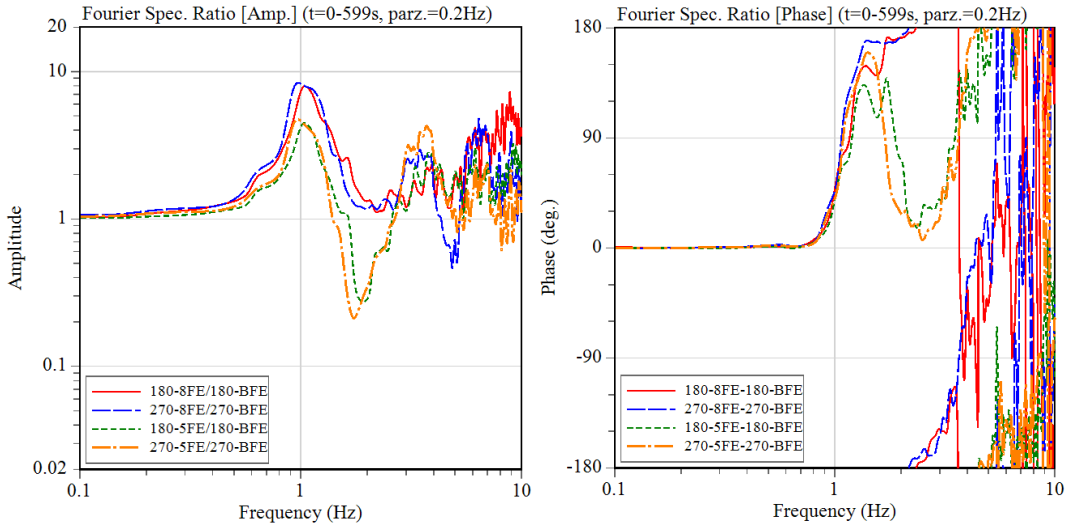


Fig. 5 – Fourier spectral ratio 8FE/BFE and 5FE/BFE for the 2011 Tohoku-oki event (using ViewWave).

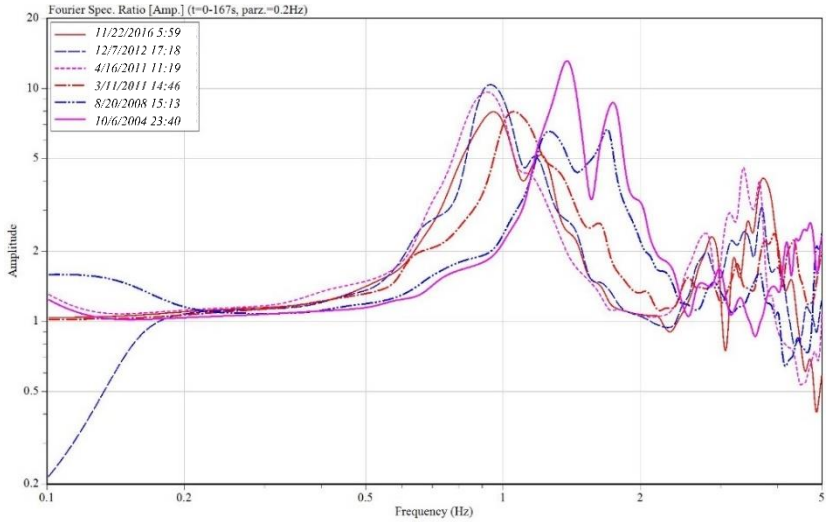


Fig. 6 – Fourier spectral ratio between 180-8FE and 180-BFE for multiple earthquakes events.

5. Training and testing the models

The first experiment compares the supervised learning techniques mentioned in Section 2. The second experiment focuses on obtaining accurate LSTM neural networks. The desktop and workstation used had the following specifications of Intel Xeon CPUs @2.60GHz (16 and 64 cores respectively), and NVIDIA Geforce

RTX 2080Ti GPU with 1545 MHz (one and four respectively). Python library PYRENN [15] was used to simulate LM-RNN, whereas the GPU-version of Google's Python library Tensorflow 2 [16] was used to simulate DNN, RNN, and LSTM. To simplify, the reliable activation function ReLU and optimizer Adam were used, as they previously displayed good performance [9]. Time-series data were scaled between -1 and 1 before training, and inversely scaled after model predictions.

5.1. Comparing between the AI algorithms

This experiment is a minimal study on predictions produced from the four neural network methods. Time-series data of the NS components of BFE, 8FE, 5FE, 2FE, and 1FE are down-sampled to 20 Hz initially. Then the sliding window size of BFE is set to 20 steps. DNN and LM-RNN models' neural network structures are set to {20,40,20,4}. RNN and LSTM models included an extra dense layer of 20 neurons before the output to become {20,40,20,20,4}. Thirty events were used for training with 20,000 epochs and batch size of 100. The event that occurred *off Sanriku* on *12/7/2012 at 17:18* was used for testing. Time-series and Fourier spectrum prediction results for 8FE (NS) only are displayed in Fig. 7, the most challenging channel to predict.

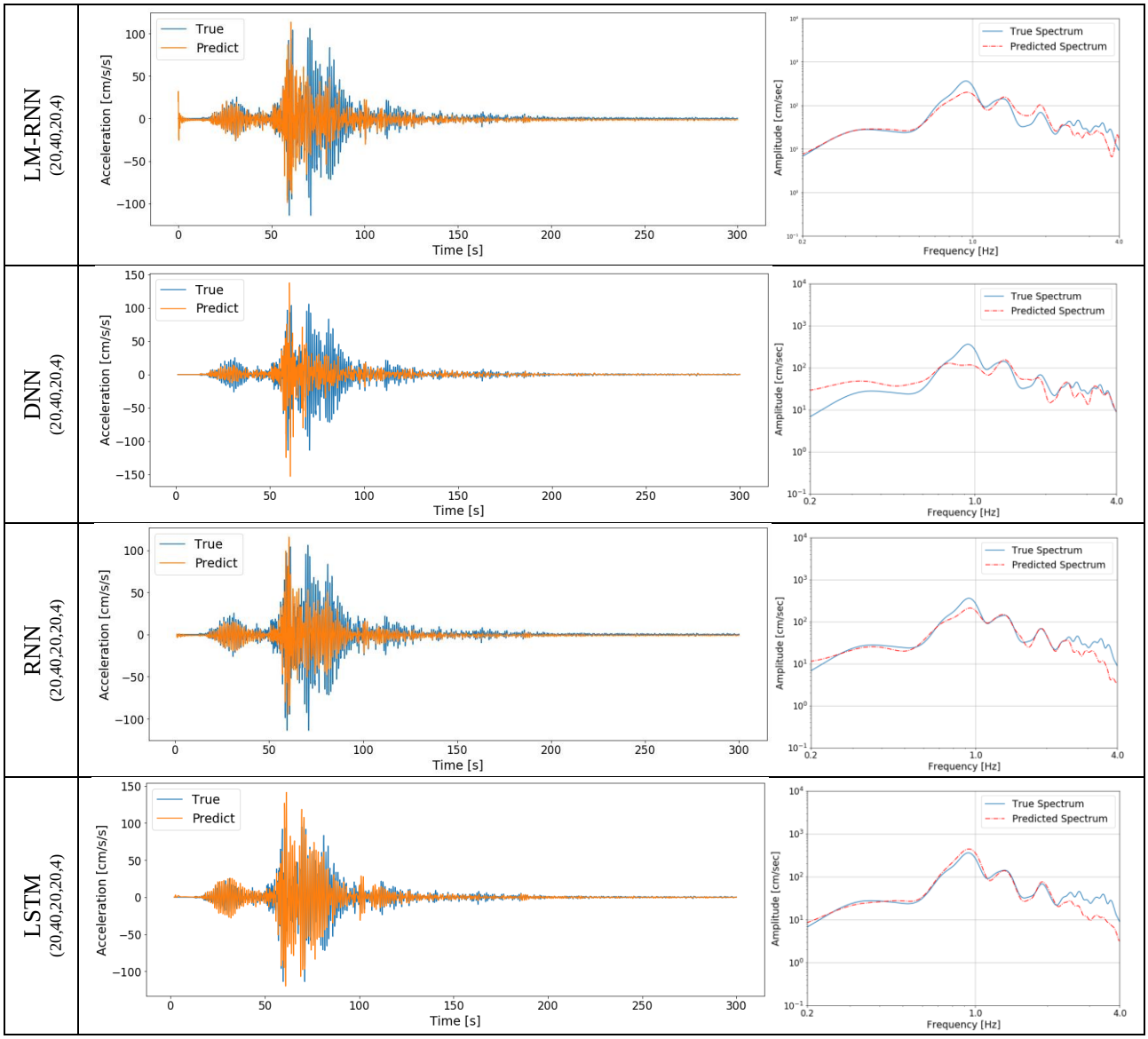


Fig. 7 – Predicted time-series (left) and their Fourier spectrum (right) for 8FE (NS).

The trainable parameters for the DNN model were 2,164 and took 0.2 hours to train on the GPU, for the RNN model were 4,164 and took 76.9 hours to train on the CPU, for the LM-RNN model took 273.8 hours to

train on the CPU, and for the LSTM model were 15,144 and took 4.43 hours to train on a GPU. Both the RNN and LM-RNN libraries only have implementations on the CPU. From this minimal study it is apparent from the time-series and Fourier spectra in Fig. 7 that although all models had similar layers/neurons structures, LSTM networks have better capabilities at learning structural response relationships with base/ground accelerations. Thus, it is more convenient to optimize the LSTM networks in the coming subsection.

5.2. Optimizing LSTM models

The following experiment focuses on finding the optimum LSTM network model for the case study. Similar to the previous experiment, time-series data of the NS and EW components of (BFE, 8FE, 5FE, 2FE, 1FE) are utilized. Table 2 provides the selection of hyperparameters' grid search values, which are implemented in iterations of LSTM networks for the NS components and EW components separately. Again, all events were used to train models, whereas the event that occurred *off Sanriku* on *12/7/2012 at 17:18* were used for testing.

Table 2 – Hyperparameters for LSTM models and their ranges.

Hyperparameter	Selection	Hyperparameter	Selection
Sliding Window Size	10, 20 (steps)	Epochs	1000, 5000, 10000, 20000
Down-sampling ratio	1/5 (20Hz), 1/10 (10Hz)	Batch size	10, 100, 1000, 5000
Layers/model	1, 2	Dropout rate	0, 0.1, 0.2, 0.5
Neurons/layer	40, 100, 200, 400	Activation Function	ReLU

Eight LSTM network models for the NS and EW components with minimum test errors are presented in Table 3 and Table 4, where the neural networks are compared according to training and test errors. The results are listed in ascending order for MSE of the testing data. For the NS component models, a window size of 10 and down-sampling from 100Hz to 20Hz seem to best method to represent inputs. The most common number of neurons in the first hidden layer is 400 neurons. It is evident that having a dropout rate between 0.2 and 0.5 is favorable. It could be noticed that if batch sizes are as small as 10, small number of epochs such as 5,000 tend to converge to the same testing errors of large batch sizes with 20,000 epochs.

Table 3 – List of the most optimum models for NS components.

	Window	Down-sample	Epochs	Batch size	Hidden Layers	Dropout	Hours	Train Error (MSE)	Test Error (MSE)
1	10	20Hz	20000	5000	200, 100, 50, 50	0.2	11.96	1.83E-05	0.00044
2	20	20Hz	40000	5000	400, 200, 100, 100	0.4	30.19	9.74E-06	0.00050
3	10	20Hz	20000	5000	100, 50, 25, 25	0.2	4.84	3.66E-05	0.00055
4	10	20Hz	20000	100	400, 200, 200	0.5	3.58	1.01E-05	0.00056
5	10	20Hz	20000	10	400, 200, 200	0.5	5.99	5.56E-06	0.00057
6	10	20Hz	5000	10	40, 20, 20	0.5	1.19	0.00012	0.00068
7	10	20Hz	20000	10	40, 20, 20	0.5	4.79	9.21E-05	0.00078
8	20	20Hz	20000	5000	200, 100, 50, 50	0.2	12.93	2.13E-05	0.00079

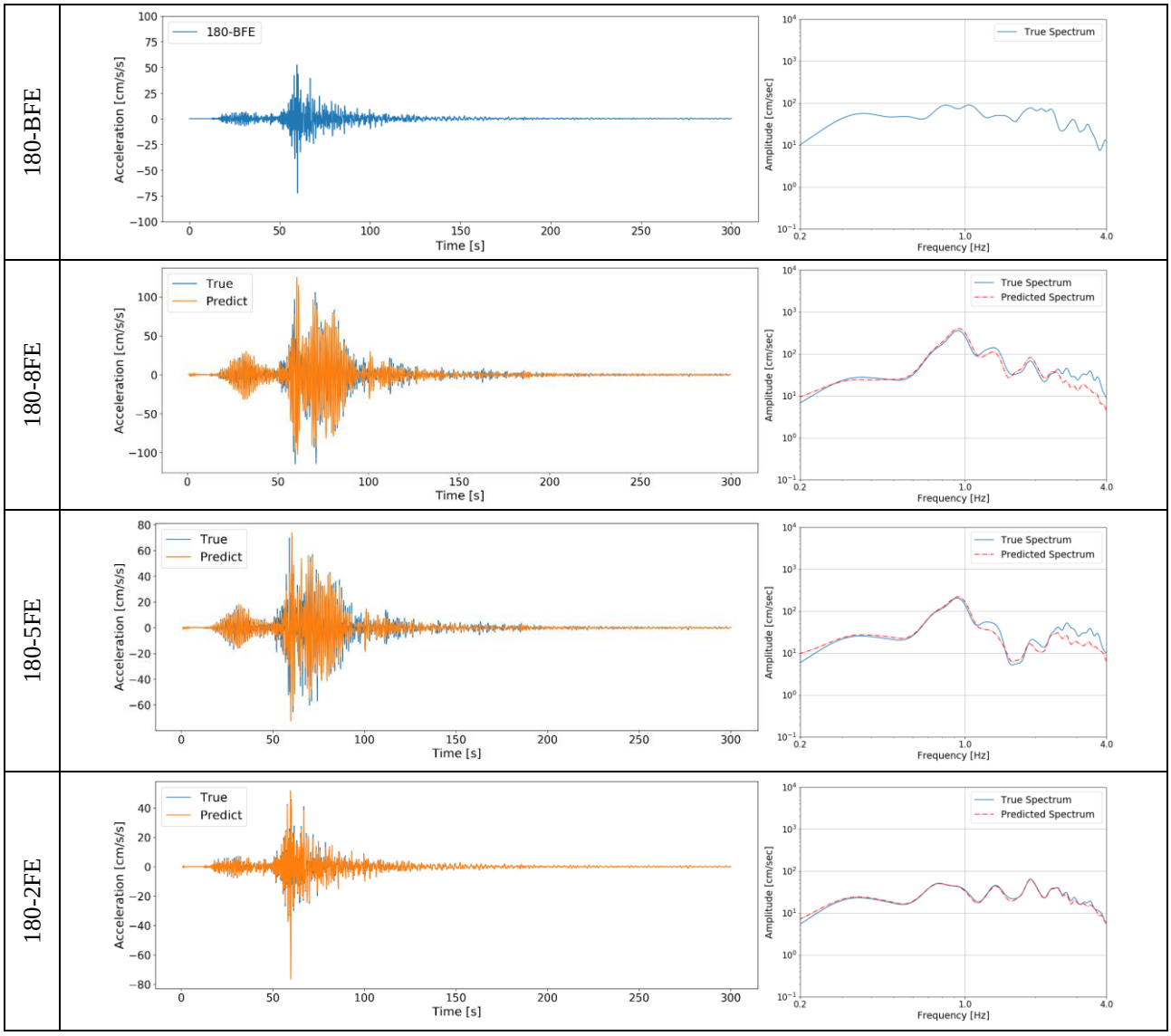
Table 4 – List of the most optimum models for EW components.

	Window	Down-sample	Epochs	Batch size	Hidden Layers	Dropout	Hours	Train Error (MSE)	Test Error (MSE)
1	10	20Hz	20000	10	400, 200, 200	0.1	3.35	1.82E-05	0.000128
2	10	10Hz	20000	100	200, 100, 100	0.5	1.87	2.38E-05	0.000129
3	10	10Hz	20000	10	400, 200, 200	0.1	1.85	1.16E-05	0.000153
4	20	20Hz	20000	100	400, 200, 100, 100	0.5	4.89	1.09E-05	0.000159
5	10	10Hz	20000	5000	200, 100, 100	0.5	0.37	0.000285	0.000165

6	10	20Hz	20000	100	200, 100, 100	0.5	3.29	8.21E-05	0.000183
7	20	20Hz	20000	100	400, 200, 200	0.5	3.59	1.22E-05	0.000187
8	10	10Hz	10000	100	200, 100, 100	0.5	0.93	6.10E-05	0.000187

According to the EW component models, a window size of 10 and down-sampling from 100Hz to either 20Hz or 10Hz best represents the inputs. The most common number of neurons in the first hidden layer is also between 400 and 200 neurons. It is also evident that having a dropout rate of 0.5 usually produces a reasonable model. In EW component models, batch sizes of 10 and 100 more commonly give the least testing errors.

To visualize the testing prediction results, the first model from each component is presented in Fig. 8 for the NS component and in Fig. 9 for the EW component. The optimum models appear to produce near-perfect estimations of the 2FE and 1FE time-series data. Performance of 8FE and 5FE are satisfactory but it is apparent from the predicted Fourier spectrum that peaks of frequencies are accurate. The main recurring similarity in the best network configurations between the NS and EW components is that a sliding window size of 10 and a sampling rate of 20Hz are the best formation of the input features. Also, these inputs appear to require at least 200 LSTM neurons in the first hidden layer of the networks to describe the features and at least 20,000 epochs for training.



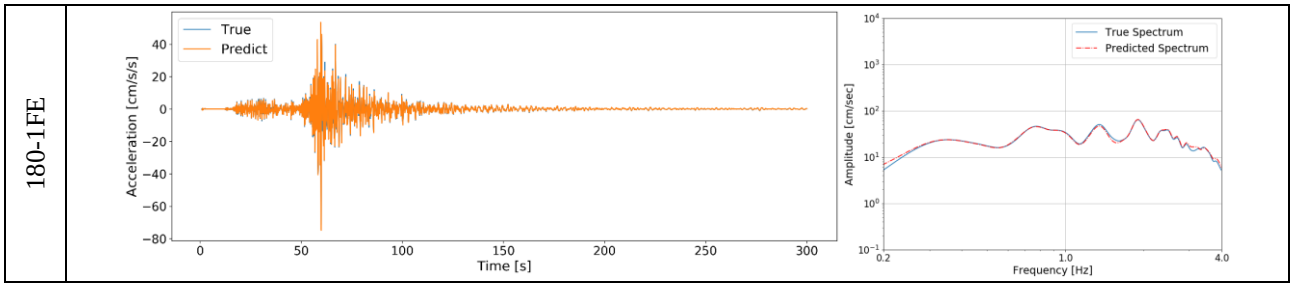


Fig. 8 – Input and predicted time-series (left) and their Fourier spectrum (right), for the NS components.

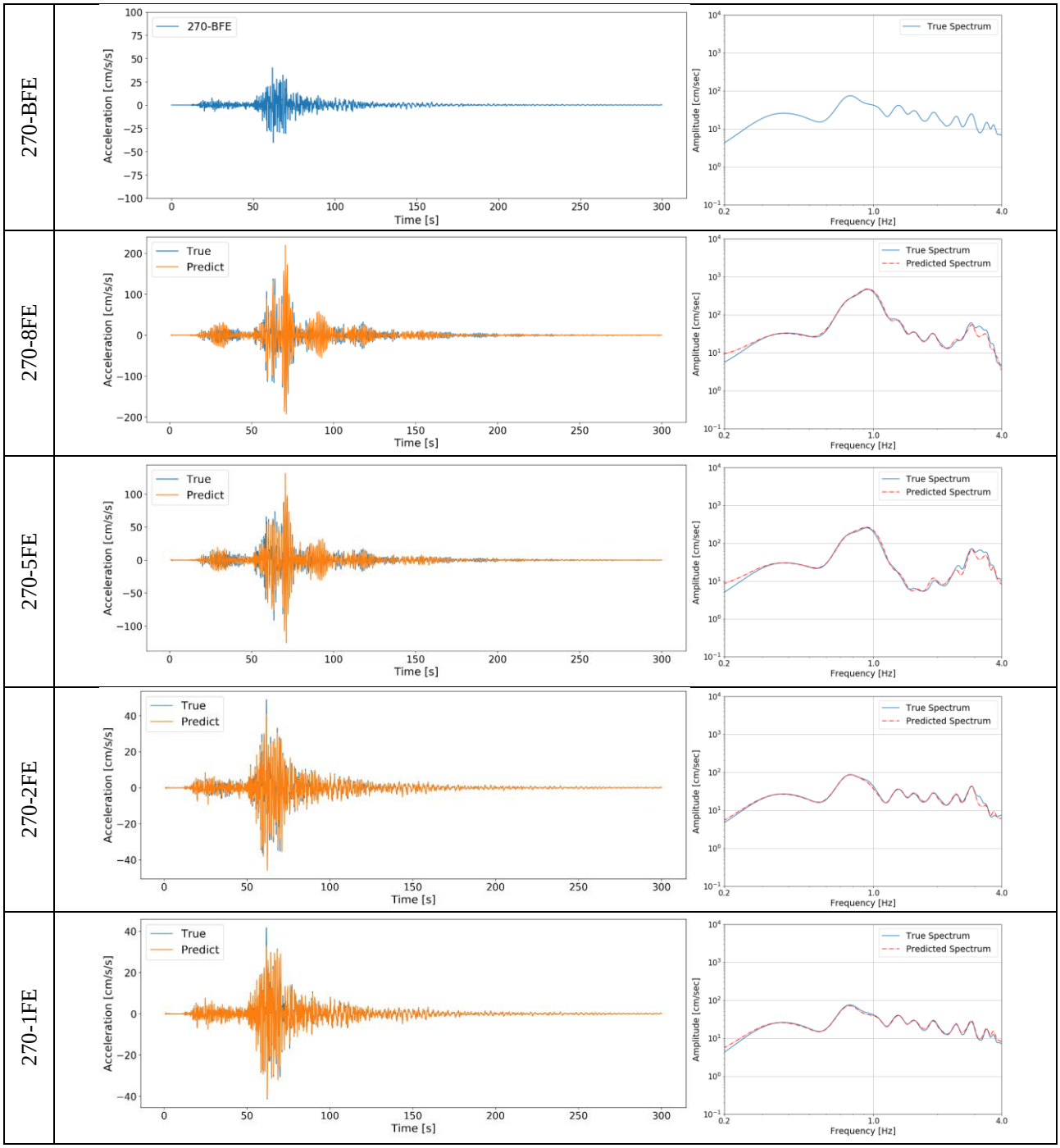


Fig. 9 – Input and predicted time-series (left) and their Fourier spectrum (right), for the EW components.

6. Conclusions and Future work

Deep learning techniques could be applied to predict the nonlinear dynamic response of structures during strong ground motions. To produce adequate deep learning models that can perform such a task, the sliding window algorithm for multiple-channel prediction is proposed. Four different supervised learning techniques are implemented into the proposed algorithm to predict multiple floors' responses from base accelerations. To validate the proposed algorithm, accelerometer data from the BRI annex building were used. Past seismic events of at least PGA 40cm/s^2 are identified. Through the first experiment, LSTM neural networks outperformed other supervised learning techniques at estimating the structural response time-series and Fourier spectra for testing data. Then, the second experiment found optimum LSTM neural networks that can predict multiple accelerometers' channels from base accelerations. Possible continuation to the current study could include investigation of LSTM-based networks such as 2D LSTM and CONV2DLSTM. These recently developed techniques could import multi-component inputs from multiple accelerometers of the building. Multi-component input features could better represent seismic coupling effects between orthogonal components. Also, a real-time online system can be developed to continuously train and predict a building's dynamic response, which could be advantageous for the development of structural health monitoring systems combined with earthquake early warning systems. The authors intend to proceed with the above studies.

7. Acknowledgements

The authors would like to acknowledge the support provided by the Building Research Institute (Japan) for providing strong motion records of the BRI annex building. This study was supported by JSPS KAKENHI Grant number 19K22002.

8. References

[1] Salehi H, Burgueño R (2018): Emerging artificial intelligence methods in structural engineering. *Engineering Structures*, **171**, 170–189.
[2] Torky AA, Aburawwash AA (2018): A Deep Learning Approach to Automated Structural Engineering of Prestressed Members. *International Journal of Structural and Civil Engineering Research*, **7**, 347–352.
[3] Kuyuk HS, Susumu O (2018): Real-time classification of earthquake using deep learning. *Procedia Computer Science*, **140**, 298–305.
[4] Goodfellow I, Bengio Y, Courville A (2016): *Deep Learning.* MIT Press, http://www.deeplearningbook.org/.
[5] Caterini AL, Chang DE (2018): *Deep Neural Networks in a Mathematical Framework: Recurrent neural networks. SpringerBriefs Computer Science*, Springer International Publishing, 59-79.
[6] Hochreiter S, Schmidhuber J (1997): Long Short-Term Memory. *Neural Computation*, **9** (8), 1735–1780.
[7] Bathe KJ, Ramm E, Wilson EL (1975): Finite element formulations for large deformation dynamic analysis. *International Journal Numerical Methods in Engineering*, **9**, 353-386.
[8] Zhang R, Chen Z, Chen S, Zheng J, Büyüköztürk O, Sun H (2019): Deep long short-term memory networks for nonlinear structural seismic response prediction. *Computers and Structures*, **220**, 55-68.
[9] Lee S, Ha J, Zokhirova M, Moon H, Lee J (2018): Background Information of Deep Learning for Structural Engineering. *Archives of Computational Methods in Engineering*, **25** (1), 121-129.
[10] Fu X, Li S, Fairbank M, Wunsch DC, Alonso E (2015): Training Recurrent Neural Networks with the Levenberg-Marquardt Algorithm for Optimal Control of a Grid-Connected Converter. *IEEE Transactions on Neural Networks and Learning Systems*, **26** (9), 1900-1912.
[11] Kashima T (2004): Dynamic Behavior of an Eight-Storey SRC Building Examined from Strong Motion Records. *13th World Confonference on Earthquake Engineering*, Vancouver, Canada.
[12] BRI Strong Motion Observation (English). https://smo.kenken.go.jp/.
[13] Kashima T, Koyama S, Okawa I (2012): Strong Motion Records in Buildings from the 2011 off the Pacific coast of Tohoku Earthquake. *Building Research Data No.135*, Building Research Institute.
[14] Kashima T, (2014): Dynamic Behaviour of SRC Buildings damaged by the 2011 Great East Japan Earthquake based on Strong Motion Records. *Second European Conference of Earthquake Engineering and Seismology*, Istanbul, Turkey.
[15] PYRENN. https://pyrenn.readthedocs.io/en/latest/.
[16] TensorFlow 2. https://www.tensorflow.org/.

