Prediction of Nonlinear Drift Demands for Buildings with Recurrent Neural Networks

K. Kocamaz[1], B. Binici[2], K. Tuncay[3]

[1]Department of Civil Engineering, METU, Ankara, Turkey, kkocamaz@metu.edu.tr
[2]Department of Civil Engineering, METU, Ankara, Turkey, binici@metu.edu.tr
[3]Department of Civil Engineering, METU, Ankara, Turkey, tuncay@metu.edu.tr

Abstract

Application of deep learning algorithms to the problems of structural engineering is an emerging research field. In this study, a deep learning algorithm, namely recurrent neural network (RNN), is applied to tackle a problem related to the assessment of reinforced concrete buildings. Inter-storey drift ratio profile of a structure is a quite important parameter while conducting assessment procedures. In general, procedures require a series of time consuming nonlinear dynamic analysis. In this study, an extensive RNN is trained to tackle these problems and provide a simple tool for assessment. Aim of the study is to predict the non-linear drift demand along the height of a structure by employing RNN for a given stiffness profile along the height, strength reduction coefficient, mass density on a floor, number of storeys. In order to train the network, a large number of nonlinear time history analyses are conducted for synthetically created building models. It is shown that RNN is able to accurately predict nonlinear drift demand profile of a structure along height without conducting tedious time history analyses. Therefore, the trained RNN can serve as a drift demand estimation tool, significantly shortening the assessment procedure.

Keywords: *recurrent neural networks, drift prediction, structural design*

1 Introduction

Application of machine learning algorithms (perceptron, decision trees etc.) and deep learning algorithms (recurrent neural networks, convolutional neural networks, generative adversarial networks etc.) to the problems of structural engineering field is newly emerging research field among structural engineers. In this study, a deep learning algorithm, namely recurrent neural networks, is applied to shorten the assessment procedure of existing structures.

For assessment procedures, inter-storey drift ratio is an important factor affecting decision making process. Assessment procedures can be categorized as qualitative and quantitative procedures. Quantitative procedures can be further categorized as model-based and measurement-based procedures. Measurement-based methods aim to assess a building by using measurements that are representation of structural behavior and seismicity of interest. On the other hand, model-based procedures often require either equivalent elastic or nonlinear analysis (Akpinar et al., 2020). In most assessment procedure provisions, including the Turkish Code, it is required to conduct equivalent elastic analysis for low-rise structures. However, a series of nonlinear dynamic analyses are necessary to estimate drift demand for mid-rise buildings. These analyses demand advanced modelling techniques that are labor intensive and time consuming.

Aim of the present study is to predict the nonlinear drift demand for each storey of an existing building accurately by using neural networks without conducting time consuming nonlinear dynamic analyses. A large number of nonlinear dynamics analyses were conducted to generate training data for the neural network. 4 distinct geographic locations and 2 different soil types were selected for seismicity. 22 ground motions were scaled accordingly. These ground motions were applied to randomly generated structures in order to form a large dataset for training.

Although there are several applications of neural network algorithms to problems related to structural engineering, the presented work seems to be the first one proposed for the assessment of existing structures using drift estimations.

2 Neural Networks

This section provides a brief introduction to two main algorithms which are artificial neural networks (ANNs) and recurrent neural networks (RNNs). Although RNNs is adopted in the study, the section starts with ANNs which can be seen as the conceptual basis for RNNs.

2.1 Artificial Neural Networks

ANN, inspired by neurons found in a biological brain, consists of nodes and layers. Every node i.e. neuron in a layer is connected to all neurons in previous layers. In the forward pass operations, multiplication and summation are applied on each neuron with trainable parameters which are known as weight and bias constants. After this operation, a selected function, known as activation function, is applied (Equation 1-2). Sigmoid and hyperbolic tangent function are the commonly activation functions which are given in Equation 1 and Equation 2

$$z_j = \sum_i (w_i \times a_{i-1} + b) \tag{1}$$

$$a_i = f(z_j) \tag{2}$$

where a_0 is input and their graph can be seen in Figure 1.

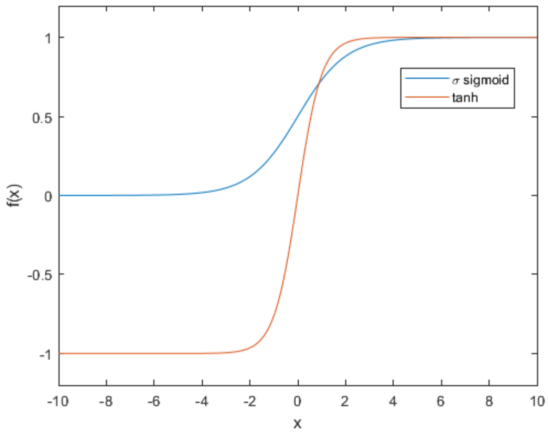

Figure 1. Sigmoid and hyperbolic tangent functions

The main objective of ANN is to minimize a loss function which is generally taken as the summation of differences between obtained output and desired output values. In order to minimize loss function, ANNs employ a backpropagation algorithm. After forward pass operations total loss is calculated. Backpropagation algorithms calculate loss for each node by employing derivatives of activation functions and, trainable parameters of the model are updated accordingly. The method has the power to find relations between input and output variables even if these relationships are highly nonlinear. After the training process, trained ANN model can be used for the prediction for input variables that ANN has not seen before.

2.2 Recurrent Neural Networks

There are three main differences between RNNs and ANNs. Firstly, RNNs enable output of the previous step to be used in the subsequent step as input. This is known as the recurrent procedure which is depicted representatively in Figure 2. Secondly, RNN cell contains persistent variables in itself as hidden state variables which evolve during the training process. Thirdly, input and output can be sequential data which means RNNs are sequence to sequence networks.

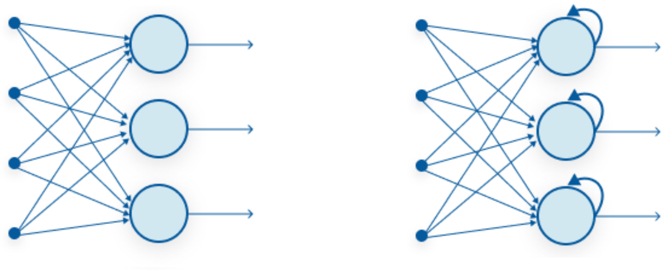

Figure 2. Artificial neural networks and recurrent neural networks for forward pass

ANN has fixed input and output shape limitation. However, in this study, our objective is to predict drift demands for each storey, thus, input and output shapes depend on the number of storeys. Apart from this limitation, RNNs can capture the temporal or spatial dependencies of input data. In the scope of this study, stiffness, number of storeys, mass of a typical floor, storey height, strength reduction factor and seismicity parameters are given as input to the algorithm and nonlinear drift demand profile is predicted by the constructed RNN. Since the inter-storey drift depends on the stiffness of the storey, stiffness of adjacent storeys and stiffness distribution over the structure, RNNs are able to catch these dependencies and they are expected to provide better results compared to estimations using ANNs. For these reasons, in this study, RNNs were adopted.

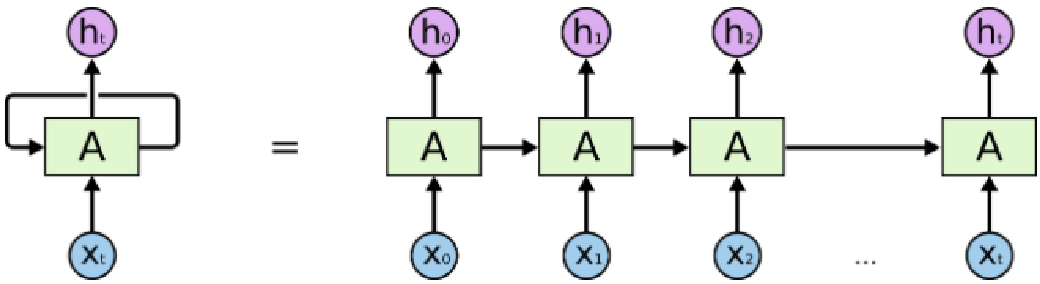

Figure 3. A recurrent neural network and its unrolled demonstration

Figure 3 shows a single RNN composed of a single cell on the left (which is the case in this study). Generally, RNNs take sequence as input, therefore, they can be depicted on the right-hand side in Figure 3 in which RNN is unrolled over the sequence or timestep. In accordance with Figure 3, for a *n*-storey structure, there are *n* steps as demonstrated on the right-hand side. Each X_i is a row vector gathered from input matrix and h_i is a scalar output for the drift for each storey. Generally, instead of “neuron”, “cell” terminology is preferred in RNNs. Recalling that RNN is a type of network, they are composed of cells. There are commonly used cells such as long short-term memory cell unit (LSTM), gated recurrent cell unit (GRU), basic RNN cell etc.

Since the basic RNN cell suffers from vanishing and exploding gradient problems which are related to the backpropagation update algorithm, LSTM cell is adopted in this study. In addition to the cell state, the LSTM cell consists of three gates. These are input gate, forget gate and output gate. LSTM solves the vanishing/exploding gradient problems by employing these gates. As in the case of a basic RNN cell, the cell state accommodates information from the previous step through the recurrent operation.

3 Literature Review

There are quite a few deep learning implementations in structural engineering as deep learning is a relatively new research subject in this field. Prediction of response of nonlinear structures under the ground motion by using machine learning algorithms is of interest of structural engineers. One of the first attempts for the dynamic response estimation of structures was presented by Wu et al. (2019). Convolutional neural network was used for response estimation and system identification for nonlinear SDOF and MDOF systems. They applied the proposed method to a number of different scenarios. For all cases, architecture of the CNN model was varied and the model was trained separately for all scenarios. Models were not generic, i.e., they were trained and conducted estimation for each case. Bas et al. (2020) attempted to predict hysteretic response of a steel element by using the LSTM model. In order to train the model, they used experimental data from two experiments and they tried to predict the response force from the imposed displacement in the experiment. The major drawback of the study is that they only predicted the second half of the response instead of the whole response from the imposed displacement. Liu et al. (2020) used recorded structural health monitoring data of a structure to predict modal parameters such as frequencies, mode shapes and damping ratio. By using the principle of orthogonality of mode shapes, they decomposed recorded signals to the modal combinations as a function of time. They implemented custom made loss functions within the artificial neural network to decompose recorded signals by enforcing orthogonality of modes. They showed applicability of the machine learning algorithms to such problems which can serve as a basis of the future research in this area. Another study on the topic of prediction of nonlinear structural response from the ground motion data was conducted by Zhang et al (2019). They employed layers of LSTM and fully connected layer on their model for prediction. They suggested two different LSTM approaches which are LSTM with standard input and LSTM with windowed input. They stated that the LSTM with windowed input has higher prediction accuracy. They applied the model to three different cases. Another attempt on this subject was presented by Zhang et al. (2020). Main idea of this study was to embed physical constraints and equilibrium (such as equation of motion) into the loss function. LSTM network was adopted and incremental dynamic analysis was performed for data generation and training.

4 Generated Dataset

In this chapter, synthetic dataset used in the training of RNNs is presented. 8000 buildings and their model parameters are randomly generated using uniform distributions. For seismicity, 4 locations (İstanbul, İzmir, Konya and Afyonkarahisar) and 2 soil conditions (ZC and ZD) are selected. Corresponding spectra is shown in Figure 4 (only for ZC). In addition, the following properties are randomly assigned to 8000 buildings:

- number of storeys is varied over 2 and 20,
- storey height is varied over 2.5 m to 4 m,
- mass density over a unit floor area varied 500 to 1000 kg/m2,
- plan area of a floor varied over 100 m2 to 1600 m2
- strength reduction factor (R) is varied over 4 and 8
- for buildings that have higher than 8 storeys, a stiffness reduction factor is generated in between 0.8 and 1 and applied to the top floor stiffness.

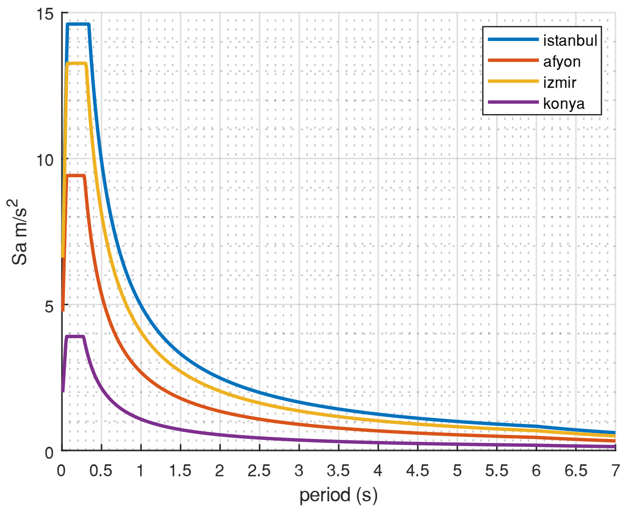

Figure 4. Considered spectrum (only shown for ZC soil type)

After randomly generating buildings with the above properties and assigned seismicity parameters, nonlinear time history analyses are conducted to form the training dataset for the RNN. Buildings were modelled as stick models. In this simplified model, each spring corresponds to a storey and modelled using the hysteretic material model in OpenSees (Mazzoni et al., 2006) software. In order to conduct the analyses, shear strength and the yield displacement should be calculated in accordance with the assigned spectrum. The following procedure is followed to calculate them:

i. Storey shear strength and yield displacement values should be consistent with the assigned spectrum. For this purpose, an empirical equation is used for approximating the period of buildings as shown in Equation 3.

$$T_{app} = 0.05 \times H^{0.75} \tag{3}$$

ii. Assuming the first mode shape and using the approximated period, stiffness is calculated.
iii. Total base shear force demand is obtained from the assigned spectrum by using period, total mass and strength reduction factor.
iv. Yield displacement (u_y) of storeys (spring/truss elements) is taken as the division of total base shear force to stiffness calculated at step ii.

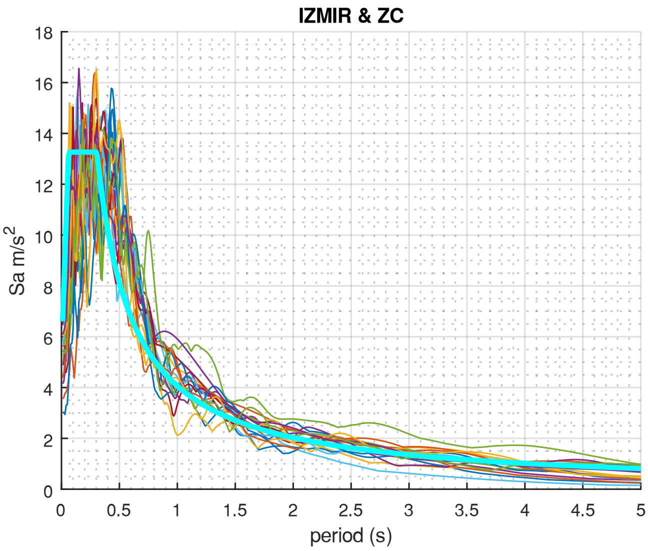

Figure 5. Linear design spectrum for location of Izmir and soil type of ZC and spectrum of matched ground motions

After calculating and assigning the storey (spring/truss element) strength and yield displacement, computational models are constructed using the OpenSees software. 22 ground motion are scaled to the 8 different spectra by preserving frequency content, as shown in Figure 5. For each building, 22-time history analyses are conducted using the scaled ground motions. Envelope of inter-storey drift demands for each analysis is gathered and averaged. Thus, a single inter storey drift profile is calculated for each building.

	stiffness	drift	#storey	m_floor	height	n	R	location	soil	spectrum
storey 13	1.528E+09	0.000334	13	545618.1	3.549001	13	4	2	1	3
storey 12	1.668E+09	0.000647	12	545618.1	3.549001	13	4	2	1	3
storey 11	1.808E+09	0.000985	11	545618.1	3.549001	13	4	2	1	3
storey 10	1.948E+09	0.001266	10	545618.1	3.549001	13	4	2	1	3
storey 9	2.087E+09	0.001437	9	545618.1	3.549001	13	4	2	1	3
storey 8	2.227E+09	0.00439	8	545618.1	3.549001	13	4	2	1	3
storey 7	2.227E+09	0.004279	7	545618.1	3.549001	13	4	2	1	3
storey 6	2.227E+09	0.003821	6	545618.1	3.549001	13	4	2	1	3
storey 5	2.227E+09	0.004277	5	545618.1	3.549001	13	4	2	1	3
storey 4	2.227E+09	0.005719	4	545618.1	3.549001	13	4	2	1	3
storey 3	2.227E+09	0.012201	3	545618.1	3.549001	13	4	2	1	3
storey 2	2.227E+09	0.017166	2	545618.1	3.549001	13	4	2	1	3
storey 1	2.227E+09	0.017936	1	545618.1	3.549001	13	4	2	1	3

Figure 6. An example of generated data for a sample building

Finally, generated data contains the following variables for a building sample (Figure 6):

- drift profile over the height,
- stiffness profile over the height,
- number of storeys,
- mass of a typical floor,
- storey height,
- reduction factor,
- seismicity variables i.e. location, soil and spectrum encode.

4 Training and Results

Input matrix was given to LSTM row by row which represents a storey and for each row one output was demanded from the network. As mentioned before, RNNs can have variable input/output during the training, therefore, prediction of a variable for each storey can easily be handled with RNNs. Also, it should be noted that RNNs can capture the dependency between inputs and outputs through their recurrent operation and persistence values embedded in them. In order to train the RNN for the drift profile prediction, the generated dataset is manipulated such that drift profile is separated as desired output and the rest are taken as input. Some of the important parameters for the model are listed below:

- There are 8000 buildings. Training set, validation set and test set were divided as 6400, 800 and 800, respectively.
- 30 epochs are performed for the training process.
- 25 LSTM unit is adopted, therefore, there are 3700 trainable parameters.
- Adam optimization algorithm is utilized for the back-propagation update.
- Loss function is defined as mean squared error between true values and predicted values of output variable. No reduction was preferred, instead, output of the loss function was left as a vector.
- Before the training operation, min-max normalization is applied to the data.

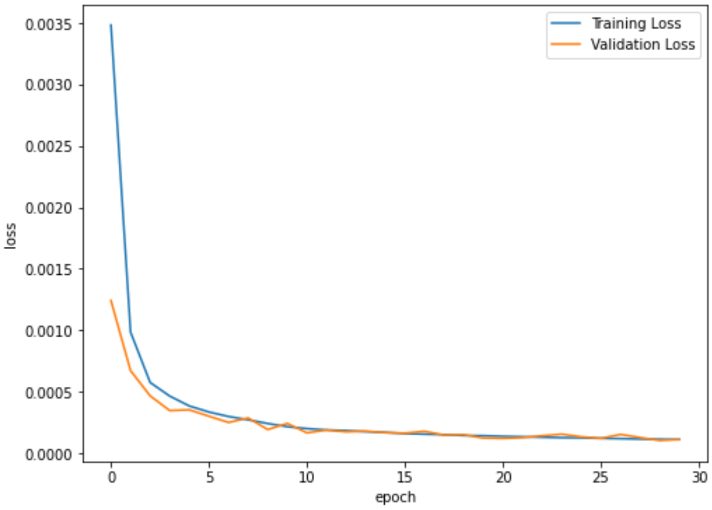

Figure 7. Training and validation loss throughout training

Evolution of the loss function during the training process is shown in Figure 7. The network does not consider test data as input throughout the training process, thus, only the prediction results for the test data are shown in the following figures. In addition to the mean of predicted drift profile, drift predictions for first storey are shown in Figure 8. As can be observed from the figures, the RNN network can predict output variables accurately.

Among the test data, two building samples are given in the Figure 9 along with the inter-storey drift demand prediction. The building on the left is located in Konya whereas the building on the right is located in İzmir. As can be seen from the figure, the RNNs give reasonable prediction for inter-storey drift profile for both buildings. Moreover, in Figure 10, maximum inter-storey drift ratio along the height for each building sample from the test data is given on the left and their mean absolute errors arisen from the predictions are given in the right. As can be seen from the Figure 10, the mean error of a building does not depend on the spectrum or location, therefore, it can be said that the trained network can produce accurate predictions for any selected spectrum and location.

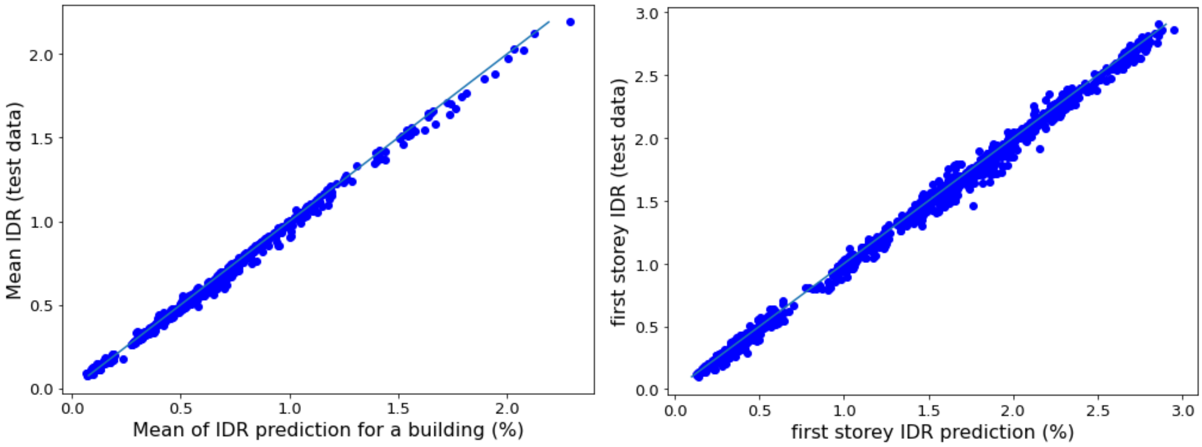

Figure 8. Predicted versus true value for mean drift ratio over the height (left), for first storey (right)

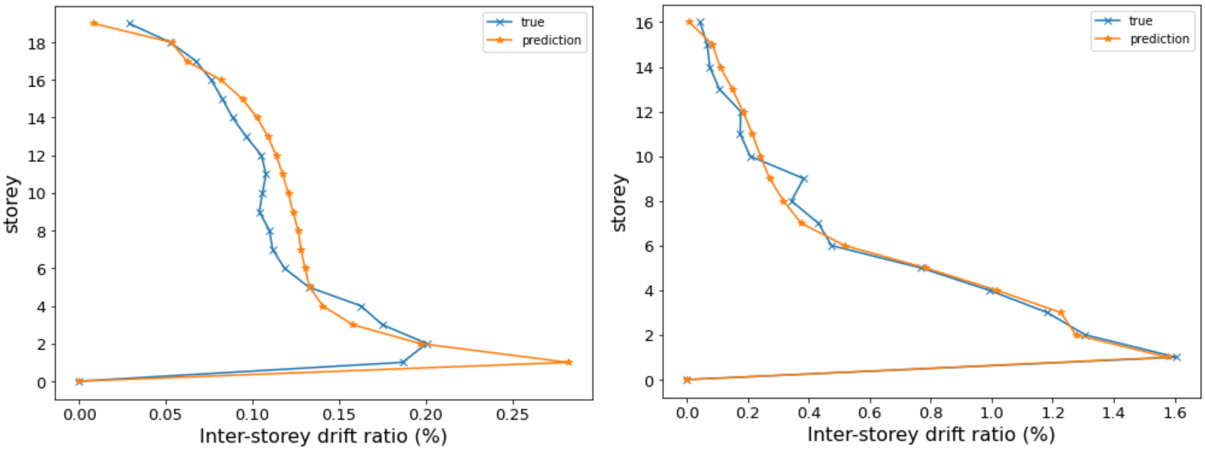

Figure 9. Predicted versus true value for inter-storey drift ratio over the height for two building samples

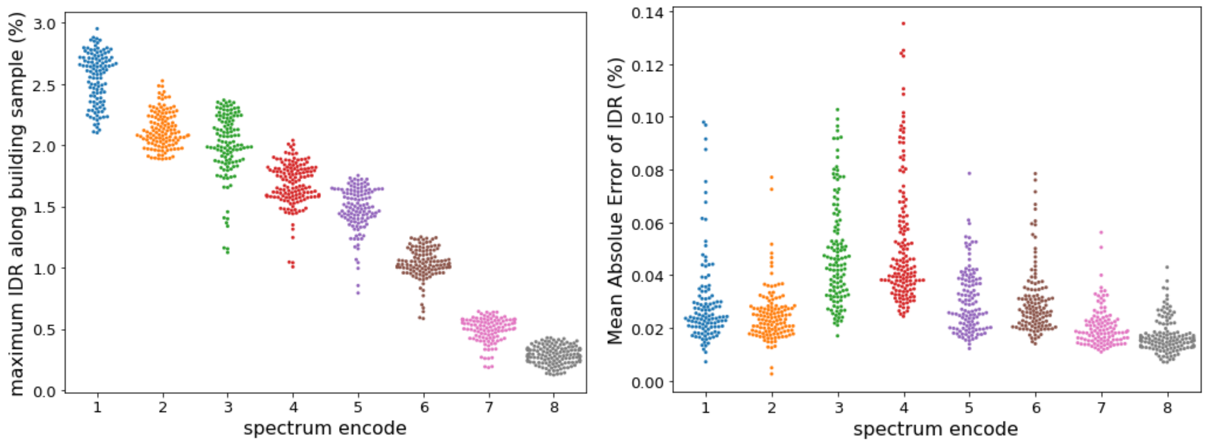

Figure 10. Maximum inter-storey drift ratio of test data versus their spectrum encodes (left), their mean absolute error after prediction versus spectrum encodes (right)

5 Conclusions

In this study, inter-storey drift profile of a building is estimated as a function of location, soil type, number of stories, strength reduction factor, and storey height. The RNN is constructed using 6400 buildings that are analyzed using the nonlinear stick model approximation. 800 test data (not used in the training set) estimations are in very good agreement with expected values. Due to the flexibility of RNNs that they can work with variable input and output shape, the trained network is not limited for the number of storeys. Therefore, unlike many examples from the literature, anyone can use the trained network without training it once again for their structure. The trained network can serve as a tool for assessment procedures which require computation of nonlinear drift demands. In addition, it can be used as an alternative to rapid assessment procedures.

References

Bao, Y., Liu, D., Tang, Z., & Li, H. (2020). Machine-learning-based methods for output only structural modal identification. arXiv preprint arXiv:2004.07644.

Bas, E., Aslangil, D., & Moustafa, M. A. (2020). Predicting Nonlinear Seismic Response of Structural Braces Using Machine Learning. arXiv e-prints, arXiv-2007.

Akpinar, U., Binici, B., Yakut, A. et al. An approximate procedure for estimating the member demands in mid-rise reinforced concrete buildings. Bull Earthquake Eng 18, 6715–6734 (2020). https://doi.org/10.1007/s10518-020-00956-3

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 & Structures, 220, 55–68. https://doi.org/10.1016/j.compstruc.2019.05.006

Zhang, R., Liu, Y., & Sun, H. (2020). Physics-informed multi-LSTM networks for metamodeling of nonlinear structures. Computer Methods in Applied Mechanics and Engineering, 369, 113226. https://doi.org/10.1016/j.cma.2020.113226

Wu, R.-T., & Jahanshahi, M. R. (2019). Deep Convolutional Neural Network for Structural Dynamic Response Estimation and System Identification. Journal of Engineering Mechanics, 145(1), 04018125. https://doi.org/10.1061/(asce)em.1943-7889.0001556

Mazzoni, S., McKenna, F., Scott, M. H., and Fenves, G. L. (2006) OpenSees Command Language Manual University of California, Berkeley, https://opensees.berkeley.edu/

Disaster and Emergency Management Presidency, Türkiye Bina Deprem Yönetmeliği, Offical Gazette, 2018