Identifying and Locating Earthquake-Induced Damage in a High-Rise Using Neural Operators

JONATHAN TA, GABRIEL PIZARRO, ELIJAH TAVARES,
YUQIAO LI, MONICA KOHLER,
NALINI VENKATASUBRAMANIAN
and SANG-WOO JUN

ABSTRACT

Rapid response after earthquakes is vital to mitigate the effects of catastrophic structural failures and to save lives. In particular, critical infrastructure relies on accurate yet low-latency damage detection to facilitate timely responses. Although traditional techniques rely on hand-crafted feature extraction and data interpretation to identify the presence of damage within a structure, deep learning has resulted in newer methods that combine feature extraction and data interpretation in one. However, these models often require exorbitant amounts of data to train, are often computationally expensive, and are difficult to cross-validate with established theory due to their black box nature. In this paper, we present our current workflow that addresses the aforementioned challenges: (i) constructing representative finite-element models of a 15-story steel frame building with simulated damage pattern scenarios in the form of weld fractures, (ii) generating a synthetic waveform dataset for earthquake-induced dynamic response using a well-known dynamic analysis simulator, OpenSees, for the finite-element models and (iii) implementing a deep learning model that performs damage identification in addition to *damage localization*, i.e., locating the damage. To construct the dataset, we apply recorded earthquake ground motions to the 15-story building model modified to include small-scale damage scenarios imposed at beam-column connections. We then compute accelerations at each of the four corners within the structure for each individual modified model and earthquake input. Our deep learning model consists of a neural operator in conjunction with a machine learning mechanism called self-attention. To our knowledge, our work is the first work exploring the use of neural operators for earthquake-specific vibration-based damage identification. In addition, we present an analysis of our model performance and model fit, and compare our model to a subset of machine learning techniques.

INTRODUCTION

The ability to identify and locate damage in structures such as buildings after earthquakes is critical for functional recovery of civil infrastructure. Vibration-based techniques are especially useful for characterizing damage, particularly for areas within structures where the damage is not visible or easily inspected. There is a rich and extensive history of studies on structural health monitoring (SHM) and damage detection techniques that focus on data-driven approaches for feature extraction, interpretation, and numerical modeling [1–7]. Combining data with traditional physics-based structural models can be difficult to implement in practice due to prohibitive computation requirements, in addition to challenges involving model uncertainty or incomplete information.

With the rapid advancement of deep learning, modern approaches using deep learning blur the boundary between feature extraction and interpretation, often combining both capabilities into one model without sacrificing model accuracy. However, deep learning suffers from its own drawbacks, most relevantly, exorbitant data requirements and sizable computational requirements.

Damage identification training data can be surprisingly difficult to obtain. Gathering and collecting real-world data often requires instrumenting real-world buildings [8] or constructing structures with the explicit purpose of collecting damage data [9]. However, using real-world data has its own challenges, as real-world buildings often have incomplete structural information. Custom-built building models tend to be either too small or too simple to be representative, ultimately becoming unsustainable for long-term study. In light of these challenges, we focus here on generating synthetic vibration datasets for customizable structures to use in a proof-of-concept study of a specific deep learning approach - neural operators - for damage detection.

Recent advances in machine learning have led to the development of neural operators, a generalized technique for learning operators, i.e., mappings between function spaces [10]. Notably, neural operators are a promising approach towards analytically solving partial differential equations (PDEs), offering advantages such as computational efficiency and resolution invariance when compared with traditional finite-element models. Thus, neural operators have seen usage in SHM as a method of learning solutions to PDEs that were previously prohibitively expensive [11]. Within the context of vibration-based damage identification, we utilize neural operators to solve the inverse problem, i.e., estimating parameter coefficients using recorded vibrations.

In this paper, we present our current workflow that addresses the aforementioned challenges: (i) constructing representative finite-element models of a 15-story steel frame

Jonathan Ta, Department of Computer Science, University of California, Irvine, Irvine, CA, USA.
Gabriel Pizarro, College of Creative Studies, University of California, Santa Barbara, Santa Barbara, CA, USA.
Elijah Tavarres, Department of Computer Science, University of California, Irvine, Irvine, CA, USA.
Yuqiao Li, Department of Computer Science, University of California, Irvine, Irvine, CA, USA.
Monica Kohler, Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA, USA
Nalini Venkatasubramanian, Department of Computer Science, University of California, Irvine, CA, USA
Sang-Woo Jun, Department of Computer Science, University of California, Irvine, CA, USA

building with simulated damage pattern scenarios in the form of weld fractures, (ii) generating a synthetic waveform dataset for earthquake-induced dynamic response using a well-known dynamic analysis simulator, OpenSees [12], for the finite-element models and (iii) implementing a deep learning model that performs damage identification in addition to *damage localization*, i.e., locating the damage. We evaluate the effectiveness of the model and benchmark our model against traditional machine learning approaches.

DAMAGE IDENTIFICATION UTILIZING NEURAL OPERATORS

Let $\mathcal{L}$ be a linear differential operator parametrized by a. Consider the partial differential equation (PDE)

$$(\mathcal{L}_a u)(x) = f(x) \tag{1}$$

where u is the PDE solution, f is the source term, and $x \in \mathbb{R}^n$. In the context of earthquake-specific damage identification, the source term f would be forces imparted by the ground motions on the building. Typically, simulators such as OpenSees will numerically solve for $u(x)$ using finite-element models. For the purpose of damage identification, we need to obtain $a(x)$, i.e., the parameter space that informs the stiffness in tension within the model. Thus, our goal is to build an operator G that learns an approximation of $(\mathcal{L}_a u)(x)$ by minimizing the loss $\|G - (\mathcal{L}_a u)\|^2$ [10].

We consider the neural operator that maps functions of $u(x)$ to $a(x)$. The operator can be constructed as the following

$$h_0(x) = (Pu)(x), \tag{2}$$

$$h_{l+1}(x) = \sigma(W_l h_l(x) + \int_D \kappa_l(x, y) h_l(y)\, \mathrm{d}y), \qquad 0 \leq l < L \tag{3}$$

$$h_{L+1}(x) = (Q h_L)(x) \tag{4}$$

where $h_{L+1}(x)$ is a learned approximation of $a(x)$, h_l are the intermediary results commonly referred to as the hidden dimensions, L is the depth of the neural operator, P, Q are point-wise operators that increase and reduce the input dimension respectively, σ is an activation function, W_l are point-wise operators that parametrize the residual connections, and κ_l are the parametric kernel functions [10]. Fourier Neural Operators (FNOs) are a particular variant of neural operators that replace the kernel integral operator with convolution operators in the Fourier domain, improving computational efficiency [13].

METHODOLOGY

Simulation

We simulate the effect of earthquake ground motions on a 15-story high-rise building and record the vibrations for each corner of the building on every floor. We use OpenSees, a framework for modeling structural response to earthquake motions using linear finite-element models to generate the simulated dataset [12]. For a given earthquake, the recorded vibrations are applied to the base of the building, equivalently across the orthogonal horizontal x, y directions. Prior to that, we randomly impose damage

Location	Station/Component	Date	Magnitude
Kobe, Japan	NIS000	Jan. 17, 1995	6.9
Izmit, Turkey	DZC270	Aug 17, 1999	7.6
Chi-chi, Taiwan	CHY080	Sep. 21, 1999	7.3
Duzce, Turkey	BOL090	Nov. 12, 1999	7.2
Hector Mine, California	HEC090	Oct. 16, 1999	7.1
Denali, Alaska	PS10-047	Nov. 3, 2002	7.9
Sierra El Mayor, Baja California	MDO000	Apr. 4, 2010	7.2
Darfield, New Zealand	GLDC35W	Sep. 4, 2010	7.1

TABLE I. Earthquakes used for simulation dataset.

across the beam elements in the corners of the building for each simulation. To simplify, we only consider damage events that reduce the stiffness in tension by half. We consider a set of ground motions M and a set of damageable elements that comprise the simulated building E. Then, each simulation can be parametrized as some $x \sim S(m, e)$ for $m \in M, e \in \mathcal{P}(E)$, i.e., for some particular ground motion, some particular combination of damaged elements, and a simulation function S.

Each of the simulations is divided into tiers of damage, with each tier comprising between one and ten damaged elements, in addition to the undamaged baseline case. For example, tier 0 would consist of the undamaged scenario, tier 1 would consist of simulations with exactly 1 damaged element, etc. We sample 100 simulations per tier and do not repeat permutations. Figure 1 demonstrates an example damage scenario from our dataset. The current synthetic dataset consists of a total of $\left(\binom{60}{0} + \binom{60}{1} + 100 \cdot 9\right) \cdot 8 = 7688$ simulations across eight ground motions and ten damage tiers.

Each simulation consists of the recorded time series $x \in \mathbb{R}^{2\times60\times4500}$ – 2 per the x, y directions, computed for all 60 corners in the 15-story building, and 4500 for a 45-second window sampled at 100 Hz. We omit z-axis (vertical) vibrations for our initial study to reduce the complexity of the problem.

Dataset

We select a subset of eight earthquakes from the NGA-West2 database [14] for use in the OpenSees simulations. The particular eight earthquakes can be seen in Table I. Each ground motion is processed through a standardized procedure. Specifically, we upsample or downsample the signal to 100 Hz. If the time series was downsampled, we pass it through a lowpass filter to remove aliasing. The resulting waveforms are then normalized to a maximum amplitude of $0.1g$ and snipped to a 45-second window. Longer earthquake records were truncated and tapered, while shorter earthquake records were padded with zeros. To regularize the model, we add a small amount of noise during training to the vibrations. This not only prevents the model from overfitting, but also has the added benefit of simulating realistic behavior where a model may be exposed to noisy signals in buildings.

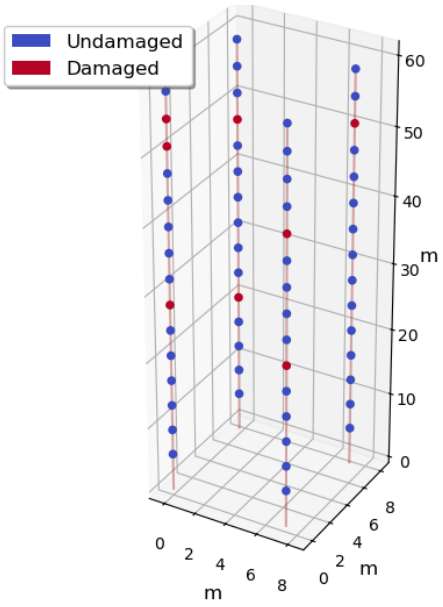

Figure 1. An example of a ground truth damage scenario from the dataset.

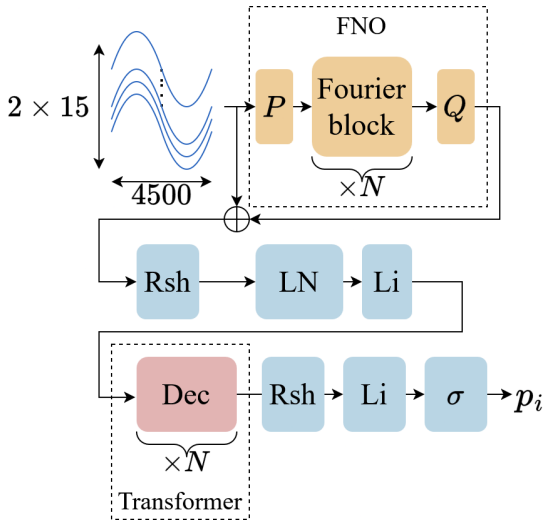

Figure 2. The proposed model architecture utilizing FNO and decoders.

Model

Our model architecture consists of an FNO in conjunction with self-attention decoder blocks [15] to predict the probability of damage for a given element. We feed in the vibration time series to the model, and output predictions p_i for each building corner element $i \in E$. Figure 2 gives a high-level overview of the model. The abbreviated labels Rsh, LN, Li, and Dec stand for Reshape, LayerNorm, Linear, and Decoder respectively. We implement our model in `pytorch` and the `neuralop` library for the Fourier neural operators [16, 17]. For simplicity and reduced input dimension, the current model is trained only on a single corner per floor of the synthetic dataset. Figure 3 shows model loss and accuracy during training. We train with a learning rate of 10^{-5} and mean-squared error loss.

EVALUATION

The original dataset is divided into subsets of 70% training, 15% test, and 15% validation, respectively. The subsequent evaluation studies use the validation dataset.

Let $\hat{y} = \{\hat{p}_i : \forall i \in E\}$ denote the model's predictions, and $y = \{p_i : \forall i \in E\}$ be the ground truth labels. We define positive predictions as $\hat{p}_i > t$, and negative predictions as $p_i \leq t$, for some threshold t. True positives (TP_i) are the instances of correctly predicted positive predictions, while false positives (FP_i) are the instances of incorrectly predicted positive predictions, i.e., $\hat{p}_i > t, p_i = 1$ for some element i, and vice versa for true negatives (TN_i) and false negatives (FN_i).

Benchmarks

We compare our work against a subset of machine learning techniques such as Con-

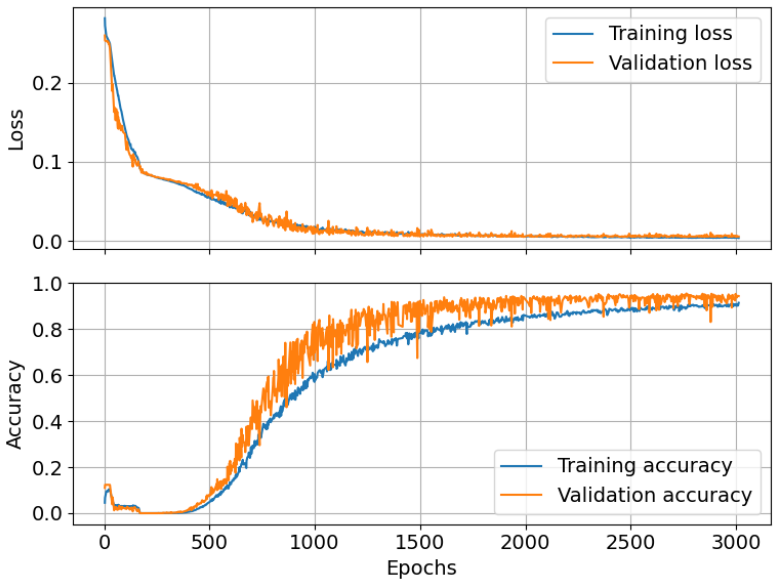

Figure 3. Model training as a function of number of epochs. Top panel: Loss. Bottom panel: Accuracy.

volutional Neural Networks (CNN) and Support Vector Machines (SVM). We briefly describe and evaluate these models on their precision, recall, and f_1 score (Eqs. 5-7). Table II lists these metrics for each model. The SVM achieves a recall of 1.0 due to the model only predicting positives, i.e., it did not learn a meaningful representation.

$$precision = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FP_i} \tag{5}$$

$$recall = \frac{\sum_i TP_i}{\sum_i TP + \sum_i FN} \tag{6}$$

$$f_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}. \tag{7}$$

CNNs are widely used in signal processing tasks due to their ability to automatically extract local and hierarchical features. In particular, 1D-CNNs have seen recent use as a method to process time series data [9]. We investigate two different variants of CNNs, CNNs with 2D and 1D kernels respectively. For 2D-CNNs, we substitute the 3 FNO layers in our model with a 3-layer 2D-CNN. For each convolution layer, we apply a 3×15 size kernel to convolve across the 15×4500 vibration time series per cardinal direction. For 1D-CNNs, we apply the convolution across each time series input per floor with a kernel size of 5. The model consists of submodules for every floor, with each submodule consisting of two 1D convolution layers, 1D Maxpooling, and 2 linear layers.

SVMs are a machine learning technique commonly used for classification and regression tasks. We extract a host of 16 features for training, including, but not limited to, mean, standard deviation, and median. Due to limitations in the SVM model, we restrict the SVM to predict damage in the building instead of the individual elements, in addition to the number of damaged points.

metric	Ours	2D-CNN	1D-CNN	SVM
precision	0.98927	0.93184	0.34415	0.5199
recall	0.93990	0.73679	0.17247	1.0000
F1	0.96395	0.82292	0.22979	0.6842

TABLE II. Model performance results.

Goodness-of-fit

The receiver operating characteristic (ROC) curve is often used to assess the performance of a classifier and its sensitivity to threshold. We use a one-versus-rest (OvR) strategy to assess multiclass ROC [18], where the true positive rate (TPR) and false positive rate (FPR) are calculated as follows:

$$TPR = \frac{1}{60} \sum_i \frac{TP_i}{TP_i + FN_i} \tag{8}$$

$$FPR = \frac{1}{60} \sum_i \frac{FP_i}{FP_i + TN_i}. \tag{9}$$

The ROC was evaluated across the possible thresholds 0.0 to 1.0 with a step size of 0.01.

Figure 4 describes the TPR for each of the possible damageable elements for a threshold value $t = 0.5$ in the validation dataset. The choice of t is akin to a confidence threshold— it indicates that predictions with $p > 0.5$ are treated as positive predictions. Surprisingly, there is no discernible pattern of the TPR across each of the corners. Figure 5 illustrates the ROC curve in relation to the other benchmarks. Note that although 2D-CNN demonstrates impressive results, this performance is due to conservative predictions that result in low false positives but a larger number of false negatives.

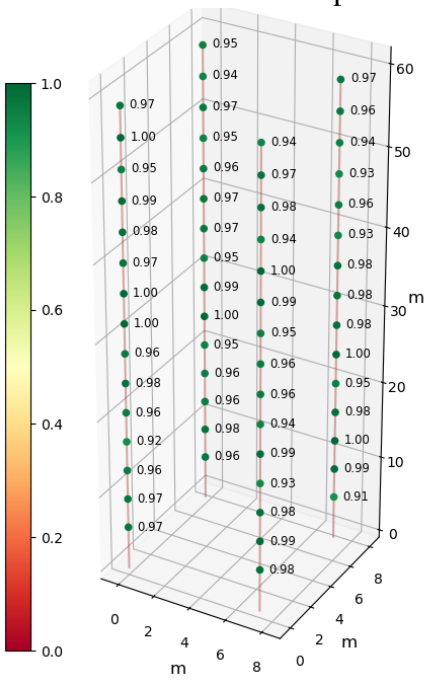

Figure 4. True Positive Rate (TPR) across damageable elements.

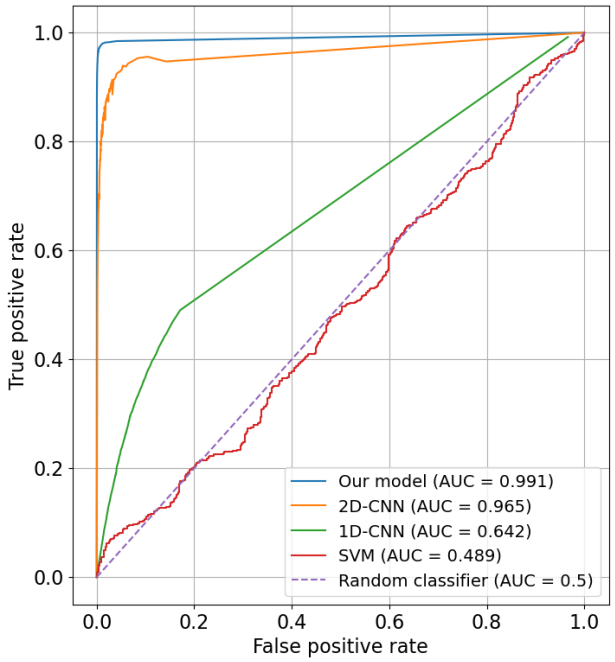

Figure 5. Receiver Operator Characteristic (ROC) across different models.

CONCLUDING REMARKS

We present our current workflow towards a methodology that generates synthetic data for earthquake-oriented damage identification and a complementary model that performs damage identification and damage localization. We find that the model performs well on the generated dataset, and we present some metrics that explore the model performance. Finally, we benchmark our model performance relative to a subset of other machine learning approaches.

REFERENCES

1. Adams, D. E. and R. J. Allemang. 1998. "Survey of Nonlinear Detection and Identification Techniques for Experimental Vibrations," *Proc. ISMA*, 23:269–281.
2. An, Y., E. Chatzi, S.-H. Sim, S. Laflamme, B. Blachowski, and J. Ou. 2019. "Recent progress and future trends on damage identification methods for bridge structures," *Structural Control and Health Monitoring*, 26(10):e2416, ISSN 1545-2263, doi:10.1002/stc.2416, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/stc.2416.
3. Carden, E. P. and P. Fanning. 2004. "Vibration Based Condition Monitoring: A Review," *Structural Health Monitoring*, 3(4):355–377, ISSN 1475-9217, doi:10.1177/1475921704047500, publisher: SAGE Publications.
4. Chang, P. C., A. Flatau, and S. C. Liu. 2003. "Review Paper: Health Monitoring of Civil Infrastructure," *Structural Health Monitoring*, 2(3):257–267, ISSN 1475-9217, doi:10.1177/1475921703036169, publisher: SAGE Publications.
5. Doebling, S. W., C. R. Farrar, M. B. Prime, and D. W. Shevitz. 1996. "Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review," Tech. Rep. LA-13070-MS, Los Alamos National Lab. (LANL), Los Alamos, NM (United States), doi:10.2172/249299.
6. Fan, W. and P. Qiao. 2011. "Vibration-based Damage Identification Methods: A Review and Comparative Study," *Structural Health Monitoring*, 10(1):83–111, ISSN 1475-9217, doi:10.1177/1475921710365419, publisher: SAGE Publications.
7. Farrar, C. R. and K. Worden. 2006. "An introduction to structural health monitoring," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):303–315, doi:10.1098/rsta.2006.1928, publisher: Royal Society.
8. Kohler, M. D., F. Filippitzis, T. Heaton, R. W. Clayton, R. Guy, J. Bunn, and K. M. Chandy. 2020. "2019 Ridgecrest Earthquake Reveals Areas of Los Angeles That Amplify Shaking of High-Rises," *Seismological Research Letters*, 91(6):3370–3380, ISSN 0895-0695, doi:10.1785/0220200170.
9. Abdeljaber, O., O. Avci, M. S. Kiranyaz, B. Boashash, H. Sodano, and D. J. Inman. 2018. "1-D CNNs for structural damage detection: Verification on a structural health monitoring benchmark data," *Neurocomputing*, 275:1308–1317, ISSN 0925-2312, doi:10.1016/j.neucom.2017.09.069.
10. Kovachki, N., Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. 2023. "Neural operator: learning maps between function spaces with applications to PDEs," *J. Mach. Learn. Res.*, 24(1):89:4061–89:4157, ISSN 1532-4435.
11. Zou, C., K. Azizzadenesheli, Z. E. Ross, and R. W. Clayton. 2024. "Deep neural Helmholtz operators for 3-D elastic wave propagation and inversion," *Geophysical Journal International*, 239(3):1469–1484, ISSN 1365-246X, doi:10.1093/gji/ggae342.
12. McKenna, F. 2011. "OpenSees: A Framework for Earthquake Engineering Simulation," *Computing in Science & Engineering*, 13(4):58–66, ISSN 1558-366X, doi:10.1109/MCSE.2011.66.
13. Li, Z., N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. 2020. "Fourier Neural Operator for Parametric Partial Differential Equations," .
14. Ancheta, T., R. Darragh, J. Stewart, E. Seyhan, B. Chiou, K. Wooddell, R. Graves, A. Kottke, D. Boore, T. Kishida, and J. Donahue. 2014. "NGA-West2 database," *Earthquake Spectra*, 30, doi:10.1193/070913EQS197M.
15. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polosukhin. 2017. "Attention is All you Need," in I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., *Advances in Neural Information Processing Systems*, Curran Associates, Inc., vol. 30.
16. 2025, "github.com/neuraloperator," `https://github.com/neuraloperator/neuraloperator`.
17. "PyTorch," `https://pytorch.org/`.
18. "Multiclass Receiver Operating Characteristic (ROC)," `https://scikit-learn/stable/auto_examples/model_selection/plot_roc.html`.