Summer 5-31-2024

Machine Learning Based Analysis Of Civil Infrastructure In The Presence Of Sparse Data

Megan Butcher
Embry-Riddle Aeronautical University, butchem3@my.erau.edu

MACHINE LEARNING BASED ANALYSIS OF CIVIL INFRASTRUCTURE IN THE PRESENCE OF SPARSE DATA

by: Megan Lee Butcher

Ashok Gurjar, Ph.D.

Civil Engineering Department Chair

Siddharth Parida, Ph.D.

Committee Chair

Dan Su, Ph.D.

Committee Member

Prashant Shekhar, Ph.D.

Committee Member

Jeff Brown, Ph.D.

Committee Member

Date: August 7, 2024

MACHINE LEARNING BASED ANALYSIS OF CIVIL INFRASTRUCTURE IN THE PRESENCE OF SPARSE DATA

Megan Lee Butcher

A thesis/dissertation submitted in partial fulfillment of the requirements
for the degree of Master of Science in Civil Engineering - Structures Track
at Embry-Riddle Aeronautical University
August 2024

Acknowledgements

It has been an immense privilege to undertake this research, and I owe much of its success to the support of Embry-Riddle Aeronautical University. Foremost, I extend my deepest gratitude to my thesis advisor, Dr. Siddharth Parida, Assistant Professor of Civil Engineering. Dr. Parida not only entrusted me with an impactful research topic but also guided me with profound insight and encouragement since May 2021. His mentorship has not only shaped my academic journey but also fostered my growth as a researcher and an individual.

In addition, I am indebted to my esteemed committee members: Dr. Jeff Brown, Professor of Civil Engineering and former Program Coordinator for the B.S. and M.S. in Civil Engineering; Dr. Dan Su, Assistant Professor of Civil Engineering; and Dr. Prashant Shekhar, Assistant Professor of Data Science/Math. Their collective wisdom, constructive feedback, and scholarly guidance were pivotal in navigating this thesis topic.

Furthermore, I extend my appreciation to all faculty members within the civil engineering department at Embry-Riddle Aeronautical University. Their unwavering support, valuable advice, and extensive knowledge significantly contributed to the development of my professional life. Their dedication to fostering academic excellence has been instrumental in my academic journey. I am also grateful to the American Society of Civil Engineers (ASCE) for providing invaluable opportunities for professional, personal, and academic growth.

Lastly, I wish to acknowledge my family, friends, and classmates whose steadfast encouragement and unwavering belief in my abilities have been a constant source of motivation throughout my educational career. Their support has been invaluable, and I am profoundly grateful for their presence in my life. I am deeply thankful to all who have supported me on this journey. Their encouragement and guidance have been instrumental in my success, and I carry their lessons and support with me as I embark on future endeavors.

Abstract

The high computational cost of estimating engineering demand parameters (EDPs) via finite element (FE) models, which incorporate uncertainties in earthquake events and material properties, limits the application of the Performance-Based Earthquake Engineering (PBEE) framework. Previous efforts to replace FE models with surrogate models have typically focused only on building parameters, necessitating re-training for new, unseen earthquakes. This paper introduces a machine learning-based surrogate model framework that addresses both earthquake and material parameter uncertainties to predict responses for unseen seismic events. Earthquakes are characterized by their projections on an orthonormal basis, computed using Singular Value Decomposition (SVD) of a representative ground motion suite, allowing for the generation of varied earthquake scenarios by sampling these weights. These weights, along with constitutive parameters, serve as inputs to the machine learning models, with EDPs as the output. Four competing machine learning models were evaluated, with deep neural networks (DNNs) demonstrating the highest accuracy. The framework's validity is shown through its successful prediction of the peak responses of one-story and three-story shear frame buildings, represented as nonlinear spring–mass–damper systems, subjected to unseen far-field ground motions. Furthermore, the study highlights the importance of rigorously characterizing forcing functions, as predictions are highly sensitive to these parameters. Machine learning tools, due to their flexibility and efficiency, have emerged as powerful alternatives in various engineering fields. In this study, the application of machine learning for predicting EDPs, while considering uncertainties in both forcing functions and model parameters, is assessed. Results indicate that DNNs perform the best among the tested models. This comprehensive framework integrates machine learning into the PBEE framework, offering a cost-effective solution for structural analysis under uncertain conditions.

Contents

List of Figures

List of Tables

1 Motivation of Research

Performance-based earthquake engineering (PBEE) is used to analyze and design structures based on their expected performance during earthquakes. This approach was developed by the Pacific Earthquake Engineering Research (PEER) center which provides data, models, and software resources to support a structured performance-based earthquake engineering approach [Cornell and Krawinkler, 2000, Snaiki and Parida, 2023a,b]. Compared to traditional seismic design methods that focus primarily on ensuring that structures remain standing, also known as life safety [Pessiki, 2017], PBEE evaluates the performance of structures across a range of performance objectives, including safety, functionality, and ability to be repaired. This approach considers the potential consequences of various seismic hazard levels and assesses how well a structure can withstand those hazards while minimizing damage and ensuring occupant safety.

This framework for the PBEE approach is illustrated in Figure 1.1 and has four sequential stages: hazard analysis, structural analysis (SA) through finite element (FE) simulation, converting engineering demand parameters (EDPs) obtained from SA into damage measures (DMs), and translating DMs into different decision variables (DVs) [Porter et al., 2004]. The PBEE framework begins by defining a ground motion intensity measure, which characterizes the critical aspects of ground motion hazard affecting structural response. EDPs are then determined to depict structural response through deformations, accelerations, or other simulated response metrics to the input ground motions. These EDPs are then correlated with DMs, describing the structure and its components' condition. This process concludes with a comprehensive probabilistic depiction of damage that calculates the DVs. These variables, are aligned with the decision-makers requirements, with metrics like repair costs, downtime, and casualty rates, to facilitate effective risk management decisions [Moehle and Deierlein, 2004].

While it serves as an exemplary framework for performance evaluation, its application is constrained by computational expenses involved with the nonlinear probabilistic finite element

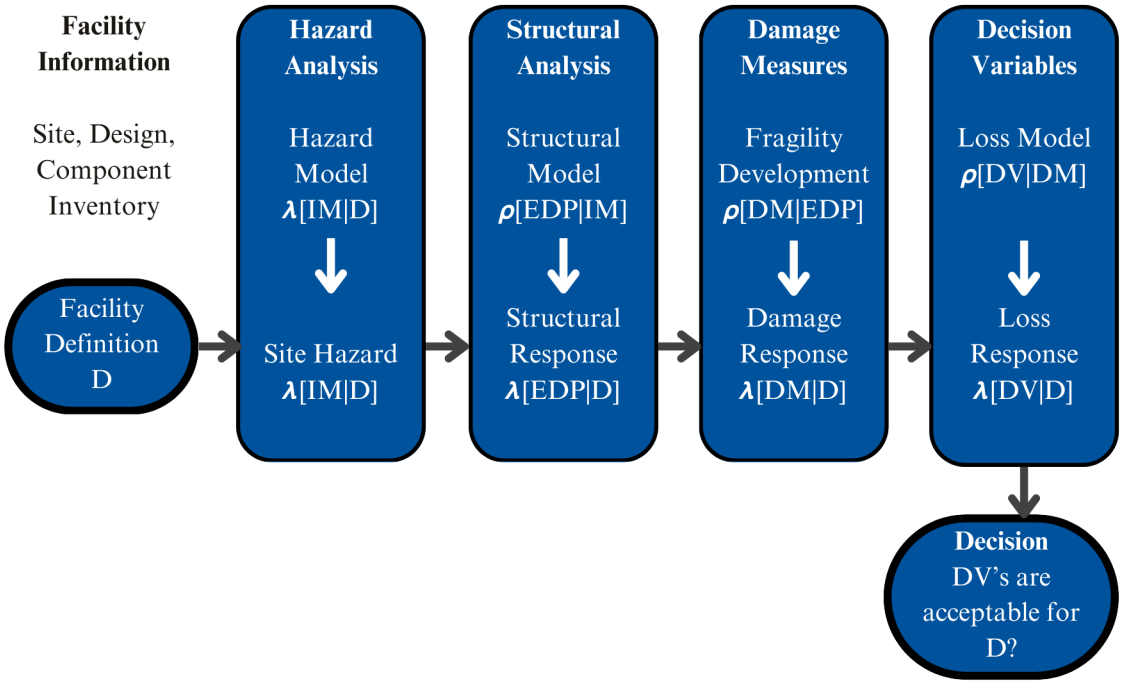


Figure 1.1: Overview of the PEER PBEE framework

simulation of the structure, required for the SA. The problem is expanded if uncertainties in future earthquakes and constitutive material parameters are considered [Parida et al., 2020, Parida, 2019, Zaker Esteghamati and Flint, 2021]. To mitigate this computational burden and facilitate swift decision-making, surrogate models can be utilized for computing engineering demand parameters (EDPs). Typically, a surrogate model seeks to establish a computationally economical mathematical function that directly links the inputs of the finite element model to its outputs, thus skipping the expensive nonlinear dynamic analysis.

This simple computational mapping found in surrogate models can replace the finite element model in the structural analysis phase of the PBEE, enabling the rapid acquisition of EDPs while accounting for uncertainties in both material parameters and future earthquakes. Frequently, this mapping function is formulated using fundamental principles of structural dynamics in conjunction with Newtonian mechanics. Such physics-based methodologies are extensively employed in

earthquake engineering practice owing to their inherent physical robustness and intuitive nature [Guan et al., 2021]. However, these approaches often rely on numerous simplistic assumptions, potentially diminishing the accuracy of response predictions. In contrast, data-driven approaches construct the underlying mathematical mapping function through training on a substantial dataset of input-output pairs [Panagiotis G. Asteris and Nikoo, 2019, Nguyen-Minh et al., 2011].

In recent research endeavors, machine learning techniques such as logistic regression (LR), decision trees (DT), random forest (RF), support vector regression (SVR), artificial neural networks (ANN), and others are have been used within the civil engineering sphere as effective data-driven surrogate models for forecasting EDPs. Their appeal lies in their straightforward implementation and capacity to capture significant levels of non-linearity [Xie et al., 2020b, Asteris et al., 2022]. This arises from multiple factors, such as adaptability, affordability in assessment, a wide array of well-established techniques, and notably, widespread accessibility facilitated by third-party toolboxes integrated into programming environments, all contributing to practical applicability [Koziel and Pietrenko-Dabrowska, 2022].

Surrogate models based on LR, RF, and SVR were employed in [Ataei and Padgett, 2015] for assessing the fragility of deck unseating failure in coastal bridges. Mangalathu and Jeon [2019a] utilized RF to create and refine bridge-specific fragility curves, while ANN [Mangalathu et al., 2018] was employed to establish multidimensional seismic fragility for single and two-column bent box-girder bridges. Segura et al. [2020] adopted a polynomial response surface surrogate model to develop fragility surfaces for the efficient seismic evaluation of gravity dams. Additionally Hwang et al. [2021], employed machine learning techniques to boost algorithms utilized for predicting responses in reinforced concrete frame buildings. These studies, along with numerous similar works [Möller et al., 2009, Kocamaz et al., 2021, Wu and Jahanshahi, 2019, Perez-Ramirez et al., 2019, Ahmed et al., 2022, Kiani et al., 2019], demonstrate the diverse applications of machine learning in structural engineering and demonstrated that once the machine learning model is properly trained, it can replicate a non-linear finite element model output accurately.

It is important to acknowledge that a significant computational investment is required initially to train machine learning models for replicating the nonlinear behavior of structures [Al-Jarrah et al., 2015]. However, the utility of these methods lies in their ability to rapidly estimate nonlinear predictions and damage assessments once trained. Nevertheless, most studies in the literature train data-driven surrogate models for specific sets of earthquakes. When faced with new, “unseen" earthquakes not included in the original training data, the surrogate model must be retrained, leading to additional computational costs. This challenge is partly due to the limited availability of high magnitude earthquake datasets [Xie et al., 2020b]. Unlike fields such as data science and computer science, where datasets are typically much larger, earthquake engineering datasets, particularly those containing recordings of large-magnitude earthquakes, are scarce. Consequently, training these machine learning models can be particularly challenging, especially given their numerous model parameters that require estimation.

Addressing the “small-data" challenge can involve employing data augmentation techniques [Shorten and Khoshgoftaar, 2019], which generate new data with similar characteristics to the available dataset. This transforms the “small-data" issue into a “big-data" scenario, facilitating more effective training of machine learning models. Several studies, as referenced in Refs. [Gidaris et al., 2015, Kyprioti and Taflanidis, 2021], utilize Kriging surrogate models in conjunction with stochastic ground motion models to predict nonlinear structural responses. These models use uncertain parameters from both ground motion and finite element models as inputs to predict structural responses. However, while this approach effectively tackles the “small-data" problem associated with retraining, it relies on stochastic ground motion models, which are typically based on empirically derived equations. This reliance may restrict their applicability, especially for structures of greater complexity, size, and nonlinearity. Additionally, the performance of Kriging models heavily depends on the specific variogram chosen to capture spatial structure [Cressie, 2015], making them less robust for modeling complex nonlinear behaviors. Importantly, when faced with an “unseen" earthquake that is not present in the original training dataset, it becomes challenging

to extract corresponding parameters for the stochastic ground motion model. As a result, there is no deterministic inverse relationship between earthquake time history and stochastic ground motion model parameters, hindering the reuse of surrogate models for predicting responses to such "unseen" earthquakes. Other studies, such as those cited in [Zhang et al., 2019, Ahmed et al., 2022] utilize advanced neural network architectures like Long Short-Term Memory cells to predict seismic responses while considering earthquake uncertainty. These studies augment small historical earthquake datasets by scaling ground motions using approaches like incremental dynamic analysis [Zhang et al., 2019] or by adding white noise [Ahmed et al., 2022]. However, these approaches involve complex networks that are computationally intensive to train. Moreover, surrogate models do not account for material parameter uncertainty, potentially limiting their applicability to specific parameter sets and requiring retraining for different scenarios. Furthermore, data augmentation techniques may not introduce sufficient variability into the training set, thereby restricting the applicability of surrogate models beyond the initially trained dataset.

To this end, this research study proposes a surrogate modeling framework based on data augmentation that:

i) Transforms a "small data" problem into a "big data" problem, allowing machine learning models to be trained for better generalization performance.

ii) Evaluates and selects the best-performing model from a set of machine learning models as the surrogate.

iii) Validates the model with unseen earthquakes not included in the training set to ensure robustness.

The organization of this thesis is as follows. Chapter 1 motivates the reader to better understand the importance of the research work presented in this thesis. Chapter 2 introduces the surrounding literature on ML models, the application of ML in the earthquake engineering field,

and explanations of specific techniques and processes used throughout this study. Chapter 3 explores various machine learning models to capture the non-linear dynamic response of structures in terms of EDPs, using traditional ground motion characteristics and material property values. Chapter 4 proposes a machine learning-based surrogate model framework utilizing SVD-enabled data augmentation. A representative suite of far-field ground motions recorded on a firm rock site was selected as the dataset. Using SVD, an orthonormal basis was identified that spans the space of the ground motion suite. The basis vector weights, assumed as random vectors, along with the constitutive parameters as random variables, were used to generate a large set of synthetic earthquakes and constitutive parameters. These synthetic inputs were fed into a finite element model. The resulting data, comprising the randomly generated weights, material parameter values, and finite element model outputs, were used to train machine learning models. Various models, including DNN, Support Vector Regression (SVR), Decision Trees (DT), and RF, were used to map the input (weights of the basis vectors and constitutive model parameters) to the output (finite element model response). Lastly, In Chapter 5 conclusions from Chapter 3. and 4. are drawn and possible future research has been discussed.

2 Literature Review

2.1 Surrogate Models using ML

Surrogate modeling is a technique that has gained traction in the civil engineering community. Traditional methods are often physically demanding and take a lot of time so alternative methods are often looked into. Surrogate models are used in a wide variety of fields and have also been used in civil engineering. Eslamlou et al.[Dadras Eslamlou and Huang, 2022] address the computational expenses of traditional models used in structural health monitoring. Ly et al. [2021] focuses on enhancing structure robustness by developing probabilistic-based soft computing models for predicting the load-carrying capacity of Composite-Filled Steel Tube (CFST) under uniaxial compression. Three hybrid Artificial Intelligence (AI) models, ANFIS-BBO, ANFIS-GA, and ANFIS-PSO, were developed and validated, incorporating Monte Carlo simulations to account for the variability of input parameters. ANFIS-PSO was identified as the most efficient model due to its high capability in quantifying the contribution of input variables to load-carrying capacity. This model also highlights the importance of cross-sectional geometry and material properties, as it has potential to estimating confidence intervals of mechanical behavior in composite members under axial loading. In Hung and Thang [2022], the use of temporal deep learning-based methods is considered for predicting dynamic responses of structures prone to wind excitation as a way to reduce computational time and maintain high accuracy of performance. This framework was verified with a case study of a 9-story RC frame structure that was able to reduce the time by three orders of magnitude and have an accurate result for maximum top-floor displacement compared to the FEM model. This framework suggests that surrogate models are practical for real, large-scale scenarios where calculations for reliability, sensitivity, or parametric analysis become too complicated and computationally expensive.

The development of these models has also led to identifying key areas of performance and research that need to be considered and developed. Al Kajbaf and Bensi proposed a surrogate

model for coastal storm surge hazard assessment and wanted to understand the performance of ANNS, Gaussian Process Regression (GPR), and SVR for predicting storm surge [Al Kajbaf and Bensi, 2020]. They found that the use of physically-motivated parameter scaling and more accurate features to inform surrogate models should be used in order to gain more accurate results and provide complete information about the performance of the surrogate models. The set of data these models are being trained on could be influential to results. Hariri-Ardebili and Mahdavi [2023] looked at the use of surrogate modeling for concrete strength prediction, it was suggested that the Kriging regression model was the primary choice of the surrogate models to predict the mechanical properties of concrete and asphalt mixtures with a performance that does better than 85% of the other soft computing algorithms. This study also suggests that further research is needed to confirm performances for other concrete databases as only one output parameter was considered in the study. Next different machine learning models used in engineering for surrogate modeling are described briefly.

2.2 Review of machine learning algorithms

Machine learning is categorized into two primary types: supervised learning and unsupervised learning. In supervised learning, the algorithm is trained with known labels, allowing it to learn from provided answers. This prior knowledge is utilized during training. Within unsupervised learning, algorithms are further divided into classification and regression based on the nature of the output [Kong et al., 2020]. Supervised learning can be subdivided into classification and regression, depending on the nature of the data (discrete or continuous) and the objectives of the task. Similarly, unsupervised learning encompasses clustering and dimensionality reduction methods. Figure 2.1 summarizes the two types of ML and some commonly used ML algorithms [Kong et al., 2020].

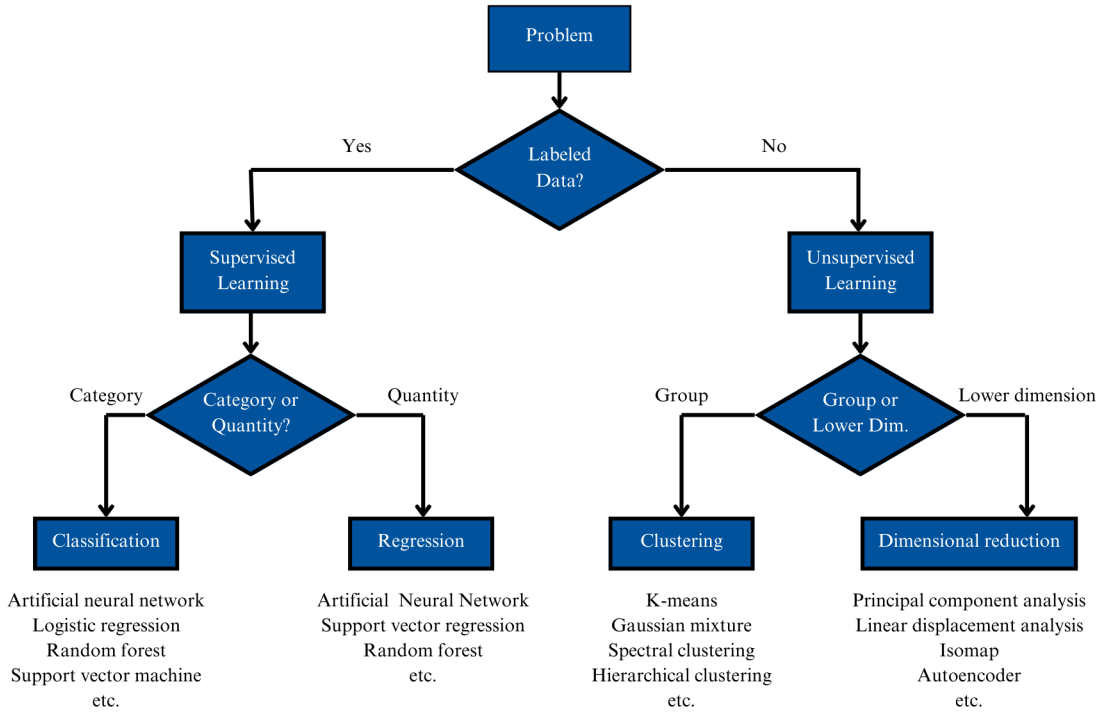


Figure 2.1: ML types and some commonly used ML algorithms [Kong et al., 2020]

The surrogate modeling problem is essentially a supervised regression problem therefore the ML models that need to be focused on are ANN, SVR, RF, and Decision Tree (DT)

2.2.1 Artificial neural network

An Artificial Neural Network (ANN) consists of interconnected artificial neurons, which are designed to mimic the action of biological neurons. ANNs are not easily defined, but they can be compared to a black box with multiple inputs and outputs, operating using a large number of parallel connected simple arithmetic units [Zupan, 1994]. The emphasis is on the network structure rather than the individual neuron's operation [Dongare et al., 2012]. In most applications, networks typically consist of three fundamental types of layers: input, hidden, and output. The input layer receives data either from input files or directly from electronic sensors in real-time scenarios. The output layer transmits information either to external systems, secondary computer processes, or

other devices like mechanical control systems. Situated between these layers, numerous hidden layers exist. These internal layers contain many interconnected neurons arranged in various structures [Maind and Wankar, 2014]. The inputs and outputs of each hidden neuron are simply routed to other neurons within the network. Figure 2.2 illustrates the setup of the basic ANN model. It can be used for tasks such as sample selection, classification, clustering, and making predictive models. ANNs are quite flexible for adaptation to different types of problems and can be custom-designed to almost any type of data representation. Within the existing literature, predictive modeling problems using ANNs are the most relevant within the engineering field [Abiodun et al., 2018].

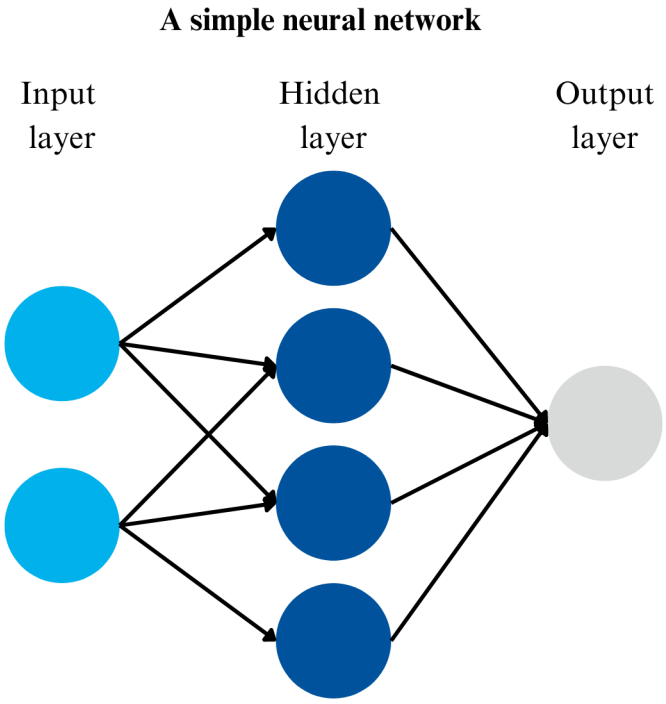


Figure 2.2: Simple neural network structure [Maind and Wankar, 2014]

2.2.2 Support vector regression

A Support Vector regression (SVR) is a supervised machine learning algorithm primarily used for regression tasks [Xie et al., 2020b]. The fundamental concept involves discovering a hyperplane

that accurately divides the d-dimensional data into its respective classes [Boswell, 2002]. However, as real-world data is often not linearly separable, SVMs introduce the concept of a “kernel-induced feature space" to transform the data into a higher-dimensional space where separation is achievable. Typically, this transformation could pose computational challenges and increase the risk of overfitting [Jakkula, 2006]. Some generalized steps to use SVM for classification and regression analysis is to first prepare the pattern (feature) matrix required. While classification and regression have different matrices, after they are prepared the data can be further partitioned into training, testing, and validation sets. Next, a kernel function is selected based on the degree of nonlinearity and parameters are selected to best suit the dataset. The algorithm is then trained with the input and output data which will define the optimal hyperplane between classes. Finally, unseen data is classified or predicted based on these factors, with errors traced back to feature extraction, kernel selection, or parameter estimation, prompting iterative refinement for enhanced accuracy [Gholami and Fakhari, 2017]. A generalized schematic of SVM in training and testing processes can be seen in Figure 2.3.

2.2.3 Decision tree and Random forest

A decision tree is a supervised learning algorithm used for classification and regression tasks in machine learning. It works by recursively partitioning the data into subsets based on the values of input features, creating a tree-like structure of decision nodes and leaf nodes [Ali et al., 2012]. An overview of the ML model’s structure can be found in Figure 2.4. At each decision node, the algorithm selects the feature that best splits the data into homogeneous subsets, typically using metrics. This process continues until a stopping criterion is met, such as reaching a maximum depth or minimum number of samples in a node [Fratello and Tagliaferri, 2018]. Decision trees are interpretable and intuitive, as they mimic human decision-making processes. However, they can be prone to overfitting, especially when the tree grows too deep, capturing noise in the data. To mitigate this issue, techniques like pruning or using ensemble methods like Random Forests are

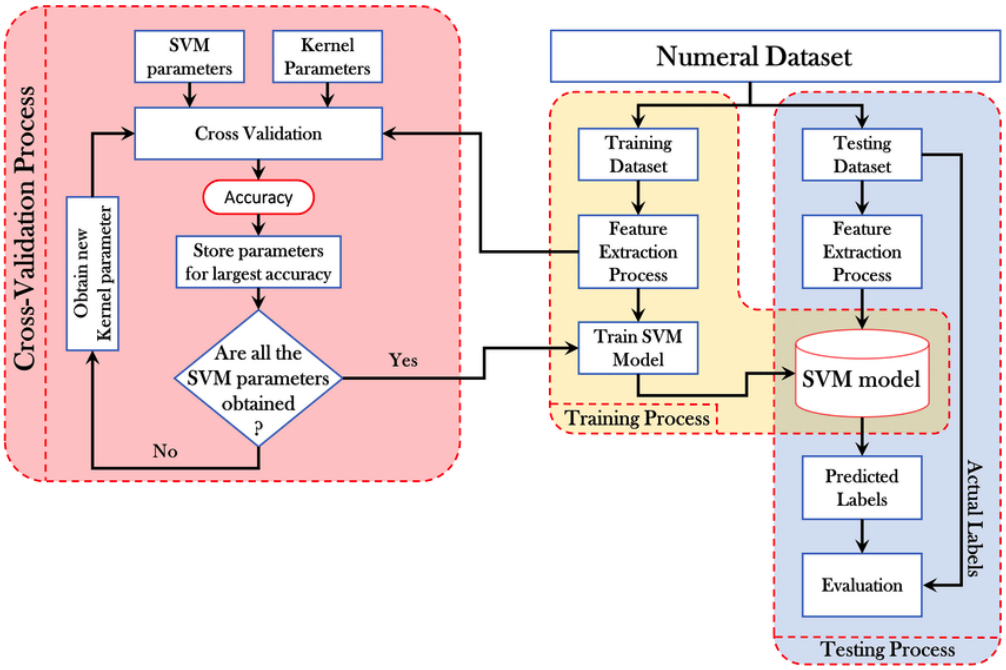


Figure 2.3: Support vector regression (SVR) schematic diagram for training and testing process [Sarraf Shirazi and Frigaard, 2021].

often employed [Speiser et al., 2019].

Random Forest is a learning technique that builds multiple decision trees during training and combines their predictions to improve accuracy, robustness, and avoid overfitting. An overview of the ML model's structure can be found in Figure 2.5. Each tree in the forest is trained on a random subset of the training data (bootstrap samples) and a random subset of the input features. This randomness introduces diversity among the trees, reducing the risk of overfitting and increasing the model's generalization ability [Louppe, 2014]. During inference, the predictions of individual trees are aggregated through averaging (for regression) or voting (for classification) to produce the final output [Kulkarni and Sinha, 2013]. Random Forests are highly flexible and perform well on a wide range of datasets without requiring extensive hyperparameter tuning. They are also resistant to overfitting and outliers due to the averaging effect of multiple trees.

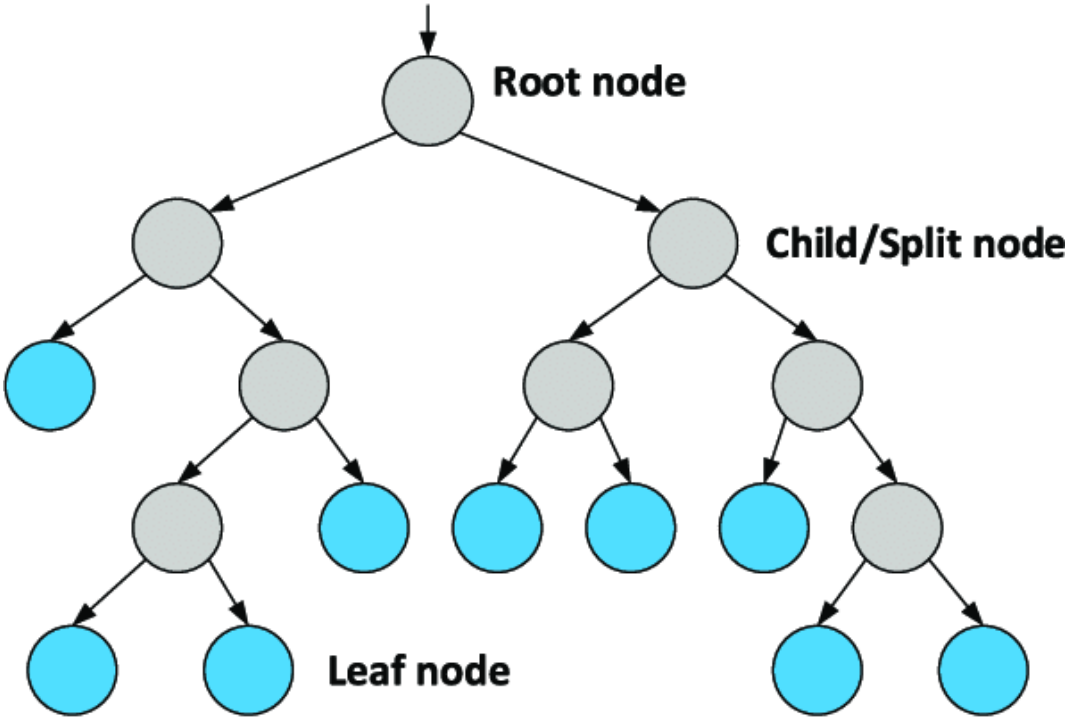


Figure 2.4: Structure of decision tree example [Camana et al., 2020].

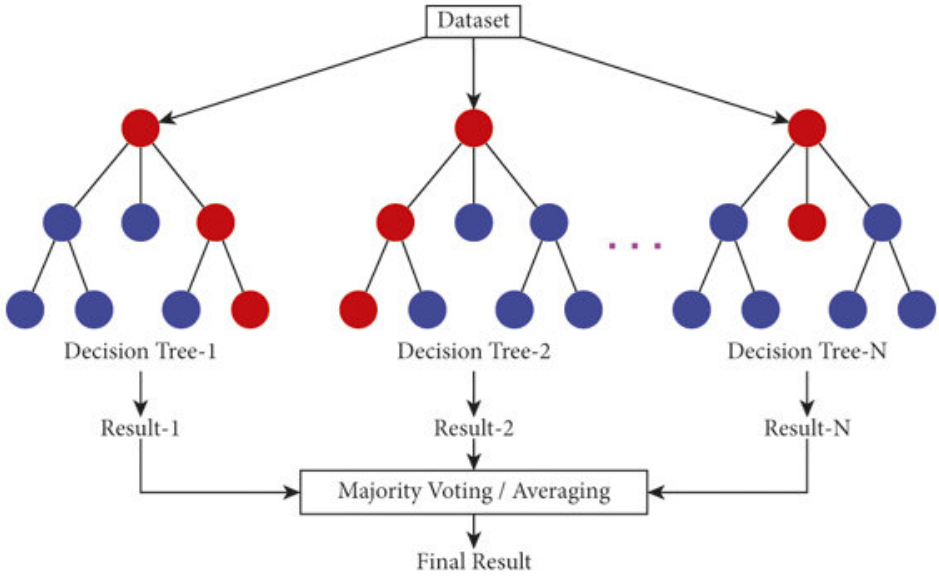


Figure 2.5: Diagram of random forest example [Khan et al., 2021].

2.3 ML-based surrogate modeling in Earthquake Engineering

These surrogate models have been used in many different applications but within the earthquake engineering community, there have been instances of surrogate models used to spearhead ground motion prediction and generation, damage detection, and seismic fragility assessment.

Ground motion prediction and generation are important components of assessing seismic risk for civil structures. Traditional empirical approaches have depended on regression analyses to establish attenuation equations. These equations relate various measures of ground motion intensity to factors such as source characteristics, distance traveled, and site conditions. [Anbazhagan et al., 2013, Atkinson and Boore, 2006]. When abundant data is available, machine learning techniques generally outperform these conventional linear regression models. Among these methods, Random Forest (RF) stands out for its superior prediction accuracy. However, linear regression remains valuable when data is limited, as its equations are based on established physical principles [Khosravikia and Clayton, 2021]

Surrogate models used in damage detection focus on creating ML models that identify, categorize, and evaluate seismic damage in civil structures. Vibration-based damage identification methods can be classified into three domains according to vibration parameters: time domain, frequency domain, and time-frequency domain approaches. Time domain methods rely on time-history responses, while frequency domain methods utilize modal parameters. Time-frequency domain techniques leverage time–frequency analytical tools. Regarding algorithms, damage detection methods can be divided into non-model-based or data-driven approaches and model-based methods[Hou and Xia, 2021]. A notable surrogate model within damage detection is Convolution Neural Networks (CNNs) as there is no manual extraction of features, meaning the the raw signals can be used as input without the need to pre-process [Avci et al., 2022].

Seismic fragility analysis is a methodology used to quantify the vulnerability of civil structures to seismic events. It assesses the likelihood of structural damage or failure at varying levels

of ground shaking intensity. Traditional methods for seismic fragility analysis, include the safety factor method, numerical simulation method, regression analysis, and maximum likelihood estimation. These methods are used to construct fragility curves, which are essential for evaluating the vulnerability of structures to seismic events [Zentner et al., 2017]. These methods and additional methods such as Intensity Measures (IMs) and Incremental Dynamic Analysis (IDA) demand sophisticated modeling processes, often reliant on high-speed computers, are commonly employed to evaluate collapse states. To expedite seismic risk assessment, innovative methods have been developed using machine learning algorithms [Kazemi et al., 2023]. In a particular study, the use of three different training algorithms for Artificial Neural Networks (ANN) in the context of fragility assessment of reinforced concrete buildings is discussed. The study compares their performance based on mean square error (MSE) and found that the ML models produced very similar fragility curves as were obtained using numerical modeling [Rasheed et al., 2022].

2.4 Data Augmentation and Feature Extractions

2.4.1 Fourier's Transformation

The Fourier Transform (FT) converts a time-domain function into a frequency-domain function, revealing the different frequency components present in the original signal [Kido, 2014]. Mathematically, the FT of a continuous function $f(t)$ is defined as $F(\omega)$, where ω represents the angular frequency [Giron-Sierra, 2017]. This transformation is expressed by the integral:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t}\, dt \tag{2.1}$$

where $e^{-i\omega t}$ serves as the kernel of the transform, incorporating complex exponential functions that oscillate at different frequencies. The result $F(\omega)$ is a complex-valued function, providing both amplitude and phase information of the frequency components. To retrieve the original time-domain function from its frequency-domain representation, the inverse Fourier Transform is

used, given by:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} \, d\omega \tag{2.2}$$

This operation reconstructs $f(t)$ by summing up all the frequency components $F(\omega)$ modulated by $e^{i\omega t}$. For discrete signals, the Discrete Fourier Transform (DFT) is used, and it can be efficiently computed using the Fast Fourier Transform (FFT) algorithm. The DFT of a discrete signal $x[n]$ is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-i\frac{2\pi}{N}kn} \tag{2.3}$$

where N is the number of samples, $x[n]$ is the signal in the time domain, and $X[k]$ represents the frequency-domain coefficients [Smith, 2007]. The DFT provides a way to analyze the spectral content of discrete signals, which is crucial in various applications such as signal processing, audio analysis, and communications. The Fourier Transform and its discrete counterpart, the DFT, are indispensable in modern science and engineering, providing deep insights into the frequency characteristics of signals and enabling efficient processing and analysis techniques.

2.4.2 Singular Value Decomposition

Singular Value Decomposition (SVD) is a fundamental matrix factorization that decomposes a matrix into three simpler matrices, providing valuable insights into the properties and structure of the original matrix. Let's denote a matrix A as having dimensions $m \times n$. This matrix represents a suite of m earthquakes with n times steps. The SVD of the matrix A can be represented as:

$$A_{n\times m} = U_{n\times m} S_{m\times m} V^T_{m\times m} \tag{2.4}$$

where U is an $m \times m$ orthogonal matrix with its columns representing the left singular vectors of A. S is an $m \times m$ square diagonal matrix with the singular values of A. These values are arranged

in descending order. V^T is the transpose of an $m \times m$ orthogonal matrix V, with its rows representing the right singular vectors of A. In order to obtain an alternate representation for columns of A to be projected onto the U matrix, $S_{m\times m}$ and $V^T_{m\times m}$ can multiplied to obtain:

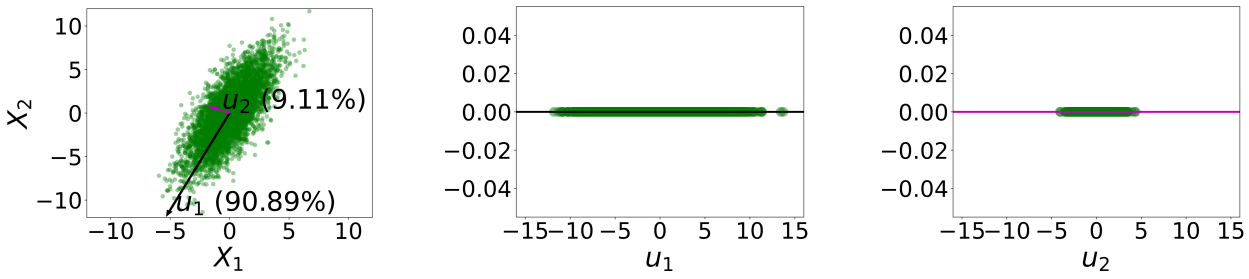
$$A_{n\times m} = U_{n\times m}\sum_{m\times m} \tag{2.5}$$

where _$m \times m$ is the product of $S_{m\times m}$ and $V^T_{m\times m}$ matrices. The columns of $\sum_{m\times m}$ represent the weights of each earthquake in the suite. This means to reproduces the *i*th earthquake in the suite the *i*th column of the $\sum$ matrix would need to be multiplied into the U basis matrix. This produces an m-dimensional encoding for each earthquake as represented by the columns of $\sum$, which can serve as a feature representation for input into a machine learning model. To construct new earthquake data, one can generate m-dimensional vectors with random weights and utilize them to generate the corresponding time histories. Additionally, any new earthquake's feature representation $P_{n\times 1}$ that the machine learning model is tasked with predicting, can be projected onto the U basis.

$$\sigma_{m\times 1} = U^T_{m\times n}P_{n\times 1} \tag{2.6}$$

where $\sigma_{m\times 1}$ a weighted vector of the new earthquake's feature representation. It is important to note that the larger and more diverse the U basis matrix is the better the weighted vectors that are being produced will turn out. A smaller U basis could result in less unique datasets that are being produced as it is only able to relate to the original suite used to create the U basis matrix.

In Figure 2.6, 1000 random samples in 2 dimensions that exhibit clear directional features or correlation are considered. After performing SVD on this 1000 × 2 matrix, we observe that the direction of the first column of U (represented as u_1) captures almost the entire variation in the dataset (91%). The direction of the second column of U (u_2) captures the remaining variance. For many practical applications involving this dataset, the first column of U alone can serve as a



(a) (b) (c)

Figure 2.6: (a) The directions of first 2 columns of U (represented u_1 as and u_2) capturing the entire variance of a random bivariate dataset. Projection of the data set on (b) u_1 preserves the maximum variation of the dataset while (c) u_2 captures the remaining small variation.

sufficient uni-variate representation.

2.4.3 Discrete Wavelet Transform

Discrete Wavelet Transform (DWT) is used as a multi-resolution analysis tool in signal processing and data compression. The earthquake data can be assumed to be generated from a function $f(t)$, one can represent it in terms of basis functions, $\phi(t)$ and $\psi(t)$, which can be scaled to give multiple resolutions of the earthquake data function. The Jth scale representation of the time-dependent earthquake function $f(t)$ can be written in terms of orthogonal basis function families that are generated by father $\phi(t)$ and mother $\psi(t)$ wavelets as seen below:

$$f(t) = \sum_k c_{a_{J,k}} \phi_{J,k}(t) + \sum_{j=1}^{J} \sum_k c_{d_{j,k}} \psi_{j,k}(t) \tag{2.7}$$

where k is the number of coefficients, $c_{a_{J,}}$ and $c_{d_{j,}}$ are the approximation and detailed coefficients at a specified scale. The father and mother wavelets generate $\phi_{J,k}(t)$ and $\psi_{j,k}(t)$ by scaling and shifting as:

$$\phi_{J,k} = 2^{-J/2} \phi(\frac{t - 2^J k}{2^J}), \tag{2.8}$$

$$\psi_{j,k} = 2^{-j/2}\psi(\frac{t-2^j k}{2^j}), j = 1,...,J \tag{2.9}$$

The shift parameter in the numerator can be represented as $\alpha = 2^j k$ and the scale parameter is $\beta = 2^j$. These parameters are responsible for manipulating the size and shape of the wavelets for example when α is increases the spread of the wavelet increases and the height decreases. These parameters apply a specific wavelet such as Daublets, Symmlets, Haar, etc. through the entire dataset at a given level.

Using 2.7 on the earthquake dataset the DWT can produce the approximate and detailed coefficients as:

$$c_{a_{J,k}} = n^{1/2}\sum_{t=1}^{n} f(t)\phi_{J,k}(t), \tag{2.10}$$

$$d_{a_{j,k}} = n^{1/2}\sum_{t=1}^{n} f(t)\psi_{j,k}(t) j = 1,...,J \tag{2.11}$$

where n is the number of points within the earthquake dataset. These approximate and detailed coefficients represent weights that describe each wavelet that goes toward the earthquake data function $f(t)$. The approximation coefficients represent the overall "structure" in the dataset at a specific level and the detailed coefficients are finer and higher frequency behavior. In order to successfully conduct a multi-scale feature extraction scheme the approximation coefficients at every level are more conducive to use. The scale J is also an important factor to consider. A high-scaled representation of the data results in very coarse or smooth features and a low-value representation results in features corresponding to noise in the data. Because these features will be used to train ML models, it is important to select the correct scaled representation of the data and will become a hyperparameter to adjust within the training period.

2.4.4 Auto-encoders

Auto-encoders are a tool used in data processing, particularly where feature extraction is needed. The initial dataset is pre-processed and split into training and testing sets. In the auto-encoder architecture, both the encoder and decoder components are constructed using convolutional layers within a neural network framework [Lange and Riedmiller, 2010]. An overview of the ML model's structure can be found in Figure 2.7. To determine optimal performance, various configurations, including both 1D and 2D setups, are tested to determine the most suitable arrangement for the dataset.

In the encoder, an initial setup of the convolutional layers is established, and a desired latent space h is determined [Zhai et al., 2018]. It should be noted that the dimension of the latent space was tested to see if that has any affect on the reconstruction results. To systematically evaluate performance, a range of setups are tested and compared. A tabulated summary of these different configurations aids in assessing their efficacy in feature extraction. The training earthquake input data X is fed into the encoder, generating a latent space representation h of the dataset. The function to describe this is $h = f_{enc}(X)$ where $f_{enc}(X)$ represents the encoder's neural network. This latent space representation is then passed to the decoder, which works to reconstruct the original data from this compressed representation. The decoder's task is to construct a signal $X^{'}$ from the latent space representation. The function to describe this process is $X^{'} = f_{dec}(h)$ where $f_{dec}(h)$ represents the decoder's neural network. This constructed signal is then compared to the original dataset, aiming to minimize the loss incurred during reconstruction. During the training stage, the model iterates through the training dataset, adjusting its parameters to minimize the loss function. This involves back-propagating the calculated loss throughout the model, giving it the opportunity to refine the ability to reconstruct the input data. Training is typically conducted over multiple epochs, with the dataset size and number of epochs carefully selected to ensure robust model convergence.

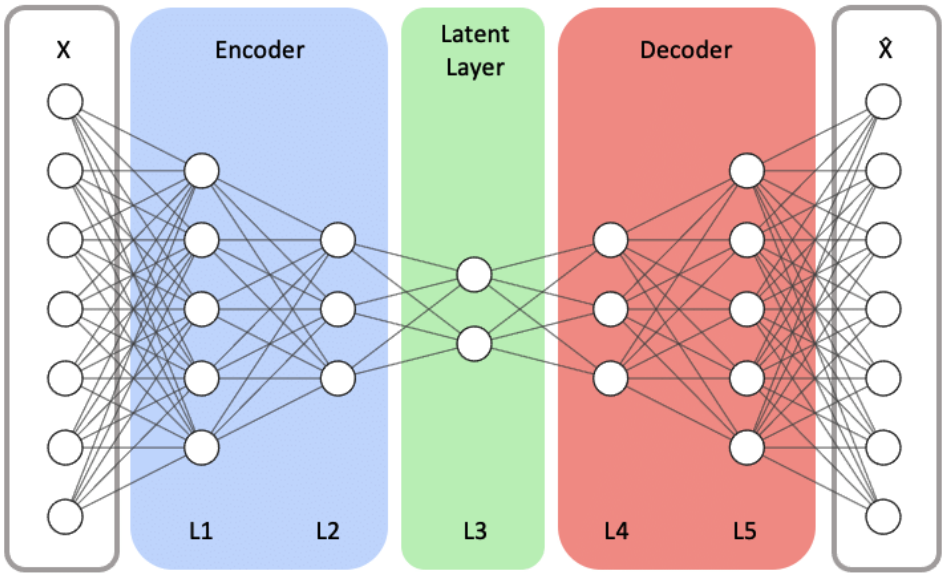


Figure 2.7: Diagram of auto-encoder [Song et al., 2021].

The objective of this auto-encoder framework is to develop an encoder capable of producing a latent space representation that can be reconstructed by the decoder. This latent space serves as a condensed feature set, which can subsequently be leveraged in various ML models for predictive tasks. The adaptability of the encoder is crucial in scenarios where additional data becomes available. The encoder must adeptly accommodate new data, generating a latent space representation that captures the features of the updated dataset. This capacity ensures the continued relevance and effectiveness of the auto-encoder in processing evolving datasets. AE can be hypothetical to obtain a condensed representation of EQ time history that can be used in ML model training

2.4.5 Generative Adversarial Networks

The Generative Adversarial Network (GAN) architecture comprises of two components: the generator G and the discriminator D [Hong et al., 2019]. An overview of the ML model's structure can be found in Figure 2.8. In this framework, the generator is provided with a random noise latent sample z from a predetermined distribution. This noise serves as the input in which the generator

constructs fake data, generating signals with the exact dimensions of the real seismic data. The generator can be represented as a function $G(z, \theta_G)$, where θ_G represents the parameters of the generator network. Simultaneously, the discriminator undergoes training to identify the characteristics of real seismic data, using the predefined dataset used for all feature extraction purposes. It takes input data x and outputs a probability $D(x, \theta_D)$ indicating the likelihood that x comes from the real data distribution. The θ_D represents the parameters of the discriminator network. At every iteration, it evaluates whether the presented data is real or fake, and then provides feedback to its own decision-making architecture and to guide the generator's learning.

The generator operates in a feedback loop, working to improve its ability to deceive the discriminator. Since real earthquake data is never seen by the generator, it relies solely on the discriminator's feedback to adjust its synthetic output. The training process of GANs involves optimizing the following minimax objective function:

$$min_G max_D V(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[log(1-D(G(z)))] \tag{2.12}$$

where $p_{data}(x)$ is the distribution of the earthquake data, $p_z(z)$ is the distribution of the random latent space noise. $D(x)$ is the output of the discriminator when given real earthquake data while $D(G(z))$ is the output of the discriminator when given fake earthquake data. The generator's proficiency level will improve over iterations to a point where the discriminator is deceived into perceiving synthetic signals as authentic. In reaching this state, the discriminator iteratively refines its discriminative abilities, adapting to the evolving capabilities of the generator. Both the generator and discriminator architectures are constructed with convolutional layers within the framework of neural networks.

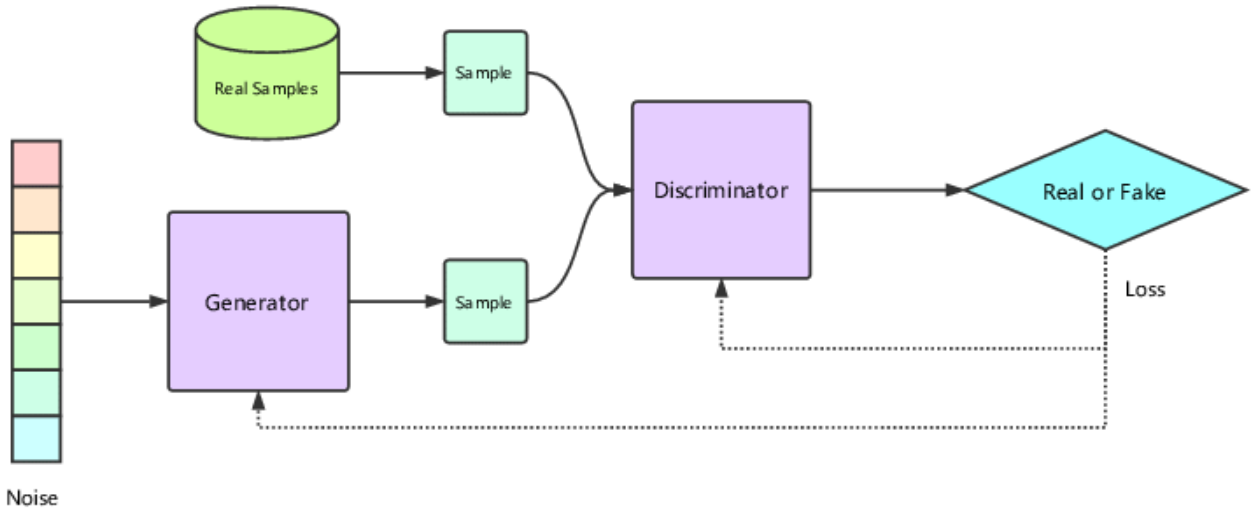


Figure 2.8: Diagram of GAN setup [Gu et al., 2018].

To understand the optimal performance of the GAN, various setups, including both 1D and 2D configurations, are tested to identify the most accurate arrangement for the dataset under consideration. The objective of this GAN framework is to train the generator to a proficiency where the discriminator is unable to reliably distinguish between real and synthetic earthquake data. Once this threshold is achieved, the generator becomes a tool for generating diverse and novel earthquake datasets. By leveraging predefined latent spaces or noise distributions, an abundance of synthetic earthquake data can be produced and used as training and testing datasets for ML models. This approach not only adds to the quantity of available data but also enhances the capabilities of earthquake-related ML applications.

3 Surrogate models using existing features to predict EDPs

The work presented in this chapter builds upon prior research and was originally presented at the 12th National Conference on Earthquake Engineering in 2022 [Parida et al., 2022]. This study introduces a novel surrogate modeling framework utilizing machine learning to address the challenges of repeated training and uncertainties in future earthquakes and material parameters. The aim is to have a surrogate model in place for the structural analysis portion of the Performance-Based Earthquake Engineering (PBEE) framework in order to be cost effective. This approach effectively tackles issues including:

a) Identifying a which traditional features are suitable to represent earthquake data and can serve as input for training ML models.

b) Resolving the “small-data" problem by generating the traditional earthquake features from the historical dataset to have a variety of training data.

d) Identifying the most suitable ML model for predicting EDPs.

3.1 Training Data

The structures observed in this study consisted of a 1-story and 2-story building, which were modeled as idealized mass-damper systems. OpenSEES was utilized to create single-degree-of-freedom (SDOF) and two-degree-of-freedom systems (2-DOF) for the structures [McKenna et al., 2010]. The Steel01 material in OpenSEES was employed to represent uniaxial bilinear steel material with kinematic hardening and no isotropic hardening. More details on Steel01 can be found in Chapter 4. To capture the damping behavior of the material, a combination of a nonlinear spring and an elastic material element with specified damping stiffness was considered, and the representative lumped mass at each story was treated as a whole.

Building Type	Forcing Function			
	Single Harmonic		Two Harmonics	
	Input	EDP	Input	EDP
One Story (SDOF)	PGA, A_1, ω_1, E, F_y, ζ	$D_{\max}$	PGA, ω_1, ω_2, A_1, A_2, E, F_y, ζ	$D_{\max}$
Two Story (2-DOF)	PGA, A_1, ω_1, E, F_y, ζ	D_{max} ISD	PGA, ω_1, ω_2, A_1, A_2, E_1, F_{y1}, ζ_1, E_2, F_{y2}, ζ_2	D_{max} ISD

Table 3.1: Input parameters and corresponding EDPs for training ML models

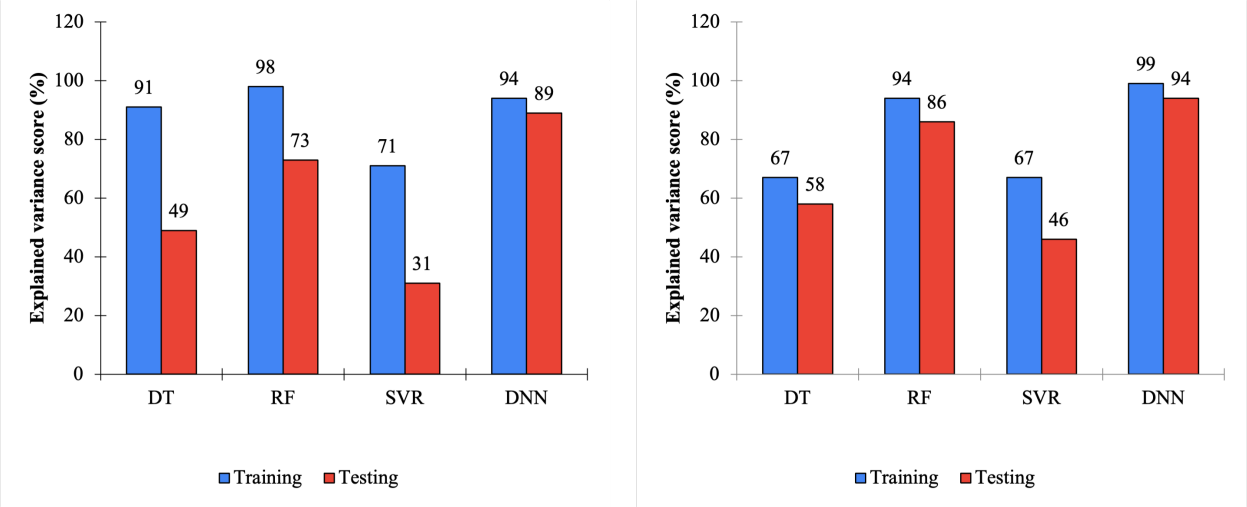
The ML models typically require a large set of input data, so to provide a wide range of data, the constitutive model parameters were represented as uniform random variables, and the range of realizations came from their probability density functions (PDFs). To ensure large nonlinear deflections, the mean elastic modulus (E) was chosen such that the natural angular frequency of the structural systems was 1s, the mean of F_y was 0.5% of the elastic modulus, and the mean of ζ was 5%. These values fell within the typical range for these types of structures. A coefficient of variation (COV) of 30% was chosen to generate the realization of the model parameters. For the 2-story (2-DOF) scenarios, the mean value of model parameters was assumed to be the same as the first, but the set of parameters for the second story was considered a separate set of random variables. This resulted in a set of model parameter values that differed for the first and second story in each realization. The use of single and two harmonic excitations was considered in the FE model. To ensure that some of the realizations would have resonance in the structure, the mean of cyclic frequency (ω_1) was taken as the first natural frequency of the structure for single harmonic cases, and the mean of the second harmonic (ω_2) was chosen to be equal to the second natural frequency of the two-story building for the two harmonic cases. The forcing functions for the realizations used a 30% COV. The Engineering Demand Parameters (EDPs) relevant to this specific topic were peak roof displacement (D_{max}) and inter-story drift (ISD). The inputs and the output for each case were summarized in Table 3.1, where PGA represented the peak ground acceleration, and A_i denoted the amplitude associated with the peak frequency of the forcing function.

3.2 Selection of ML Model

Four competing ML models discussed in the previous chapters: Decision Trees (DT), Random Forest (RF), Support Vector Regression (SVR), and Deep Neural Network (DNN)s [Géron, 2022] will be used and systematically select the best model based on their performances. Further information on how these ML Models work can be found in the literature in previous chapters. It is important to understand that the hyper-parameters within these models have a great affect on the model output. Within the DNN model, the number of hidden layers, number of neurons in each layer, activation function, etc. are the hyperparameters that needs to be tuned to ensure that the performance of the model is good enough for training and testing. To best see if the models were tuned accurately, the data that was generated was split into training and testing in a 75:25 ratio. Each ML model was trained with the training data and the performance scores for both training and testing data were observed for different hyper-parameter configurations. To capture an interpretable score to determine performance explained variance ratio [Géron, 2022] and R^2 El-Sayed et al. [2023] was used as the metric. The best model was chosen based on the performance of the testing data set.

To avoid overfitting the models, the model that had the best performance based on the testing data set was selected as the model best suited. Figure 3.1 shows two examples of different hyper-parameters configurations for the models considered in both one-story and two harmonic forcing cases. Figure 3.1(a) shows a situation where overfitting is prevalent since it does well on training data but not as good within testing data while Figure 3.1(b) is a better balance between training and test data scores. Looking at RF for example, in Figure 3.1(a), the RF consists of an ensemble of 150 trees resulting in better training performance compared to the ensemble of 100 trees in 3.1(b). The difference is that the RF performs better with the testing case in Figure 3.1(b), so the hyper-parameters of Figure 3.1(b) have better performance than Figure 3.1(a) within the performance metric that has been set. It should be noted that the positive change in performance plateaus at

about 15,000 data points.



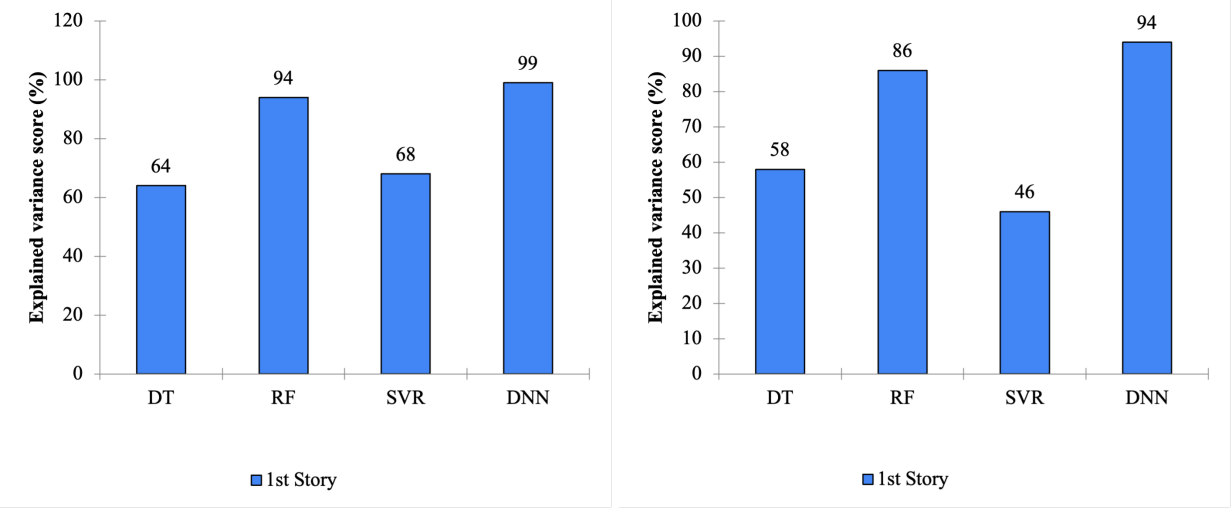
(a) (b)

Figure 3.1: (a) Example of bad performing ML model (b) Example of ML model with superior performance

3.3 Results

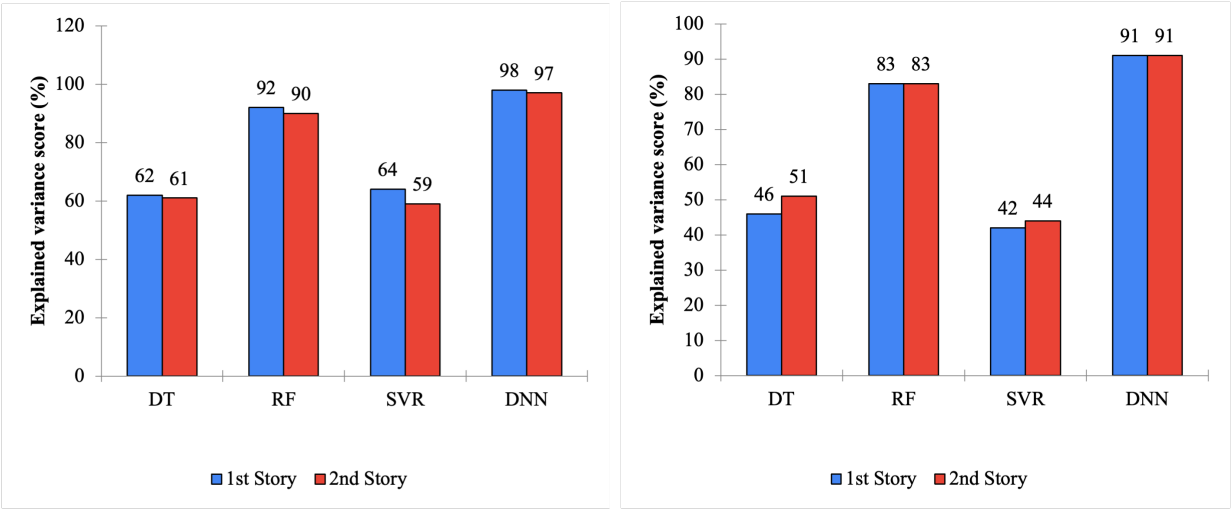
The performances of the ML models chosen are shown below in Figure 3.2. In order to compare the best configurations for each chosen ML model, iterations of tuning hyper-parameters were conducted till it resulted in a suitable setup for each ML model. This concept was previous discussed where the “poor" DNN in Figure 3.1(a) had a single layer of 50 neurons with ReLU activation functions and a “good" DNN had three layers of neurons each consisting of 75 neurons.

After finalizing the tuning, it can be seen in Figure 3.1(b) that DNN performs the best, followed by random forest, support vector regression and decision tree, in that order, for all the cases considered in this study. Comparing single harmonic and two harmonic excitations, there is a decrease in model performance that can be observed. Feature importance was conducted in order to understand this decrease in performance that was occurring using the random forest model [Mangalathu and Jeon, 2019b]. From this, the excitation function, PGA, peak frequencies,



(a) (b)

Figure 3.2: Performance of various ML models on testing data, (a) SDOF system with single harmonic forcing, (b) SDOF system with two harmonic forcings.



(a) (b)

Figure 3.3: Performance of various ML models on testing data, (a) 2DOF system with single harmonic forcing, and (b) 2-DOF system with two harmonic forcings.

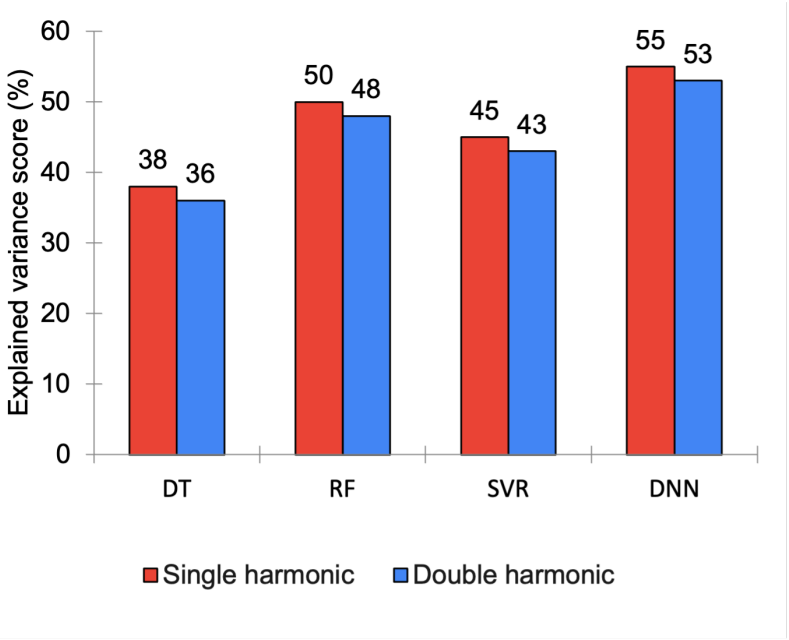


Figure 3.4: Performance of various ML models on seismic testing data, SDOF system with single harmonic and two harmonic forcings

and corresponding amplitudes were the most important features that the predictions relied on. The importance and affect listed inputs were further explored by using local differentiation to determine the sensitivity of the inputs. It was observed that several different models were highly sensitive to the change of variation in inputs for certain input intervals as compared to others.

3.3.1 Application to real EQ data

A suite of real earthquake time history that was similar to the training data was also tested. Using Fourier's Transform, the frequencies and amplitudes were determines for each realization and used as input data for the various ML models. This resulted in poor predictions ranging from 35% to 55% as seen in Figure 3.4.

The poor results from the real earthquake suite raise the necessity of exploring new ways to extract data features from these suites that can be used to accurately portray the characterization of the excitation time history in terms of parameters that can train ML models. Once an accurate

data feature extraction method can be identified and used to train ML models it can be used to predict EDPs to be used in PBEE. Additionally, the bad results might be due to the limited set of earthquake data. The means of feature extraction methods using Fourier transform cannot be used in data augmentation as it does not capture the essential part of the historical EQ set that an ML surrogate model can understand and be trained with.

In the next chapter, using SVD-enabled feature extraction will be presented as a technique that can be used in both feature extraction but also data augmentation.

4 SVD enabled data augmentation for machine learning-based surrogate modeling of non-linear structures

4.1 Systematic development of the surrogate model

The work discussed in this chapter pertains to the publication [Parida et al., 2023]. This work proposes a novel surrogate modeling framework based on ML to tackle the challenges associated with repeated training and uncertainties in future earthquakes and material parameters. The goal is to seamlessly integrate this surrogate model into the PBEE framework for cost-effective structural analysis. Essentially, this approach addresses four simultaneous issues:

a) Identifying a comprehensive set of features that effectively represent historical earthquake data and can serve as input for training ML models. Moreover, these features should be extractable from “unseen" earthquakes to predict response, establishing an inverse relationship between earthquake time histories and features.

b) Resolving the “small-data" problem by augmenting small earthquake datasets into larger ones while encompassing significant variability.

c) Determining a set of material parameters to which the finite element model output is sensitive.

d) Identifying the most suitable ML model for predicting EDPs.

4.1.1 Selecting features and generating training data

As discussed briefly at the end of the last chapter, input features used to train ML models are highly sensitive and need to accurately capture characteristics of earthquake time history and building model parameters. Once this set of input features is acceptable, different realizations of the variables are given to the FE model to produce EDPs used to determine the accuracy of the ML model’s

performance. A supervised learning method is chosen [Zaker Esteghamati and Flint, 2021, Hwang et al., 2021, Möller et al., 2009]. All input features chosen for the ML models need to be all-encompassing to predict EDPs with high degrees of accuracy. These inputs should have the ability to be used in data augmentation that can be used to create realizations for the FE model to use and produce the corresponding EDP's outputs. Traditional Earthquake input features include peak frequency, corresponding Fourier amplitude, peak ground acceleration (PGA), peak ground velocity (PGV), pseudo-spectral acceleration, Arias intensity, and spectral moment, among others [Kramer, 1996, Bose et al., 2019]. These single-value input features were explored in the previous chapter and was found that these features cannot generate earthquake time-series data for different realizations of input parameters. An inverse relationship does not exist between the intensity measures and earthquake time-series data which is a vital relationship needed for data augmentation. This inverse relationship is imperative to generating enough training data for the ML model which is a topic that is discussed in the cited literature previously. As concluded in the previous chapter the traditional intensity measures of earthquake characterization often lead to ML models that are highly sensitive to the input parameters so this has driven the need to consider alternative techniques for ground motion characterization. The method used to extract these data features is singular value decomposition (SVD). Details on this data augmentation process can be found in the previous chapters.

4.1.2 Selecting optimal suitable ML models

As discussed in previous chapters, tuning the hyper-parameters is a task that needs to be meticulously conducted as it can significantly affect the surrogate model's accuracy and efficiency. The ML models that will be considered are Decision Trees (DT), Random Forest (RF), Support Vector Regression (SVR), and Deep Neural Networks (DNN). The framework of this work can also allow other ML models to be interpreted if appropriate but these 4 models are within the scope of this work. A brief description of the models can be found in the previous chapters while a detailed

description of these models can be found in [Géron, 2022, Goodfellow et al., 2016, Hastie et al., 2009, Xie et al., 2020a, Asteris and Mokos, 2020, Lu et al., 2020]. With these models, the dataset will be divided in training and testing sets. The optimal values for the hyperparameters will be determined for each ML model, using cross-validation in DT, RF, and SVR [Géron, 2022, Ng, 1997]. Cross-validation is the method of training a ML model on a subset of the training data and evaluating its performance on the subset of training data (cross-validation set) that the ML model has not seen. By cross-validating multiple times, the set of parameters that yield the most optimal and generalized model can be produced [Mohri et al., 2018]. For DNN, a heuristic approach is is the most widely used and accepted method to determine the number of neurons and layer combinations [Mitropoulou and Papadrakakis, 2011]. Once the optimal ML models are selected and trained, the performance of these models can be assessed using the testing data.

4.1.3 Validation of selected ML model

The ML models are validated using testing data. The quality of the testing dataset is important as it should consist of input-output ordered pairs that have not been seen by the ML model. A successful validation from the ML model allows it to be deployed within the PBEE framework. Unsuccessful validation of the ML model requires the input set, Θ, to be altered with a higher number of features or a set of higher-quality features that will retrain the ML model. The framework for this section is illustrated in Figure 4.1 and the application of this framework will be discussed in the upcoming sections.

4.2 Results and discussion

4.2.1 1-story and 3-story building FE models

The most detailed and accurate seismic analysis can be found in a three-dimensional FE model as the seismic demands at a component level for beam and column members can be observed if

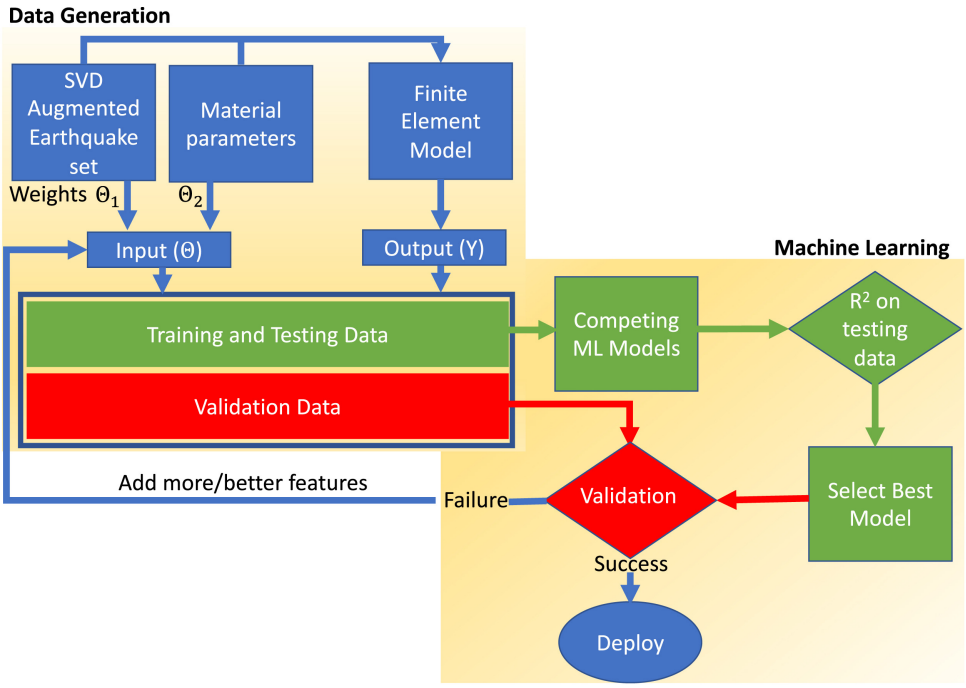


Figure 4.1: Flowchart illustrating the proposed framework.

distributed plastic elements are used. For typical seismic analysis, the global seismic response and performance of the structure are most important as they have correlating EDPs such as the story drifts and floor accelerations. The use of a simplified model can accurately capture the nonlinear response of the structure in terms of drifts and acceleration. A one-story and three-story nonlinear frame buildings are considered for the proposed framework. A simplified nonlinear spring–mass–damper model in OpenSees [McKenna et al., 2000] is used as the structure. These models are also referred to as "stick" models (see Figure 4.2) and are used widely to perform nonlinear seismic analysis [Roh et al., 2013, Gaetani d'Aragona et al., 2021, Liu et al., 2012]. These springs are assumed to be an uniaxial bilinear steel material with kinematic hardening while isotropic hardening is not considered in this study without the loss of generalizing. This is implemented using the Steel01 material model in OpenSees [Menegotto, 1973]. The Steel01 material model in OpenSEES is an isotropic elastic-perfectly plastic model commonly used to simulate the behavior of steel in structural analysis. It assumes linear elastic behavior up to the yield point,

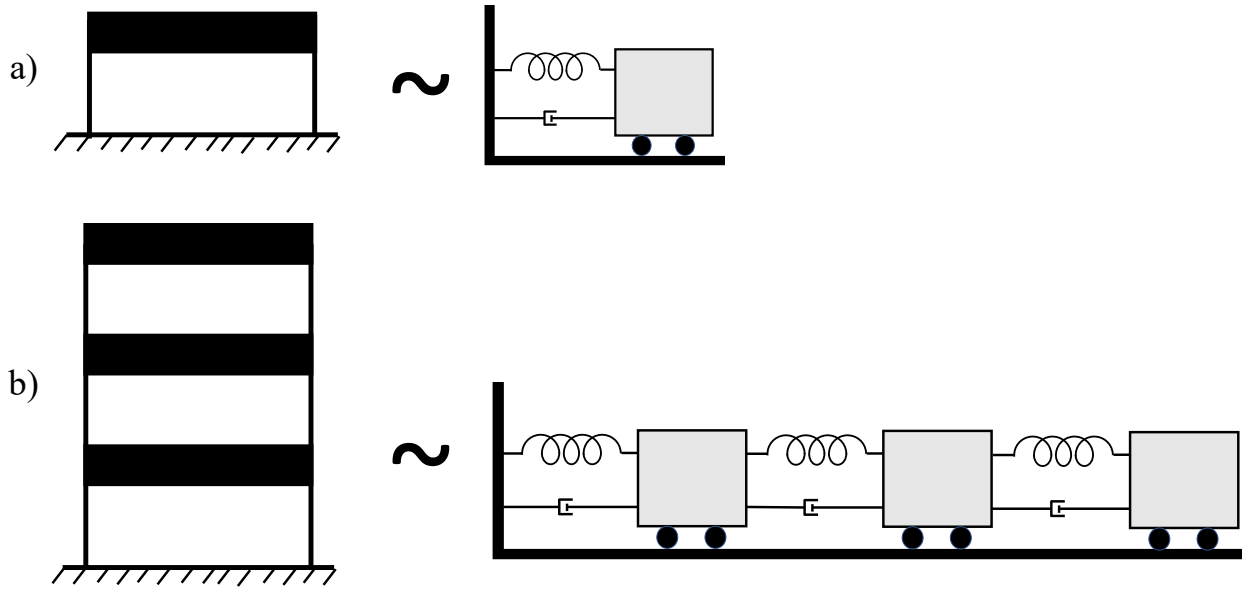


Figure 4.2: Approximation of a) Single bay one story shear frame as 1Dof non-linear spring mass damper system and b) single bay three-story shear frame as 3Dof non-linear spring mass damper system.

followed by perfect plastic deformation beyond yield. Figure 4.3 describes the stress-strain curve for Steel01 material. The material properties required for Steel01 include the elastic modulus (E), yield strength (f_y), and strain hardening ratio (b). Dampering is minimized using viscous material in conjunction with zero-length elements in OpenSEES. The objective of this work is to precisely replicate the response of these finite element models to ground motion utilizing ML techniques, given a set of constitutive model parameters and earthquake features.

4.2.2 Choice of the initial suite of ground motion

Selecting a set of appropriate ground motions for training and testing of surrogate models must encompass a wide range of historical earthquakes. To this end, 22 pairs of ground motions with magnitudes ranging from 6.5 to 7.6, recorded on firm soil (rock or stiff, 180 m/s), were pulled from the FEMA P695 far-field suite. Table 4.1 provides a list of these ground motions and their characteristics. All motions were captured at sites located 10 km or more away from fault rup-

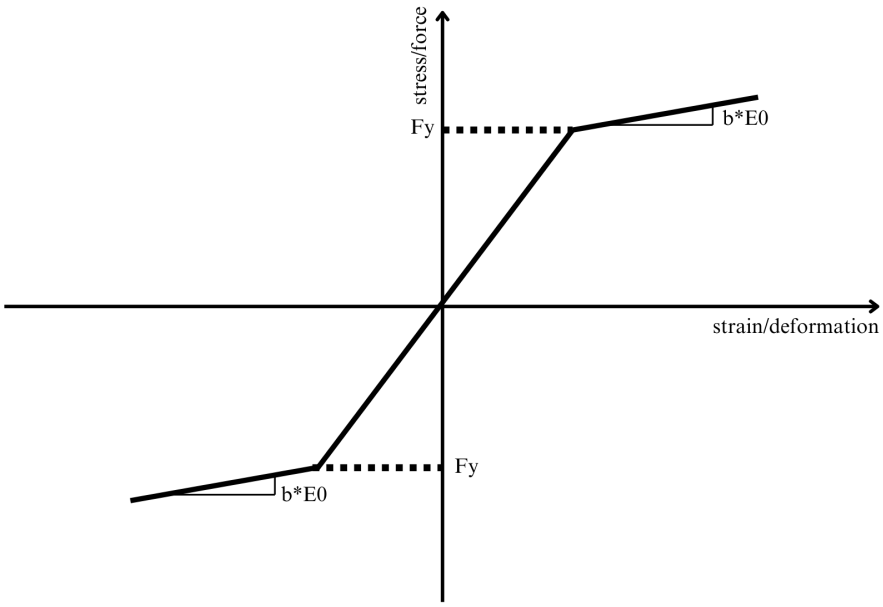


Figure 4.3: Stress-Strain curve diagram of Steel01 material.

ture (including strike-slip and reverse-thrust faults), which classifies them as "far-field" motions. To have a well-rounded representation of various recorded earthquakes, this set includes far-field records drawn from the majority of large-magnitude events in the PEER NGA database [Chiou et al., 2008]. The differences in event magnitude, source distance, source type, and site conditions of these ground motions account for the record-to-record variability required to capture uncertainties in ground motion. Further details on the criteria for record selection can be found in [ATC, 2009]. Figure 4.4 displays the acceleration time histories of the earthquakes in the suite.

4.2.3 Characterization of ground motion using SVD

The FEMA P695 far-field suite chosen was first processed to facilitate SVD analysis. This dataset has varying sampling frequencies and record lengths, initially interpolated to a standard sampling frequency and record length. The chosen sampling frequency was set at 50 Hz (time step of 0.02 s), aligning with the smallest sampling frequency of the records. The chosen time history length

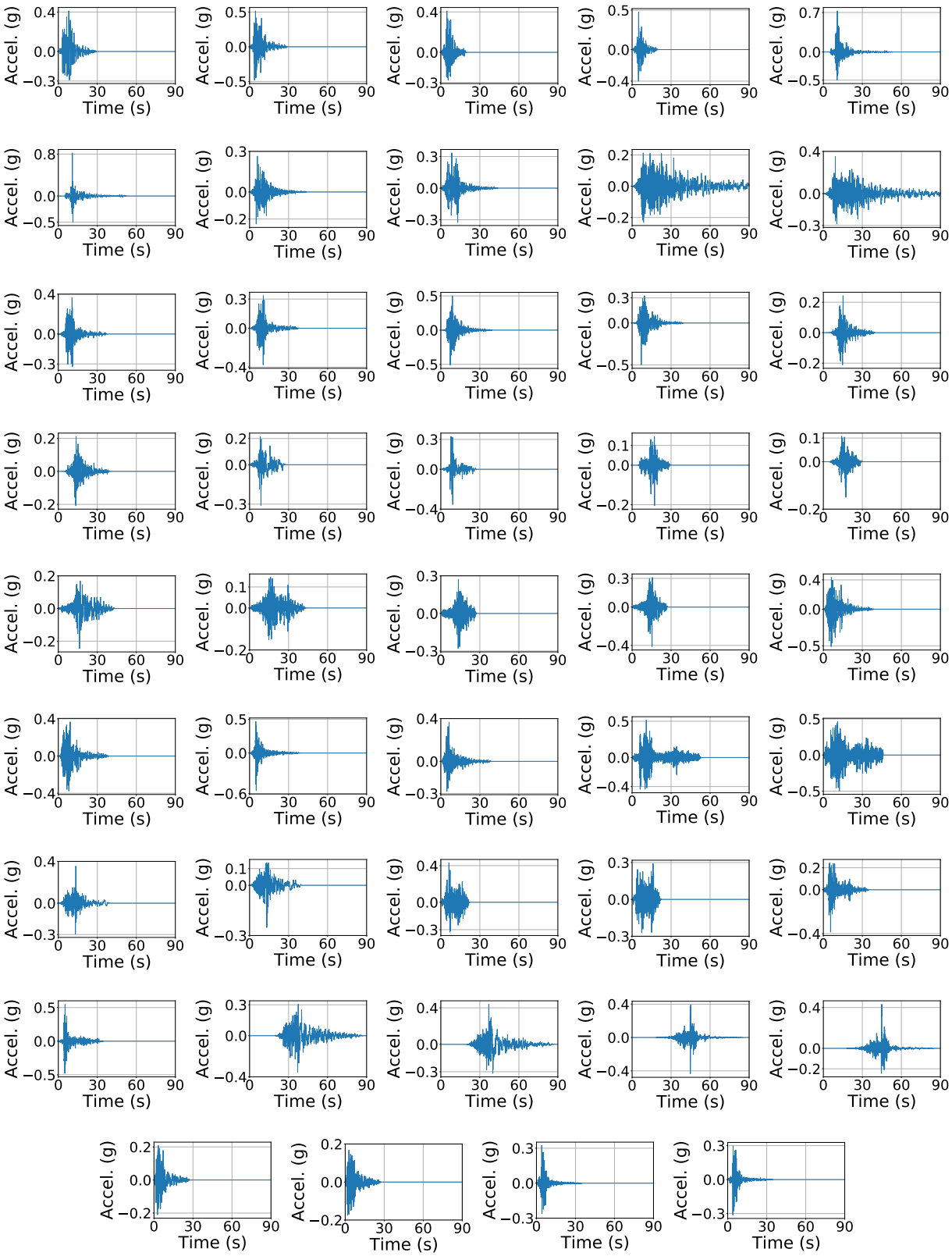
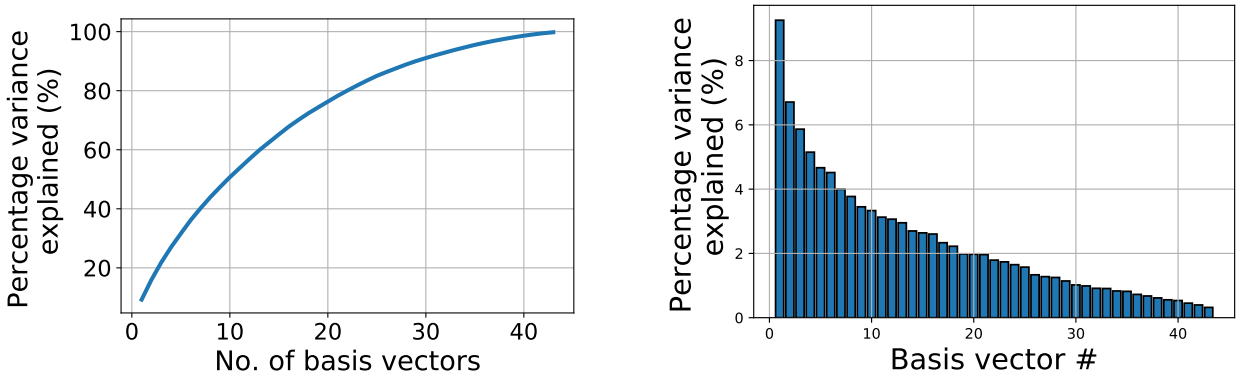


Figure 4.4: The FEMA P695 earthquake suite.



(a) (b)

Figure 4.5: (a) Cumulative percentage of variance explained by increasing the number of basis vectors used. (b) Contribution of each basis vector to the total explained variance.

was designated as 90 seconds, representing the longest record length in the suite. For data with a duration of less than 90 seconds, zero padding was used. This results in $A_{4500\times44}$ in Equation 2.5, with n, the number of time steps (4500), and m, the number of earthquakes (44). SVD was then performed on matrix A, creating an orthonormal basis matrix $U_{4500\times44}$ that has 44 columns. The basis vectors were placed in descending order of their singular values and the cumulative explained variance was plotted against the number of basis vectors in Figure 4.5a. It was determined that the first 40 basis vectors account for 99% of the variation in the ground motion data. This is illustrated in Figure 4.5b, where the percentage of explained variance is plotted against the basis vector number. The contribution of the last 4 vectors is negligible meaning only the first 40 vectors were selected as the basis vectors for the ground motion suite, resulting in a $U_{4500\times40}$ orthonormal basis matrix. These 40 basis vectors are shown in Figure 4.6. 40 basis vectors were selected to reduce the size of the U matrix but this size can be reduced even further with a concurrent reduction in the level of accuracy.

Table 4.1: FEMA P695 far-field ground motions

					PGA (g)	
ID	Event	Station	Soil	M	Major	Minor
1	Northridge, 1994	Beverly Hills - Mulhol	D	6.7	0.516	0.416
2	Northridge, 1994	Canyon County - WLC	D	6.7	0.482	0.410
3	Duzce, 1999	Bolu	D	7.1	0.822	0.728
4	Hector Mine, 1999	Hector	C	7.1	0.337	0.266
5	Imperial Valley, 1979	Delta	D	6.5	0.351	0.238
6	Imperial Valley, 1979	El Centro Array #11	D	6.5	0.380	0.364
7	Kobe, Japan, 1995	Nishi-Akashi	C	6.9	0.509	0.503
8	Kobe, Japan, 1995	Shin-Osaka	D	6.9	0.243	0.212
9	Kocaeli, Turkey, 1999	Duzce	D	7.5	0.358	0.312
10	Kocaeli, Turkey, 1999	Arcelik	C	7.5	0.219	0.150
11	Landers, 1992	Yermo Fire Station	D	7.3	0.245	0.152
12	Landers, 1992	Coolwater	D	7.3	0.417	0.283
13	Loma Prieta, 1989	Capitola	D	6.9	0.529	0.443
14	Loma Prieta, 1989	Gilroy Array #3	D	6.9	0.555	0.367
15	Manji, Iran, 1990	Abbar	C	7.4	0.515	0.496
16	Superstition Hills, 1987	El Centro Imp. Co.	D	6.5	0.358	0.258
17	Superstition Hills, 1987	Poe Road (temp)	D	6.5	0.446	0.300
18	Cape Mendocino, 1992	Rio Dell Overpass	D	7.0	0.549	0.385
19	Chi-Chi, Taiwan, 1999	CHY101	D	7.6	0.440	0.353
20	Chi-Chi, Taiwan, 1999	TCU045	C	7.6	0.512	0.474
21	San Fernando, 1971	LA - Hollywood Stor	D	6.6	0.210	0.174
22	Friuli, Italy, 1976	Tolmezzo	C	6.5	0.351	0.315

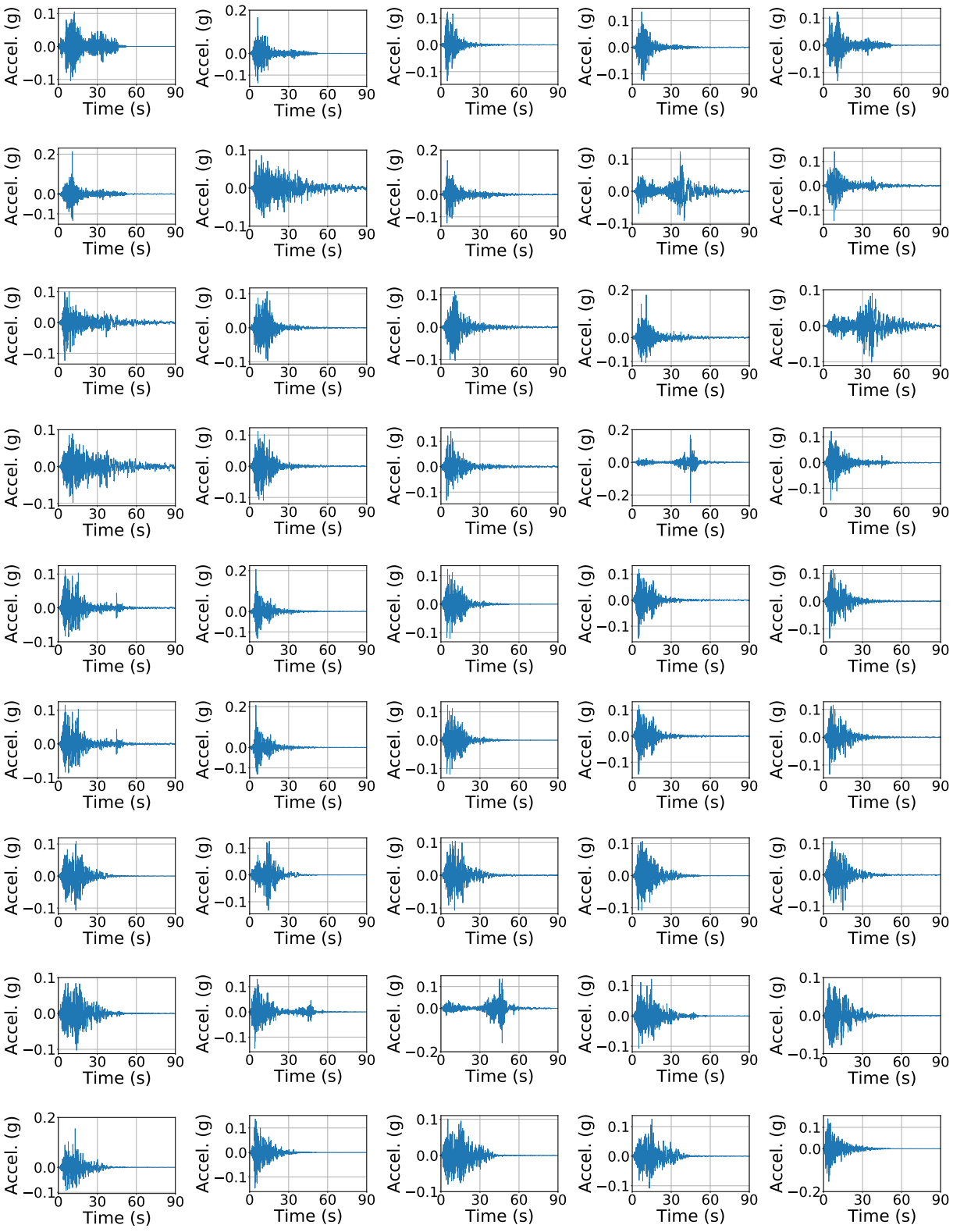


Figure 4.6: First forty basis vectors for FEMA P695 earthquake suite which are responsible for 99% of variability in the suite.Each plot represents a column of the $U_{4500\times40}$ matrix.

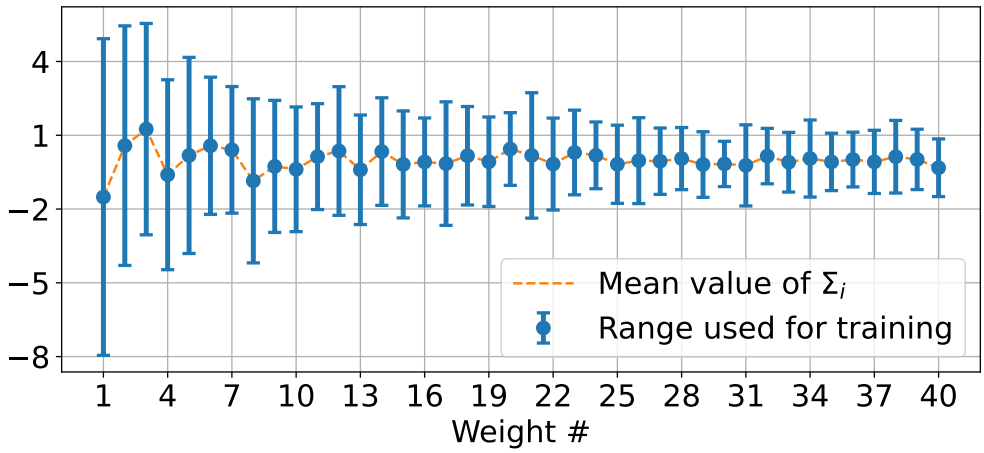


Figure 4.7: Mean values of each row of Σ along with the bounds used for creating augmented dataset.

4.2.4 Training and testing ML models

The ML models need to be trained using large sets of input-output pairs, the input being material parameters and ground motion features from the SVD analysis and the output being peak inter-story drift (ISD) and peak floor acceleration. These two output parameters are commonly used in structural engineering to represent structural and non-structural performance. Using SVD, described in the previous section, the FEMA P695 far-field ground motion suite was projected onto the U basis and produced a set of 40 weights that corresponded to each column vector in the basis and also related to the columns of the weight matrix $\Sigma_{40\times44}$ in Equation 2.5. Every column of the weight matrix Σ can be assumed to have a uniform random vector $\Theta_{1_{40x1}}$ realization that is bounded by the minimum and maximum values of the row Σ. The mean values and bounds of Θ_1 are shown in Figure 4.7. These bounds can be altered to account for a wider range of earthquake variation but this configuration was kept to reflect the data being used. With the mean values and bounds of Θ_1 known, a set of new earthquakes can be produced by randomly generating values for Θ_1 and multiplying it with U. Examples of theses generated earthquakes can be seen in Figure 4.8

The next step is to account for constitutive model parameters of lateral story stiffness (K), story yielding shear force (F_y), and equivalent viscous damping ratio (ζ) along with their respective

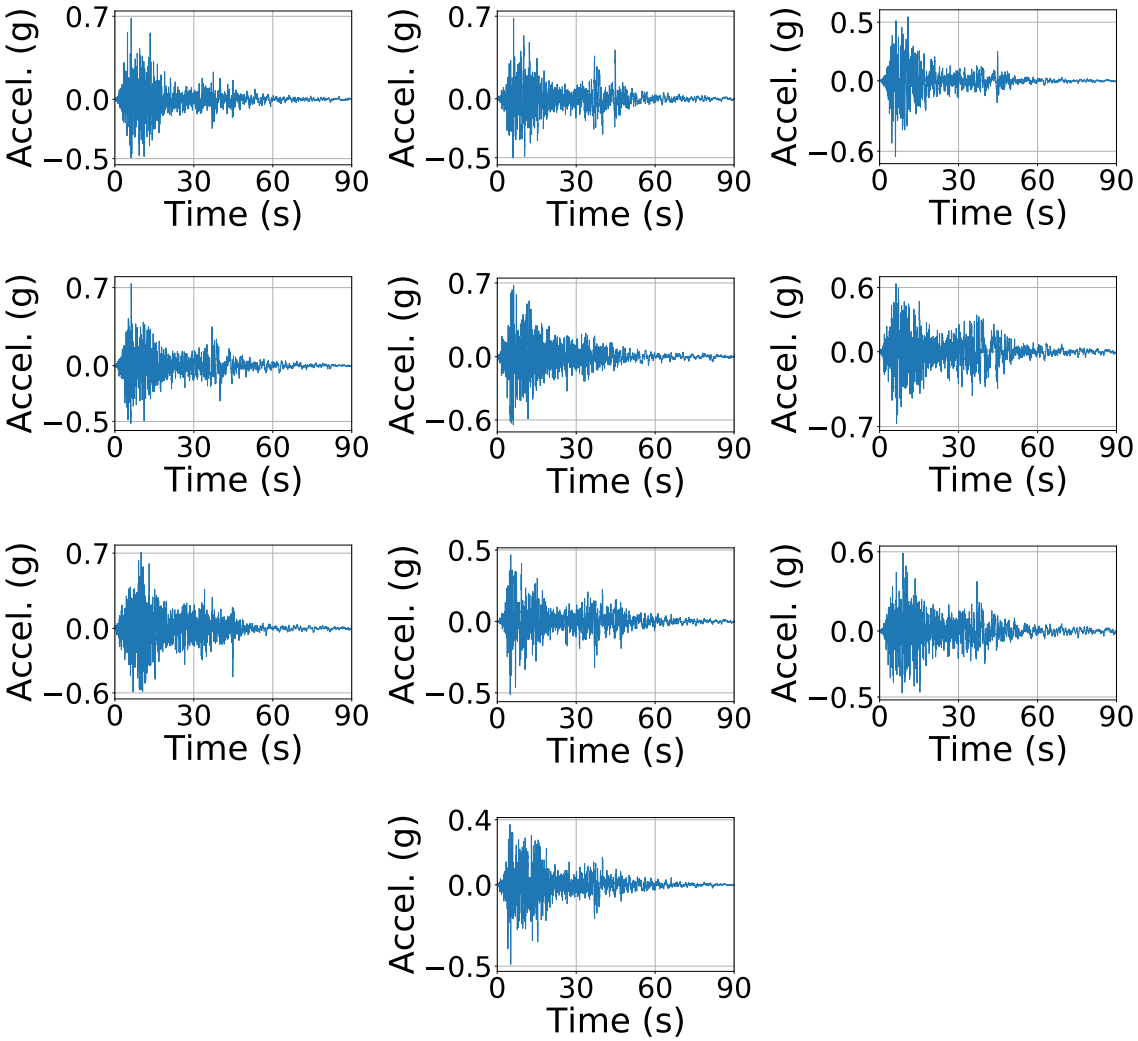


Figure 4.8: Earthquake realizations obtained by multiplying random samples of Θ_1 with U.

Table 4.2: Mean and bounds of the structural parameters.

Parameter	Mean	Bounds
Lateral story stiffness(K)	40 MN/m	[32 MN/m, 48 MN/m]
Story yielding shear force (F_y)	0.28 MN	[0.21 MN, 0.35 MN]
Damping ratio (ξ)	0.05	[0.04, 0.06]

uncertainties. These Uniform random variables were normalized by the mass of each story which is assumed to be 1.0 kN-s^2/m. By normalizing these values by the mass the diagonal mass matrix turned into the identity matrix. A mean value of $K = 40$ MN/m and a variation of $\pm$ 20% was used so that the mean fundamental period of the structure would be 1 s, which represents the fundamental period of buildings. The mean story yielding lateral force used was $F_y = 0.28$ MN with bounds that are $\pm$ 20% of the mean. This was chosen so that the response of the FE models reaches the nonlinear regime for the majority of the realizations in order to have the ML models trained for non-linearity. The mean of ζ was 0.05 with $\pm$ 20% variation. A summary of the constitutive model parameters is shown in Table 4.2.

For the ML model, the material parameter input resulted in a uniform random vector $\Theta_{2_{3x1}}$ for the 1-story model, and $\Theta_{2_{9x1}}$ for the 3-story model. The final input vector for the ML model Θ can be written as $\Theta = [\Theta_1, \Theta_2]^T$ for both the 1-story model (43 dimensional vector) and the 3-story model (49 dimensional vector).

500,000 realizations were generated from the input random vector Θ and its PDF as this number of realizations produced the best results across all ML models considered. It should be noted that a convergence study was not conducted to determine the number of points for individual ML models as it is not within the scope of the work. Next, the FE model was simulated by corresponding to the set of Θ and the structural response corresponding EDPs of peak ISD and floor acceleration were obtained. This resulted in a 2-dimensional output vector corresponding to each Θ. These simulations were run on 64 Intel i9 computing cores with 64 GB of RAM which

Table 4.3: Hyperparameter values that yield the best results for each ML model

Machine Learning Model	Hyperparameters	Values
Decision trees (DT)	Maximum depth	[100]
Random forests (RF)	Number of trees	[250]
Support vector regression (SVR)	λ and ε	[1.5,0.5]
Deep neural networks (DNN)	Number of layers and neurons in each layer	[10, 500]

took a total simulation time of around 4 hours. The force deformation plots that relate to the earthquake time histories generated in Figure 4.8 are presented in Figure 4.9

There was highly non-linear behavior evident in approximately 99% of all realizations from the earthquakes and material parameters. The 500,000 input-output data points were split into training data (90%) and testing data (10%). The ML models can now be trained on these input-output order pairs and the performance can be determined from the testing input–output order pairs. The performance metric chosen was R^2 and can be described in Equation 4.1

$$R^2(Y,\hat{Y}) = 1 - \frac{Variance(Y-(\hat{Y}))}{Variance(Y)} \tag{4.1}$$

where the numerator is the error variance between the FE model output, Y, and ML model output, $\hat{Y}$.

As mentioned earlier, to choose among the competing ML models the best set of hyperparameters for each model type must be determined. Once this is determined, the model that has the least error within its optimal model configuration is considered the best overall ML model. A cross-validation study, as described earlier, was conducted to find the best hyper-parameters for SVR, DT, and RF. This was achieved using a built-in cross-validation function within the Scikit-Learn library [Pedregosa et al., 2011]. The values of hyperparameters for each ML model are listed in Table 4.3.

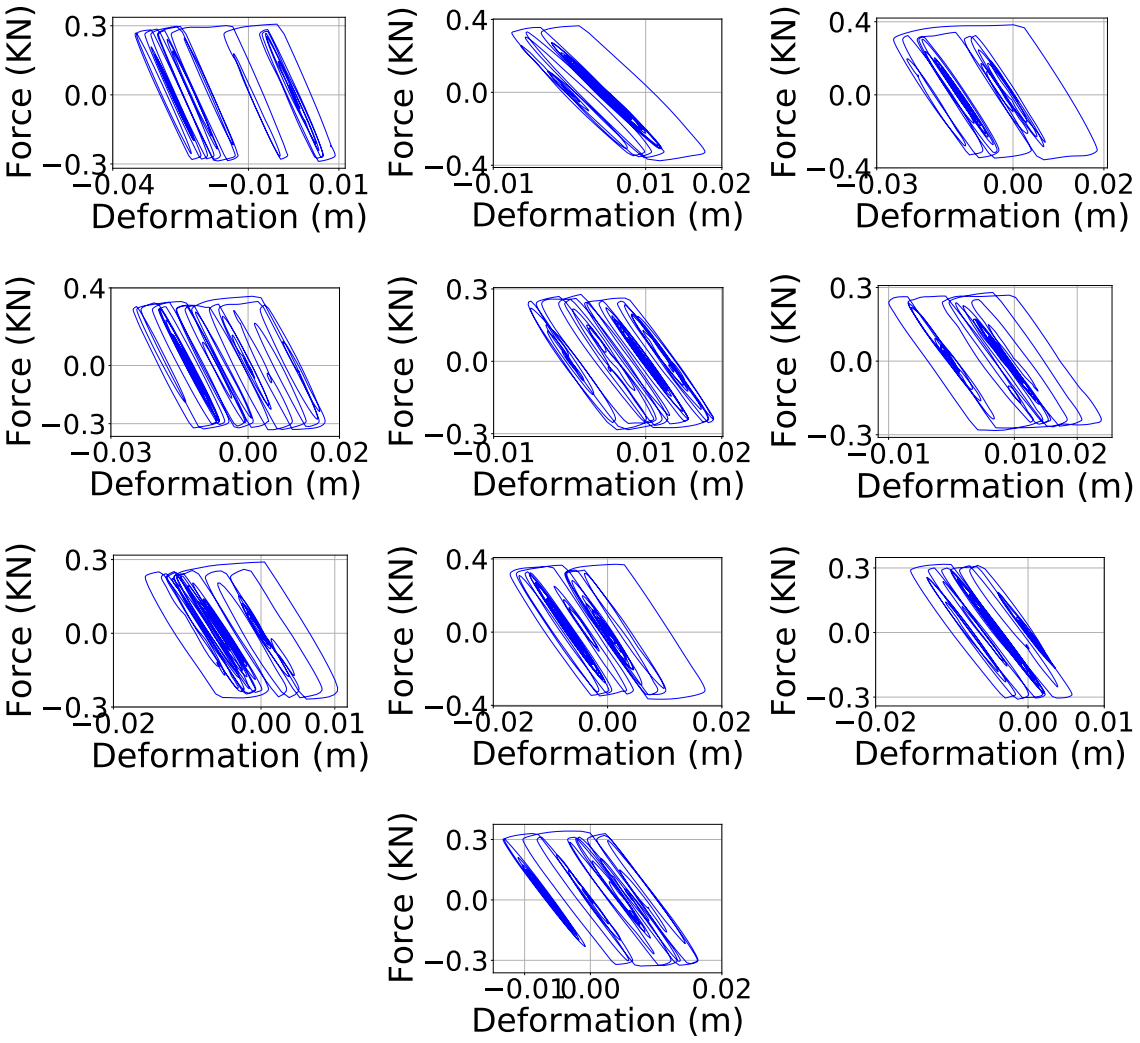


Figure 4.9: Force-deformation plots relating to realizations of earthquakes shown in Figure 4.8 and material parameters sampled from Table 4.2.

DNN hyper-parameters were chosen heuristically by varying the numbers of layers and neurons in each layer until a suitable level of performance was achieved as shown in Figure 4.10. As the number of trainable parameters increased due to varying the number of layers and neurons in each layer, the variation of error also increased. This can be seen with a DNN that has $3,634,802$ trainable parameters (10 layers with 600 neurons) compared to a DNN that has $1,527,002$ trainable parameters (8 layers with 200 neurons).

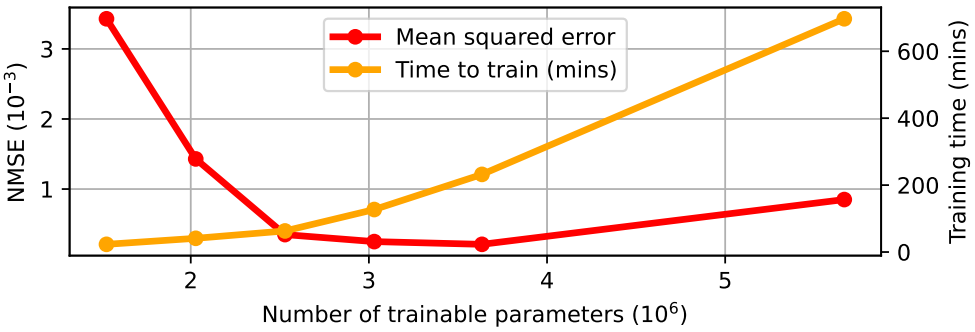


Figure 4.10: Variation of normalized mean squared error and time take for training with increasing size of DNN as measured by the number of trainable parameters in the network.

Figure 4.10 also shows how the amount of time to train the networks is related to the number of trainable parameters. It is seen generally that as the depth of the neural network increases, the error decreased while also seeing an increase in the training time needed to train the network. Therefore the optimal combination of hyperparameters for DNN was found to have 10 layers with 500 neurons in each layer, resulting in $2,529,002$ trainable parameters. After a point, increasing the depth of the network led to an increase in error which could be due to the lack of data to train such a large number of training parameters. Considering all the ML models chosen, Figure 4.11 shows the computational complexity of each ML model in terms of the time taken to train each model. The simplest structure, DT, trains the quickest, and the most complicated, DNN, takes the longest to train.

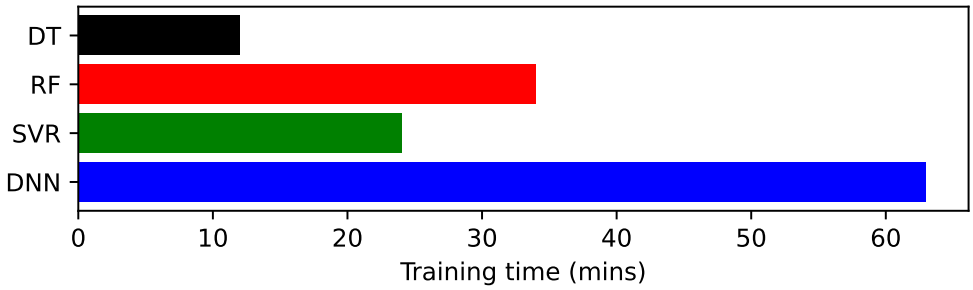


Figure 4.11: Time taken to train each ML model with the chosen optimal hyperparameters.

Figure 4.12 and Figure 4.13 show the R^2 error for each of the ML models in configuration with the best hyperparameter configuration. Figure 4.12(a) corresponds to the training error for the one-story building and Figure 4.12(b) corresponds to the testing error for the one-story building. Similarly, Figure 4.13(a) corresponds to the training error for the three-story building and Figure 4.13(b) corresponds to the testing error for the three-story building.

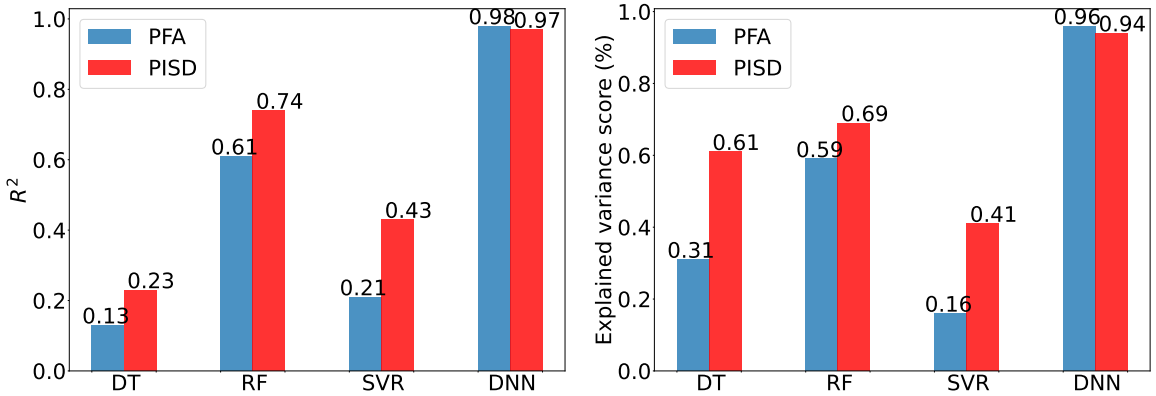


Figure 4.12: Performance of various ML models in predicting peak floor acceleration (PFA) and peak inter-story drift (PISD) in terms of R^2 for (a) training data and (b) testing data for the one-story building.

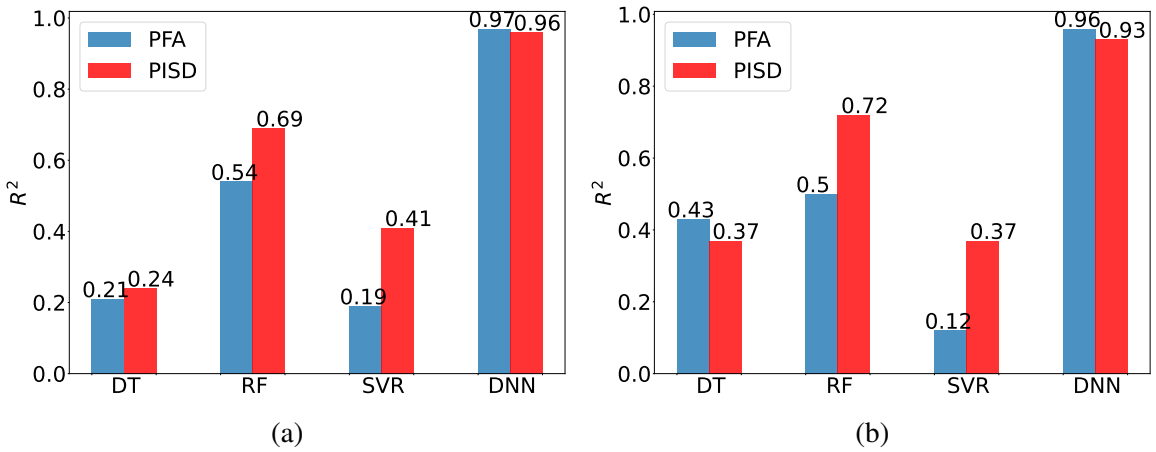


Figure 4.13: Performance of various ML models in predicting peak floor acceleration (PFA) and peak inter-story drift (PISD) in terms of R^2 for (a) training data and (b) testing data for the three-story building.

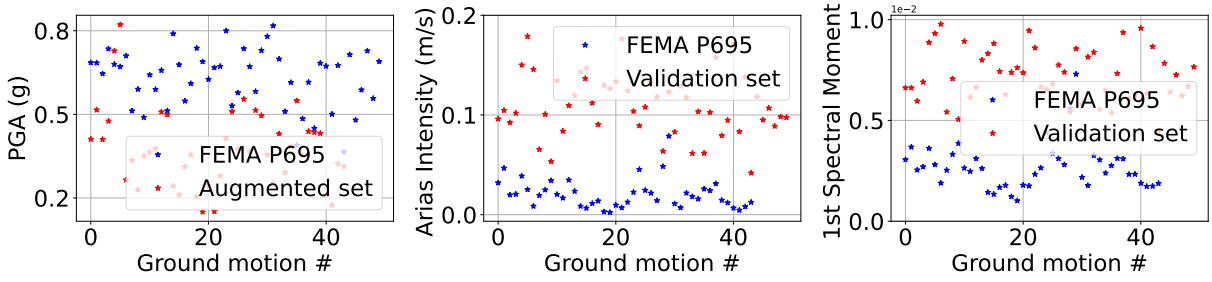
From these results, it can be concluded that DNN has the best performance in both training and testing, followed by RF, SVR, and DT in both 1-story and 3-story buildings. Due to the highly non-linear nature of the finite element response, it was expected that DNN would be able to perform better compared to the other ML models. The use of multiple layers with non-linear functions allows DNNS to process intricate data correlation. The stochastic nature of gradient-based training methods used in the DNN encourages robust generalization to unseen data instances, reducing the risk of overfitting. Note that the DNN architecture comprises 10 layers, each having 500 neurons, as the accuracy performance justifies this configuration. This is further shown by a performance comparison against DT in Figure 4.13, where DT shows a large difference in performance between training and testing errors which indicates overfitting.

Before this DNN model can be deployed to be used in the PBEE framework, it must be validated. While the testing set of data did not get used in the training sequence it did get used to select hyper-parameters of the SVR, RF, and DT models. The convergence of error for the testing set was used to stop the forward and backward propagation loops for the DNN model which have leaked information from the test set to the training set. This DNN model could also coincidentally

perform well on a particular test set. A validation process is key to being able to deploy the best-performing ML model chosen, ie. DNN, to be used in the prediction of EDPs.

4.2.5 Unseen earthquakes and parameters for validation

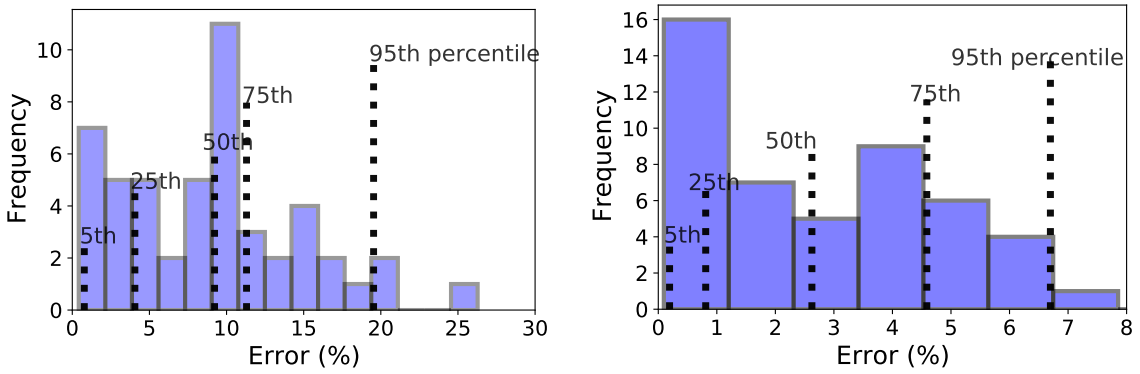
The validation process accounts for 50 ground motions (Θ_1) and 50 material parameters (Θ_2) to create 50 realizations of Θ. It is ideal for these generated ground motions to be different from the original suite of 22 ground motion pairs. Figure 4.14(a), 4.14(b), and 4.14(c) show the peak ground acceleration (PGA), Arias Intensity (IA), and first spectral moment, respectively, of the generated earthquake motions compared to the original suite of earthquakes used. Looking at Figure 4.14, it can be seen that there are significant differences between the generated ground motions and the original earthquake suite so it can be considered as a new unseen earthquake dataset.



(a) (b) (c)

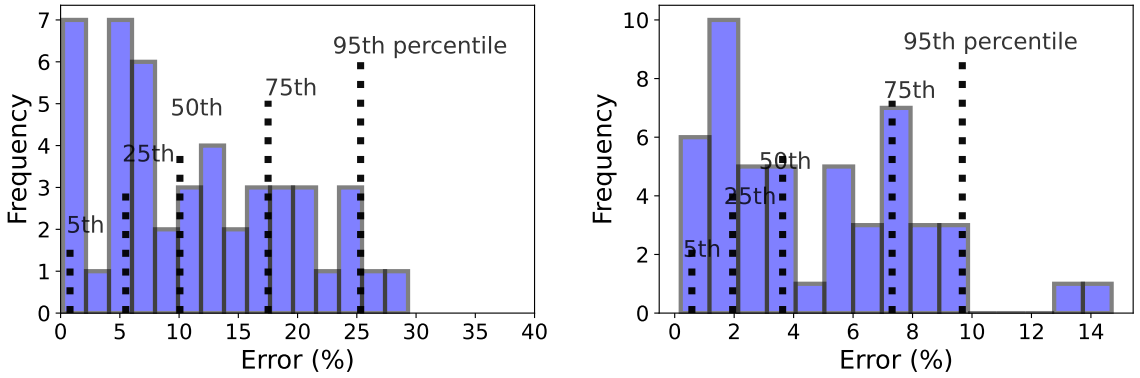
Figure 4.14: Comparison of ground motion intensity measures of (a) PGA, (b) Arias intensity, and (c) 1st spectral moment of the validation dataset with FEMA P695 ground motion suite.

The realization was used in the FE models to obtain the true value of the EDPs (Y). The DNN model predicted the output $\hat{Y}$ using the input Θ. $\hat{Y}$ and Y are then compared to determine the prediction accuracy.



(a) (b)

Figure 4.15: Histogram of prediction error (%) between DNN and FE estimate for (a) peak ISD and (b) peak floor acceleration of the one-story building when using generated earthquake.



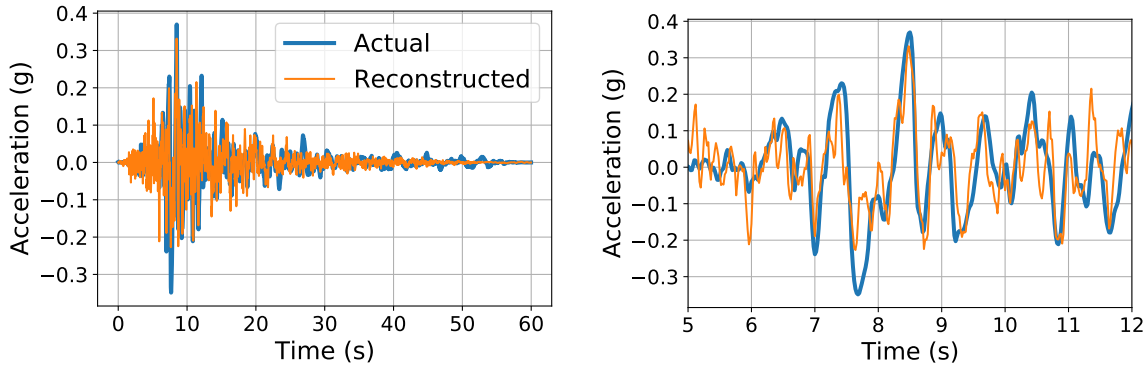
(a) (b)

Figure 4.16: Histogram of prediction error (%) between DNN and FE estimate for (a) peak ISD and (b) peak floor acceleration of the three-story building when using generated earthquake.

The median error was 9% for peak ISD and 3% for peak floor acceleration for the 1-story building. For the 3-story building scenario, a median error of 10% was observed for peak ISD and 3% for peak floor acceleration. Figure 4.15 and Figure 4.16 depict the histograms of the errors in predicting the outputs for the 1-story and 3-story buildings, respectively. For both structural systems, the 95th percentile error is less than 25% for peak ISD and less than 10% for peak floor acceleration.

4.2.6 Prediction for Loma Prieta earthquake

For a more rigorous validation, the DNN models were tasked with predicting the response for the Loma Prieta earthquake recorded at the station Hollister—South and Pine. This specific earthquake is not within the original suite so an accurate prediction would suggest that this ML model is robust. The Loma Prieta earthquake was projected onto the U basis matrix to get the weights according to Equation 2.5. These weights were then multiplied back into the U basis to create the reconstructed time history which can be seen in Figure 4.17 as it is superimposed over the original ground motion.

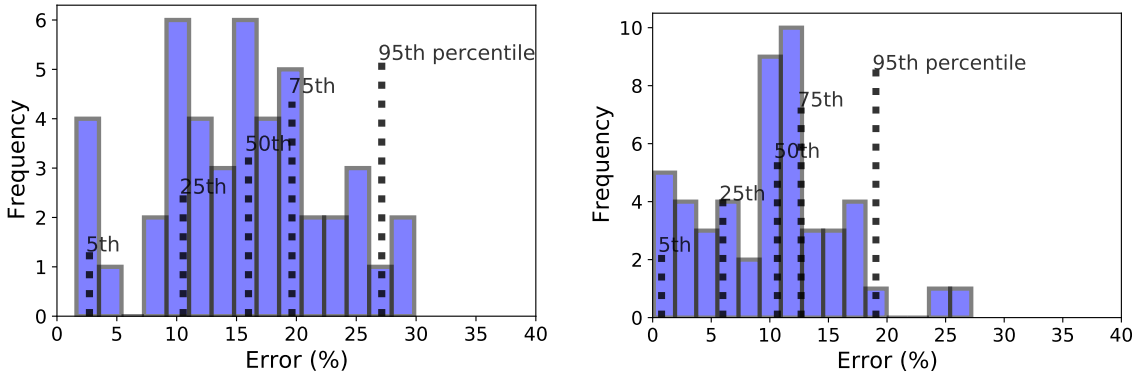


(a) (b)

Figure 4.17: (a) Comparison of Loma-Prieta earthquake and reconstructed earthquake using U basis and weights obtained by projecting the earthquake onto U basis. (b) Zoomed-in strong motion portion of the time history.

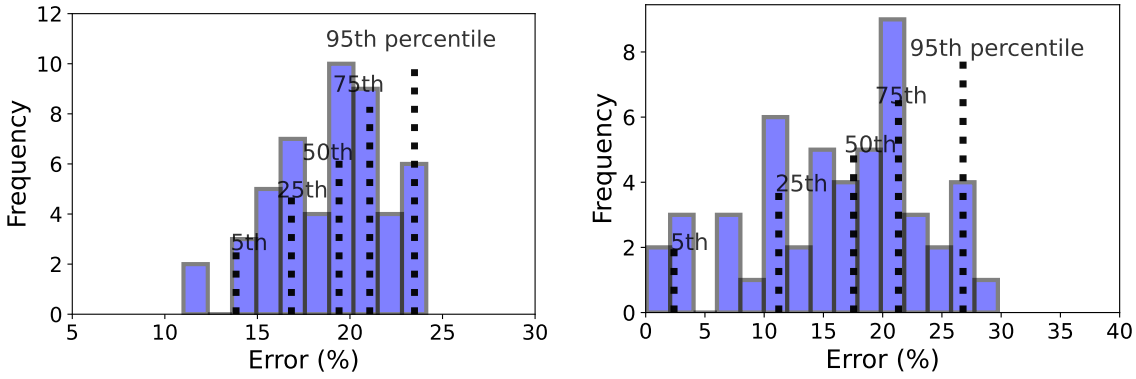
Using SVD to obtain weights of the Loma-Prieta earthquake the actual record can be reproduced with high accuracy and capture the important features of the ground motion. There is an observed level of high frequencies in the reconstructed motion but, this should not be a problem for predicting EDPs as only the weights on the U basis are used as input for the ML model. Comparing the intensity measures for the original earthquake and reconstructed earthquake a difference of 5%, 4%, and 8% error for PGA, Arias intensity, and spectral moment respectively was observed. Using the original Loma-Prieta earthquake and 50 realizations of Θ_2 in the FE models, the output Y was obtained. The inputs Θ were provided to the ML model to produce an output $\hat{Y}$. For the 1-story

building, the surrogate model predicted the peak ISD and the acceleration with a median error of 16% and 10%, respectively, while the 3-story building, had median errors of 19% and 17%, respectively. Figure 4.18 and Figure 4.19 illustrate the histograms of the prediction errors for the 1-story and 3-story buildings, respectively.



(a) (b)

Figure 4.18: Histogram of prediction error (%) between DNN and FE estimate for (a) peak ISD and (b) peak floor acceleration of the one-story building using Loma Prieta earthquake.



(a) (b)

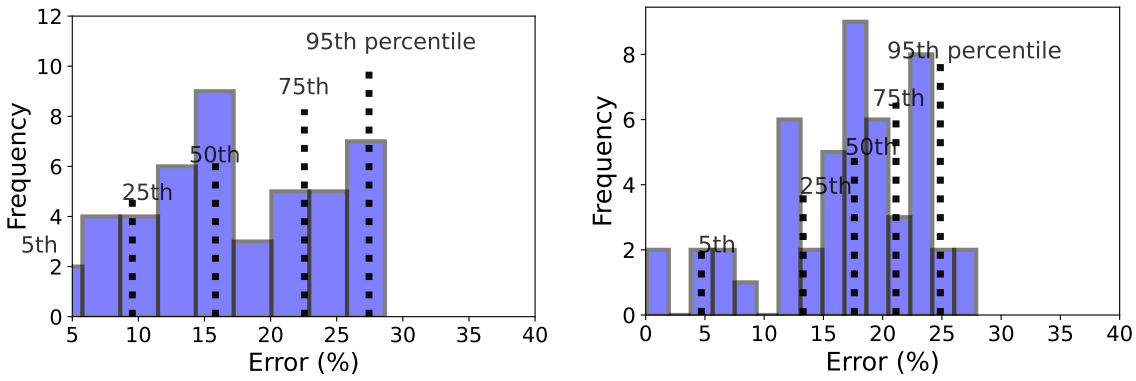
Figure 4.19: Histogram of prediction error (%) between DNN and FE estimate for (a) peak ISD and (b) peak floor acceleration of the three-story building using Loma Prieta earthquake.

The 95th percentile error is less than 30% for both peak ISD and peak floor acceleration. This is considered very successful as this ground motion was not a part of the initial suite of the 22

pairs and suggests that the application of this framework to predict structural response to far-field motions recorded at firm rock sites is attainable.

4.2.7 Ibarra-Medina-Krawlinkler deterioration model

While the robustness of the ML model to unseen earthquake sets was verified, the sensitivity to more complex material models and parameters needs to be considered. A three-story shear frame using the modified Ibarra-Medina-Krawinkler (IMK) deterioration model with bilinear hysteretic response ("Bilin" material in OpenSees) was considered in this section [Lignos and Krawinkler, 2013]. The IMK deterioration model in OpenSEES is an advanced material model that simulates the behavior of deterioration of reinforced concrete and steel elements under seismic ground motions [Lignos and Krawinkler, 2011]. It accounts for deterioration phenomena such as stiffness degradation, strength degradation, and energy dissipation capacity reduction over time due to cyclic loading [Ibarra et al., 2005]. The IMK model considers the deterioration of both concrete and steel components separately and typically requires input parameters such as concrete and steel strengths, ductility capacities, deterioration parameters, and degradation rules. The deterioration of material properties is typically modeled using degradation laws or curves that define how the material properties vary with increasing damage. The FE model and ML model was rerun with the new material considered using the procedure developed in previous sections. The histograms of the prediction errors corresponding to the 3-story building with the modified IMK deterioration model are shown in Figure 4.20 below. The respective median errors are 16% and 17% and the 95th percentile error is less than 27% for both peak ISD and peak floor acceleration. These errors are within the range of the FE models developed with Steel01, showing the robustness of the proposed framework for predicting the nonlinear response using deterioration hysteretic modeling.



(a) (b)

Figure 4.20: Histogram of prediction error (%) between DNN and FE estimate for (a) peak ISD and (b) peak floor acceleration of the three-story building using modified Ibarra-Medina-Krawlinkler deterioration model.

The findings from sections 4.2.5, Unseen Earthquakes and Parameters for Validation, 4.2.6, Prediction for Loma Prieta Earthquake, and 4.2.7, Ibarra-Medina-Krawinkler Deterioration Model, demonstrate that the median errors in peak floor acceleration typically remain lower than those in peak inter-story drift (ISD), with only one exception observed for the 3-story building subjected to Loma-Prieta ground motion. In this case, the difference in median error, approximately 1%, is deemed statistically insignificant. It is important to emphasize that the approach is data-driven, aimed at capturing uncertainties in earthquake and material parameters. Hence, errors are reported in terms of statistical measures such as the median. The errors observed are specific to the analyzed ground motion, and strongly influenced by variations in weights on the U basis used for training. The consistent observation of median errors below 16% throughout this study underscores the robustness of the U basis in characterizing the “unseen" Loma-Prieta earthquake.

5 Conclusion and Future Works

PBEE framework is a decision-making tool used in earthquake engineering. Use of PBEE is often limited by its high computational cost stemming from probabilistic non-linear finite element analysis used in the framework. These FE models take into account uncertainties in both earthquakes and constitutive parameters. This research proposes a surrogate modeling framework based on data augmentation that

i) Converts a “small data" problem to a “big data" problem so that ML models can be trained to have better generation performance.

ii) Chooses from a set of ML models and selects the model that best performs as a surrogate model and

iii) Validates with unseen earthquakes outside the set used for training, to check for the robustness of the model.

Chapter 3 attempts to use the different ML models to capture the non-linear dynamic response of structures in terms of EDPs given traditional ground motion characteristics and material property values. DNNs was determined to have the best performance followed by random forest in predicting the EDPs chosen. It was also observed that the predictions were more sensitive to the forcing function parameters than to the building model parameters. The traditional ground motion characterization techniques like Arias intensity, PGA, PGV, etc. were relevant, but, limited in their use in training ML models for generalization to prediction for unseen earthquakes.

To this end, in Chapter 4, a ML-based surrogate model framework based on SVD-enabled data augmentation is proposed. A representative suite of far-field ground motions recorded on a firm rock site was selected as the dataset. Using SVD, an orthonormal basis was chosen that spans the space of the ground motion suite. The weights of the basis vectors were assumed as random vectors along with the constitutive parameters as random variables to generate a large

set of earthquakes and constitutive parameters. The finite element model is fed these randomly generated earthquakes and constitutive parameters. The randomly generated weights, material parameter values and, the finite element model output were used as training data for the ML model. DNN, SVR, DT, and RF were used as ML models to map the input (weights of the basis vectors and constitutive model parameters) to the output (finite element model response).

R^2 was used as the performance metric to determine the best model. One-story and three-story buildings represented by spring–mass–damper systems were used as the structures to be subjected to far-field ground motions and ultimately predict peak ISD and peak floor acceleration. Among the competing set of ML models, DNN showed it was able to be used to accurately estimate the non-linear response of the buildings subject to unseen earthquakes. A validation stage was also conducted to estimate building response for earthquakes and model parameters that were not a part of the training set. The DNN could predict the response of the FE models with reasonable accuracy (median error less than 20%). This proposed framework supported by the validation results on unseen earthquakes and model parameters provides a firm basis for the validity and applicability of the ML-based surrogate model in predicting non-linear building response.

5.1 Future Works

While the use of the ground motion set, ML models, and SVD-enabled data augmentation methods yielded good results, this framework has limitations that should be expanded on in future research endeavors.

- The use of the far-field motions in this study was for an initial starting point of the proposed framework and was a good representation of the characteristics of far-field motions on firm soil representing soil classes C and D and originating from strike-slip and reverse thrust faults. The use of ground motion suites such as near-field motions or motions with long-period pulses (expected in soft soil) in the training phase could further expand on the

robustness of the existing framework.

- The use of SVD for data augmentation proved to be highly successful but this method is a linear representation. Future studies will utilize non-linear representations such as radial basis functions, Gaussian functions [Parida, 2019, Parida et al., 2018, 2019], Fourier basis, wavelet transform, and auto-encoders [Liou et al., 2014]. Generative Adversarial networks will also be explored to create generated earthquake suites[Marano et al., 2024].

- This framework will extend to more complex, high-fidelity finite element models. Computational efficiency will be achieved by employing multi-fidelity deep neural network surrogate models, where the surrogate for a low-fidelity model will guide the solution direction for the surrogate of a high-fidelity model. This approach will help achieve convergence with a limited number of training data points.

References

Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11), 2018.

Bilal Ahmed, Sujith Mangalathu, and Jong-Su Jeon. Seismic damage state predictions of reinforced concrete structures using stacked long short-term memory neural networks. *Journal of Building Engineering*, 46:103737, 2022. ISSN 2352-7102. doi: https://doi.org/10.1016/j.jobe.2021.103737. URL https://www.sciencedirect.com/science/article/pii/S2352710221015953.

Omar Y. Al-Jarrah, Paul D. Yoo, Sami Muhaidat, George K. Karagiannidis, and Kamal Taha. Efficient machine learning for big data: A review. *Big Data Research*, 2(3):87–93, 2015. ISSN 2214-5796. doi: https://doi.org/10.1016/j.bdr.2015.04.001. URL https://www.sciencedirect.com/science/article/pii/S2214579615000271. Big Data, Analytics, and High-Performance Computing.

Azin Al Kajbaf and Michelle Bensi. Application of surrogate models in estimation of storm surge:a comparative assessment. *Applied Soft Computing*, 91:106184, 2020. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2020.106184. URL https://www.sciencedirect.com/science/article/pii/S1568494620301241.

Jehad Ali, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)*, 9(5):272, 2012.

P. Anbazhagan, Abhishek Kumar, and T.G. Sitharam. Ground motion prediction equation considering combined dataset of recorded and simulated ground motions. *Soil Dynamics and Earthquake Engineering*, 53:92–108, 2013. ISSN 0267-7261. doi: https://doi.org/10.1016/j.soildyn.2013.0

6.003. URL https://www.sciencedirect.com/science/article/pii/S0267726113001
36X.

Panagiotis G Asteris and Vaseilios G Mokos. Concrete compressive strength using artificial neural networks. *Neural Computing and Applications*, 32(15):11807–11826, 2020.

Panagiotis G. Asteris, Fariz Iskandar Mohd Rizal, Mohammadreza Koopialipoor, Panayiotis C. Roussis, Maria Ferentinou, Danial Jahed Armaghani, and Behrouz Gordan. Slope stability classification under seismic conditions using several tree-based intelligent techniques. *Applied Sciences*, 12(3), 2022. ISSN 2076-3417. doi: 10.3390/app12031753. URL https://www.mdpi.com/2076-3417/12/3/1753.

Navid Ataei and Jamie E. Padgett. Fragility surrogate models for coastal bridges in hurricane prone zones. *Engineering Structures*, 103:203–213, 2015. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2015.07.002. URL https://www.sciencedirect.com/science/article/pii/S0141029615004356.

ML ATC. Fema p695 recommended methodology for quantification of building system performance and response parameters. 2009.

Gail M. Atkinson and David M. Boore. Earthquake Ground-Motion Prediction Equations for Eastern North America. *Bulletin of the Seismological Society of America*, 96(6):2181–2205, 12 2006. ISSN 0037-1106. doi: 10.1785/0120050245. URL https://doi.org/10.1785/0120050245.

Onur Avci, Osama Abdeljaber, and Serkan Kiranyaz. An overview of deep learning methods used in vibration-based damage detection in civil engineering. In Kirk Grimmelsman, editor, *Dynamics of Civil Structures, Volume 2*, pages 93–98, Cham, 2022. Springer International Publishing. ISBN 978-3-030-77143-0.

S Bose, A Stavridis, PC Anastasopoulos, and K Sett. Surrogate statistical model of a school building in nepal using asce 41-17. In *2nd International Conference on Natural Hazards and Infrastructure, Chania, Greece*, 2019.

Dustin Boswell. Introduction to support vector machines. *Departement of Computer Science and Engineering University of California San Diego*, 11, 2002.

Mario Camana, Saeed Ahmed, Carla García, and Insoo Koo. Extremely randomized trees-based scheme for stealthy cyber-attack detection in smart grid networks. *IEEE Access*, PP:1–1, 01 2020. doi: 10.1109/ACCESS.2020.2968934.

Brian Chiou, Robert Darragh, Nick Gregor, and Walter Silva. Nga project strong-motion database. *Earthquake Spectra*, 24(1):23–44, 2008.

C. A. Cornell and H. Krawinkler. Progress and challenges in seismic performance assessment, 2000. URL `http://peer.berkeley.edu/news/2000spring/index.html`. PEER Center News.

Noel Cressie. *Statistics for spatial data*. John Wiley & Sons, 2015.

Armin Dadras Eslamlou and Shiping Huang. Artificial-neural-network-based surrogate models for structural health monitoring of civil structures: A literature review. *Buildings*, 12(12), 2022. ISSN 2075-5309. doi: 10.3390/buildings12122067. URL `https://www.mdpi.com/2075-5309/12/12/2067`.

AD Dongare, RR Kharde, and Amit D Kachare. Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1):189–194, 2012.

Marwa M. H. El-Sayed, Siddharth S. Parida, Prashant Shekhar, Amy Sullivan, and Christopher J. Hennigan. Predicting atmospheric water-soluble organic mass reversibly partitioned to aerosol

liquid water in the eastern united states. *Environmental Science & Technology*, 57(46):18151–18161, 2023. doi: 10.1021/acs.est.3c01259. URL https://doi.org/10.1021/acs.est.3c01259. PMID: 37952161.

Michele Fratello and Roberto Tagliaferri. Decision trees and random forests. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, 1(S 3), 2018.

Marco Gaetani d'Aragona, Maria Polese, Marco Di Ludovico, and Andrea Prota. The use of stick-it model for the prediction of direct economic losses. *Earthquake Engineering & Structural Dynamics*, 50(7):1884–1907, 2021.

Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022.

Raoof Gholami and Nikoo Fakhari. Chapter 27 - support vector machine: Principles, parameters, and applications. In Pijush Samui, Sanjiban Sekhar, and Valentina E. Balas, editors, *Handbook of Neural Computation*, pages 515–535. Academic Press, 2017. ISBN 978-0-12-811318-9. doi: https://doi.org/10.1016/B978-0-12-811318-9.00027-2. URL https://www.sciencedirect.com/science/article/pii/B9780128113189000272.

Ioannis Gidaris, Alexandros A. Taflanidis, and George P. Mavroeidis. Kriging metamodeling in seismic risk assessment based on stochastic ground motion models. *Earthquake Engineering & Structural Dynamics*, 44(14):2377–2399, 2015. doi: https://doi.org/10.1002/eqe.2586. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/eqe.2586.

Jose Maria Giron-Sierra. *Time-Frequency Analysis*, pages 357–494. Springer Singapore, Singapore, 2017. ISBN 978-981-10-2534-1. doi: 10.1007/978-981-10-2534-1_7. URL https://doi.org/10.1007/978-981-10-2534-1_7.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Yuxuan Gu, Qixin Chen, Kai Liu, Le Xie, and Chongqing Kang. Gan-based model for residential load generation considering typical consumption patterns. 11 2018. doi: 10.1109/ISGT.2019.8791575.

Xingquan Guan, Henry Burton, Mehrdad Shokrabadi, and Zhengxiang Yi. Seismic drift demand estimation for steel moment frame buildings: From mechanics-based to data-driven models. *Journal of Structural Engineering*, 147(6):04021058, 2021. doi: 10.1061/(ASCE)ST.1943-541X.0003004. URL https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29ST.1943-541X.0003004.

Mohammad Amin Hariri-Ardebili and Golsa Mahdavi. Generalized uncertainty in surrogate models for concrete strength prediction. *Engineering Applications of Artificial Intelligence*, 122: 106155, 2023. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2023.106155. URL https://www.sciencedirect.com/science/article/pii/S0952197623003391.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Yongjun Hong, Uiwon Hwang, Jaeyoon Yoo, and Sungroh Yoon. How generative adversarial networks and their variants work: An overview. *ACM Comput. Surv.*, 52(1), feb 2019. ISSN 0360-0300. doi: 10.1145/3301282. URL https://doi.org/10.1145/3301282.

Rongrong Hou and Yong Xia. Review on the new development of vibration-based damage identification for civil engineering structures: 2010–2019. *Journal of Sound and Vibration*, 491: 115741, 2021. ISSN 0022-460X. doi: https://doi.org/10.1016/j.jsv.2020.115741. URL https://www.sciencedirect.com/science/article/pii/S0022460X2030571X.

Dang Viet Hung and Nguyen Truong Thang. Predicting dynamic responses of frame structures subjected to stochastic wind loads using temporal surrogate model. *Journal of Science and*

Technology in Civil Engineering (JSTCE) - HUCE, 16(2):106–116, Apr. 2022. doi: 10.31814/stce.huce(nuce)2022-16(2)-09. URL https://stce.huce.edu.vn/index.php/en/article/view/2269.

Seong-Hoon Hwang, Sujith Mangalathu, Jiuk Shin, and Jong-Su Jeon. Machine learning-based approaches for seismic demand and collapse of ductile reinforced concrete building frames. *Journal of Building Engineering*, 34:101905, 2021. ISSN 2352-7102. doi: https://doi.org/10.1016/j.jobe.2020.101905. URL https://www.sciencedirect.com/science/article/pii/S2352710220335385.

Luis F Ibarra, Ricardo A Medina, and Helmut Krawinkler. Hysteretic models that incorporate strength and stiffness deterioration. *Earthquake engineering & structural dynamics*, 34(12): 1489–1511, 2005.

Vikramaditya Jakkula. Tutorial on support vector machine (svm). *School of EECS, Washington State University*, 37(2.5):3, 2006.

F. Kazemi, N. Asgarkhani, and R. Jankowski. Machine learning-based seismic fragility and seismic vulnerability assessment of reinforced concrete structures. *Soil Dynamics and Earthquake Engineering*, 166:107761, 2023. ISSN 0267-7261. doi: https://doi.org/10.1016/j.soildyn.2023.107761. URL https://www.sciencedirect.com/science/article/pii/S0267726123000064.

Muhammad Yaseen Khan, Abdul Qayoom, Muhammad Nizami, Muhammad Shoaib Siddiqui, Shaukat Wasi, and Khaliq-Ur-Rahman Raazi Syed. Automated prediction of good dictionary examples (gdex): A comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques. *Complexity*, 09 2021. doi: 10.1155/2021/2553199.

Farid Khosravikia and Patricia Clayton. Machine learning in ground motion prediction. *Computers & Geosciences*, 148:104700, 2021. ISSN 0098-3004. doi: https://doi.org/10.1016/j.cageo.2021

.104700. URL https://www.sciencedirect.com/science/article/pii/S0098300421000157.

Jalal Kiani, Charles Camp, and Shahram Pezeshk. On the application of machine learning techniques to derive seismic fragility curves. *Computers & Structures*, 218:108–122, 2019. ISSN 0045-7949. doi: https://doi.org/10.1016/j.compstruc.2019.03.004. URL https://www.sciencedirect.com/science/article/pii/S0045794918318650.

Keniti Kido. *Digital Fourier analysis: fundamentals*. Springer, 2014.

Korhan Kocamaz, Barış Binici, and Kağan Tuncay. Prediction of nonlinear drift demands for buildings with recurrent neural networks. 2021.

Q. Kong, T. Siauw, and A. Bayen. *Python Programming and Numerical Methods: A Guide for Engineers and Scientists*. Elsevier Science, 2020. ISBN 9780128195499. URL https://books.google.com/books?id=cZ4LEAAAQBAJ.

Slawomir Koziel and Anna Pietrenko-Dabrowska. *Surrogate Modeling for High-Frequency Design*. WORLD SCIENTIFIC (EUROPE), 2022. doi: 10.1142/q0317. URL https://www.worldscientific.com/doi/abs/10.1142/q0317.

Steven Lawrence Kramer. *Geotechnical earthquake engineering*. Pearson Education India, 1996.

Vrushali Y Kulkarni and Pradeep K Sinha. Random forest classifiers: a survey and future research directions. *Int. J. Adv. Comput*, 36(1):1144–1153, 2013.

Aikaterini P. Kyprioti and Alexandros A. Taflanidis. Kriging metamodeling for seismic response distribution estimation. *Earthquake Engineering & Structural Dynamics*, 50(13):3550–3576, 2021. doi: https://doi.org/10.1002/eqe.3522. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/eqe.3522.

Sascha Lange and Martin Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2010. doi: 10.1109/IJCNN.2010.5596468.

Dimitrios G Lignos and Helmut Krawinkler. Deterioration modeling of steel components in support of collapse prediction of steel moment frames under earthquake loading. *Journal of Structural Engineering*, 137(11):1291–1302, 2011.

Dimitrios G Lignos and Helmut Krawinkler. Development and utilization of structural component databases for performance-based earthquake engineering. *Journal of Structural Engineering*, 139(8):1382–1394, 2013.

Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. Autoencoder for words. *Neurocomputing*, 139:84–96, 2014.

Y Liu, R Chen, Y Jiang, and W Liu. Lumped-mass stick modeling of building structures with mixed wall-column components. *Proceeding of the 15 WCEE LISBOA*, 2012.

Gilles Louppe. Understanding random forests. *Cornell University Library*, 10, 2014.

Shasha Lu, Mohammadreza Koopialipoor, Panagiotis G Asteris, Maziyar Bahri, and Danial Jahed Armaghani. A novel feature selection approach based on tree models for evaluating the punching shear capacity of steel fiber-reinforced concrete flat slabs. *Materials*, 13(17):3902, 2020.

Hai-Bang Ly, Binh Thai Pham, Lu Minh Le, Tien-Thinh Le, Vuong Minh Le, and Panagiotis G Asteris. Estimation of axial load-carrying capacity of concrete-filled steel tubes using surrogate models. *Neural Computing and Applications*, 33:3437–3458, 2021.

Sonali B Maind and Priyanka Wankar. Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(1): 96–100, 2014.

Sujith Mangalathu and Jong-Su Jeon. Stripe-based fragility analysis of multispan concrete bridge classes using machine learning techniques. *Earthquake Engineering & Structural Dynamics*, 48 (11):1238–1255, 2019a. doi: https://doi.org/10.1002/eqe.3183. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/eqe.3183.

Sujith Mangalathu and Jong-Su Jeon. Stripe-based fragility analysis of multispan concrete bridge classes using machine learning techniques. *Earthquake Engineering & Structural Dynamics*, 48 (11):1238–1255, 2019b.

Sujith Mangalathu, Gwanghee Heo, and Jong-Su Jeon. Artificial neural network based multi-dimensional fragility development of skewed concrete bridge classes. *Engineering Structures*, 162:166–176, 2018. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2018.01.053. URL https://www.sciencedirect.com/science/article/pii/S0141029617326275.

Giuseppe Carlo Marano, Marco Martino Rosso, Angelo Aloisio, and Giansalvo Cirrincione. Generative adversarial networks review in earthquake-related engineering fields. *Bulletin of Earthquake Engineering*, 22(7):3511–3562, 2024.

Frank McKenna, GL Fenves, MH Scott, and B Jeremic. Open system for earthquake engineering simulation (opensees). *Pacific Earthquake Engineering Research Center, Univ. of California, Berkeley, CA*, 2000.

Frank McKenna, Michael H. Scott, and Gregory L. Fenves. Nonlinear finite-element analysis software architecture using object composition. *Journal of Computing in Civil Engineering*, 24 (1):95–107, 2010. doi: 10.1061/(ASCE)CP.1943-5487.0000002. URL https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29CP.1943-5487.0000002.

Marco Menegotto. Method of analysis for cyclically loaded rc plane frames including changes in geometry and non-elastic behavior of elements under combined normal force and bending. In

Proc. of IABSE Symposium on Resistance and Ultimate Deformability of Structures Acted on by Well Defined Repeated Loads, 1973, 1973.

Chara Ch Mitropoulou and Manolis Papadrakakis. Developing fragility curves based on neural network ida predictions. *Engineering Structures*, 33(12):3409–3421, 2011.

Jack Moehle and Gregory G Deierlein. A framework methodology for performance-based earthquake engineering. In *13th world conference on earthquake engineering*, volume 679, page 12. WCEE Vancouver, 2004.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Oscar Möller, Ricardo O. Foschi, Marcelo Rubinstein, and Laura Quiroz. Seismic structural reliability using different nonlinear dynamic response surface approximations. *Structural Safety*, 31 (5):432–442, 2009. ISSN 0167-4730. doi: https://doi.org/10.1016/j.strusafe.2008.12.001. URL https://www.sciencedirect.com/science/article/pii/S0167473008001094.

Andrew Y Ng. Preventing" overfitting" of cross-validation data. In *ICML*, volume 97, pages 245–253. Citeseer, 1997.

L. Nguyen-Minh, M. Rovňák, T. Tran-Quoc, and K. Nguyenkim. Punching shear resistance of steel fiber reinforced concrete flat slabs. *Procedia Engineering*, 14:1830–1837, 2011. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2011.07.230. URL https://www.sciencedirect.com/science/article/pii/S1877705811013075. The Proceedings of the Twelfth East Asia-Pacific Conference on Structural Engineering and Construction.

Mehdi Nikoo Liborio Cavaleri Panagiotis G. Asteris, Saeed Nozhati and Mohammad Nikoo. Krill herd algorithm-based neural network in structural seismic reliability evaluation. *Mechanics of*

Advanced Materials and Structures, 26(13):1146–1153, 2019. doi: 10.1080/15376494.2018.1430874. URL https://doi.org/10.1080/15376494.2018.1430874.

S. S. Parida. *Model-data fusion for probabilistic analysis of civil infrastructures*. PhD thesis, University at Buffalo, The State University of New York, Buffalo, NY, 2019.

Siddharth S Parida, Kallol Sett, and Puneet Singla. An efficient pde-constrained stochastic inverse algorithm for probabilistic geotechnical site characterization using geophysical measurements. *Soil Dynamics and Earthquake Engineering*, 109:132–149, 2018.

Siddharth S Parida, Kallol Sett, and Puneet Singla. Model-data fusion for spatial and statistical characterization of soil parameters from geophysical measurements. *Soil Dynamics and Earthquake Engineering*, 124:35–57, 2019.

Siddharth S. Parida, Alexandros Nikellis, Kallol Sett, and Puneet Singla. Model-data fusion for seismic performance evaluation of an instrumented highway bridge. *Earthquake Engineering & Structural Dynamics*, 49(14):1559–1578, 2020. doi: https://doi.org/10.1002/eqe.3317. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/eqe.3317.

Siddharth S. Parida, Supratik Bose, Megan Butcher, Georgios Apostolakis, and Prashant Shekhar. Svd enabled data augmentation for machine learning based surrogate modeling of non-linear structures. *Engineering Structures*, 280:115600, 2023. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2023.115600. URL https://www.sciencedirect.com/science/article/pii/S0141029623000147.

SS Parida, M Butcher, S Bose, G Apostolakis, and P Shekhar. Machine learning based surrogate model to predict engineering demand parameters. In *12th national conference on earthquake engineering. Salt Lake City, Utah, USA*, 2022.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion,

Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, and Vincent Dubourg. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Carlos A. Perez-Ramirez, Juan P. Amezquita-Sanchez, Martin Valtierra-Rodriguez, Hojjat Adeli, Aurelio Dominguez-Gonzalez, and Rene J. Romero-Troncoso. Recurrent neural network model with bayesian training and mutual information for response prediction of large buildings. *Engineering Structures*, 178:603–615, 2019. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2018.10.065. URL https://www.sciencedirect.com/science/article/pii/S0141029618307235.

Stephen Pessiki. Sustainable seismic design. *Procedia Engineering*, 171:33–39, 2017. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2017.01.307. URL https://www.sciencedirect.com/science/article/pii/S1877705817303077. The 3rd International Conference on Sustainable Civil Engineering Structures and Construction Materials - Sustainable Structures for Future Generations.

Keith A. Porter, James L. Beck, Jianye Y. Ching, Judith Mitrani-Reiser, Masaki Miyamura, Atsushi Kusaka, and Yoshiyuki Hyodo. Real-time loss estimation for instrumented buildings. Technical Report 8, Report EERL 2004, 2004.

Abdur Rasheed, Muhammad Usman, Muhammad Zain, and Nadeem Iqbal. Machine learning-based fragility assessment of reinforced concrete buildings. *Computational Intelligence and Neuroscience*, 2022, 2022.

Hwasung Roh, Huseok Lee, and Jong Seh Lee. New lumped-mass-stick model based on modal characteristics of structures: development and application to a nuclear containment building. *Earthquake Engineering and Engineering Vibration*, 12(2):307–317, 2013.

Alireza Sarraf Shirazi and Ian Frigaard. Slurrynet: Predicting critical velocities and frictional pressure drops in oilfield suspension flows. *Energies*, 14:1263, 02 2021. doi: 10.3390/en14051263.

RL Segura, JE Padgett, and P Paultre. Fragility surfaces for efficient seismic assessment of gravity dams via surrogate modeling. In *The 17th World Conference on Earthquake Engineering, Sendai, Japan*, 2020.

Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019. doi: 10.1186/s40537-019-0197-0. URL https://doi.org/10.1186/s40537-019-0197-0.

Julius O Smith. *Mathematics of the discrete Fourier transform (DFT): with audio applications*. Julius Smith, 2007.

Reda Snaiki and Siddharth S. Parida. A data-driven physics-informed stochastic framework for hurricane-induced risk estimation of transmission tower-line systems under a changing climate. *Engineering Structures*, 280:115673, 2023a. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2023.115673. URL https://www.sciencedirect.com/science/article/pii/S0141029623000871.

Reda Snaiki and Siddharth S. Parida. Climate change effects on loss assessment and mitigation of residential buildings due to hurricane wind. *Journal of Building Engineering*, 69:106256, 2023b. ISSN 2352-7102. doi: https://doi.org/10.1016/j.jobe.2023.106256. URL https://www.sciencedirect.com/science/article/pii/S2352710223004357.

Youngrok Song, Sangwon Hyun, and Yun-Gyung Cheong. Analysis of autoencoders for network intrusion detection. *Sensors*, 21:4294, 06 2021. doi: 10.3390/s21134294.

Jaime Lynn Speiser, Michael E. Miller, Janet Tooze, and Edward Ip. A comparison of random

forest variable selection methods for classification prediction modeling. *Expert Systems with Applications*, 134:93–101, 2019. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2019.05.028. URL https://www.sciencedirect.com/science/article/pii/S0957417419303574.

Rih-Teng Wu and Mohammad R. Jahanshahi. Deep convolutional neural network for structural dynamic response estimation and system identification. *Journal of Engineering Mechanics*, 145 (1):04018125, 2019. doi: 10.1061/(ASCE)EM.1943-7889.0001556. URL https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29EM.1943-7889.0001556.

Yazhou Xie, Majid Ebad Sichani, Jamie E Padgett, and Reginald DesRoches. The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. *Earthquake Spectra*, 36(4):1769–1801, 2020a. doi: 10.1177/8755293020919419. URL https://doi.org/10.1177/8755293020919419.

Yazhou Xie, Majid Ebad Sichani, Jamie E Padgett, and Reginald DesRoches. The promise of implementing machine learning in earthquake engineering: A state-of-the-art review. *Earthquake Spectra*, 36(4):1769–1801, 2020b. doi: 10.1177/8755293020919419. URL https://doi.org/10.1177/8755293020919419.

Mohsen Zaker Esteghamati and Madeleine M. Flint. Developing data-driven surrogate models for holistic performance-based assessment of mid-rise rc frame buildings at early design. *Engineering Structures*, 245:112971, 2021. ISSN 0141-0296. doi: https://doi.org/10.1016/j.engstruct.2021.112971. URL https://www.sciencedirect.com/science/article/pii/S0141029621011147.

Irmela Zentner, Max Gündel, and Nicolas Bonfils. Fragility analysis methods: Review of existing approaches and application. *Nuclear Engineering and Design*, 323:245–258, 2017. ISSN 0029-

5493. doi: https://doi.org/10.1016/j.nucengdes.2016.12.021. URL https://www.sciencedirect.com/science/article/pii/S0029549316305209.

Junhai Zhai, Sufang Zhang, Junfen Chen, and Qiang He. Autoencoder and its various variants. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 415–419, 2018. doi: 10.1109/SMC.2018.00080.

Ruiyang Zhang, Zhao Chen, Su Chen, Jingwei Zheng, Oral Büyüköztürk, and Hao Sun. Deep long short-term memory networks for nonlinear structural seismic response prediction. *Computers & Structures*, 220:55–68, 2019. ISSN 0045-7949. doi: https://doi.org/10.1016/j.compstruc.2019.05.006. URL https://www.sciencedirect.com/science/article/pii/S0045794919302263.

Jure Zupan. Introduction to artificial neural network (ann) methods: what they are and how to use them. *Acta Chimica Slovenica*, 41(3):327, 1994.