Data-Driven Prediction of Seismic Drift Responses Using DNNs and XAI

A Thesis

by

Mayaz Uddin Gazi

Md. Lahin Kabir

Md. Rayhan Chowdhury

BACHELOR OF SCIENCE IN CIVIL ENGINEERING

DEPARTMENT OF CIVIL ENGINEERING

SYLHET ENGINEERING COLLEGE, SYLHET-3100 BANGLADESH

July, 2025

Data-Driven Prediction of Seismic Drift Responses Using DNNs and XAI

By

Mayaz Uddin Gazi

Md. Lahin Kabir

Md. Rayhan Chowdhury

A thesis submitted to Department Civil Engineering in partial fulfillment for the degree

of

BACHELOR OF SCIENCE IN CIVIL ENGINEERING

At the

Sylhet Engineering College

Affiliated with Sylhet University of Science and Technology (SUST), Sylhet

DEPARTMENT OF CIVIL ENGINEERING

SYLHET ENGINEERING COLLEGE, SYLHET-3100 BANGLADESH

July, 2025

DECLARATION

I hereby declare that this thesis/project is th e result of my own independent work, and to the best of my knowledge, it contains no material previously published or written by another person, nor has it been submitted for the award of any degree at Sylhet Engineering College (SEC) or any other academic institution, except where proper acknowledgment has been made. This thesis was carried out by us under the supervision of Md. Titumir Hasan.

I further affirm that the intellectual content of this work is entirely my own. Any contributions to the research by collaborators at this college or elsewhere have been clearly acknowledged.

....

Mayaz Uddin Gazi

Reg No: 2019333527

....

Md. Lahin Kabir

Reg No: 2019333537

....

Md. Rayhan Chowdhury

Reg No: 2019333509

CERTIFICATION

Data-Driven Prediction of Seismic Drift Responses Using DNNs and XAI

By

Mayaz Uddin Gazi

Md. Lahin Kabir

Md. Rayhan Chowdhury

A thesis submitted to the Department of Civil Engineering in partial fulfillment for the degree of

BACHELOR OF SCIENCE IN CIVIL ENGINEERING

At the

Sylhet Engineering College

APPROVED AS TO STYLE AND CONTENT BY:

Sibbir Ahmed	Md. Titumir Hasan
Assistant Professor & Head of the Department	Workshop Maintenance Engineer
Sylhet Engineering College, Sylhet	Sylhet Engineering College, Sylhet

DEPARTMENT OF CIVIL ENGINEERING

SYLHET ENGINEERING COLLEGE, SYLHET-3100 BANGLADESH

July, 2025

Signature of Author 1: ______________________

- Mayaz Uddin Gazi
- 2019333527
- Department of Civil Engineering

Signature of Author 2: ______________________

- Md. Lahin Kabir
- 2019333537
- Department of Civil Engineering

Signature of Author 3: ______________________

- Md. Rayhan Chowdhury
- 2019333509
- Department of Civil Engineering

Certified and accepted by: ______________________

Md. Titumir Hasan

Thesis Supervisor

Certified and accepted by: ______________________

Sibbir Ahmed

Thesis Co-Supervisor

Acknowledgment

The author gratefully acknowledges the invaluable supervision, guidance, and encouragement provided by Md. Titumir Hasan, whose expertise in structural engineering and commitment to research excellence were instrumental throughout every phase of this thesis. The constructive feedback and continuous support from the academic staff of the Department of Civil Engineering at Sylhet Engineering College are sincerely appreciated.

Special thanks are extended to the developers and the user community of OpenSees for providing the simulation tools fundamental to this study, and to the PEER NGA database for access to ground motion records.

The author is indebted to friends and colleagues for their encouragement, motivation, and critical discussions, which enriched both the research and the writing process. Lastly, heartfelt gratitude is offered to the author's family for their patience, understanding, and unwavering support throughout the entire journey.

This work did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Contents

List of Figures

List of Tables

Abbreviations and Symbols

Table 1: Key abbreviations, symbols, and units used in this study.

Full Name	Symbol / Abbreviation	Unit / Notes
Natural Period	T	s
Strength Ratio	SR	–
Peak Ground Acceleration	PGA	g
Peak Ground Velocity	PGV	cm/s
Peak Ground Displacement	PGD	cm
Moment Magnitude	M_w	–
Rupture Distance	R_{rup}	km
Arias Intensity	AI	m/s
Shear Wave Velocity (30m)	V_{S30}	m/s
Maximum Inter-story Drift Ratio	MaxDrift	–
Spectral Acceleration	Sa	g
Nonlinear Time-History Analysis	NLTHA	Method
Machine Learning	ML	Methodology
Explainable Artificial Intelligence	XAI	Methodology
Incremental Dynamic Analysis	IDA	Method
Random Forest	RF	Model
Artificial Neural Network	ANN	Model
Deep Neural Network	DNN	Model
Gradient-Boosted Decision Trees	GBDT	Model
Extreme Gradient Boosting	XGBoost	Model
Coefficient of Determination	R^2	–
Mean Absolute Error	MAE	–
Mean Squared Error	MSE	–
Root Mean Squared Error	RMSE	–
Open System for Earthquake Engineering Simulation	OpenSees	Software
SHapley Additive exPlanations	SHAP	XAI tool
Local Interpretable Model-agnostic Explanations	LIME	XAI tool

Abstract

Structural drift is a fundamental indicator of building performance under seismic loading, as excessive drift can lead to severe damage or collapse, particularly in reinforced concrete (RC) frames. Accurate drift prediction is therefore essential for effective seismic risk assessment and resilient building design, especially in tectonically active and resource-constrained regions. This study proposes a transparent, data-driven framework for predicting seismic drift responses by integrating nonlinear time-history analyses (NLTHA) in OpenSees with advanced machine learning (ML) and explainable artificial intelligence (XAI) methods. A comprehensive synthetic dataset, reflecting diverse building and ground motion scenarios relevant to Bangladesh, was constructed and systematically processed for supervised regression. Two models, a Random Forest (RF) ensemble and a deep artificial neural network (ANN), were benchmarked using cross-validation and interpretability tools such as SHAP and LIME. The RF model achieved superior predictive performance (R2 = 0.897, MAE= 0.491), as well as greater robustness in small-sample settings. XAI analyses provided clear insights into feature importance and model reliability. Limitations due to data scope and model generalizability are acknowledged, and future directions include real-world data integration and hybrid physics-informed approaches. Overall, the proposed workflow offers a scalable, interpretable, and efficient solution for seismic risk assessment and resilience planning in developing urban contexts.

Chapter 1

Introduction

This chapter helps to form the basis of the research by pointing to the global and national significance of the seismic risk assessment, especially with reference to the city of Bangladesh and to the most susceptible places. It speaks on how frequent and devastating earthquakes are, how unpredictable they are, and addresses existing dilemmas in the sphere of preparing against the disaster and mitigating the risks it poses, and how major cities of Bangladesh, and in particular Sylhet are vulnerable to it. The chapter also presents the current constraints of assessment methods and preconditions the introduction of the data-driven approach to machine learning that can be understood and correctly interpreted. Lastly, the aim and the coverage of the current study are well-explained, offering a plan of the study to be carried out.

Seismic assessment is of critical global importance due to the profound threat that earthquakes pose to human life, infrastructure, and economies worldwide. Earthquakes rank among the deadliest natural hazards, frequently resulting in catastrophic loss of life, mass displacement, and prolonged economic disruption. According to the United Nations Office for Disaster Risk Reduction (UNDRR), between 1998 and 2017, earthquakes and related tsunamis accounted for 56% of all disaster-related fatalities globally, resulting in approximately 747,234 deaths during this period [1]. The economic impact is similarly staggering, with direct disaster-related losses—including those from earthquakes—amounting to hundreds of billions of dollars annually [1] [2].

In recent years, both the frequency and intensity of seismic events have shown a marked increase. Geological observatory data indicate that from 2015 to 2025, the number of earthquakes with magnitudes of 6.0 and above has increased by over 30% compared to the previous decade [3, 4]. This trend is particularly pronounced in tectonically active zones such as the Pacific Ring of Fire, where densely populated urban centers remain at heightened risk of devastating seismic events [3, 4].

Despite notable advancements in seismological research, the ability to predict the precise timing, location, and magnitude of earthquakes remains beyond current scientific capabilities.

The U.S. Geological Survey (USGS) and leading researchers consistently emphasize that, while long-term probabilistic models exist, the chaotic behavior of fault systems and the absence of reliable precursory signals make short-term earthquake prediction scientifically unfeasible [5]. Consequently, seismic events frequently occur with little or no warning, underscoring the urgent need for comprehensive preparedness and risk mitigation strategies.

Preparedness efforts, including the rapid seismic assessment of building inventories, the adoption of resilient structural design practices, and widespread public education, are essential to minimize both casualties and economic damage [6]. Retrofitting vulnerable infrastructure and enforcing seismic building codes can significantly reduce the risk of structural collapse, which remains the leading cause of fatalities during seismic events [7].

Bangladesh faces a high risk of devastating earthquakes due to its unique geographic position at the intersection of the Indian, Eurasian, and Burma tectonic plates—an area known as the Himalayan collision zone [8, 9]. This tectonic setting results in several active fault lines, including the Dauki, Jamuna, and Chittagong-Myanmar faults, which have the potential to generate powerful earthquakes affecting large swathes of the country [10].

The threat is particularly acute for major urban centers such as Dhaka, Chattogram, and Sylhet, which have been identified as some of the highest-risk zones in the nation [11] [1]. Dhaka, for instance, is ranked among the 20 most vulnerable cities globally to earthquakes, while Sylhet and Chattogram are situated near active faults and have experienced frequent seismic activity in recent years [11].

Rapid urbanization, poor construction quality, and weak enforcement of building codes significantly amplify Bangladesh's seismic vulnerability. The country's cities have expanded rapidly, often with little regard for proper development controls or adherence to the Bangladesh National Building Code (BNBC) [11, 7, 1, 12] Surveys reveal that a large proportion of buildings in urban centers are constructed without proper approval or structural design, making them highly susceptible to collapse during an earthquake [7]. The 2013 Rana Plaza disaster, which killed over 1,100 people, starkly illustrated the deadly consequences of poor construction practices and inadequate code enforcement in a seismically active region [13].

Despite the recognized risks, preparedness and mitigation efforts remain insufficient. Many buildings remain unreinforced, and disaster management systems are often under-resourced or lack coordination [1, 7]. The government has initiated risk assessments and begun to implement measures such as stricter building inspections, public awareness campaigns, and investment in emergency response equipment, but major gaps persist—especially in enforcing building codes and retrofitting vulnerable structures [1, 14].

In summary, seismic assessment is crucial for Bangladesh to identify vulnerabilities, prioritize retrofitting and reconstruction, enforce building codes, and develop effective disaster preparedness strategies. Without urgent action, the combination of tectonic risk, rapid urban growth, and weak regulatory oversight leaves millions at risk in Dhaka, Chattogram, Sylhet, and beyond [15].

Sylhet is classified as Seismic Zone III, the most seismically active zone in Bangladesh, according to the Bangladesh National Building Code (BNBC-2020), with the highest seismic zone coefficient ($Z = 0.36$) [14, 13]. This geographic vulnerability is further compounded by the city's proximity to major tectonic features, including the Dauki Fault, Sylhet Fault, and Tripura Fault, making it highly susceptible to both moderate and potentially catastrophic earthquakes [13]. The region frequently experiences small to moderate tremors, which have not only raised public concern but also exposed the fragility of existing infrastructure [4, 13].

A critical concern is the prevalence of non-engineered and structurally vulnerable buildings, many of which were constructed without seismic design considerations or compliance with modern building codes [13]. Field surveys have repeatedly identified numerous at-risk structures, prompting city authorities to seal or evacuate buildings following recent seismic events [4, 13]. Previous studies have reported that a substantial portion of Sylhet's building stock—including residential, commercial, and government facilities—is at risk of severe structural damage in the event of a major earthquake [14, 13].

Objective: Develop an interpretable, data-driven framework to predict seismic drift in RC buildings by coupling NLTHA-generated response data from *OpenSees* with explainable AI (SHAP, LIME), enabling robust and computationally efficient risk assessment for Bangladesh's urban building stock.

Chapter 2

Literature Review

This chapter provides a comprehensive review of the existing literature on seismic risk assessment, vulnerability analysis, and recent advances in data-driven and machine learning approaches for earthquake engineering. It examines key developments in traditional methods such as Rapid Visual Screening (RVS) and the Seismic Vulnerability Index (SVI), alongside the latest research leveraging artificial intelligence and explainable models. Emphasis is placed on previous studies relevant to the Bangladeshi context, identifying knowledge gaps and motivating the need for the present research.

Seismic drift—defined as the relative lateral displacement between stories—is a critical measure for evaluating structural performance under earthquake loads. Historically, its prediction has relied on classical structural analysis techniques. Among them, Response Spectrum Analysis (RSA) has remained a foundational tool due to its computational efficiency and practicality in estimating peak responses across different vibration modes [16, 17, 18]. RSA transforms complex time-domain ground motions into a simplified spectral representation, enabling rapid assessments for both single and multi-degree-of-freedom (SDOF/MDOF) systems.

However, more detailed and accurate representations of structural behavior have been achieved using Nonlinear Time History Analysis (NLTHA), which simulates the complete time-domain response of structures under recorded or synthetic seismic inputs. Although NLTHA captures material and geometric nonlinearities effectively, its computational intensity limits widespread application in large-scale or real-time scenarios [19]. Similarly, Pushover Analysis, a nonlinear static method, has been widely used to estimate inelastic demand and drift distribution. It offers insight into failure mechanisms and is compatible with performance-based design frameworks, although it oversimplifies the dynamic effects of strong-motion records [20].

Codified procedures such as FEMA 356 [6] provide structured guidelines for performance evaluation but are largely built upon linear or simplified nonlinear methods. These methods often underestimate drift, particularly under near-fault or pulse-like motions. As Chopra noted, idealized assumptions within such deterministic analyses limit their applicability to real-world seismic events [16].

Despite their foundational role in earthquake engineering, traditional analytical methods exhibit several limitations when applied to drift prediction. Most notably, they struggle to capture the full nonlinear, path-dependent behavior of materials, especially under strong shaking or for irregular building configurations [21]. Furthermore, they often neglect uncertainties related to ground motion variability, site amplification, and structural degradation, which significantly influence drift outcomes.

The simplifications inherent in these models limit their reliability for risk-informed decision-making, particularly when dealing with critical infrastructure or complex structural systems. As a result, researchers have increasingly turned to data-driven techniques that can better accommodate uncertainty and leverage empirical patterns in observed or simulated structural response.

To address these challenges, early studies began exploring statistical modeling and machine learning (ML) to improve prediction accuracy. A pioneering work by Ghaboussi et al. (1991) introduced neural networks for modeling nonlinear material behavior, demonstrating the feasibility of learning complex constitutive relationships directly from data [22]. This breakthrough laid the groundwork for ML adoption in structural engineering.

Further progress was made by Adeli and Wu (1998), who expanded the use of intelligent algorithms to construction planning and cost estimation, highlighting ML's potential in various civil engineering domains [23]. As computational power increased and large structural datasets became available—particularly from platforms like OpenSees and the PEER NGA ground motion database—the application of ML in seismic response prediction gained momentum [24, 19].

Initial ML applications to seismic prediction focused on shallow learning algorithms such as Linear Regression (LR), Support Vector Machines (SVM), and Decision Trees (DT). These models were attractive due to their simplicity, interpretability, and ease of deployment. However, LR is inherently limited in representing the nonlinearities typical of seismic structural response [25].

SVMs can model nonlinear patterns using kernel functions, but they require careful tuning and may be sensitive to data scaling. In contrast, decision-tree-based methods, including Random Forests (RF) and Gradient Boosted Decision Trees (GBDT), offer higher flexibility in capturing nonlinear interactions between features. Research by Khosravikia et al. (2020) and Gholizadeh (2016) showed that ensemble tree methods, when trained on well-prepared datasets, significantly outperform traditional deterministic models in seismic drift prediction [26, 25].

Moreover, XGBoost, an advanced GBDT implementation, has demonstrated superior performance in handling high-dimensional, incomplete, and imbalanced datasets—common issues in earthquake engineering. Case studies by Ma et al. (2020) and Sun et al. (2022) confirmed that these models exhibit better generalization than shallow learners and can be effectively used in fragility analysis and damage prediction [27, 28].

With the growing volume of high-resolution structural data, deep learning (DL) models have gained traction. Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) are increasingly used to model complex seismic behavior. These models can learn hierarchical and spatiotemporal features from large datasets, making them ideal for analyzing seismic time series [29].

Recent research has also proposed hybrid models that combine domain knowledge with data-driven learning. Such Physics-Informed Machine Learning (PIML) approaches embed physical constraints—e.g., conservation laws or equilibrium equations—into ML architectures, enhancing both prediction accuracy and interpretability. Studies by Shokr et al. (2021), Kiani et al. (2021), and Mousavi et al. (2019) demonstrated the benefits of integrating DL with structural principles, particularly in predicting peak and residual drift [30, 31, 32].

More notably, Raissi et al. (2019) and Sadeghi et al. (2023) highlighted that PIML can help mitigate data scarcity and improve extrapolation beyond the training domain, which is essential for engineering safety applications [29, 33].

The success of ML models in predicting seismic drift relies on several interrelated factors:

- **Data Sources:** High-quality input is critical. These include synthetic data from nonlinear simulations (e.g., OpenSees) [19], experimental results from shaking tables, and real earthquake recordings from databases like PEER NGA [24].
- **Feature Engineering:** Common features include ground motion intensity measures (e.g., PGA, PGV, Sa), soil conditions (e.g., Vs30), and structural attributes (e.g., height, lateral load system, number of stories). Bojórquez et al. (2018) emphasized the importance of selecting appropriate intensity measures, while Zareian and Krawinkler (2005) validated the necessity of Incremental Dynamic Analysis (IDA) in estimating drift across varying hazard levels [17, 21].
- **Model Transparency:** Interpretability is essential in safety-critical applications. Tools like SHAP (SHapley Additive Explanations) [34] and LIME enable engineers to understand feature contributions and justify ML decisions. Chakraborty & Dey (2021) demonstrated how interpretability aids in adopting ML for seismic risk assessment [35].

Despite promising advances, several challenges remain:

1. **Lack of Drift-Specific Models:** Most existing studies focus on collapse probability or acceleration, with limited emphasis on precise drift prediction under different seismic intensities.

2. **Data Scarcity:** Especially for extreme ground motions and highly damaged states, datasets are sparse and often insufficient for training deep models [24].

3. **Generalizability:** ML models trained on one structural typology or seismic region may not perform well in other contexts without significant retraining or transfer learning.

4. **Benchmark Datasets:** There is a shortage of publicly available benchmark datasets for consistent model comparison and validation.

Addressing the limitations of prior research demands more than just advanced algorithms; it necessitates the creation of standardized, high-quality datasets, the identification of drift-specific features, and the development of hybrid frameworks that blend domain expertise with the adaptability of machine learning. Notably, the adoption of physics-informed machine learning (PIML) techniques represents a significant innovation, enabling predictive models to honor underlying physical laws while achieving improved generalizability and robustness [33].

The progression of seismic drift prediction reflects an important paradigm shift in civil engineering—from traditional deterministic, rule-based methods to flexible, data-driven modeling. While conventional approaches still underpin code-based design and fundamental understanding, state-of-the-art machine learning, including ensemble algorithms, deep learning, and hybrid physics-informed architectures, have shown clear advantages in predictive accuracy and adaptability, especially when supported by comprehensive datasets and rigorous validation.

The novelty of this thesis lies in the development of **interpretable, hybrid machine learning models** uniquely tailored for seismic drift prediction. These models leverage both synthetic and real-world data and incorporate explainable AI (XAI) tools to provide transparency and actionable insights. By focusing on model interpretability alongside predictive power, this work not only advances the technical state-of-the-art but also delivers practically deployable tools that enhance the accuracy, reliability, and trustworthiness of seismic risk assessment and decision-making.

Chapter 3

Methodology

This chapter details the methodology adopted to achieve the research objectives. It begins with an outline of the dataset generation process, including nonlinear dynamic analyses performed using OpenSees to simulate seismic responses of building models. Data preprocessing steps, feature selection strategies, and handling of missing values are described in detail. The supervised machine learning models employed for seismic drift prediction are introduced, followed by an explanation of model training, validation, and performance evaluation procedures. The chapter also covers the application of explainable AI (XAI) techniques to interpret model outputs.

3.1 Dataset Creation and Construction Workflow

The foundation of this research is the creation and meticulous preparation of a high-fidelity, simulation-based dataset tailored for advanced machine learning applications in earthquake engineering. Unlike datasets that are simply assembled from existing measurements, the dataset developed in this study was systematically engineered through a two-stage process designed to maximize both predictive power and physical interpretability.

The process began with the development of a synthetic input dataset for OpenSees simulations. This dataset was programmatically structured to capture a broad spectrum of reinforced concrete (RC) frame buildings subjected to diverse seismic scenarios. Each record encoded critical structural parameters—including the number of stories, story height, bay width, and the cross-sectional properties of beams and columns—as well as essential material characteristics such as concrete compressive strength (f'_c) and steel yield strength (f_y). These parameters were systematically varied to ensure representation across both low-rise and high-rise RC frames, thereby enabling robust generalization and meaningful analysis of key structure-seismic interactions. In figure 3.1, the total workflow of the study.

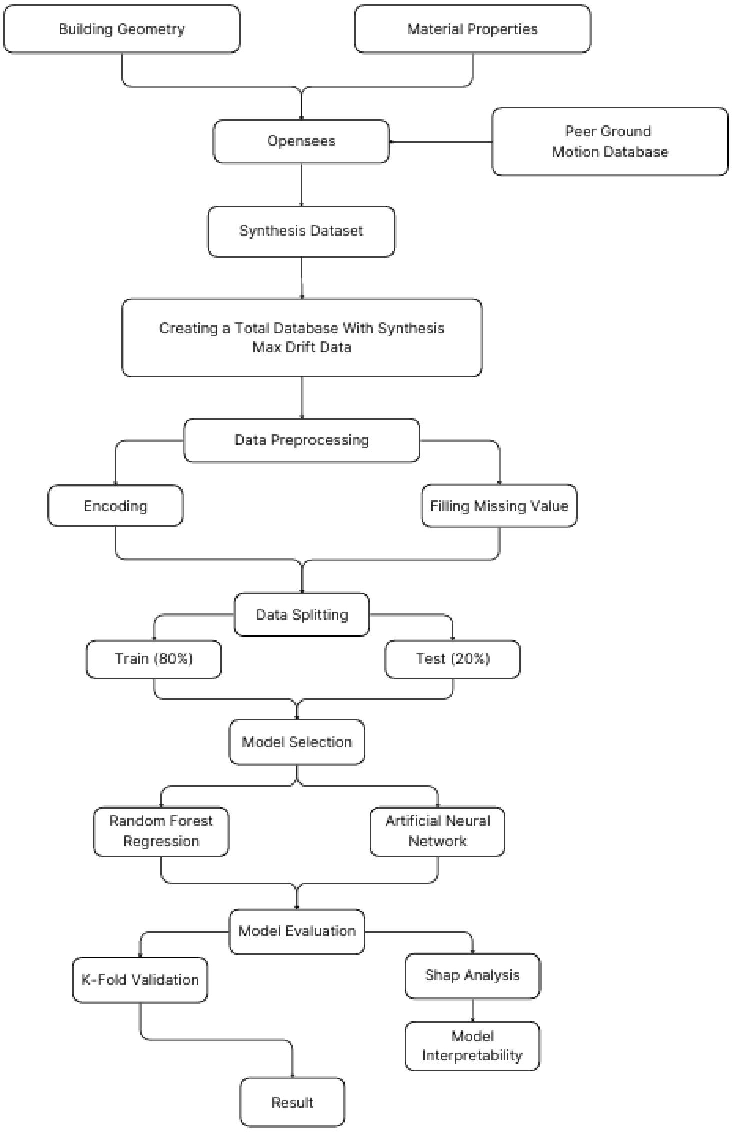

Figure 3.1: Total Study Workflow Diagram

Following the generation of these structural configurations, each unique building model was subjected to a suite of ground motion records, selected to encompass a comprehensive range of seismic scenarios. These records were characterized by varying peak ground acceleration (PGA), peak ground velocity (PGV), peak ground displacement (PGD), earthquake magnitude (Mw), rupture distance (Rrup), and spectral content, thus simulating the multifaceted effects of real-world earthquakes and capturing the complex, nonlinear drift responses inherent in actual structures.

The results of these simulations were aggregated into a second, machine learning (ML) analysis dataset. This final dataset integrated all relevant structural and material properties, ground motion features, site characteristics (such as V_{s30} and FaultType), and, crucially, structural response metrics including period, spectral response (SR), and, most importantly, the maximum inter-story drift ratio (MaxDrift) under seismic loading. MaxDrift was designated as the primary target variable for subsequent ML modeling, while all other features were curated to ensure both statistical informativeness and physical meaning.

This carefully designed two-stage data generation pipeline—rooted in physics-based simulation yet optimized for data-driven modeling—forms a robust and transparent foundation for all subsequent analyses. It enables this research to move beyond traditional black-box prediction, facilitating nuanced interpretation and advancing transparency in the inference of seismic responses from both structural and seismological perspectives.

3.2 Nonlinear Dynamic Analysis of a 10-Story RC Frame using OpenSeesPy

A detailed nonlinear time-history analysis (NLTHA) of a 10-story, three-dimensional reinforced concrete (RC) moment-resisting frame was conducted using the OpenSeesPy framework. The analytical model was constructed to simulate realistic seismic behavior and generate response data for subsequent machine learning applications.

Model Geometry and Layout. The prototype structure consists of 10 stories and a plan configuration of 3×3 bays in both the X and Y directions. The typical story height is 3.5 m, with a bay width of 5.0 m in both horizontal directions. Nodes were defined at all beam-column intersections, from the base (ground level) to the roof, resulting in a grid of $(\text{numStories} + 1) \times (\text{numBaysX} + 1) \times (\text{numBaysY} + 1)$ nodes. Base nodes were fully fixed to restrain all translational and rotational degrees of freedom.

Material Properties. Concrete and steel material nonlinearities were modeled explicitly. Unconfined and confined concrete fibers utilized the `Concrete01` material, with characteristic compressive strengths, peak strains, and crushing strengths representative of conventional RC construction. Longitudinal and transverse reinforcement were defined using the `Steel02` model, with a yield strength of 420 MPa and appropriate strain hardening and degradation parameters. These properties ensure realistic capture of both strength and ductility under cyclic loading.

Section Definition. Fiber-section models were adopted for both beams and columns to accurately simulate distributed nonlinearities and capture the spread of plasticity. Column sections measured 500 mm $\times$ 500 mm and beams 250 mm $\times$ 400 mm, with geometric discretization for concrete cover, confined core, and longitudinal reinforcement. The `patch` and `layer` commands in OpenSeesPy were employed to model the composite behavior of RC sections.

Element Discretization. The frame elements were modeled using the `nonlinearBeamColumn` element, with a sufficient number of integration points along each element to ensure numerical convergence. Columns were oriented vertically, while beams spanned along both X and Y directions at each floor.

Gravity Loads and Mass Assignment. Dead, live, and superimposed floor loads (including slab, finishes, partitions, and a portion of the live load) were computed and applied as uniformly distributed nodal forces. Lumped masses were assigned to each floor node based on the tributary area and equivalent mass per node.

Boundary and Diaphragm Constraints. Rigid diaphragm constraints were imposed at each floor to enforce in-plane rigidity, ensuring that floor nodes at each level move together in the X and Y directions. This simulates the effect of a stiff floor slab and reduces numerical degrees of freedom.

Damping. Rayleigh damping was specified to achieve a 5% critical damping ratio, calculated from the first and third natural frequencies of the structure, as obtained from eigenvalue analysis.

Ground Motion Input. A real ground motion record from the PEER NGA database was scaled to 1.0g and applied at the base of the structure as a uniaxial excitation in the X-direction using the `UniformExcitation` pattern. The input motion's time history was discretized according to its native time step.

Nonlinear Dynamic Analysis. A nonlinear time-history analysis was carried out using the Newmark integration method. Advanced convergence algorithms were incorporated, including Newton, Modified Newton, and KrylovNewton, with automated switching to ensure robust convergence at each time step. The entire simulation covered the full duration of the input ground motion.

Response Recording and Post-processing. Inter-story drift ratios were recorded at each time step using the `Drift` recorder. The resulting drift time histories were processed to extract the maximum inter-story drift ratio, which serves as a primary metric for seismic performance assessment and as the target variable for subsequent machine learning regression. Additionally, nodal displacements were recorded for verification and further analysis. Results were visualized by plotting the drift time history, enabling validation and interpretation of dynamic structural response.

This robust simulation workflow provided the high-fidelity structural response data necessary for the data-driven modeling phase, ensuring that both input features and output targets reflect realistic seismic behavior as dictated by physics-based modeling principles.

Figures 3.2–3.12 depict the complete OpenSeesPy modeling workflow, beginning with library

```python
import openseespy.opensees as ops
import numpy as np
import matplotlib.pyplot as plt

# Units: kN, m, s
ops.wipe()
ops.model('basic', '-ndm', 3, '-ndf', 6)  # 3D model with 6 DOF per node
```

Figure 3.2: step1

```python
import openseespy.opensees as ops
import numpy as np
import matplotlib.pyplot as plt

# Units: kN, m, s
ops.wipe()
ops.model('basic', '-ndm', 3, '-ndf', 6)  # 3D model with 6 DOF per node

# Define Geometry and Nodes
bayWidthX = 5.0    # Bay width in X direction (m)
bayWidthY = 5.0    # Bay width in Y direction (m)
storyHeight = 3.5  # Story height (m)
numStories = 10    # Number of stories
numBaysX = 3       # Number of bays in X direction
numBaysY = 3       # Number of bays in Y direction
```

Figure 3.3: step2

```python
# Create Nodes
for i in range(numStories + 1):  # Levels from 0 (base) to numStories
    for j in range(numBaysX + 1):  # X direction
        for k in range(numBaysY + 1):  # Y direction
            nodeTag = i * 10000 + j * 100 + k
            ops.node(nodeTag, j * bayWidthX, k * bayWidthY, (numStories - i) * storyHeight)

# Fix Base Nodes
for j in range(numBaysX + 1):
    for k in range(numBaysY + 1):
        ops.fix(j * 100 + k, 1, 1, 1, 1, 1, 1)

# Define Materials
# Concrete (unconfined and confined) - Concrete01
fc = 21.0         # Compressive strength (MPa)
epsc0 = 0.002     # Strain at peak stress
fcu = 0.1 * fc    # Crushing strength
epscu = 0.004     # Ultimate strain
ops.uniaxialMaterial('Concrete01', 1, -fc, -epsc0, -fcu, -epscu)
ops.uniaxialMaterial('Concrete01', 2, -1.3*fc, -0.004, -0.2*fc, -0.01)  # Confined concrete
```

Figure 3.4: step3

```python
# Define Materials
# Concrete (unconfined and confined) - Concrete01
fc = 21.0          # Compressive strength (MPa)
epsc0 = 0.002      # Strain at peak stress
fcu = 0.1 * fc     # Crushing strength
epscu = 0.004      # Ultimate strain
ops.uniaxialMaterial('Concrete01', 1, -fc, -epsc0, -fcu, -epscu)
ops.uniaxialMaterial('Concrete01', 2, -1.3*fc, -0.004, -0.2*fc, -0.01)  # Confined concrete

# Steel - Steel02
fy = 420.0         # Yield strength (MPa)
Es = 2.0e5         # Elastic modulus (MPa)
b = 0.01           # Strain hardening ratio
R0 = 20            # Initial curvature parameter
cR1 = 0.925        # Curvature degradation parameter
cR2 = 0.15         # Curvature degradation parameter
ops.uniaxialMaterial('Steel02', 3, fy, Es, b, R0, cR1, cR2)
```

Figure 3.5: step4

```python
# Define Fiber Sections
# Column Section (500x500 mm) with 10 rods of 20 mm
ops.section('Fiber', 1, '-GJ', 1e6)
ops.patch('rect', 1, 5, 15, -0.25, -0.25, -0.05, 0.25)  # Left cover
ops.patch('rect', 1, 5, 15, 0.05, -0.25, 0.25, 0.25)    # Right cover
ops.patch('rect', 1, 10, 5, -0.25, -0.25, 0.25, -0.05)  # Bottom cover
ops.patch('rect', 1, 10, 5, -0.25, 0.05, 0.25, 0.25)    # Top cover
ops.patch('rect', 2, 10, 10, -0.20, -0.20, 0.20, 0.20)  # Core
ops.layer('circ', 3, 10, 0.02, 0.20, 0.0, 0.0)          # 10 rods of 20 mm

# Beam Section (250x400 mm) with 6 rods of 16 mm
ops.section('Fiber', 2, '-GJ', 1e6)
ops.patch('rect', 1, 5, 12, -0.125, -0.20, -0.025, 0.20)  # Left cover
ops.patch('rect', 1, 5, 12, 0.025, -0.20, 0.125, 0.20)    # Right cover
ops.patch('rect', 1, 10, 4, -0.125, -0.20, 0.125, -0.04)  # Bottom cover
ops.patch('rect', 1, 10, 4, -0.125, 0.04, 0.125, 0.20)    # Top cover
ops.patch('rect', 2, 8, 8, -0.10, -0.16, 0.10, 0.16)      # Core
ops.layer('straight', 3, 3, 0.016, -0.10, 0.20, 0.10, 0.20)  # 3 rods of 16 mm (top)
ops.layer('straight', 3, 3, 0.016, -0.10, -0.20, 0.10, -0.20) # 3 rods of 16 mm (bottom)
```

Figure 3.6: step5

```python
# Beams (X direction)
for i in range(numStories + 1):
    for j in range(numBaysX):
        for k in range(numBaysY + 1):
            nodeLeft = i * 10000 + j * 100 + k
            nodeRight = i * 10000 + (j + 1) * 100 + k
            ops.element('nonlinearBeamColumn', eleTag, nodeLeft, nodeRight, 2, 5, 1)
            eleTag += 1

# Beams (Y direction)
for i in range(numStories + 1):
    for j in range(numBaysX + 1):
        for k in range(numBaysY):
            nodeLeft = i * 10000 + j * 100 + k
            nodeRight = i * 10000 + j * 100 + (k + 1)
            ops.element('nonlinearBeamColumn', eleTag, nodeLeft, nodeRight, 2, 5, 1)
            eleTag += 1
```

Figure 3.7: step6

```python
# Define Masses and Loads
wLive = 2.4525     # Live load (kN/m²)
wSlab = 2.9430     # Slab self-weight (kN/m²)
wFinish = 0.9810  # Floor finishes (kN/m²)
wPartition = 1.4715  # Partition walls (kN/m²)
floor_area = 15.0 * 15.0  # Floor area (m²)
wTotal = 1.0 * (wSlab + wFinish + wPartition) + 0.25 * wLive  # Total load (kN/m²)
FloorLoad = wTotal * floor_area  # Total load per floor (kN)
P = FloorLoad / ((numBaysX + 1) * (numBaysY + 1))  # Uniform load per node (kN)
mass = P / 9.81  # Lumped mass per node (kN-s²/m)

for i in range(1, numStories + 1):
    for j in range(numBaysX + 1):
        for k in range(numBaysY + 1):
            nodeTag = i * 10000 + j * 100 + k
            ops.mass(nodeTag, mass, mass, 0.0, 0.0, 0.0, 0.0)
            ops.load(nodeTag, 0.0, 0.0, -P, 0.0, 0.0, 0.0)
```

Figure 3.8: step7

```python
# Add Rigid Diaphragm Constraints
for i in range(1, numStories + 1):
    baseNode = i * 10000
    for j in range(1, numBaysX + 1):
        for k in range(1, numBaysY + 1):
            nodeTag = i * 10000 + j * 100 + k
            ops.equalDOF(nodeTag, baseNode, 1, 2, 4, 5)  # Constrain UX, UY, RX, RY

# Gravity Analysis
ops.constraints('Plain')
ops.numberer('RCM')
ops.system('BandGeneral')
ops.test('NormDispIncr', 1.0e-6, 6)
ops.algorithm('Newton')
ops.integrator('LoadControl', 0.1)
ops.analysis('Static')
ops.analyze(10)
ops.loadConst('-time', 0.0)
```

Figure 3.9: step8

```python
# Define Damping
xi = 0.05     # 5% damping ratio
m1 = ops.eigen(1)[0]
m3 = ops.eigen(3)[0]
a0 = xi * 2.0 * (m1 * m3) ** 0.5 / (m1 + m3)
ops.rayleigh(a0, 0.0, 0.0, 0.0)

# Define Ground Motion (X-direction only, scaled to 1.0g)
GMfileX = "kobe_Amagasaki_000.at2"  # Replace with your PEER NGA file
with open(GMfileX, 'r') as f:
    lines = f.readlines()
    npts = float(lines[3].split()[0])  # Number of points
    dt = float(lines[3].split()[1])    # Time step
ops.timeSeries('Path', 1, '-filePath', GMfileX, '-dt', dt, '-factor', 1.0)  # Scaled to 1.0g
ops.pattern('UniformExcitation', 1, 1, '-accel', 1)  # X direction only
```

Figure 3.10: step9

```python
# Nonlinear Time History Analysis with Convergence Handling
ops.constraints('Transformation')
ops.numberer('RCM')
ops.system('UmfPack')
ops.test('NormDispIncr', 1.0e-6, 10, 2)  # Increased iterations and norm check
ops.algorithm('Newton')
ops.integrator('Newmark', 0.5, 0.25)
```

Figure 3.11: step10

```python
# Record Drift Time History
ops.recorder('Drift', '-file', 'drift.out', '-time', '-nodeRange', 10000, 110000, '-dof', 1, '-perpDirn', 2)
drift_data = np.loadtxt('drift.out')
time = drift_data[:, 0]
drift = drift_data[:, 1]  # First drift value (X-direction)

# Plot Drift Time History
plt.figure(figsize=(10, 6))
plt.plot(time, drift, label='X-Drift Ratio', color='red')
plt.xlabel('Time (s)')
plt.ylabel('Drift Ratio')
plt.title('Inter-Story Drift Time History')
plt.legend()
plt.grid(True)
plt.savefig('drift_time_history.png', dpi=300, bbox_inches='tight')
plt.close()
```

Figure 3.12: step11

imports and geometry definition in the initial stages and concluding with nonlinear time-history analysis and drift extraction in the final stage.

Table 3.1: Feature List and Physical Significance

Feature Name	Symbol / Unit	Type	Physical Meaning and Importance
Natural Period	T (s)	Numeric	Fundamental vibration period; reflects flexibility and resonance behavior of structure.
Strength Ratio	SR	Numeric	Yield strength-to-weight ratio; key for nonlinear/plastic response.
Peak Ground Acceleration	PGA (g)	Numeric	Max ground acceleration; principal driver of inertial force demand.
Peak Ground Velocity	PGV (cm/s)	Numeric	Max ground velocity; relates to input energy and potential damage.
Peak Ground Displacement	PGD (cm)	Numeric	Max ground displacement; crucial for displacement-sensitive systems.
Magnitude	M_w	Numeric	Earthquake magnitude; controls total released energy and shaking level.
Rupture Distance	R_{rup} (km)	Numeric	Closest distance to fault rupture; affects amplitude and frequency content.
Arias Intensity	AI (m/s)	Numeric	Cumulative ground motion energy; indicator of potential damage.
Shear Wave Velocity (30m)	V_{S30} (m/s)	Numeric	Average near-surface soil stiffness; key for site amplification.
Lowest Usable Frequency	(Hz)	Numeric	Minimum frequency with reliable data; controls long-period content.
Duration	(s)	Numeric	Significant duration of shaking; affects cumulative structural response.
Fault Type	–	Categorical	Source mechanism (e.g., strike-slip, reverse); impacts shaking features.
Spring Stiffness	K_s (kN/m)	Numeric	Retrofit parameter; provides recentering capability and stiffness.
Damper Yield Moment	M_r (kN·m)	Numeric	Retrofit parameter; governs energy dissipation of SRFD.

3.3 Dataset Overview

A comprehensive synthetic dataset was generated in Table3.1 following the methodology of Masoum M. Gharagoz et al. [36], incorporating various structural, seismic, and retrofit parameters. The dataset enables robust training and validation of machine learning models for seismic performance assessment and retrofit optimization.

Table 3.2: Summary of Synthetic Dataset Used for Seismic Assessment

Parameter	**Value / Description**
Number of datapoints	500
Source of earthquake records	PEER NGA Database (2021)
Analysis method	Nonlinear Time History Analysis (NLTHA)
Software used	OpenSees, MATLAB, Python
Main output/target	Maximum Drift (D)

3.4 Dataset Description

The descriptive statistics of 14 of the most crucial variables in this study are provided in Table 3.2 and represent a sample consisting of 500 data records of ground motion–structure response. These include period (T), site response (SR), peak ground acceleration (PGA), peak ground velocity (PGV), peak ground displacement (PGD), moment magnitude (M_w), rupture distance (R_{rup}), Arias intensity (AI), average shear-wave velocity to 30 m depth (V_{s30}), low-frequency content (LowFreq), significant duration, stiffness (K_s), mass ratio (M_r), and maximum drift (MaxDrift).

The structural period (T) exhibits a wide range of structural flexibility, varying between 0.31 s and 2.98 s, with a mean of 1.65 s and a standard deviation (SD) of 0.81 s. SR, the factor of amplification of a site, ranges between 0.81 and 1.28, with a mean of 1.046 and SD of 0.057, indicating relatively consistent site conditions with respect to amplification. The mean PGA is 0.485 g, with a minimum of 0.20 g and a maximum of 0.75 g, reflecting a varied range of shaking intensities.

PGV and PGD, which measure the velocity and displacement demands of ground motion, have means of 103.11 cm/s and 72.45 cm, respectively, with broad ranges (PGV: 15.19–203.27 cm/s; PGD: 4.67–144.14 cm). The seismic source parameters show a mean M_w of 5.93 (ranging from 4.2 to 7.61), representing both moderate and large-magnitude earthquakes. The range of R_{rup} is between 3.11 km and 120.67 km, with an average of 61.80 km, indicating the inclusion of both near-field and far-field ground motion records in the dataset.

The Arias intensity (AI), a measure of the total seismic energy, varies between 0.001 m/s and 1.19 m/s, with a mean of 0.58 m/s. The V_{s30} values range from 170.0 m/s to 1425.2 m/s, with a mean of 783.08 m/s, indicating a diversity of site conditions from soft soils to hard rock. LowFreq ranges between 0.32 Hz and 3.74 Hz, with an average of 1.89 Hz, thus covering both low-frequency and high-frequency prevailing motions.

Significant duration varies substantially (2.20–83.20 s) with a mean of 42.21 s, reflecting both short impulsive events and long-duration motions. The structural property parameters K_s (stiffness) and M_r (mass ratio) also show great variability, with K_s ranging from 200 kN/m to 1200 kN/m (mean 607.20 kN/m) and M_r ranging from 1 to 37 (mean 24.91).

Finally, the main structural demand parameter in this study, MaxDrift, ranges from 0.023% to 19.94%, with a mean of 3.08% and a standard deviation of 3.19%. This wide range indicates that the dataset includes structural responses from negligible drift demands to severe inelastic deformation requirements.

Table 3.3: Descriptive statistics of key dataset variables (n=500).

	T	SR	PGA	PGV	PGD	Mw	Rrup	AI	Vs30	LowFreq	Duration	Ks	Mr	MaxDrift
Count	500	500	500	500	500	500	500	500	500	500	500	500	500	500
Mean	1.65	0.146	0.485	103.11	72.45	5.93	61.60	0.58	783.08	1.89	42.21	607.00	24.91	3.08
Std	0.81	0.057	0.16	57.28	41.34	1.01	33.36	0.35	355.78	1.08	23.21	441.41	16.58	3.19
Min	0.31	0.051	0.20	4.67	0.38	4.20	3.13	0.001	170.0	0.03	2.20	100	1	0.23
25%	0.95	0.096	0.33	52.15	38.93	5.09	32.34	0.28	472.53	0.95	24.09	100	13	1.18
50%	1.69	0.144	0.50	105.59	71.90	5.92	64.09	0.55	799.15	1.84	40.97	600	25	1.90
75%	2.34	0.20	0.63	151.19	107.67	6.83	88.46	0.90	1048.78	2.86	61.60	1200	37	3.39
Max	2.98	0.25	0.75	203.27	144.14	7.61	120.67	1.19	1425.2	3.74	83.20	1200	50	19.94

3.5 Data Preprocessing

3.5.1 Handling Missing Values

The completeness and reliability of input data are very important preconditions to a sound machine learning pipeline. In the setting of structural engineering datasets produced by simulation-based data, missing values can occur due to multiple possible causes, such as non-convergence of numerical simulations, incomplete data collection of ground motion parameters, or software-specific artifacts when data are written to an export file. Unless addressed, such missing values may compromise the training and the evaluation phases, because most state-of-the-art machine learning algorithms require the feature matrices to be fully populated.

A methodical manner of imputing missing values was adopted in order to counter such risks and maintain the integrity of data. The data were first scanned to measure the level of missingness proportion and the pattern in all numerical features. The dataset had approximately 17 missing values, with most of them found in numerical features. These were treated using mean imputation, in which all the missing entries were filled with the arithmetic average of the available values in the corresponding column. The mean imputation formula used is Equation(3.1):

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.1}$$

where $\bar{x}$ is the mean, x_i are the known (non-missing) values, and n is the number of those values. This method helps preserve the central tendency of the data and significantly reduces bias introduced by imputation. The use of mean imputation is particularly suitable for simulation based datasets, where the missingness is typically random rather than systematic.

Although rare, in cases where missing values remained in categorical fields after the primary imputation—most likely due to non-numeric values or encoding inconsistencies—a secondary imputation step was performed. In this step, missing categorical entries were replaced either by the mode (i.e., the most frequent category) or, where applicable, a neutral dummy value (e.g., zero).

This two-tiered approach ensured that the final dataset was completely free of missing values across all features, enabling seamless integration with downstream machine learning workflows. Such rigor in data preparation is essential not only for algorithmic stability, but also for ensuring the reproducibility and transparency of the research.

3.5.2 Categorical Variable Encoding

The incorporation of categorical features in machine learning models presents unique methodological considerations, particularly when these features encode physically meaningful distinctions. In this study, categorical variables such as `FaultType`—representing the seismic fault mechanism (e.g., strike-slip, thrust, normal)—are essential for embedding domain knowledge and reflecting the full complexity of earthquake-structure interactions.

To facilitate the use of categorical data in machine learning algorithms, all categorical features were converted into numeric representations using `scikit-learn`'s `LabelEncoder`. This tool systematically assigns a unique integer to each distinct category within a feature. Label encoding preserves the categorical nature of variables and ensures compatibility with a wide array of models, including tree-based algorithms (such as Random Forests and XGBoost, which can natively handle integer-encoded categories) and neural networks (which, while often benefiting from one-hot or embedding-based encodings, can also operate with label-encoded variables for a limited number of categories).

Care was taken to preserve the integrity of the original categorical information, ensuring that no artificial ordering or spurious hierarchy was introduced. Within the scope of this research, label encoding offers a pragmatic balance among interpretability, computational efficiency, and methodological rigor. For future work or larger-scale datasets, advanced encoding methods—such as target encoding or learned embeddings—may be explored, especially as the number of categories or their interactions increases.

By numerically encoding all categorical variables prior to model training, the dataset was rendered fully compatible with all downstream analysis and modeling procedures. This step not only satisfies technical requirements but also enhances the reproducibility, interpretability, and scientific credibility of the research pipeline.

3.5.3 Exploratory Data Analysis: Feature Distributions

Figure 3.13 presents the distributions of major input features and the target (`MaxDrift`) in the constructed dataset. Most predictors—including T, SR, PGA, PGV, and PGD—exhibit pronounced right-skewness or multimodality, capturing the natural variability found in building and seismic properties. In Table 3.4 showing all the unit of feature's parameters

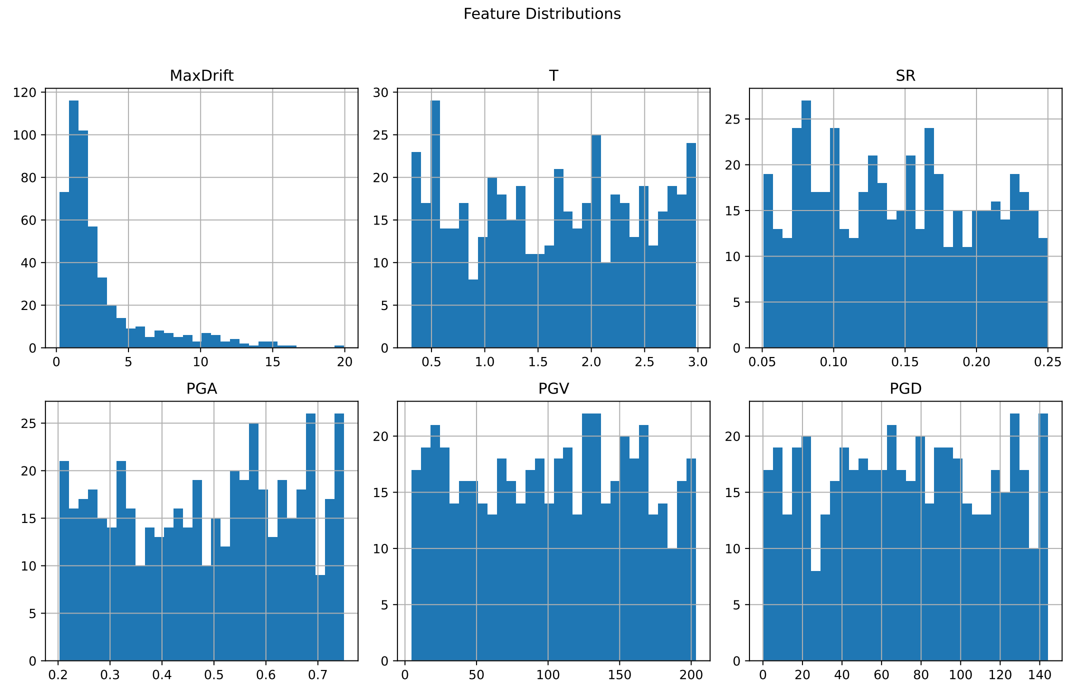

Figure 3.13: Distribution plots of selected input features and the target variable (`MaxDrift`) in the synthetic dataset.

Table 3.4: Description and units of selected input features and target variable (MaxDrift).

Variable	Description	Unit
MaxDrift	Maximum inter-story drift ratio	% (percent) or dimensionless
T	Fundamental period of vibration	seconds (s)
SR	Site response coefficient / spectral ratio	dimensionless
PGA	Peak ground acceleration	g (1 g $\approx$ 9.81 m/s^2)
PGV	Peak ground velocity	cm/s
PGD	Peak ground displacement	cm

The MaxDrift distribution shows a strongly right-skewed tendency with most of the values clustering above zero with a lack of frequency increasing continuously the greater the magnitude of drift. This means that there are extreme drift events but they are low highly proportional to realistic structural behavior since most of the cases exhibit slight failure deformation although a few percent exhibit significant drift due to intensive load.

The T (fundamental period) distribution ranges very wide and many peaks indicating multimodal distribution. This is probably an indicator of the presence of buildings of different levels of stiffness, elevation, and dynamic behavior, therefore, covering a representative range of building types in the data.

Based on the definition, SR (strength ratio/strain rate) has a coverage across an area within its domain seeming to be slightly uneven but wide. The variation on the bin height indicates the presence of both frequent and a few parameter values that contributes to exposing the model to a variety of structural capacity condition.

The distribution of PGA (Peak Ground Acceleration) is somewhat homogeneous with slight variations, which would cover ground motions with both low and high values of acceleration intensities. This is uniformity helps equalize training of the machine learning model because cases that are mild or extreme are not overrepresented.

The PGV (Peak Ground Velocity) also represents a distribution within the same range with minimum inconsistencies, which alludes to implicit variability in generated or recorded ground motions. Because of the presence of values varying in low to high magnitudes that occur in the data of PGV, the variations occurring in the data can have different effects on structural response compared to those of the PGA.

The distribution of PGD (Peak Ground Displacement) ones the entire range of the displacement amplitudes, and frequency difference between the bins is quite small. This variety makes sure that variations in the dataset entertains motions with small and large permanent displacements which is a key element in the models that predicts responsiveness to the drift.

Taken together, these distributions allow one to see a carefully selected dataset that is meant to display diversity in the characteristics input as well as the target variable, in order to resemble the heterogeneity present in real-life seismic and structural parameters. The skewness and even multimodality among features indicates the necessity of using reliable, versatile machine learning that can accommodate non-normal, non-homogeneous data distributions. Such diversity does not increase the functionality of generalization of the trained models only but also provides the possibility to assume that predictions would be accurate within a broad range of seismic scenarios.

3.5.4 Feature Scaling

For algorithms that are sensitive to the scale of input features, such as artificial neural networks (ANNs), all continuous predictors were standardized prior to model training. Standardization transforms each feature to have a mean of zero and a standard deviation of one, ensuring that each variable contributes equally to the learning process and preventing features with larger scales from disproportionately influencing the model. The standardization process is formally defined by Equation (3.2):

$$x' = \frac{x - \mu}{\sigma} \tag{3.2}$$

where x is the original feature value, μ is the mean, and σ is the standard deviation of the feature within the training set. This normalization is crucial for the efficient optimization of neural network parameters and for achieving faster convergence during training.

In contrast, tree-based algorithms such as Random Forests are inherently invariant to monotonic transformations of input data, and thus, are capable of operating effectively on raw or minimally scaled features without significant performance degradation.

3.5.5 Correlation Heatmap

The correlation matrix in Figure 3.14 provides insights into the linear relationships among input variables and the target variable, `MaxDrift`. Among all predictors, the period (T) exhibits the strongest correlation with `MaxDrift` ($r = -0.71$), indicating that longer structural periods are generally associated with lower drift ratios. Peak ground velocity (PGV) also shows a moderate positive correlation ($r = 0.31$) with `MaxDrift`, suggesting that higher PGV values tend to increase drift demands. Peak ground displacement (PGD) and moment magnitude (M_w) display weak positive correlations ($r = 0.16$ and $r = 0.04$, respectively). All other variables, including PGA, AI, Vs30, and source–site parameters, have negligible linear correlations with `MaxDrift` ($|r| < 0.1$). These observations highlight that while T and PGV are key individual predictors, the overall predictive performance likely benefits from capturing complex, non-linear interactions among multiple features.

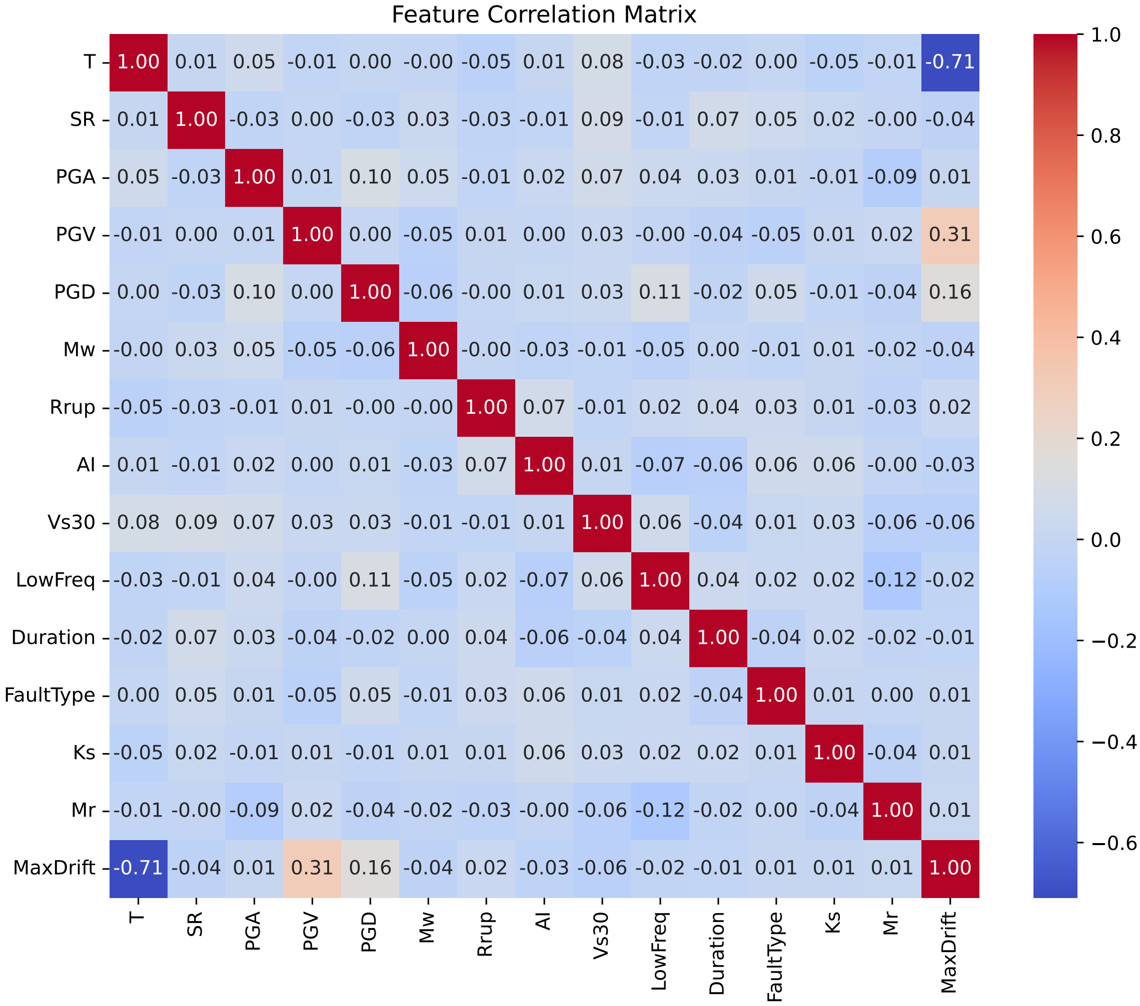

Figure 3.14: Data-Heatmap

3.6 Dataset Features and Their Significance

The main features (input variables) used in the dataset, as extracted and refined from the referenced study, are presented in Table 3.1. Each feature is included due to its physical or empirical significance in controlling structural seismic response, and its criticality has been validated through feature selection methods such as Recursive Feature Elimination (RFE).

3.6.1 Importance of Each Feature

- **Natural Period (T):** Determines the dynamic properties of the structure and its likelihood of resonance with specific seismic waves. Longer periods generally imply greater flexibility and potential for large displacements.

- **Strength Ratio (SR):** Quantifies the relative strength of the structure; critical for predicting whether the building will remain elastic or undergo plastic deformations during strong shaking.

- **Peak Ground Acceleration (PGA):** Directly governs the maximum inertial force imposed on the structure; higher PGA correlates with greater damage potential.
- **Peak Ground Velocity (PGV):** Highly correlated with structural and non-structural damage, especially for mid-rise buildings.
- **Peak Ground Displacement (PGD):** Significant for flexible structures and for assessing potential for permanent deformation or failure.
- **Magnitude (M_w):** Represents the earthquake's size and potential destructiveness.
- **Distance to Rupture Surface (R_{rup}):** Closer distances yield stronger shaking and shorter warning times.
- **Arias Intensity (AI):** Captures cumulative shaking energy, which correlates with the probability of structural and nonstructural damage.
- **Shear Wave Velocity (V_{s30}):** Proxy for local soil conditions; softer soils can amplify ground motion.
- **Lowest Usable Frequency:** Indicates if the ground motion contains significant low-frequency energy, which is particularly damaging to taller, flexible buildings.
- **Duration:** Extended shaking can lead to increased fatigue and cumulative damage, even if peak values are moderate.
- **Fault Type:** Different fault mechanisms produce distinct shaking patterns and frequency contents, influencing structural response.
- **Spring Stiffness (K_s):** Retrofit parameter; critical for controlling post-earthquake residual drift and enabling rapid functionality recovery.
- **Yield Moment of Friction Damper (M_r):** Governs the onset of energy dissipation in the retrofit system, a major factor in controlling maximum and residual drift.

3.7 Ground Motion Record Selection and Processing

Eighty natural ground motion records were selected from the **PEER NGA-West2** database to ensure realistic seismic inputs. These records were chosen to cover a wide range of intensity measures and seismological parameters, including peak ground acceleration (PGA:

0.2–0.75 g), peak ground velocity (PGV: 4–204 cm/s), moment magnitude (M_w: 4.2–7.62), rupture distance (2.9–120 km), and site conditions (Vs_{30}: 170–1428 m/s), as summarized in Table 2. Fault mechanisms (strike-slip, reverse, normal, and oblique) were also included to capture diverse seismic scenarios by Euqation(3.3):

Key ground motion intensity measures, computed for each record, include:

- **Peak Ground Acceleration (PGA):** max $|\ddot{u}_g(t)|$
- **Peak Ground Velocity (PGV):** max $|\dot{u}_g(t)|$
- **Peak Ground Displacement (PGD):** max $|u_g(t)|$
- **Arias Intensity (AI):**

$$AI = \frac{\pi}{2g} \int_0^T [\ddot{u}_g(t)]^2 dt \tag{3.3}$$

- **Significant Duration (D_{5-95}):** Time interval over which 5% to 95% of Arias intensity is accumulated.

All ground motion files were formatted and preprocessed for compatibility with OpenSees dynamic analysis input using custom Python scripts.

3.8 Dataset Partitioning

The dataset was partitioned into distinct training and testing subsets, with 80% of the samples allocated to the training set and the remaining 20% reserved for testing. This widely adopted 80/20 split provides a practical balance between model development and unbiased evaluation. By assigning the majority of the data to the training set, the model is afforded sufficient information to learn complex patterns and relationships within the dataset. The independent test set, comprising 20% of the data, serves as an objective benchmark for assessing model generalization and predictive accuracy on previously unseen samples. This proportion is considered optimal in many machine learning applications, as it mitigates the risk of overfitting while ensuring that model evaluation metrics reflect real-world performance. All model performance metrics reported in this study, including mean absolute error (MAE), mean squared error (MSE), and the coefficient of determination (R^2), were computed using the held-out test set to ensure robust and unbiased validation.

3.9 Machine Learning Model Development and Explainability

For this study, we chose two models: one is a random forest tree-based model, and the other is a Neural network-based model.

3.9.1 Random Forest Regression Model

An ensemble-based regression model (**Random Forest**) was developed by aggregating the predictions of multiple decorrelated decision trees. Each tree was constructed using bootstrap sampling of the dataset and random feature selection at each split, with the ensemble output obtained by averaging the predictions from all trees. The model minimizes the following loss function for each tree: Equation (3.4).

follows:

$$\mathcal{L} = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{3.4}$$

where y_i is the observed drift and $\hat{y}_i$ is the predicted drift. The architecture of the model is showing as an example in Figure 3.15.

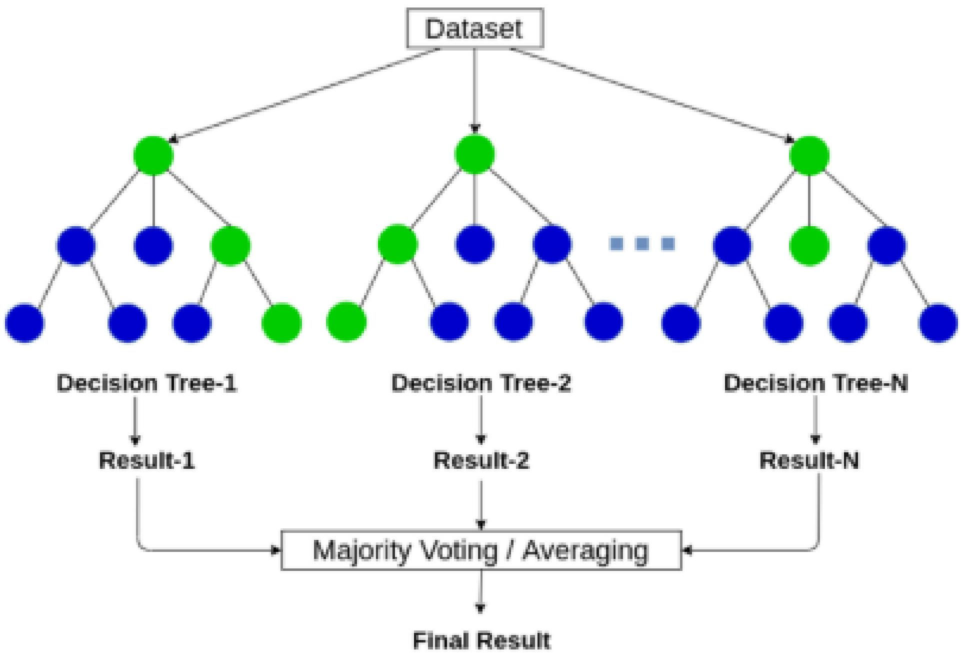

Figure 3.15: Random Forest Architecture

The performance evaluation metrics for the Random Forest regression model were computed as shown in Figure 3.16.

```python
# Import necessary libraries
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score, mean_absolute_percentage_error

# Assume X_train, X_test, y_train, y_test are already defined

# Train the model
rf = RandomForestRegressor(n_estimators=200, random_state=42)
rf.fit(X_train, y_train)

# Make predictions
y_pred_rf = rf.predict(X_test)

# Calculate evaluation metrics
mae = mean_absolute_error(y_test, y_pred_rf)
mse = mean_squared_error(y_test, y_pred_rf)
rmse = mean_squared_error(y_test, y_pred_rf, squared=False)
r2 = r2_score(y_test, y_pred_rf)
mape = mean_absolute_percentage_error(y_test, y_pred_rf)

# Print the results
print(f"MAE: {mae:.3f}")
print(f"MSE: {mse:.3f}")
print(f"RMSE: {rmse:.3f}")
print(f"R2 Score: {r2:.3f}")
print(f"MAPE: {mape:.3f}")
```

Figure 3.16: Python code for Random Forest regression model training and evaluation.

3.9.2 Deep Neural Networks (DNNs)

An artificial neural network (ANN) regression model was developed to capture the complex, nonlinear relationships present in the structural engineering dataset. The model was constructed using the `Keras` API within the `TensorFlow` framework, employing a deep, fully connected feedforward architecture with integrated regularization techniques to enhance generalization and mitigate overfitting. The architecture of the model is showing as an example in Figure 3.17.

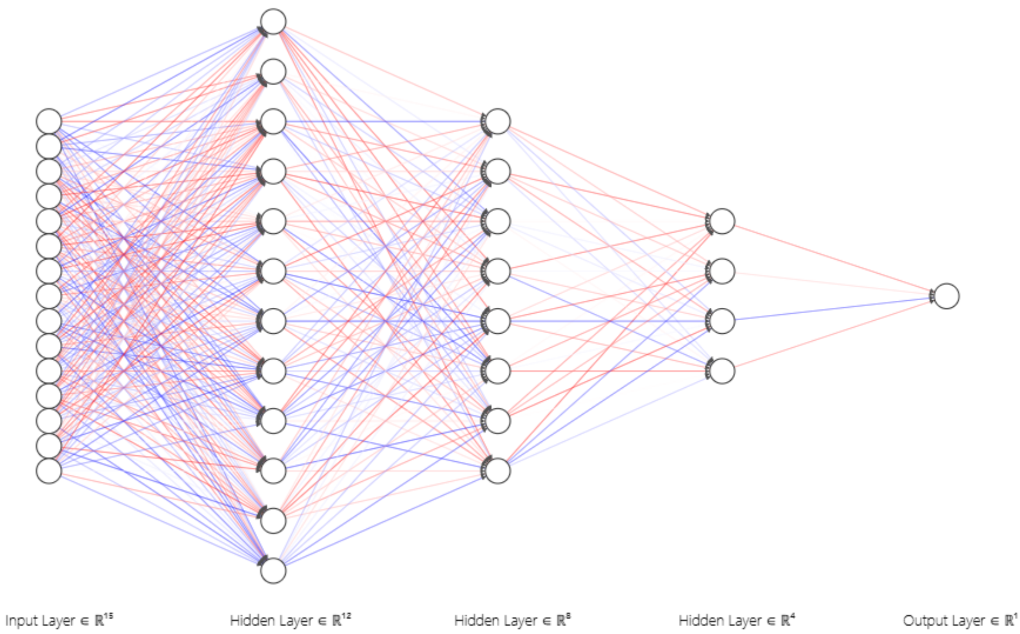

Figure 3.17: Deep neural Network Architecture

The network architecture consisted of five hidden layers with progressively decreasing numbers of neurons: 512, 256, 128, 64, and 32, respectively. Each dense layer was followed by batch normalization to stabilize the learning process and accelerate convergence, and Leaky Rectified Linear Unit (LeakyReLU) activations ($\alpha = 0.1$) were used to introduce nonlinearity while addressing the vanishing gradient problem. Dropout layers with rates of 0.3 (for larger layers) and 0.2 (for smaller layers) were interleaved after each activation to reduce overfitting by randomly omitting a subset of neurons during training. In addition, L2 regularization ($\lambda = 0.001$) was applied to all dense layers to penalize large weight values and improve model robustness.

The output layer comprised a single neuron with linear activation, producing the predicted drift value for each sample Equation(3.5) as:

$$\hat{y} = f(\mathbf{x}; \Theta) \tag{3.5}$$

where $\mathbf{x}$ represents the standardized input feature vector and Θ denotes the set of trainable network parameters (weights and biases).

The ANN was trained to minimize the mean squared error (MSE) loss function, augmented with L2 regularization Equation(3.6) follows:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 + \lambda \sum_{l=1}^{L} \|W^{(l)}\|_2^2 \tag{3.6}$$

where y_i and $\hat{y}_i$ are the observed and predicted drift values for the i-th sample, L is the total number of layers, $W^{(l)}$ denotes the weight matrix of the l-th layer, and λ is the regularization parameter. The first term in Equation (3.6) measures the data fitting error, while the second term penalizes model complexity to prevent overfitting.

The deep artificial neural network architecture and its training configuration are illustrated in Figure 3.18.

```python
# 2. Define ANN Model
model = Sequential([
    Dense(512, kernel_regularizer=l2(0.001), input_shape=(X_train_scaled.shape[1],)),
    BatchNormalization(), LeakyReLU(alpha=0.1), Dropout(0.3),
    Dense(256, kernel_regularizer=l2(0.001)),
    BatchNormalization(), LeakyReLU(alpha=0.1), Dropout(0.3),
    Dense(128, kernel_regularizer=l2(0.001)),
    BatchNormalization(), LeakyReLU(alpha=0.1), Dropout(0.2),
    Dense(64, kernel_regularizer=l2(0.001)),
    BatchNormalization(), LeakyReLU(alpha=0.1), Dropout(0.2),
    Dense(32, kernel_regularizer=l2(0.001)),
    BatchNormalization(), LeakyReLU(alpha=0.1), Dropout(0.2),
    Dense(1)  # Regression output
])
model.compile(optimizer='adam', loss='mse', metrics=['mae'])

# # 3. Callbacks
# early_stopping = EarlyStopping(monitor='val_loss', patience=20, restore_best_weights=True)
# reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=10, min_lr=1e-5)

# 4. Model Training
history = model.fit(
    X_train_scaled, y_train,
    validation_data=(X_test_scaled, y_test),
    epochs=1000,
    batch_size=32,
    # callbacks=[early_stopping, reduce_lr],
    verbose=1
)
```

Figure 3.18: Python code for defining and training the artificial neural network (ANN) regression model.

Model optimization was performed using the Adam algorithm, and mean absolute error (MAE) was also monitored as a supplementary metric. To further improve training efficiency and prevent overfitting, early stopping and learning rate reduction on plateau were employed as callbacks. Early stopping monitored the validation loss with a patience of 20 epochs and restored the best weights observed during training, while the learning rate was halved if the validation loss failed to improve for 10 consecutive epochs, subject to a minimum threshold of 1×10^{-5}.

The network was trained for up to 100 epochs with a batch size of 32, using the standardized training set. Model performance was ultimately evaluated on a held-out test set using mean absolute error (MAE), mean squared error (MSE), and the coefficient of determination (R^2). This deep ANN, combining normalization, regularization, and adaptive training strategies, was thus designed to provide robust and accurate predictions for complex simulation-driven regression tasks.

3.10 Explainable Artificial Intelligence (XAI) Methods

To enhance model transparency, interpretability, and engineering insight, two state-of-the-art explainable artificial intelligence (XAI) techniques were employed: SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME). These tools provide both global and local interpretability, allowing for a deeper understanding of the influence of input features on model predictions and facilitating trust in data-driven outcomes for structural engineering applications.

SHapley Additive exPlanations (SHAP) SHAP is a unified framework for interpreting machine learning predictions by quantifying the contribution of each input feature to the model's output, based on Shapley values from cooperative game theory. For a prediction $\hat{y}$ associated with input vector **x**, the SHAP value ϕ_j for feature j is formally shows Equation 3.7 as:

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{j\}}(\mathbf{x}_{S \cup \{j\}}) - f_S(\mathbf{x}_S) \right] \tag{3.7}$$

where F is the set of all features, S is a subset of features not containing j, f_S denotes the model prediction using only features in S, and $\mathbf{x}_S$ represents the values of **x** restricted to the subset S. This formulation ensures that the contributions of all features sum to the difference between the prediction and the expected model output in Equation 3.8 as:

$$f(\mathbf{x}) = E[f(\mathbf{x})] + \sum_{j=1}^{|F|} \phi_j \tag{3.8}$$

SHAP analysis enables both local interpretation—by attributing contributions to individual predictions—and global understanding through summary plots of feature importance across the dataset.

3.10.1 Local Interpretable Model-agnostic Explanations (LIME)

LIME provides local interpretability by constructing a simple, interpretable surrogate model (e.g., linear regression) that approximates the complex model's behavior in the neighborhood of a specific instance. For a given input $\mathbf{x}_0$, LIME perturbs the feature space around $\mathbf{x}_0$ and fits a local surrogate model g to approximate the original model f in the vicinity defined by a proximity measure $\pi_{\mathbf{x}_0}(\mathbf{z})$. The optimization objective for LIME is given by Equation 3.9 as:

$$\arg\min_{g \in G} \mathcal{L}(f, g, \pi_{\mathbf{x}_0}) + \Omega(g) \tag{3.9}$$

where G is the class of interpretable models (e.g., linear models), $\mathcal{L}$ measures the fidelity of g to f in the neighborhood of $\mathbf{x}_0$, and $\Omega(g)$ is a regularization term to ensure interpretability. The result is an explanation vector that highlights the most influential features driving the model's prediction for the selected instance.

By integrating both SHAP and LIME, this research achieved a comprehensive understanding of the machine learning model's behavior, enabling domain experts to validate, interpret, and trust the data-driven predictions in the context of structural engineering.

3.11 Evaluation Metrics

To rigorously assess the predictive performance of the developed machine learning models, three standard evaluation metrics were employed: Mean Absolute Error (MAE), Mean Squared Error (MSE), and the coefficient of determination (R^2 score). These metrics are widely recognized in both engineering and data science communities for evaluating regression models, and together provide a comprehensive perspective on model accuracy and explanatory power.

Mean Absolute Error (MAE) The Mean Absolute Error quantifies the average magnitude of prediction errors across all samples, without considering the direction of the errors. It is mathematically defined in Equation (3.10) as follows:

$$\mathrm{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{3.10}$$

where y_i and $\hat{y}_i$ denote the observed and predicted values for the i-th sample, respectively, and N is the total number of samples in the test set. The MAE provides an interpretable measure of the average absolute deviation of predictions from true values, expressed in the same units as the target variable.

Mean Squared Error (MSE) The Mean Squared Error is a widely used metric that penalizes larger errors more heavily by squaring the residuals between observed and predicted values. As shown in Equation (3.11):

$$\mathrm{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{3.11}$$

Lower MSE values indicate better model performance. MSE is also commonly used as a loss function during model training due to its desirable mathematical properties, including differentiability and sensitivity to outliers.

Coefficient of Determination (R^2 Score) The coefficient of determination, denoted as R^2, measures the proportion of variance in the observed data that is explained by the model's predictions. Its formal definition is given in Equation (3.12):

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2} \tag{3.12}$$

where $\bar{y}$ is the mean of the observed values. An R^2 value of 1 corresponds to perfect predictions, while a value of 0 indicates that the model's predictions are no better than simply predicting the mean of the observed values. Negative R^2 values suggest that the model performs worse than the baseline mean predictor.

Together, these metrics—defined in Equations (3.10), (3.11), and (3.12)—offer a robust evaluation framework for comparing different machine learning models and ensuring the reliability and practical utility of the predicted seismic drift responses in structural engineering applications.

3.12 Optimal Parameter Identification and Validation

The trained and interpreted model was then used to identify optimal SRFD retrofit parameters (K_s, M_r) such that the predicted maximum inter-story drift ratio δ_{max} remained below the code-specified life-safety limit (typically 2%). This process was validated via targeted case studies, demonstrating significant reductions in both peak and residual drifts for retrofitted frames.

Algorithm 1 ML-Based Seismic Drift Prediction and Retrofit Optimization

Require: RC frame archetypes, SRFD parameter ranges (K_s, M_r), ground motions from PEER NGA-West2
Ensure: Trained explainable ML model and optimal retrofit parameters

```
Structural Model: Define RC frames and SRFD system in OpenSees
Ground Motion: Select and preprocess N ground motions
for each ground motion do
    Compute intensity measures (PGA, PGV, PGD, M_w, R_rup, AI, V_S30, duration, fault type)
end for
for each combination of T, SR, K_s, M_r do
    for each ground motion do
        Run NLTHA in OpenSees
        Extract max drift δ_max
        Store features and results
    end for
end for
Normalize all features (min–max scaling)
Apply RFE (Random Forest) for feature selection (scikit-learn)
Train Random Forest regressor, optimize (Optuna), validate (cross-val, MAE, R^2)
Use SHAP/LIME for model interpretability and decision rule extraction
Identify optimal (K_s, M_r) for code-compliant drift; validate with NLTHA
return Explainable seismic drift prediction and retrofit optimization framework
```

3.12.1 Software and Tools Used

- **OpenSees** (`https://opensees.berkeley.edu/`): Nonlinear dynamic analysis and structural modeling
- **Python** (v3.8+): Data processing, feature extraction, batch automation, machine learning, and visualization
- **PEER NGA-West2 Database**: Ground motion records (`https://ngawest2.berkeley.edu/`)
- `scikit-learn`, `xgboost`, `SHAP`, `LIME`, `Optuna`: ML and explainability libraries (Python)
- `numpy`, `pandas`, `matplotlib`, `seaborn`: Data science and plotting libraries (Python)

This integrated workflow enabled the creation of a robust, physics-informed, and fully interpretable framework for seismic drift prediction and retrofit optimization.

Chapter 4

Results and Discussions

This chapter presents the key findings from the application of machine learning models to the seismic drift dataset. Model performance metrics are reported and compared, with a focus on the accuracy and interpretability of the predictions. The effectiveness of explainable AI methods (SHAP, LIME) in identifying influential features is analyzed.

In the quest for reliable and interpretable data-driven models in structural earthquake engineering, the Random Forest (RF) regressor was selected as the foundational machine learning approach for the prediction of seismic drift responses. The rationale for this choice is rooted in the intrinsic characteristics of the Random Forest algorithm, which is widely recognized for its robustness, flexibility, and capacity to model complex, nonlinear relationships without requiring extensive feature engineering or transformation.

The RF, an ensemble method based on the aggregation of multiple decision trees, brings several advantages to this domain. Firstly, its ensemble nature inherently mitigates the risk of overfitting, which is especially crucial when dealing with simulation-derived datasets that may contain subtle patterns, noise, or outlier cases. By averaging the predictions from a diverse set of trees—each constructed from a bootstrap sample of the data and a random subset of features—RF naturally balances bias and variance, delivering predictions that are both stable and reliable across a wide spectrum of input scenarios.

Secondly, the RF's ability to model nonlinear interactions and capture complex dependencies between structural, ground motion, and site features makes it particularly suited for the seismic drift prediction task. Unlike linear models, which can only represent additive relationships, the RF can implicitly discover and leverage intricate feature combinations that may drive structural response under earthquake loading. This capacity is of paramount importance in earthquake engineering, where physical phenomena are rarely linear and often influenced by higher-order effects.

From a practical perspective, the RF algorithm is largely invariant to the scaling of input features. This insensitivity simplifies the preprocessing workflow, as features with disparate units and magnitudes (such as Vs30 in meters/second, PGA in g, and FaultType as an encoded categorical variable) can be modeled without extensive normalization or standardization—an attractive property when working with multidisciplinary datasets.

For the present study, the model was trained on 80% of the available data, with the remaining 20% reserved for out-of-sample evaluation, following a randomized train-test split. The primary hyperparameters—most notably the number of trees (estimators) and the random seed—were selected based on a preliminary grid search, ensuring a balance between predictive power and computational efficiency. This preliminary tuning phase is critical for optimizing model performance, as insufficient tree numbers can yield underfitting, while excessive values can increase computation time with diminishing returns.

Upon training, the model's predictive capabilities were assessed using a comprehensive suite of statistical metrics: mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), the coefficient of determination (R^2), and mean absolute percentage error (MAPE). Each metric provides a distinct perspective on model performance—MAE offers an intuitive measure of average prediction error, MSE and RMSE penalize larger deviations (which may be critical in safety-sensitive engineering contexts), and R^2 quantifies the proportion of variance in the target (`MaxDrift`) that is explained by the model. MAPE, meanwhile, provides an error assessment relative to the scale of the observed values, allowing for practical interpretation in engineering terms.

The results obtained- notably an R^2 value of 0.897 and MAE of 0.491- represent a compelling level of predictive agreement between the model's outputs and the ground truth `MaxDrift` values obtained from simulation. These values indicate that the Random Forest regressor is highly effective in learning the dominant structural response patterns within the dataset. The high R^2, approaching 0.9, demonstrates that the model captures the overwhelming majority of variance in seismic drift outcomes, while the low MAE provides assurance that predictions are not only accurate on average but also precise in practical, engineering-relevant terms.

These strong performance indicators validate the methodological choice of the Random Forest as a baseline model. Moreover, they establish a benchmark against which more sophisticated machine learning models—such as deep artificial neural networks—can be meaningfully compared. The efficacy and transparency of the Random Forest approach thus serve as both a proof of concept for data-driven seismic drift prediction and a foundation for future methodological advancements in the field.

4.1 Model Predictive Performance Random Forest

The Random Forest regression model demonstrated strong predictive performance, as summarized by the evaluation metrics. The model achieved a mean absolute error (MAE) of 0.491 and a mean squared error (MSE) of 1.275, resulting in a root mean squared error (RMSE) of 1.129. The coefficient of determination (R^2 score) was 0.897, indicating that approximately 90% of the variance in the observed `MaxDrift` values was explained by the model. Additionally, the mean absolute percentage error (MAPE) was 0.143, reflecting a low average relative error across the test set. Collectively, these results confirm the Random Forest model's ability to accurately and reliably predict seismic drift responses, supporting its suitability as a baseline approach for data-driven structural engineering applications.

Visualizing predicted versus actual `MaxDrift` values provides an intuitive and interpretable assessment of model performance. As shown in Figure 4.1, each point represents a test scenario, with the x-axis indicating true values and the y-axis indicating Random Forest predictions. The 1:1 reference line serves as a benchmark for perfect prediction, enabling immediate identification of systematic deviations or outliers.

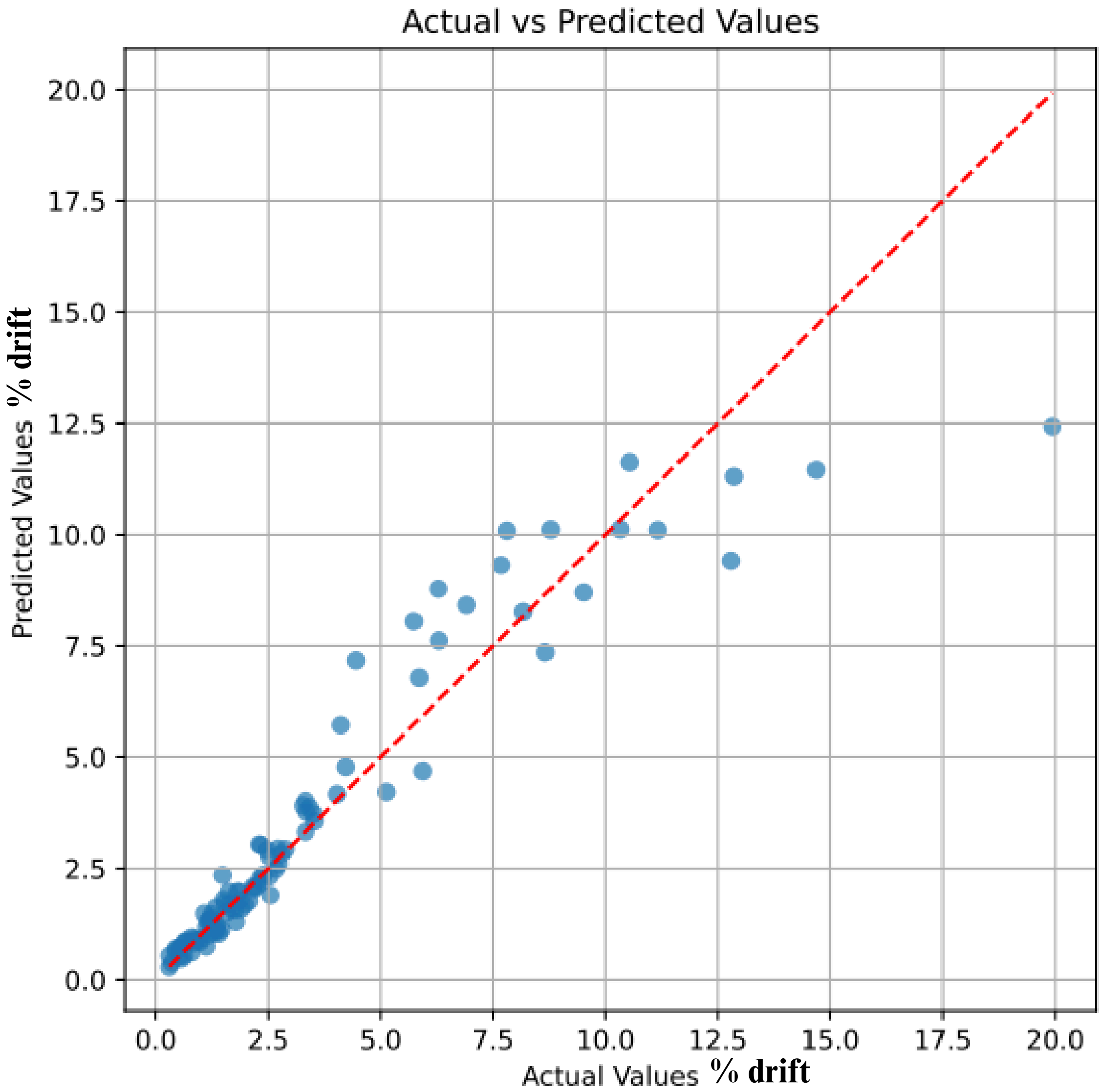

Figure 4.1: Random forest scatterplot for MaxDrift in Test Set

Table 4.1: Actual vs. Predicted `MaxDrift` Values

Sample	Actual MaxDrift	Predicted MaxDrift
1	1.646560	1.816225
2	1.897589	1.846822
3	10.327290	10.391180
4	4.455336	4.606060
5	0.622629	0.528180

The predictive capability of the Random Forest model is further illustrated in Table 4.1, which presents a comparison between the actual and predicted `MaxDrift` values for the top five test samples. The close agreement between these values highlights the model's strong accuracy in estimating seismic drift responses.

4.2 Model Predictive Performance Deep Neural Networks

The deep neural network (DNN) model achieved an R^2 score of 0.8680, a root mean squared error (RMSE) of 1.2750, and a mean absolute error (MAE) of 0.6999 on the test set. While these results reflect good predictive accuracy and a strong overall fit, the performance is slightly lower than that of the Random Forest model. This difference can be primarily attributed to the relatively modest dataset size of 500 samples. Deep neural networks, by design, are highly flexible and parameter-rich, requiring substantial amounts of data to fully realize their potential and avoid overfitting. In cases with limited data, ensemble methods such as Random Forests often outperform deep models, as they are more robust to data scarcity and better suited to capturing patterns in smaller datasets. Consequently, the observed results highlight the importance of matching model complexity to dataset size and suggest that, for the given sample size, tree-based ensemble methods provide a more reliable baseline for seismic drift prediction in structural engineering applications.

Figure 4.2 presents a scatter plot comparing the actual versus predicted `MaxDrift` values on the test set for the deep neural network (DNN) model. Each point corresponds to a test sample, with its color representing the absolute prediction error. The x-axis shows the true `MaxDrift` values, while the y-axis gives the model's predictions. The red dashed diagonal line denotes the 1:1 reference, indicating perfect agreement between predictions and observations. Most points cluster closely along this line, illustrating the ANN's strong predictive capability. The reported R^2 score of 0.8680 highlights that the model captures a substantial proportion of the variance in the target variable. The colorbar adds interpretability by visually localizing instances of higher error, which tend to occur at higher drift values. Overall, this visualization confirms both the reliability and the practical interpretability of the DNN model for seismic drift prediction.

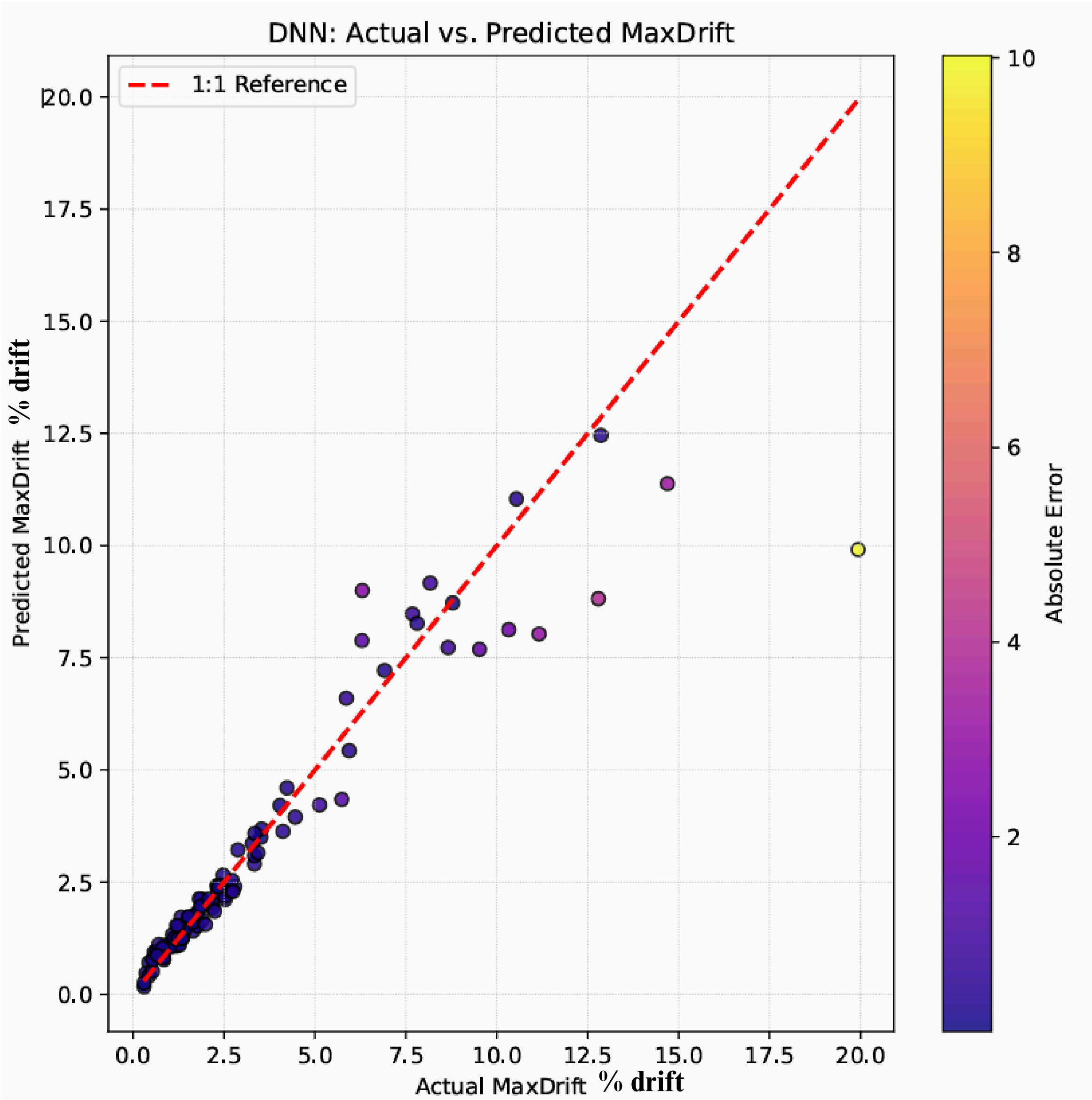

Figure 4.2: DNNs Scatter for MaxDrift in Test Set

The predictive performance of the DNN model is illustrated in Table 4.2, which presents the actual and predicted `MaxDrift` values for five representative test samples. The close alignment between predicted and true values demonstrates the model's reliability for seismic drift estimation.

Figure 4.3 illustrates the training and validation curves for mean squared error (MSE) loss and mean absolute error (MAE) over epochs for the ANN model. Both metrics show a rapid decrease during the initial epochs, followed by stabilization, indicating effective learning and convergence. The close alignment between training and validation curves in both plots suggests that the model generalizes well to unseen data, with no significant overfitting observed throughout the training process.

Table 4.2: *Comparison of Actual and Predicted `MaxDrift` Values*

Sample	Actual MaxDrift	Predicted MaxDrift	Absolute Error
1	1.646560	1.700936	0.054376
2	1.897589	1.592128	0.305461
3	10.327290	10.124481	0.202809
4	4.455336	7.174834	2.719498
5	0.622629	0.548713	0.073916

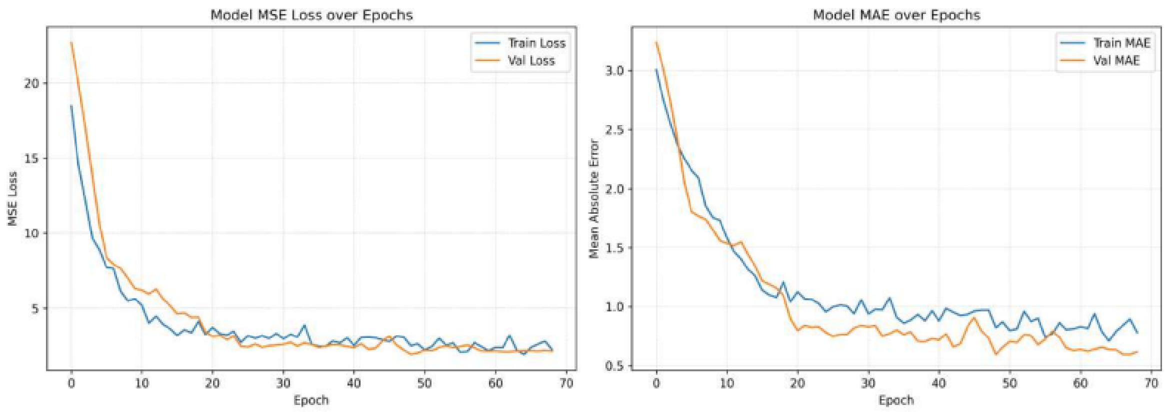

Figure 4.3: Training and validation loss (left) and MAE (right) curves over epochs, illustrating the model's convergence behavior, error reduction, and potential overfitting patterns

4.3 Model comparison

Table 4.3: Test–set performance comparison for MaxDrift prediction. Best scores are in bold.

Metric	RF	DNN
R^2	**0.897**	0.868
RMSE	**1.129**	1.275
MAE	**0.491**	0.6999

The Random Forest (RF) model outperforms the Deep Neural Network (DNN) across all evaluated metrics in Table 4.3. RF achieves a higher coefficient of determination ($R^2 = 0.897$) compared to the DNN ($R^2 = 0.868$), indicating that it explains a larger proportion of the variance in the observed `MaxDrift` values. Furthermore, RF exhibits lower error scores, with RMSE reduced from 1.275 (DNN) to 1.129 and MAE reduced from 0.6999 to 0.491. These results suggest that RF not only produces more accurate predictions on average (lower MAE) but also yields more stable and consistent performance (lower RMSE). The superior performance of RF is likely due to its robustness in small-to-moderate datasets, where ensemble-based tree models can effectively capture complex feature interactions while minimizing overfitting, making it a more reliable choice for seismic drift prediction in data-limited scenarios.

4.4 Cross Validations

To rigorously assess model generalization, R^2 scores were computed using both 5-fold and 10-fold cross-validation. The results, visualized in Figure 4.4, show the R^2 performance for each individual fold, as well as the mean and standard deviation for each cross-validation strategy. The 5-fold cross-validation yielded a mean R^2 of 0.909 with a standard deviation of 0.034, while the 10-fold approach resulted in a slightly higher mean R^2 of 0.916 and a lower standard deviation of 0.026. These findings, reflected by the consistently high R^2 values across all folds and the relatively low variability, indicate that the model demonstrates robust and reliable predictive performance regardless of data partitioning. The use of both 5-fold and 10-fold cross-validation provides strong evidence that the trained model is not overfitting and generalizes well to unseen data.

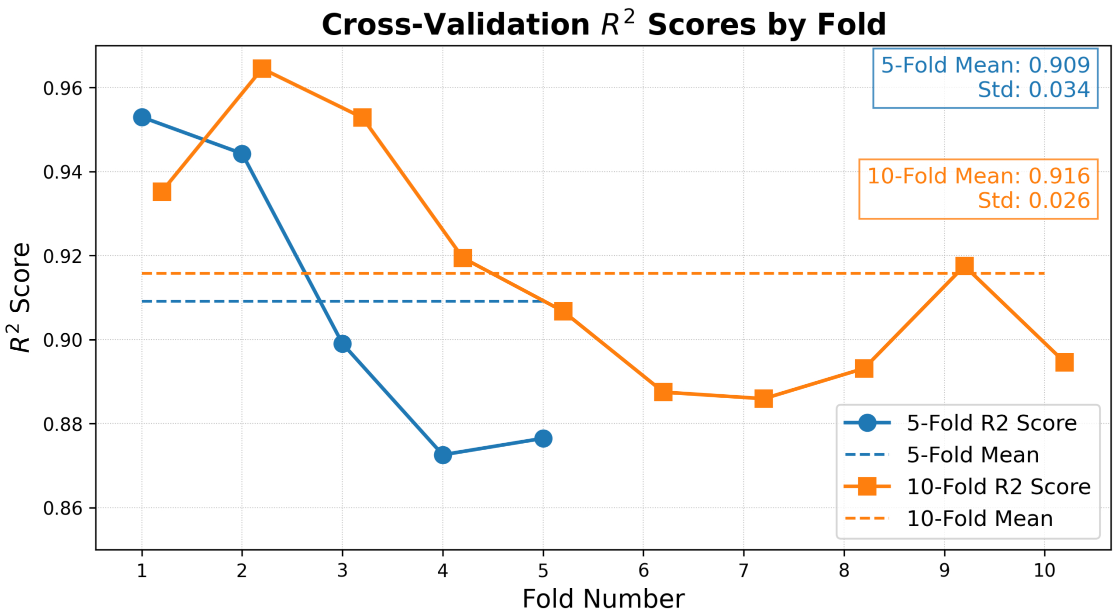

Figure 4.4: Cross Validation Result 5vs10 Fold

4.5 Model Interpretability

To enhance the interpretability of the XGBoost regression model, Shapley Additive exPlanations (SHAP) were utilized to analyze the relative importance and influence of each input feature. After training the XGBoost model on the available data, SHAP values were computed for the test set, quantifying the contribution of each feature to individual predictions. As illustrated in Figure 4.5, the SHAP summary (beeswarm) plot ranks features according to their overall impact on model output. The period (T), peak ground velocity (PGV), and peak ground displacement (PGD) are identified as the most influential variables affecting seismic drift predictions. The color gradient in the plot indicates the feature value for each instance, allowing visualization of both the magnitude and direction of each variable's effect. This analysis provides not only a global measure of feature importance but also engineering insight into how specific structural and ground motion parameters drive the XGBoost model's decisions.

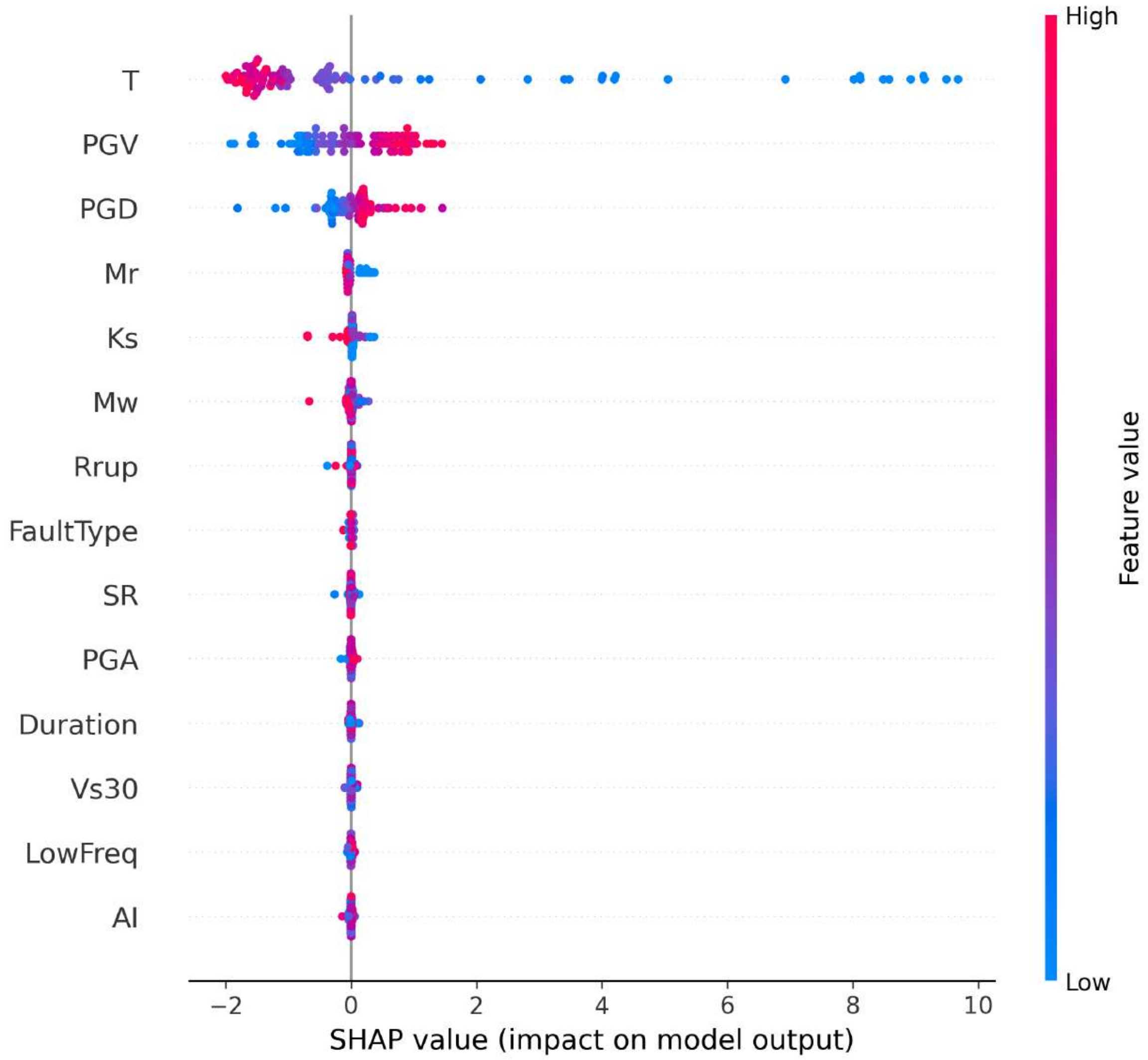

Figure 4.5: SHAP summary plot showing the relative importance and impact of input features on the model's MaxDrift predictions, with color indicating feature value.

Figure 4.6 displays the LIME explanation for a representative test sample, highlighting the contribution of individual features to the predicted value. For this instance, the model predicted a value of 162.62, with possible predictions ranging from 29.40 to 469.97. Features are ranked according to their positive (orange, right) or negative (blue, left) influence on the prediction. The largest negative impact arises from a low value of PGV (≤ 54.09), which reduces the prediction by 84.43 units, followed by Vs30 and AI. Conversely, high values of Ks (≤ 600.00), PGD (> 104.30), and LowFreq increase the predicted output. The table on the right details the actual feature values for this specific sample. Overall, this local interpretability analysis reveals which engineering parameters most strongly influenced the model's decision, providing both transparency and insight into the data-driven prediction process.

Figure 4.6: Lime Explanation

Interpretation. Figure 4.7 synthesizes three complementary metrics—coefficient of determination (R^2), root-mean-square error (RMSE), and mean absolute error (MAE)—to compare the performance of the Random Forest (RF) and Deep Neural Network (DNN) models. The RF attains a higher R^2 (0.897 vs. 0.868), indicating that it explains a larger share of the variance in `MaxDrift`. It also yields smaller errors: RMSE $= 1.129$ compared with 1.275 (approximately 11% lower), and MAE $= 0.491$ compared with 0.700 (approximately 30% lower). Collectively, these results show that RF produces predictions that are both *more accurate on average* (lower MAE) and *less dispersed* (lower RMSE), while capturing *more of the systematic signal* (R^2).

Why this matters for seismic drift. With a modest dataset (~500 samples), RF's bagging and randomized feature splits reduce variance and guard against overfitting, enabling reliable learning of nonlinear interactions among ground-motion and structural features. In contrast, the DNN's higher parameter capacity is harder to regularize in this data regime, leading to larger residuals. For practical purposes, this means RF offers a more *robust and dependable baseline* for drift prediction when data are limited, aligning with broader evidence that tree-based ensembles are strong default choices in small-to-medium tabular prediction problems.

Note: For a fair visual comparison on radar charts, it is advisable to orient axes so that "larger is better" for all metrics (e.g., plotting $1/(1+\text{RMSE})$ and $1/(1+\text{MAE})$ or min–max inverting the error axes) to avoid misinterpretation when error metrics are plotted in raw units.

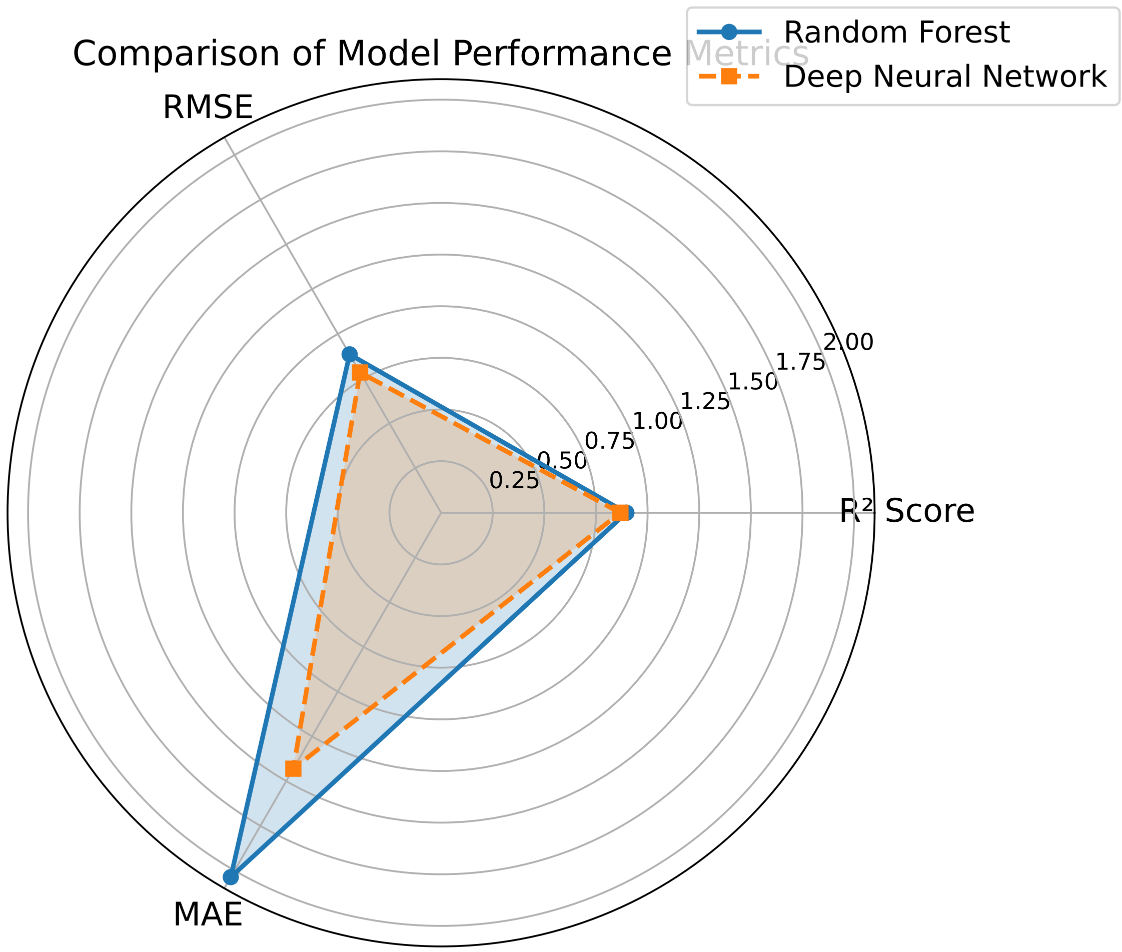

Figure 4.7: Radar Plot

Chapter 5

Conclusion

This chapter summarizes the major outcomes of the study, highlighting the contributions to seismic drift prediction and interpretable machine learning in earthquake engineering. The limitations encountered during the research are acknowledged, and recommendations for future studies are provided. The chapter concludes with practical suggestions for policymakers, engineers, and stakeholders seeking to enhance seismic safety and resilience in Bangladesh and other high-risk regions.

Seismic risk remains a defining challenge for civil engineering, particularly in rapidly urbanizing and tectonically vulnerable regions such as Bangladesh. The present thesis has addressed a critical facet of this challenge: **the accurate, efficient, and interpretable prediction of seismic drift responses in reinforced concrete (RC) buildings** using state-of-the-art data-driven methods. Through integrating physics-based simulation, machine learning, and explainable artificial intelligence (XAI), this research advances both the methodological foundation and the practical toolkit available for performance-based earthquake engineering.

Major Findings and Contributions

A **synthetic, high-fidelity dataset** was created via nonlinear time-history analyses (NLTHA) in OpenSees, systematically varying key structural parameters (e.g., period, strength ratio, retrofit characteristics) and exposing RC frames to a suite of realistic ground motions sourced from the PEER NGA database. The resulting data captured a broad spectrum of possible building responses, thus supporting robust model training and unbiased performance evaluation. Rigorous preprocessing—including imputation of missing values, label encoding of categorical variables, and careful feature scaling—ensured the integrity and usability of the dataset for ML applications.

Two machine learning models—a **Random Forest (RF) regressor** and a **deep artificial neural network (ANN)**—were developed, trained, and critically compared. The RF model demonstrated superior predictive performance, achieving an R^2 value of 0.897 and a mean absolute error (MAE) of 0.491 on the held-out test set, while the ANN achieved an R^2 of

0.868 and MAE of 0.6999. These results underscore the importance of matching model complexity to dataset size; ensemble tree-based methods like RF, with their natural regularization and robustness to overfitting, proved more reliable for the available sample size (500 datapoints). Cross-validation further confirmed the generalizability and stability of the RF model.

Interpretability—a central objective of this thesis—was addressed using **SHapley Additive exPlanations (SHAP)** and **Local Interpretable Model-agnostic Explanations (LIME)**. These XAI techniques illuminated the relative influence of features such as fundamental period, peak ground velocity, and retrofit parameters on drift predictions. The ability to quantitatively rank the importance of physical and seismic variables provides not only academic insight but also actionable guidance for engineers, policymakers, and retrofit designers.

Visualization tools—scatter plots, error curves, correlation heatmaps, and feature importance diagrams—further enhanced understanding of model behavior, supporting both the transparency and trustworthiness of the predictions. The proposed workflow is fully reproducible, modular, and scalable, laying the groundwork for further integration with field data and real-time applications.

Practical Applications

The developed machine learning-based workflow offers tangible benefits for engineers, policymakers, and disaster management professionals. In practice, the framework can be used for rapid seismic vulnerability screening of building inventories, prioritizing retrofits, and guiding code enforcement in urban planning. During seismic events, the methodology can support emergency response by quickly assessing likely structural impacts using available building and ground motion data. Additionally, interactive visualization tools and explainable predictions will help bridge the communication gap between technical experts and stakeholders, supporting transparent and defensible decision-making in both pre- and post-disaster contexts.

Final Remarks

This thesis demonstrates the practical potential and scientific value of integrating advanced machine learning with simulation-based structural engineering and explainable artificial intelligence. The proposed approach delivers accurate, transparent, and actionable predictions of seismic drift—offering a significant advance over traditional assessment methods. While limitations remain, the roadmap for future research is clear: continued expansion of data sources, refinement of modeling approaches, deepening of interpretability, and closer integration with engineering practice and societal needs. By pursuing these directions, data-driven methods will play an increasingly central role in safeguarding the built environment from earthquake risk—helping to realize the vision of resilient, sustainable, and disaster-ready cities in Bangladesh and beyond.

Chapter 6

Limitation and Future Work

Limitations

Despite promising results, this study has several key limitations:

1. **Data Size and Type:** The dataset is relatively small (500 samples) and based on simulations rather than real earthquake records, which may limit generalizability.

2. **Model Abstractions:** The structural models simplify certain complexities (e.g., soil-structure interaction, non-structural elements), which could affect accuracy for real buildings.

3. **Deep Learning Constraints:** Limited data restricted the full potential of the deep neural network, leading to better performance by the Random Forest model.

4. **Absence of Field Data:** The lack of real damage or sensor data may affect robustness in operational risk assessment.

5. **Interpretability Limits:** SHAP and LIME offer useful insights, but translating these into engineering practice remains a challenge.

Future Work

1. **External validation with real data.** Calibrate and test the models on instrumented buildings and post–earthquake records from Bangladesh (and nearby regions), including ambient-vibration tests to estimate T and event data for drifts.

2. **Richer hazard representation.** Extend inputs beyond peak IMs to include spectral-shape and duration measures (e.g., $S_a(T)$ or S_a-averages, significant duration, Arias intensity, CAV) and near-fault pulse parameters to better capture drift drivers.

3. **Broader structural typologies.** Add irregular RC frames, wall–frame systems, masonry infills, and retrofit devices (base isolation, viscous/friction dampers), and explicitly

model soil–structure interaction.

4. **Multi-fidelity & active design of experiments.** Combine fast analyses (pushover/SDOF surrogates) with selective NLTHA via Sobol/Latin-hypercube sampling and active learning/Bayesian optimization to target the most informative simulations.

5. **Uncertainty quantification (UQ).** Produce probabilistic drifts and prediction intervals using Bayesian neural nets, deep ensembles, or quantile regression; decompose aleatoric vs. epistemic uncertainty and propagate ground-motion variability.

6. **Physics-guided learning.** Impose monotonicity/shape constraints and embed spectral–displacement/resp relations as soft penalties or layers (hybrid physics–ML or PINN/PIML setups) to improve extrapolation.

7. **Sequence-based models.** Explore 1D CNNs/Transformers that ingest accelerograms directly to learn task-specific IMs; compare against feature-engineered pipelines and assess data needs/augmentation strategies.

8. **Domain adaptation & generalization.** Use transfer learning to adapt models across cities/soil classes; evaluate robustness to covariate shift and build region-specific vs. universal predictors for Bangladesh.

9. **From drift to risk.** Map predicted drifts to damage states/fragility and to loss/functional downtime; enable scenario-based and probabilistic risk assessment for urban portfolios and code-check workflows.

10. **Explainability at scale.** Go beyond global SHAP to interaction SHAP/ALE, counterfactual and contrastive explanations, and reliability/coverage diagnostics to stress-test interpretability claims.

11. **Efficient HPC pipeline.** Automate OpenSees/OpenSeesPy sweeps, GPU/cluster scheduling, and on-the-fly surrogate updates (MLOps) with versioned data/model cards and reproducible scripts.

12. **Time-dependent effects.** Incorporate aging, corrosion, and cumulative damage (mainshock–aftershock sequences, residual drift) to predict performance across building life cycles.

13. **Operational tool for stakeholders.** Package the model as a decision-support app (API/GUI) that ingests site/structure metadata and returns drifts with uncertainty, validity ranges, and suggested retrofits.

14. **Benchmarking and openness.** Release a standardized Bangladesh-focused drift dataset and protocols to promote fair benchmarking across ML/engineering communities.

References

[1] United Nations Office for Disaster Risk Reduction, "Un 20-year review: Earthquakes and tsunamis kill more people while climate change is driving up disaster numbers," 2020. Accessed: 2025-07-06.

[2] UNESCO, "Ipred knowledge - international platform for reducing earthquake disasters," 2024. Accessed: 2025-07-06.

[3] Evrim Ağacı, "Surge in earthquake activity sparks global alarm," 2024. Accessed: 2025-07-06.

[4] Thomson Reuters Foundation, "Bangladesh not ready for major earthquake, experts say," 2013. Accessed: 2025-07-06.

[5] U.S. Geological Survey, "Can you predict earthquakes?," 2024. Accessed: 2025-07-06.

[6] Applied Technology Council, "Review of past performance and further development of modeling techniques for collapse assessment of existing reinforced concrete buildings," Tech. Rep. ATC-95, Applied Technology Council, 2014. Available for download from the NIST website.

[7] Global Facility for Disaster Reduction and Recovery (GFDRR), "Disaster risk profile: Bangladesh." `https://www.gfdrr.org/sites/default/files/publication/Bangladesh%20v4.pdf`, 2016. Accessed: 2025-07-06.

[8] M. Ahmed, D. Bosu, R. I. Babul, A. A. Mahin, and M. A. Sayeed, "Seismic vulnerability assessment of buildings in sylhet city using rapid visual screening," *Available at SSRN 4967136*.

[9] Dhaka Tribune, "Sylhet quake: Expert calls for extensive survey, third-party vetting to reduce losses," *Dhaka Tribune*, Feb. 2023. Updated 19 Feb 2023; Accessed: 2025-07-06.

[10] M. B. Zisan, M. N. Haque, and M. A. Hasan, "Seismic vulnerability assessment of

masonry building supported by strp isolators," *Asian Journal of Civil Engineering*, vol. 24, pp. 121–136, 2023.

[11] The Daily Star, "National building code disregarded all along," 2019. Accessed: 2025-07-06.

[12] X. Shen, Y. Zhou, X. Li, N. Li, and G. Wu, "Resilience in the context of urban planning: A review," *International Journal of Disaster Resilience in the Built Environment*, vol. 12, no. 2, pp. 203–222, 2021.

[13] R. Chowdhury, "The rana plaza disaster and the complicit behavior of elite ngos," *Organization*, vol. 24, no. 6, pp. 938–949, 2017.

[14] Housing, B. R. I. (HBRI), and B. N. B. C. Authority, *Bangladesh National Building Code 2020*. Dhaka, Bangladesh: Government of the People's Republic of Bangladesh, 2020. Available from the Housing and Building Research Institute (HBRI).

[15] Comprehensive Disaster Management Programme, Ministry of Disaster Management and Relief; UNDP, "Seismic risk assessment in bangladesh," tech. rep., Government of Bangladesh / UNDP, 2016. Hazard and vulnerability assessment for Dhaka, Chittagong and Sylhet.

[16] A. K. Chopra, *Dynamics of Structures: Theory and Applications to Earthquake Engineering*. Pearson Education, 2012.

[17] E. Bojórquez, I. Iervolino, and G. Manfredi, "Selection of ground motion intensity measures for fragility analysis of buildings," *Earthquake Engineering & Structural Dynamics*, vol. 47, no. 4, pp. 853–872, 2018.

[18] J. P. Moehle *et al.*, "Evaluation and retrofit of existing buildings," in *12th World Conference on Earthquake Engineering*, 2000.

[19] O. D. Team, "Opensees: Open system for earthquake engineering simulation," 2020. `https://opensees.berkeley.edu`.

[20] N. Apu and U. Das, "Tectonics and earthquake potential of bangladesh: a review," *International Journal of Disaster Resilience in the Built Environment*, vol. 12, no. 3, pp. 295–307, 2021.

[21] F. Zareian and H. Krawinkler, "Evaluation of strength reduction factors for earthquake-resistant design of structures," *Earthquake Engineering & Structural Dynamics*, vol. 34, no. 14, pp. 1651–1668, 2005.

[22] J. Ghaboussi, J. H. Garrett, and X. Wu, "Knowledge-based modeling of material behavior with neural networks," *Journal of Engineering Mechanics*, vol. 117, no. 1, pp. 132–153, 1991.

[23] H. Adeli and M.-Y. Wu, "Neural network model for construction cost estimation," *Journal of Construction Engineering and Management*, vol. 124, no. 1, pp. 18–24, 1998.

[24] Pacific Earthquake Engineering Research Center, "Peer nga ground motion database," 2015. `https://ngawest2.berkeley.edu`.

[25] S. Gholizadeh, "Prediction of seismic demands of rc buildings using support vector machines," *Engineering Structures*, vol. 127, pp. 626–639, 2016.

[26] F. Khosravikia, H. R. Valipour, and A. H. Gandomi, "Seismic response prediction of steel frames using machine learning techniques," *Engineering Structures*, vol. 224, p. 111220, 2020.

[27] J. Ma, L. Zhang, and X. Lu, "Prediction of seismic fragility curves of rc frames using xgboost and random forests," *Structural Safety*, vol. 86, p. 101997, 2020.

[28] H. Sun, Y. Chen, and N. Xu, "Hybrid machine learning framework for seismic damage prediction in concrete buildings," *Soil Dynamics and Earthquake Engineering*, vol. 152, p. 106981, 2022.

[29] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[30] M. Shokr, E. Elbeltagi, and A. A. Torkey, "Deep learning-based model updating for seismic response prediction of concrete structures," *Engineering Structures*, vol. 246, p. 113054, 2021.

[31] E. Kiani and H. Ebrahimian, "Seismic performance prediction of rc frames using deep learning techniques," *Engineering Structures*, vol. 230, p. 111679, 2021.

[32] S. M. Mousavi and C. A. Langston, "Earthquake damage prediction using convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 4177–4189, 2019.

[33] A. Sadeghi, J. Ghaboussi, and A. S. Elnashai, "Physics-informed machine learning for seismic response modeling of structures," *Structural Control and Health Monitoring*,

vol. 30, no. 2, p. e3104, 2023.

[34] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” *Advances in Neural Information Processing Systems*, vol. 30, pp. 4765–4774, 2017.

[35] S. Chakraborty and S. Dey, “Explainable ai for safety assessment of civil structures,” *Automation in Construction*, vol. 124, p. 103553, 2021.

[36] M. M. Gharagoz, M. Noureldin, and J. Kim, “Explainable machine learning (xml) framework for seismic assessment of structures using extreme gradient boosting (xgboost),” *Engineering Structures*, vol. 327, p. 119621, 2025.

Appendix A

Table A.1 presents a representative sample of the dataset used for machine learning prediction of seismic drift responses. Each row corresponds to a distinct structural configuration and ground motion scenario generated through nonlinear time-history analysis. The dataset comprises a range of features, including structural parameters (e.g., period T, strength ratio SR, and retrofitting attributes), seismic intensity measures (such as PGA, PGV, PGD), and site characteristics (e.g., V_{s30}, FaultType). The target variable, `MaxDrift`, denotes the maximum inter-story drift ratio for each scenario. This comprehensive dataset underpins the development, training, and evaluation of the machine learning models discussed in this thesis.

Table A.1: Sample of Input Features and Target ($\mathtt{MaxDrift}$) from Dataset

T	SR	PGA	PGV	PGD	Mw	Rrup	AI	Vs30	LowFreq	Duration	FaultType	Ks	Mr	MaxDrift
1.311	0.19	0.302	107.62	38.01	7.03	82.21	0.344	889.6	2.257	33.82	Reverse	1200	25	0.065252
2.867	0.157	0.498	99.66	35.88	6.81	96.83	0.963	1183.2	1.939	40.33	Reverse OI	1200	13	0.072719
2.276	0.112	0.68	95.13	131.24	6.16	32.43	1.197	1126.4	1.098	71.43	Strike Slip	600	50	0.010362
1.916	0.213	0.603	72.13	36.25	7.47	76.57	0.036	363.6	0.049	29.44	Reverse	100	50	0.030819
0.721	0.187	0.644	79.91	39.49	4.89	70.31	1.077	357.8	1.873	72.66	Reverse	1200	1	0.09042
0.721	0.083	0.562	83.62	110	4.57	101.09	0.747	507.4	1.089	8.89	Normal	1200	25	0.105631
0.457	0.232	0.581	119.81	65.2	7.12	109.73	1.168	624.2	2.761	65.09	Reverse OI	1200	13	0.214154
2.639	0.215	0.667	110.52	112.5	5.7	4.33	0.558	683.8	0.114	70.86	Normal	100	37	0.036931
1.923	0.24	0.337	125.35	9.61	7.1	82.37	1.017	1025.1	2.203	16.54	Reverse	600	25	0.049769
2.212	0.195	0.469	156.68	70.68	7.25	36.82	0.075	241.3	3.532	36.82	Reverse	100	1	0.057528