Check for updates

NIST Special Publication
NIST SP 1321

A Technical Framework to Map Functional Recovery Performance Objectives to Prescriptive Seismic Design Provisions for Buildings

Dustin Cook
Siamak Sattar

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.1321

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Special Publication
NIST SP 1321

A Technical Framework to Map Functional Recovery Performance Objectives to Prescriptive Seismic Design Provisions for Buildings

Dustin Cook
Siamak Sattar
Earthquake Engineering Group
Engineering Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.1321

September 2024

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

NIST SP 1321
September 2024

Certain commercial equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

NIST Technical Series Policies
Copyright, Use, and Licensing Statements
NIST Technical Series Publication Identifier Syntax

Publication History
Approved by the NIST Editorial Review Board on 2024-07-18

How to Cite this NIST Technical Series Publication
Cook D, Sattar S (2024) A Technical Framework to Map Functional Recovery Performance Outcomes to Prescriptive Seismic Design Provisions for Buildings. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 1321. https://doi.org/10.6028/NIST.SP.1321

NIST Author ORCID iDs
Dustin Cook: 0000-0002-4841-4515
Siamak Sattar: 0000-0003-1096-7015.

Contact Information
dustin.cook@nist.gov

Abstract

Designing buildings for improved functional recovery represents a major shift in the current design paradigm. Given the novelty of the functional recovery building performance objective, current design provisions may be ineffective at controlling the underlying damage mechanisms that influence building function and recovery. Therefore, new design provisions, developed based on a fundamental understanding of building recovery behavior, are needed to meet target performance objectives for new buildings.

This report presents a technical framework to quantify minimum prescriptive design requirements that satisfy target functional recovery performance objectives for new buildings. At its core, the framework utilizes a machine learning algorithm trained on the simulated performance outcomes of a set of archetype building models to identify building characteristics that form the boundary between acceptable and unacceptable building performance. The trained algorithm serves as a decision-support tool for subject matter experts in making recommendations to update building code provisions through a consensus-based process.

In this report, the framework is exercised on a case study archetype design space to illustrate the framework's implementation and interpretation of results. The proposed framework aims to provide a technical link between probabilistic functional recovery models and practical design implementation strategies for improving community resilience and reducing disaster impacts.

Keywords

Functional Recovery, Building Codes and Standards, Seismic Design Provisions, Machine Learning, Decision Trees

Table of Contents

List of Tables

List of Figures

1. Introduction

Since the adoption of the first California earthquake codes in the early 20th century, engineers and researchers have focused on improving the collapse resistance of buildings to safeguard lives in future earthquakes. In the U.S., the building code has evolved through a consensus-based process of professional engineers, academics, and government researchers. Updates to the building code are motivated by research findings, developments in new construction techniques, and observations of building performance in earthquakes. The overall process relies heavily on the collective judgment of subject matter experts (SME) to translate the lessons learned into technical design provisions. Historically, building codes are adopted and enforced by state and local jurisdictions as minimum safety standards [1] and have a major impact on construction practice and building design.

While current building codes are intended to prevent building collapse and ensure people can evacuate safely, buildings may still sustain severe damage and be unrepairable or unusable after an earthquake. Interruptions to the normal function of buildings and the operation of critical facilities can cause cascading social disruption and economic loss [2], especially for more vulnerable community members; these impacts have been most notably demonstrated in the 1994 Northridge, 1995 Kobe, and 2010 Christchurch earthquakes.

To improve the performance of the U.S. building stock and mitigate these potentially devastating community impacts, many within the earthquake engineering field have focused on developing a new functional recovery performance objective for the design of new buildings, which explicitly targets faster building recovery after an earthquake. Expanding beyond traditional safety-based objectives, functional recovery is defined as “a post-earthquake performance state in which a building… is maintained, or restored, to safely and adequately support the basic intended functions associated with [its] pre-earthquake use or occupancy” [3].

Designing buildings for improved functional recovery represents a major shift in the current design paradigm [4]. Building function depends on many factors not explicitly considered in structural design, including damage to and the operation of many nonstructural systems, the supply of external utilities from building lifelines, and occupant-specific requirements for building use. While performance-based frameworks allow engineers to explicitly quantify building-specific performance, performance-based design requires additional analysis effort that is not typically encompassed by typical project budgets; most new buildings designed in the U.S. rely on prescriptive building-code requirements rather than performance-based requirements. Therefore, widespread adoption of functional recovery performance targets will only be realized if new prescriptive design requirements are defined.

Code-based prescriptive design requirements outline a set of deterministic procedures that implicitly provide a predefined acceptable level of reliability against earthquake-induced failure. Given the novelty of the functional recovery building performance objective, current design provisions may be ineffective at controlling the underlying damage mechanisms that influence building function and recovery. Therefore, new design provisions, developed based on a fundamental understanding of building recovery behavior, are needed to meet target performance objectives for new buildings.

This report presents a technical framework to quantify minimum prescriptive design requirements that satisfy target functional recovery performance objectives for new buildings. At its core, the framework utilizes a machine learning model, such as decision trees, trained on the simulated performance outcomes of a set of archetype building models to identify building characteristics that form the boundary between acceptable and unacceptable building performance. Essentially, the framework maps a set of archetype performance models to minimum design requirements that satisfy a target performance objective.

The objective of this framework is to act as a decision support tool for SMEs in making recommendations to update building code provisions through a consensus-based process. The collective experience established by the SME in the code development process is crucial to constrain and generalize individual research contributions applied to the field of structural design and construction; it would be impractical, and perhaps impossible, to adequately model the extent of expertise captured by the body of SMEs in the code update process. However, all building code recommendations should first and foremost be supported by data, whether observational, experimental, or analytical. Therefore, the machine learning model proposed in this framework is used as a robust and repeatable approach to explain trends to support data-informed decision-making in the code update process; the machine learning model is not intended to act as a prediction agent or surrogate model to replace SME judgment and design expertise. The proposed framework provides the technical basis for developing minimum design requirements for various building systems to meet consistent target functional recovery performance levels across each system.

Assessing and improving community recovery is a multidisciplinary and complex problem comprising technical, organizational, social, and economic aspects [5]. The proposed research aims to provide a technical link between probabilistic functional recovery models and practical design implementation strategies for improving community resilience and reducing disaster impacts. This research will help satisfy the technical research needs for improving community recovery and building performance outlined by NIST [6].

1.1. Scope of Report

This report outlines a detailed technical framework for determining prescriptive design requirements from analytical simulations of performance-based models. Additionally, the framework is exercised on a case study archetype design space to illustrate its implementation and interpretation of results. The proposed framework is intended to be used in future studies for a broader range of building archetypes beyond those presented in this report's case study.

Section 2 of this report provides additional background information on U.S. building codes and standards, recovery-based modeling, FEMA P-695, and machine learning techniques. The technical framework is presented in detail in Section 3, and the case study methods and results are presented in Section 4.

While the framework presented in this report focuses on recovery-based modeling and functional recovery building performance objectives for seismic provisions, the approach used here is general and applicable to developing design requirements for other performance objectives or hazards. This report focuses on prescriptive design requirements for new buildings following the ASCE/SEI 7 standard [7]. However, a similar approach could be followed to quantify design requirements for other standards, such as performance-based requirements for existing buildings in the ASCE/SEI 41 standard [8].

2. Background

2.1. Seismic Building Codes and Standards in the U.S.

In the U.S., seismic awareness grew in the early 20th century, spurred by events like the devastating 1906 San Francisco, 1925 Santa Barbara, and the 1933 Long Beach earthquakes. These disasters highlighted the urgent need for more robust regulations to mitigate the seismic risks to buildings and communities. The first provisions specifying minimum lateral seismic loads—up to 10 % of the building's seismic weight—were introduced with the inaugural version of the Uniform Building Code (UBC) in 1927. However, the provisions were part of a non-mandatory appendix, and adoption of the seismic provisions was sparse among local California jurisdictions. The deadly performance of unreinforced masonry schools in the 1933 Long Beach earthquake catapulted California into the mandatory era of seismic regulations with the introduction of the 1933 UBC, the Field Act, and the Riley Act [9].

The 1960s through the 1990s marked a pivotal era in the evolution of the seismic design provisions for new buildings. Following significant earthquakes in Alaska and California—most notably the 1964 Alaska, 1971 San Fernando, 1989 Loma Prieta, and 1994 Northridge Earthquakes—the UBC and other material-specific standards were updated to include more advanced analysis methods and special seismic detailing requirements, reflecting lessons learned from earthquake reconnaissance and a general growing understanding of ground motion behavior, building response, and structural member capacity.

In its current form, the International Building Code (IBC), first replacing the UBC in 2000, is a model code widely adopted by states and local jurisdictions for building design, construction, maintenance, repair, and demolition. The code provides minimum requirements to ensure buildings meet life-safety performance objectives. The code references the ASCE/SEI 7 standard for Minimum Design Loads and Associated Criteria for Buildings as the technical basis for its seismic provisions.

While current seismic provisions are intended to enhance building safety, only limited scope is provided for damage protection and continued building occupancy. Most buildings satisfy minimum seismic design loads by detailing structural components to have sufficient ductility capacity to withstand a design-level earthquake with an acceptably low probability of collapse; buildings that meet these minimum safety requirements may still sustain severe damage and be unrepairable or unusable after an earthquake. While design provisions for Risk Category IV structures and special state-specific provisions, like California's design criteria for hospitals, provide additional requirements to target immediate occupancy after an earthquake, it is unclear whether those provisions adequately ensure enhanced functional recovery performance.

The notion that building codes do not directly define requirements to limit damage and protect property in disasters runs counter to public expectations of the building code. Davis & Porter [10] surveyed around 500 adults in California and the New Madrid seismic region near Memphis, Tennessee. Results from the survey show that a significant portion of the public already believes that new buildings are designed to be occupiable and functional after earthquakes. Most surveyed indicated they would prefer occupiable and functional building performance targets, compared to the current status quo, even if the change resulted in increased construction and rental costs.

2.2. Performance-Based Earthquake Engineering

In contrast to traditional prescriptive design requirements outlined in the building code, performance-based design allows engineers to explicitly design buildings to meet a specific performance target by leveraging advanced analytical methods. The PEER performance-based framework, which was later formalized in FEMA P-58 [11], [12], introduced a four-stage process (Fig. 1) of integrating hazards, structural response, and component-level damage and consequence models into probabilistic performance assessment methodology [13], [14], [15]; the hazard analysis uses probabilistic seismic hazard analysis [16] to quantify the intensity of shaking given the seismicity of the building site; the structural analysis quantifies the building's response, in terms of engineering demand parameters (EDPs), given the intensity of shaking; the damage analysis follows a component-level assembly-based procedure [17], where the damageability of each structural and nonstructural component within the building is represented by a fragility curve, which probabilistically relates EDPs from structural analysis to discrete component damage states; the loss analysis assembles component level damage consequences into building performance metrics that are communicated to stakeholders and decision makers in terms of post-earthquake repair costs, repair times, casualties, and potential unsafe placards.

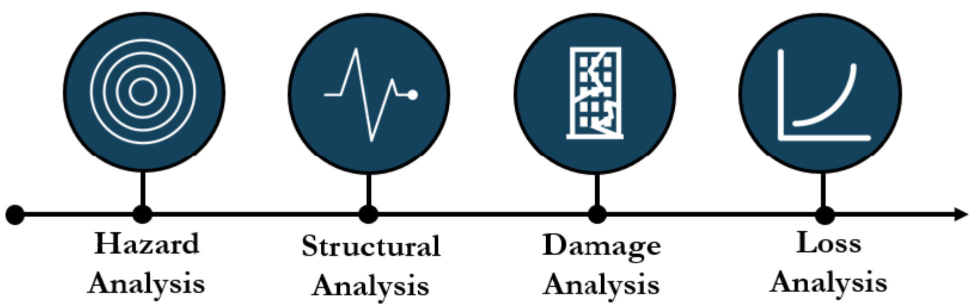

Fig. 1. PEER framework for performance-based earthquake engineering

To facilitate the component-level calculations, the FEMA P-58 project assembled data on hundreds of structural and nonstructural building components into a fragility database, including fragilities based on experimental data, empirical evidence, and expert judgment. Each component within the fragility database has associated consequence functions that quantify the repair costs, repair times, potential casualties, and likelihood of triggering an unsafe placard, given each component’s discrete damage states. Total building losses are an aggregation of all component-level losses within the building. The FEMA P-58 assessment methodology uses a Monte Carlo simulation to account for and propagate the uncertainty in each step of the process and probabilistically quantify building performance.

The critical point of departure of performance-based design from prescriptive design is the ability to set specific building performance objectives, explicitly tune any given design to meet those performance goals, and communicate design performance into meaningful decision metrics. However, flexibility in performance-based design is also a fundamental limitation. Setting performance standards in a national building code framework that adequately ensures acceptable performance without the need for design peer review is difficult. Long and uncertain peer review processes and the added cost of additional modeling are usually only economically viable for large projects in high seismic settings.

2.3. FEMA P-695

FEMA P-695 [18] is a performance assessment methodology that serves as a rational framework to determine minimum prescriptive design values (building system response parameters) for lateral force-resisting systems to satisfy the collapse performance required for adoption into the seismic provisions of the ASCE/SEI 7 standard. More specifically, the methodology assesses whether system-specific R, Cd, and Omega factors from Chapter 12 of ASCE/SEI 7 provide acceptable collapse performance for a given lateral system. At its core, the methodology identifies a process for developing a design space of archetype buildings, quantifying each archetype’s collapse potential for a given ground motion intensity, and assessing the collective archetype performance to determine whether the proposed design parameters meet the intended performance.

To facilitate the assessment and generalize the collapse performance of a given lateral system, the analyst develops a set of archetype models to represent typical variations in a lateral system. A complete archetype design space can comprise 20-30 models, representing variations in building height, seismic design category, fundamental period, or building configuration; special attention should be given to system-specific design variations that may significantly impact performance. Each archetype is organized into performance groups based on shared features.

Once the archetype design space is finalized, a set of building system response parameters (R, Cd, and Omega) are proposed, and each archetype model is designed and detailed to satisfy the minimum seismic provisions according to Chapter 12 of ASCE/SEI 7. Using nonlinear response history analysis according to the state-of-the-art modeling practices, the collapse performance of each archetype model is assessed for a suite of 44 ground motions, scaled to the risk-targeted maximum considered earthquake (MCE_R).

To assess the adequacy of the proposed building system response parameters, the collective archetype performance, in terms of collapse potential, is compared to the minimum acceptable performance required to satisfy the intent of the building code. To satisfy the criteria, the average collapse probability of all models within a performance group should be less than 10 % and no more than 20 % for any one given model. All performance groups assessed must meet the above criteria for the proposed building system response parameters to be deemed acceptable. If an archetype set, designed to a given combination of R, Cd, and Omega factors (mainly the R factor), meets the assessment criteria, the design values can be adopted into the seismic provisions of the building code (pending an SME peer review through the building provision update process). If the models do not meet the criteria, new design factors must be proposed, and the process must be repeated. FEMA P-695 remains the benchmark methodology for assessing life-safety prescriptive design values and introducing new lateral systems into the building code.

2.4. Functional Recovery

Even as the evolution of modern building codes has improved the collapse resistance of the general building stock, the potential for widespread damage in earthquakes still presents a significant risk to community recovery and economic stability; damage to buildings and critical facilities can interrupt business, displace families, and significantly disrupt economies and normal community function for years to come [2]. In the 1989 Loma Prieta earthquake, it took up to 10 years to repair damage to schools, affordable housing, and highways, resulting in the permanent closure of many buildings [19]. The population in the Kobe region dropped by 2.5 % after the earthquake and lost 10 % of its businesses, with a disproportionate impact on smaller businesses and lower-income families [20]. More recently, after 90 % of the buildings were red or yellow-tagged following the 2010 Christchurch earthquake, the central business district remained closed for over two years, causing the permanent closure of 11 % of the city's businesses [21].

Given evidence from these earthquakes, attempts to support recovery-based design objectives in building codes, standards, and practices have increased in the past few years. In a multi-agency effort, the National Institute of Standards and Technology (NIST) and the Federal Emergency Management Agency (FEMA)—two federal agency members of the National Earthquake Hazards Reduction Program (NEHRP)—published the NIST SP1254/FEMA P-2090 report (NIST 2021), which outlines options for improving the built environment and critical infrastructure to reflect performance goals stated in terms of post-earthquake re-occupancy and functional recovery time. In particular, the report recommends designing new buildings, retrofitting existing buildings, and upgrading lifeline infrastructure to meet recovery-based performance objectives, in addition to promoting pre-disaster planning, education and outreach, and access to financial resources.

For buildings, functional recovery is a post-earthquake performance state in which a building is maintained or restored to safely and adequately support the basic intended functions associated with its pre-earthquake use or occupancy, as illustrated in Fig. 2. In other words, functional recovery represents a building's capacity to maintain or rapidly restore its primary use after an earthquake, focusing mainly on the performance of the building's structure and nonstructural systems rather than explicitly representing household or community recovery. Therefore, the functional recovery performance objective represents a shift in building design towards improving post-earthquake outcomes for communities by collectively enhancing the recovery-based performance of the general building stock, all while maintaining the same rigorous life safety performance objectives.

While the general goals for functional recovery are clear, the mechanisms for implementing those goals within buildings and lifeline infrastructure are still under development [22], [23]. The framework proposed in this report aims to help translate functional recovery goals into technical design provisions that satisfy specific functional recovery performance objectives, leveraging the performance-based earthquake engineering framework.

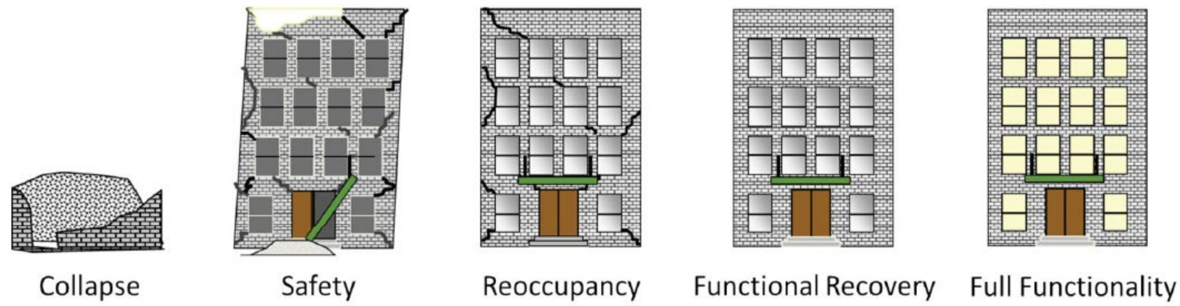

Fig. 2. Illustration of safety- and recovery-based performance states for buildings [3].

As discussed in the previous section, performance-based earthquake engineering and design presents a quantitative approach to explicitly assess building performance that reflects building-specific design characteristics. Several recent methods have emerged that extend the FEMA P-58 assessment methodology to quantify the performance of a building in terms of the post-earthquake loss of building function and time to restore the building to a desired functional state [24], [25], [26], [27]. In these methods, relationships are developed between damage to structural and nonstructural systems and building function, using either component-level damage-state classifications or through a series of hierarchical fault trees. Uncertainties in ground motion hazard, building response, component damage, component repair time, building repair schedule, and construction impeding factors are propagated through a Monte Carlo simulation to produce probabilistic performance outcomes. These performance-based methods facilitate a technical link between building-specific characteristics, such as building configuration or seismic design values, and functional recovery performance. Several methods are already used in engineering practice to design and communicate building performance for explicit recovery-based goals [28]. However, such methods often require expertise, computational bandwidth, and funding well beyond the scope of typical design projects, limiting widespread adoption in practice.

2.5. Machine Learning

Machine learning (ML) generally describes a class of algorithms that actively learn from exposure to data and can be used to develop models for pattern recognition, regression, and classification. In machine learning, data is the raw material, and the applicability of various machine learning algorithms to solve complex problems depends on the extent and quality of data available. In practice, the data used for machine learning applications can range from sensor readings, images, text, simulated outputs from analytical models, survey results, or any other type of information relevant to the problem at hand, for example, the various seismic design characteristics of a set of buildings. Depending on the type and extent of available data and the specific question to be addressed, various machine learning algorithms can be leveraged to solve a given problem, each with its own distinct advantages. This diverse landscape can be broadly grouped into three fundamental categories: supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning is the process of mapping input data, or data features, to predefined labeled outputs, or targets, for a given dataset, thereby learning patterns and relationships between various data features and target values or classifications. Supervised learning boasts several prominent techniques, such as logistic regression, which excels in binary classification problems, and decision trees, renowned for their interpretability and ability to handle complex feature interactions [29]. The ensemble method known as random forests harnesses the collective power of multiple decision trees to improve predictive accuracy [30]. Inspired by the human brain's structure, neural networks have revolutionized fields like image recognition and natural language processing [31].

In contrast, unsupervised learning takes a different route, working with unlabeled data to uncover hidden structures and patterns within the information. Unsupervised learning encompasses techniques like clustering and dimensionality reduction. K-means clustering, for example, groups similar data points together, enabling the discovery of natural clusters in the data. Principal Component Analysis (PCA) reduces the dimensionality of high-dimensional data while retaining essential information, aiding in data visualization and feature selection [32].

Reinforcement learning, on the other hand, focuses on training agents to make sequences of decisions by interacting with the environment and receiving feedback in the form of rewards or penalties. Applications of reinforcement learning include game playing (e.g., AlphaGo), robotics, and autonomous control systems.

Training a machine learning model is a multi-stage process that involves data preprocessing, feature engineering, algorithm tuning, training, and testing. Throughout the process, careful attention should be given to balancing the need for model performance, data quality, and computational efficiency while avoiding bias through model generality. Specific techniques for data normalization, feature extraction, dimensionality reduction, model regularization, and model validation and performance evaluation are well documented in the general literature [33].

While high predictive performance is a distinct advantage of many ML algorithms, model transparency is critical for domains where interpretability and accountability are paramount. Transparency in ML algorithms refers to the ability to understand and interpret the model's decision-making process. The complexity of a model directly impacts its interpretability, and there is often a trade-off between model performance and model transparency. Many algorithms can help explain trends, extract feature importance, and describe decision rules for complex ML models such as deep neural networks. However, explainable AI (XAI) can only go so far in making complex ML models interpretable by humans. Alternatively, simpler algorithms, such as decision trees, are naturally transparent in how they make decisions and formulate predictions and, therefore, may be preferred for situations where machine learning algorithms are being used as decision support tools for SMEs.

Machine learning has evolved from a specialized field within computer science to a widely adopted set of tools that help solve complex problems and support data-driven decision-making across countless sectors of industry, academia, and public policy. Machine learning packages are widely available in many science and engineering programming tools such as Python or MATLAB and represent an easily accessible approach for synthesizing trends from a large dataset, especially for highly dimensional, nonlinear problems.

Though data availability and privacy issues have plagued the widespread application of ML to many structural engineering and natural hazards problems, the use of ML in these fields dates back to the 1980s. In early applications, structural engineering researchers demonstrated using ML models to help solve classic mechanics-based design problems [34]. More recently, with the widespread expansion of data collection tools and computational resources, the application of ML in structural engineering and natural hazards research and practice has grown to cover topics such as structural response prediction, surrogate modeling, design optimization, hazard forecasting, synthesizing recordings from experimental tests into analytical models, structural health monitoring and predictive maintenance, data collection and classification from disaster reconnaissance, and fragility model development for buildings, dams, and other infrastructure [35], [36], [37], [38], [39], [40].

2.5.1. Decision Trees

Decision trees are a powerful class of supervised learning algorithms that identify rules for splitting labeled data into hierarchical tree structures. Decision trees partition data using principles of information theory to sort through data features and arrive at an end decision point, or leaf node, associated with a specific prediction or outcome. All nodes along a branch that lead to a particular prediction outcome outline a set criterion required to satisfy the given output based on the features used to train the model.

Decision trees are constructed through recursive binary splitting of the input space (feature values). The process starts with the root node, representing the first split for the entire training data set. Each internal decision node selects an additional splitting criterion to divide the data further; the end of a particular branch is referred to as a leaf node, where the data is classified into one dominant class (Fig. 3). At each node, the algorithm will select the split that results in the highest information gain (or Gini Index); the information gained by a given split is typically quantified as the difference in entropy (or Gini Impurity) between the parent and child nodes.

Decision trees are not as high performing of prediction algorithms relative to other machine learning techniques, such as deep neural networks or random forests, and therefore are typically not the first choice for surrogate models or where high prediction accuracy is required. However, decision trees are unique in that they are completely transparent; it is easy for a human to determine precisely what decisions are being made by the model and why. This makes decision tree models an excellent decision support tool. For this reason, the proposed framework uses a decision tree algorithm as the mechanism by which to identify trends in design characteristics that lead to acceptable building performance.

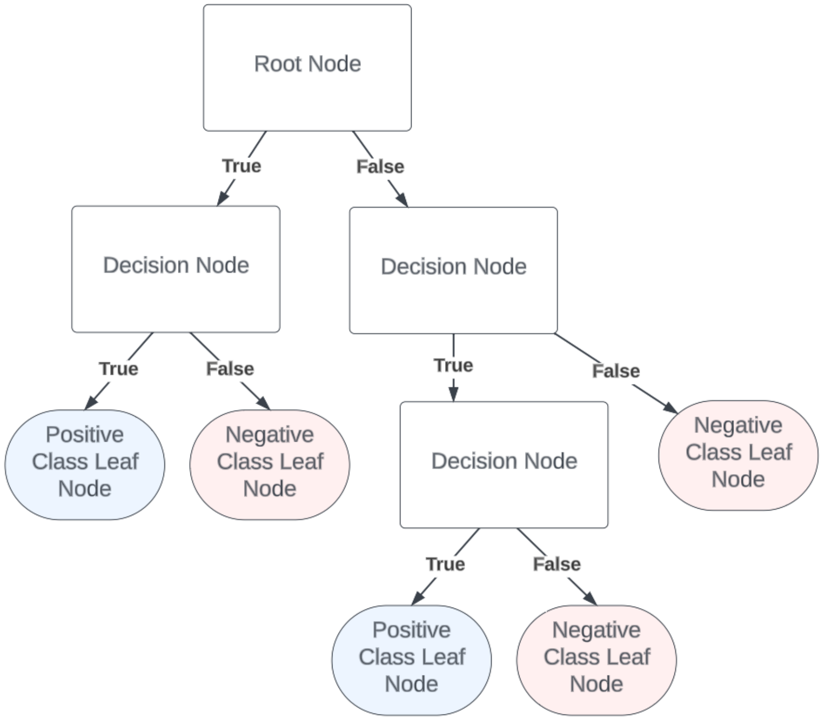

Fig. 3. Illustration of a basic decision tree structure.

3. Framework

3.1. Framework Overview

The proposed framework presents a quantitative methodology to determine prescriptive design values required to achieve target performance objectives for a class of buildings. While the overall framework presented in this report generally applies to other hazards and building performance objectives, we focus on applications to the functional recovery performance of buildings in earthquakes. More specifically, the framework is used to develop force- and deformation-based minimum design requirements for structural and nonstructural building components and systems to be adopted into the seismic design provisions of the building code.

In its fundamental objective and architecture, the framework follows a similar path to that of the FEMA P-695 assessment methodology, which is used to develop prescriptive design requirements—namely the R-factor, Cd, and Omega—for collapse prevention performance objectives in the building code. In both approaches, a set of archetype building models are assembled to represent the design and performance attributes of a particular building class or lateral system, the performance of each archetype is simulated using a performance-based approach, and the performance outcomes are compared to a target performance objective to assess the acceptability of a given set of prescriptive design values. The framework proposed in this approach deviates from the FEMA P-695 methodology in two key ways: (1) we are interested in design parameters that improve building recovery rather than collapse and use a modeling framework to probabilistically quantify functional recovery time, considering damage to structural and nonstructural building components alike, rather than just collapse potential; (2) we use a decision tree algorithm, trained to the simulated archetype performance outcomes, to identify relationships between prescriptive design parameters and acceptable performance levels. In leveraging the decision tree model, the proposed framework presents a robust and repeatable process to identify the most influential and informative design parameters in a highly dimensional design space and translate trends usually only captured by building-specific performance models into prescriptive-based requirements.

Fig. 4 presents a diagram illustrating the architecture of the proposed framework. To facilitate the quantification of functional recovery times among the various archetype models, we use the performance-based assessment methodology presented in Cook et al. [25], as implemented in ATC-138 [41], which extends the FEMA P-58 methodology to probabilistically simulate building performance in terms of re-occupancy and functional recovery time across various ground motion intensities.

The feedback loop between SMEs and the decision tree model is crucial to the framework's application in future building codes. Recognizing that not all aspects of building performance, practical construction constraints, economics, and design optimization can adequately be captured in the building performance model, the SME remains a pivotal gatekeeper to translate model outcomes into practical and effective seismic provisions. Therefore, a heuristic regularization cycle is placed within the proposed framework to integrate and learn from SME feedback and constraints.

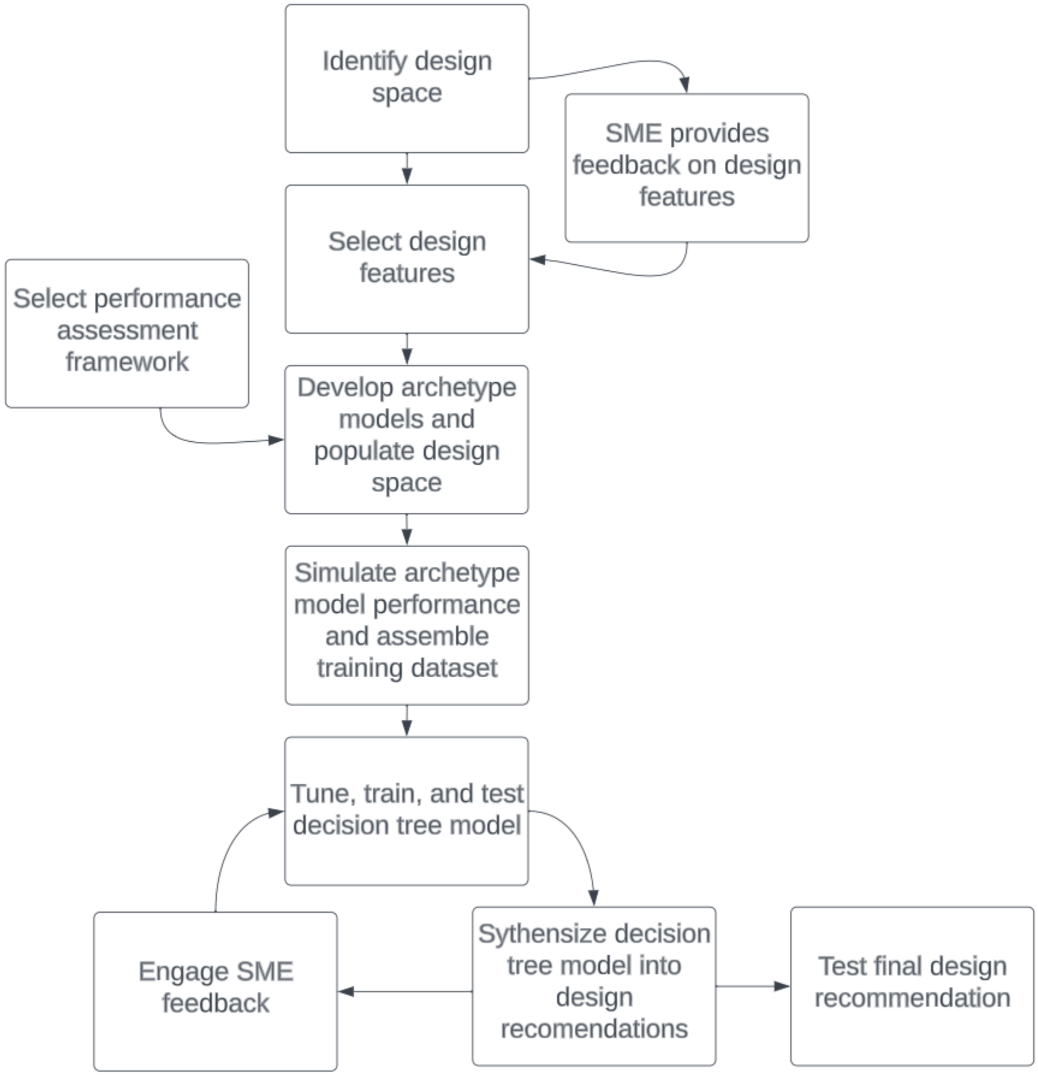

Fig. 4. Workflow of the proposed framework.

Each step of the framework is discussed in detail in the following sections: Sec. 3.2 on defining the design parameters of interest and archetype design space, Sec. 3.3 on developing a set of archetype models to represent the design space, Sec. 3.4 on simulating the recovery performance of each model, and Sec. 3.5 on training, testing, and interpreting the decision tree model to develop minimum design requirements for a given recovery-based acceptance criteria.

3.2. Design Space Identification

The first step of the process is identifying the design space of interest. In this process, the analyst will identify building characteristics, such as seismic design parameters or other architectural characteristics, that may substantially impact the building's target performance. All building characteristics identified here are used to populate a large design space with archetype models; the number of building characteristics explored defines the number of dimensions within the design space. While seemingly trivial, this step is critical as it establishes the scope and parameters for the problem the analyst is attempting to solve.

Of the building characteristics of interest, design features are the characteristics directly used to train the decision tree model. Therefore, the design features define the scope of the design parameters available to the framework to make final design recommendations. Each design feature used within the framework should meet the following criteria:

1. The design features should influence the performance target of interest.
2. The impact of the design features should be captured in the archetype model and adequately reflected in its simulated performance.
3. The design feature should be within the purview of the building code and design engineer.

In the first criterion, it will likely be challenging to know ahead of time exactly which design parameters impact target performance. Therefore, the analyst should identify many design parameters that may be of interest, using feedback from SMEs to guide their selection. In this process, it is typically better to select more design features than is necessary, as the decision tree model will automatically prioritize design features that are most informative and remove features that do not influence performance substantially. However, large design feature sets can exponentially grow the design space, making the problem computationally intractable. Section 3.3 below discusses several strategies for populating large design spaces in a scalable fashion.

Some design features of interest will not satisfy criteria 2 and 3 above and, therefore, cannot be directly incorporated into this framework. For example, disaster preparedness planning might substantially impact recovery, but the recovery simulation model may not adequately capture its impact; similarly, the building code may not be capable of regulating it. Solutions to these performance impacts require either judgment-based stop-gap measures by SMEs or further study before action can be taken.

In the FEMA P-695 methodology, prescriptive design factors R, Cd, and Omega are selected by an SME as the critical design parameters of interest before the method's development. These design parameters are the primary structure-level requirements the building code uses to establish minimum design loads to control collapse risk among various structural systems. Additionally, other building characteristics, such as seismic design category, building height, and structural configuration, populate a design space of archetypes to identify any significant variations in collapse performance among typical design cases.

For recovery-based modeling, many additional factors beyond the collapse reliability can influence a building’s functional recovery performance; some example factors include the distribution of drifts and accelerations on each story, the repairability of damaged structural components, the continued operation of nonstructural components and systems, backup systems and equipment, or selective additional quality control and assurance in the design process. Before the start of the proposed assessment, the analyst should engage with SMEs to identify critical design features and building characteristics of interest. This will help expedite the later SME feedback loop to refine the trained decision tree model into design recommendations.

3.2.1. Feature Selection

Feature selection is the process of reducing model dimensionality by either trimming features that have little impact on the performance target or extracting new features that have a superior correlation with the performance target. Feature selection is critical in training a machine learning model by reducing model dimensionality, improving computation efficiency, improving model interpretability (simplicity), and improving model generality (reducing overfitting).

Feature selection is typically employed in several ways:

1. Filter methods are employed prior to training the model to identify features that exhibit strong correlations or dependencies with the target variable. Examples of filter methods include correlation or the chi-square test.
2. Wrapper methods are employed during model tuning to identify the simplest subset of features that lead to models with acceptable model performance.
3. Embedded methods are directly integrated into the training of the machine learning model, such as in Lasso or Ridge regression (L1 and L2 regularization).

On the other hand, feature extraction is a process by which new features that show superior correlation to a given model target are created from a subset of existing features. Feature extraction can be performed manually by defining new features as some linear or nonlinear combination of existing features based on domain-specific knowledge (typically done for regression analysis of continuous variables to identify nonlinear relationships) or through algorithmic means such as principal component analysis (PCA). For example, maximum story drift demands do not always relate the best to building damage and recovery time [42]; one could use PCA to extract new EDP features from feature vectors of peak story drift (SDR) and peak floor acceleration (PFA) when performing a regression analysis to relate peak structural response with recovery time [43].

In the proposed framework, we employ two primary feature selection approaches: (1) a wrapper approach that automatically trims features with low influence by adding fitting constraints to regularize the decision tree within the tuning process and (2) manual, post-training heuristic regularization employed through a feedback loop with SME; the heuristic regularization process is discussed in detail in Sec. 3.6. We do not employ any feature extraction within the proposed framework. Though feature extraction is effective at boosting model performance, extracted features may be less intuitive or interpretable for the end user, especially when using complex algorithms such as deep neural networks.

Additionally, before training, some pre-performance simulation analysis can be performed to identify the influential domain space of each design feature. While optional, identifying the influential domain space of each design feature can substantially reduce the computational cost of simulating the performance of each archetype model within the design space. As suggested by Issa et al. [37], the initial domain of a design feature should first be bounded based on data or judgment that reflects the physical range of the feature that is possible or practical; however, an additional sensitivity assessment can be performed to identify the domain over which each design feature influences the performance target of interest. Here, an inverse one-at-a-time (OAT) sensitivity analysis can be performed on a subset of archetype models, where all design features are set to their maximum value (the value that leads to the smallest recovery time), then each design feature is reduced independently, over a proposed domain to identify the lower limit value; Design feature values below the lower limit always cause unacceptable performance and therefore are unnecessary to assess in the performance simulation of the entire design space. Additionally, the sensitivity study should be checked to ensure that acceptable performance is reached when all design features are set to their maximum value.

The general procedure for feature selection within this framework is as follows:

1. Identify important features and feature domains based on feedback from SMEs
2. Perform a feature-specific sensitivity analysis to verify the target acceptance criteria are within the feature design space.
3. Constrain the model during model tuning (Sec. 3.5) to automatically select only the most influential features that lead to high model performance (wrapper technique).
4. Engage in a feedback loop within SMEs to perform heuristic regularization on model features, which helps to simplify the model into design recommendations for adoption into the building code update process.

3.3. Archetype Development

For each design feature and building characteristic within the identified design space, an archetype model should be created that adequately represents its design and response characteristics. Archetype models should be developed to represent many different values over the domain of each design feature and building characteristics of interest at adequately discretized increments to capture important changes in performance. These models should follow best modeling practices according to the assessment methodology used to simulate performance.

In the FEMA P-695 assessment methodology, archetype models are categorized into performance groups based on the common characteristics among various archetypes. The performance grouping is used to check the acceptability of a set of design features; the average collapse probability of all models in a performance group should be less than 10 %, and no one model should be greater than 20 %. Performance groups are usually separated by characteristics such as framing layout and seismic design category, while the number of stories varies within a performance group; otherwise, exactly what defines a performance group is a bit subjective.

In the proposed framework, the performance group concept is not required. All archetype models within the design space are simulated using the same assessment methodology, and a decision tree model is trained on the performance outcomes and design features of all archetype models. The influence of each design feature on archetype performance is automatically captured within the decision tree to develop a hierarchical structure that naturally separates the archetypes into two classification groups: models that meet the target performance objective and those that do not.

For recovery-based modeling, each archetype should be represented by a performance model, similar to what is defined by FEMA P-58 [12]. A building performance model consists of an analytical structural model capable of capturing peak structural responses (EDPs) for a given ground motion intensity and a component fragility and consequence model that captures component damage given those structural responses. For typical applications, this would involve (1) creating an explicit structural design that conforms to current design standards and represents the selected design features, (2) constructing a mathematical model of structural response to capture peak EDPs, and (3) populating the building performance model with structural and nonstructural components according to typical construction. For the case study in Sec. 4, we model each archetype using a simplified elastic representation of a building to provide an illustrative example of the end-to-end process; nonlinear building models should be used for the actual application of the proposed framework.

3.3.1. Design of Computer Experiments

For hyperdimensional design spaces, i.e., where there are many design features and building characteristics of interest, the number of archetype models required to represent each discrete combination of design features grows exponentially. For example, a design space with eight design features of interest, each with ten discrete values along the design feature domain, results in 10^8 unique combinations of design features within the design space (i.e., using a grid search).

In the proposed framework, the design space represents the scope of the problem attempting to be solved; the analyst seeks to develop relationships between design features and archetype performance, but only for buildings within the limits of the design space. To provide sufficient input data for the decision tree, archetype models should be dispersed throughout all corners of the design space, but perfect distribution is not required. The decision tree model only needs enough data points to extract trends and separate poor performers. Therefore, other space-filling techniques, such as Monte Carlo Simulation (MCS) or Latin Hypercube Simulation (LHS), can provide much more efficient ways to populate the design space. Still, each has limitations based on the specific nuances of the design space of interest.

To take advantage of efficient simulation methods while simultaneously circumventing their limitations, the framework adopts a hybrid Design of Computer Experiments (DoCE) approach [44] to develop a sample protocol based on random, pseudo-random, and hand-selected protocols. This approach provides a computationally efficient means of exploring a wide range of conditions, leveraging domain-specific knowledge to fill the design space. While the exact DoCE is specific to the given design space and performance objective of interest, an analyst can leverage the following types of sampling protocols to develop an efficient sampling design:

- Hand-selected protocols: usually using a uniform scaling protocol, where all features are scaled simultaneously and uniformly, or a one-at-a-time (OAT) protocol, where only one feature is varied while other features are held constant. Hand-selected protocols usually require a domain-specific knowledge of the performance assessment methodology to help identify regions of the design space where performance outcomes may be more sensitive to variations in design features (and may not be well captured by random protocols).
- Random protocols: usually a Monte Carlo or Latin Hypercube simulation.
- Pseudo-random protocols: a combination of hand-selected protocols with some randomness simulated within the selection process.

Section 4 details the exact DoCE protocols used in the illustrative case study.

3.4. Performance Simulation

Once the design space has been established, the performance of each archetype model needs to be simulated to quantify the performance outcomes used to train the decision tree model. As previously mentioned, the performance simulation method should be capable of capturing variations in building performance stemming from the various design features and building characteristics of each archetype model within the design space. When quantifying functional recovery time, the building performance model should capture how variations in structural design impact building response and structural damage, as well as how variations in nonstructural design impact the continuous operation of critical building systems. Therefore, to quantify the functional recovery time of each archetype model, we use the performance-based functional recovery method outlined by Cook et al. [25], as implemented in ATC-138 [41]. This assessment methodology extends the FEMA P-58 performance-based method to consider consequences, such as re-occupancy and functional recovery time, in a probabilistic analytical assessment framework; other recently developed methods could be used as well [24], [26], [27].

At its core, the FEMA P-58-based performance assessment methodology requires two primary inputs: (1) a mathematical model of the structural lateral system to capture peak structural responses in terms of story drifts and floor accelerations and (2) an assembly of component fragility models to capture the damage fragility and damage consequences of each critical structural and nonstructural component in the building; tenant contents are typically not considered when modeling building functional recovery time. However, beyond these two core inputs, there are many other inputs that make up the building performance model. For example, in the Cook et al. method, the analyst must also model the distribution of tenant units based on the building's occupancy and any impeding factors that delay the start of repairs. While the methodology provides recommended default values for most assessment inputs, the analyst should carefully document each assumption made and clearly communicate them to the end user of the decision tree model. If additional modeling assumptions are deemed important for functional recovery and are within the purview of the building code, they should be explicitly modeled as design features of interest and captured as variations in the archetype design space.

When incorporating design features into the building performance model, some expert judgment may be required on the part of the analyst as to exactly how variations in a given design feature are reflected in the building performance model. For example, it may be impractical to improve the acceleration capacity of a particular nonstructural component in a linear-continuous fashion due to traditional construction constraints. One option could be to directly model the various construction constraints that make such improvements impractical. However, it is difficult to realistically capture these constraints. Instead, an analyst may ignore these various construction constraints and model the nonstructural improvements as theoretical scalars on component capacity, thereby modeling the minimum increase in capacity required to achieve acceptable performance instead of directly modeling explicit design requirements. In other words, define the minimum force- and displacement-based prescriptive capacity required to achieve acceptable performance, leaving the detailing required to achieve said capacity up to the engineer and material standards. In training the decision tree model, an SME can identify any impractical enhancements, constrain the inputs, and retrain the model to derive a more practical set of recommendations during the feedback loop. To aid the SME in interpreting decision tree outcomes, the analyst should carefully document and communicate all assumptions used when modeling various design features in the building performance model.

The performance of each archetype within the design space should be simulated using the same performance assessment framework and default assumptions, where appropriate. Once the performance of all archetype models is simulated, the performance outcomes are collected into a database alongside design features to be used for training the decision tree model. Given the probabilistic nature of the performance assessment methodology, the extracted performance outcomes from each archetype model should reflect the target reliability level of the performance objective. For example, suppose the target performance objective is that there should be less than a 10 % chance of exceeding 30 days of functional recovery time. In that case, the 90th percentile functional recovery time should be collected from each archetype model. Additional details on the performance simulation assumptions made in the illustrative case study example are provided in Sec. 4.

3.5. Decision Tree Modeling

Once the database of performance targets and design features is assembled, the decision tree model is ready to be trained. Fundamentally, the decision tree model serves as a platform to synthesize trends in performance outcomes from a highly dimensional and often nonlinear design space in a robust and repeatable fashion. The goal of the decision tree model is not to serve as a prediction tool to replace SMEs in the code update process but instead to utilize the decision tree hierarchy as a decision support tool to highlight critical trends and aid in data-informed decision-making. For many problems, training a machine learning model is a very straightforward process, with extensive computational resources available to automate much of the training process. Decision trees, in particular, are both straightforward to train and intrinsically transparent and, therefore, easy for end users to interpret and verify.

The process for training an interpretable decision tree within the proposed framework is outlined below:

1. Identify decision tree performance metrics.
2. Tune the model hyperparameters.
3. Train the model on the training data subset.
4. Test the model on the testing data subset.
5. Interpret and synthesize the decision tree model into design recommendations.
6. SME feedback loop and final model testing.

3.5.1. Model Performance Metrics

When training a machine learning model, there are many possible ways to quantify the predictive performance of a given model. Model performance metrics such as accuracy, precision, recall, specificity, and the F1-score are commonly used to test the performance of a decision tree model, tune the model hyperparameters for superior performance, and assess whether the model sufficiently meets its intended use. Each metric quantifies some number of desired outcomes over some normalizing factor. For example, accuracy quantifies the total number of correct predictions over the total number of predictions made; precision quantifies the ratio of predicted positives that were correct; recall quantifies the ratio of actual positives that were predicted correctly. Each metric can be sensitive to different forms of bias; therefore, depending on the application of the specific problem at hand, there is no best universal metric.

For the application of developing prescriptive code minimum requirements, we want to avoid recommendations that lead to under-conservative outcomes (i.e., these are minimum requirements, by definition); in other words, we are more concerned about misclassifying buildings as acceptable when they don't actually have acceptable performance, opposed to misclassifying buildings as unacceptable when they actually have acceptable performance. Therefore, precision is used as the primary decision tree performance metric in the proposed framework. Training a model for high precision, or the number of buildings that were correctly predicted as acceptable over the total number of buildings that were predicted as acceptable in general (whether correctly or incorrectly), biases the model towards conservative recommendations, meaning that if a model meets these recommendations, it is unlikely that it will actually fail the performance criteria. Additionally, we also use the F1-score as a secondary tuning and testing metric. The F1-score is the harmonic mean of precision and recall and helps to balance some of the natural tradeoffs between high precision and high recall, and therefore, helps us not get too conservative when tuning the model.

3.5.2. Tuning

Machine learning models, including decision trees, all have hyperparameters that control the way the model learns and help to balance the model's tendencies to overfit or underfit the training data. For decision trees, these hyperparameters include the splitting metric (e.g., Gini Impurity vs. Entropy), the maximum tree depth (maximum number of splits), the minimum number of observations required to formulate a leaf node, the minimum information gain required to justify a split, and the maximum number of features available for training.

Tuning hyperparameters essentially boils down to a sensitivity study to find the combination of hyperparameters that leads to the best balance between decision tree performance and model simplicity for the problem at hand. Techniques such as grid search, random search, or Bayesian optimization are commonly employed to facilitate the assessment of various hyperparameters. In addition, K-fold cross-validation is often used to prevent biasing the model to one particular partition of training and testing data. In K-fold cross-validation, the data is split into K number of bins, and a decision tree model is trained on K-1 bins and tested on the remaining bin (e.g., in 5-fold cross-validation, the data is split into five bins, four are used for training, and one for testing). The model is then retrained across all subsequent split combinations (e.g., the bin that was previously used for testing is now used for training, and a new bin is used for testing), and the vector of model performance metric outcomes is aggregated to summarize the results (typically averaged).

For developing prescriptive design recommendations from recovery-based building performance models, we recommend tuning the following hyperparameters as they had the most significant impact on model performance:

- The maximum tree depth.
- The minimum leaf size.
- The maximum number of features.

Decision tree hyperparameters are usually very straightforward to define in most decision tree software packages. The search domain of each hyperparameter should be established by the analyst, and the decision tree should be trained and tested for each hyperparameter combination (depending on the search technique) and cross-validation fold. Decision tree performance is measured based on the precision and F1-score, as previously stated, averaged across all cross-validation folds for a given hyperparameter combination.

The best set of hyperparameters, which balance performance and model simplicity, can either be selected manually or algorithmically. For consistency between decision tree models and future assessments, we recommend establishing an algorithmic set of rules for selecting hyperparameters. For the prescriptive design framework established here, the following process is outlined to select hyperparameters:

1. Set a minimum primary performance metric criterion, e.g., precision is greater than 0.9, and only select hyperparameters above that minimum performance.

2. For the remaining hyperparameter combinations, establish a tolerance for the secondary performance metric, e.g., the F1-score within 0.01 of the max score, and only select hyperparameters that are within that set tolerance of the highest scoring secondary performance metric.
3. Of the remaining hyperparameter combinations, select the hyperparameters that lead to the simplest (most constrained) model. “Simple” is not always easily defined in a multi-dimensional hyperparameter space; Therefore, establish a set priority between all hyperparameters. Based on the recommendations above, we first prioritize simplicity due to the minimum leaf size, then the maximum tree depth, then the maximum number of features.

The best set of hyperparameters is not necessarily the one that leads to the highest primary performance metrics; often, there are other hyperparameter combinations that result in negligible difference in model performance but lead to much simpler and more efficient decision tree models.

3.5.3. Training

Training a decision tree is a relatively straightforward process. The construction of a decision tree involves recursive binary splitting of the design space based on the design features and performance targets. The process starts with the root node, which represents the entire training data set. At each internal node, the data is divided into two or more child nodes based on the splitting criteria (e.g., Gini Index or Information Gain). Each split of the decision tree aims to select the design feature and feature criteria that lead to the highest level of separation (information gained) in the target performance outcomes. The tree continues to grow until a given branch reaches homogeneity of a particular prediction class (pure leaf node) or the growth is constrained by one of the predefined hyperparameters. Most scientific programming platforms (e.g., Python, MATLAB, R) have readily available packages that allow analysts to easily tune and train decision trees to a given dataset.

Prior to training a decision tree on the simulated recovery models, the archetype model performance outcomes should be classified into two bins: acceptable or unacceptable performance—a decision tree is fundamentally a classification algorithm, making predictions on binary or multi-class classifications rather than continuous random variables as in a regression algorithm. The performance should be classified based on the performance objective of interest, e.g., a 10 % chance of exceeding 30 days of functional recovery time; if the 90th percentile functional recovery time of a given model is less than 30 days, it is classified as acceptable performance.

The database of design features and archetype performance outcomes (target classifications) is then split into training and testing data. We used an 80/20 split of training and testing data; other applications may warrant a different data split. For the purpose of code provision decision support, we do not recommend using k-fold cross-validation for the final model training. The goal of the proposed framework is to use the hierarchy and splits of the decision tree itself, rather than its direct predictions, to inform minimum code requirements. K-fold cross-validation will result in a k-number of trees, each with a potentially slightly different tree structure, making the resulting tree more difficult to interpret. Instead, one tree should be trained on 80 % of the data, selected randomly, and tested on the remaining 20 %. In the final testing of the model, after the SME feedback loop is complete, a k-fold cross-validation sensitivity assessment could be performed to ensure the performance of the final model does not depend significantly on the initial training split. The decision tree should be trained using the best hyperparameter set identified in the tuning process.

3.5.4. Testing

Once the model is successfully trained, the remaining testing data partition is used to quantify the decision tree's performance. Testing the decision tree on a separate set of data helps to ensure the model maintains generality in its predictions and is not overfit to a select set of training data. During the testing process, each observation (row) of design features within the testing dataset is fed into the decision tree model to classify whether the given set of design features results in acceptable performance. The predicted classifications are then compared to the actual classifications to assess model performance based on the metric of interest, e.g., precision and F1-score. The model performance should be similar to the performance of the selected best hyperparameter set in model tuning.

3.5.5. Interpretation

The resulting outcome of the decision tree training process is a hierarchical set of minimum design requirements selected from the various design features provided, which, if followed, result in acceptable target performance for a given class of buildings. These design requirements come from the interpretation of the decision tree itself rather than relying on the decision tree model as a predictive tool. Essentially, the decision tree is a tool that systematically synthesizes trends in the underlying relationship between design features and simulated recovery performance and organizes them into a hierarchical tree structure to define the various combinations of design features and feature values that lead to acceptable performance.

To illustrate how the decision tree is used to inform design requirements, Fig. 5 shows an example decision tree. In the decision tree, the root node, internal nodes, and leaf nodes are all connected by branches. Each leaf node classifies the branch into one of two classes: acceptable (pass) or unacceptable performance (fail), based on the various design features, namely, design drift, seismic importance factor (Ie), and the maximum number of stories. All the nodes along the branch define the various conditions, in terms of minimum values of a given design feature, required to end up in a particular leaf node. By taking all of the leaf nodes that result in acceptable performance (a total of five passing leaves in Fig. 5), we can reorganize their branches into a set of rules, as shown in Table 1. These rules now formulate first-pass prescriptive design values that are required to meet the specified target performance objective. The decision tree and design recommendations in Fig. 5 and Table 1 are purely illustrative and are not based on outcomes from performance simulations.

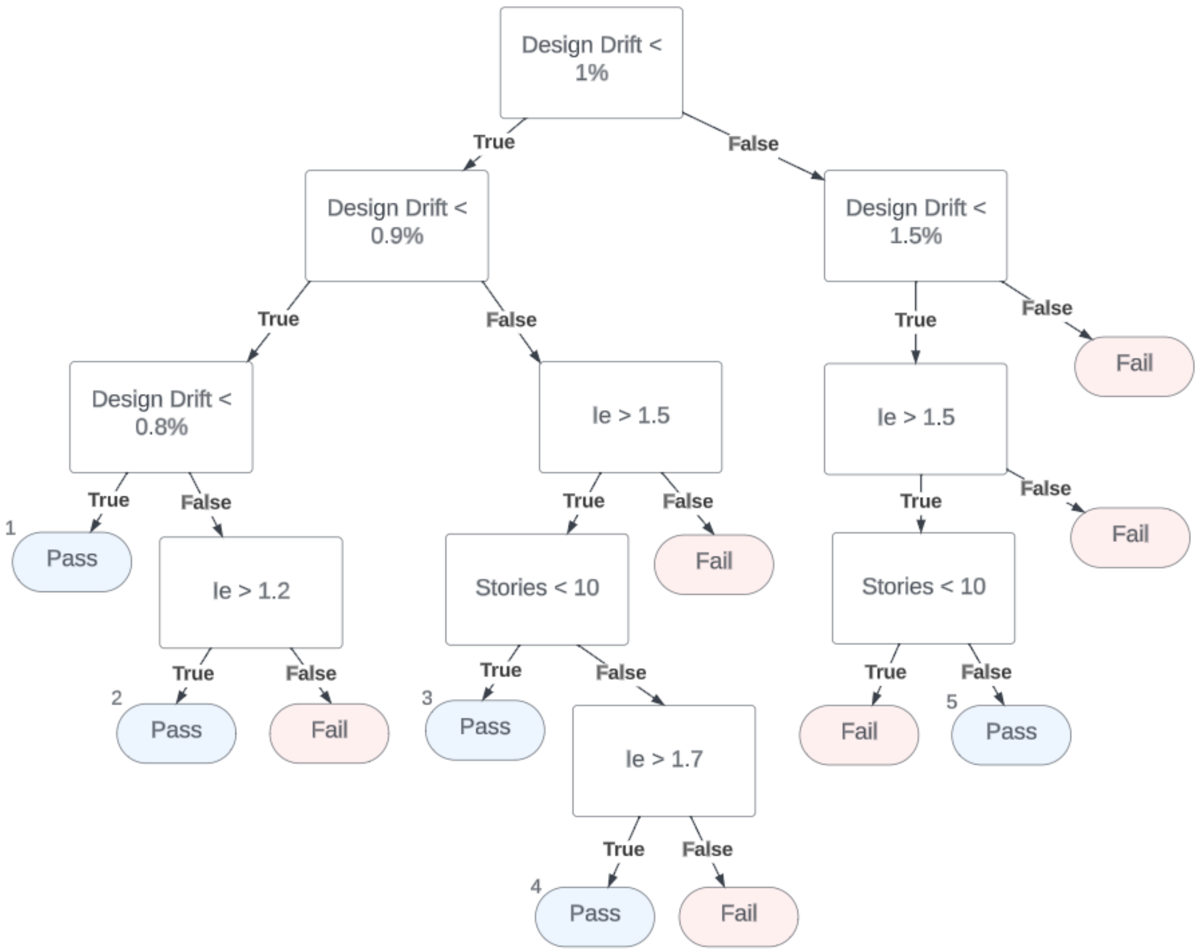

Fig. 5. Example decision tree model.

Typically, there are some common-sense modifications the analyst can make to the list of rules to simplify the passing leaves into a more compact form. For example, leaves 3 and 5 are redundant; any model passing leaf three will automatically pass leaf 5. Therefore, leaf three can be absorbed into leaf 5. Additionally, the difference in required design drift between leaves 1 and 2 is negligible and will likely round to 0.75 % (pending SME feedback). Therefore, leaf 2 can be absorbed into leaf 1. The resulting simplified set of example design criteria is shown in Table 2. To improve interpretability, analysts should avoid overly deep trees, which provide additional complexity without adding much value.

As long as the decision tree model performance metrics, quantified as part of the model testing process, are sufficiently high, e.g., model precision is greater than 0.9, and F1-score is greater than 0.75, the design requirements derived from the decision tree present a reliable set of recommendations to achieve target functional recovery performance. To compare the recommended design requirements among various performance metrics, the analyst can simply retrain a new decision tree model using the same simulated data but for a new performance metric, e.g., a 75 % chance of exceeding 30 days of functional recovery instead of 90 %; examples of this comparison are provided in Sec. 4.5.

Table 1. Example decision tree model transformed into a tabular format for all passing leaves.

Leaf	Branch Criteria
1	Design Drift < 0.8 %
2	Design Drift < 0.9 % and Ie > 1.2
3	Design Drift < 1 %, Ie > 1.5, and Number of Stories < 10
4	Design Drift < 1 %, Ie > 1.7
5	Design Drift < 1.5 %, Ie > 1.5, and Number of Stories < 10

Care should be taken not to apply the design recommendations beyond their intended scope. Similar to the FEMA P-695 assessment methodology, the proposed framework quantifies performance for a select design space of archetype buildings; the recommendations from the decision tree model should not be applied to buildings that are outside the scope of the design space. Additionally, the functional recovery performance of each archetype is simulated based on a select performance-based assessment methodology; recommendations from the decision tree model are therefore dependent on the modeling assumptions used to simulate performance and should not be applied to cases that are outside the scope of the performance assessment methodology. SME feedback is critical to shaping the scope and applicability of the results from any decision tree model.

Table 2. Building design criteria required to meet acceptable performance from example decision tree model.

Design Check	Passing Criteria
1	Design Drift < 0.75 %
2	For buildings less than ten stories: Design Drift < 1.5 %, and Ie > 1.5
3	For buildings ten stories or more: Design Drift < 1 %, and Ie > 1.7

3.5.6. Subject Matter Expert Feedback Loop

Given the limited capabilities of analytical models to capture each and every aspect of the design and construction process, engineering experience must remain the gatekeeper in the code update process. The goal of this framework is to act as a decision-support tool and provide a robust process in which to systematically synthesize outcomes and trends from a large set of simulated data. Therefore, a key part of the proposed framework is the interactive SME feedback loop, which further constraints, retrains, and retests the model based on engineering experience. This process evolves the recommendations from a theoretical set of design values to minimum design requirements that are directly applicable to the design process and adoption into the design standard.

The first step of the SME feedback loop is to review the design space, performance simulation assumptions, model training process, and model recommendations with a group of established

SMEs. It is helpful if the SME are already familiar with the design space, design features selected, and performance simulation process. After the review, SMEs should critique and constrain the decision tree model in the following ways:

- Removing design features that have little impact on the results and, therefore, overly complicate final recommendations.
- Rounding minimum design values to their closest practical implementations—it may be desirable to round towards a conservative recommendation given the scope of the building code.
- Combining branches or rules that are similar into one general or simplified rule.
- Restructuring the tree's hierarchy. For example, to modify recommendations to fit within the current structure of the seismic provision, it may be useful for SMEs to constrain the structure of the recommendations in a particular hierarchy (e.g., force the root split of the decision tree to be the building's structural system—this could be done by training multiple trees, one for each structural system).

At this point, the SME should also think about the scope and applicability of the simulated data and whether or not additional studies should be assessed to capture key limitations. Ideally, the SME should not be providing additional design features to add to the framework, as the SME should have already been engaged in the discussion when the initial set of design features was proposed.

Once the SMEs have proposed the additional practical constraints, the analyst should update their testing and training dataset accordingly and retrain and retest the model. We call this process of constraining, retraining, and retesting the model heuristic regularization, relying on SME experience to help generalize outcomes from the analytical model and overcome limitations. This heuristic regularization process should be repeated with the SME group until a consensus is reached. After the final model is developed, the decision tree performance metrics should be retested on the original dataset using k-fold cross-validation. Model performance should remain acceptably high across all training bins.

4. Illustrative Case Study

To illustrate the process of selecting design features, developing an archetype design space, simulating archetype performance, training a decision tree model, and interpreting results, this section exercises the proposed framework on a simplified case study set of archetypes. The archetype set developed for the purpose of this case study uses elastic multi-degree-of-freedom (MDOF) models over a range of various response amplitudes and response shapes to investigate recovery performance across a general range of response behaviors. Beyond this, no explicit structural designs were created for this case study. Therefore, the results of the case study are meant to illustrate the proposed framework and do not serve as recommendations for updating future seismic provisions.

For the purpose of this case study, we define a target recovery performance objective where a building should have less than a 25 % chance of exceeding 30 days of functional recovery time given a ground motion equal to two-thirds of the current design earthquake. Thirty days is deemed a good separator of rapid vs long recovery based on previous studies that characterized recovery-based outcomes from performance-based assessment methods [45]. Two-thirds of the current design earthquake is selected as previous efforts have proposed functional recovery ground motions somewhere between life-safety design ground motions and serviceability ground motions [46]. Additionally, our simplified case study models assume elastic response; therefore, quantifying the performance at smaller ground motions is more representative. For the purpose of defining the design earthquake and developing building performance models for each archetype, the seismic design category is set as the generic Dmax condition defined in FEMA P-695 [18].

4.1. Target Design Parameters and Archetype Design Space

The first step of the process is identifying the design space of interest. The dimensions of the design space are characterized by the particular design features of interest as well as other building characteristics of interest that may influence recovery but are not directly used as design features. The design space is limited by the extent of the archetype models considered and the additional assumptions made in the archetype performance simulation.

Following the criteria in Sec. 3.2, we identified four design features of interest, as outlined in Table 3. These design features are identified based on the author’s domain-specific knowledge of the ASCE/SEI 7 design provisions and functional recovery performance assessment methods. Each design feature is anticipated to impact recovery performance, is capable of being adequately represented in the building performance model, and falls within the purview of the structural design engineer, as described in Table 3.

Buildings are typically comprised of many nonstructural systems and components, each of which is not necessarily required to follow the same design requirements. Therefore, the nonstructural design features in Table 3 are implemented as independent design features (sub-features) to control the design and performance of the various nonstructural systems within the building, bringing the total number of design features within the design space to 18 (2 structural features and 16 nonstructural sub-features). Table 4 shows the nonstructural system sub-features which are varied within the design space. Section 4.2 provides additional details into how each nonstructural design sub-feature is incorporated into the building performance model.

In addition to the design variables considered above, we also investigate the impact of the response mode, e.g., shear- vs flexure-type response fundamental mode shapes [47], on functional recovery, but do not explicitly consider it as a design feature when training the decision tree. Table 5 shows all of the archetype characteristics that are varied in the case study assessment, which defines the design space.

Table 3. Selected case study design features.

ID	Design Feature	Anticipated Impact on Functional Recovery	Model Implementation	Potential Practical Implementation
1	Height Limit	For some buildings, it may be difficult to achieve acceptably low functional recovery times using prescriptive means.	Quantify performance across various building heights, from two to 20 stories.	Updated height limits equivalent to Chapter 12 of ASCE/SEI 7.
2	Structural Drift Limit	Reduce damage to structural and nonstructural components but potentially increase acceleration-based damage.	Modify the fundamental period of the structural response model such that peak drifts are below the drift limit.	Updated drift limits equivalent to Chapter 12 of ASCE/SEI 7.
3	Nonstructural Component Capacity	Reduced probability of component, anchorage, or brace failure for various nonstructural components within the building.	Modify the expected capacity of nonstructural fragilities by the input factor.	Updated Ip factors, deformation requirements, and/or detailing requirements equivalent to Chapter 13 of ASCE/SEI 7.
4	Nonstructural Equipment Prequalification Program	Reduce the probability of equipment failure for shaking levels below the expected capacity.	Increase capacity and reduce uncertainty based on a tested prequalification level.	Require certain equipment to be selected from a list of prequalified components (OSP) or be subject to additional testing.

4.2. Target Design Parameters and Archetype Design Space

To simulate the functional recovery time of each archetype model within the design space, we use the performance assessment method outlined in Cook et al. [25] as implemented in the ATC-138 assessment methodology [41]. The two fundamental inputs required to perform such an assessment are (1) estimating the distribution of structural response, in terms of peak story drift ratio and peak floor accelerations, for a given ground shaking intensity, and (2) modeling the fragility and damage consequence of each structural and nonstructural component within the building.

Given the goal of this case study is to illustrate the application of the proposed framework, we did not go through the process of developing explicit structural design and nonlinear response models to represent each structural variant within the design space. Instead, we leverage an approximate elastic response estimation method presented by Miranda and Taghavi [48], following the process outlined below:

1. For a given archetype's number of stories and displaced shape characteristic (e.g., shear- vs flexure-type response), quantify the approximate shape factor and approximate participation factor for the first three modes, according to Miranda and Taghavi [48].
2. Based on the first mode shape factor, calculate the target roof displacement that would satisfy the peak story drift limit.
3. Calculate the fundamental building period that would result in such a roof displacement, given two-thirds of the Dmax design ground motion; estimate the first three modal periods according to Miranda and Taghavi [48].
4. Perform elastic modal response history analysis using the first three response modes and the 44 ground motions from the FEMA far field set [18] scaled to two-thirds of the Dmax shaking intensity at the first-mode period.
5. Use the simulated response profiles, in terms of both peak story drift ratios and peak floor accelerations, as inputs to the archetype performance models.

The response models assessed in this case study are simplified elastic models that do not explicitly reflect the building code design constraints and are unable to capture nonlinear behaviors; future assessments should develop explicit structural designs using the latest seismic provisions and nonlinear response models, similar to what was done for FEMA P-695. However, while the selected structural assessment process outlined here is fundamentally approximate, it allows us to use a consistent physics-based framework to facilitate comparisons between simulated recovery outcomes across various structural response behaviors. Given the range of structural response assessed in this study (i.e., an elastic response below the design earthquake), we assume that collapse and residual drift effects are negligible for all archetype models.

Fundamentally, the recovery-performance assessment framework used in this study is a component assembly procedure [17], where damage to each component in the building is simulated given the estimated structural response, and the final building level is then based on an aggregation of component-level damage and consequences. Following the recovery method outlined in Cook et al. [25], we use the structural and nonstructural component fragility and consequence database assembled as part of the FEMA P-58 [12] framework and populate building components throughout each archetype following the recommended values from the FEMA P-58-3 normative quantities tool. For the purposes of this study, we assume all buildings are multi-family residential occupancy with no vertical or horizontal irregularities. Additionally, to quantify the capacity of anchorage components, we follow the process outlined in FEMA P-58, Volume 2, Section 7.3.1, for calculating component fragility curves from code minimum capacities.

Table 4. Nonstructural design features.

Sub-feature ID	Design Feature ID	Nonstructural System	Nonstructural Component Action
1	3	Cladding	Cladding deformation capacity
2	3	Cladding	Cladding anchorage capacity
3	3	Interior Partitions	Partition wall deformation capacity
4	3	Stairs	Stair deformation capacity
5	3	Elevators	Elevator anchorage capacity
6	4	Elevators	Elevator equipment acceleration capacity prequalification testing
7	3	Domestic Plumbing	Potable plumbing pipe and brace capacity
8	3	Domestic Plumbing	Sanitary plumbing pipe and brace capacity
9	3	Active Fire Suppression	Fire sprinkler drop acceleration capacity
10	3	Active Fire Suppression	Fire sprinkler pipe and brace capacity
11	3	HVAC	HVAC fluid distribution pipe and brace capacity
12	3	HVAC	HVAC equipment anchorage capacity
13	3	HVAC	HVAC air distribution duct and brace capacity
14	4	HVAC	HVAC equipment acceleration capacity prequalification testing
15	3	Electrical	Electrical equipment anchorage capacity
16	4	Electrical	Electrical equipment acceleration capacity prequalification testing

In addition to the response estimation and component population model, there are many additional input factors that can influence building recovery within the recovery-performance assessment method outlined in Cook et al. [25]. Unless otherwise stated, we followed the default assumptions prescribed in the ATC-138 implementation of the method [41], including impeding factor models, long lead times for select nonstructural equipment, tenant-specific functional recovery requirements for multi-family residential buildings, and the use of temporary repair measures to mitigate the consequences of certain component damage. Future applications of the proposed framework should carefully review any assumed input into the recovery model and incorporate additional building characteristics directly into the design space as needed.

For the nonstructural design features considered in this study, nonstructural component capacity enhancements (design feature 3 from Table 3) are incorporated into the model by directly amplifying the median component capacity by the input factors. Therefore, the recommendations provided by the decision tree model for nonstructural component capacity will represent minimum increases in component capacity beyond the current state of design, which are required to achieve target recovery objectives; this amplification may or may not proportionally translate to existing nonstructural design factors such as the Ip factor in Chapter 13 of ASCE/SEI 7 due to variations in component overstrength, nominal design and construction practices, and alternative failure modes. Additional judgment and analysis are needed to translate these capacity requirements into design requirements for future design provisions. Prequalified equipment (design feature 4 from Table 3) is represented within the building performance model by developing new fragility curves with high capacity and reduced uncertainty, reflecting the existing California Department of Health Care Access and Information (HCAI) Special Seismic Certification Preapproval program for prequalifying equipment in hospitals. The prequalification level can then be amplified beyond current code requirements using the design feature input factor. The prequalification fragility development follows the process outlined in the ATC-138 method to define new fragility models based on prequalification testing.

Table 5. Case study design space.

Archetype Characteristic Variable	Domain
Response mode	[shear, flexure]
Number of stories	∈ [2, 20]
Peak story drift ratio	∈ [0.02, 0.005]
Nonstructural component capacity	∈ [1, 3]
Nonstructural equipment prequalification	[true, false]
Nonstructural equipment prequalification level	∈ [1, 3] | prequalified=true

4.2.1. Sensitivity Assessment

Following the procedure outlined in Sec. 3.2, we perform a sensitivity study to help refine the domains of the design features in Table 3 to be within the range of influence over our target performance objective and reduce some computation expense in the simulation of the full design space. In Phase I of the sensitivity study, we constrain all nonstructural capacity factors to be equal and uniformly scale them across the range of their proposed domain; here, all applicable nonstructural equipment is set as prequalified. This process is repeated across each structural variant and at several increments across the drift limit domain. Fig. 6 shows the aggregated outcomes of Phase I of the sensitivity study, where each cell represents the shortest functional recovery time (75th percentile) for all models falling into the respective cell; dark-shaded cells represent times that exceed the performance target of 30 days. Based on this assessment, most design feature combinations result in simulated recovery outcomes within the range of the target performance objective, with the exception of archetype models with drifts greater than 1.5 % and nonstructural capacity factors less than 2. Therefore, these sections of the design space do not need to be considered in the simulation of the full design space.

In Phase II of the sensitivity assessment, we perform an inverse one-at-a-time (OAT) assessment where all nonstructural design features are set to their maximum value, and then each design feature is reduced, independently, over their proposed domain. This process is repeated across each structural variant and at several increments across the drift limit domain. Fig. **7** shows the aggregated outcomes of Phase II of the sensitivity study, where each cell represents the longest functional recovery time (75th percentile) for all models falling into the respective cell; dark-shaded cells represent times that exceed the performance target of 30 days. If all of the nonstructural component modifications pass the performance objective across their domain and across various drift limits, it is unlikely that they need to be improved beyond current design requirements and, therefore, can be removed from the full design space simulation. The sensitivity study is an optional assessment as it primarily serves as a mechanism to make the simulation of the full design space more computationally efficient.

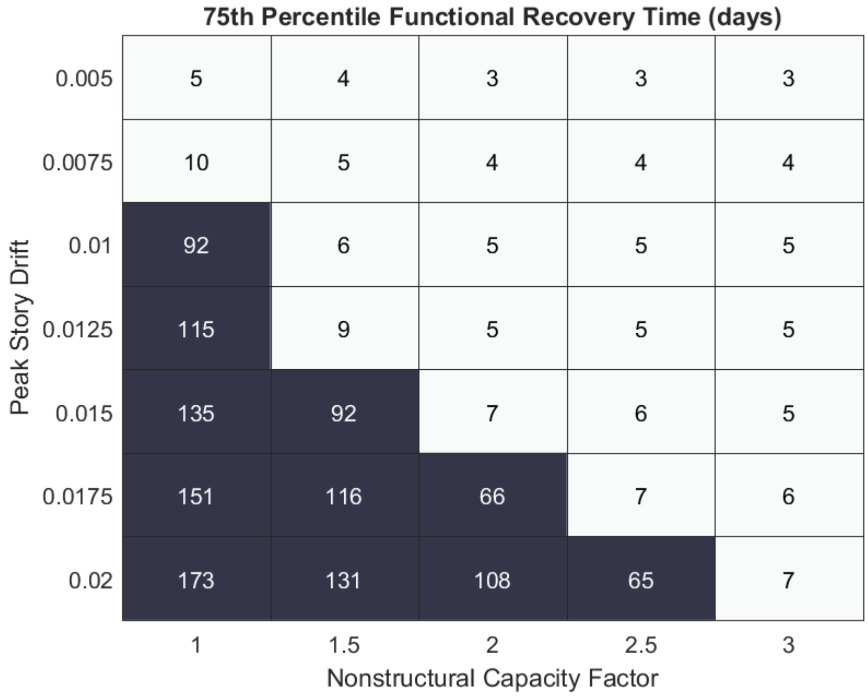

Fig. 6. Minimum functional recovery outcomes (75th percentile) across archetype models.

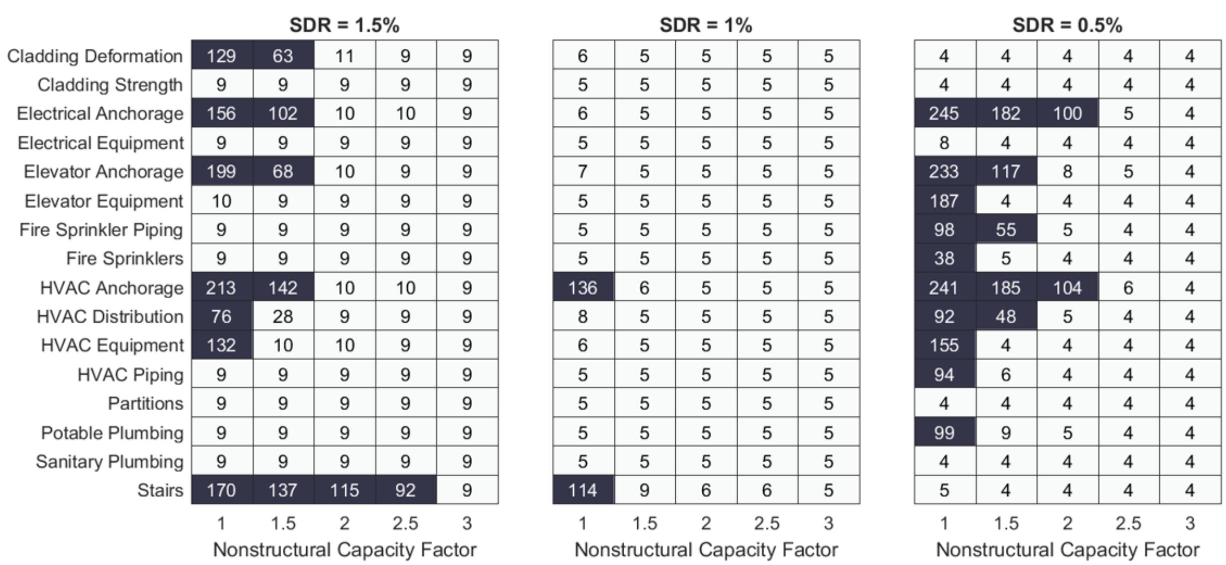

Fig. 7. Maximum functional recovery outcomes (75th percentile) across archetype models.

4.2.2. Design of Computer Experiments

Simulating the recovery performance of the full design space with 18 independent design features, using a full grid search, and assuming an average of six discretizations per feature would require approximately 100 trillion archetype model simulations; this is computationally intractable for most applications. Instead, we populate the design space using a collection of hand selection and random sampling approaches, as discussed in Sec. 3.3.1 and outlined in Table 6. This sampling approach allows us to efficiently fill the design space without significantly compromising the decision tree's performance; the selected protocol in Table 6 samples a total of 92,584 archetype models to population the design space. Not all sampled archetypes have unique structural designs; much of the design space is populated by variations in nonstructural components and detailing.

Table 6. Design space sampling protocol.

Sample Group	Sample Type	Drift sampling	Nonstructural sampling	Group Sample Size
Baseline	Hand selected	Grid search from Є[0.02, 0.005] at increments of 0.0025	NA	22
Inverse OAT	Hand selected	Grid search from Є[0.02, 0.005] at increments of 0.0025	Scale each nonstructural sub-feature one at a time (OAT) across the domain Є[1, 3] at increments of 0.25, where all other sub-features are held at their maximum value.	22,176
Uniform Scaling	Hand selected	Grid search from Є [0.02, 0.005] at increments of 0.0025	Uniform scaling across all nonstructural sub-features across the domain Є[1, 3] at increments of 0.25.	1,386
Inverse pRand MAT	Pseudo random	Grid search from Є[0.02, 0.005] at increments of 0.0025	Inverse OAT, using multiple nonstructural sub-features at a time. Grid search of the number of sub-features from Є[2, n-1]. Random search of each selected nonstructural sub-features capacity modification factor from Є[1, 3]; selected groups are correlated.	49,000
Monte Carlo	Random	Random search from Є[0.02, 0.005]	Random search across all nonstructural sub-features, where the nonstructural capacity factor varies from Є[1, 3]. Nonstructural sub-features are assumed to be independent.	10,000
LHS	Random	Random search from Є[0.02, 0.005]	Random search across all nonstructural sub-features, where the nonstructural capacity factor varies from Є[1, 3]. Nonstructural sub-features are assumed to be independent.	10,000

The sampling protocol used in this study consists of three hand-selected groups, two random sampling groups, and one pseudo-random group that combines a grid search with a correlated Monte Carlo sampling approach. The hand-selected groups are based on the authors' domain-specific knowledge of the functional recovery performance method and are intended to identify specific boundaries of the design space that may significantly impact outcomes. For example, the inverse one-at-at-time (OAT) approach targets the point where the weakness of one nonstructural system alone leads to unacceptable performance. The random sampling approaches help to populate the remainder of the design space.

4.3. Decision Tree Modeling

After the recovery performance of each archetype model in the design space is simulated, we classify each performance outcome as either passing or failing the target performance objective and assemble all design features and target performance classes into one large database for training the decision tree.

4.3.1. Visualizing the Simulated Performance

Fig. 8 plots the classification of the simulated recovery performance for the entire design space with respect to peak story drift and the capacity factor applied to the deformation capacity of the stairs; the plot illustrates both the hand-selected and randomly sampled approach within the sampling protocol, where the dark vertical lines represent grid searches at specific bins of story drift, and the gaps between the lines are filled in by additional random sampling. Here, we see a clear trend where archetype models tend to fail the performance objective for larger drifts and stairs with small capacity factors and pass for more stringent drift limits and larger stair capacity factors. However, there is no definitive boundary, and mixed regions of passing and failing models remain. Therefore, additional design factors beyond the drift limit and stair capacity factor are required to further separate the performance of the archetype models within the design space; the decision tree model facilitates the process of separating the data considering the highly dimensional design space.

4.3.2. Tuning, Training, and Testing

Using the assembled design feature and target classification database, we tune, train, and test the decision tree model to the data, according to the process outlined in Sec. 3.5. The decision tree model is first tuned using 10-fold cross-validation and a grid search over the hyperparameters: minimum leaf size, maximum number of tree splits, and maximum number of features. The curves in Fig. 9 show how the resulting decision tree model's precision and F1-score change with variations in the assessed hyperparameters.

To define the best hyperparameters to use for training the decision tree, only hyperparameters that resulted in precision scores above 0.9, then F1-scores within 0.01 of the maximum remaining F1-scores are selected. Among the remaining tuning cases, the hyperparameters that lead to the simplest models are selected, prioritizing large sample leaves, then small tree depths, and then limiting the number of features. From this process, we selected the following tuning parameters:

- Minimum leaf size: 1000

- Maximum number of tree splits: 16
- Maximum number of features: No limit

Given the prioritization amongst the hyperparameters, no limitations were put on the number of features. This is by design as constraining both the leaf size and the tree depth will automatically force the decision tree to only use the most influential features within the given constraints.

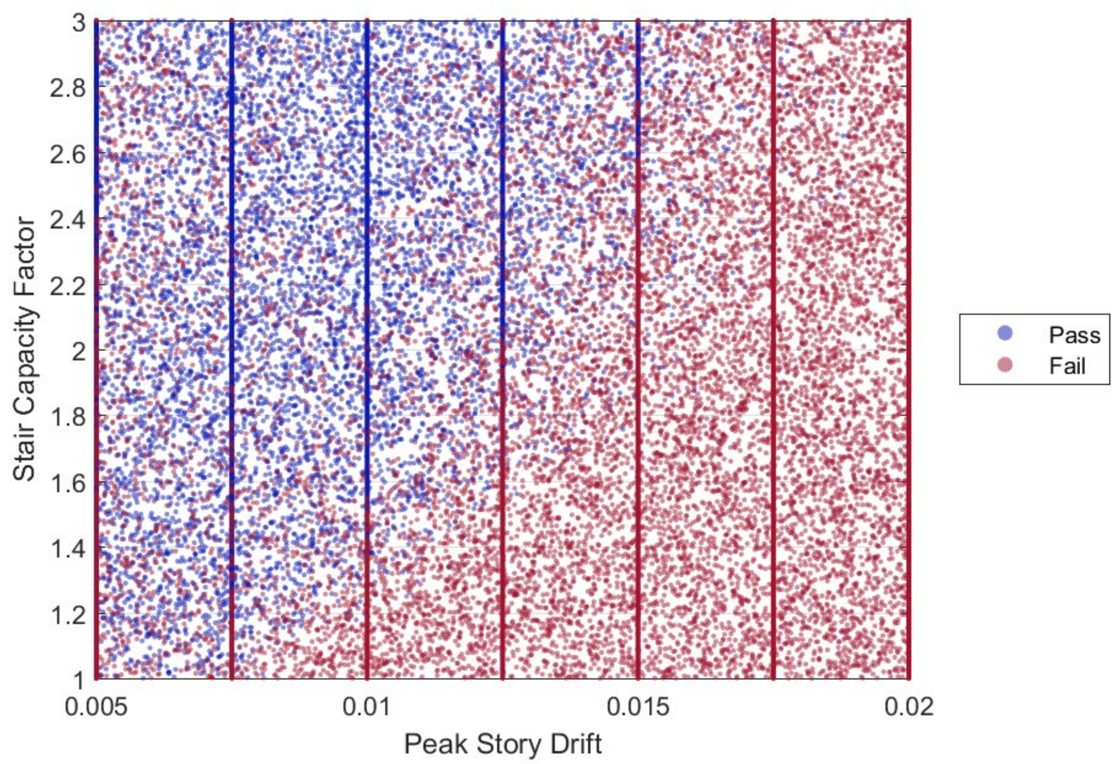

Fig. 8. Visualization of simulated recovery performance classification with respect to two design features: peak story drift and the deformation capacity of the stairs. Models labeled as passing have less than a 25 % chance of exceeding 30 days of functional recovery time.

Once the hyperparameters have been set, the decision tree is trained on an 80/20 split of the data, where 80 % of the data is randomly selected for training, and the other 20 % is set aside for testing. For the case study assessed here, we trained the decision tree using the *fitctree* function in MATLAB; similar packages are available in other common high-level programming software such as Python or R. The trained decision tree model is shown in Fig. 10. Based on the simulated data and hyperparameter constraints, the decision tree identifies five influential design features: drift limits, HVAC equipment prequalification, HVAC anchorage capacity, stair deformation capacity, and building height limits. Through the hierarchy of the tree, various combinations of required design feature values lead to five leaf nodes that satisfy the recovery performance objective of less than a 25 % probability of exceeding 30 days of recovery given the input ground motion.

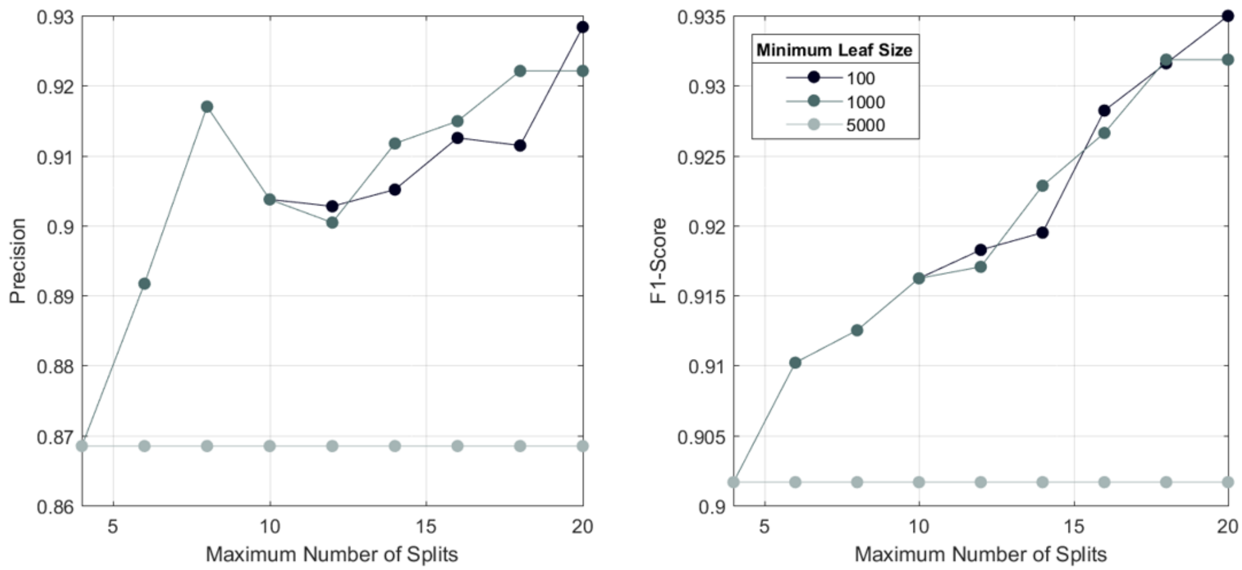

Fig. 9. Decision tree performance curves from tuning process.

The trained decision tree model is then tested on the testing data partition that was previously set aside before testing. In this testing process, the design features of each archetype model in the testing data set are assessed through the decision tree model to formulate a predicted outcome class: passing or failing the recovery performance acceptance criteria. Those predicted classifications are then compared to the original classifications from each archetype recovery performance simulation in the form of a confusion matrix, as shown in Fig. 11. The confusion matrix compares the total number of outcomes of each class from the decision tree's prediction (vertical axis) to the total number of outcomes of each class from the underlying simulated data (horizontal axis). From the confusion matrix, decision tree performance metrics such as Precision and F1-score are calculated and shown in Fig. 11; for the case study assessed here, the trained model maintains relatively high precision and F1 score.

4.4. Output Interpretation and Heuristic Regularization

As discussed in Sec. 3.5.5, we can reorganize the structure of the decision tree model from the previous section into a tabularized set of design feature values required to meet the target recovery performance objective, as shown in Table 7.

The interpretation of outcomes from the decision tree model depends on how each feature is represented in the building performance model. Here, the nonstructural capacity factors represent theoretical changes in nonstructural capacity; therefore, this decision tree leaves represent the minimum capacity increase required to achieve target functional recovery performance. This may or may not be achieved by specific provision updates such as increases in the Ip factor. For example, increasing stair deformation capacity by 1.85 in leaf 1 of Table 7 may or may not be effectively achieved by specifying that high-level design requirements for stair deformation compatibilities equal to 1.85 times the current drift limit; specific detailing requirements may need to be established to achieve such performance in reality.

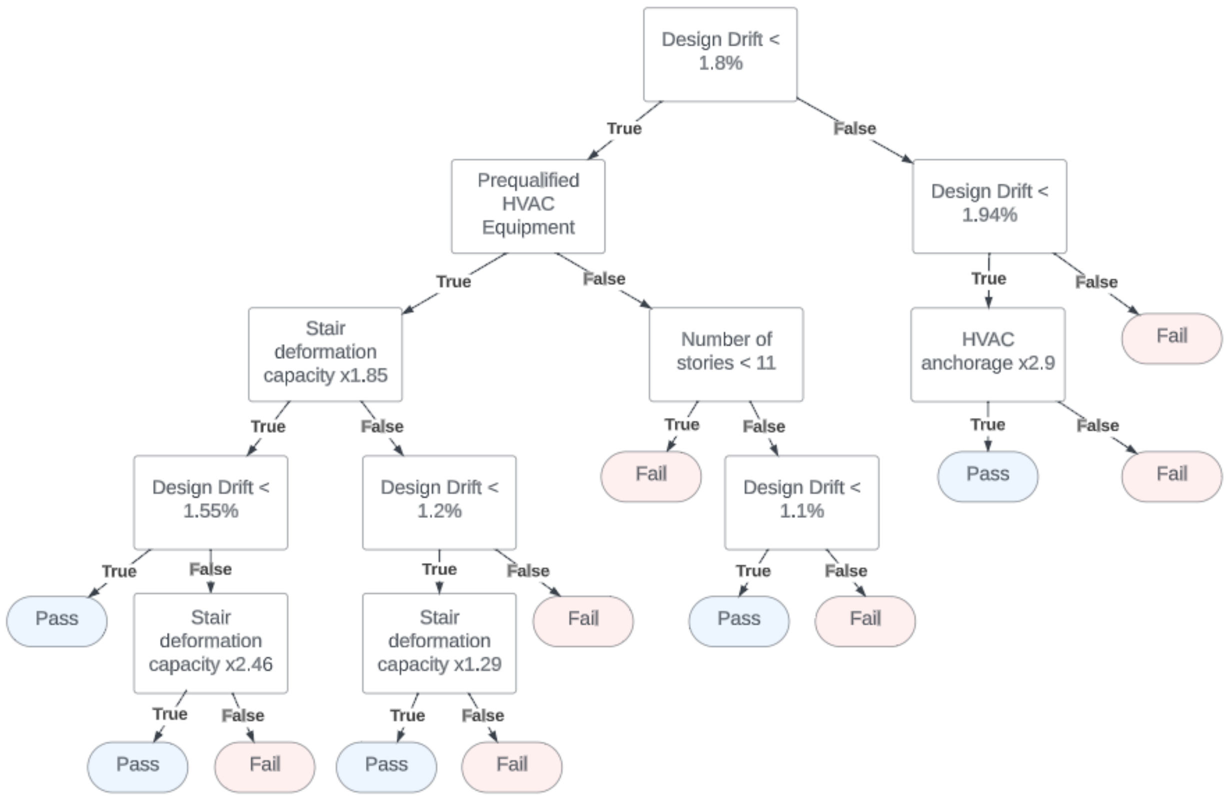

Fig. 10. Decision tree model that was trained to simulated functional recovery results from the case study archetype design space. Branches of the decision tree identify building design features that result in a less than 25 % chance of exceeding 30 days of functional recovery time given two-thirds of the design earthquake.

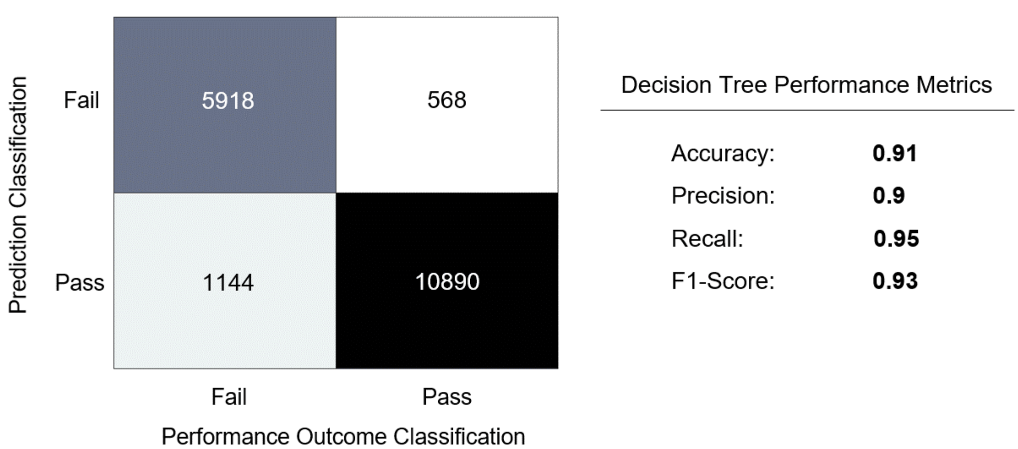

Fig. 11. Confusion matrix and performance metrics showing the testing outcomes from the trained decision tree model. Each number with the confusion matrix corresponds to the total number of archetype performance outcomes observed within each case.

In the heuristic regularization process, SMEs should have a thorough understanding of how each design feature is modeled in the recovery simulation to adequately interpret and constrain the decision tree model. For the purpose of this case study, we did not engage directly with a group of SME experts. Instead, we developed a set of simplifying guidelines that we assumed to have come from an SME group and used those guidelines to further constrain and evolve the design recommendations. The assumed rules are as follows:

1. All drift limits should be in increments of 0.25 %, rounding conservatively.
2. All nonstructural capacity modification factors should be in increments of 0.25, rounding conservatively.
3. Where two criteria are similar, attempt to combine the criteria into one general criteria.

Following the above guidelines, we are able to simplify the branch criteria in Table 7, into a set of three design criteria by (1) combining leaves 1 and 2 and (2) removing leaf 4 in favor of leaf 3. The resulting recommended design criteria are shown in Table 8.

Table 7. Decision tree structure reorganized into a list of all passing leaves.

Leaf	Branch Criteria
1	Design drift < 1.55 %, stair deformation capacity increased by 1.85 factor, prequalified HVAC equipment.
2	Design drift < 1.8 %, stair deformation capacity increased by 2.46 factor, prequalified HVAC equipment.
3	Design drift < 1.2 %, stair deformation capacity increased by 1.29 factor, prequalified HVAC equipment.
4	For buildings 11 stories or greater, design drift < 1.1 %, prequalified HVAC equipment.
5	Design Drift < 1.94 %, HVAC anchorage capacity increased by 2.9 factor.

Table 8. Building design criteria required to meet acceptable performance from example decision tree model.

Design Check	Passing Criteria
1	Design drift < 1. 5%, stair deformation capacity increased by 2.5 factor, prequalified HVAC equipment.
2	Design drift < 1 %, stair deformation capacity increased by 1.5 factor, prequalified HVAC equipment.
3	HVAC anchorage capacity increased by a factor of 3.

4.4.1. Testing Final Recommendations

Once the heuristic regularization process is complete, the final set of recommendations should be retested using the testing dataset. This additional testing is to ensure that the process of simplifying the model based on engineering judgment does not result in unexpected reductions in the performance of the decision tree model. Testing the regularized model follows the same process as testing the original trained decision tree model; each observation of archetype model design features in the testing dataset is assessed using the criteria established in Table 8. The predicted classifications are then compared to the original classifications and assembled into a confusion matrix. This process is repeated using 10-fold cross-validation to ensure the final tested values are not biased to one particular partition of the data. Fig. 12 shows the confusion matrix for one of the tested partitions of the recommended design criteria from Table 8. The final design criteria show a minor drop in F1-score but a slight increase in precision compared to the original model. This change is due to the impact of the conservative bias during the heuristic regularization process, reflecting the goal of the final design recommendation to minimize unconservative results (i.e., the performance goal of prescriptive design requirements is high precision). The tested decision tree performance is similar across all tested partitions.

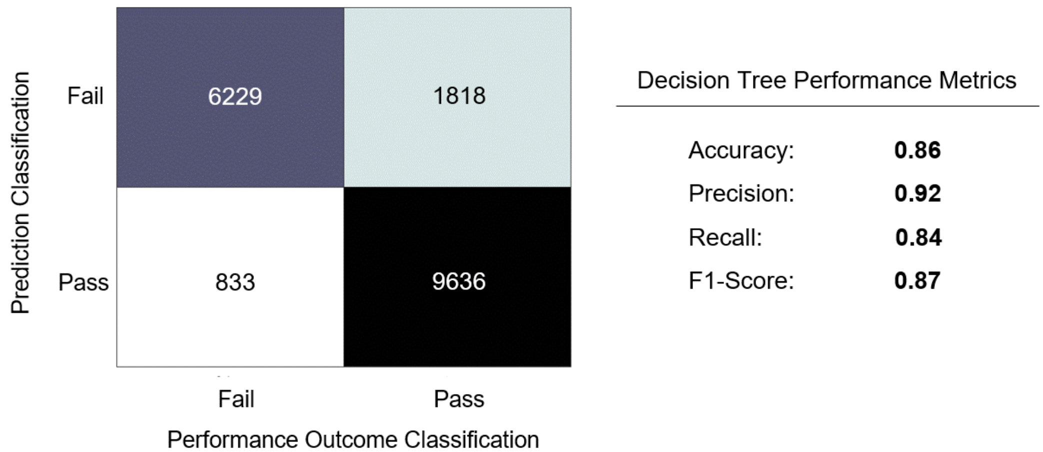

Fig. 12. Confusion matrix and performance metrics of the final recommended design criteria.

4.5. Testing Alternative Performance Metrics

Once the performance of all archetype models in the design space has been simulated, the process of training, testing, and regularizing a decision tree model into design recommendations can be easily repeated across various target performance metrics using the same simulated data. Analysts can compare design requirements across various performance metrics, such as targeting red tags or reoccupancy instead of functional recovery, various recovery time targets, or various reliability targets. Table 9 shows the recommended design values from decision tree models fit to the simulated archetype models assessed in this case study, but each targeting different performance objectives. From this comparison we can observe how baseline models (i.e., those meeting basic life safety design requirements) already satisfy a functional recovery performance target of a 50 % chance of exceeding 120 days. As the recovery time and reliability targets become more stringent, the design values required to meet the given performance objective become more rigorous.

Table 9. Comparison of recommended design values (simplified from full decision tree) from various performance objectives.

Reliability Target	30-Day Functional Recovery Target	120-Day Functional Recovery Target
10 % Reliability	Drift < 0.75 %, stairs x1.5 — Drift < 1.25 %, stairs x2, prequalified HVAC, HVAC anchorage x1.5 — Drift < 1.5 %, stairs x3, cladding deformation x3	Drift < 1 %, stairs x1.25 — Drift < 1.5 %, stairs x2, prequalified HVAC
25 % Reliability	Drift < 1.5 %, stairs x2.5, prequalified HVAC — Drift < 1 %, stairs x1.5, prequalified HVAC — HVAC anchorage x3	Drift < 1.75 %
50 % Reliability	Drift < 1.25 % — Drift < 1.5 % and stairs x1.5 — Drift < 1.75 %, stairs x1.5, prequalified HVAC — Stairs x2.25, prequalified HVAC, cladding deformation x1.5	Baseline design satisfies

5. Conclusions

This report presents a technical framework that utilizes the performance-based earthquake engineering methodology and machine learning techniques to quantify minimum prescriptive design requirements that satisfy target functional recovery performance objectives for new buildings. More specifically, the framework provides an analytical process for training a decision tree to identify key building design characteristics that satisfy minimum performance objectives based on the collective simulated performance of a set of archetype buildings. While the overall framework presented in this report is generally applicable to multiple hazards and building performance objectives, we focus here on applications to the functional recovery performance of buildings in earthquakes, using the performance-based assessment methodology presented in Cook et al. [25], as implemented in ATC 138 [41].

In its underlying architecture, the framework is similar to the FEMA P-695 assessment methodology, where the performance of a set of archetype buildings is compared to a target performance objective to assess the acceptability of a given set of prescriptive design values. However, in leveraging the decision tree model, the proposed framework presents a robust and repeatable process to identify the most influential and informative design parameters in a highly dimensional design space and translate trends normally only captured by building-specific performance models into prescriptive-based requirements. In the various sections of this report, we outline the process for selecting design features and building characteristics of interests to define the design space; developing building performance models to represent each archetype building within the design space; simulating from the design space in a computationally efficient manner; tuning, training, and testing a decision tree model; and constraining decision tree outcomes based on SME domain knowledge to formulate recommended design requirements. In Sec. 4 of this report, the framework is exercised on a set of simplified elastic building models to illustrate the application and outcomes of the proposed framework.

Key to the framework is the feedback loop between SMEs and the decision tree model. Not all aspects of building performance, construction constraints, economics, and design optimization can adequately be captured in the performance simulation. Therefore, SMEs remain key gatekeepers to translate model outcomes into practical and effective seismic provisions.
The heuristic regularization cycle actively integrates SME feedback within the decision tree training process to evolve the decision tree model from a theoretical set of design values to minimum design requirements that are directly applicable to the design process and adoption into the building code.

Fundamentally, the analytical framework presented in this report provides a decision support tool for experts engaged in the building code update process to translate highly dimensional and nonlinear performance trends into transparent design recommendations.

5.1. Framework Limitations

The design recommendations developed from the proposed analytical framework depend on the specific performance-based assessment methodology used to simulate the performance of each archetype model within the design space. Therefore, any design requirement recommended by the trained decision tree model is subject to any and all limitations and modeling assumptions of the underlying performance assessment framework. Analysts should be careful to document and communicate all key modeling assumptions as they directly influence the scope and applicability of the design recommendations. For example, in the illustrative case study, we used a design feature that directly amplified the existing nonstructural component's capacity (fragility median) by a given factor of interest. Therefore, the resulting design requirements indicate the minimum change in component capacity required to meet the recovery goals. Actual design requirements, such as amplifying the minimum design force by an increased Ip factor in Chapter 13 of ASCE/SEI 7, may or may not proportionally relate to component capacity if other construction constraints or failure modes control. Care should be given to understand how the proposed nonstructural design changes relate to "physical" installation differences and what requirements or clauses should be added to current design provisions to best provide the required increase in component capacity.

Design recommendations from the decision tree model are only applicable to buildings within the acceptable scope of the design space assessed. It is not possible to assess all buildings that are permissible in current seismic design provisions, therefore, early engagement with SMEs will help to identify an appropriate design space that is sufficient to address the research goals in the development of new design provisions.

Additionally, a decision tree is a form of a greedy algorithm; it will select the most informative split in the data at each node but will not recursively optimize the entire tree. Therefore, the recommended outcomes from the decision tree model represent one possible solution to separate the design features that led to acceptable performance but may not be the only solution and may not be the optimal solution (which depends on an objective function). Tree-based models can also be sensitive to noise in the data and may fluctuate based on selected hyperparameters. However, simplifications to the decision tree model made in the heuristic regularization process are expected to limit the impact of model fluctuations. Additionally, the definition of what constitutes a reliable split or sufficient model performance can be quite subjective.

5.2. Alternative Approaches

Beyond the analytical framework presented in this study, there are alternative approaches that could be pursued, targeting the development of similar prescriptive design goals, each with its own advantages and shortcomings. For example, analysts could follow something closer to the FEMA P-695 [18] process by grouping archetypes into bins and setting criteria for minimum and average performance. Alternatively, if a consensus objective function is available, an optimization-based approach, similar to that proposed by Issa et al. [37], could help provide specific solutions that minimize conditions such as construction cost or design complexity. If an analyst is interested in mapping continuous building response values with functional recovery as a continuous variable, a regression approach similar to that proposed by Kolozvari & Terzic [43] could be used. Finally, the archetype's performance could be defined by a closed-form solution, and design features that satisfy specific performance objectives could simply be back-calculated. For example, Mieler et al. [21] proposed a framework in which the fault trees that are used to model building functionality given component damage could be inversed to identify minimum component capacities required to achieve specific building function reliability levels. However, to solve the inversed tree, the analysis would need to define specific component damage correlations and a subjective weighting function to converge to a single solution.

Within the goal of developing functional recovery design criteria, there is currently no one consensus target performance objective. Instead of explicitly quantifying functional recovery time, alternative metrics, such as targeting significant jumps in recovery time uncertainty or targeting specific damage states, such as structural damage that requires immediate repair, could be investigated. While assessing various performance objectives is within the application of the proposed framework, the analysts should take care to ensure the performance assessment methodology properly reflects the target performance objective. If the performance objective varies significantly, alternative frameworks may be preferred. For example, Perrone et al. [49] developed a conceptual framework to set nonstructural performance factors analogous to structural system factors via FEMA P-695 by explicitly considering the floor response spectrum and specific variations in nonstructural system configuration. The framework is intended to quantify design factors for individual nonstructural components rather than explicitly target building-level recovery goals.

5.3. Future Work

Assessing and improving community recovery is a multidisciplinary and complex problem made up of technical, organizational, social, and economic aspects [5]. The goal of the proposed framework is to provide a decision support tool to serve as a technical link between probabilistic functional recovery models and practical design implementation strategies for buildings to improve community resilience and reduce disaster impacts. While the report exercises the proposed framework for a specific example archetype set and target performance objective, more work is needed to expand the application of the framework to a more robust set of models with wider applications.

Given the lack of consensus around target performance objectives for the recovery-based design of new buildings, additional target performance objectives, reliabilities, recovery times, and conditional ground motion intensities should be assessed to provide adequate decision support data for the code update process. In this process, the analysts should be sure to explore additional design features, building characteristics, and modeling assumptions beyond what was presented in the illustrative case study. Additionally, special consideration should be given to develop specific design requirements for nonstructural components that are flagged by future applications of this framework, perhaps leveraging the methodology presented by Perrone et al. [49], stop-gap measures, or design guidelines derived based on expert opinion.

References

[1] Y. Zhang, J. F. Fung, K. J. Johnson, and S. Sattar, "Review of Seismic Risk Mitigation Policies in Earthquake-Prone Countries: Lessons for Earthquake Resilience in the United States," *Journal of Earthquake Engineering*, vol. 26, no. 12, pp. 6208–6235, 2022, doi: 10.1080/13632469.2021.1911889.

[2] M. W. Mieler and J. Mitrani-Reiser, "Review of the State of the Art in Assessing Earthquake-Induced Loss of Functionality in Buildings," *Journal of Structural Engineering*, vol. 144, no. 3, 2017.

[3] NIST and FEMA, *NIST-FEMA Special Publication FEMA P-2090/NIST SP-1254: Recommended Options for Improving the Built Environment for Post-Earthquake Reoccupancy and Functional Recovery Time*. Gaithersburg, MD: National Institute of Standards and Technology, 2021.

[4] S. Sattar *et al.*, "Recommended Options for Improving the Functional Recovery of the Built Environment," 17th World Conference on Earthquake Engineering, 2020.

[5] M. Bruneau *et al.*, "A Framework to Quantitatively Assess and Enhance the Seismic Resilience of Communities," *Earthquake Spectra*, vol. 19, no. 4. Nov. 2003.

[6] S. Sattar *et al.*, "Research needs to support immediate occupancy building performance objective following natural hazard events," Gaithersburg, MD, Aug. 2018. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1224.pdf

[7] ASCE, *ASCE/SEI 7-22: Minimum design loads and associated criteria for buildings and other structures.* 2022.

[8] ASCE, *ASCE 41-17: Seismic Evaluation and Retrofit of Existing Buildings*. Restin, Virginia: American Society of Civil Engineers, 2017.

[9] FEMA, *FEMA P-2156, The Role of the NEHRP Recommended Seismic Provisions in the Development of Nationwide Seismic Building Code Regulations: A Thirty-Five-Year Retrospective*. Federal Emergency Management Agency, 2021.

[10] M. Davis and K. Porter, "The public's role in seismic design provisions," *Earthquake Spectra*, vol. 32, no. 3, 2016.

[11] FEMA, *FEMA P-58: Seismic Performance Assessment of Buildings*. Federal Emergency Management Agency, 2012.

[12] FEMA, *FEMA P-58: Seismic Performance Assessment of Buildings: Second Edition*. Federal Emergency Management Agency, 2018.

[13] K. A. Porter, "An Overview of PEER's Performance-Based Earthquake Engineering Methodology," in *9th International Conference on Application in Statistics and Probability in Civil Engineering*, San Francisco, 2003.

[14] G. G. Deierlein and J. Moehle, "A Framework Methodology for Performance-Based Earthquake Engineering," in *13th World Conference on Earthquake Engineering*, Vancouver: EERI, 2004.

[15] J. Mitrani-Reiser, "An Ounce of Prevention: Probabilistic Loss Estimation for Performance-Based Earthquake Engineering. Ph.D. Thesis," California Institute of Technology, 2007.

[16] C. A. Cornell, "Engineering Seismic Risk Analysis," *Bulletin of the Seismological Society of America*, vol. 58, no. 5, 1960.

[17] K. Porter, A. Kiremidjian, and J. LeGrue, "Assembly-Based Vulnerability of Buildings and Its Use in Performance Evaluation," *Earthquake Spectra*, vol. 17, no. 2, 2001.

[18] FEMA, *FEMA P-695: Quantification of Building Seismic Performance Factors.*, no. June. Federal Emergency Management Agency, 2009.

[19] M. C. Comerio, "Estimating downtime in loss modeling," *Earthquake Spectra*, vol. 22, no. 2, May 2006.

[20] S. E. Chang, *Urban disaster recovery: a measurement framework and its application to the 1995 Kobe earthquake*. Oxford: Blackwell Publishing, 2010.

[21] M. W. Mieler, S. R. Uma, and J. Mitrani-Reiser, "Using Failure Analysis Tools to Establish Seismic Resilience Objectives for Building Components and Systems," *Bulletin of the New Zealand Society for Earthquake Engineering*, vol. 49, no. 1, 2016.

[22] ICC, *A National Approach to Seismic Functional Recovery for New Construction: A roundtable discussion convened by the International Code Council and California Building Officials*. International Code Council, 2020. [Online]. Available: www.iccsafe.org

[23] NIBS, *Functional Recovery Planning Committee Report*. National Institute of Building Sciences, 2023.

[24] I. Almufti and M. Willford, "REDi Rating System: Resilience-based Earthquake Design Initiative for the Next Generation of Buildings," 2013.

[25] D. T. Cook, A. B. Liel, C. B. Haselton, and M. Koliou, "A framework for operationalizing the assessment of post-earthquake functional recovery of buildings," *Earthquake Spectra*, vol. 1, no. 36, Aug. 2022, doi: 10.1177/87552930221081538.

[26] C. Molina-Hutt, T. Vahanvaty, and P. Kourehpaz, "An Analytical Framework to Assess Earthquake Induced Downtime and Model Recovery of Buildings," *Earthquake Spectra*, 2022.

[27] V. Terzic and P. K. Villanueva, "Method for probabilistic evaluation of post-earthquake functionality of building systems," *Eng Struct*, vol. 241, p. 112370, Aug. 2021, doi: https://doi.org/10.1016/j.engstruct.2021.112370.

[28] C. Haselton, "Resilient Design for Functional Recovery – Approach and Design Steps Using ATC-138 (and FEMA P-58), and Example Projects Implementing this in U.S. Practice," in *Canadian Conference - Pacific Conference on Earthquake Engineering*, Vancouver: Canadian Association for Earthquake Engineering, Jun. 2023.

[29] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification And Regression Trees*. Routledge, 1984. doi: 10.1201/9781315139470.

[30] L. Breiman, "Random Forests," *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[31] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[32] I. Jolliffe, "Principal Component Analysis," in *Wiley StatsRef: Statistics Reference Online*, Wiley, 2014. doi: 10.1002/9781118445112.stat06472.

[33] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY: Springer New York, 2009. doi: 10.1007/978-0-387-84858-7.

[34] H. Adeli and C. Yeh, "Perceptron Learning in Engineering Design," *Computer-Aided Civil and Infrastructure Engineering*, vol. 4, no. 4, pp. 247–256, 1989.

[35] S. Guikema, “Artificial Intelligence for Natural Hazards Risk Analysis: Potential, Challenges, and Research Needs,” *Risk Analysis*, vol. 40, no. 6, pp. 1117–1123, Jun. 2020, doi: 10.1111/risa.13476.

[36] M. A. Hariri-Ardebili, G. Mahdavi, L. K. Nuss, and U. Lall, “The role of artificial intelligence and digital technologies in dam engineering: Narrative review and outlook,” *Eng Appl Artif Intell*, vol. 126, Nov. 2023, doi: 10.1016/j.engappai.2023.106813.

[37] O. Issa, R. Silva-Lopez, J. W. Baker, and H. V. Burton, “Machine-learning-based optimization framework to support recovery-based design,” *Earthq Eng Struct Dyn*, vol. 52, no. 11, pp. 3256–3280, Sep. 2023, doi: 10.1002/eqe.3860.

[38] J. Kiani, C. Camp, and S. Pezeshk, “On the application of machine learning techniques to derive seismic fragility curves,” *Comput Struct*, vol. 218, pp. 108–122, Jul. 2019, doi: 10.1016/j.compstruc.2019.03.004.

[39] P. Kourehpaz and C. Molina Hutt, “Machine Learning for Enhanced Regional Seismic Risk Assessments,” *Journal of Structural Engineering*, vol. 148, no. 9, Sep. 2022, doi: 10.1061/(asce)st.1943-541x.0003421.

[40] H. Sun, H. V. Burton, and H. Huang, “Machine learning applications for building structural design and performance assessment: State-of-the-art review,” *Journal of Building Engineering*, vol. 33. Elsevier Ltd, Jan. 01, 2021. doi: 10.1016/j.jobe.2020.101816.

[41] ATC, *ATC-138-3: Seismic Performance Assessment of Buildings Volume 8 – Methodology for Assessment of Functional Recovery Time*. Applied Technology Council, 2021. [Online]. Available: www.ATCouncil.org

[42] F. Naeim, “Performance of Extensively Instrumented Buildings During the January 17, 1994 Northridge Earthquake,” 1997.

[43] K. Kolozvari and V. Terzic, “Methodology for developing practical recovery-based design requirements for buildings,” *Eng Struct*, vol. 274, Jan. 2023, doi: 10.1016/j.engstruct.2022.115102.

[44] H. Burton, H. Xu, and Z. Yi, “Design of computer experiments for developing seismic surrogate models,” *Earthquake Spectra*, vol. 38, no. 1, pp. 384–406, Feb. 2022, doi: 10.1177/87552930211033309.

[45] D. Cook, A. Liel, and A. Safiey, “Functional Recovery in Modern Reinforced Concrete Buildings,” *In Review*, 2023.

[46] NIST, “NIST GCR 12-917-20 Tentative Framework for Development of Advanced Seismic Design Criteria for New Buildings,” Gaithersburg, 2012.

[47] E. Miranda, “Approximate Seismic Lateral Deformation Demands in Multistory Buildings,” *Journal of Structural Engineering*, vol. 125, no. 4, 1999.

[48] E. Miranda and S. Taghavi, “Approximate Floor Acceleration Demands in Multistory Buildings. I: Formulation,” *Journal of Structural Engineering*, vol. 131, no. 2, pp. 203–211, Feb. 2005, doi: 10.1061/(asce)0733-9445(2005)131:2(203).

[49] D. Perrone, D. Rodriguez, A. Filiatrault, E. Brunesi, C. Beiter, and R. Piccinin, “A Framework for the Quantification of Non-Structural Seismic Performance Factors,” *Journal of Earthquake Engineering*, vol. 26, no. 16, pp. 8468–8494, 2022, doi: 10.1080/13632469.2021.1991516.