

Article

Reanalysis of Reinforced Concrete Frames via a Three-Layer Machine Learning Framework: Sensitivity-Based Features and Model Interpretability

Yohannes L. Alemu ^{1,*} , Bedilu Habte ², Girum Urgessa ³ , Christian Walther ¹ and Tom Lahmer ¹ 

¹ Institute of Structural Mechanics, Bauhaus University, 99423 Weimar, Germany; christian.walther@uni-weimar.de (C.W.); tom.lahmer@uni-weimar.de (T.L.)

² School of Civil and Environmental Engineering, Addis Ababa Institute of Technology, Addis Ababa 1000, Ethiopia; bedilu.habte@aau.edu.et

³ Department of Civil and Environmental Engineering, College of Engineering and Architecture, Howard University, Washington, DC 20059, USA; girum.urgessa@howard.edu

* Correspondence: yohannes.lisanework.alemu@uni-weimar.de

Abstract

Structural reanalysis involves repeated evaluation of structural responses under iterative design changes. It is a major computational burden in structural optimization, sensitivity analysis, and health monitoring. The three-layer architecture, which comprises the stiffness, displacement, and force layers, is motivated by the governing structural mechanics relationship $F = K \cdot U$, which establishes stiffness and displacement as natural intermediate quantities for predicting internal forces. This physics-informed hierarchy reduces dependence on large training datasets while preserving predictive accuracy across all response quantities. The framework predicts member-level stiffness statistics, nodal displacements, and internal forces through three sequential layers: stiffness, displacement, and force. Each layer enriches the feature set of the layer above. Sensitivity-based secondary inputs are derived analytically from closed-form expressions relating cross-sectional dimensions to stiffness and displacement changes. This embeds structural mechanics knowledge directly into the feature engineering process without additional analyses. Member stiffness matrices are recovered as submatrices of the global stiffness matrix, encoding inter-member structural context into each member's representation. The framework is implemented on a six-floor, three-bay reinforced concrete frame of 42 members. Training uses 1890 data points from 45 finite element iterations. The Random Forest model achieves R^2 scores of 0.99, 0.98, and 0.91 for axial force, shear force, and bending moment, respectively, on unseen validation data. Once trained on 45 FE iterations, the framework evaluates any number of candidate cross-sectional configurations in a single batch inference pass, enabling a shift from sequential solver-driven reanalysis to model-driven batch optimization. The proposed framework offers a scalable, interpretable, and physics-consistent alternative to both classical reanalysis methods and purely data-driven models, with direct applicability to structural size optimization and structural health monitoring workflows.

Keywords: three-layer framework of models; data driven machine learning; reanalysis of structures; reinforced concrete structures; structural optimization; structural health monitoring; damage identification; model interpretability



Academic Editor: Muhammad Junaid Munir

Received: 1 April 2026

Revised: 10 May 2026

Accepted: 14 May 2026

Published: 17 May 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Structures undergo repeated analysis under iterative changes to loading, geometry, material properties, boundary conditions, connectivity, and other factors. Such repeated evaluation is essential in structural optimization, sensitivity analysis, damage identification, and structural health monitoring. Complete reanalysis using finite element software involves solving large systems of equations at every iteration. This is computationally expensive and hinders the automation of design workflows.

Several reanalysis methods have been developed to reduce this cost. They are largely based on the Sherman–Morrison and Woodbury matrix inversion update formulae. Linear and nonlinear formulations of these approaches are reviewed in [1]. They are classified as direct (exact) and iterative (approximate). Direct methods provide exact solutions but are restricted to small-scale structural changes affecting only a few members. Eigenvalue reanalysis approaches have also been reviewed and compared [2], where the Padé approximation method and the Kirsch combined method demonstrated superior performance.

For large or global structural changes, approximation methods are more applicable. The combined approximations reanalysis method (CARM) combines the efficiency of local approximations with the accuracy of global approaches [3,4]. However, CARM still requires solving reduced equilibrium equations. The number of basis vectors depends on the step size of structural changes. A larger step size demands more basis vectors. This either forces a variable number of basis vectors per iteration, or requires a fixed step size throughout. This is a significant constraint in size optimization, where depth-to-width proportionality varies freely across iterations. Furthermore, CARM requires explicit formulation and partial inversion of the assembled stiffness matrix through the computation of δK , adding further implementation complexity. Recent model-based reanalysis approaches have also been explored to reduce this cost, though they still rely on repeated reduced-order solves during the model construction phase [5].

Machine learning takes a different approach. ML models rely on pattern recognition rather than equation solving and do not require explicit matrix formulation or inversion. Several ML-based approaches have been explored for structural reanalysis. Jenkins [6] introduced a neural network iterative method for direct reanalysis of framed structures, incorporating force and displacement controls to handle changes in loading, supports, and member connectivity. Nodal equilibrium served as the convergence criterion, with displacements and forces within an acceptable error range taken as reanalysis outputs [7]. Ahmad et al. [8] developed one- and two-hidden-layer artificial neural networks to predict load-carrying capacity of RC members from cross-sectional and material strength inputs, achieving good prediction accuracy. However, these methods remain partly computation-dependent and in several cases require more time than direct FEM computation. Tandale et al. [9] trained neural networks to compute element stiffness matrices via displacement-to-force derivations, but required 80,000 data points for even simple truss structures.

Data-driven ML models address these limitations directly. They are independent of equilibrium equation computation entirely. Once trained, they produce predictions through direct inference. Critically, a single trained model can simultaneously process any desired number of design iterations in one pass. Unlike sequential computational reanalysis, which evaluates one iteration at a time, this approach scales efficiently with the number of required reanalyses. In structural size optimization workflows, frame reanalysis may be required across thousands or tens of thousands of cross-sectional size iterations, making this difference practically significant [10].

Existing ML-based reanalysis approaches share a common limitation: they treat structural response prediction as a single-step mapping from input parameters to output quantities, without leveraging the hierarchical physical relationships that govern structural

behavior. This overlooks the well-established dependency chain in which cross-sectional geometry determines member stiffness, stiffness governs nodal deformation, and deformation drives internal forces. Bypassing this chain forces the model to learn long-range input–output mappings from data alone, increasing data requirements and reducing physical interpretability. The framework proposed in this work deals with this gap by following the physical sequence from stiffness to displacement to force.

This paper presents a three-layer data-driven ML framework for the reanalysis of reinforced concrete framed structures. The framework sequentially predicts member stiffness statistics, nodal displacements, and internal forces through a hierarchically organized stiffness, displacement, and force layer. Each lower layer enriches the feature set of the layer above. The derivation of sensitivity-based secondary input features from closed-form expressions relating cross-sectional dimensions to stiffness and displacement changes is an important contribution. Member stiffness matrices are recovered as sub-matrices of the global stiffness matrix using nodal degree-of-freedom indices, encoding inter-member structural context into each member’s feature representation. Model interpretability is examined through feature importance rank, Spearman’s rank, and mutual information analysis across all three layers. The framework is validated on a six-floor, three-bay RC building frame using 1890 data points from 45 finite element iterations.

A notable operational characteristic of the proposed framework is that it reduces the dependence of the optimization algorithm on the structural solver. In conventional reanalysis, each candidate configuration requires a FEM solve, coupling the two processes tightly. Once trained, the proposed framework can process multiple candidate configurations in a single inference pass, at low marginal cost. This may facilitate the use of population-based optimization strategies that can be computationally demanding under sequential FEM reanalysis.

The contributions of this work are as follows:

- A three-layer ML framework structured according to the $F = K \cdot U$ mechanics relationship, enabling hierarchical feature enrichment without additional structural analyses;
- Sensitivity-based secondary input features derived analytically from closed-form stiffness and displacement expressions, embedding domain knowledge directly into the feature engineering process;
- A quantitative demonstration that batch ML inference reduces total reanalysis cost relative to sequential FEM beyond a break-even of approximately $N \approx 10$ configurations, with speedups reaching one to two orders of magnitude at $N = 10^4$ – 10^5 , enabling model-driven optimization workflows where tens of thousands of candidate configurations may require evaluation;
- A member-level stiffness representation extracted as sub-matrices of the global stiffness matrix, encoding inter-member structural context into each member’s feature vector;
- A multi-method interpretability analysis using feature importance rank, Spearman correlation, and mutual information, which reveals the collective discriminative role of nodal coordinate features across all three layers.

2. Machine Learning Models for Structural Reanalysis

Machine learning has seen growing application in structural engineering, spanning structural health monitoring [11], damage detection, and design optimization [12,13]. Its integration into reanalysis workflows is motivated by the ability to replace equation-solving with pattern recognition, enabling predictions without explicit matrix formulation [14]. Several regression models are evaluated in this work for their suitability to the reanalysis task. Their selection is guided by three criteria: the ability to handle nonlinear input–output relationships, robustness under limited training data, and computational efficiency during

inference. Physics-informed approaches that embed structural mechanics knowledge into the learning process have also demonstrated strong performance for real-time structural response prediction [15]. The candidate models are Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting (GB), Extreme Gradient Boosting (XGB), and Neural Networks (NN).

SVM regression constructs a decision boundary in a transformed feature space using a kernel function [16,17], making it suitable for the nonlinear relationship between cross-sectional properties and internal forces [18]. However, SVM is comparatively slower to train as dataset size increases, limiting its scalability in large reanalysis workflows. Ensemble methods combine multiple weak learners to improve prediction quality and generalization [19]. RF applies bagging over an ensemble of decision trees trained on random feature subsets [12], reducing overfitting and improving generalization under limited data. GB arranges trees sequentially so that each tree corrects the residual errors of the previous one [20], improving accuracy on complex nonlinear mappings. XGB extends GB with L_1 and L_2 regularization terms that penalize model complexity and reduce overfitting [21]; comprehensive reviews of ensemble methods are provided in [19,22]. NN consists of stacked layers of weights updated to minimize a defined loss function [23], with demonstrated application in structural health monitoring and response prediction [13,24]. A single hidden layer may suffice to capture the nonlinear relationships present in the reanalysis task, while deeper architectures offer greater flexibility at increased data and training cost.

A comparative evaluation of all five models on the stiffness layer targets is reported in Table 4. RF achieved the highest R^2 values and is adopted for all subsequent layers, consistent with findings from recent studies on RC structural response prediction where tree-based ensembles consistently outperform other regression algorithms under limited data conditions [25–27].

3. The Proposed Reanalysis Framework of Machine Learning Models

The objective of the reanalysis framework is to predict internal responses such as axial forces, bending moments, and shear forces of structural members. Stiffness and displacement are chosen as fundamental features based on domain knowledge, directly reflecting the governing structural mechanics relationship. For a given function f :

$$\text{internal force} = f(\text{stiffness, displacement}). \quad (1)$$

The product of member stiffness and nodal displacement governs nodal force. Though internal forces are computed from the developed strain, they remain a function of member stiffness and end-node displacements, as expressed in Equation (1). Accordingly, a three-layer ML model framework consisting of a stiffness layer, a displacement layer, and a force layer is devised. The features considered at each layer are either direct results of the variables in Equation (1) or their sensitivity-based augmentations.

3.1. Framework Integration

The general integration of the proposed framework into both iterative and non-iterative reanalysis workflows is shown in Figure 1. In conventional FEM-based workflows, each change to a cross-sectional design variable triggers a full system solution, making the total computational cost proportional to the number of design iterations. The proposed framework decouples training cost from deployment cost entirely: once trained on a modest number of FEM iterations, it produces predictions for any number of new cross-sectional configurations in a single inference pass, at a cost that is independent of the number of reanalyses required. This makes the framework particularly well-suited to

size optimization workflows, where tens of thousands of candidate configurations may need to be evaluated. Conventional FEM-based reanalysis evaluates one iteration at a time, requiring a full or reduced equation to be solved at each step. The proposed framework replaces this with direct inference. Iterative changes in cross-sectional input variables lead to iterative changes in structural response. In iterative mode, the framework tracks these changes sequentially. In non-iterative mode, predictions for any desired set of design configurations can be generated simultaneously in a single inference pass, without following a parameter change path. This significantly reduces computational cost in optimization workflows where reanalysis may be required across thousands of candidate configurations.

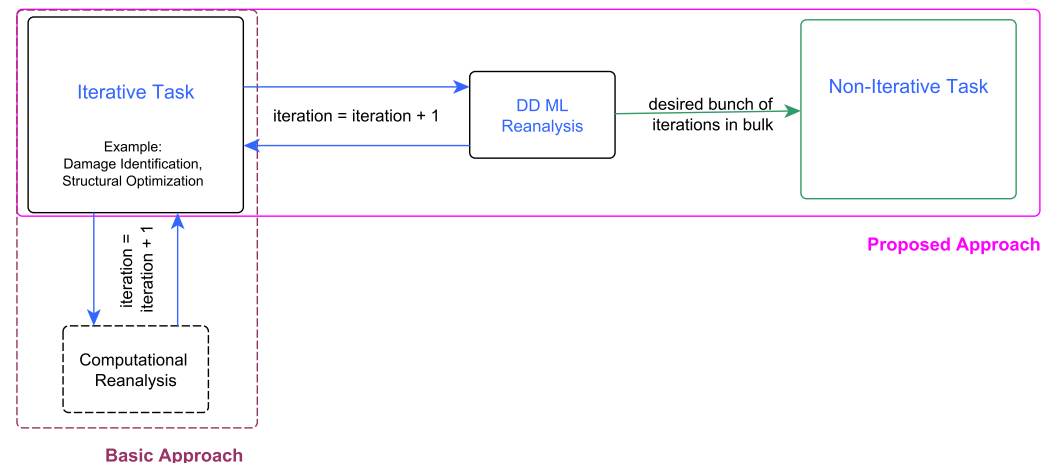


Figure 1. Generic integration of the machine learning reanalysis framework into iterative and non-iterative tasks.

In the present implementation, the framework is trained once on 45 finite element reanalysis iterations of the six-floor, three-bay frame and then deployed in the non-iterative branch of Figure 1 to generate predictions for any desired set of cross-sectional configurations in a single batch inference pass. This non-iterative path replaces sequential solver-driven reanalysis with model-driven batch optimization, decoupling deployment cost from the number of required reanalyses and making large design studies computationally tractable.

3.2. Layer-Wise Architecture

The architecture of the framework is shown in Figure 2. At the stiffness layer, statistical descriptors of member stiffness matrices are predicted from the primary parametric inputs driving the reanalysis task. Representing each member stiffness matrix through its statistical moments, specifically the mean and standard deviation of the diagonal and lower triangular entries, serves two purposes. First, it reduces the dimensionality of the target vector from the full $n \times n$ matrix to four scalars per member, making the prediction task tractable under limited training data. Second, the diagonal entries encode member axial and flexural stiffness, while the off-diagonal entries capture coupling between degrees of freedom; their first two statistical moments therefore retain the distributional characteristics most relevant to predicting downstream displacement and force responses. At the displacement and force layers, nodal displacements and internal forces are predicted directly.

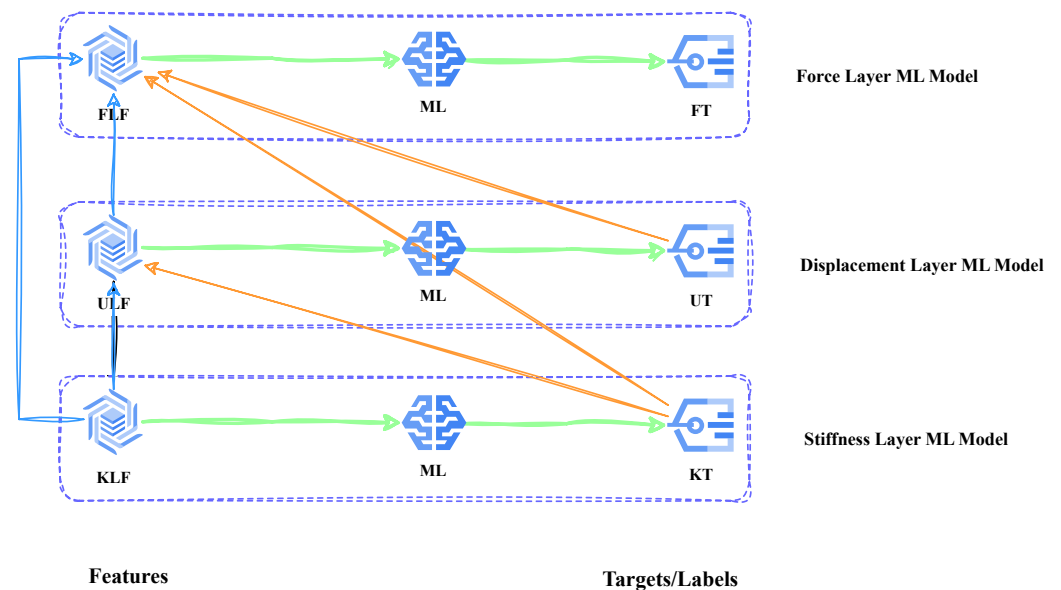


Figure 2. A three-layer framework of machine learning models with stiffness, displacement and force layer features (K_{LF} , U_{LF} , F_{LF}) and their respective targets (K_T , U_T , F_T). The green arrows represent input and output for the machine learning models; the blue arrows show input features from lower layers used on upper layers; the orange arrows show output features from lower layers used on upper layers.

3.3. Sequential Feature Enrichment

A distinguishing property of the framework is that each layer receives progressively richer features as one advances toward the force layer. Lower-layer predictions are propagated upward and combined with the layer's own features, so the force layer leverages the complete set of stiffness, displacement, and sensitivity features simultaneously. This increases the likelihood of higher prediction accuracy at the most critical output layer.

During training, upper layers receive ground-truth inputs from lower layers; during deployment, they receive predicted values that carry accumulated error from upstream. This distributional shift is mitigated by two properties of the framework: lower-layer outputs serve only as intermediate inputs, making the force layer robust to moderate upstream error provided the correct trend is captured; and sensitivity-based secondary inputs are computed independently of layer predictions, providing a stabilizing signal at each layer.

4. Features and Labels

4.1. Features

The stiffness of a structure is directly influenced by the material properties, degrees of freedom, geometry, and connectivity of its members. Nodal coordinate information further identifies inter-member dependencies and enables consistent member identification across iterative dataset generations. These features are considered fundamental and appear in all three layers of the framework. Where material properties are uniform across all iterations, they need not be included as features.

Table 1 presents the feature categories and their corresponding features across the three layers. Input features are classified as primary and secondary, where secondary inputs capture the sensitivity of target variables to changes in the primary inputs at each layer.

Sensitivity features with respect to area and moment of inertia introduce inputs that capture the patterns arising from changes in cross-sectional variables. An important advantage of this design is that even approximate predictions at the lower layers can serve

as effective inputs for the upper layer, provided they capture the overall trend. Since stiffness and displacement layer outputs are used only as intermediate inputs, rough but trend-consistent predictions remain sufficient for accurate force layer output.

Table 1. Feature categories and selected features.

Feature Category	Features	Framework Layer
Geometry	depth, width, length and their derivatives	All layers
Stiffness	lower triangular stiffness mean, standard deviation	Stiffness
Displacement	deflections and rotations	Displacement
Sensitivities	stiffness and displacement with respect to area and moment of inertia	Stiffness and Displacement
Force	augmentation of stiffness, displacement and force members, nodal connectivity, degree of freedom,	Force
Structural information	nodal coordinates	All layers

The relative change of stiffness with respect to the primary input variables is introduced as a secondary input in the stiffness layer; similarly, displacement sensitivities are introduced at the displacement layer. Part of the collected dataset is reserved as secondary input, enabling sensitivity computation across all three layers. The stiffness, displacement, and force values in the secondary input are direct outputs of structural analyses, from which updated responses are computed following changes in the input variables.

$$K_{i+1} = K_i + \frac{dK}{dx}(x_{i+1} - x_i),$$

and

$$U_{i+1} = U_i + \frac{dU}{dx}(x_{i+1} - x_i),$$
(2)

where K and U represent stiffness and displacement respectively, and x is the derivation variable, in this case cross-sectional area A and moment of inertia I . The stiffness and displacement corresponding to a given set of cross-sections x_i constitute the secondary inputs, which are updated for a new set x_{i+1} . Recursive application of Equation (2) generates feature vectors for subsequent iterations. The sensitivity measures $\frac{dK}{dx}$ and $\frac{dU}{dx}$ are derived in the following subsections.

4.1.1. Stiffness Layer Sensitivity

In the following expressions, subscript s denotes the reference cross-section at the secondary input iteration, against which updated cross-sectional sizes are compared; thus K_s , D_s , b_s , A_s , and I_s are the stiffness, depth, width, area, and moment of inertia at the secondary input reference state. Given rectangular cross-sections of members and applying the chain rule, simplified for changes related to cross-sectional area and moment of inertia:

$$\frac{d}{dA}[(A - A_s)K_s], \quad \text{and} \quad \frac{d}{dI}[(I - I_s)K_s],$$

For a rectangular cross-section, $A = bD$, so that $D = A/b$ and $b = A/D$. Setting $K = (A - A_s)K_s$ and applying the chain rule:

$$\frac{\partial K}{\partial A} = \frac{\partial K}{\partial D} \frac{\partial D}{\partial A} + \frac{\partial K}{\partial b} \frac{\partial b}{\partial A},$$

where

$$\frac{\partial K}{\partial D} = (b - b_s)K_s, \quad \frac{\partial D}{\partial A} = \frac{1}{b}, \quad \text{for } D = \frac{A}{b},$$

$$\frac{\partial K}{\partial b} = (D - D_s)K_s, \quad \frac{\partial b}{\partial A} = \frac{1}{D}, \quad \text{for } b = \frac{A}{D}.$$

Substituting and collecting terms yields:

$$\frac{\partial K}{\partial A} = \left[2 - \frac{b_s}{b} - \frac{D_s}{D} \right] K_s. \quad (3)$$

For a rectangular cross-section, $I = bD^3/12$, so that $D = (12I/b)^{1/3}$ and $b = 12I/D^3$. Setting $K = (I - I_s)K_s$ and applying the chain rule:

$$\frac{\partial K}{\partial I} = \frac{\partial K}{\partial D} \frac{\partial D}{\partial I} + \frac{\partial K}{\partial b} \frac{\partial b}{\partial I},$$

where

$$\begin{aligned} \frac{\partial K}{\partial D} &= (bD^2 - b_s D_s^2) \frac{K_s}{4}, \quad \frac{\partial D}{\partial I} = \left(\frac{12}{b} \right)^{\frac{1}{3}} \frac{1}{3I^{\frac{2}{3}}}, \quad \text{for } D = \left(\frac{12I}{b} \right)^{1/3}, \\ \frac{\partial K}{\partial b} &= (D^3 - D_s^3) \frac{K_s}{12}, \quad \frac{\partial b}{\partial I} = \frac{12}{D^3}, \quad \text{for } b = \frac{12I}{D^3}. \end{aligned}$$

Substituting and collecting terms yields:

$$\frac{\partial K}{\partial I} = \left[\frac{0.191}{b^{1/3} I^{2/3}} (bD^2 - b_s D_s^2) + \left(1 - \frac{D_s^3}{D^3} \right) \right] K_s. \quad (4)$$

where A is area and I is the moment of inertia. Equations (3) and (4) give the change in stiffness of the secondary inputs corresponding to the cross-sectional changes of each structural member. The updated stiffness values for updated depth D and width b are summed to the existing secondary input values, which then appear in the stiffness layer feature vectors.

4.1.2. Displacement Layer Sensitivity

The displacement sensitivities are derived from:

$$\frac{d}{dA} \left[\frac{U_s}{(A - A_s)} \right], \quad \text{and} \quad \frac{d}{dI} \left[\frac{U_s}{(I - I_s)} \right].$$

The total displacements in the secondary input, accounting for both the existing value and the sensitivity contribution, are given in Equations (5) and (6).

$$\frac{\partial U}{\partial A} = \left[-2 + \frac{b_s}{b} + \frac{D_s}{D} \right] U_s, \quad (5)$$

and

$$\frac{\partial U}{\partial I} = \left[-4 + \frac{3b_s D_s^2}{b D^2} + \frac{D_s^3}{D^3} \right] U_s. \quad (6)$$

For the ML model, reasonable approximations suffice, as the goal is to capture trends rather than exact outputs. The $1/(A - A_s)$ and $1/(I - I_s)$ denominators become numerically unstable as cross-sectional sizes approach their secondary input reference values, producing arbitrarily large sensitivity estimates that would introduce outliers into the feature space. Since the goal of the sensitivity features is to capture directional trends in stiffness and displacement change rather than exact magnitudes, these denominators are excluded. The remaining terms preserve the sign and relative scale of the sensitivity, which is sufficient for the ML model to leverage the feature effectively.

4.1.3. Force Layer Sensitivity

At the force layer, predictions from both lower layers, which have already accounted for their respective sensitivities, are augmented with the force values of the secondary inputs:

$$P_{i+1} = (K_{i+1} \times U_{i+1}) \times P_i. \quad (7)$$

Equation (7) is not a physically exact expression for force updating; rather, it constructs a composite feature that combines predicted stiffness and displacement trends with a reference force from the secondary input. The structural physics is encapsulated in P_i , which is a direct output of finite element analysis. The product $K_{i+1} \times U_{i+1}$ acts as a scaling signal that reflects how the force is expected to change relative to the reference state, while P_i anchors the prediction to a physically meaningful magnitude. This formulation is intentionally approximate: it provides the force layer with a physics-informed trend signal without requiring an additional structural solve.

4.2. Labels

Multiple label configurations were tested and evaluated. Statistical descriptors, mean and standard deviation, are used as labels for the stiffness layer, applied to both the diagonal and lower triangular stiffness matrices. For the displacement and force layers, output quantities are used directly as target labels: translations and rotations at the displacement layer, and axial force, shear force, and bending moment at the force layer.

The target labels in Table 2 are used across the three layers of the framework. Since lower-layer predictions serve primarily as inputs to the upper layers, alternative targets such as eigenvectors could also be explored in future work. Target transformations are applied to improve numerical conditioning during training. Stiffness matrix entries span several orders of magnitude; applying a logarithm of square root transformation compresses this range and reduces the influence of high-magnitude entries on the loss function, preventing the model from disproportionately optimizing for large stiffness values. Displacement values, by contrast, are small in magnitude and tightly clustered; a sigmoid transformation rescales them to the (0, 1) interval, amplifying inter-sample variation and improving gradient signal during training. No transformation is applied to force layer targets, as internal force values are well-distributed across the training dataset, as shown in Figure 10.

Table 2. Target label variables for each layer in the proposed framework.

Framework Layer	Target Labels	No. of Targets	Transformation
Stiffness layer	Diagonal stiffness matrix: mean	4	logarithm of square root
	Diagonal stiffness matrix: standard deviation		
	Lower triangular stiffness matrix: mean		
	Lower triangular stiffness matrix: standard deviation		
Displacement layer	Translations	6	sigmoid
	Rotations		
Force layer	Internal forces	3	—

Since each layer predicts multiple target columns simultaneously from a single feature table, the framework significantly reduces the required number of feature tables. Without this design, 13 separate feature tables would be needed to accommodate all 13 target variables. The hierarchical framework reduces this to 3.

4.3. Feature Tables

A core contribution of this work is the generation of layer-wise feature tables directly from a small set of primary variables, without requiring structural analysis software during deployment. Unlike conventional approaches, the feature tables used as input for prediction are themselves built sequentially from framework predictions.

Regarding member type, beams and columns may be modeled separately or together. From a domain knowledge perspective, separate treatment is reasonable: moment and shear govern beam behavior, while axial force and moment dominate in columns under limited lateral loading, and their cross-sectional geometries may differ substantially. However, training separate models risks dataset scarcity, particularly when the total number of iterations is limited. Training a unified model increases dataset size and may improve generalization by exposing the model to the combined variability of both member types. The implications of this choice are examined in Section 5.2.

A 2D matrix representation where rows correspond to structural members at each iteration and columns correspond to features is used. In this representation, instances of the same member recur after every n rows, where n is the total number of structural members.

The stiffness layer feature table is shown in Figure 3a. Secondary inputs are computed using Equations (3) and (4). The displacement layer feature table in Figure 3b incorporates stiffness layer targets as inputs, with secondary inputs computed using Equations (5) and (6). The force layer feature table in Figure 3c includes targets from both lower layers, with force sensitivity augmentation computed using Equation (7). As the number of iterations used as secondary inputs n_i increases, the feature table sizes across all three layers grow accordingly.

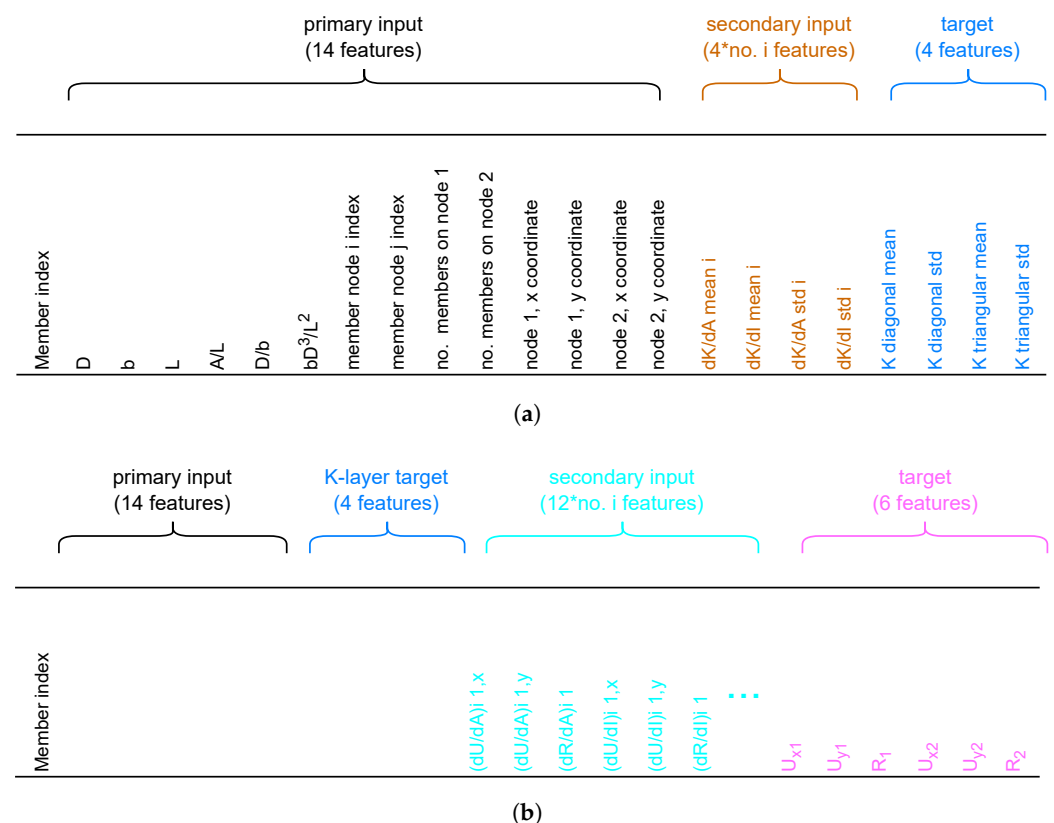


Figure 3. Cont.

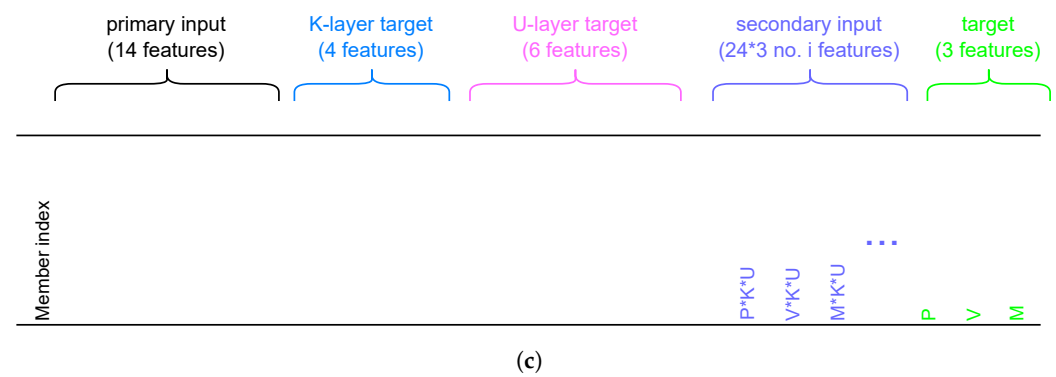


Figure 3. Feature tables for each layer of the proposed framework. (a) Feature table for the stiffness layer. (b) Feature table for the displacement layer. (c) Feature table for the force layer.

4.4. General Procedure

The general application procedure is summarized in Table 3.

Table 3. Summarized procedure to implement the three-layer framework.

No.	Main Activity	Activity	Sub-Activity
1	Data Collection	FEM-based structural simulation and analysis for a selected number of iterations.	The main variables changing in each iteration are cross-sectional sizes. Step size and cross-sectional limits for each member type (beam or column) shall be defined. Sampling approach: a mix of random and uniform distribution.
		Extracting required data from the simulation and analysis.	Cross-section, member length, stiffness matrix, nodal coordinates, nodal and member connectivity, degree of freedom, deformation results, and internal force results are extracted.
2	Data Preparation	Preparing the extracted data in a data frame format.	Extracting member stiffness from the global stiffness matrix using connectivity and degree-of-freedom information. Nodal coordinates, connectivity, geometry, deformation, and force for each member are retained.
3	Feature Tables and ML Model Training	Splitting into primary and secondary inputs.	Around 80% of data used as primary input computation. Equations (3)–(6) applied to compute secondary inputs.
		Setting targets for each framework layer.	Targets follow Figure 3.
		Preferred regression model trained sequentially along the layers, with results evaluated.	Appropriate evaluation metrics applied.
4	Framework Deployment for Reanalysis Prediction	For a given set of cross-sections for members of the same structure: input data provided and predictions generated.	Primary input updated with new cross-sectional sizes. Secondary inputs for the first two layers updated based on current cross-sectional variations. Stiffness layer targets predicted. Displacement layer targets predicted using stiffness layer outputs. Force layer secondary input updated and force targets predicted.

5. Application of the Framework on a 2D RC Building Frame

The proposed three-layer DD ML prediction framework is implemented on a 2D RC building frame. The task is to predict member-level internal responses using machine learning models trained on hand-crafted features.

5.1. Dataset

The frame shown in Figure 4 consists of 42 members: 18 beams and 24 columns. Loading information is adopted from the 2D case study in [28]. The structure is modeled and analyzed in SAP2000 v24 using the finite element method. Member section properties, geometry, connectivity, degree-of-freedom indices, joint coordinates, stiffness matrices, displacements, and internal forces at specified points are extracted from the software output tables.

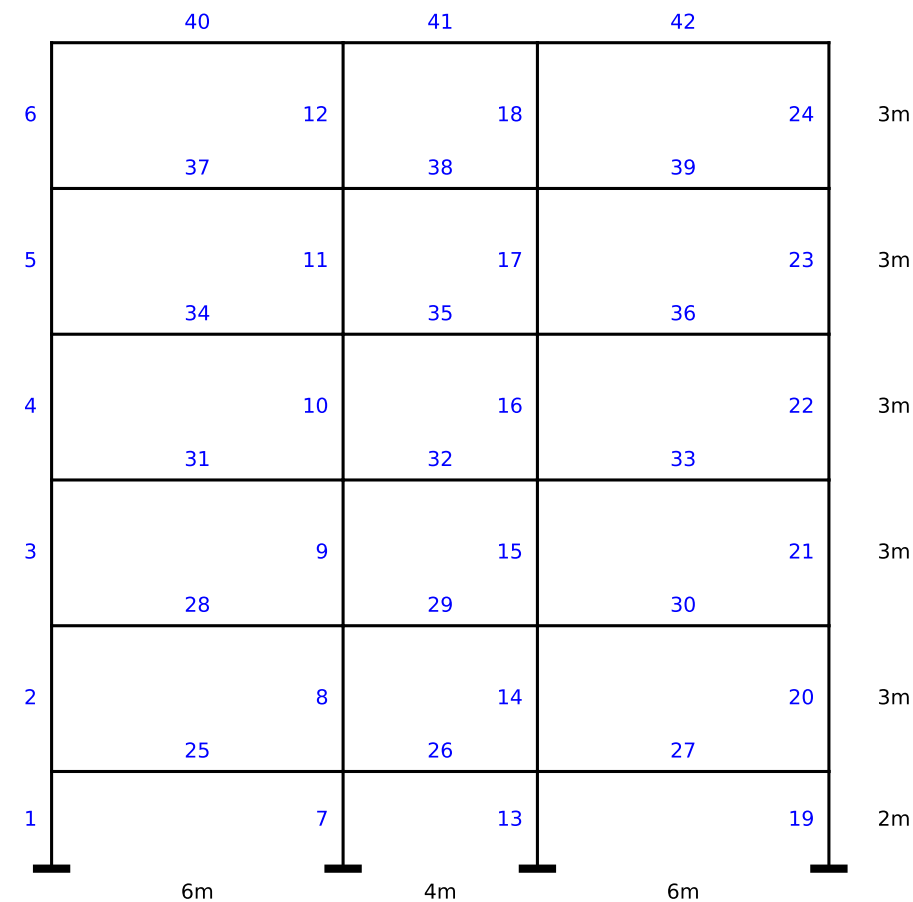


Figure 4. Considered two-dimensional reinforced concrete frame.

The frame considered in Figure 4 is fixed at the base and the beam column connection is semi-rigid. A design load of 17.85 kN/m permanent load and 3 kN/m live load were applied based on the loading in [28] Table 1.

A total of 1890 data points are collected from 45 iterations. The relatively modest dataset size is made tractable by the physics-informed feature engineering adopted in this framework. Sensitivity-based secondary inputs, derived from closed-form expressions relating cross-sectional dimensions to stiffness and displacement changes, embed structural mechanics knowledge directly into the feature representation. This reduces the model’s reliance on large volumes of data to infer underlying physical relationships empirically, distinguishing the proposed approach from purely data-driven methods that typically require significantly larger training sets. The number of iterations was incrementally

increased until satisfactory model performance was achieved. The dataset is split into three groups: 37 iterations for primary input, 4 iterations for secondary input computation, and the remaining 4 for deployment predictions and model evaluation on unseen data.

The present numerical experiments are conducted on a single six-floor, three-bay RC frame, which serves as a demonstrator for the proposed pipeline. In envisaged applications, the framework is not intended to generalize from this trained model to arbitrary new structures. Instead, a separate dataset would be collected and the three-layer pipeline retrained for each new structural configuration of interest. The current study therefore evaluates the methodology at the level of a single representative frame, rather than claiming cross-structure generalization.

Model accuracy generally improves as the number of iterations increases. The required threshold varies with structural type, topology, geometry, and material composition. However, this dependency may be moderated by the fact that data from each structural member is generated across multiple iterations, ensuring balanced member representation. The total number of members also influences the dataset size required.

The step size, defined as the rate of change of each design variable per iteration, directly affects dataset granularity. Larger step sizes reduce the number of collected data points; smaller step sizes increase computational effort. Using a random step size within defined bounds offers flexibility and may improve model robustness, but carries the risk of coarse sampling in certain regions of the design space

$$\text{bin width} = \frac{3.5 \sigma(X)}{n(X)^{1/3}}, \quad (8)$$

where σ is the standard deviation of vector X and n is its length.

The underlying distribution of the dataset is assessed using Scott's rule, which is optimal for near-normal data. The bin width, as given in Equation (8), depends on the standard deviation and length of the dataset vector. Cross-sectional sizes are randomly generated within lower and upper limits of 0.2 m and 1.0 m for both width and depth. Three sampling ranges are used: tertile subranges (0.2 to 1/3, 1/3 to 2/3, 2/3 to 1 m); two broader groups (0.2 to 2/3 and 2/3 to 1 m); and fully random generation within the global bounds. Each iteration is designed to contribute to coverage of the design space.

The four secondary input iterations are sampled from the same ranges. Figure 5 shows the depth, width, and area distributions of the input dataset, with bin widths determined by Scott's rule. A trade-off between normal and uniform distributions is targeted. The corresponding internal force distributions are shown in Figure 6. These distributions depend on member configuration, loading, boundary conditions, and the overall structural analysis. Beam and column members are displayed together. The region of higher density in the axial force distribution corresponds to beam members, while the larger, more uniformly distributed values represent columns.

Member stiffness extraction from the global stiffness matrix is performed in MATLAB R2018b. ML scripts are written in Python 3.9.18. The RF, GB, XGB, and SVM models were implemented using scikit-learn v1.2.2. The NN baseline was implemented using TensorFlow v2.10.0. All features are normalized using z-score standardization prior to training. Dataset representation uses a 2D matrix format.

The structural analysis is performed under linear elastic static conditions. Gravity loads (dead and live) are applied following the load configuration. Lateral loads are not considered; the frame is symmetric and lateral displacements are negligible. Nonlinear effects including geometric nonlinearity, material nonlinearity, P-delta effects, and plastic hinge formation are outside the scope of the present work. This represents a deliberate scoping decision: establishing the framework's predictive accuracy and computational

advantages under linear elastic conditions provides a necessary and reproducible baseline before extension to nonlinear behaviour, which is the subject of ongoing work. The framework architecture is not inherently restricted to linear analysis; substituting nonlinear FEM data as the training source would extend it directly.

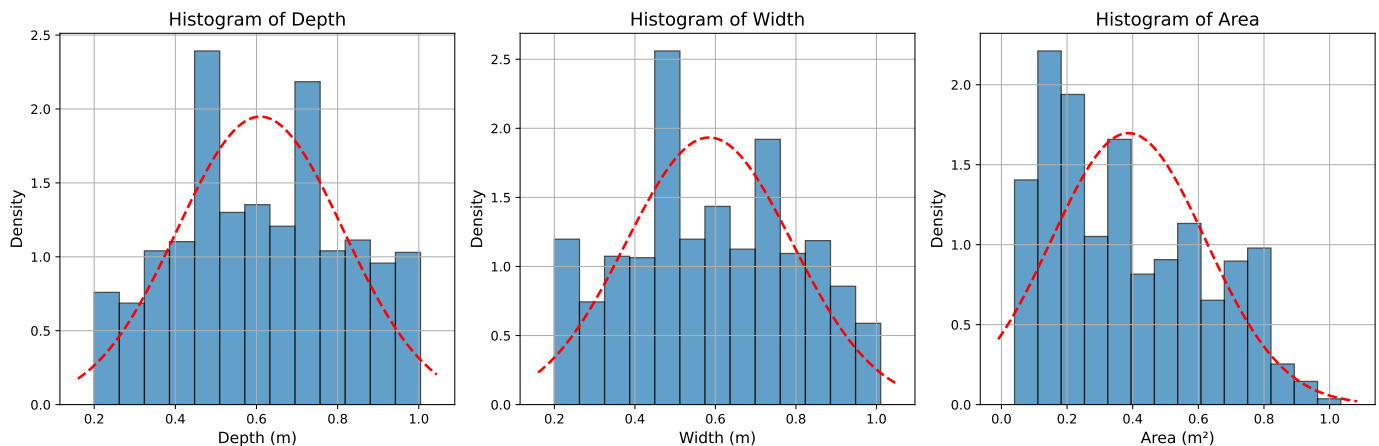


Figure 5. Member cross-sectional dataset distribution.

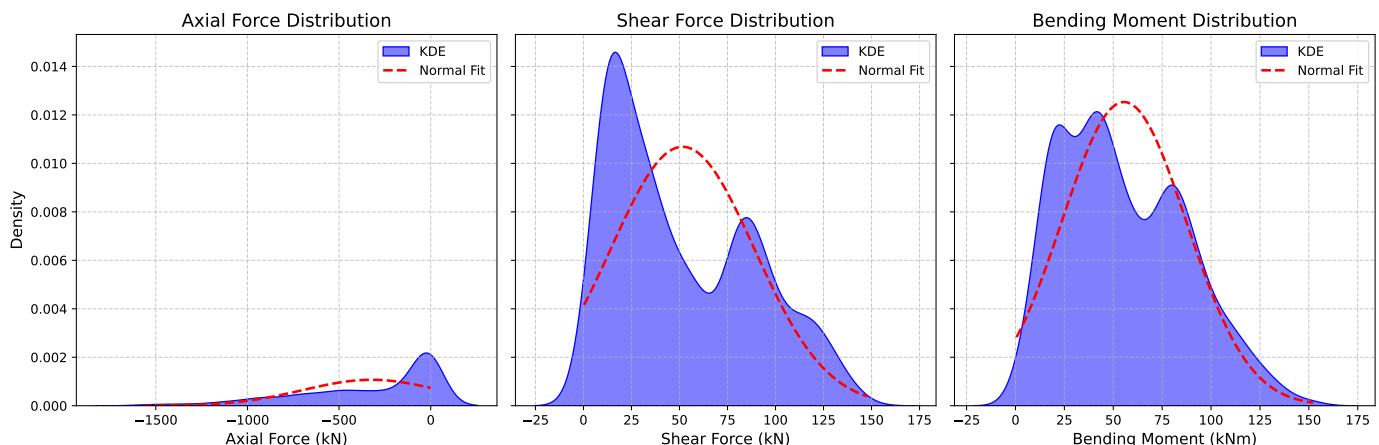


Figure 6. Internal force data distribution.

5.2. Experimentation

5.2.1. Number of Secondary Input Iterations

To determine the optimal number of secondary input iterations n_i , the stiffness layer is used as a reference. The identified optimal value is then applied uniformly to the remaining layers to reduce implementation complexity. For larger or more complex structures, an independent search for n_i at each layer may be more appropriate.

Values of n_i from 1 to 4 are evaluated in Figure 7. The configuration with $n_i = 4$ returned slightly higher performance, though the differences across all tested values are modest. A small degree of feature redundancy is nonetheless considered beneficial for model stability.

Accordingly, $n_i = 4$ is adopted for all three layers. As n_i increases, additional features are introduced and may continue to contribute positively up to a point. Beyond this, diminishing variation between primary and secondary input cross-sections reduces informational gain, and excessive feature redundancy may suppress attention toward the most informative features and impair training.

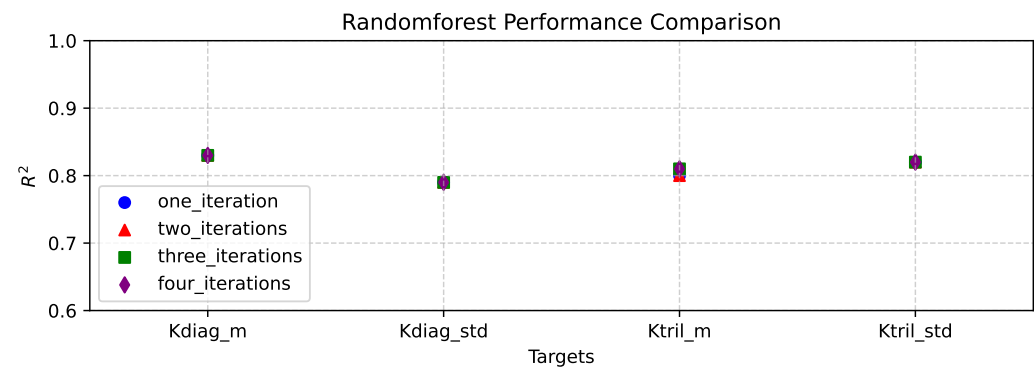


Figure 7. R^2 for Random Forest models with varying numbers of secondary input iterations in the stiffness layer.

5.2.2. Separate Member Type Consideration

Separate RF models for beams and columns are trained at the stiffness layer to assess the effect of member-type separation. In this configuration, 666 beam instances and 888 column instances are available for training.

As shown in Figure 8, performance increases from the beam-only model to the column-only model, and further improves when both member types are combined. The limited dataset size appears to restrict separate models from outperforming the combined case. Training separate models is therefore not pursued for the stiffness and displacement layers. However, given that lower-story columns resist larger axial forces than upper-story ones, a separate column model may provide some advantage at the force layer, which is examined in Section 5.3.

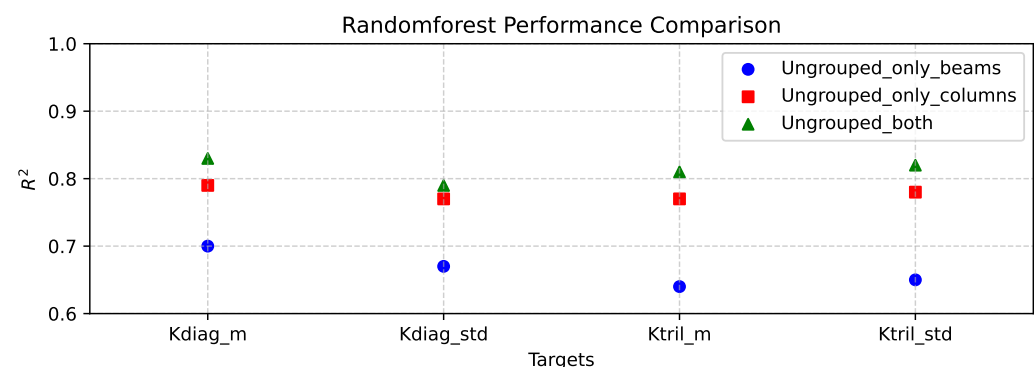


Figure 8. R^2 for beam-only, column-only, and combined cases in the stiffness layer.

5.2.3. Grouping Members by Iteration

Dataset splitting into training, validation, and test sets is performed at the iteration level rather than the individual data point level. All 42 members belonging to a given iteration are assigned together to a single split partition, preserving structural coherence across splits.

While grouped splitting ensures proportional distribution of member configurations across partitions, results on unseen deployment data indicate that the ungrouped approach yields better generalization. This is attributed to the greater diversity of member configurations seen during training under unconstrained splitting. When members from the same iteration are kept together, the training set contains fewer distinct cross-sectional combinations, reducing the model's exposure to the full variability of the design space. Ungrouped splitting distributes individual member instances more evenly across partitions, improving generalization to unseen cross-sectional configurations at deployment. This comes at the cost of mild structural coherence within each partition, which is acceptable

given that the model operates at the member level rather than the system level. As shown in Figure 9, the ungrouped RF model achieves higher R^2 on unseen data. The ungrouped approach is therefore adopted for all three layers.

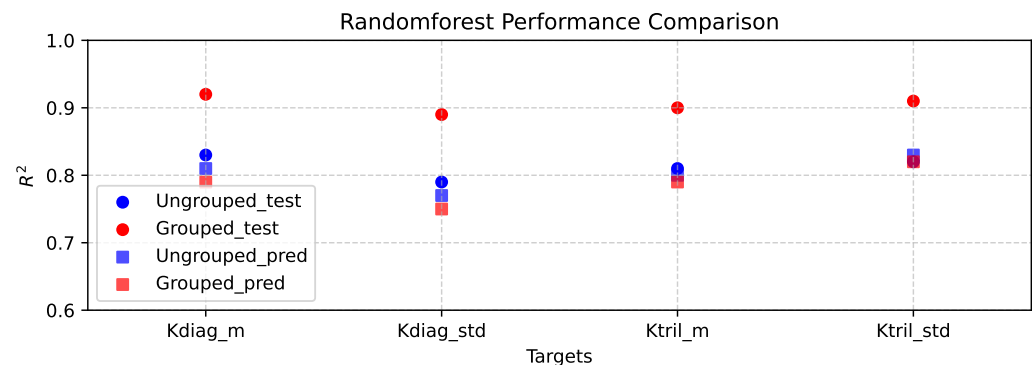


Figure 9. R^2 for grouped and ungrouped splitting in the stiffness layer.

5.2.4. Regression Algorithm Selection

SVM, RF, GB, XGB, and NN models are evaluated on the stiffness layer targets (Table 4). Performance differences are modest across algorithms. SVM is comparatively slower to train. XGB introduces L1/L2 regularization but does not show meaningful improvement over GB. NN results are comparable with greater architectural flexibility.

Table 4. Main hyperparameters of the models.

Model	Key Hyperparameters
Random Forest (RF)	$n_{\text{estimators}} = 200$; max_depth = 10; min_samples_leaf = 2; max_features = 'sqrt'; 5-fold CV.
Gradient Boosting (GB)	$n_{\text{estimators}} = 200$; max_depth = 10; learning_rate = 0.1; 5-fold CV.
XGBoost (XGB)	$n_{\text{estimators}} = 200$; max_depth = 10; learning_rate = 0.1; 5-fold CV.
Support Vector Regression (SVR)	kernel = rbf; $\epsilon = 0.1$; 5-fold CV.
Neural Network (NN)	Dense(n_{features})–Dense(20)–Dense(n_{outputs}), ReLU/linear; dropout = 0.20; L1/L2 regularization; Adam (learning_rate = 0.015); epochs = 200; batch_size = 32; 5-fold CV.

Table 5 reports R^2 and RMSE for all five models across the four stiffness layer targets: mean and standard deviation of the diagonal and lower triangular stiffness matrices. All targets are evaluated after applying a logarithm of square root transformation. RF achieves the highest R^2 values and is adopted for all subsequent layers. This is consistent with findings from recent studies on RC structural response prediction, where RF and tree-based ensembles consistently outperform other regression algorithms under limited data conditions [25–27].

Table 5. Algorithm performance comparison on stiffness layer targets.

Model	R^2				RMSE			
	$K_{\text{diag_m}}$	$K_{\text{diag_std}}$	$K_{\text{tril_m}}$	$K_{\text{tril_std}}$	$K_{\text{diag_m}}$	$K_{\text{diag_std}}$	$K_{\text{tril_m}}$	$K_{\text{tril_std}}$
RF	0.83	0.79	0.81	0.82	0.14	0.13	0.16	0.13
GB	0.80	0.74	0.75	0.77	0.16	0.14	0.18	0.15
XGB	0.79	0.75	0.75	0.78	0.16	0.14	0.18	0.15
SVM	0.83	0.78	0.80	0.81	0.14	0.13	0.16	0.14
NN	0.75	0.66	0.73	0.73	0.16	0.14	0.17	0.12

5.2.5. End-to-End Baseline Comparison

Table 6 compares the proposed three-layer framework against a direct end-to-end Random Forest baseline, in which internal forces are predicted in a single step from the primary input features and secondary input features of the stiffness and displacement layers, bypassing the intermediate stiffness and displacement layers. The three-layer framework achieves substantially lower RMSE for axial force on both the test set (25.25 vs. 69.24) and the unseen deployment set (23.94 vs. 64.58), confirming that the hierarchical feature enrichment which propagates predicted stiffness and displacement together with their sensitivity-based secondary input features as inputs to the force layer provides a meaningful predictive advantage over a flat mapping. Improvements for shear force and bending moment are more modest, as these quantities are less sensitive to the axial force distribution that challenges the end-to-end model. The consistency between test and unseen performance across both models suggests that the accuracy gains are structural rather than artefacts of the train–test split.

Table 6. Proposed framework performance comparison to an end-to-end baseline.

Model	Stage	Metrics	Axial Force	Shear Force	Bending Moment
Three-layer framework	Test	RMSE	25.25	5.65	9
		R^2	0.995	0.977	0.918
	Unseen	RMSE	23.94	5.4	9.17
		R^2	0.99	0.98	0.91
End-to-end baseline	Test	RMSE	69.24	6.81	10.15
		R^2	0.965	0.966	0.895
	Unseen	RMSE	64.58	6.28	10.62
		R^2	0.965	0.97	0.885

5.3. Results and Discussion

Results for the displacement and force layers are based on Random Forest models. As the frame is symmetric and lateral loads are not considered, lateral displacements are negligible and excluded. Only vertical displacements and rotations are included in the displacement layer. At the force layer, shear force and bending moment are the primary responses of interest for beam members under gravity loading; for columns, all three internal force components, axial force, shear force, and bending moment, are considered.

5.3.1. Layer-Wise Model Performance

Table 7 summarizes model performance across all three layers. All targets achieve $R^2 > 0.79$, with the force layer achieving R^2 values of 0.99, 0.98, and 0.91 for axial force, shear force, and bending moment, respectively. The elevated RMSE for axial force is attributable to the nature of the structural configuration: lower-story columns accumulate larger axial loads under gravity loading, producing high-magnitude outliers in the target distribution, as visible in Figure 6. No transformation is applied to force layer targets, so absolute errors are reported directly. Given the smaller dataset size and the randomness of cross-sectional sampling, these results are considered promising.

Table 7. Model performance metrics for each framework layer.

Metric	Stiffness Layer				Displacement Layer				Force Layer		
	K_{diag_m}	K_{diag_std}	K_{tril_m}	K_{tril_std}	U_{y1}	R_{z1}	U_{y2}	R_{z2}	P	V	M
R^2	0.83	0.79	0.81	0.82	0.98	0.92	0.86	0.90	0.99	0.98	0.91
RMSE	0.14	0.13	0.16	0.13	0.018	0.066	0.013	0.08	23.94	5.4	9.17
MAE	0.11	0.09	0.11	0.1	0.008	0.039	0.005	0.048	14.56	3.97	6.41

The relatively high R^2 alongside elevated RMSE and MAE for axial force is a consequence of the bimodal target distribution visible in Figure 10: beam members carry near-zero axial force, while lower-story columns accumulate axial loads exceeding 1000 kN under gravity loading. This wide dynamic range inflates absolute error metrics even when relative prediction accuracy is high. The R^2 value of 0.99 confirms that the model correctly captures the variance structure of the axial force distribution; the RMSE should therefore be interpreted in the context of the target range rather than as an indicator of poor model fit. The predicted values are plotted against true values for each target in Figure 10. Proximity of each data point to the fitted line indicates prediction quality; the spread of points along the line reflects the distribution of cross-sectional sizes in the tested dataset and does not indicate model error.

The stiffness layer predictions in Figure 10a show close linear alignment across all four targets, with some scatter at extreme values. This does not significantly degrade force layer performance, because outputs from the stiffness layer serve only as intermediate inputs on the force layer. Displacement layer fitting lines in Figure 10b show tighter agreement, consistent with the high R^2 values reported. Force layer fitting lines in Figure 10c confirm acceptable prediction quality across all three internal force components.

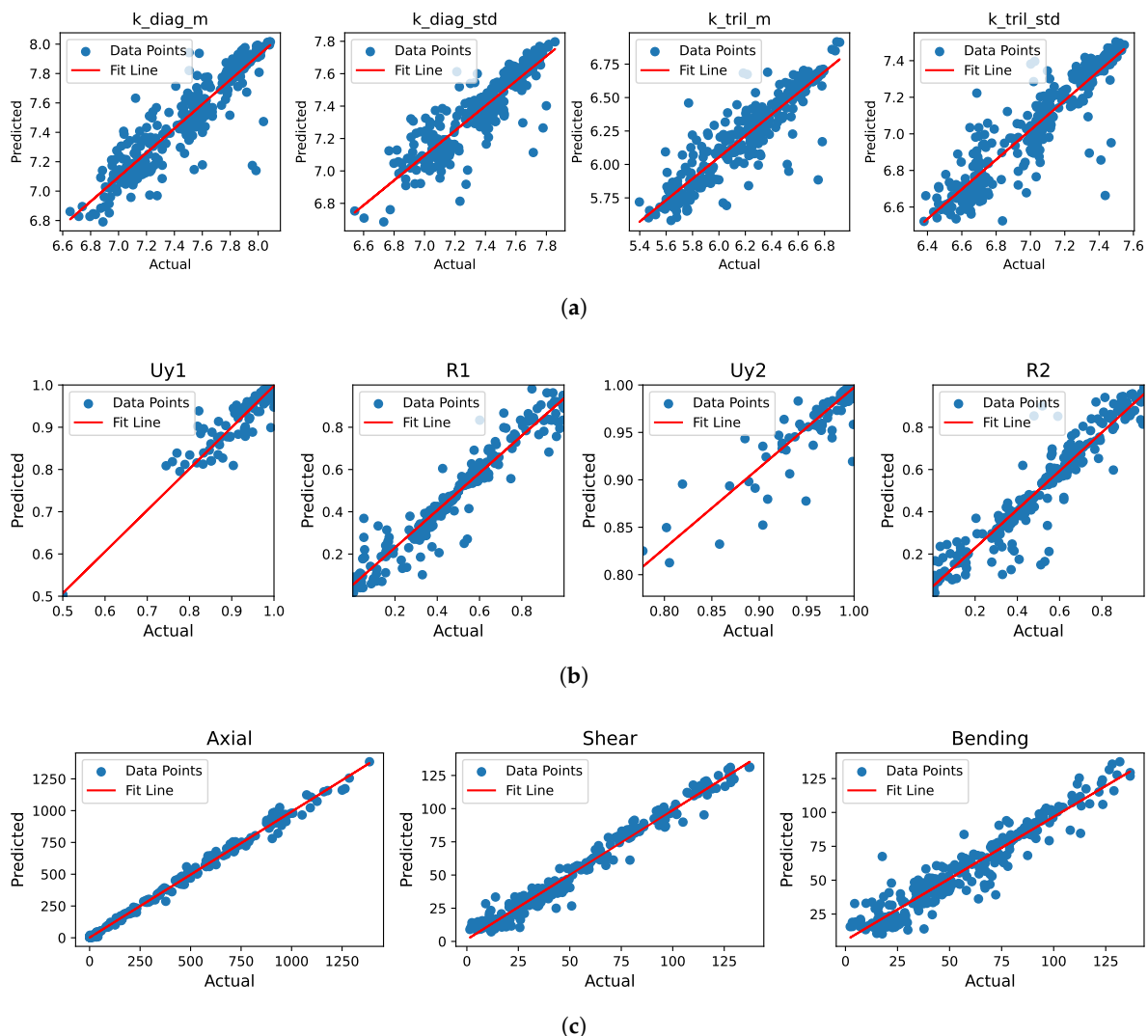


Figure 10. Predicted vs. true value fitting lines for each framework layer. (a) Stiffness layer: predicted vs. true values (logarithm of the square root transformation). (b) Displacement layer: predicted vs. true values (sigmoid transformation). (c) Force layer: predicted vs. true values.

5.3.2. Force Layer Performance by Member Type

The force layer is further evaluated using three model configurations: beam members only, column members only, and all members combined. Results are reported for both test data and four additional unseen deployment iterations in Table 8.

Table 8. Force layer performance by member type on test and unseen data.

Metric	Data	Beams Only			Columns Only			Beams and Columns		
		<i>P</i>	<i>V</i>	<i>M</i>	<i>P</i>	<i>V</i>	<i>M</i>	<i>P</i>	<i>V</i>	<i>M</i>
RMSE	Test	6.93	5.92	8.99	33.77	5.28	9.06	25.27	5.65	9.01
R^2		0.71	0.93	0.87	0.99	0.82	0.84	0.99	0.98	0.92
RMSE	Unseen	7.81	5.56	8.90	30.19	4.90	9.04	23.94	5.40	9.17
R^2		0.66	0.94	0.90	0.99	0.85	0.83	0.99	0.98	0.91

The combined model achieves the best overall performance across all three force components. The column-only model yields comparable R^2 for axial force but lower scores for shear and bending moment. The beam-only model achieves low axial force R^2 , which is expected given that axial forces are small and near-uniform in beams under gravity loading. The consistency between test and unseen data performance confirms model stability and generalization capability. Separating member types may improve individual metric quality but doubles the number of required models and is not warranted given the current dataset size.

5.3.3. Multi-Iteration Prediction on Unseen Data

The trained framework is deployed on four additional unseen iterations comprising 168 members (72 beams and 96 columns). This demonstrates the key operational advantage of the framework: predictions for multiple design iterations are produced simultaneously in a single inference pass, unlike sequential FEM-based reanalysis which processes one iteration at a time.

Figure 11 shows the box plots of true and predicted values for each internal force component. Median values and interquartile ranges of true and predicted pairs align closely across all three targets. Some conservatism is observed in the fourth quartile of shear force and bending moment predictions, consistent with the model's tendency to slightly underestimate extreme values.

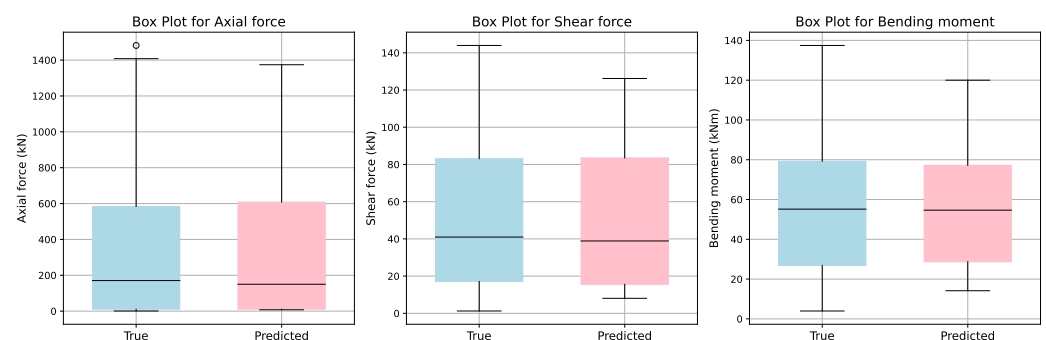


Figure 11. Box plot of true and predicted internal forces on unseen data.

Once trained, the framework generates predictions for any number of cross-sectional size variations in a single inference pass, whereas FEM-based reanalysis requires a full equation solve for each change sequentially. This gap widens considerably when frame reanalysis is required across thousands or tens of thousands of cross-sectional size iterations, as is typical in structural size optimization workflows.

5.3.4. Computational Efficiency Analysis

In addition to evaluating accuracy, we benchmarked the wall-clock inference time of the three-layer random forest (RF) model against SAP2000 linear-static analysis for the same 3-bay, 42-element RC frame on the same machine. To obtain realistic deployment timings while avoiding cross-language overhead and GUI latency, we measured only the pure inference phase for the ML framework and adopted a conservative per-frame estimate of 100 ms for SAP2000's linear static solve. This 100 ms value lies well below the 1–2 s bracket implied by the SAP2000 log timestamps for the full GUI-driven run, thereby favouring the FEM baseline in the comparison. In both cases, timings correspond to processing a single 42-row frame model, with the ML framework evaluated in batched mode for increasing numbers of candidate cross-sectional configurations.

Table 9 reports the measured wall-clock times for the three RF layers (stiffness, displacement, and force) and their sum across batch sizes from $N = 1$ to $N = 10^5$ frames. For a single frame, the total inference time is 0.27 s (≈ 265 ms), dominated by the displacement and force layers; this is already of the same order as the conservative 100 ms SAP2000 baseline. As N increases, the marginal cost per additional frame drops markedly due to efficient batched prediction, with the average per-frame cost falling from 265 ms at $N = 1$ to 31.8 ms at $N = 10$, 8.3 ms at $N = 100$, and stabilizing around 4.8–5.0 ms per frame for $N \geq 10^3$. The total processing time for 10^5 frames is 492 s, which corresponds to an average of 4.9 ms per frame for the entire three-layer pipeline.

Table 9. Wall-clock inference times of the three-layer RF framework for different numbers of frames N , compared to a conservative SAP2000 baseline of 100 ms per frame.

N Frames	t_k (s)	t_u (s)	t_p (s)	t_{total} (s)	$t_{\text{per frame}}$ (ms)	t_{SAP} (ms/frame)
10^0	0.083	0.122	0.060	0.265	265.181	100
10^1	0.097	0.142	0.079	0.318	31.824	100
10^2	0.242	0.367	0.221	0.830	8.302	100
10^3	1.052	1.862	2.142	5.056	5.056	100
10^4	10.039	19.248	19.044	48.332	4.833	100
10^5	103.610	194.264	194.523	492.397	4.924	100

Using the conservative 100 ms per-frame SAP2000 assumption, the ML framework and FEM solver are roughly comparable at very small batch sizes, but the model rapidly becomes more efficient as soon as multiple candidates are evaluated. For $N = 10$ frames, SAP2000 would require about 1.0 s, whereas the ML framework needs only 0.32 s, yielding a speedup of approximately $3\times$. At $N = 10^2$, SAP2000 would need about 10 s versus 0.83 s for the model (roughly $12\times$ faster), and at $N = 10^4$ the difference grows to 1000 s versus 48.3 s (about $21\times$ faster). For candidate configurations $N = 10^5$ —a regime typical of cross-sectional size optimization workflows—the SAP2000 baseline would require on the order of 10,000 s (≈ 2.8 h), whereas the three-layer ML framework finishes in about 492 s (≈ 8.2 min), corresponding to a speedup of roughly $20\times$ even under the conservative 100 ms FEM assumption.

The relative advantage of the model is expected to grow further for larger and more complex structures. For SAP2000 and similar FEM solvers, the cost of element formation, stiffness assembly, and equation solution generally increases at least linearly, and often superlinearly, with the number of degrees of freedom, and becomes more pronounced when geometric or material nonlinearity is included. By contrast, once trained, the three-layer RF framework performs inference with essentially linear complexity in the number of input samples, with no dependence on the underlying stiffness matrix size or the details of the constitutive model. Even if the model were extended to approximate nonlinear response quantities (for example, by retraining on nonlinear FEM simulations), the deployment cost

would remain governed by the same inexpensive tree traversals used here. As a result, the gap between classical FEM and the proposed ML framework is expected to widen markedly for larger frames, refined meshes, and nonlinear analyses, making the model particularly attractive for large-scale parametric studies and size optimization tasks where thousands to millions of candidate configurations must be evaluated.

Figure 12 visualizes the computation time between the computational software SAP2000 and the proposed machine learning reanalysis framework in milli-seconds.

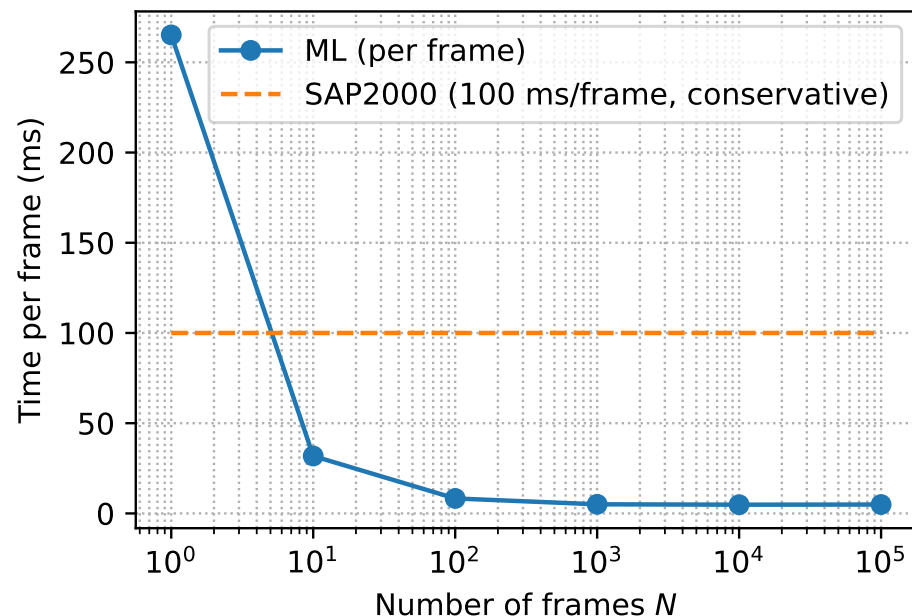


Figure 12. Box plot of true and predicted internal forces on unseen data.

5.3.5. Error Propagation Analysis

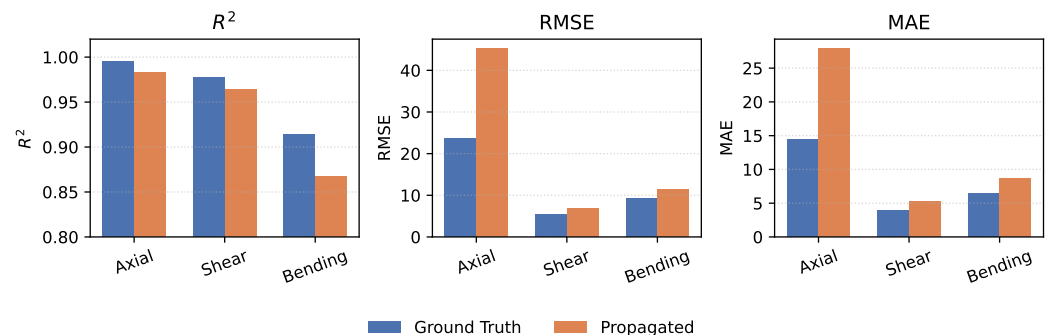
In addition to reporting the force-layer performance under ideal conditions (i.e., when the ground-truth stiffness and displacement fields from finite element analysis are used as inputs), we explicitly quantified the effect of error propagation through the three-layer model framework. In the deployed setting, the force model does not receive FEM-based K and U , but instead the predictions of the stiffness and displacement models. To emulate this realistic scenario, we replaced the ground-truth stiffness and displacement entries in the force-layer feature matrix by the corresponding predictions K_{pred} and U_{pred} obtained from the first and second layers, while keeping all other force-layer features unchanged. The force-layer regressor itself was not retrained; it was evaluated twice on the same test frames: once with ground-truth intermediate quantities, and once with their model-based predictions.

Table 10 reports the resulting metrics for each internal force component and their mean across all components. When using ground-truth stiffness and displacement, the force layer achieves a mean coefficient of determination of $R^2 = 0.9626$, with a mean root-mean-square error (RMSE) of 12.77 and mean absolute error (MAE) of 8.29 on the held-out test set. Under propagated inputs, where both stiffness and displacement are provided by the upstream models, the mean R^2 decreases to 0.9382, while the mean RMSE and MAE increase to 21.16 and 13.97, respectively. The detailed per-component results show modest deterioration for all three internal-force outputs, with R^2 drops between approximately 0.01 and 0.05 and corresponding increases in RMSE and MAE, with the third component being most sensitive to upstream prediction noise.

Table 10. Force layer performance with ground-truth vs. propagated stiffness and displacement inputs.

Internal Force	R^2_{GT}	R^2_{prop}	ΔR^2	$RMSE_{GT}$	$RMSE_{prop}$	$\Delta RMSE$	MAE_{GT}	MAE_{prop}	ΔMAE
Axial	0.9953	0.9830	−0.0123	23.7616	45.2304	21.4688	14.5088	27.8998	13.3910
Shear	0.9777	0.9641	−0.0136	5.3968	6.8458	1.4489	3.9716	5.3022	1.3306
Bending	0.9147	0.8674	−0.0473	9.1366	11.3892	2.2526	6.3899	8.7164	2.3265
Mean	0.9626	0.9382	−0.0244	12.7650	21.1551	8.3901	8.2901	13.9728	5.6827

Overall, this analysis demonstrates that chaining the three models in a fully sequential manner incurs a measurable but moderate degradation in force-prediction accuracy compared to the idealized setting with perfect intermediate fields. Crucially, the force layer retains high fidelity even in deployment mode, with all components maintaining $R^2 > 0.86$ and the mean R^2 remaining above 0.93 as shown on Figure 13 despite accumulated upstream errors. This provides a quantitative bound on error propagation and shows that the proposed layered model architecture is robust enough for practical use, while clearly distinguishing between the upper performance bound (ground-truth intermediates) and the more realistic, end-to-end performance using only model-predicted quantities.

**Figure 13.** Box plot of true and predicted internal forces on unseen data.

5.4. Model Performance After Synthetic Data Enrichment

A synthetic dataset from the existing dataset distribution was generated to examine the dataset effect. Accordingly, 2100 datapoints for 50 frame iterations (analysis) were introduced to the force layer and the output was recorded.

Based on the results presented in Table 11, the synthetic data enriched model achieves notably lower RMSE and higher R^2 on the test set relative to the model trained on collected data only, indicating improved in-distribution fit. However, performance on the unseen deployment set remains comparable between the two configurations, with differences falling within a narrow margin across all three force components. This suggests that the enrichment does not confer a generalization advantage under the present dataset conditions.

Table 11. Force layer performance of synthetic data enriched model.

Metric	Data	P	Original Data			Synthetic Data Enriched		
			V	M		P	V	M
RMSE	Test	25.25	5.65	9		15.53	3.77	6.03
R^2	Test	0.995	0.977	0.918		0.998	0.99	0.963
RMSE	Unseen	25.27	5.65	9.01		25.16	5.54	9.34
R^2	Unseen	0.99	0.98	0.91		0.995	0.977	0.911

A plausible explanation is that the 45 collected iterations, sampled across broader cross-sectional bounds, have already provided adequate coverage of the design space

relevant to this six-floor, three-bay frame. When the collected dataset sufficiently spans the cross-sectional sample space of the frame members, additional synthetic samples introduce limited new information, and the corresponding gain in unseen-set performance is marginal. The result implies that the benefit of synthetic enrichment is most pronounced when the original dataset undersamples certain regions of the design space, and diminishes as dataset coverage approaches saturation.

5.5. Interpretability

Model interpretability is examined through feature importance, Spearman's rank correlation, mutual information, and nodal coordinate contribution analysis. Pearson correlation is excluded, as it is limited to linear relationships and the problem is inherently nonlinear. For the stiffness layer, all 30 features are evaluated. For the displacement and force layers, where feature counts reach 66 and 214 respectively, only features with importance scores of 0.01 or above are retained for Spearman analysis. Across all layers, feature indices 1 to 14 represent primary inputs. Exceptions are noted in Figure 3: stiffness layer targets occupy indices 15–18 in the displacement layer, and stiffness and displacement layer targets occupy indices 15–22 in the force layer.

5.5.1. Feature Importance

As expected, depth D , width b , and the moment of inertia derived from them contribute most significantly at the stiffness layer, as shown in Figure 14a. Secondary input stiffness statistics also rank highly. Member length, connectivity, and nodal coordinates contribute less, which is consistent with their invariance across iterations.

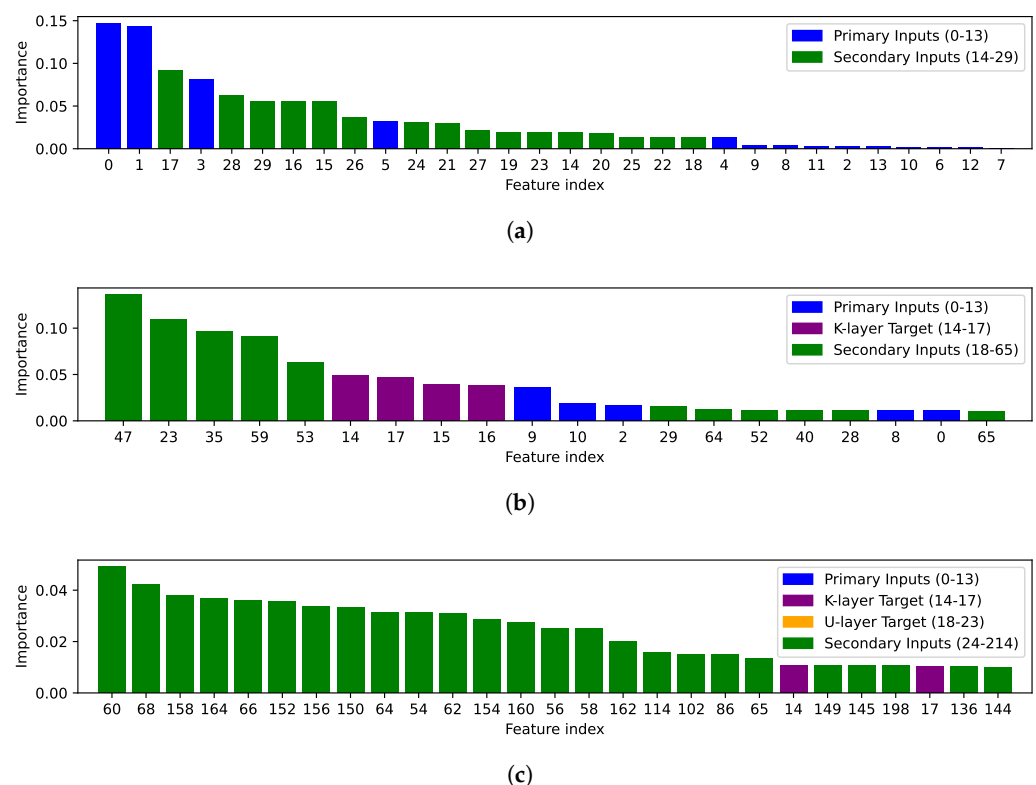


Figure 14. Feature importance scores across all framework layers. (a) Stiffness layer feature importance. (b) Displacement layer feature importance. (c) Force layer feature importance.

At the displacement and force layers (Figure 14b,c), secondary input features surpass primary inputs in importance. This reflects the improved pattern-capturing capability introduced by the sensitivity-derived features as the framework progresses upward. Al-

though individual importance scores are small, their cumulative contribution across all features remains substantial.

5.5.2. Spearman's Rank Correlation

Spearman's rank correlation coefficient is defined as:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}. \quad (9)$$

where d_i is the rank difference between corresponding observations and n is the number of observations.

At the stiffness layer, Figure 15a confirms the dominance of cross-sectional depth, width, and moment of inertia, consistent with Figure 14a. At the displacement layer, Figure 15b shows a more uniform distribution of Spearman scores among the top features. This divergence from feature importance rank is expected: feature importance accounts for variable interactions through MSE/MAE decomposition, while Spearman's correlation captures only monotonic relationships. At the force layer, Figure 15c shows the clear dominance of secondary inputs, with a negative correlation between these features and axial force predictions.

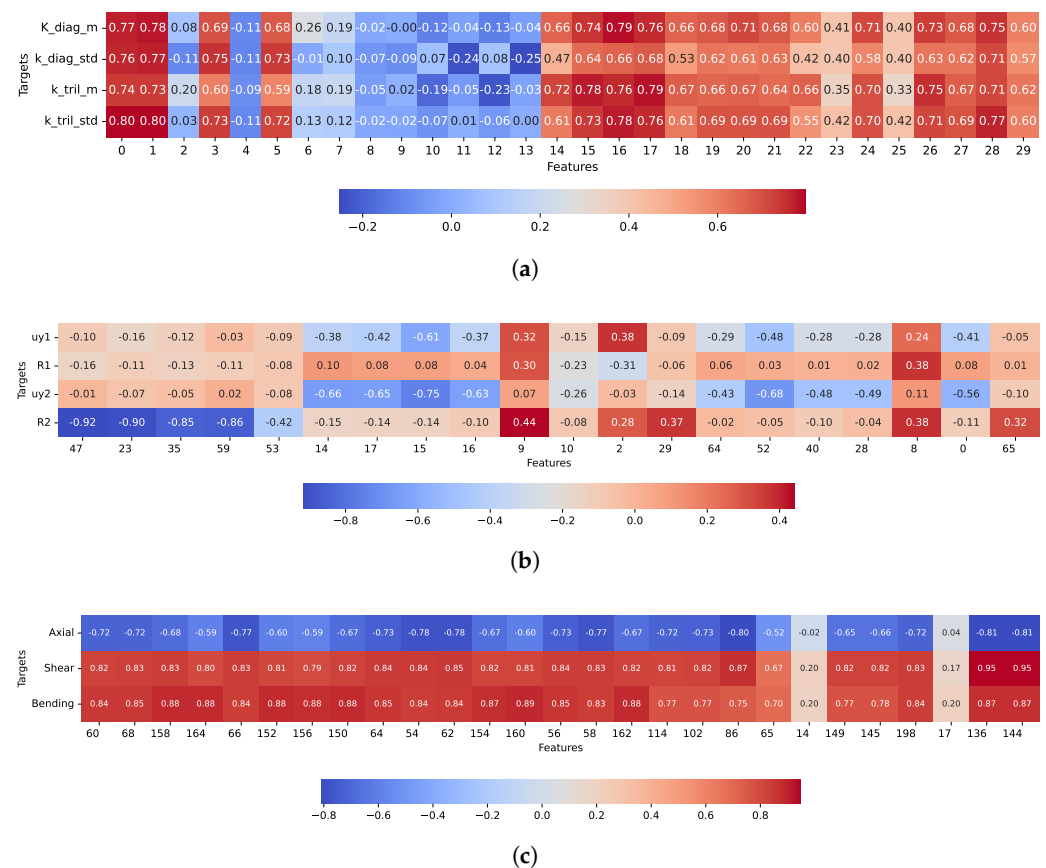


Figure 15. Spearman's rank correlation across all framework layers. (a) Stiffness layer Spearman rank. (b) Displacement layer Spearman rank. (c) Force layer Spearman rank.

Feature-to-feature Spearman correlations at the force layer are visualized for the top-ranked features in Figure 16. Intra-feature correlation is high across nearly all selected features, with the exception of stiffness layer targets at indices 14 and 17. This high intra-feature correlation is consistent with the elevated feature importance scores in Figure 14c and indicates that the selected features carry coherent, complementary information.

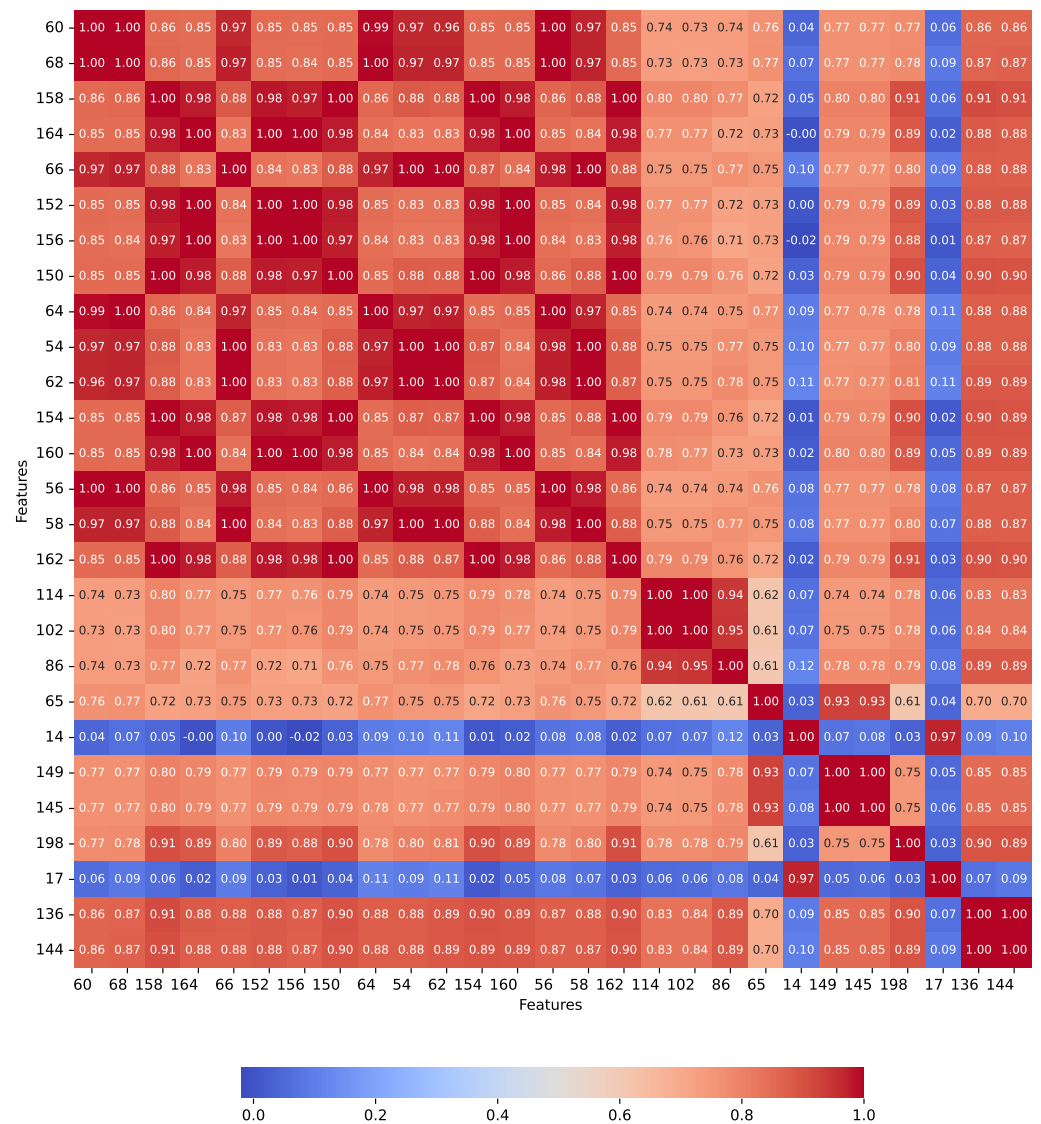


Figure 16. Spearman's rank correlation among top force layer features.

5.5.3. Mutual Information

Mutual information quantifies the reduction in uncertainty about one variable given knowledge of another, capturing nonlinear dependencies that correlation measures cannot:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) \quad (10)$$

where $p(x,y)$ is the joint probability distribution of X and Y , and $p(x)$ and $p(y)$ are their respective marginal distributions.

Figure 17 shows the mutual information matrix for the force layer. Features selected based on importance scores contribute nearly uniformly to each target. Their contribution to axial force prediction is comparatively larger, consistent with the higher R^2 and wider value range of the axial force target.

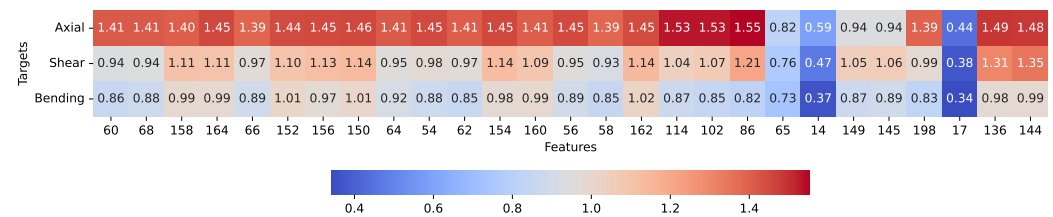


Figure 17. Force layer mutual information matrix.

5.5.4. Role of Nodal Coordinate Features

Nodal coordinates are included as primary input features. Each member has two end nodes with X and Y components, yielding four coordinate features per member at feature indices 10 to 13. Nodal coordinate values are label-encoded, assigning a unique integer identifier to each distinct x and y coordinate in the frame. This treats node positions as categorical location identifiers rather than continuous geometric quantities, enabling the model to distinguish members by their position within the structural topology. These encoded values remain constant across all iterations, as nodal geometry does not change under cross-sectional size variations. Their feature importance scores are accordingly low, as confirmed in Figure 14.

Despite their low individual importance scores, nodal coordinates play a structural role in the model. Comparing Figure 14a with Figure 18, their inclusion elevates the maximum importance score from below 0.10 to approximately 0.15. More notably, secondary features carrying sensitivity information score higher in the presence of nodal coordinates: features at indices 15, 16, 28, and 29 exceed 0.05 with nodal coordinates included but fall below 0.05 when they are excluded. This suggests that nodal coordinates, despite low direct importance, improve the discriminative capacity of other features by encoding positional context.

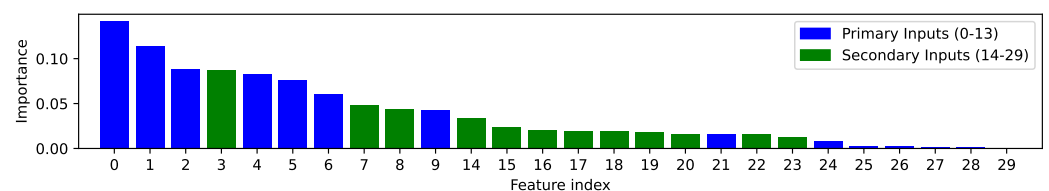


Figure 18. Stiffness layer feature importance without nodal coordinate features (indices 10–13).

5.5.5. Cumulative Feature Contribution at the Force Layer

Figure 19 summarizes the percentage contribution of all 214 force layer features across the three target variables. The importance score range across features is narrow, remaining below 0.05. Features with individually small scores nonetheless contribute cumulatively to prediction quality. This justifies retaining a broad feature set rather than aggressively pruning low-importance features.

Results from all three methods agree. Sensitivity-based features dominate individual importance rank, while nodal coordinate features contribute primarily through collective interaction with other features rather than independently. This suggests that the framework's predictive power derives from two complementary sources: the physics-informed sensitivity features, which encode explicit structural mechanics knowledge, and the coordinate features, which encode implicit topological context that no single feature can capture alone. This distinction has practical implications for feature selection in future extensions of the framework: removing coordinate features based on low individual importance scores would degrade model performance despite their apparently negligible standalone contribution.

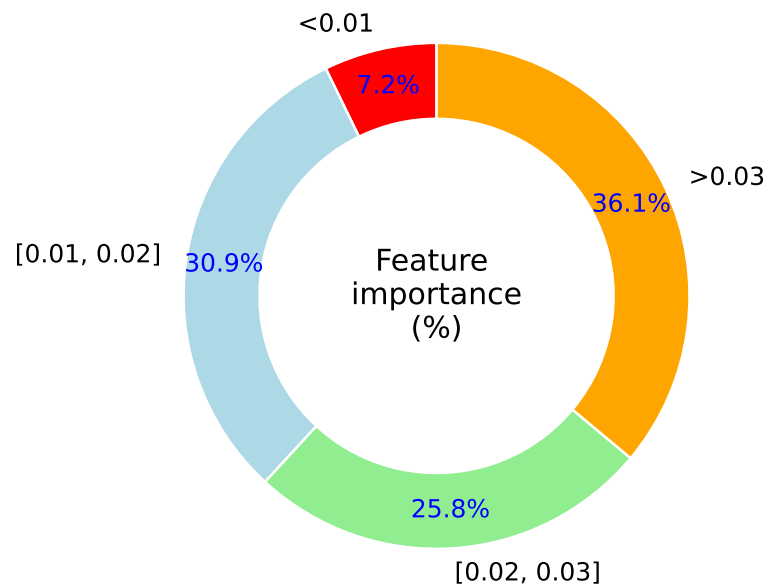


Figure 19. Force layer feature importance percentage contribution across all 214 features.

6. Conclusions

A novel three-layer framework of data-driven ML models is proposed for the re-analysis of framed structures, targeting the prediction of member-level internal forces. Once the dataset is collected, statistical and pattern recognition techniques are used to extract features and train the models. Sensitivity analysis is additionally incorporated to derive key physical representations, grounding the framework in structural mechanics. The approach serves as a computationally efficient alternative to conventional reanalysis methods, particularly in the context of structural size optimization where reanalysis is performed repeatedly.

The hierarchical three-layer architecture consisting of stiffness, displacement, and force layers is designed such that predictions from lower layers enrich the feature set of the layer above. This progressive feature accumulation allows the force layer to leverage the most informative inputs from the entire framework, expanding its feature space and enhancing prediction quality. Feature tables at each layer are not preassembled from known quantities; rather, they are constructed sequentially from predictions made within the framework itself, starting from a limited set of primary variables. The secondary input mechanism is specifically devised to support this process by encoding sensitivity information into the feature representation. It is noted that stiffness layer targets achieve R^2 values in the range of 0.79–0.83, and while this intermediate accuracy is sufficient for the force layer to produce high-quality predictions, the propagation of stiffness prediction error through the hierarchy represents a known limitation of the sequential architecture. This is partially mitigated by the independently computed sensitivity features, which provide a stabilizing signal at each layer that does not depend on upstream predictions.

Given only cross-sectional dimensions, member lengths, and nodal coordinate information, the framework predicts internal responses across multiple targets simultaneously. Unlike computational reanalysis, which requires a full equation solve for each cross-sectional size change sequentially, the trained framework produces predictions for any desired number of cross-sectional configurations in a single batch inference pass by following the non-iterative deployment path. This asymptotic advantage becomes substantial when frame reanalysis is required across thousands or tens of thousands of configurations, as is typical in structural size optimization workflows, and creates a pathway toward replacing iterative optimization workflows with iteration-free prediction approaches.

The models achieve R^2 values of 0.99, 0.98, and 0.91 for axial force, shear force, and bending moment, respectively, on unseen data, demonstrating promising performance given the limited dataset size and the randomness of cross-sectional sampling. The current implementation is validated on a single six-floor, three-bay 2D reinforced concrete frame under gravity loading only, using linear static analysis with fixed-base boundary conditions. These conditions represent a deliberate scope limitation: the framework has not been evaluated for lateral or dynamic loading, nonlinear material behavior, or non-symmetric structural configurations, and its generalizability to such cases requires further investigation. A further limitation is that the numerical experiments are restricted to one RC frame; transfer of a model trained on one structure to distinct structural configurations is outside the scope of this work. The intended use case is to retrain the framework on structure-specific datasets, reusing the same pipeline rather than a single universal model. Performance can be further improved with larger datasets and finer sampling interval calibration, and the framework is in principle applicable to 3D frames, with the corresponding increase in input dimensionality and target variables.

Model interpretability is examined through feature importance, Spearman's rank correlation, and mutual information. The three methods consistently corroborate each other's findings, confirming the reliability of the interpretability analysis. A notable finding is that features with low individual importance scores can still enhance the interpretability of other features. Nodal coordinate features exemplify this role: despite low direct importance, their inclusion elevates the discriminative capacity of sensitivity-derived secondary features, as demonstrated in the case study.

Future work should address the optimal sampling interval for a given cross-sectional variable range, the minimum dataset size required for a target accuracy level, and the optimal number and distribution of secondary input iterations. Quantifying error propagation across layers and its sensitivity to upstream prediction quality represents a particularly important direction. Extensions to laterally loaded frames, dynamic analysis, non-symmetric structural configurations, and 3D frames represent natural directions for further development.

Author Contributions: Conceptualisation, Y.L.A.; methodology, Y.L.A.; resources, Y.L.A., B.H., G.U., C.W. and T.L.; software, Y.L.A.; validation, Y.L.A.; writing—original draft preparation, Y.L.A.; writing—review and editing, Y.L.A., B.H., G.U., C.W. and T.L.; and supervision, Y.L.A., B.H., G.U., C.W. and T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a special fund of the German government at the BMBF [grant number: 16DKWN078A].



Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in this research is available upon request to the corresponding author.

Acknowledgments: This work was supported by a special BMBF fund for the Intelligente Methoden zur automatischen und nachvollziehbaren Analyse umfangreicher Infrastruktur-, Verkehrs- und Umweltmessdaten (InMeA) project (grant number: 16DKWN078A). The BMBF is therefore acknowledged for the funding provided.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Akgün, M.A.; Garcelon, J.H.; Haftka, R.T. Fast exact linear and non-linear structural reanalysis and the Sherman–Morrison–Woodbury formulas. *Int. J. Numer. Methods Eng.* **2001**, *50*, 1587–1606. [\[CrossRef\]](#)
2. Chen, S.; Yang, X.; Lian, H. Comparison of several eigenvalue reanalysis methods for modified structures. *Struct. Multidiscip. Optim.* **2000**, *20*, 253–259. [\[CrossRef\]](#)
3. Kirsch, U. *Reanalysis of Structures*; Springer: Berlin/Heidelberg, Germany, 2008.
4. Kirsch, U. Combined approximations—A general reanalysis approach for structural optimization. *Struct. Multidiscip. Optim.* **2000**, *20*, 97–106. [\[CrossRef\]](#)
5. Lee, M.; Jung, Y.; Choi, J.; Lee, I. A reanalysis-based multi-fidelity (RBMF) model framework for efficient structural optimization. *Comput. Struct.* **2022**, *273*, 106895. [\[CrossRef\]](#)
6. Jenkins, W. A neural network for structural re-analysis. *Comput. Struct.* **1999**, *72*, 687–698. [\[CrossRef\]](#)
7. Jenkins, W. Structural reanalysis using a neural network-based iterative method. *J. Struct. Eng.* **2002**, *128*, 946–950. [\[CrossRef\]](#)
8. Ahmad, A.; Cotsovos, D.M.; Lagaros, N.D. Framework for the development of artificial neural networks for predicting the load carrying capacity of RC members. *SN Appl. Sci.* **2020**, *2*, 545. [\[CrossRef\]](#)
9. Tandale, S.B.; Markert, B.; Stoffel, M. Smart stiffness computation of one-dimensional finite elements. *Mech. Res. Commun.* **2022**, *119*, 103817. [\[CrossRef\]](#)
10. Kaveh, A.; Eslamlou, A.D. *Metaheuristic Optimization Algorithms in Civil Engineering: New Applications*; Springer: Berlin/Heidelberg, Germany, 2020.
11. Bao, Y.; Li, H. Machine learning paradigm for structural health monitoring. *Struct. Health Monit.* **2021**, *20*, 1353–1372. [\[CrossRef\]](#)
12. Thai, H.T. Machine learning for structural engineering: A state-of-the-art review. In *Proceedings of the Structures*; Elsevier: Amsterdam, The Netherlands, 2022; Volume 38, pp. 448–491.
13. Alemu, Y.L.; Lahmer, T.; Walther, C. Damage Detection with Data-Driven Machine Learning Models on an Experimental Structure. *Eng* **2024**, *5*, 629–656. [\[CrossRef\]](#)
14. Champaney, V.; Chinesta, F.; Cueto, E. Engineering empowered by physics-based and data-driven hybrid models: A methodological overview. *Int. J. Mater. Form.* **2022**, *15*, 31. [\[CrossRef\]](#)
15. Zhou, Y.; Meng, S.; Lou, Y.; Kong, Q. Physics-informed deep learning-based real-time structural response prediction method. *Engineering* **2024**, *35*, 140–157. [\[CrossRef\]](#)
16. Truong, V.H.; Pham, H.A. Support vector machine for regression of ultimate strength of trusses: A comparative study. *Eng. J.* **2021**, *25*, 157–166. [\[CrossRef\]](#)
17. Yetilmezsoy, K.; Sihag, P.; Kıyan, E.; Doran, B. A benchmark comparison and optimization of Gaussian process regression, support vector machines, and M5P tree model in approximation of the lateral confinement coefficient for CFRP-wrapped rectangular/square RC columns. *Eng. Struct.* **2021**, *246*, 113106. [\[CrossRef\]](#)
18. Lu, S.; Jiang, M.; Wang, X.; Yu, H.; Su, C. Damage degree prediction method of CFRP structure based on fiber Bragg grating and epsilon-support vector regression. *Optik* **2019**, *180*, 244–253. [\[CrossRef\]](#)
19. Khan, A.A.; Chaudhari, O.; Chandra, R. A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. *Expert Syst. Appl.* **2024**, *244*, 122778. [\[CrossRef\]](#)
20. Louk, M.H.L.; Tama, B.A. Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system. *Expert Syst. Appl.* **2023**, *213*, 119030. [\[CrossRef\]](#)
21. Bentéjac, C.; Csörgő, A.; Martínez-Muñoz, G. A comparative analysis of gradient boosting algorithms. *Artif. Intell. Rev.* **2021**, *54*, 1937–1967. [\[CrossRef\]](#)
22. Ganaie, M.A.; Hu, M.; Malik, A.K.; Tanveer, M.; Suganthan, P.N. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105151. [\[CrossRef\]](#)
23. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Volume 1.
24. Cianci, E.; Civera, M.; De Biagi, V.; Chiaia, B. Physics-informed machine learning for the structural health monitoring and early warning of a long highway viaduct with displacement transducers. *Mech. Syst. Signal Process.* **2026**, *242*, 113659. [\[CrossRef\]](#)

25. Demir, A.; Sahin, E.K.; Demir, S. Advanced tree-based machine learning methods for predicting the seismic response of regular and irregular RC frames. *Structures* **2024**, *64*, 106524. [[CrossRef](#)]
26. Yahiaoui, A.; Dorbani, S.; Yahiaoui, L. Machine learning techniques to predict the fundamental period of infilled reinforced concrete frame buildings. *Structures* **2023**, *54*, 918–927. [[CrossRef](#)]
27. Kazemi, F.; Asgarkhani, N.; Jankowski, R. Machine learning-based seismic fragility and seismic vulnerability assessment of reinforced concrete structures. *Soil Dyn. Earthq. Eng.* **2023**, *166*, 107761. [[CrossRef](#)]
28. Alemu, Y.L.; Habte, B.; Lahmer, T.; Urgessa, G. Priority Criteria (PC) Based Particle Swarm Optimization of Reinforced Concrete Frames (PCPSO). *CivilEng* **2023**, *4*, 679–701. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.