Preprints.org

Article

Not peer-reviewed version

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

Yohannes L. Alemu *, Bedilu Habte, Girum Urgessa, Christian Walther, Tom Lahmer

Posted Date: 2 April 2026

doi: 10.20944/preprints202604.0174.v1

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

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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.

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, Germany
[2] School of Civil and Environmental Engineering, Addis Ababa Institute of Technology, Ethiopia
[3] Department of Civil and Environmental Engineering, College of Engineering and Architecture, Howard University, Washington, DC 20059, USA
* 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 1,890 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, the framework predicts any number of design iterations in a single inference pass. This substantially reduces the computational cost of reanalysis-based workflows. The proposed framework offers a scalable, interpretable, and physics-consistent alternative to both classical reanalysis methods and purely data-driven surrogate 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

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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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 surrogate-based reanalysis approaches have also been explored to reduce this cost, though they still rely on repeated reduced-order solves during the surrogate 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 behaviour. 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 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 submatrices of the global stiffness matrix using nodal degree-of-freedom indices, encoding inter-member

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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 1,890 data points from 45 finite element iterations.

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 member-level stiffness representation extracted as submatrices 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: 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].

2.1. Support Vector Machine

Support Vector Machine (SVM) regression constructs a decision boundary in a transformed feature space using a kernel function [16,17]. The ε-insensitive tube formulation allows control over prediction tolerance, making SVM suitable for tasks where outlier sensitivity must be managed [18]. For structural reanalysis, the nonlinear relationship between cross-sectional properties and internal forces makes kernel-based transformation relevant. However, SVM is comparatively slower to train as dataset size increases, which limits its scalability in large reanalysis workflows.

2.2. Ensemble Methods

Ensemble methods combine multiple weak learners to improve prediction quality and generalization [19]. Two principal strategies are used: bagging, which builds parallel decision trees on bootstrap samples and aggregates their outputs; and boosting, which arranges trees sequentially such that each tree corrects the errors of the previous one [20].

2.2.1. Random Forest

Random Forest (RF) applies bagging over an ensemble of decision trees, with each tree trained on a random feature subset [12]. The resulting diversity reduces overfitting and improves generalization, which is particularly beneficial when training data is limited. RF returns the average prediction across all trees for regression tasks. Its simplicity, interpretability, and robustness make it well-suited for the feature-rich, moderate-sized dataset generated in this work.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

2.2.2. Gradient Boosting

Gradient Boosting (GB) arranges weak decision trees in series, where each successive tree is trained to minimize the residual error of the previous one [20]. This sequential correction improves accuracy on complex nonlinear mappings typical of structural response prediction.

2.2.3. Extreme Gradient Boosting

Extreme Gradient Boosting (XGB) extends GB with L1 and L2 regularization terms that penalize model complexity, reducing overfitting [21]. It offers additional hyperparameter control and computational efficiency. Comprehensive reviews of ensemble methods are provided in [19,22].

2.3. Neural Networks

Neural Networks (NN) consist of an input layer, one or more hidden layers, and an output layer [23]. Weights and biases are updated through backpropagation to minimize a defined loss function. Their application in structural engineering has expanded to health monitoring and response prediction [13,24]. For the reanalysis task, a single hidden layer architecture may suffice to capture the nonlinear relationships present, while deeper architectures offer more flexibility at the cost of increased training complexity and data requirements.

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:

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

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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

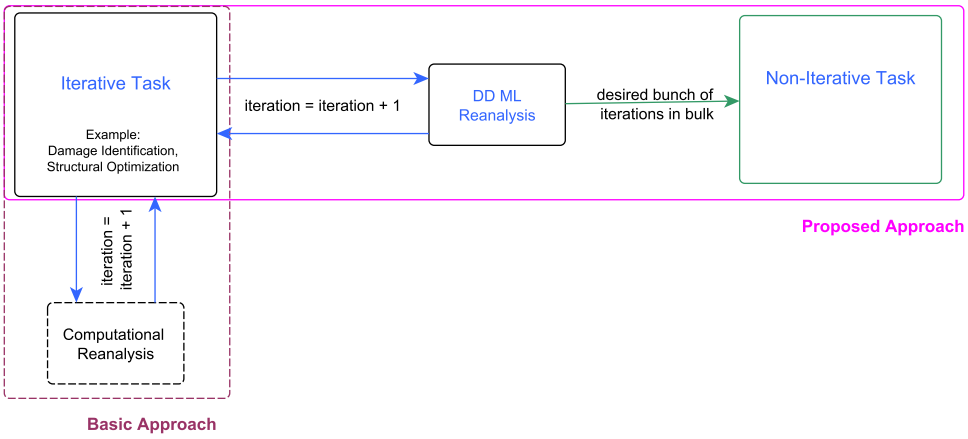


Figure 1. Generic integration of the machine learning reanalysis framework into iterative and non-iterative tasks

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..

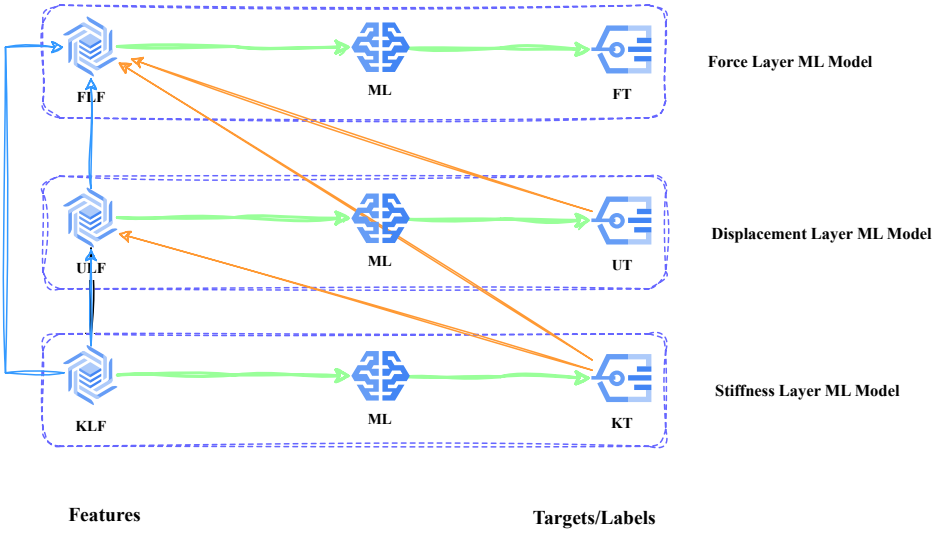


Figure 2. A three-layer framework of machine learning models with stiffness, displacement and force layer features (K_{LF}, U_{LF}, F_{LF}) and their respective targets (K_T, U_T, F_T)

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,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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 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.

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	Force
Structural information	members, nodal connectivity, degree of freedom, nodal coordinates	All layers

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.

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.

$$K_{i+1} = K_i + \frac{dK}{dx}(x_{i+1} - x_i), \quad \text{and} \quad U_{i+1} = U_i + \frac{dU}{dx}(x_{i+1} - x_i), \tag{2}$$

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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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

Given rectangular cross-sections of members and applying the chain rule, simplified for changes related to cross-sectional area and moment of inertia:

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

we get

$$\frac{\partial K}{\partial A} = \left[2 - \frac{b_s}{b} - \frac{D_s}{D}\right]K_s, \tag{3}$$

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

where A is area and I is moment of inertia. Equation 3 and Equation 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 Equation 5 and Equation 6.

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

and

$$\frac{\partial U}{\partial I} = \left[-4 + \frac{3b_sD_s^2}{bD^2} + \frac{D_s^3}{D^3}\right]U_s. \tag{6}$$

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:

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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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 optimising 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 Diagonal stiffness matrix: standard deviation Lower triangular stiffness matrix: mean Lower triangular stiffness matrix: standard deviation	4	logarithm of square root
Displacement layer	Translations Rotations	6	sigmoid
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.

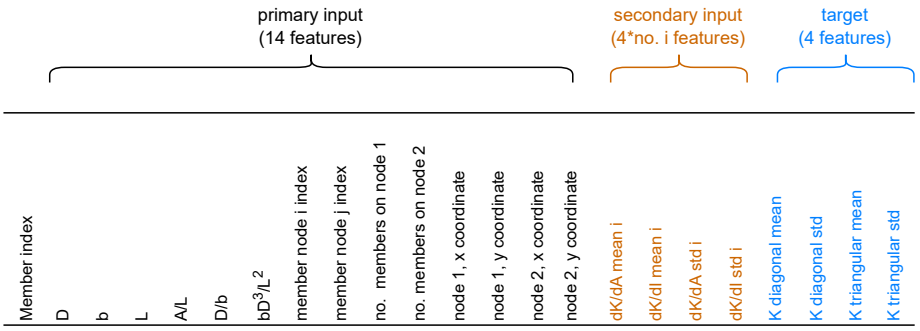
Regarding member type, beams and columns may be modeled separately or together. From a domain knowledge perspective, separate treatment is reasonable: moment and shear govern beam behavior, while axial force and moment dominate in columns under limited lateral loading, and their cross-sectional geometries 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 subsection 5.2.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

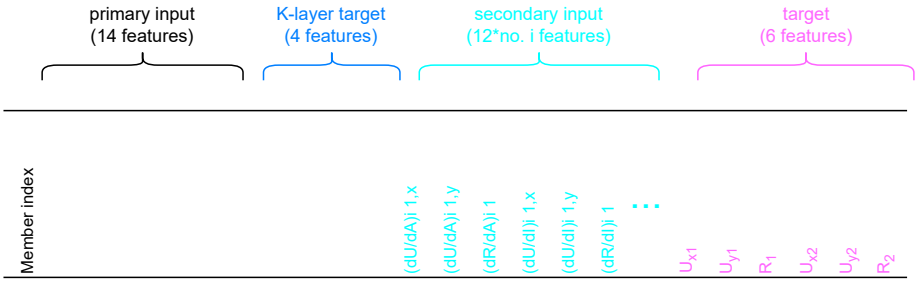
Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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.

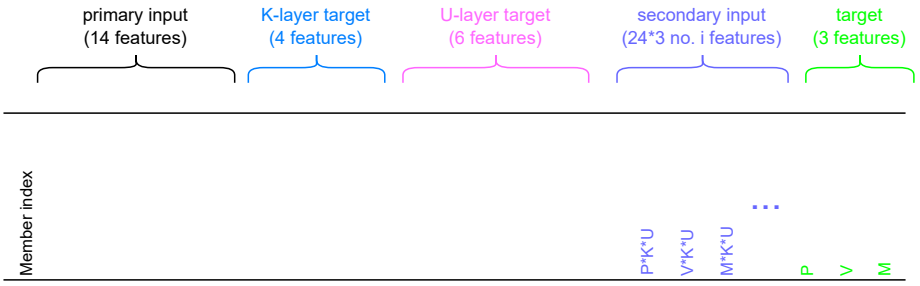
The stiffness layer feature table is shown in Figure 3a. Secondary inputs are computed using Equation 3 and Equation 4. The displacement layer feature table in Figure 3b incorporates stiffness layer targets as inputs, with secondary inputs computed using Equation 5 and Equation 6. The force layer feature 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.



(**a**) Feature table for stiffness layer



(**b**) Feature table for displacement layer



(**c**) Feature table for force layer

Figure 3. Feature tables for each layer of the proposed framework

4.4. General Procedure

The general application procedure is summarized in Table 3.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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.	Arround 80% of data used as primary input computation. Equation 3–Equation 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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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 [25]. 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.

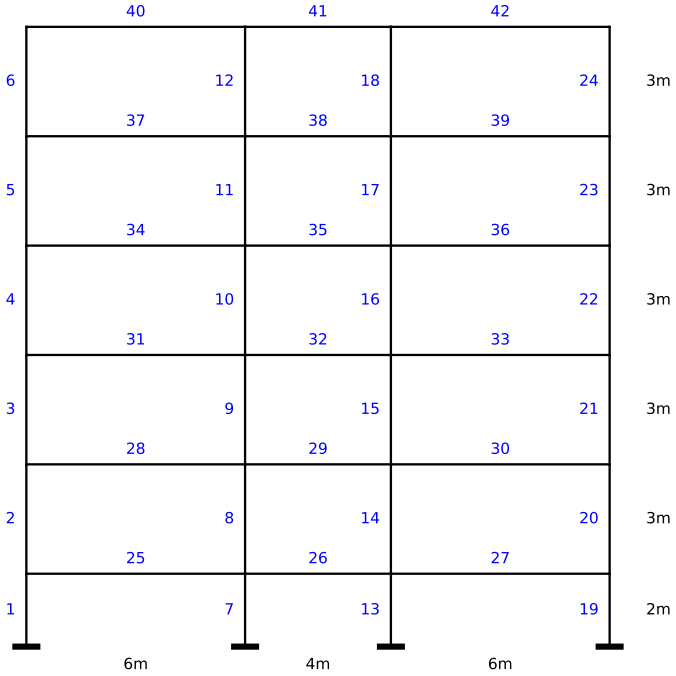


Figure 4. Considered 2-dimensional reinforced concrete frame

A total of 1,890 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.

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

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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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.

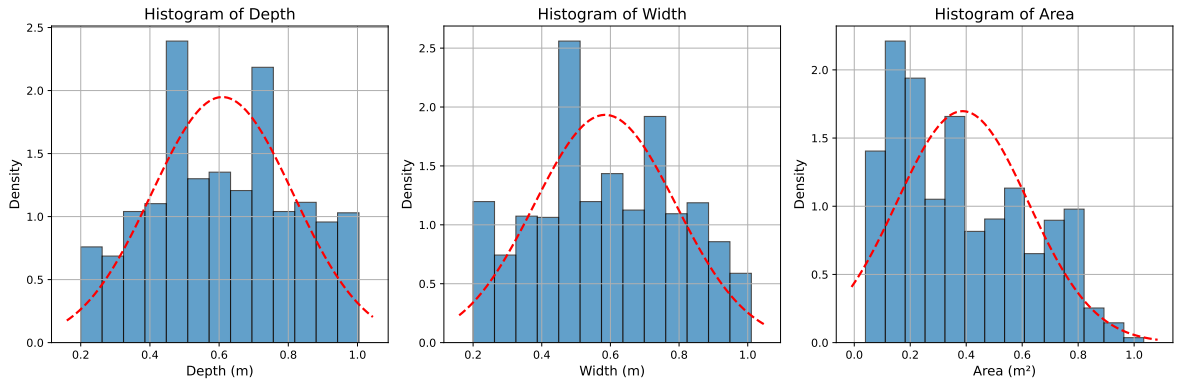


Figure 5. Member cross-sectional dataset distribution

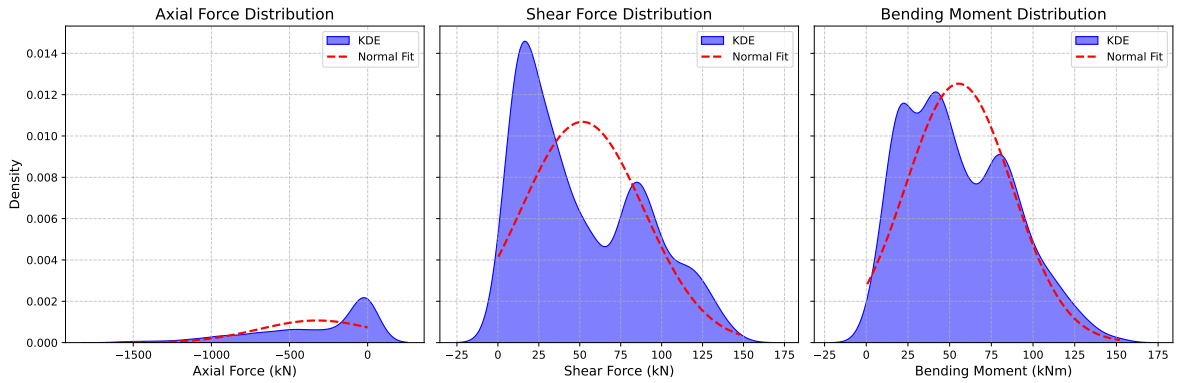


Figure 6. Internal force data distribution

Member stiffness extraction from the global stiffness matrix is performed in MATLAB R2018b. ML scripts are written in Python 3.9.18 using TensorFlow v2.10.0. All features are normalized using z-score standardization prior to training. Dataset representation uses a 2D matrix format.

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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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.

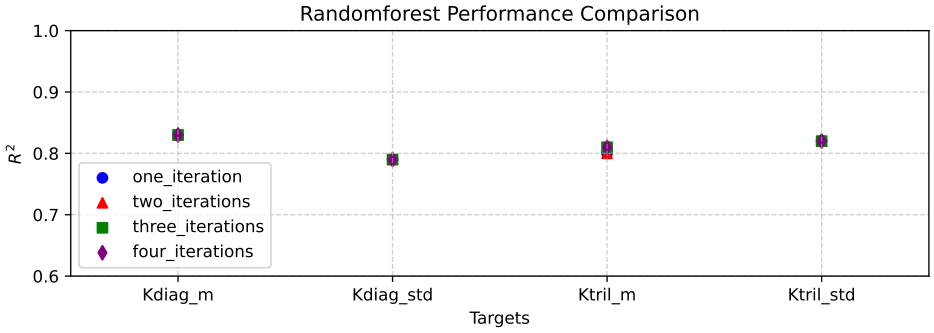


Figure 7. R^2 for Random Forest models with varying numbers of secondary input iterations in the stiffness layer

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.

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 subsection 5.3.

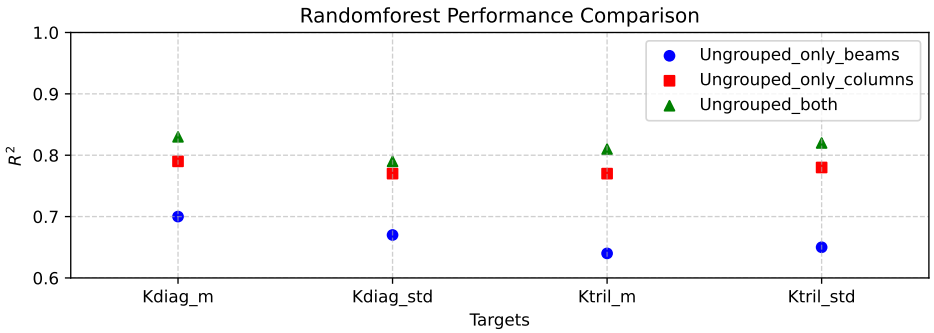


Figure 8. R^2 for beam-only, column-only, and combined cases in the stiffness layer

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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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.

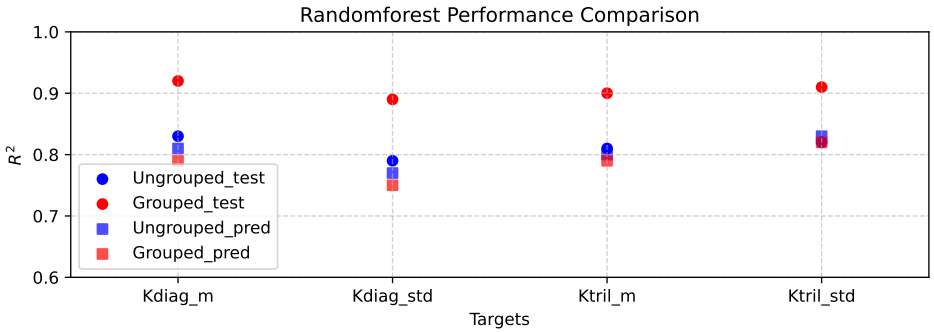


Figure 9. R^2 for grouped and ungrouped splitting in the stiffness layer

5.2.4. Regression Algorithm Selection

SVM, RF, GB, XGB, and NN models are evaluated on the stiffness layer targets. 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 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 [26,27].

Table 4. Algorithm performance comparison on stiffness layer targets

Model	R^2				RMSE			
	K_{diag_m}	K_{diag_std}	K_{tril_m}	K_{tril_std}	K_{diag_m}	K_{diag_std}	K_{tril_m}	K_{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

5.3. 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 5 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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

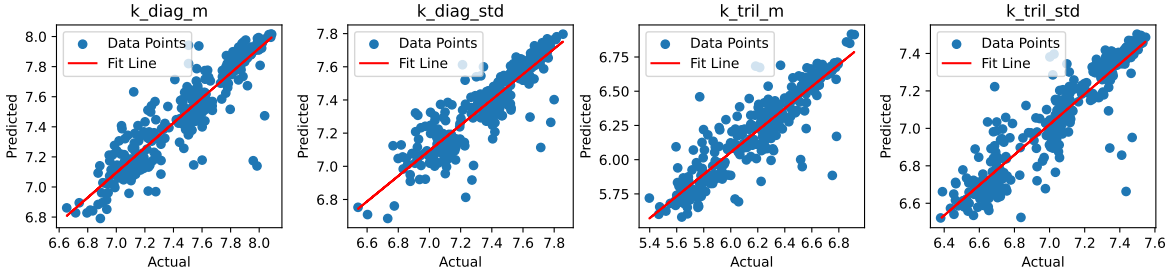
Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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.

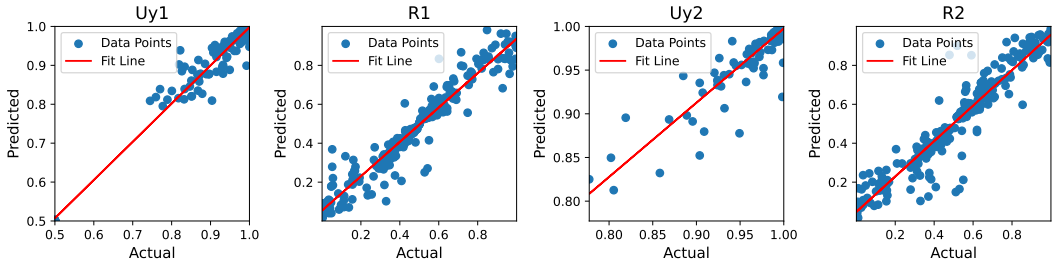
Table 5. Model performance metrics for each framework layer

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

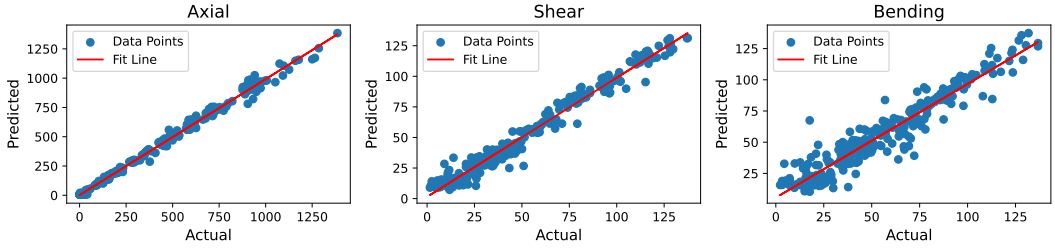
The relatively high R^2 alongside elevated RMSE and MAE for axial force is a consequence of the bimodal target distribution visible in Figure 10: beam members carry near-zero axial force, while lower-story columns accumulate axial loads exceeding 1000 kN under gravity loading. This wide dynamic range 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.



(**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

Figure 10. Predicted vs. true value fitting lines for each framework layer

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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.

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 6.

Table 6. Force layer performance by member type on test and unseen data

Metric	Data	Beams only			Columns only			Beams and columns		
		P	***V***	***M***	***P***	***V***	***M***	***P***	***V***	***M***
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.

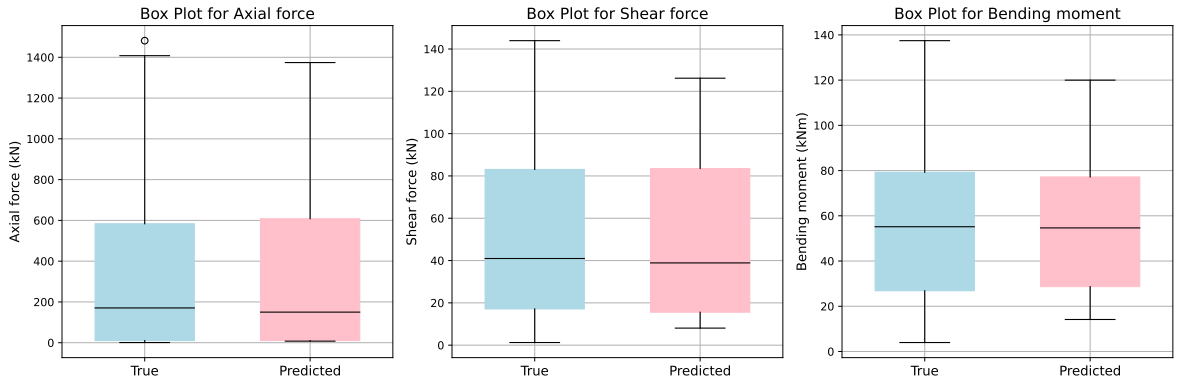


Figure 11. Box plot of true and predicted internal forces on unseen data

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

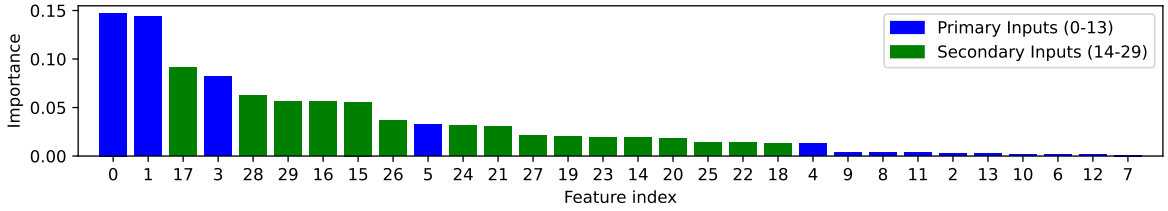
Once trained, the framework generates predictions for any number of cross-sectional size variations in a single inference pass, whereas FEM-based reanalysis requires a full equation solve for each change sequentially. This gap widens considerably when frame reanalysis is required across thousands or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

5.4. Interpretability

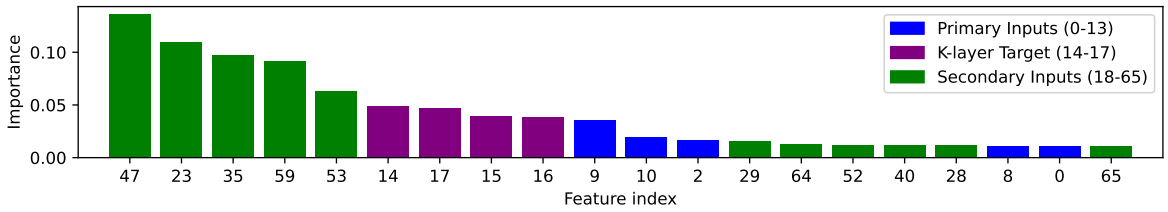
Model interpretability is examined through feature importance, Spearman's rank correlation, mutual information, and nodal coordinate contribution analysis. Pearson correlation is excluded, as it is limited to linear relationships and the problem is inherently nonlinear. For the stiffness layer, all 30 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.

5.4.1. Feature Importance

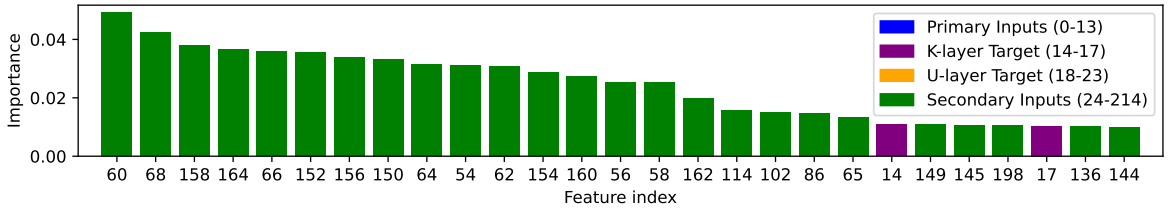
As expected, depth D, width b, and the moment of inertia derived from them contribute most significantly at the stiffness layer, as shown in Figure 12a. Secondary input stiffness statistics also rank highly. Member length, connectivity, and nodal coordinates contribute less, which is consistent with their invariance across iterations.



(**a**) Stiffness layer feature importance



(**b**) Displacement layer feature importance



(**c**) Force layer feature importance

Figure 12. Feature importance scores across all framework layers

At the displacement and force layers (Figure 12b, Figure 12c), 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. Although individual importance scores are small, their cumulative contribution across all features remains substantial.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

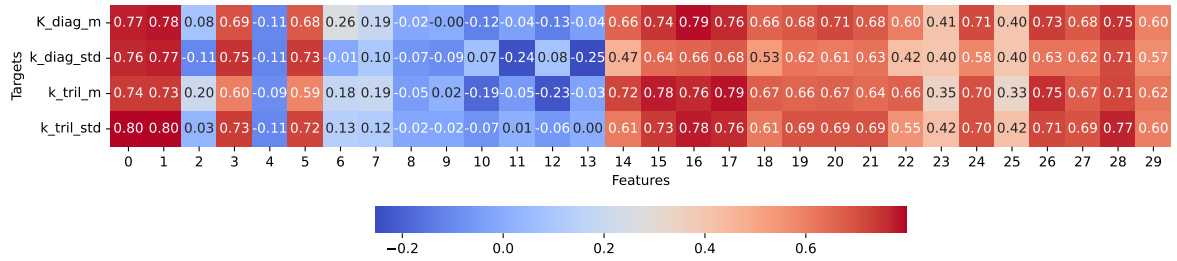
5.4.2. Spearman's Rank Correlation

Spearman's rank correlation coefficient is defined as:

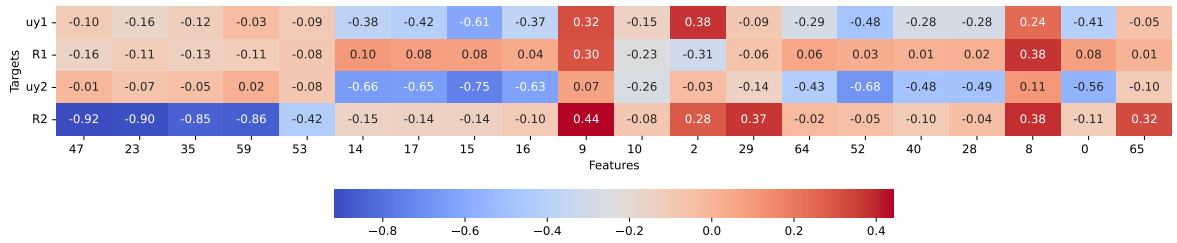
$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}. \tag{9}$$

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

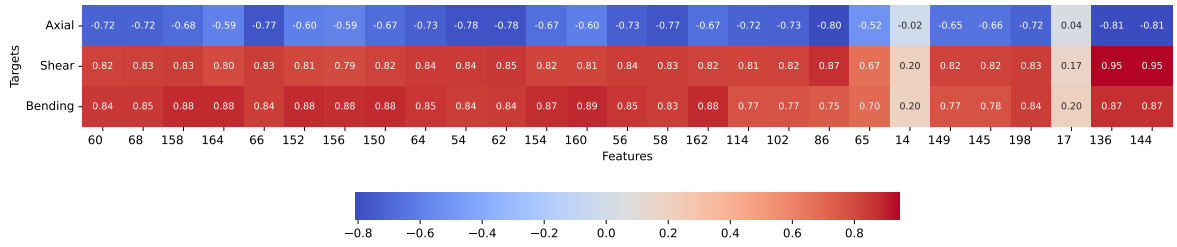
At the stiffness layer, Figure 13a confirms the dominance of cross-sectional depth, width, and moment of inertia, consistent with Figure 12a. At the displacement layer, Figure 13b shows a more uniform distribution of Spearman scores among the top features. This divergence from feature importance rank 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 13c shows the clear dominance of secondary inputs, with a negative correlation between these features and axial force predictions.



(**a**) Stiffness layer Spearman rank



(**b**) Displacement layer Spearman rank



(**c**) Force layer Spearman rank

Figure 13. Spearman's rank correlation across all framework layers

Feature-to-feature Spearman correlations at the force layer are visualized for the top-ranked features in Figure 14. 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 12c and indicates that the selected features carry coherent, complementary information.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

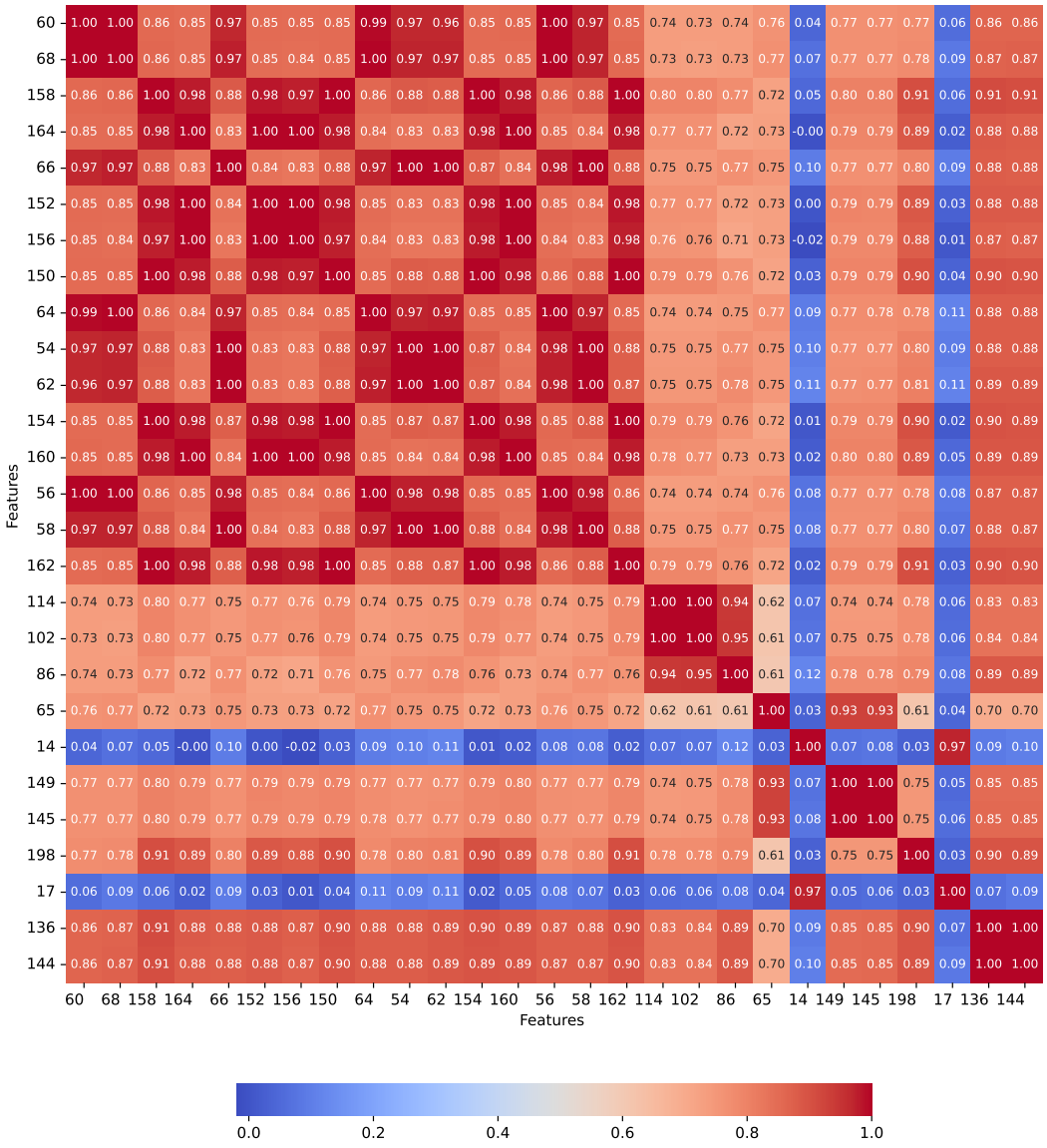


Figure 14. Spearman's rank correlation among top force layer features

5.4.3. Mutual Information

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

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

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

Figure 15 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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

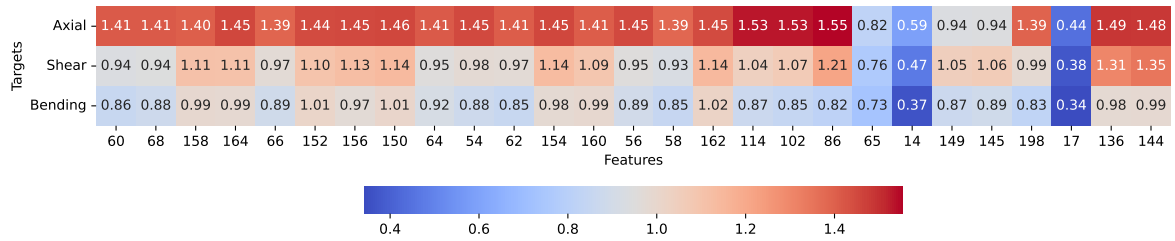


Figure 15. Force layer mutual information matrix

5.4.4. 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 12.

Despite their low individual importance scores, nodal coordinates play a structural role in the model. Comparing Figure 12a with Figure 16, 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.

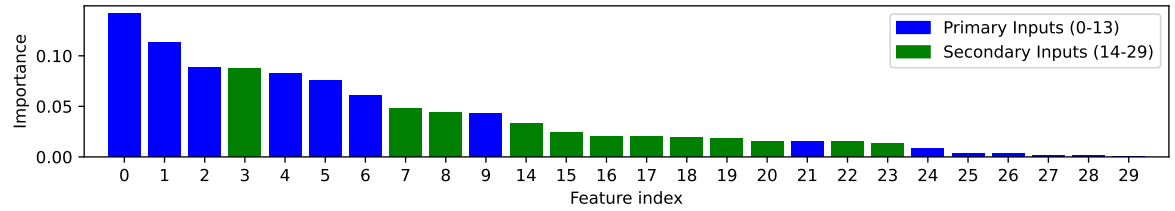


Figure 16. Stiffness layer feature importance without nodal coordinate features (indices 10–13)

5.4.5. Cumulative Feature Contribution at the Force Layer

Figure 17 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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

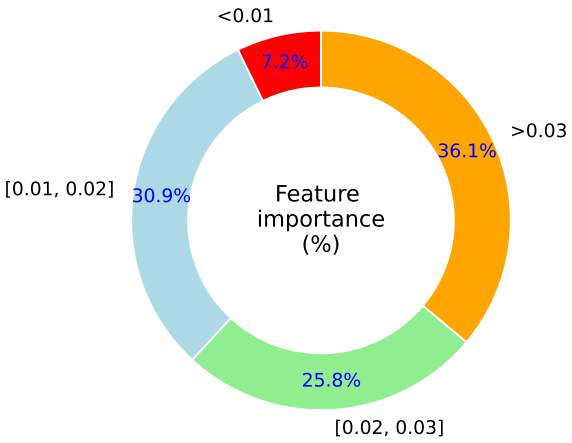


Figure 17. Force layer feature importance percentage contribution across all 214 features

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.

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.; 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: 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 Analyze umfangreicher Infrastruktur-, Verkehrs- und Umweltmessdaten (InMeA) project [grant number: 16DKWN078A]. The BMBF is therefore acknowledged for the funding provided.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

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. *International Journal for Numerical Methods in Engineering* **2001**, *50*, 1587–1606.
2. Chen, S.; Yang, X.; Lian, H. Comparison of several eigenvalue reanalysis methods for modified structures. *Structural and Multidisciplinary optimization* **2000**, *20*, 253–259.
3. Kirsch, U. *Reanalysis of structures*; Springer, 2008.
4. Kirsch, U. Combined approximations–a general reanalysis approach for structural optimization. *Structural and Multidisciplinary Optimization* **2000**, *20*, 97–106.
5. Lee, M.; Jung, Y.; Choi, J.; Lee, I. A reanalysis-based multi-fidelity (RBMF) surrogate framework for efficient structural optimization. *Computers & Structures* **2022**, *273*, 106895.
6. Jenkins, W. A neural network for structural re-analysis. *Computers & Structures* **1999**, *72*, 687–698.
7. Jenkins, W. Structural reanalysis using a neural network-based iterative method. *Journal of Structural Engineering* **2002**, *128*, 946–950.
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 Applied Sciences* **2020**, *2*, 1–21.
9. Tandale, S.B.; Markert, B.; Stoffel, M. Smart stiffness computation of one-dimensional finite elements. *Mechanics Research Communications* **2022**, *119*, 103817.
10. Kaveh, A.; Eslamlou, A.D. *Metaheuristic optimization algorithms in civil engineering: new applications*; Springer, 2020.
11. Bao, Y.; Li, H. Machine learning paradigm for structural health monitoring. *Structural health monitoring* **2021**, *20*, 1353–1372.
12. Thai, H.T. Machine learning for structural engineering: A state-of-the-art review. In Proceedings of the Structures. Elsevier, 2022, Vol. 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.
14. Champaney, V.; Chinesta, F.; Cueto, E. Engineering empowered by physics-based and data-driven hybrid models: A methodological overview. *International Journal of Material Forming* **2022**, *15*, 31.
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.
16. Truong, V.H.; Pham, H.A. Support vector machine for regression of ultimate strength of trusses: A comparative study. *Engineering Journal* **2021**, *25*, 157–166.
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. *Engineering Structures* **2021**, *246*, 113106.
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.
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 Systems with Applications* **2023**, p. 122778.
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 Systems with Applications* **2023**, *213*, 119030.
21. Bentéjac, C.; Csörgő, A.; Martínez-Muñoz, G. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review* **2021**, *54*, 1937–1967.
22. Ganaie, M.A.; Hu, M.; Malik, A.K.; Tanveer, M.; Suganthan, P.N. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence* **2022**, *115*, 105151.
23. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep learning*; Vol. 1, MIT press Cambridge, 2016.
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. *Mechanical Systems and Signal Processing* **2026**, *242*, 113659.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2026 doi:10.20944/preprints202604.0174.v1

25. 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.
26. Demir, A.; Sahin, E.K.; Demir, S. Advanced tree-based machine learning methods for predicting the seismic response of regular and irregular RC frames. In Proceedings of the Structures. Elsevier, 2024, Vol. 64, p. 106524.
27. Yahiaoui, A.; Dorbani, S.; Yahiaoui, L. Machine learning techniques to predict the fundamental period of infilled reinforced concrete frame buildings. In Proceedings of the Structures. Elsevier, 2023, Vol. 54, pp. 918–927.

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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.