



Developing a robust machine learning framework for predicting the behavior of large-scale structure

Siamak Talatahari^a, Fang Chen^a, Amir H. Gandomi^{a,b,*}

^a Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, 2007, Australia

^b University Research and Innovation Center (EKIK), Óbuda University, 1034, Budapest, Hungary

ARTICLE INFO

Keywords:

Analyzing framework
Structural analysis and design
Machine learning
Large-scale structures
Performance predicting

ABSTRACT

A rapid and accurate analysis is crucial to structural design and control. This paper presents a comprehensive and robust framework for the analysis and prediction of the behavior of large-scale structures. To ensure accurate prediction of structural responses, the proposed framework integrates sample selection, dimensionality reduction, and advanced machine learning analysis techniques. Three different types of structures are used as numerical examples: a tower truss structure, a steel building, and a reinforced concrete structure. The framework is trained using 26 different machine-learning methods and validated using a comprehensive set of performance metrics. Additionally, an enhanced machine learning method is introduced to achieve more accurate results. This technique leverages chaos game optimization to automate the parameter updating of the machine learning method. Shapley Additive Explanations (SHAP), as the interpretability technique, was incorporated into the framework to quantify each feature's contribution, helping engineers identify key factors influencing structural behavior and ensuring safer, more efficient designs. The validation results demonstrate the high accuracy of the proposed framework in predicting the behavior of large-scale structures. The implications for structural engineering are significant, as this framework has the potential to enhance the design and assessment of large-scale structures, thereby enhancing their safety, durability, and performance.

1. Introduction

Designing engineering problems is a complicated and challenging task [1] due to the existing large number of variables (large problem spaces), various influencing parameters (such as properties of materials, loading definitions, and geometry conditions), complicated relationships between the variables/parameters (nonlinear response space), and hardships in creating or solving the related equations (large uncertainty). Furthermore, the decisions (selecting appropriate designs) made in the design process should be grounded in a clear understanding of the problem, which requires engineers to thoroughly comprehend the problem and its implications to identify the most effective solutions.

This requires a deep understanding of the physical nature of the problem, its governing equations, and the limitations of the available technology and materials ensuring that the chosen design solutions are both feasible and optimal. The problem can often be so complex that it may not be easy to define a meaningful description of it. For instance, designing buildings requires a profound

* Corresponding author. Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, 2007, Australia
E-mail addresses: siamak.talatahari@mq.edu.au, siamak.talat@gmail.com (S. Talatahari), Fang.Chen@uts.edu.au (F. Chen), gandomi@uts.edu.au (A.H. Gandomi).

<https://doi.org/10.1016/j.job.2025.112204>

Received 13 September 2024; Received in revised form 3 February 2025; Accepted 22 February 2025

Available online 25 February 2025

2352-7102/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

understanding of the physics behind structures, the loads they will endure, the properties of the materials that will be used, and the limitations of available construction techniques. Considering all these factors, a designer must be well-informed and recognize the complexities of the task to create an effective design. Our ability to assess designs accurately is facilitated through structural analysis, which provides a comprehensive understanding of the problem. This not only allows for more efficient and cost-effective design but also helps identify potential weaknesses in the structure, enabling the development of solutions to address them. Ultimately, structural analysis ensures the safety, durability, and longevity of structures making it crucial to employ suitable techniques for analyzing and designing structures.

Due to the iterative nature of the optimization methods in structural design problems, it becomes critical to employ a prompt and reliable analysis technique as part of the optimization process [2]. This ensures that each iteration is based on accurate and timely assessments, leading to more efficient and effective design outcomes. This analysis technique must be capable of rapidly delivering performance data to guide the optimization process effectively [3,4]. It must also be able to handle complex design problems while maintaining computational efficiency. Given that the optimization process relies on performance feedback to update design parameters, an unreliable or slow analysis technique can significantly hinder efficiency, preventing the design from reaching its optimal solution in a timely manner. Furthermore, an inability to handle complex design problems can lead to incorrect results and ultimately, a sub-optimal design. For example, a design optimization technique relying on a finite element analysis (FEA) might not be suitable for problems requiring optimization over large geometries with fine details, as the computational time required can be prohibitive. In the same way, structural damage analysis, construction stage design, and nonlinear analysis often require iterative analysis processes, further emphasizing the importance of a robust and efficient analysis method [5]. Structural damage analysis involves assessing how structures respond to loads after sustaining damage. As a result of the design process of construction stages, engineers can determine the safest and most efficient method of building a structure. Nonlinear analysis, on the other hand, focuses on understanding the behavior of structures under varying loads ensuring they can withstand forces such as earthquakes and wind. Direct engineering analysis methods are formulated based on well-established Partial Differential Equations (PDE) and experimental research [6]. However, these traditional methods are computationally expensive, and involve complex matrix relationships that are difficult to simplify without sacrificing accuracy.

Although advancements in the hardware and software for computational machines have significantly enhanced the capacity for engineering analysis in recent decades, only a few academic studies have addressed complex and practical problems. This indicates that the full potential of computational tools for engineering analysis remains underutilized. The primary challenge lies in the fact that the algorithms involved are often slow and computationally expensive, making them difficult to process in real-world projects. While these algorithms could be accelerated through parallel computing or specialized hardware, these methods come with challenges such as complex implementation and high costs. The utilization of AI-based learning methods offers a promising solution to this problem, as these methods are known for their significant ability to reduce the complexity of engineering challenges [7], [8]. While early AI applications in engineering, such as damage detection [9], health monitoring [10], optimization [11], strength and resistance prediction [12], and reliability studies [13], were not directly linked to the structural analysis and design process addressed in this study, recent developments have focused on integrating AI methods into structural analysis and design, [14,15].

Recent advancements in computational power and storage have catalyzed the development of sophisticated AI tools and techniques capable of addressing highly nonlinear learning problems in engineering design [16–18]. These advancements have significantly revitalized the role of AI in engineering analysis and design, making it increasingly integral to modern engineering practices. For instance, there have been notable developments in applying AI to seismic engineering. Examples include assessing the seismic limit-state capacity of steel moment-resisting frames [48], evaluating seismic fragility and vulnerability of reinforced concrete structures [48], analyzing the seismic response of steel buckling-restrained braced frames [20], reinforced concrete shear walls [21], and performing residual drift and seismic risk assessments of steel moment-resisting frames considering soil-structure interaction [22]. Despite these advancements, challenges persist in using AI methods for structural analysis and design, including high computational demands and the difficulty of fully capturing the complexities of real-world structures [23]. Overcoming these challenges requires further research to enhance AI algorithms' efficiency and improve their ability to model and simulate complex structural behaviors under diverse conditions.

AI-based learning methods can excel at processing large volumes of data quickly and accurately, allowing them to identify patterns and efficiently find solutions that would be difficult for traditional methods. In addition, AI-based learning methods can easily adapt to new situations and incorporate data from multiple sources to determine the best solution. AI is especially valuable for establishing approximate solutions to problems where input-output datasets are available or can be obtained at a limited cost. The contribution of this research is to develop an AI framework designed to provide tools suitable for achieving fast and accurate structure analysis. The framework aims to improve structural analysis by integrating sample selection, dimensionality reduction, and optimization strategies, enhancing predictive performance. The study focuses on exploring the potential of these methods to provide reliable insights into structural responses, such as displacement and drift, under various loading conditions. Ultimately, the goal is to contribute to the design and assessment of large-scale structures, offering a tool that can improve safety, efficiency, and durability in structural engineering practices.

2. Conventional structural analysis methods

Nowadays, nearly the entire process of analysis and control for many engineering problems is based on developed engineering concepts and well-established rules, supported by a strong mathematical foundation. In general, solving the associated partial differential equations (PDEs) analytically is often impossible for many engineering problems. Instead, engineers rely on various

numerical approximation and discretization methods. Among these, the finite element method (FEM) is one of the well-known and widely accepted approaches. Additionally, graph-based and reanalysis techniques are two other prominent types of conventional methods. In FEM, the problem’s domain is divided into a countable and finite number of elements, each described by simple equations [24]. Various FEMs have been extensively developed and applied to engineering problems, enabling them to address complex problems; However, significant challenges remain in balancing computational efficiency and accuracy during calculations [25]. Consequently, FEMs may not be the ideal choice for iterative design proposes.

Graph-based Optimum Analysis (GOA) was developed by Kaveh and colleagues [26]. By leveraging concepts of symmetry or regularity within the framework of graph theory, they suggested some methods to enhance the analysis process [27]. Additionally, they addressed symmetric problems with minor deviations, such as small cutout modifications or the inclusion of non-regular members [26]. While graph-theoretical methods initially appear highly valuable for reanalyzing purposes, they face two key limitations: these methods are neither applicable nor advantageous for asymmetric problems and they have been designed for flexibility-based methods rather than stiffness-based ones. This is significant because stiffness methods, due to their simplicity and compatibility with computer programming, form the foundation of nearly all commercial and academic analysis software.

Reanalysis techniques have been developed to reduce computational loads when evaluating the responses of engineering problems that undergo changes in their condition. These techniques utilize the solution of the original problem without resolving the modified one [28]. Most reanalysis methods focus on the local (low-rank) modifications [29], often employing Sherman–Morrison–Woodbury (SMW) formulas, which are generally suitable for localized changes [30]. While a limited number of studies address large changes [31], these approaches typically involve a significant increase in computational efforts, and the accuracy of approximations is not guaranteed [32]. Moreover, reanalysis methods become impractical when modification involves a large number of variables.

In conclusion, while modern FEM methods deliver reliable results and are widely used to predict structural behavior under various loads, their high computational cost remains a significant drawback. Similarly, although GOA and reanalysis techniques can reduce computational efforts through certain simplifications, they are neither sufficiently reliable nor broadly applicable making them impractical for large-scale structures or complex nonlinear conditions. These limitations present considerable challenges when using conventional methods for redesign or optimization problems, which typically require iterative processes. Additionally, traditional methods rely on predefined models and require detailed structural parameters, limiting their adaptability to diverse scenarios. Furthermore, manual adjustments by engineers introduce inefficiencies and potential errors. While machine learning methods have been introduced to enhance predictive capabilities, they often lack interpretability, making it difficult for engineers to understand the reasoning behind predictions.

The proposed framework addresses these challenges by integrating FEM with ML techniques, improving efficiency, flexibility, and interpretability. By leveraging ML, computational time is significantly reduced, enabling rapid predictions without repeated FEM simulations. The framework introduces sample selection and dimensionality reduction techniques to streamline data processing while maintaining predictive accuracy. To enhance transparency, Shapley Additive Explanations (SHAP) are applied, providing a clear understanding of how different features influence model predictions. By combining these advancements, the framework offers a more efficient and adaptive solution for structural analysis. It not only improves computational efficiency but also provides engineers with a better understanding of structural behavior through interpretable machine learning models. This approach ensures that structural designs can be analyzed more effectively, making it a valuable tool for real-world engineering applications where accuracy, scalability, and explainability are essential.

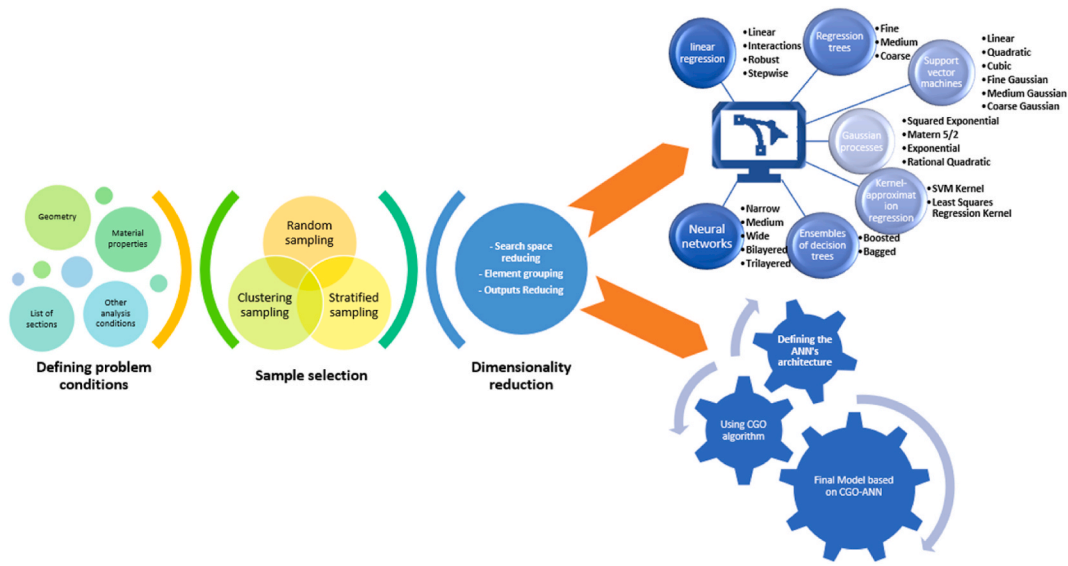


Fig. 1. An overview of the developed framework.

3. Methodology

This research aims to develop a robust framework for efficient and accurate structural analysis tools that integrate advanced methodologies to facilitate the analysis of large-scale structures, enabling rapid access to structural responses. The proposed framework bridges the gap between traditional structural analysis methods and the increasing demand for more efficient, accurate, and iterative design processes in engineering. While traditional methods, such as FEM, can be computationally expensive and time-consuming, especially for large-scale structures and optimization tasks, our approach combines structural analysis with advanced machine learning (ML) techniques to address these challenges. By leveraging these techniques, the framework establishes a data-driven approach to understanding the relationship between structural analysis inputs and outputs. Fig. 1 illustrates an overview of the developed framework.

The following outlines the key steps in developing this framework:

- A) **Defining Problem Conditions:** The primary objective is to create a robust technology that provides structural analysis results suitable for use in an iterative process. For this purpose, elements' sections are taken as inputs, while other parameters such as material properties, geometry, a list of sections, and other design conditions are considered constant.
- B) **Sample Selection:** To implement ML-based techniques, it is essential to generate initial datasets containing both inputs and outputs. In this study, we propose various sample selection techniques to achieve this effectively.
- C) **Dimensionality Reduction:** Practically, many elements have the same sections, simplifying implementation and control during construction. For design optimization, it is crucial to group elements with identical sections into "Design groups". This helps reduce the size of the problem's inputs and to achieve this, we can employ effective grouping techniques. Furthermore, the number of outputs is minimized based on structural conditions.
- D) **Developing Machine Learning Technique:** A dataset of inputs and outputs obtained from finite element simulations is used to train different machine learning techniques. Once trained, the ML model can predict the behavior of future structural designs without the need for computationally expensive finite element simulations. To improve accuracy, we have developed an advanced machine-learning model in addition to 26 other applied techniques.
- E) **Model Interpretability:** As the final step, SHAP (Shapley Additive Explanations) was incorporated as the interpretability technique to quantify each feature's contribution to the model's predictions. This analysis helps engineers identify the most

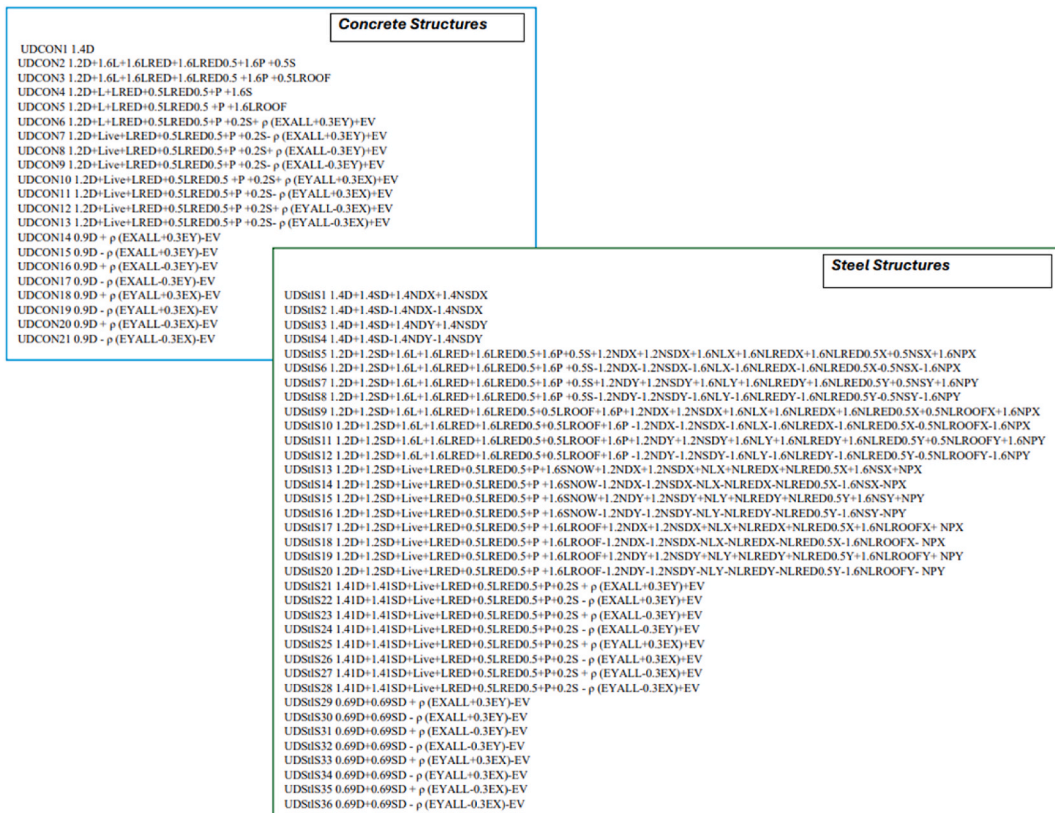


Fig. 2. Typical load combinations for structural design according to design codes.

significant features influencing structural behavior, providing valuable insights into critical design parameters necessary for ensuring safety and performance.

The following subsections describe these steps in more detail.

3.1. Defining Problem Conditions

In the analysis and design of a structure, problem conditions refer to the set of requirements and constraints that must be met to ensure the structure is safe, functional, and efficient. These conditions typically include factors such as loading conditions, material properties, geometric constraints, and analysis/design objectives. By defining the problem conditions, engineers can gain a comprehensive understanding of the search space and the constraints that need to be considered when developing a final design proposal. This information is crucial for the development of a successful analysis approach and ensures that the structure meets all the requirements for its intended use.

For large-scale structures, it is often assumed that the geometry and load conditions are defined and constant, and the material properties are determined. However, our framework differs from the classical approach in defining the load conditions: In the practical designing process, it is essential to consider the load combinations recommended in the relevant design code. For instance, the ACI code for concrete buildings [33] and the AISC code [34] for steel structures emphasize the need to consider combinations of seismic, dead, and live loads. These combinations ensure the structure is designed to withstand all possible scenarios of applied forces, as illustrated in Fig. 2. The number of possible combinations for load conditions can exceed 20 and developing models for each of these conditions is time-consuming. To address this challenge, this paper redefines the load conditions for developing the framework. We define the concept of Initial Load Patterns (ILPs), where by combining ILPs, the final response to a specific load combination can be derived. This approach reduces the number of load conditions to six for building structures, including four lateral loads (force and moment in two directions within each story), along with dead and live loads as main ILPs. This simplification is particularly advantageous for ML-based methods, as it reduces the problem complexity, making it easier to achieve good performance.

3.2. Sample selection

Sample selection is a critical step for training machine learning models, yet it often receives limited attention in previous studies on applying ML techniques to structural analysis. The primary goal of sample selection is to choose a representative subset of data from a larger dataset for the training process. This allows the model to learn from data that accurately reflects the broader population from which the data is drawn. Moreover, sample selection can help reduce the size of the dataset, saving time and computational resources. Selecting the right samples is essential, as it directly impacts the accuracy and generalizability of the model. If the chosen samples do not represent the overall distribution of the data, the model may perform poorly on new, unseen data. Conversely, selecting too many samples can lead to longer training times and overfitting, reducing the model's ability to generalize effectively.

There are several methods for selecting samples, each with its advantages depending on the specific requirements of the problem. To highlight the importance of sample selection in solving engineering analysis problems, we utilized three well-known techniques: random sampling, stratified sampling, and clustering. Random sampling involves selecting a subset of data points at random, ensuring simplicity but potentially lacking representativeness. Stratified sampling, on the other hand, selects samples from different classes or groups to ensure adequate representation from all categories. Clustering groups similar samples together, aiming to create a more

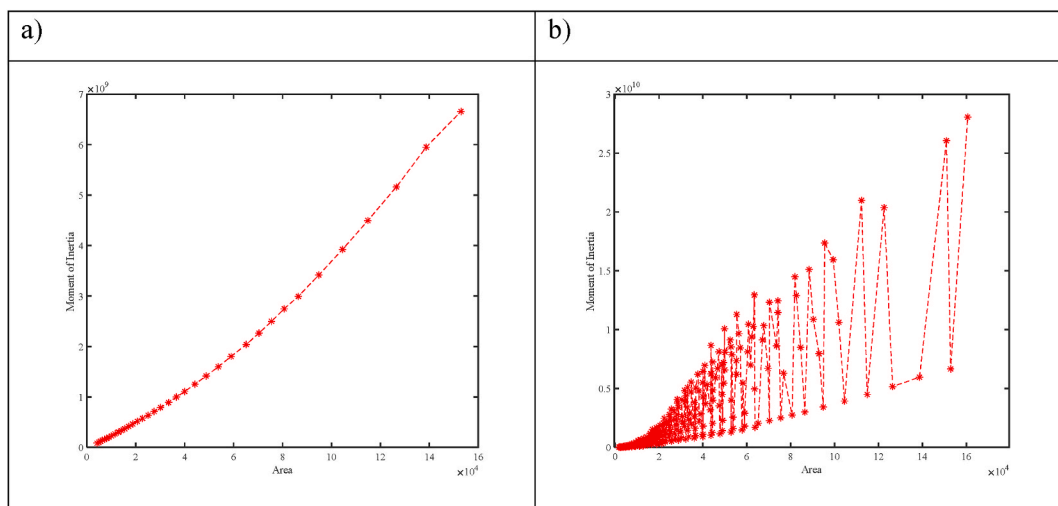


Fig. 3. Area-Moment of Inertia for steel W-shaped sections.

diverse and well-rounded sample set. All three samples are examined and compared in this study to evaluate their impact on the performance of machine learning models for structural analysis.

3.3. Dimensionality reduction

Dimensionality reduction plays a pivotal role in the success of ML-based methods. In structural design, the process of reducing dimensionality can generally be divided into three key steps.

- **Reduce search space by updating the list of sections:** The number of available sections in structural design plays a pivotal role in shaping the design space. In the realm of steel structures, the availability of W-sections defined by design codes provides us with a vast array of nearly 276 sections [34,35]. From a structural perspective, a certain section group, such as the W14 sections, possesses inherent characteristics that render them more suitable for specific applications, like columns, than others. To illustrate this concept, refer to Fig. 3, which depicts the relationship between the moment of inertia and the area of sections. It becomes evident that deriving a function to describe the moment of inertia for the W14 group (Fig. 3(a)) is a more straightforward endeavor than for the general case encompassing all sections (Fig. 3(b)). This realization allows engineers to strategically narrow down the potential sections for various structural elements, thereby streamlining the design process, and making the application of machine learning techniques to develop robust models more feasible.

In concrete structural design, the arrangement and size of rebars play a critical role in controlling element-specific constraints. As a result, engineers are required to define the section with at least a predetermined arrangement of rebars. However, rebars do not play a significant role in controlling the drift of structures. In such cases, sections without rebars can be utilized alone to achieve the desired results. This is because the main role of rebars is to resist tensile forces, whereas the drift of structures is primarily controlled by the strength of the concrete, which remains unaffected by the presence of rebars. Therefore, the presence of rebars does not necessarily need to be accounted for drift controlling. The ability to reduce the search space by utilizing general sections based solely on their dimensions, without delving into the intricate arrangement of rebars, presents a practical approach. For instance, the number of sections with different arrangements of rebars for square-shaped sections is equal to 540, while without considering rebars and their arrangement, only 18 sections exist, as presented in Fig. 4. The initial pair of numerals in Fig. 4 denotes the corresponding dimensions of the respective sections, while the subsequent pair represents the number of rebars oriented along the horizontal and vertical axes, respectively. Similarly, for rectangular-shaped sections that comply with design code constraints [33], the number of possible sections is significantly reduced—from 8346 when accounting for rebar sizes and arrangements to just 318 when rebar details are excluded. It should be noted that the size of the search space in an optimization process is equal to:

$$Largeness_{Space} = No_section^{No_element} \tag{1}$$






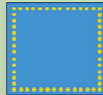
a)		b)	
Section Name Dimension-based (cm)	Section Shape	Section Name Rebar-based	Section Shape
35 X 35		35 X 35 – 3,3	
40 X 40		35 X 35 – 3,4	
.		.	
.		.	
.		.	
120 X 120		120 X 120 – 14,14	

Fig. 4. List of square-shaped section for concrete structures: a) dimension-based sections; b) rebar-based sections.

in which, $Largeness_{space}$ represents the size of the search space encompassing all possible designs for the structure; $No_section$ denotes the total number of defined sections and $No_elemnet$ indicates the total number of elements. Generally, it is evident that a smaller number of sections will lead to a more effective design with a reduced computational cost.

- Reduce the number of inputs by element grouping:

To reduce the dimensionality of structures, the second step involves minimizing the number of inputs by grouping elements. In structural analysis, each element is typically defined by several properties, such as material type, section shape, section size, and boundary conditions. However, not all properties have an equal influence on the structure's responses. Grouping elements based on their inherent properties can significantly reduce the number of inputs required for analysis and design. In data science, when a problem involves numerous features, it is a common practice to employ feature selection/extracting techniques aiming at identifying and selecting the most relevant features, thereby reducing complexity. Similarly, feature selection (specifically, feature grouping) and element grouping adhere to the same underlying concept, ultimately addressing the same problem. Grouping techniques for structural elements can vary depending on the specific application and design requirements [36], [37–40]:

- *Geometry-based grouping*: In this method, the elements of a structure are divided into distinct sets according to their geometry, and the elements within each set are grouped together. This technique particularly is useful for large structures with regular geometries where it helps streamline the analysis by organizing elements with similar geometric characteristics into manageable groups.
- *Functional grouping*: This technique involves grouping elements based on their function or role in structure. For example, elements that primarily resist bending forces, such as beams, can be grouped, while elements that primarily resist axial forces, such as columns, can be grouped separately.
- *Load-based grouping*: In this technique, elements are grouped according to the loads they are likely to experience. This grouping is typically determined through structural analysis, helping to align the design and analysis with the expected load distribution across the structure.
- *Hierarchical grouping*: This method groups elements based on their proximity to one another which can be achieved using clustering algorithms or other proximity-based methods. Defining the criteria for proximity is crucial, as it should take into account practical conditions and architectural requirements, to ensure the groupings are relevant and effective.

By reducing the number of inputs, the computational cost and complexity of analysis and design can be significantly lowered. This simplification can result in faster and more efficient analysis and design processes, while also minimizing the risk of errors and inaccuracies.

- Reduce the number of outputs by using problem-based conditions:

To create an efficient model for structures, reducing the number of required outputs is essential. Certain problem-based conditions can help achieve this goal. For instance, incorporating diaphragm properties into the analysis process can be effective. Diaphragms are crucial components of building structures, and interestingly, including them in the analysis can reduce the overall number of outputs required. Diaphragms act as continuous elements, allowing the analysis process to concentrate on their central points. By determining the response of the center of mass, the response in other degrees of freedom can be effortlessly determined, eliminating the need to analyze all degrees of freedom. This reduces the complexity of the defined framework and can lead to significant cost and time savings. By concentrating the analysis process on the diaphragms and their associated connections, structural responses can be accurately predicted.

3.4. Developing machine learning techniques

This paper utilizes 26 algorithms from various categories of machine-learning techniques to address the objectives of the research. These models include linear regression, regression trees, support vector machines, Gaussian processes, kernel-approximation regression models, ensembles of decision trees, and neural networks. The linear regression model uses linear equations where the output is a linear function of the input variables [41]. A regression tree is a tree-like structure that uses decision trees to analyze regression data [41]. In support vector machines, a hyperplane or set of hyperplanes in a high-dimensional space is constructed, [42]. As nonparametric models, Gaussian processes consist of a finite set of random variables with a joint Gaussian distribution [43]. A kernel approximation regression model uses a kernel function to map input variables into a high-dimensional space [44], which then approximates the output function as a linear combination of kernel functions. The ensemble of decision trees method involves a collection of decision trees, each trained on a different subset of the training data [45]. The ensemble output is the weighted sum of the outputs of the individual decision trees. Neural network models [42], inspired by the structure and function of the brain, consist of artificial neurons arranged in multiple layers connected by weighted links.

It is worth noting that many other methods, such as those developed by Refs. [19], can also be easily incorporated into the developed framework. In this study, the methods are selected based on the following considerations:

- **Diversity of Methods:** The selected machine learning methods for this study represent a wide array of approaches, each with distinct strengths. These include traditional models, such as support vector machines (SVM), decision trees, and neural networks, alongside more modern techniques, such as ensemble methods and the developed algorithm.
- **Complementary Strengths:** Each method provides complementary strengths depending on the nature of the data and the problem at hand. For instance, tree-based models excel at handling nonlinear relationships in data, while SVMs are particularly effective for high-dimensional spaces. Neural networks, on the other hand, offer robust performance for complex, large datasets, and are capable of capturing intricate patterns that might be missed by traditional methods.
- **Model Flexibility:** The proposed framework allows engineers to select the most appropriate method based on the specific characteristics of their structural data, such as the number of features, the size of the dataset, and the desired prediction accuracy. This flexibility ensures that the chosen method is optimized for the problem, improving the efficiency and accuracy of the analysis.
- **Validation Across Metrics:** The methods were validated against a comprehensive set of performance metrics, including prediction accuracy, speed, and robustness. This thorough evaluation demonstrated the advantages of each method in different scenarios, confirming their applicability to various types of structural analysis tasks. By comparing their performances, we ensure that the most suitable method is applied to each problem.
- **Improvement Potential:** Combining or modifying existing methods can significantly enhance the accuracy and robustness of predictions. For instance, this study introduces a novel approach by integrating Chaos Game Optimization [7] with ANNs, which markedly improves results and facilitates more effective structural analysis outcomes. This example demonstrates the potential for numerous other efficient improvement processes to be explored and implemented in the developed framework.

The primary goal of this study is to develop a general framework in which various machine-learning methods can be applied to predict the behavior of large-scale structures. This framework is designed to be flexible, allowing researchers and engineers to choose the most appropriate method for their specific problem based on the characteristics of the data and the desired outcomes. Each of the 26 methods used in this study offers distinct properties, making them suitable for different types of structural analysis problems:

- **Linear Regression Models (LR):** Best for linear or near-linear relationships, LR provides simple and interpretable solutions, capturing key factors affecting structural behavior and handling more complex interactions with added terms.
- **Regression Trees (RT):** Effective for modeling non-linear relationships, RTs offer clear interpretability and versatility with continuous and discrete variables, making them suitable for diverse structural analysis tasks.
- **Support Vector Machines (SVM):** SVMs excel in high-dimensional problems, handling distinct boundaries well, and are ideal for structural tasks like failure mode classification or material behavior analysis.
- **Ensemble Methods (EM):** Robust for large, complex datasets, EMs (boosted and bagged) reduce bias and variance, adapting well to noisy data and effectively capturing non-linear relationships in structural analysis.
- **Gaussian Process Regression (GPR):** GPR is optimal for tasks requiring smooth predictions and uncertainty quantification, such as safety or risk assessment in structural analysis.
- **Artificial Neural Networks (ANN):** ANNs handle highly non-linear, complex datasets, excelling in dynamic load modeling and intricate structural behaviors, with performance relying on proper architecture and training.

In addition to the aforementioned technologies, this paper presents an enhanced artificial neural network model. By integrating Chaos Game Optimization (CGO), a simple yet powerful optimization algorithm [7], [46], with ANNs a more powerful and efficient ML-based method is developed. This process unfolds across several levels, as described below.

-First Step: Defining the ANN's architecture

The first step is to define the ANN's architecture, including the number of layers, the number of nodes in each layer, and the activation functions. The next step involves defining the fitness function, which will be used to evaluate the performance of the ANN. Fitness functions are based on metrics such as accuracy, error rate, or other performance measures.

-Second Step: Optimization

Once the ANN architecture and fitness function are defined, the optimization process begins. Initially, a random set of solutions is generated, with each solution representing a set of weights and biases that govern the ANN's behavior. The performance of these initial solutions is subsequently evaluated using the fitness function.

The CGO algorithm begins its iterative process following the evaluation of the initial solutions. During each iteration, a new set of solutions is generated by applying the core principles of the CGO algorithm as follows [7]:

$$Seed_i^1 = X_i + \alpha_i \times (\beta_i \times GB - \gamma_i \times MG_i), i = 1, 2, \dots, n. \quad (2)$$

$$Seed_i^2 = GB + \alpha_i \times (\beta_i \times X_i - \gamma_i \times MG_i), i = 1, 2, \dots, n. \quad (3)$$

$$Seed_i^3 = MG_i + \alpha_i \times (\beta_i \times X_i - \gamma_i \times GB), i = 1, 2, \dots, n. \quad (4)$$

$$Seed_{ij}^4 = rand(Lb_j, Ub_j) \quad i = 1, 2, \dots, n. \quad (5)$$

where, X_i represents the i th solution candidate, GB denotes the best solution found globally so far, and MG_i is the mean value calculated from a set of initial eligible points. α_i is the randomly generated factor that models the movement limitations of the seeds while β_i and γ_i are random integer (either 0 or 1) that simulate the probabilistic nature of rolling a die. $rand(Lb_j, Ub_j)$ generate random values for dimension j within the range of Lb_j and Ub_j .

To evaluate the new set of solutions, the fitness function is applied after their generation. In the subsequent step, the best-performing solutions are selected and used to generate the next set of solutions. This process is repeated until the fitness function achieves a specified improvement or the algorithm reaches the predefined number of iterations.

CGO's ability to efficiently explore the solution space allows it to converge to high-quality solutions more rapidly. Furthermore, it is not only effective but also relatively straightforward to implement. The algorithm relies on simple mathematical operations and does not require specialized hardware or software, making it accessible and practical for a wide range of applications.

3.5. Model interpretability

Understanding and interpreting machine learning models is crucial, especially in high-stakes applications like structural engineering, where decision-making relies on accurate and explainable predictions. Model interpretability ensures that engineers, researchers, and stakeholders can trust the model's outputs and understand the rationale behind predictions. Without proper

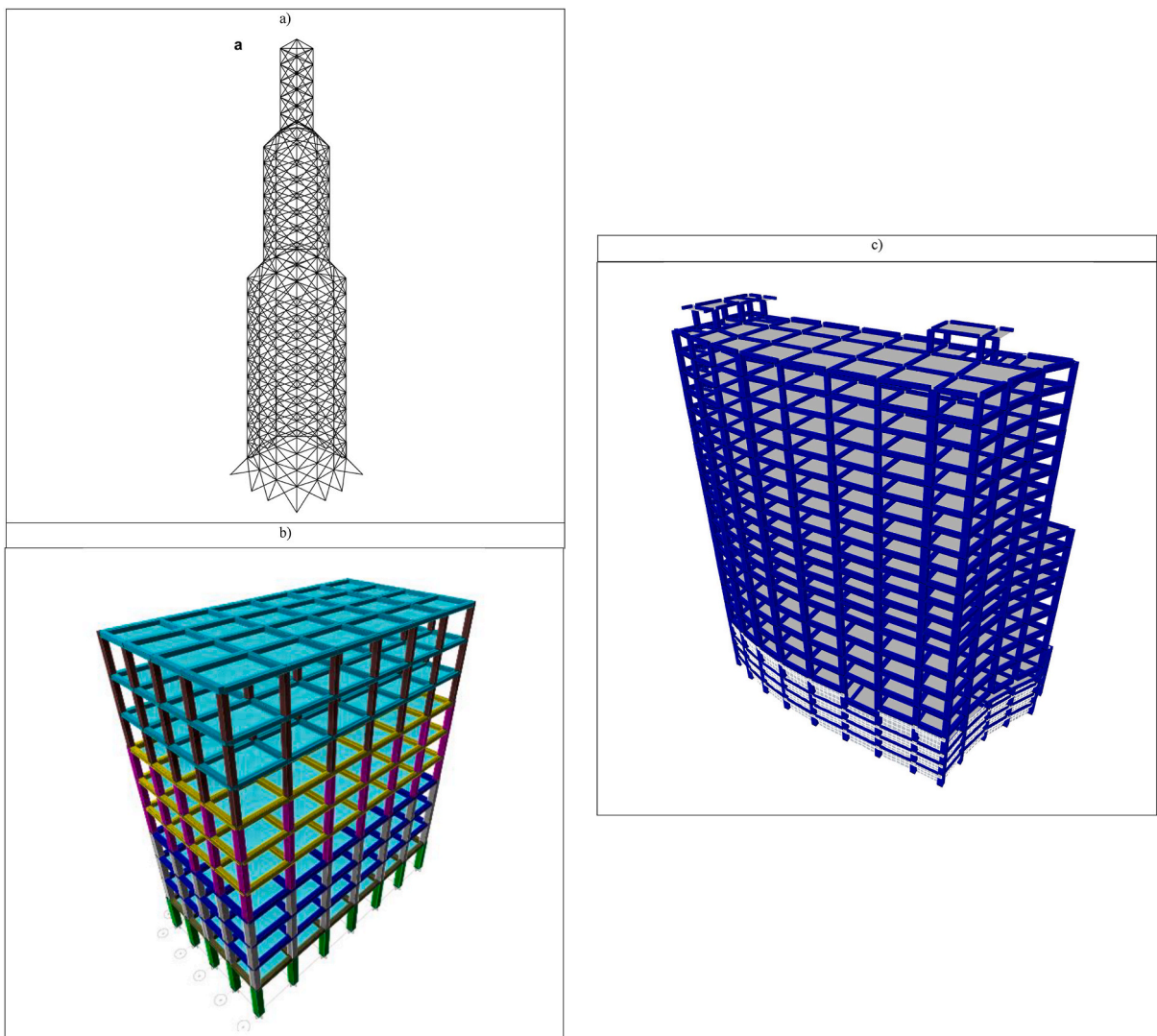


Fig. 5. 3D view of the structures: a) 26-story tower spatial truss structure, b) 10-story concrete building, c) 22-story steel building.

interpretability, even highly accurate models may be difficult to validate, leading to skepticism regarding their reliability and practical applicability. Several techniques exist for enhancing the interpretability of ML models, each offering unique insights into how models make decisions. Feature importance methods, such as permutation importance and gradient-based attribution techniques, help identify the most influential features. Partial Dependence Plots (PDP) and Individual Conditional Expectation (ICE) plots illustrate how specific features influence predictions. Local interpretable model-agnostic explanations (LIME) approximate complex models with simpler interpretable models to provide local explanations. While these techniques offer valuable insights, they often have limitations in capturing global feature importance and interactions comprehensively.

Among these techniques, Shapley values, rooted in cooperative game theory, provide a theoretically sound and mathematically rigorous approach to model interpretability. Unlike other methods, SHAP (Shapley Additive Explanations) ensures fair and consistent attribution of each feature's contribution by considering all possible feature combinations. This method helps quantify how much each feature contributes to a model's prediction compared to a baseline, offering a transparent and reliable explanation of model behavior [47]. SHAP is particularly advantageous because:

- It considers feature interactions, allowing for a deeper understanding of how multiple features jointly influence predictions.
- It provides both local and global interpretability, meaning it can explain individual predictions as well as overall feature importance.
- It ensures fairness and consistency, as Shapley values adhere to fundamental properties like additivity and symmetry, making them more reliable than heuristic-based importance measures.

In this study, we employ SHAP to analyze feature contributions in the developed framework, allowing us to interpret the model's decision-making process with precision. The computed Shapley values help identify the most and least influential features, guiding optimization efforts and enabling data-driven decisions in structural engineering. By leveraging SHAP, we enhance the model's transparency, making its predictions more interpretable, trustworthy, and practically useful in real-world applications.

4. Numerical investigation

4.1. Case studies

This section presents three real-size structures utilized to evaluate the effectiveness of the developed framework. The first design example is a 26-story-tower spatial truss structure consisting of 942 elements and 244 nodes. The allowable cross-sectional areas for this example are continuous, ranging from 0.6452 cm² to 129.032 cm². The second example is a 10-story concrete building which includes 350 columns and 580 beams. Lastly, the third example is a 22-story steel building, comprising 2990 elements. The schematic views of these examples are illustrated in Fig. 5.

There are numerous metrics for evaluating the performance of machine learning methods, as discussed in Refs. [19]. However, in this study, we used four well-known metrics, which can easily be substituted with others if needed. To generalize the developed framework, one might need to consider additional evaluation metrics, as outlined in Refs. [19]. These metrics are: the coefficient of determination (R-squared or R²), root mean squared error (RMSE), Mean Absolute Error (MAE), and Mean Squared Error (MSE). R-squared measures how well the model fits the data, with values ranging from 0 (no fit) to 1 (perfect fit). RMSE quantifies the average magnitude of prediction errors, with lower values indicating greater accuracy. MAE calculates the average absolute difference between actual and predicted values, while MSE measures the average squared differences, giving more weight to larger errors and making it sensitive to outliers. Together, these metrics provide a comprehensive assessment of model accuracy and fit.

The primary aim of this study is to establish a generalizable framework for structural optimization in the design process. For optimization, we typically follow an iterative process, focusing on reaching a near-optimal solution through a pseudo-dynamic process, as outlined in the manuscript. This method allows us to refine the solution progressively, ensuring computational efficiency. Once a near-optimal solution is reached, we can incorporate dynamic load considerations, as suggested in Refs. [19]. This stepwise approach ensures that the framework is adaptable to real-world dynamic conditions while optimizing computational resources, making it practical for large-scale structural analysis. The flexibility of our framework allows for its extension to nonlinear and dynamic scenarios. For example, machine learning models within the framework can be trained on datasets derived from nonlinear structural analyses, accounting for phenomena such as material nonlinearity, large deformations, or plasticity. Similarly, the framework can be expanded to accommodate dynamic load conditions (e.g., seismic or wind loads), enabling models to learn and predict responses under time-dependent forces.

The target parameters for prediction in our study include both the top-story displacement and the inter-story drifts, as these are critical indicators of structural behavior. Additionally, other important parameters, such as base shear, stress, and natural frequency, can also be considered for similar purposes. However, for this study, we focused on top-story displacement and inter-story drifts to demonstrate the effectiveness of the proposed framework. It is important to note that all data points (1500 data designs) utilized in this study were generated through finite element methods (FEM), making FEM the benchmark for evaluating the accuracy and effectiveness of the proposed machine learning models.

4.2. Results and discussion

This section summarizes our findings and demonstrates how each example contributes to evaluating different aspects of the

developed framework. Specifically:

- The first example is used to examine the influence of the feature selection mechanisms in the framework.
- The second example investigates the performance of the developed algorithm compared to other methods. Additionally, this example is used to study sample selection techniques.
- Finally, the third example explores the importance of dimensionality reduction in improving the framework's efficiency and accuracy.

These examples collectively provide a comprehensive assessment of the proposed framework's components and their contributions to its overall effectiveness.

4.2.1. Feature selection

In this section, the data set from the first example is utilized. Fig. 6 illustrates the steps involved in this part of the framework:

- **Grouping Elements:** In the initial step, elements are grouped based on engineering knowledge and considering the geometry, load conditions, and functional aspects of the structure.
- **Data Generation:** Using advanced finite element method techniques, the initial dataset is generated capturing the behavior and characteristics of the structure under various conditions.
- **Data Division:** The generated dataset is then divided into two subsets: training and test data sets. These subsets remain consistent across all scenarios.
- **Feature Selection:** The final step before developing machine learning techniques involves applying feature selection methods. Four different feature selection scenarios are considered:
 - **Minimum Redundancy Maximum Relevance (mRMR):** This method selects features that are highly relevant to the target variable while minimizing redundancy among them. It calculates relevance and redundancy scores for each feature and selects those with the highest relevance scores and lowest redundancy scores for further analysis or modeling.
 - **Relief Method:** This method evaluates the correlation between all features and selects those with low correlation scores. This ensures the selected features are independent and effectively represent the dataset's characteristics.
 - **F-test Method:** The F-test method assesses whether a statistically significant difference exists between selected feature groups. It compares the variance between the feature groups with the variance within each group, determining if the feature values vary significantly in relation to the target variable. Features with higher F-statistics are deemed more relevant for the model, while those with lower F-statistics are considered less important and may be excluded.
 - **All Features:** In this scenario, all features derived from the engineering grouping step are included without any additional selection.

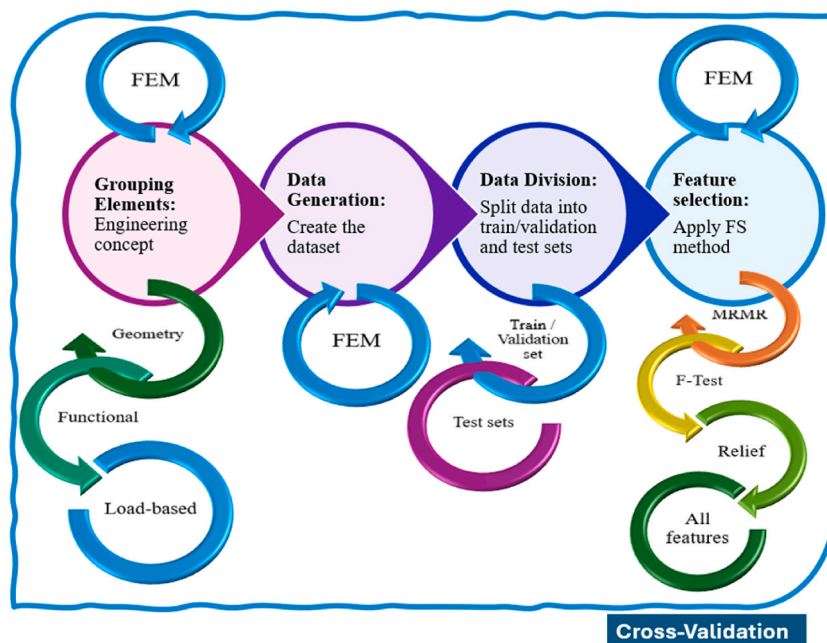


Fig. 6. Steps for feature selection methods.

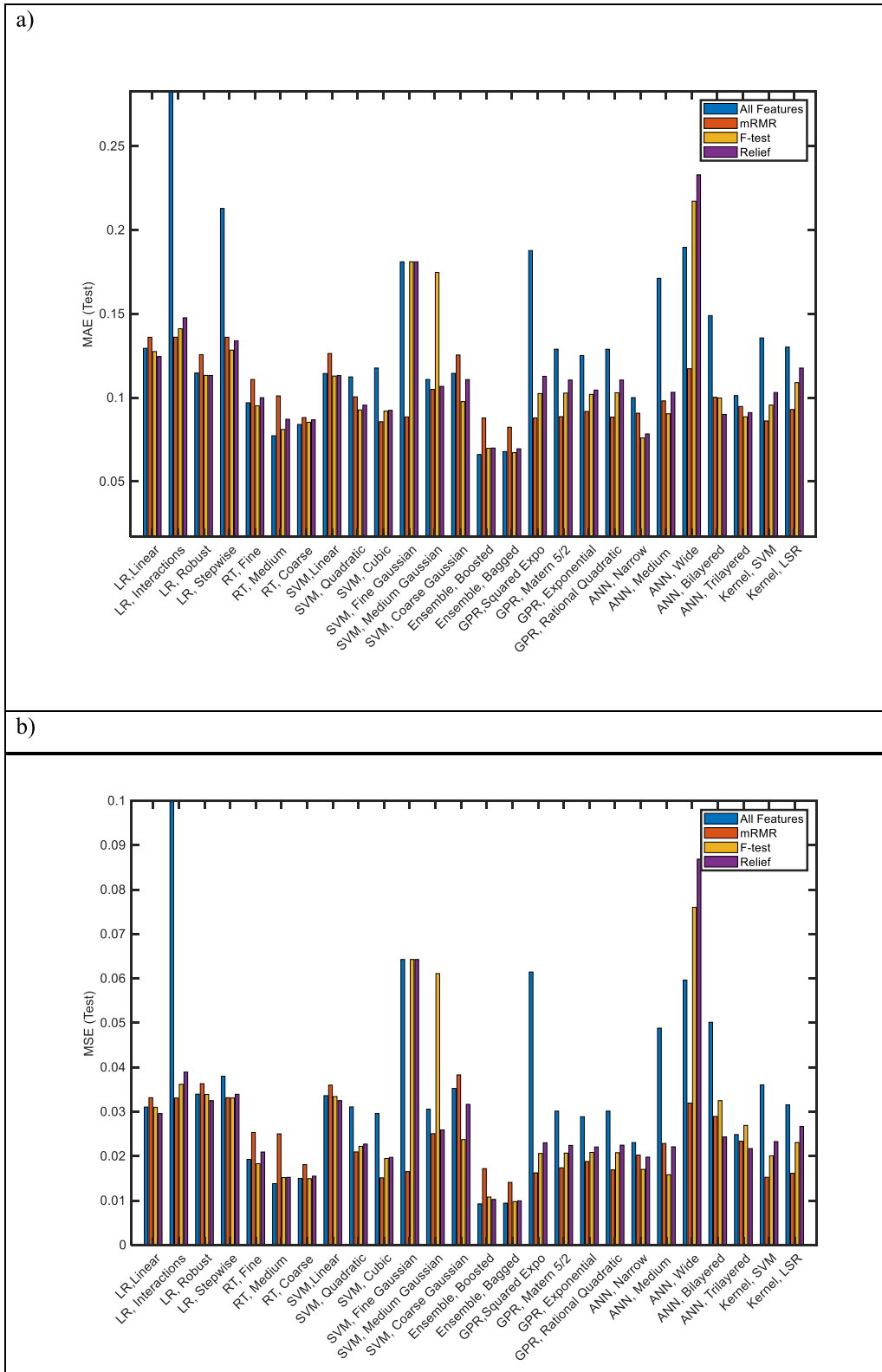


Fig. 7. The test results for different feature selection scenarios: a) MAE, b)MSE, c)RMSE, d)R-squared.

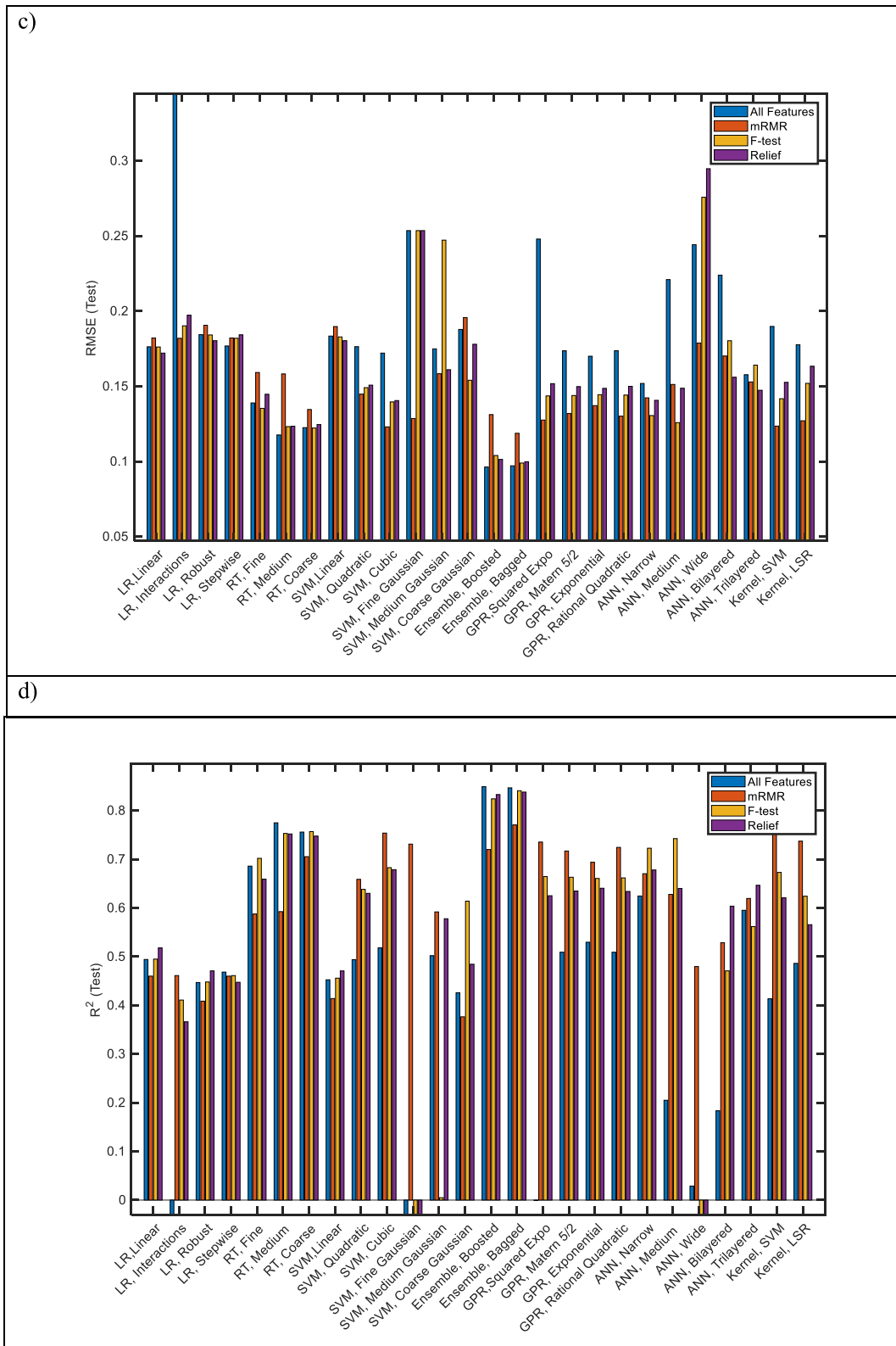


Fig. 7. (continued).

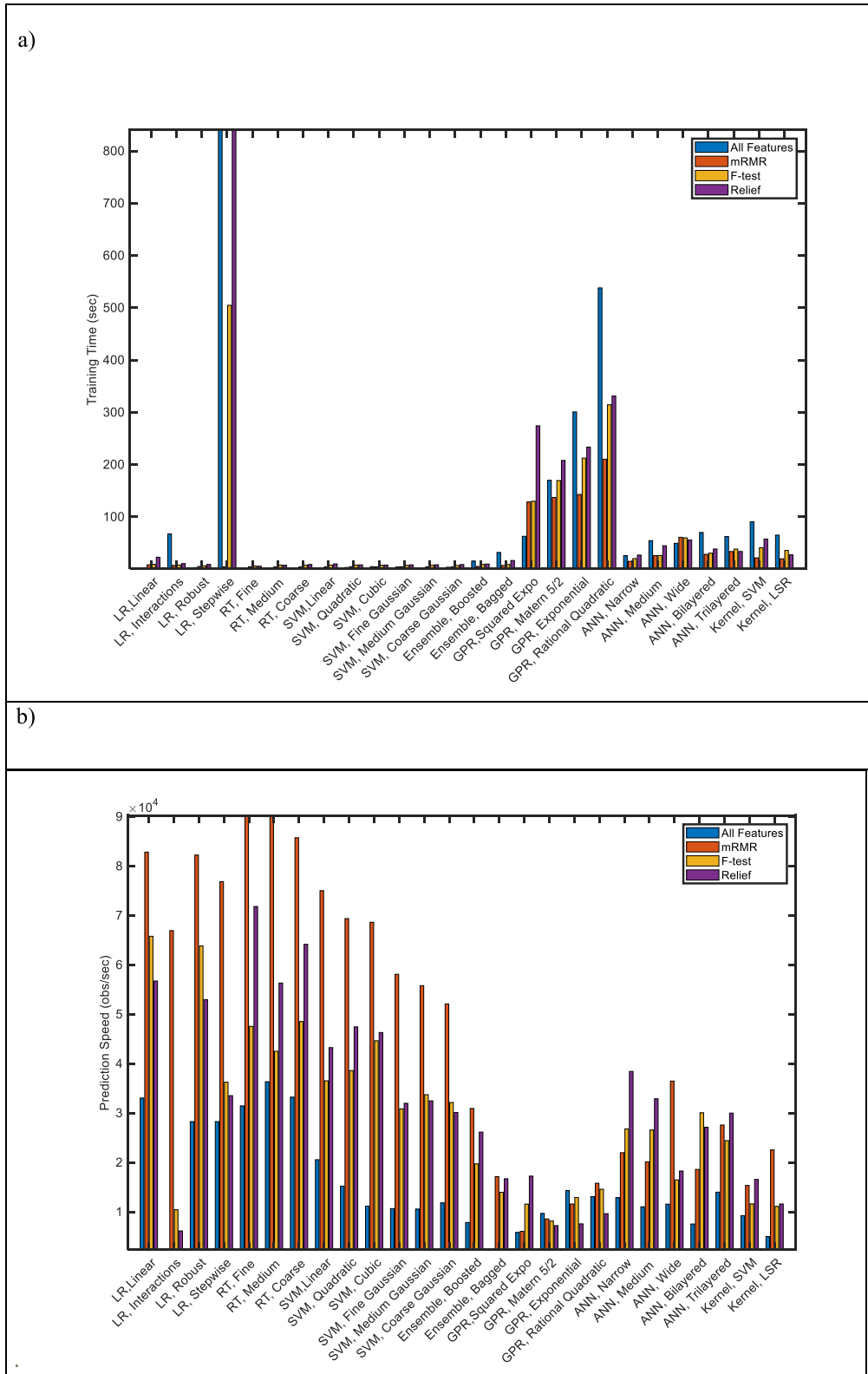


Fig. 8. a) The training time and b) prediction speeds for different feature selection scenarios.

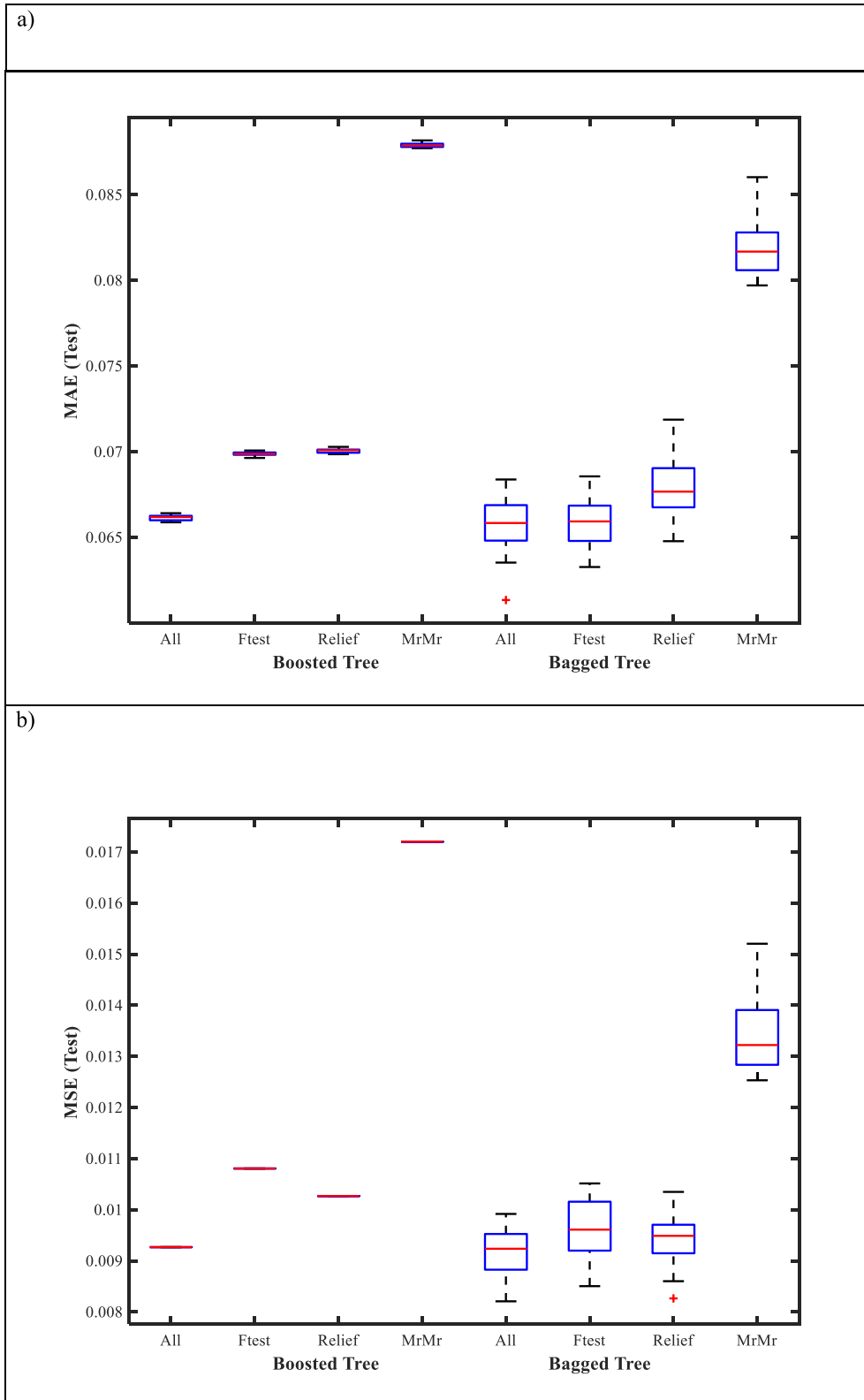


Fig. 9. Statistical study obtained by Ensembled methods using different feature selection scenarios.

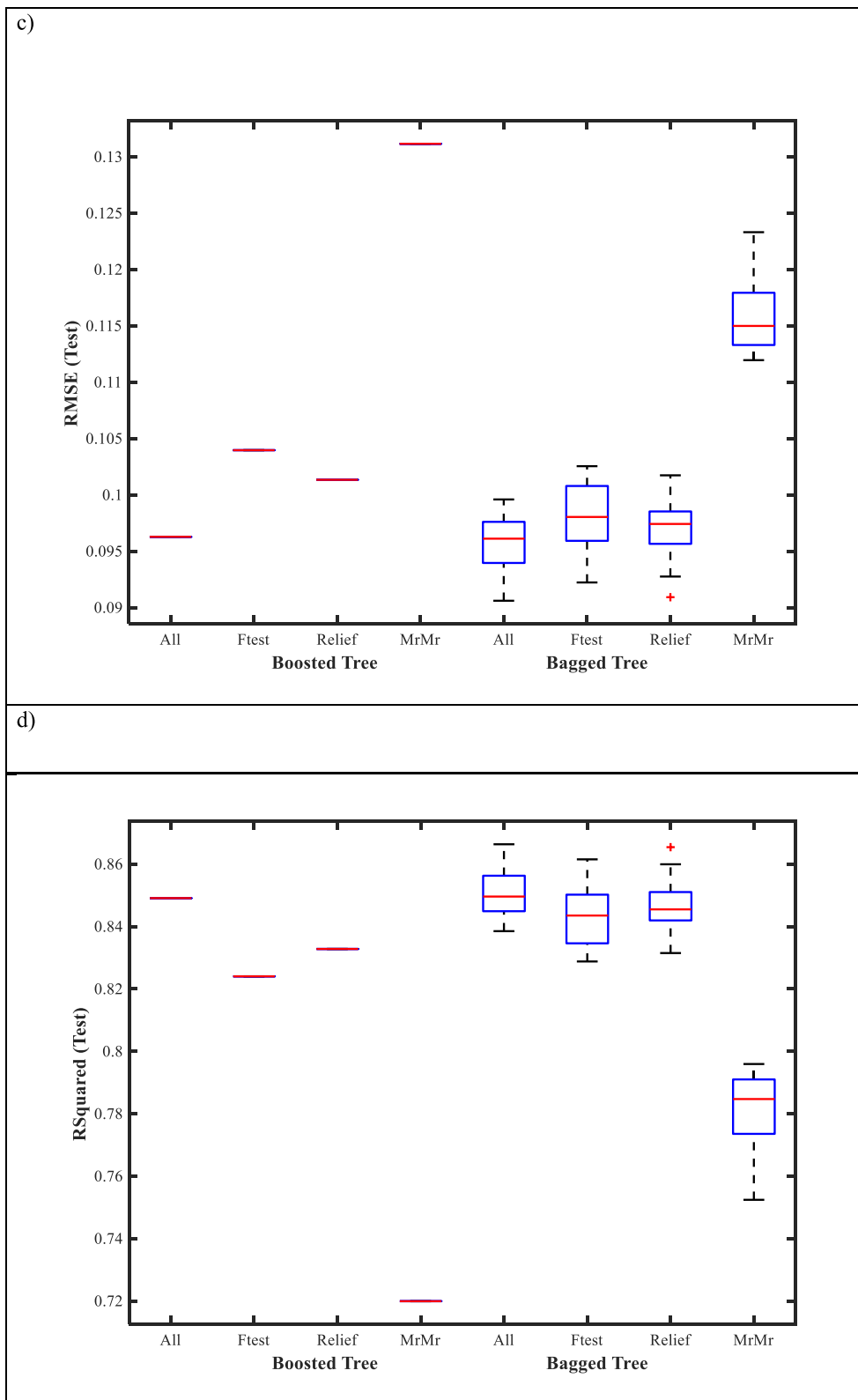


Fig. 9. (continued).

- **Cross-Validation:** To ensure a robust and fair comparison, cross-validation is employed, further enhancing the reliability of the performance evaluations.

The test results of these four feature selection scenarios are presented in Fig. 7. The bar charts in the figure provide a detailed comparison of the MAE(6a), MSE (6b), RMSE (6C), and R-squared (6d) for different machine learning models. In brief, including all features without selection leads to suboptimal performance due to irrelevant or redundant features introducing noise and complexity. In contrast, feature selection methods (mRMR, F-Test, Relief) result in lower error values, highlighting the importance of selecting relevant features. Narrowing the feature set to the most relevant ones improves model accuracy and minimizes prediction errors.

The model-specific Observations indicate that feature selection consistently improves performance across various models. For linear regression models, including the standard Linear Interactions, Robust, or Stepwise Regression, feature selection leads to improvements in error metrics. This process is especially advantageous for the Interactions linear regression model, as it reduces noise and focuses on the most relevant features, resulting in notable performance improvements (approximately 85 %, 75 %, and 84 % improvement for MAE, MSE, and RMSE, respectively). Regression tree models, such as Fine, Medium, and Coarse, show either maintenance or improvement in error metrics, with the Relief method often yielding the best results. Support Vector Machines (SVM) exhibit a significant reduction in error metrics, especially when fine and medium Gaussian kernels are used. The most significant improvement is observed with mRMR, where errors are reduced by 27 %, 18 %, and 32 % for cubic SVM, and by 51 %, 28 %, and 49 % for Fine Gaussian SVM in terms of MAE, MSE, and RMSE, respectively. When feature selection methods are applied, boosted and bagged ensemble methods generally exhibit slight increases in error metrics, except for mRMR-based models. Specifically, for boosted (and bagged) ensemble methods, mRMR results in increases of 24 % (17 %) for MAE, 46 % (33 %) for MSE, and 15 % (18 %) for RMSE. In contrast, other FS-based methods show much smaller increases, limited to a maximum of 5 %, 7 %, and 9 % for MAE, MSE, and RMSE, respectively, highlighting the relatively stable performance of non-mRMR approaches. Gaussian Process Regression (GPR) models benefit from feature selection, although the extent of improvement varies across kernels. For example, the improvements for GPR Squared Expo are 46 %, 67 %, and 43 % on average for MAE, MSE, and RMSE, respectively. Artificial Neural Networks (ANN) show mixed results, with certain architectures like the Quadratic model experiencing substantial improvements, while others, such as Medium and Narrow models, show more modest gains. Finally, the performance of kernel methods improved across almost all metrics and for all feature selection methods. Among these, mRMR achieved the best results, with improvements of 36 %, 57 %, and 35 % for MAE, MSE, and RMSE, respectively.

For a more detailed comparison and to gain valuable insights, we examine the performance of three feature selection methods. When comparing the FS methods, mRMR consistently yields the lowest error values across several models and metrics, demonstrating its effectiveness in selecting the most relevant features while minimizing redundancy. F-Test, on the other hand, provides competitive results, often closely aligning with mRMR, suggesting that statistical significance testing is a strong and reliable method for feature selection. Relief, while performing well in most cases, may not always achieve the same level of error reduction as mRMR or F-Test.

The mean improvement in the MAE metric is 35 %, 31 %, and 30 % for the mRMR, F-Test, and Relief methods, respectively, when compared to the model without any feature selection method. For the MSE metric, the improvements are even more pronounced, with values of 74 %, 68 %, and 70 %, respectively. Similarly, for the RMSE metric, the mean improvements are 30 %, 26 %, and 25 %. These results indicate that feature selection significantly enhances the model's prediction accuracy across all evaluated metrics.

Additionally, the improvement in the standard deviation (STD) of the results highlights the increased reliability of predictions. For the MAE and RMSE metrics, the reductions in STD are approximately 89 %, 80 %, and 81 % for mRMR, F-Test, and Relief, respectively. For the MSE metric, the STD reductions are even higher, at 97 %, 94 %, and 95 %. This substantial decrease in variability across multiple runs demonstrates that the models with feature selection methods not only provide more accurate predictions (as evidenced by the mean values) but also produce more consistent and dependable results. In summary, the significant improvements in both the mean values and the standard deviations indicate that feature selection methods are crucial for enhancing the performance and reliability of machine learning models. The reduced variability is especially important in practical applications, where consistent performance across different scenarios is critical. Among the methods evaluated, mRMR consistently delivers the most substantial improvements, further emphasizing its robustness in selecting relevant features and eliminating redundancies.

Fig. 8 illustrates the training time and prediction speed of all models across four predefined FS scenarios. The results indicate that sample selection generally enhances the training process by reducing the number of features which often leads to faster training speeds. On average, training time decreases by 65 %, 34 %, and 12 % for the mRMR, F-Test, and Relief methods, respectively, compared to the model with all features. Conversely, prediction speed for these methods increases by nearly 100 % compared to the model without feature selection. It is worth noting that while the prediction time remains relatively insignificant compared to the original finite element method (FEM), the increase in prediction time is not particularly critical, given the importance of other parameters such as accuracy. Moreover, the prediction process is still significantly faster than the original FEM, enabling real-time or near-real-time analysis, which is essential for applications like rapid design iterations. To sum up, the analysis of error metrics across different models and FS methods confirms that applying feature selection methods can significantly improve the performance of machine learning models. Furthermore, feature selection methods are straightforward to implement and help reduce the computational complexity of the model. Therefore, incorporating feature selection should be a key consideration when developing efficient, accurate, and computationally optimized machine learning models. a)

The statistical analysis of Ensemble models (both Bagged and Boosted), which demonstrate the best performance among all methods, is illustrated in Fig. 9. The results reveal that the range of solutions across different runs remains relatively consistent even after implementing feature selection methods, indicating that the robustness of the algorithms is preserved. This consistency underscores the reliability of the models, suggesting their resilience to variations introduced by feature selection techniques.

4.2.2. Number of samples

Another crucial consideration in developing machine learning models is the number of samples. In structural design problems, where samples are often generated using computational approaches like FEM, there is a preference for a limited number of samples to reduce computational costs. However, a small sample size can adversely affect the model's performance, especially in the current problems where high accuracy is critical.

To investigate this important parameter, we defined various scenarios with sample sizes ranging from 500 to 5000 in increments of 500, resulting in 10 different scenarios. For evaluation, we employed multiple metrics as discussed in Kazemi et al. [21]. Table 1 summarizes these metrics, providing an overview of the performance across scenarios. This analysis focused on the Bagged Ensemble model, identified as the best-performing method in the previous section.

In this study, we utilized Pareto Analysis and the Knee Detection Methods to identify the optimal sample size for training machine learning models. Pareto Analysis was applied to rank sample sizes based on their contribution to model performance, as measured by evaluation metrics as provided in Table 1. This helped identify the most influential sample sizes that accounted for the majority of performance improvements, ensuring a focus on the most impactful configurations. Additionally, the Knee Detection method was employed to pinpoint the "knee point" in performance curves, where increasing the sample size yields diminishing returns. By combining these methods, we determined the ideal trade-off between computational cost and model accuracy, enabling efficient resource allocation while maintaining high performance.

According to Fig. 10 and based on the Knee Detection method, the scenario with 1500 samples emerges as the most optimal choice for the majority of performance indicators. Specifically, it yields the best results for key evaluation metrics, including MAE, MSE, RMSE, R-squared, MRAE, RRMSE, MBE, SD, and Un95. This indicates that 1500 samples provide a well-balanced dataset, ensuring both predictive accuracy and efficiency. For the remaining four out of thirteen indicators, other sample sizes—such as 2000 and 500—show slightly better performance. However, the differences between these scenarios and the 1500-sample case are relatively small, suggesting that increasing or decreasing the sample size beyond this point does not lead to substantial improvements in overall model performance.

Similarly, Fig. 11 demonstrates that the Pareto Analysis method also identifies 1500 samples as the most effective choice for the majority of evaluation metrics, reinforcing the findings from the Knee Detection method. While certain indicators show marginally improved results with alternative sample sizes, the variations remain insignificant. This suggests that deviating from the 1500-sample scenario does not yield considerable benefits and may introduce unnecessary computational complexity. Larger datasets might offer slight improvements for a few specific indicators but do not necessarily enhance performance for many others, while also increasing computational demands. On the other hand, smaller datasets could risk lower predictive reliability. The results emphasize that a moderate dataset size, such as 1500 samples, provides an effective balance between accuracy and efficiency, making it a practical choice for structural analysis applications.

Table 1
The utilized statistical indicators.

Description	Equation
Mean absolute error	$MAE = \frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $
Mean squared error	$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$
Root Mean squared error	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$
Coefficient Determination	$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$
Mean absolute relative error	$MARE = \frac{1}{n} \sum_{i=1}^n \left \frac{y_i - \hat{y}_i}{y_i} \right $
Mean square relative error	$MSRE = \frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{y_i} \right)^2$
Root mean squared relative error	$RMSRE = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{y_i} \right)^2}$
Relative root mean square error	$RRMSE = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}}{\sum_{i=1}^n y_i} \times 100$
Mean bias error	$MBE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)$
Maximum absolute relative error	$erMAX = \max \left(\frac{y_i - \hat{y}_i}{y_i} \right)$
Residual Standard Deviation	$SD = \sqrt{\frac{1}{n-2} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$
Uncertainty at 95 %	$Un_{95} = \sqrt{1.96(SD^2 + RMSE^2)}$
t-Statistic	$t-stat = \sqrt{\frac{(n-1)MBE^2}{RMSE^2 - MBE^2}}$

In which, y_i and \hat{y}_i are the actual and predicted values, respectively. n is the total number of data points.

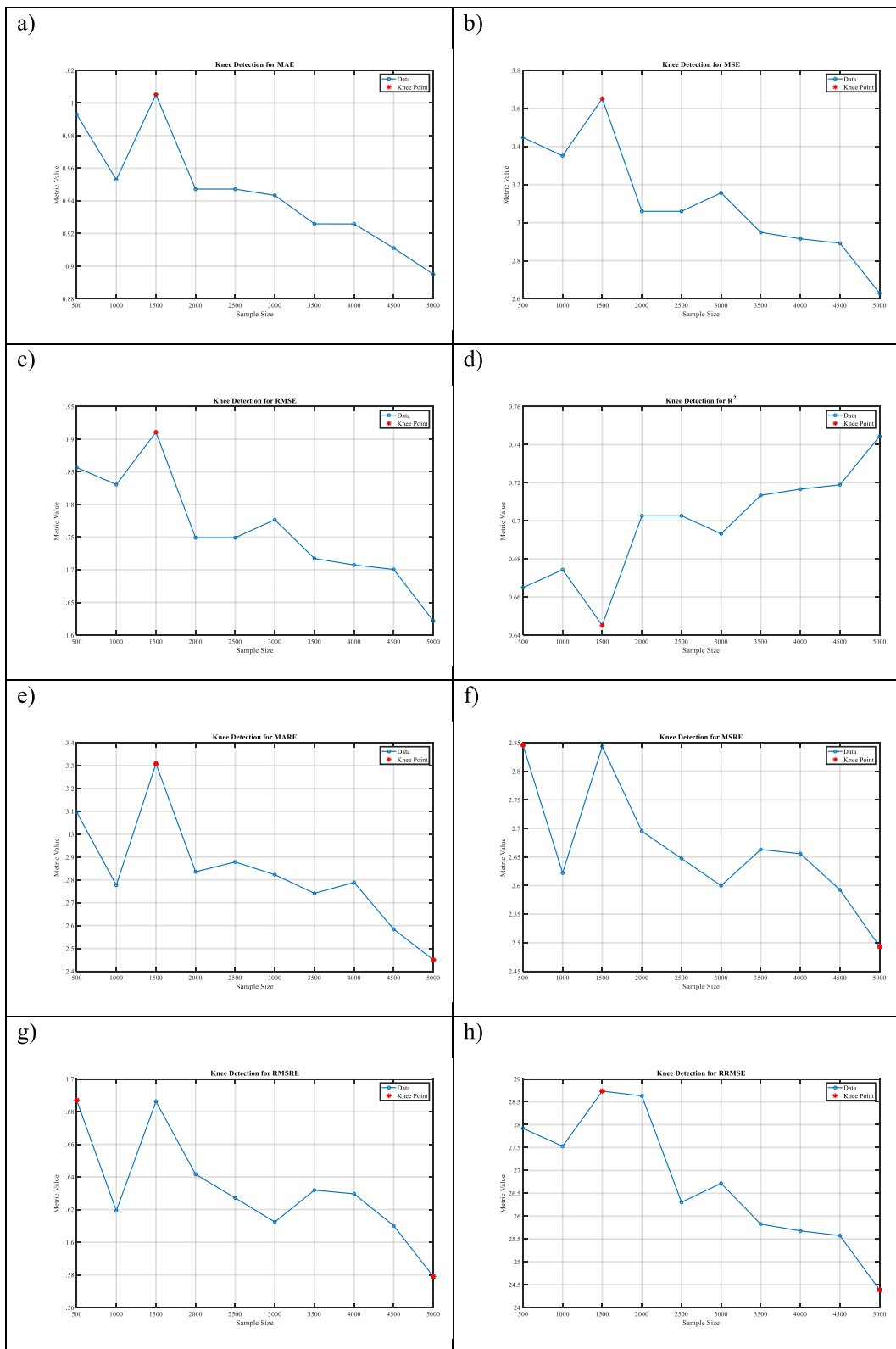


Fig. 10. Results for Knee Detection Method for a) MAE, b) MSE, c) RMSE, d) R² e) MARE, f) MSRE, g) RMSRE, h) RRMSE, i) MBE, j) erMAX, k) SD, l) Un95, m) t-stat, n) Prediction Speed, o) Training time a).

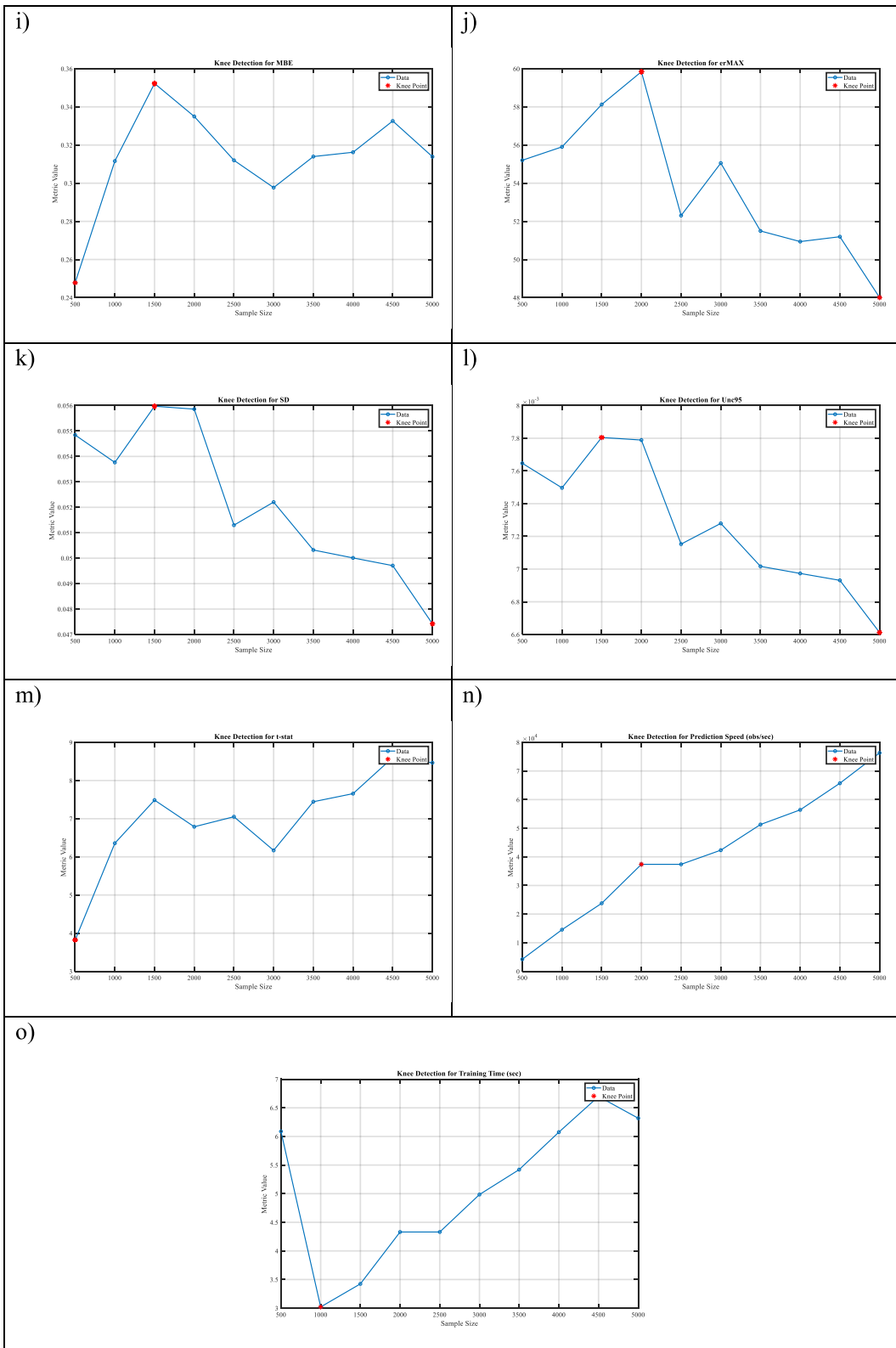


Fig. 10. (continued).

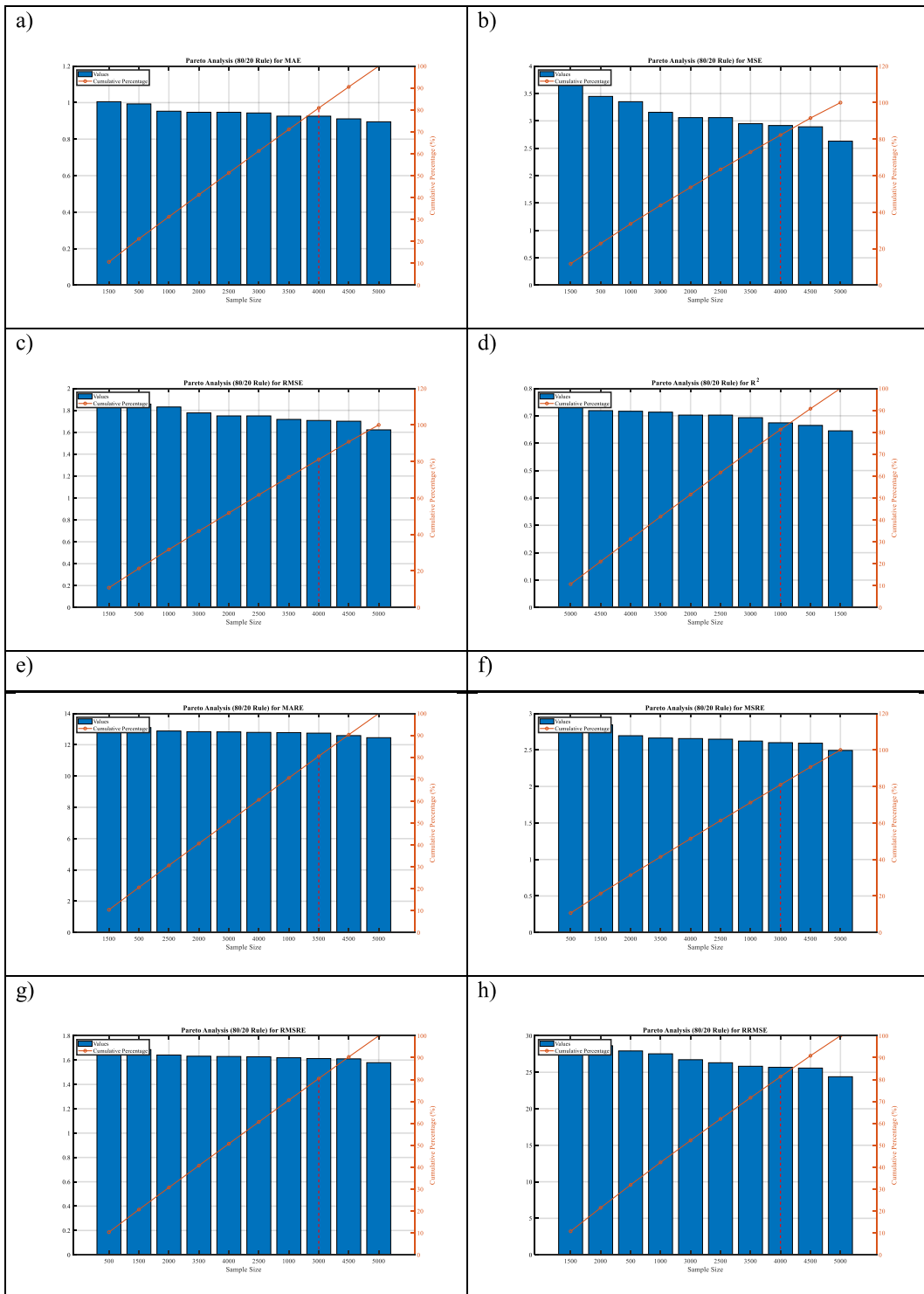


Fig. 11. Results for Pareto Analysis Method (80/20 Rule) for a) MAE, b) MSE, c) RMSE, d) R², e) MARE, f) MSRE, g) RMSRE, h) RRMSE, i) MBE, j) erMAX, k) SD, l) Un95, m) t-stat, n) Prediction Speed, o) Training time.

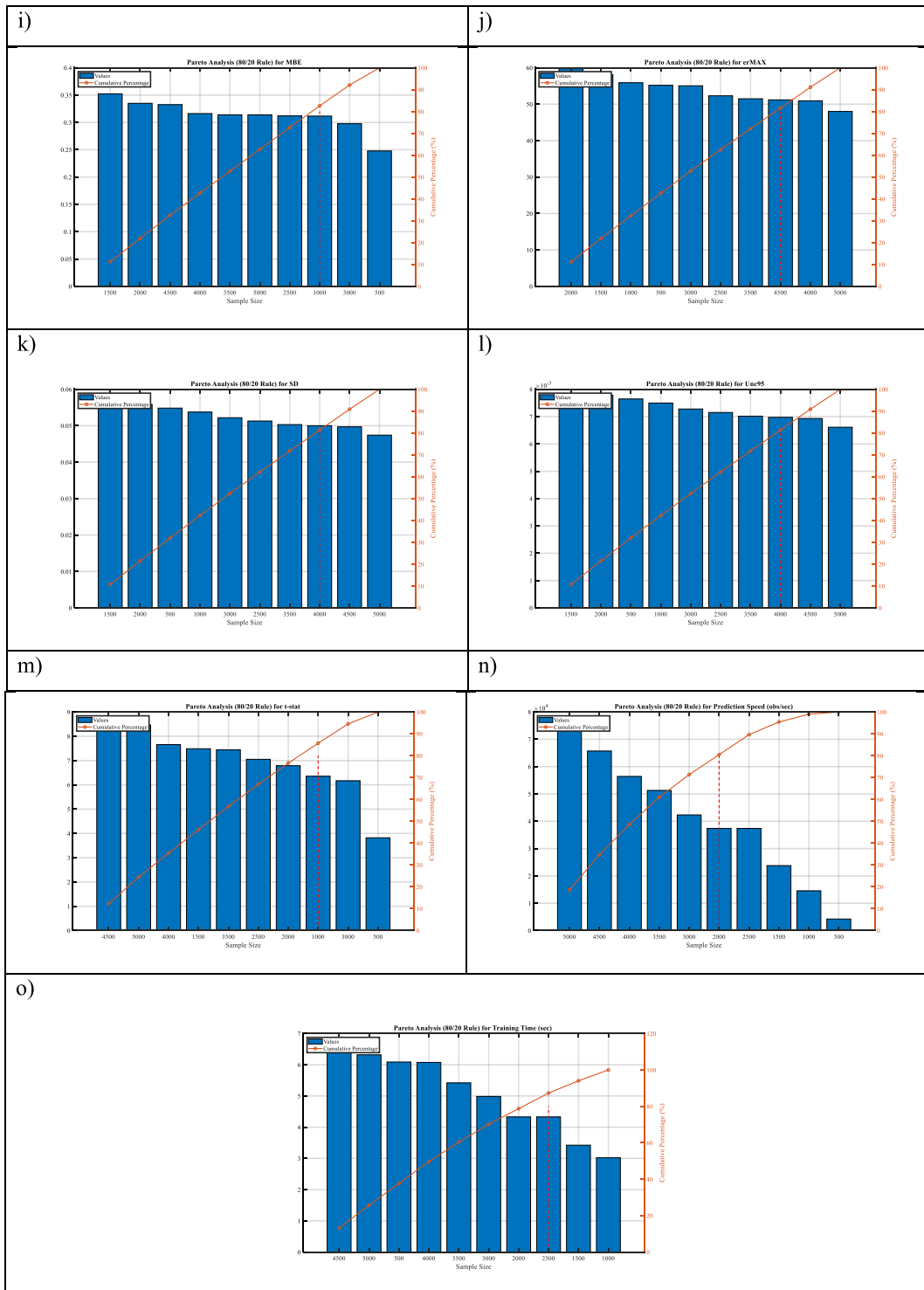


Fig. 11. (continued).

4.2.3. Performance of different machine learning methods

Accurate predictions and fast inference times are essential for structural analysis and real-time applications. Including training time in the evaluation provides valuable insight into the computational demands of different methods. This balanced evaluation provides a comprehensive understanding of each method's suitability for structural analysis tasks. Table 2 compares the performance of 26

Table 2
The result of different ML-based methods.

Methods	RMSE (Validation)	MSE (Validation)	RSquared (Validation)	MAE (Validation)	MAE (Test)	MSE (Test)	RMSE (Test)	RSquared (Test)	Prediction Speed (obs/sec)	Training Time (sec)
Linear Regression										
<i>Linear</i>	1.438E-01	2.068E-02	4.723E-01	9.216E-02	8.582E-02	1.551E-02	1.245E-01	5.121E-01	9.948E+03	1.111E+01
<i>Interactions</i>	1.380E-01	1.906E-02	5.136E-01	9.334E-02	8.232E-02	1.363E-02	1.167E-01	5.712E-01	1.351E+03	1.981E+01
<i>Robust</i>	1.632E-01	2.665E-02	3.199E-01	7.954E-02	7.181E-02	1.981E-02	1.408E-01	3.766E-01	5.495E+03	1.780E+01
<i>Stepwise</i>	1.253E-01	1.571E-02	5.990E-01	8.083E-02	7.381E-02	1.126E-02	1.061E-01	6.455E-01	2.542E+04	1.844E+03
Regression Tree										
<i>Fine</i>	9.745E-02	9.496E-03	7.576E-01	4.350E-02	3.722E-02	5.448E-03	7.381E-02	8.286E-01	9.824E+03	1.679E+01
<i>Medium</i>	9.360E-02	8.761E-03	7.764E-01	4.308E-02	3.496E-02	4.531E-03	6.731E-02	8.574E-01	1.023E+04	1.259E+01
<i>Coarse</i>	9.957E-02	9.915E-03	7.469E-01	4.699E-02	4.189E-02	6.609E-03	8.130E-02	7.920E-01	3.823E+03	1.179E+01
SVM										
<i>Linear</i>	1.592E-01	2.533E-02	3.534E-01	7.942E-02	7.133E-02	1.847E-02	1.359E-01	4.188E-01	5.940E+03	1.882E+01
<i>Quadratic</i>	1.358E-01	1.844E-02	5.294E-01	6.118E-02	5.152E-02	1.197E-02	1.094E-01	6.233E-01	6.947E+03	9.860E+00
<i>Cubic</i>	1.236E-01	1.527E-02	6.102E-01	5.699E-02	4.755E-02	9.325E-03	9.656E-02	7.066E-01	6.291E+03	1.799E+01
<i>Fine Gaussian</i>	2.024E-01	4.095E-02	-4.524E-02	1.169E-01	1.096E-01	3.294E-02	1.815E-01	-3.670E-02	5.982E+03	1.770E+01
<i>Medium Gaussian</i>	1.370E-01	1.877E-02	5.208E-01	6.325E-02	5.437E-02	1.237E-02	1.112E-01	6.106E-01	1.037E+04	1.746E+01
<i>Coarse Gaussian</i>	1.620E-01	2.625E-02	3.301E-01	7.781E-02	6.954E-02	1.903E-02	1.379E-01	4.012E-01	1.573E+04	1.679E+01
Ensemble methods										
<i>Boosted</i>	6.840E-02	4.678E-03	8.806E-01	3.136E-02	2.666E-02	2.476E-03	4.976E-02	9.221E-01	1.438E+04	1.606E+01
<i>Bagged</i>	9.259E-02	8.572E-03	7.812E-01	4.135E-02	3.680E-02	5.140E-03	7.169E-02	8.383E-01	9.687E+03	1.538E+01
GPR										
<i>Squared Exponential</i>	1.132E-01	1.281E-02	6.730E-01	6.666E-02	5.884E-02	8.211E-03	9.061E-02	7.416E-01	2.700E+04	1.138E+03
<i>Matern 5/2</i>	1.121E-01	1.257E-02	6.791E-01	6.523E-02	5.762E-02	8.104E-03	9.002E-02	7.450E-01	1.241E+04	2.166E+02
<i>Exponential</i>	1.182E-01	1.398E-02	6.432E-01	6.670E-02	6.006E-02	9.278E-03	9.632E-02	7.080E-01	1.421E+04	2.918E+02
<i>Rational Quadratic</i>	1.124E-01	1.264E-02	6.773E-01	6.578E-02	5.767E-02	8.110E-03	9.005E-02	7.448E-01	1.544E+04	4.549E+02
Neural Network										
<i>Narrow</i>	1.043E-01	1.088E-02	7.223E-01	4.175E-02	3.131E-02	3.372E-03	5.807E-02	8.939E-01	2.808E+04	1.675E+01
<i>Medium Neural</i>	8.850E-02	7.833E-03	8.001E-01	3.700E-02	3.207E-02	7.306E-03	8.547E-02	7.701E-01	1.138E+04	5.051E+01

(continued on next page)

Table 2 (continued)

Methods	RMSE (Validation)	MSE (Validation)	RSquared (Validation)	MAE (Validation)	MAE (Test)	MSE (Test)	RMSE (Test)	RSquared (Test)	Prediction Speed (obs/sec)	Training Time (sec)
<i>Wide</i>	1.115E-01	1.242E-02	6.829E-01	7.706E-02	5.632E-02	6.370E-03	7.981E-02	7.996E-01	2.823E+04	7.927E+01
<i>Bilayered</i>	1.044E-01	1.089E-02	7.220E-01	4.482E-02	3.360E-02	5.338E-03	7.306E-02	8.320E-01	2.797E+04	1.983E+01
<i>Trilayered</i>	8.982E-02	8.068E-03	7.941E-01	3.954E-02	3.717E-02	4.410E-03	6.641E-02	8.612E-01	1.712E+04	5.376E+01
Kernel Approximation Regrattion										
<i>SVM</i>	1.325E-01	1.757E-02	5.516E-01	7.065E-02	6.291E-02	1.180E-02	1.086E-01	6.288E-01	1.645E+04	4.162E+01
<i>Least Squares Regression</i>	1.311E-01	1.718E-02	5.616E-01	7.844E-02	6.975E-02	1.175E-02	1.084E-01	6.302E-01	1.070E+04	2.593E+01
CSS-ANN	8.332E-3	5.236E-04	9.883E-01	8.625E-03	7.699E-3	4.412E-04	9.656E-3	9.867E-01	2.505E+04	1.865E+01

machine learning methods using the RMSE, MSE, R-squared, and MAE as key metrics for evaluating predictive accuracy. Alongside these, training time and prediction speed are reported to assess computational efficiency. The results incorporate both validation and test datasets for the second example, ensuring consistent datasets are used across all methods to ensure fairness. While the validation results reflect the effectiveness of the training process, the test results are pivotal in gauging the predictive performance of the models. By analyzing both predictive accuracy and computational efficiency, we provide a holistic evaluation of the methods' applicability for structural analysis tasks, emphasizing their potential for accurate, efficient, and real-time decision-making.

The results demonstrate varying levels of success across the evaluated methods. Linear regression approaches, including linear, interactive, robust, and stepwise models, exhibit limited predictive capability for the task at hand. Among them, Stepwise Linear Regression emerges as the best performer, achieves an R-squared of 0.6455 and an RMSE of 0.106 on the test dataset. However, despite its relative advantage within this group, the accuracy achieved by these methods falls short of meeting the stringent requirements of structural analysis, where high precision is paramount.

Regression tree methods, particularly median regression trees, demonstrate improved performance, achieving an R-squared of 0.8574 on the test set and 0.7764 on the validation dataset. However, an approximate error rate of 15 % restricts their applicability for structural analysis tasks without further optimization or model tuning.

Support Vector Machine (SVM) methods, encompassing six variants, fail to deliver satisfactory results. Even the top-performing Cubic SVM achieves an R-squared of only 0.7066 for testing and 0.6102 for validation. These findings indicate that SVM methods are not well-suited for this problem type. The underperformance of SVM methods can be attributed to their limited ability to model the complex, non-linear, and high-dimensional nature of the problem. Additionally, SVMs often face challenges with large datasets and high computational demands, particularly when fine-tuning hyperparameters. The rigidity of kernel functions further restricts their adaptability, making them less suitable for capturing intricate, non-linear patterns essential for accurate structural analysis.

Among the evaluated methods, ensemble approaches, particularly the Boosted ensemble method, demonstrate exceptional performance. The Boosted ensemble model achieves the highest R-squared value of 0.9221 on the test dataset and excels across all evaluated metrics. Its superior performance is attributed to the ability to aggregate multiple trees, enhancing both accuracy and robustness.

Neural networks, widely used in structural design and analysis [49] also show promising results. Various architectures were tested, with most achieving R-squared values near or above 0.80 on the test dataset. Notably, the Narrow Neural Network reached R-squared values approaching 0.90. Despite these strong results, the developed CGO-ANN model surpasses even the best-performing neural networks in this category, highlighting its effectiveness for the task.

Gaussian Process Regression (GPR) and Kernel Approximation Regression methods deliver moderate performance but fail to meet the precision required for structural analysis applications. Their practical utility is further constrained by long training times and relatively higher error metrics, making them less suitable for large-scale structural problems.

4.2.4. Performance evaluation for CGO-ANN

Table 2 provides the performance evaluation for the developed CGO-ANN as well, which emerges as the best-performing model. It achieves the highest R-squared of 0.9867 on the test dataset, with an RMSE of $9.656E-3$ and a mean absolute error of $7.699E-3$. Additionally, its computational efficiency is notable, with training and prediction times comparable to other methods.

Although other ANN architectures, such as Narrow, Medium, and Trilayered Neural Networks, demonstrate solid performance—evidenced by the Trilayered ANN achieving an R-squared value of 0.8612 during testing—these models do not match the CGO-ANN in terms of precision and consistency, especially when it comes to minimizing error metrics. This underscores the advantage of integrating Chaos Game Optimization, which enhances the model's ability to fine-tune weights and biases more effectively, resulting in superior performance and more accurate predictions. While ensemble methods-based models, such as the Boosted ensemble method, demonstrate excellent R-squared values (up to 0.9221 in testing) and relatively low RMSE, their prediction accuracy is still inferior to CSS-ANN. Gaussian Process Regression and Support Vector Machines also exhibit higher error metrics and longer training times making them less efficient for large-scale structural analyses.

The performance comparison also reveals notable differences in training time and prediction speed, emphasizing the computational efficiency of each method. The CSS-ANN model demonstrates a balanced approach, achieving remarkable accuracy with a relatively moderate training time of 18.65 s. This is particularly significant given its superior predictive performance, as the optimization process enabled by CGO allows the model to converge efficiently without excessive computational cost. In contrast, traditional ANN models show varying training times. For instance, the Medium Neural Network, despite delivering reasonable accuracy, requires 50.51 s for training, while the Trilayered ANN takes 53.76 s. The ensemble models, such as the Boosted ensemble method, offer competitive prediction speeds (14,380 observations/sec) and moderate training times (16.06 s). However, they fall short of achieving the precision of CGO-ANN. The GPR methods and kernel-based approaches suffer from significantly longer training times. For example, the Squared Exponential GPR requires 1138 s for training, making it impractical for large-scale applications. Similarly, kernel approximation regression methods like SVM take 41.62 s to train, with lower predictive accuracy compared to CGO-ANN. Overall, CGO-ANN's superior balance between accuracy, robustness, and computational efficiency underscores its potential as a state-of-the-art method for structural analysis and optimization.

It should be noted that the flexibility and adaptability of our framework, not only integrates CGO with ANNs but also offers the potential for future improvements and the application of various machine learning techniques. We highlight that the purpose of integrating CGO with ANNs is to show that machine learning models can be optimized and enhanced within the proposed framework. This approach allows for continuous improvement and customization of machine learning methods based on the specific characteristics of the structural problem at hand. The developed framework is not limited to a fixed set of algorithms but is designed to be

expansive and adaptable, allowing researchers and practitioners to integrate other state-of-the-art machine learning models as well as optimization techniques to improve performance over time. This flexibility ensures that the framework can be extended in the future with additional optimization methods, data-driven models, or advanced machine learning strategies, thus allowing the proposed system to remain relevant and scalable as new techniques emerge in the field of structural analysis and AI. This reinforces the idea that the framework serves as a dynamic tool for applying and advancing machine learning methods, establishing it as a valuable contribution to the field of structural engineering.

4.2.5. Shapely additive explanations (SHAP)

Shapley values, rooted in cooperative game theory, offer a robust method for understanding the contribution of each feature to a model's prediction [47]. These values attribute a fair and consistent share of the total prediction to each feature by considering all possible combinations of features. The Shapley value for a particular feature quantifies how much that feature contributes to the model's prediction compared to the baseline, helping to explain the "why" behind each prediction in a transparent manner [47]. In our study, Fig. 12 provides a comprehensive visualization of Shapley values for each feature, offering a detailed analysis of how different input variables influence the model's predictions. By showing the Shapley values for each feature, Fig. 12 allows us to understand the relative importance of each variable in driving the model's decisions across different predictions. This is essential for gaining insights into the internal workings of the machine learning model, particularly when dealing with complex systems like structural optimization. For example, features 37, 43, 49, and 55 exhibit a high range of Shapley values, indicating that they have a significant influence on the model's output. The larger the range of the Shapley value, the more critical the feature's role in determining the prediction. These insights provide us with a clearer understanding of which features are contributing most to the model's decision-making process, helping to identify key factors that could be pivotal in refining the design or improving the model's performance.

Additionally, analyzing the Shapley values goes beyond simply identifying important features; it also sheds light on the direction of influence each feature has on the model's predictions. For instance, some features, such as feature numbers 7, 15, 58, and 33, exhibit a positive contribution, with Shapley values of 146, 134, 132, and 131 positives out of a total of 200. This indicates that increasing the values of these features is likely to improve the model's predictions. On the other hand, certain features like numbers 50, 31, 44, and 43 show negative Shapley values, with 169, 160, 158, and 152 negative values, respectively. These features negatively impact the model's performance, suggesting that increasing their values could decrease the accuracy of the predictions. This directional insight into the relationship between features and the model's output is essential for understanding how each feature influences the prediction, providing valuable information for model optimization and design adjustments.

Fig. 13 further enriches our analysis by displaying the mean absolute Shapley values for each feature, providing a clear aggregate measure of each feature's overall importance. The mean absolute value reflects the magnitude of a feature's contribution to the model's predictions, regardless of whether that contribution is positive or negative. This simplification allows for a more intuitive understanding of which features are the most influential across the entire dataset. By focusing on the magnitude of these contributions, we can pinpoint the key features that have the greatest impact on the model's behavior. In our analysis, the 16 most important features are highlighted in Fig. 13(a), with the top four—features 43, 37, 49, and 55—standing out due to their significantly higher mean absolute Shapley values. These features emerge as the most crucial, underscoring their pivotal role in the model's decision-making

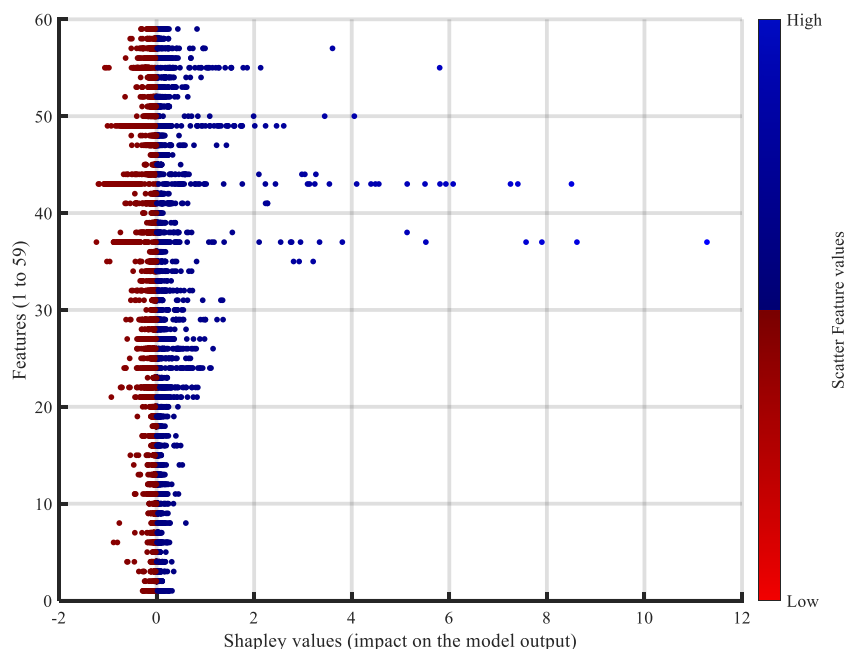


Fig. 12. The importance effect of features of test data using Shapley values for all features.

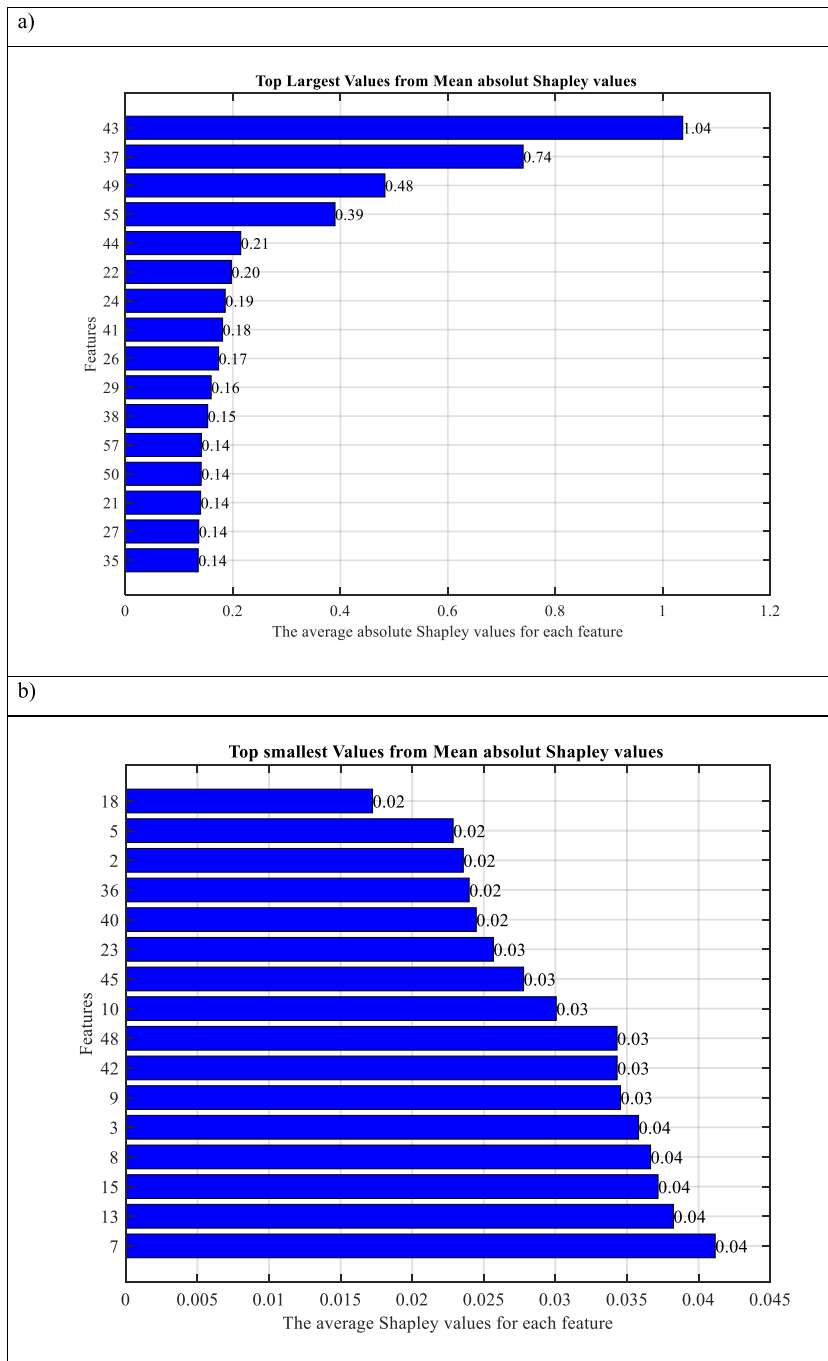


Fig. 13. The average absolute Shapley values for top feature.

process. This approach allows us to prioritize these critical features, ensuring they are the focal point for optimization and further refinement. The use of Shapley values in this context not only provides a deeper understanding of the model but also enhances its interpretability, ultimately building trust in its predictions. This is particularly valuable in structural design, where precise and transparent decision-making is essential.

Another key advantage of Shapley values is their ability to identify features that contribute minimally or redundantly to the model's performance. Features with very small Shapley values suggest that their impact on the model's predictions is negligible and can be considered for removal in future iterations. This process of feature selection helps streamline the model, reducing unnecessary complexity while maintaining predictive accuracy. Fig. 13(b) highlights several features, such as features 7, 13, 15, 8, and 3, which exhibit very small Shapley values and thus have minimal contribution to the model. These features can be safely neglected in future

model refinements. In addition to these, the figure also identifies 11 other features with similarly low Shapley values. Removing these less influential features not only simplifies the model but also reduces the risk of overfitting, enhancing its generalizability. Moreover, this approach boosts computational efficiency, which is particularly important when working with large datasets or in real-time optimization scenarios.

The results of the CGO-ANN method developed with varying numbers of neurons are also presented in Table 3. This demonstrates that we only need 20 neurons to achieve optimal performance, indicating that high complexity is not necessary for achieving excellent results with this method. The results demonstrate that the CGO-ANN method is a promising approach for predicting the response of concrete building structures.

4.2.6. Performance of different sampling methods

Based on Fig. 14, we evaluated the impact of different sampling methods on the performance of the CGO-ANN model. In the proposed framework, sampling plays a crucial role in generating an initial dataset that ensures high-quality machine learning results. The Figure demonstrates that the choice of sampling method significantly influences the success rate of the CGO-ANN model. In other words, a well-balanced dataset enhances the model's ability to generalize effectively, leading to more accurate predictions.

To investigate this, we examined three widely used sampling methods: random sampling, cluster sampling, and stratified sampling. Random sampling involves selecting individuals randomly from the population, ensuring that every instance has an equal probability of being picked. Cluster sampling, in contrast, divides the population into smaller groups or clusters, selecting a subset of these clusters to represent the entire dataset. Stratified sampling further refines this approach by dividing the population into distinct subgroups (or strata) based on shared characteristics, then randomly selecting samples from each subgroup to maintain proportional representation. These sampling methods were applied for all examples. For a fair comparison, we used the same set of data for each test.

According to the statistical evaluation, the cluster sampling technique emerged as the most effective approach for this specific model, producing the lowest MSE, RMSE and highest R-square values. This suggests that cluster sampling successfully captures the underlying patterns in the data while maintaining computational efficiency. Stratified sampling also demonstrated strong performance but proved to be computationally demanding, reducing its overall practicality for large-scale structural optimization tasks. Random sampling, while straightforward and easy to implement, did not perform as well as the other two methods due to its potential to overlook critical variations in the dataset.

4.2.7. Dimensionality reduction

Towards the end of this section, we present the results of the third example, focusing on the impact of different dimensionality reduction strategies. Initially, we explore the reduction of search space which is a critical factor in improving the efficiency of optimization algorithms. In the first scenario, we consider all 267 available sections for all elements in the structural design. The results are depicted in Fig. 15(a). This represents a comprehensive approach, where the algorithm evaluates the full range of possible section configurations. However, as the complexity of the search space increases, the algorithm's performance may be impacted due to the vast number of potential solutions.

For, the R-squared value increases from 0.88 % to 0.98 % for the test sets, indicating a marked enhancement in predictive accuracy. In the subsequent scenario, we decrease the number of sections considered for columns and beams based on their specific

Table 3
The result of CGO-ANN with different numbers of neurons.

Step	Number of Neurons	MSE	R ²
Train	5	3.000E-04	9.920E-01
Validation		1.400E-03	9.584E-01
Test		1.700E-03	9.594E-01
Train	10	6.000E-04	9.916E-01
Validation		2.300E-03	9.366E-01
Test		9.000E-04	9.636E-01
Train	20	2.773E-04	9.918E-01
Validation		4.032E-04	9.883E-01
Test		4.412E-04	9.867E-01
Train	30	4.000E-04	9.903E-01
Validation		1.800E-03	9.475E-01
Test		2.400E-03	9.427E-01
Train	40	4.000E-04	9.890E-01
Validation		2.400E-03	8.907E-01
Test		3.500E-03	9.282E-01
Train	50	2.000E-04	9.962E-01
Validation		3.300E-03	9.246E-01
Test		3.100E-03	8.933E-01

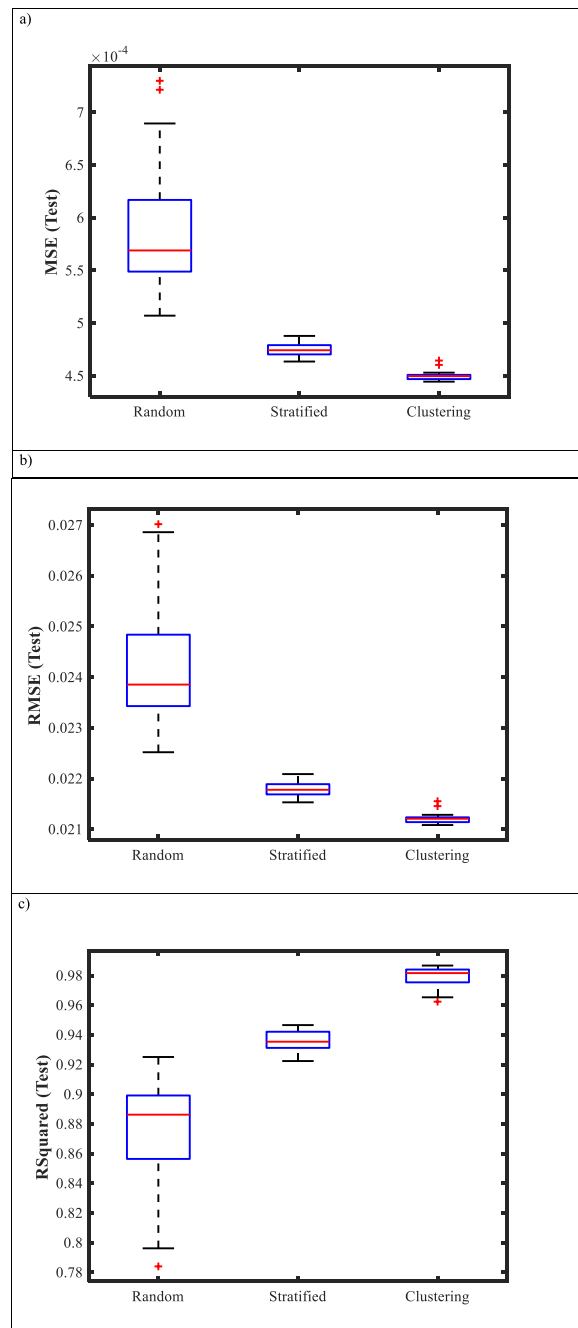


Fig. 14. Statistical study for CGO-ANN for different sampling methods.

properties. This step is designed to narrow down the search space, focusing on sections that are most relevant to the design constraints and performance requirements. The resulting performance is presented in Fig. 15(b). It is clear that limiting the search space leads to significant improvements in the algorithm's performance. For example, the R-squared value increases from 0.88 % to 0.98 % for test sets, indicating a marked enhancement in predictive accuracy.

This improvement illustrates that reducing the search space not only makes the algorithm more efficient but also more accurate. The ability to focus on a smaller, more targeted subset of sections allows the algorithm to identify optimal solutions more quickly, leading to faster convergence and better performance. Furthermore, the results highlight the algorithm's robustness in adapting to changes in section properties, reinforcing its flexibility and reliability in handling variations in structural design. This adaptability is essential for ensuring that the optimization process remains effective across a range of scenarios, demonstrating the algorithm's potential for real-world applications in structural engineering.

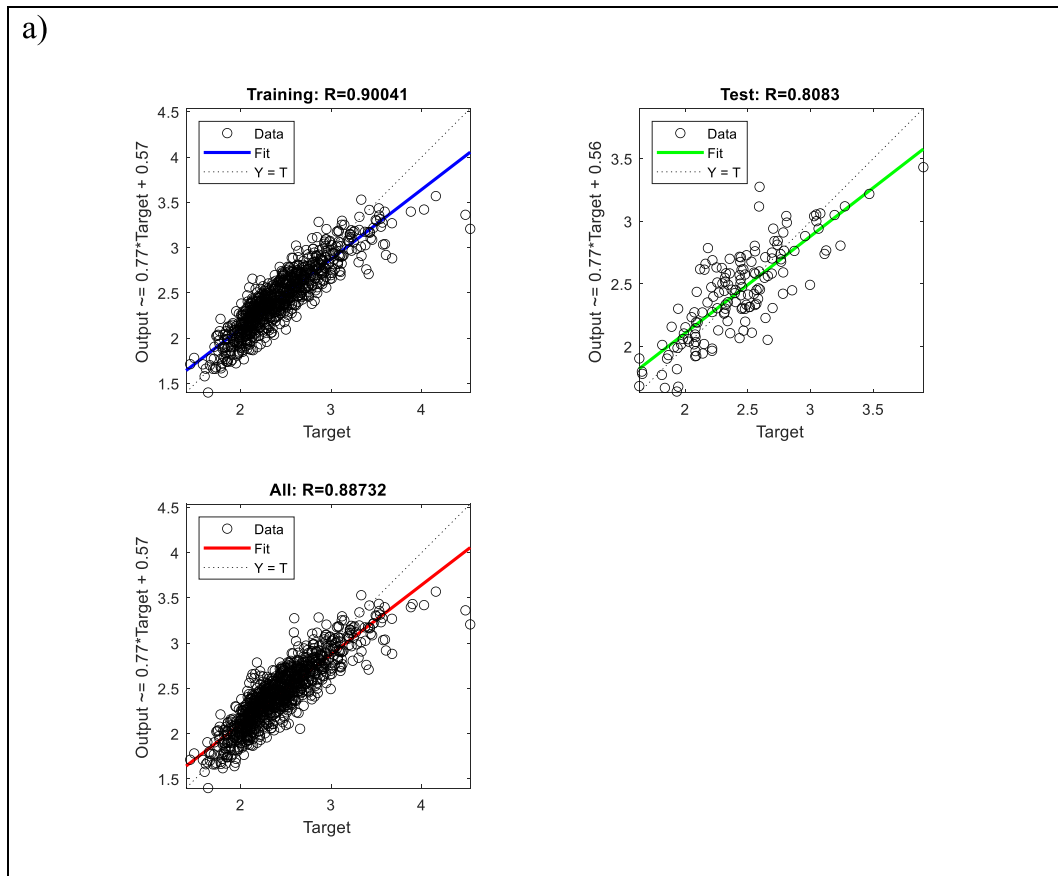


Fig. 15. Results for the models with dimensional reducing: a) using All sections; b) using selected sections.

Finally, Fig. 16 presents the results of a model with an increased number of outputs, raised to 22 (representing the number of stories in the building). Interestingly, the accuracy of the model improved compared to the results obtained with a single output (Fig. 15(a)). This finding runs counter to our initial expectations, as increasing the number of outputs generally introduces additional complexity, which can often lead to a decrease in model performance.

One potential explanation for this unexpected improvement could lie in the expansion of the datasets used for model training. With more outputs, the model has access to a greater volume of data, which could contribute to a more robust learning process. Additionally, the comparison indices used to assess model performance are based on an average view of the entire dataset, and this broader scope might have led to better overall results. However, while this improvement is evident in the numerical analysis, it's important to note that from an engineering perspective, average performance is not the most critical metric compared to maximum error.

In structural design, engineers place a greater emphasis on the maximum error, as it represents the worst-case scenario that could potentially have the most significant impact on the structure's performance. The maximum error serves as a primary constraint in the design process, ensuring that the design remains within acceptable safety margins and performance standards.

Therefore, while the increase in outputs may yield numerical improvements, the model's real-world applicability must be carefully evaluated with engineering-specific metrics, particularly the maximum error. This ensures that the model can be relied upon to deliver accurate and safe results in practical applications. To sum up, when developing models with a large number of outputs, a thorough examination of both numerical performance and key engineering measures is essential for ensuring reliability and robustness in real-world scenarios.

5. Conclusion remarks

In this study, we present a comprehensive framework for analyzing large-scale structures and predicting their behavior. The proposed framework integrates various techniques, including sample selection, dimensionality reduction, and machine learning analysis techniques to improve the efficiency and accuracy of structural predictions. By utilizing sample selection techniques, we can reduce the input size of the problem, while the machine learning model and other techniques enable accurate predictions of future

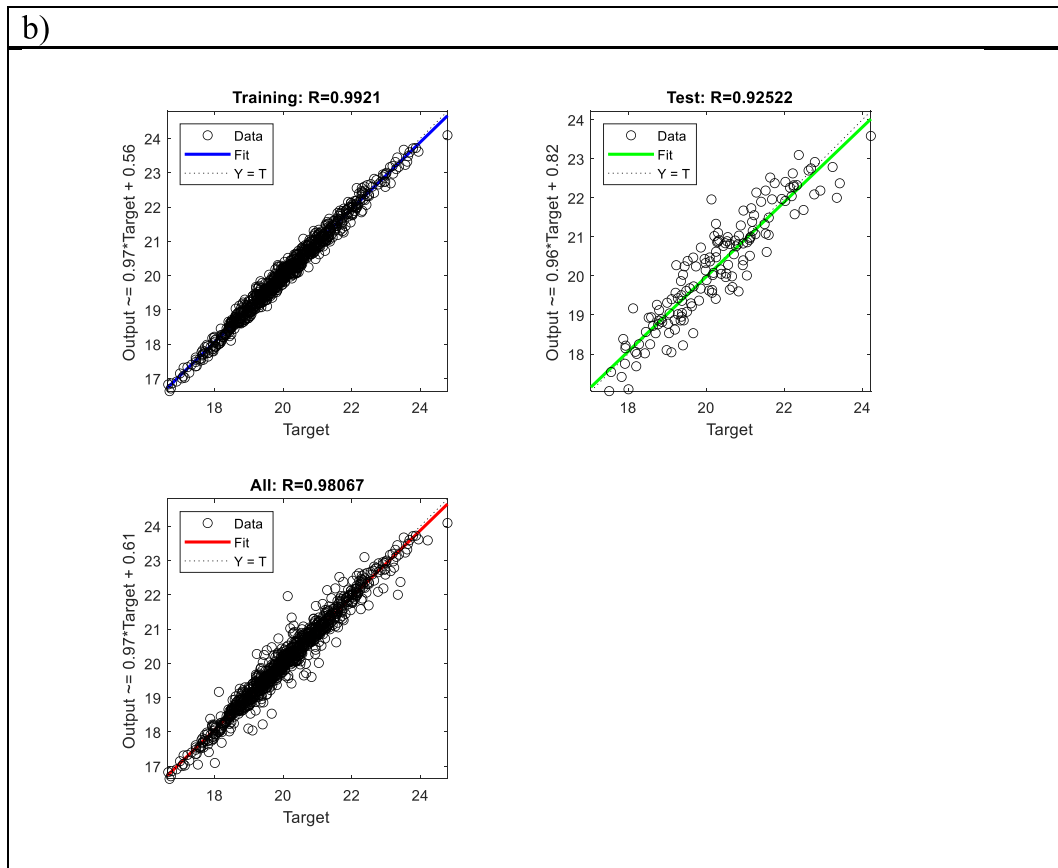


Fig. 15. (continued).

structural designs without relying on computationally expensive finite element simulations.

The primary strength of the framework lies in its adaptive nature, which allows it to seamlessly switch between different methods based on the specific requirements of the problem. This flexibility ensures that researchers and practitioners can optimize their model choices, leading to more accurate predictions and better-informed structural decisions. The adaptability of the framework ensures that it can effectively handle varying problem complexities and data characteristics.

The framework has been trained using 26 different machine-learning methods and validated using a comprehensive set of performance metrics. Additionally, an improved machine learning method is introduced that further enhances prediction accuracy. The parameter optimization of the machine learning model is automated through the use of chaos game optimization, further refining the model's performance. The results demonstrate the high accuracy of the proposed framework in predicting the behavior of large-scale structures.

In engineering applications, particularly in structural design and assessment, the interpretability of machine learning predictions is essential for ensuring that engineers can trust and effectively use the model's outputs. To address this challenge, we incorporated Shapley Additive Explanations (SHAP) into the framework, enabling us to quantify the contribution of each input feature to the model's predictions. This analysis helps engineers identify the most significant features influencing structural behavior, providing critical insights into the design parameters necessary to ensure safety and performance.

While the framework significantly enhances the design and analysis of such structures, improving their safety and durability, it does not explore all aspects of the framework in depth. Future studies should investigate and improve each step of the proposed techniques. It can focus on extending the framework to accommodate a wider range of structural types and more complex loading scenarios, such as dynamic and non-linear loads. Advanced machine learning techniques, including deep learning and ensemble methods, could be incorporated to further improve the predictive accuracy and robustness of the framework. Additionally, testing the framework with real-world data, particularly from structural health monitoring systems, would validate its applicability in practical design scenarios and enhance its generalization to actual conditions. Such testing could pave the way for real-time design and assessment of large-scale structures. Furthermore, integrating real-time data from sensors and Internet of Things (IoT) devices would allow for continuous monitoring and prediction, making the framework highly suitable for smart infrastructure systems and adaptive decision-making.

Exploring the influence of environmental factors, material properties, and aging effects on structural behavior could further

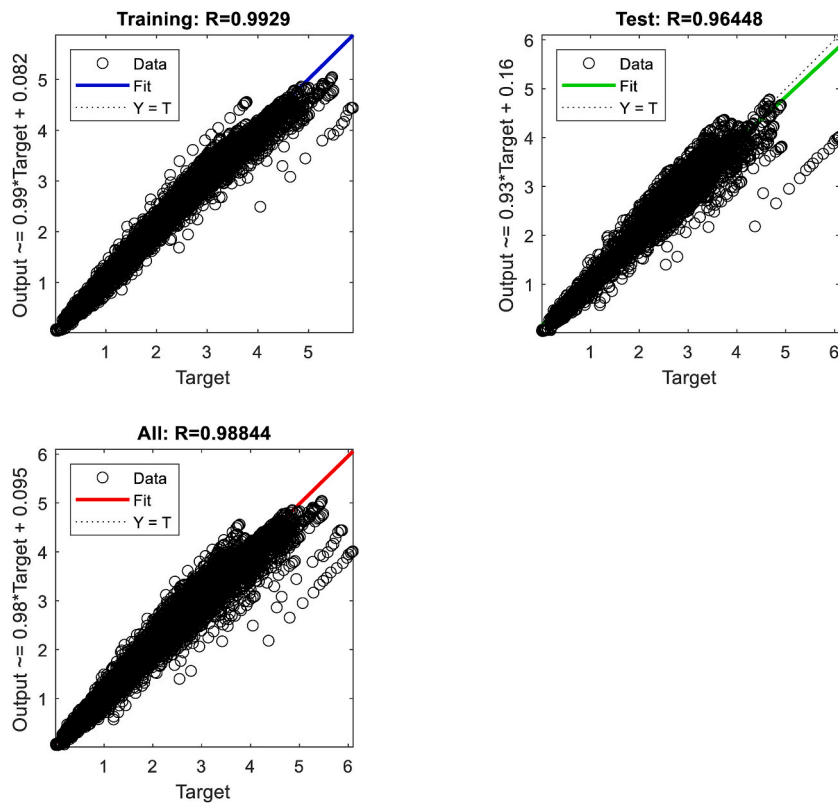


Fig. 16. Results for the model with the increased number of outputs.

enhance the framework's comprehensiveness and applicability. Incorporating topology optimization would also provide valuable insights into material distribution and structural performance, enabling more efficient and innovative designs. Additionally, collaborating with structural design software developers could facilitate the framework's practical application in industry, improving the design, safety, and longevity of large-scale structures. Finally, future work could apply other interpretability techniques, such as feature importance analysis using Permutation Importance, to offer further clarity on how predictions are made.

CRediT authorship contribution statement

Siamak Talatahari: Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Fang Chen:** Writing – review & editing, Supervision, Funding acquisition. **Amir H. Gandomi:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors acknowledge that this research is supported by the Australian Government Research Training Program Scholarship.

Data availability

Data will be made available on request.

References

- [1] A.R. Kashani, C.V. Camp, M. Rostamian, K. Azizi, A.H. Gandomi, Population-based optimization in structural engineering: a review, *Artif. Intell. Rev.* (2021) 1–108.

- [2] N. Sadrekarimi, S. Talatahari, B.F. Azar, A. Gandomi, A surrogate merit function developed for structural weight optimization problems, *Soft Comput.* 27 (2023) 1533–1563.
- [3] Y. Zhang, M. Xiao, Z. Ding, M. Xu, G. Jiang, L. Gao, Dynamic response-oriented multiscale topology optimization for geometrically asymmetric sandwich structures with graded cellular cores, *Comput. Methods Appl. Mech. Eng.* 416 (2023) 116367.
- [4] Z. Pan, L.-W. Zhang, K. Liew, Adaptive surrogate-based harmony search algorithm for design optimization of variable stiffness composite materials, *Comput. Methods Appl. Mech. Eng.* 379 (2021) 113754.
- [5] U. Kirsch, A unified reanalysis approach for structural analysis, design, and optimization, *Struct. Multidiscip. Optim.* 25 (2003) 67–85.
- [6] R.C. Coates, M. Coutie, F.K. Kong, *Structural Analysis*, CRC Press, 2018.
- [7] S. Talatahari, M. Azizi, Chaos Game Optimization: a novel metaheuristic algorithm, *Artif. Intell. Rev.* 54 (2021) 917–1004.
- [8] W. Zhang, Y. Wang, S.-K. Youn, X. Guo, Machine learning powered sketch aided design via topology optimization, *Comput. Methods Appl. Mech. Eng.* 419 (2024) 116651.
- [9] P.C. Lazaridis, I.E. Kavvadias, K. Demertzis, L. Iliadis, L.K. Vassiliadis, Structural damage prediction of a reinforced concrete frame under single and multiple seismic events using machine learning algorithms, *Appl. Sci.* 12 (2022) 3845.
- [10] G. Santarsiero, M. Mishra, M.K. Singh, A. Masi, Structural health monitoring of exterior beam–column subassemblies through detailed numerical modelling and using various machine learning techniques, *Machine Learning with Applications* 6 (2021) 100190.
- [11] G.K. Bekas, G.E. Stavroulakis, Machine learning and optimality in multi storey reinforced concrete frames, *Infrastructures* 2 (2017) 6.
- [12] C. Cakiroglu, K. Islam, G. Bekdas, U. Isikdag, S. Mangalathu, Explainable machine learning models for predicting the axial compression capacity of concrete filled steel tubular columns, *Constr. Build. Mater.* 356 (2022) 129227.
- [13] M. Flah, I. Nunez, W. Ben Chaabene, M.L. Nehdi, Machine learning algorithms in civil structural health monitoring: a systematic review, *Arch. Comput. Methods Eng.* 28 (2021) 2621–2643.
- [14] H.-T. Thai, Machine learning for structural engineering: a state-of-the-art review, *Structures*, Elsevier, 2022, pp. 448–491.
- [15] D.-C. Feng, S.-Z. Chen, E. Tacioglu, Deep learning-enhanced efficient seismic analysis of structures with multi-fidelity modeling strategies, *Comput. Methods Appl. Mech. Eng.* 421 (2024) 116775.
- [16] H. Sun, H.V. Burton, H. Huang, Machine learning applications for building structural design and performance assessment: state-of-the-art review, *J. Build. Eng.* 33 (2021) 101816.
- [17] C. Xu, B.T. Cao, Y. Yuan, G. Meschke, Transfer learning based physics-informed neural networks for solving inverse problems in engineering structures under different loading scenarios, *Comput. Methods Appl. Mech. Eng.* 405 (2023) 115852.
- [18] K. Mostafa, I. Zisis, M.A. Moustafa, Machine learning techniques in structural wind engineering: a state-of-the-art review, *Appl. Sci.* 12 (2022) 5232.
- [19] F. Kazemi, R. Jankowski, Machine learning-based prediction of seismic limit-state capacity of steel moment-resisting frames considering soil-structure interaction, *Comput. Struct.* 274 (2023) 106886.
- [20] N. Asgarkhani, F. Kazemi, A. Jakubczyk-Galczyńska, B. Mohebi, R. Jankowski, Seismic response and performance prediction of steel buckling-restrained braced frames using machine-learning methods, *Eng. Appl. Artif. Intell.* 128 (2024) 107388.
- [21] F. Kazemi, N. Asgarkhani, R. Jankowski, Optimization-based stacked machine-learning method for seismic probability and risk assessment of reinforced concrete shear walls, *Expert Syst. Appl.* 255 (2024) 124897.
- [22] N. Asgarkhani, F. Kazemi, R. Jankowski, Machine learning-based prediction of residual drift and seismic risk assessment of steel moment-resisting frames considering soil-structure interaction, *Comput. Struct.* 289 (2023) 107181.
- [23] F.V. Senhora, H. Chi, Y. Zhang, L. Mirabella, T.L.E. Tang, G.H. Paulino, Machine learning for topology optimization: physics-based learning through an independent training strategy, *Comput. Methods Appl. Mech. Eng.* 398 (2022) 115116.
- [24] W.K. Liu, S. Li, H.S. Park, Eighty years of the finite element method: birth, evolution, and future, *Arch. Comput. Methods Eng.* (2022) 1–23.
- [25] K.-J. Bathe, *Finite Element Procedures*, Klaus-Jurgen Bathe, 2006.
- [26] A. Kaveh, H. Rahami, I. Shojaei, *Swift Analysis of Civil Engineering Structures Using Graph Theory Methods*, Springer, 2020.
- [27] A. Kaveh, *Optimal Analysis of Structures by Concepts of Symmetry and Regularity*, Springer, 2013.
- [28] U. Kirsch, *Reanalysis of Structures*, Springer, 2008.
- [29] D. Li, Q. Zhou, G. Chen, Y. Li, Structural dynamic reanalysis method for transonic aeroelastic analysis with global structural modifications, *J. Fluid Struct.* 74 (2017) 306–320.
- [30] M.A. Akgün, J.H. Garcelon, R.T. Haftka, Fast exact linear and non-linear structural reanalysis and the Sherman–Morrison–Woodbury formulas, *Int. J. Numer. Methods Eng.* 50 (2001) 1587–1606.
- [31] W. Zuo, Z. Yu, S. Zhao, W. Zhang, A hybrid Fox and Kirsch's reduced basis method for structural static reanalysis, *Struct. Multidiscip. Optim.* 46 (2012) 261–272.
- [32] G. Gao, H. Wang, E. Li, G. Li, An exact block-based reanalysis method for local modifications, *Comput. Struct.* 158 (2015) 369–380.
- [33] ACI COMMITTEE, *Building Code Requirements for Structural Concrete (ACI 318-08) and Commentary*, American Concrete Institute, 2008.
- [34] A.I.O.S. Construction, AISC, *Specification for Structural Steel Buildings*, AISC, Chicago, IL, USA, 2022.
- [35] R.F. Design, *Specification for Structural Steel Buildings*, American Institute of Steel Construction, Chicago (IL, USA), 2005.
- [36] M. Kripka, G.F. Medeiros, A.C. Lemonge, Use of optimization for automatic grouping of beam cross-section dimensions in reinforced concrete building structures, *Eng. Struct.* 99 (2015) 311–318.
- [37] J.E. Herencia, R.T. Haftka, Structural optimization with limited number of element properties, *Struct. Multidiscip. Optim.* 41 (2010) 817–820.
- [38] J.P. Carvalho, A.C. Lemonge, E.C. Carvalho, P.H. Hallak, H.S. Bernardino, Truss optimization with multiple frequency constraints and automatic member grouping, *Struct. Multidiscip. Optim.* 57 (2018) 547–577.
- [39] A.C. Lemonge, H.J. Barbosa, A.L. Coutinho, C.C. Borges, Multiple cardinality constraints and automatic member grouping in the optimal design of steel framed structures, *Eng. Struct.* 33 (2011) 433–444.
- [40] J.P. Carvalho, A.C. Lemonge, P.H. Hallak, D.E. Vargas, A differential evolution to find the best material groupings in truss optimization, *EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization*, Springer, 2019, pp. 102–114.
- [41] F. Jalayer, R. DE Risi, G. Manfredi, Bayesian Cloud Analysis: efficient structural fragility assessment using linear regression, *Bull. Earthq. Eng.* 13 (2015) 1183–1203.
- [42] H. Lou, B. Gao, F. Jin, Y. Wan, Y. Wang, Shear wall layout optimization strategy for high-rise buildings based on conceptual design and data-driven tabu search, *Comput. Struct.* 250 (2021) 106546.
- [43] B. Do, M. Ohsaki, M. Yamakawa, Sequential mixture of Gaussian processes and saddlepoint approximation for reliability-based design optimization of structures, *Struct. Multidiscip. Optim.* 64 (2021) 625–648.
- [44] H.S. Marie, K. Abu EL-Hassan, E.M. Almetwally, M.A. EL-Mandouh, Joint shear strength prediction of beam-column connections using machine learning via experimental results, *Case Stud. Constr. Mater.* 17 (2022) e01463.
- [45] A. Karbassi, B. Mohebi, S. Rezaee, P. Lestuzzi, Damage prediction for regular reinforced concrete buildings using the decision tree algorithm, *Comput. Struct.* 130 (2014) 46–56.
- [46] S. Talatahari, M. Azizi, Optimization of constrained mathematical and engineering design problems using chaos game optimization, *Comput. Ind. Eng.* 145 (2020) 106560.
- [47] M. Scott, L. Su-In, A unified approach to interpreting model predictions, *Adv. Neural Inf. Process. Syst.* 30 (2017) 4765–4774.
- [48] F. Kazemi, N. Asgarkhani, R. Jankowski, Machine learning-based seismic fragility and seismic vulnerability assessment of reinforced concrete structures, *Soil Dynam. Earthq. Eng.* 166 (2023) 107761.
- [49] M. Alanani, A. Elshaer, ANN-based optimization framework for the design of wind load resisting system of tall buildings, *Eng. Struct.* 285 (2023) 116032.