Smart Structures and Systems, *Vol. 36, No. 1 (2025) 1-22*
https://doi.org/10.12989/sss.2025.36.1.001

Prediction of design and optimization of steel structures using machine learning and artificial neural networks

Mohamad N. Askar*, Ahmad Ghareeb [a] and Mohammed H. Serror [b]

Cairo University, Faculty of Engineering, Structural Engineering Department, Giza, 12613, Egypt

(*Received January 11, 2025, Revised May 27, 2025, Accepted July 18, 2025*)

Abstract. In the field of structural engineering, traditional design and optimization methodologies are being transformed by the integration of machine learning (ML) and deep learning (DL) techniques. The increasing complexity of structural systems driven by variations in geometric parameters such as eave height, span length, frame spacing, and support conditions, as well as the growing demand for cost-effective solutions and reduced design computation time, has prompted engineers to adopt innovative approaches, including predictive algorithms for steel section selection. A comprehensive dataset was generated using finite element modeling (FEM) to represent a wide range of structural configurations. These configurations were validated through manual calculations to ensure data accuracy for ML and DL training. The trained models analyze structural parameters to predict optimal section dimensions, addressing nuanced design requirements. Various ML algorithms, including Polynomial Regression, LightGBM, XGBoost, Random Forest, and artificial neural networks (ANN) were employed for predicting column and beam dimensions. Among them, Random Forest achieved the highest accuracy (Adjusted R^2 = 94.132%, MAE = 0.336), followed by ANN (Adjusted R^2 = 91.63%, MAE = 0.212). A graphical user interface (GUI) was also developed to bridge the gap between model predictions and practical implementation, enabling engineers to design cost-effective and resilient structures in compliance with AISC, ASCE, and ECP standards.

Keywords: algorithms; ANN; data base; deep learning; frames; machine learning; optimization; steel

1. Introduction

The optimization of steel structures has long been a critical focus in structural engineering, driven by the need to enhance both material efficiency and structural resilience. This study explores the application of machine learning (ML) and deep learning (DL) techniques. Conventional design methods, which often rely on iterative calculations and code-compliant checks (Abarkan *et al*. 2024b), to streamline and optimize the design process for steel frame structures, providing an innovative approach that balances strength, stability, and material efficiency. Conventional design methods are time-consuming but offer greater flexibility, particularly when dealing with complex loading scenarios and stringent deflection and drift constraints. for Approximation of Steel Moment Frames Using Artificial Neural Networks (Noori and Varaee 2022), Prediction of Mechanical Properties of Steel Fiber-reinforced Concrete Using Convolutional Neural Network (CNN) (Kavya *et al*. 2022), and finally Optimization of Truss Structures (Barakat *et al*. 2022).

Utilizing Python and the Anastruct library for automated frame analysis (Basta *et al*. 2020), we developed a robust workflow that calculates and verifies frame dimensions across these cases. This dataset was subsequently applied to a series of ML algorithms GA and PSO (Zakhera *et al*. 2024) and DL algorithm DNN (Torky and Aburawwash 2018) and Artificial Neural Network (ANN). Predictive models are designed to recognize patterns within the data, providing fast and accurate predictions of section dimensions based on loading conditions and frame configurations.

Inspired by recent work on adaptive optimization (Senatore *et al*. 2025), this study aims to extend predictive methodologies to practical structural applications by focusing on feasible section sizing and layout efficiency under variable loading conditions.

It is explicitly stated that this study adopts a surrogate-driven prediction framework, which utilizes machine learning models to rapidly predict feasible structural sections based on a pre-generated dataset comprising optimized section designs. This methodology is fundamentally distinct from explicit mathematical optimization formulations that solve strict objective functions under rigorous constraints, as in the case of Saleh *et al*. (2024). Instead, it focuses on providing fast, data-driven predictions to support efficient engineering decisions.

Mathematical structural optimization focuses on rigorously minimizing objective functions such as weight or cost under strict design constraints. Techniques like gradient-based methods, genetic algorithms, and particle swarm optimization are widely used to optimize steel and

*Corresponding author,
Ph.D. Candidate, Structural Director,
E-mail: mohamad.202110479@eng-st.cu.edu.eg
[a] Assistant Professor
[b] Professor

ISSN: 1738-1584 (Print), 1738-1991 (Online)

concrete structures. While these methods ensure precision and theoretical robustness, they often demand high computational resources and are sensitive to problem setup. Despite these challenges, they remain fundamental to structural optimization research and inform the development of newer data-driven or hybrid strategies. "Topology Optimization of Adaptive Structures: New Limits of Material Economy" by Senatore and Wang (2024) represents one of the recent advances in mathematical optimization of structural systems.

This approach also provides a deeper understanding of the relationships between frame geometry, load conditions, and structural constraints. To ensure practical application, Graphical User Interface (GUI) was planned Cicconi *et al*. (2016) to be used with minimal computational effort. This approach aims to bridge the gap between traditional code-based design and modern predictive modelling and provides a scalable solution that integrates data-driven insights with engineering design requirements.

By leveraging advanced predictive techniques and adhering to rigorous design standards as in the work of Dhiman *et al*. (2019), this work offers a scalable and adaptive solution to meet the evolving demands of structural engineering safety and optimization.

The used algorithms in Machine and Deep Learning techniques as following:

- Polynomial Regression: A simple baseline model assuming a fixed mathematical relationship between features and outputs. While useful for reference, it struggles with complex, nonlinear interactions.
- LightGBM (Light Gradient Boosting Machine): A fast and efficient gradient boosting model suited for high-dimensional data, capable of capturing intricate feature relationships with strong predictive accuracy.
- XGBoost: A robust gradient boosting algorithm that handles outliers well and provides superior fine-tuning for complex datasets (Salehi & Burgueño, 2018).
- Random Forest: An ensemble of decision trees that effectively models nonlinear dependencies, offering strong generalization and resilience to overfitting.

Artificial Neural Network (ANN): A deep learning model with multiple hidden layers, capable of recognizing intricate structural patterns often missed by traditional Machine Learning (ML) models.

Machine learning models are embedded directly within design workflows to produce on-demand predictions across different structural zones (Ahmed and Fekry 2024). Deep Learning and Structural Optimization of Steel Frames.

This study leverages Python and the Anastruct library to automate frame analysis, generating a dataset of 39,000 cases. Current research highlights the growing application of ML/DL in structural engineering. For instance, Abarkan *et al*. (2024a) demonstrated ML's effectiveness in optimizing stainless steel columns, while Málaga-Chuquitaype (2022) reviewed its potential in structural design. Nguyen *et al*. (2020) applied ANN to predict axial compressive capacity, and Salehi and Burgueño (2018) explored emerging Artificial Intelligence (AI) methods in structural analysis.

In this paper, we will introduce in section 2 the Methodology, the needed cases data from finite element models, and the GUI presentation utilized to demonstrate the model outputs. Section 3 presents our methodology for machine learning philosophy and used algorithms, section 4 presents the Deep Learning model, section 5 illustrates overall challenges, section 6 shows results, discussions, and comparisons between different methods of real and predicted weights and sectional dimensions. Section 8 contains the recommendation and the future works.

2. Methodology

2.1 Parametric analysis of regional steel profiles for optimization

To establish a foundational understanding for this study, an extensive market analysis was conducted, centering on the design practices for steel structures, with a particular emphasis on the Middle Eastern context. This preliminary research yielded significant insights into the dominant design parameters, material specifications, and structural configurations prevalent in regional steel construction. Based on this market assessment, the scope of our analysis

Table 1 Common column dimensions for spans (20,35 m)

Section	Parameter	Span 20 m	Span 35 m
Top section	Hw (mm)	600	900
	tw (mm)	5	8
	bf (mm)	200	300
	tf (mm)	10	12
Bottom section	Hw (mm)	250	600
	tw (mm)	4	8
	bf (mm)	175	250
	tf (mm)	8	10

Table 2 Beams common dimensions for spans (20,35 m)

Section	Parameter	Span 20 m	Span 35 m
Section (1)	Hw (mm)	600	900
	tw (mm)	5	14
	bf (mm)	200	300
	tf (mm)	10	35
Section (4)	Hw (mm)	500	800
	tw (mm)	4	12
	bf (mm)	175	300
	tf (mm)	10	12
Section (3)	Hw (mm)	400	700
	tw (mm)	4	10
	bf (mm)	175	300
	tf (mm)	6	20

Table 3 Parameters used in the FEM generation

Parameter	Range / value	Description
Eave height (m)	5 – 10	Height of the eave
Bay span (m)	20 – 35	Distance between bays
Spacing (m)	5 – 10	Distance between frames
Dead loads (kN/m²)	0.25	Representing metal cladding - Super imposed dead load
Live loads (kN/m²)	0.6	Load due to occupancy
Collateral loads (kN/m²)	0.1 - 0.5	Additional loads such as mechanical equipment or ceiling systems
Wind loads	ASCE, ECP	Load due to wind pressure
Earthquake loads	ASCE, ECP	Load due to seismic activity

was expanded to encompass a comprehensive range of I-shaped steel elements, including beams and columns, with spans between 20 and 35 meters. This range is reflective of the typical demands observed in regional construction projects (Bonafide Research 2025).

Through this expanded dataset, we aim to advance the optimization of structural design using Machine Learning (ML) & Deep learning (DL) integrated with the Finite Element Method (FEM). This combined approach enhances both precision and efficiency in structural design, particularly under varying load conditions. Structural checks—including evaluations of local buckling, moment capacity, and shear capacity—were conducted for welded steel sections, ensuring that the results are applicable across diverse steel frame designs in Tables 1-2.

2.2 Finite Element Model (FEM) setup and design parameters for data base

To establish a robust foundation for this study, we developed twelve finite element models representing a typical single-frame warehouse structure. Each model consisted of two columns and two beams, connected by a central slope, to reflect a realistic industrial framing configuration. The models were constructed with varying eave heights (5 m to 10 m) and bay spans (20 m to 35 m) to capture a diverse range of geometric configurations. These variations allowed us to explore the behavior of different structural layouts under common load scenarios, with design parameters aligned to industry standards, including dead loads for metal cladding, live loads from occupancy, and collateral loads accounting for auxiliary equipment. Additionally, environmental load factors, such as wind and earthquake forces in Table 3 and Figs. 1-2-19 were incorporated following American Institute of Steel Construction (AISC) guidelines to ensure that the models adhered to practical safety and stability requirements.

2.2.1 Material definition and design functions

Material properties; Steel Fy 52: Weight Per unit Volume: 77 KN/m3, Modulus of Elasticity: 200,000 MPa, Poisson's Ratio: 0.3, Coefficient of thermal Expansion: 0.0000117, Shear Modulus: 76903.07 MPa. Parameters, boundaries conditions, and loads assignment as Tables 3-4, Slope is fixed with a value of 1:10, Spacing between columns are fixed with value of 6 m.

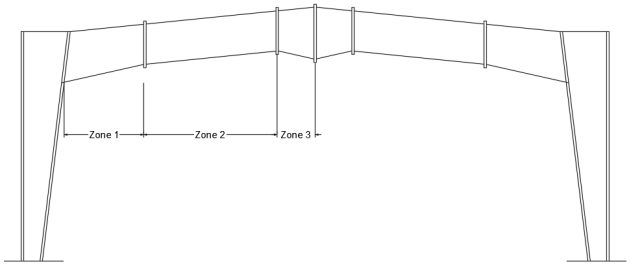


Fig. 1 General frame structure

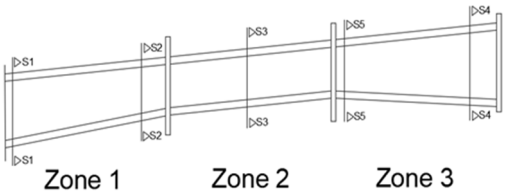


Fig. 2 Beam element zones sebmentations

Applying AISC Code American Institute of Steel Construction (2022), requirements for all stress types such as tension, shear, buckling limitations, serviceability limits, and finally the combined forces through the following Eqs. (1)-(2); the code checks are embedded within the design coding library, which is integrated into the design dataset and utilized throughout the verification process.

$$\text{For} \quad \frac{P_r}{P_c} \geq 0.2$$
$$\frac{P_r}{P_c} + \frac{8}{9}\left(\frac{M_{rx}}{M_{cx}} + \frac{M_{ry}}{M_{cy}}\right) \leq 1.0 \tag{1}$$

$$\text{For} \quad \frac{P_r}{P_c} < 0.2$$
$$\frac{P_r}{2P_c} + \left(\frac{M_{rx}}{M_{cx}} + \frac{M_{ry}}{M_{cy}}\right) \leq 1.0 \tag{2}$$

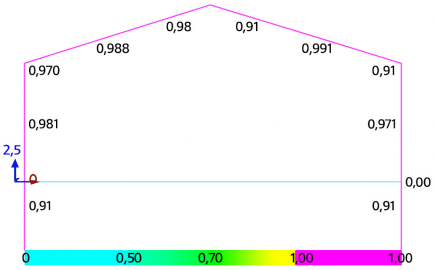
Pr = required axial compressive strength from 2nd ORDER ANALYSIS (from LRFD load combinations).

required axial compressive strength from 2nd ORDER ANALYSIS (from ASD load combinations).

Pc = available design axial compressive strength LRFD (strength from Chapter E).

```python
### Section Design According to AISC

14 usages
def design_section(hw, tf, tw, bf, P_u, M_u, V_u, Fyield, E, step, phi_c=0.85, phi_m=0.9, phi_v=1.0):
    hw = hw * 1000
    tf = tf * 1000
    tw = tw * 1000
    bf = bf * 1000
    Fyield = Fyield * 100  # Convert Fy to MPa

    """
    Function to design a steel section based on combined shear, axial, and bending forces.

    Parameters:
    hw : float
        Initial height of the web in mm (without subtracting tf).
    tf : float
        Initial flange thickness in mm.
    tw : float
        Initial web thickness in mm.
    bf : float
        Initial flange width in mm.
    P_u : float
        Applied axial load in kN.
    M_u : float
        Applied bending moment in kN-m.
    V_u : float
        Applied shear load in kN.
    Fyield : float
        Yield strength of steel in MPa (default is 345 MPa).
    E : float
        Modulus of Elasticity in MPa (default is 200,000 MPa).
    step : float
        Step size for incrementing hw during iteration
    phi_c : float
        Resistance factor for compression (default is 0.85).
    phi_m : float
        Resistance factor for moment (default is 0.9).
    phi_v : float
        Resistance factor for shear (default is 1.0).
```

(a) Sample python code for steel section design nased on AISC criteria



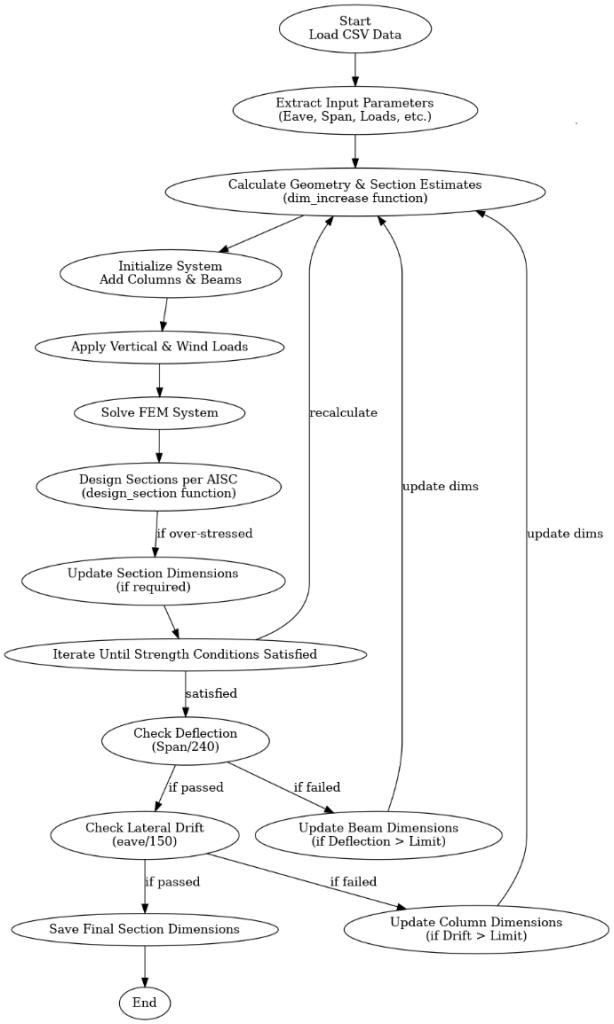
(d) PMM ratios of all elements from Etabs for Model 1 under strength design criteria

```python
import sys
from pythonnet import load
import comtypes.client

load("coreclr")
import clr

# Set the following path to the installed ETABS program directory
clr.AddReference(R'C:\Program Files\Computers and Structures\ETABS 22\ETABSv1.dll')
from ETABSv1 import *

# Attach to an existing instance of ETABS
AttachToInstance = True

if AttachToInstance:
    try:
        # Create API helper object
        helper = cHelper(Helper())

        # Get the active ETABS object
        myETABSObject = cOAPI(helper.GetObject("CSI.ETABS.API.ETABSObject"))
    except Exception as e:
        print("Failed to attach to an existing instance of ETABS.")
        sys.exit(-1)
else:
    print("Set AttachToInstance to True to attach to a running instance.")
    sys.exit(-1)

# Create SapModel object
SapModel = cSapModel(myETABSObject.SapModel)

# Set units using the enum directly
SapModel.SetPresentUnits(eUnits.kN_m_C)

# Run analysis
SapModel.Analyze.RunAnalysis()

# Set design code
```

(b) Python code example for initializing ETABS API

```python
#run model (this will create the
analysis model)
Analyze = cAnAlyze(SapModel.Analyze)
ret = Analyze.RunAnalysis()

ret = SapModel.DesignSteel.SetCode("AISC
360_22")
ret = SapModel.DesignSteel.StartDesign()
```

(c) Python script executing structural analysis and steel design using ETABS API with AISC 360-22 Code



(e) Flowchart of AISC design steps in the coding process

Fig. 3 Automated steel design workflow using python scripting and AISC code integration with ETABS API

Mr = required flexural strength (from LRFD load combinations).

Mc = available design flexural strength LRFD (strength from Chapter F).

The all-base FEM (Finite Element Method) ensures the safety of all structural models, as demonstrated by the strength-to-demand ratios presented in Fig. 3(d) considering Strength requirement only while Fig. 3(a) represents sample

Table 4 Different analysis cases

Models no.	Span (m)	Eave (m)	Live load (kN/m^2)	Dead load (kN/m^2)	Collateral load (kN/m^2)	Wind speed (m/s)	Unsupported length (m)	Support
Model 1	20	5	0.6	0.25	0.6	42	1.5	Hinged
Model 2	25	5	0.6	0.25	0.6	42	1.5	Hinged
Model 3	30	5	0.6	0.25	0.6	42	1.5	Hinged
Model 4	35	5	0.6	0.25	0.6	42	1.5	Hinged
Model 5	20	8	0.6	0.25	0.6	42	1.5	Fixed
Model 6	25	8	0.6	0.25	0.6	42	1.5	Fixed
Model 7	30	8	0.6	0.25	0.6	42	1.5	Fixed
Model 8	35	8	0.6	0.25	0.6	42	1.5	Fixed
Model 9	20	10	0.6	0.25	0.6	42	1.5	Fixed
Model 10	25	10	0.6	0.25	0.6	42	1.5	Fixed
Model 11	30	10	0.6	0.25	0.6	42	1.5	Fixed
Model 12	35	10	0.6	0.25	0.6	42	1.5	Fixed

Table 5 ETABS manual traditional steps column sections trials PMM ratio (Span = 35 m, Eave = 10 m)

Step	C6 hw (mm)	C6 tw (mm)	C1 bf (mm)	C6 tf Exterior (mm)	C6 tf Interior (mm)	C7 hw (mm)	C7 tw (mm)	C7 bf (mm)	C7 tf Exterior (mm)	C7 tf Interior (mm)	PMM
Trial 1	400	8	200	12	10	800	8	250	12	10	2.538
Trial 2	500	10	200	14	12	1000	8	250	14	12	1.11
Trial 3	500	12	250	16	12	1000	12	250	16	12	0.737
Trial 4	500	12	250	14	12	1000	12	250	14	12	0.772
Trial 5	500	10	200	14	12	1000	10	200	14	12	0.924

Table 6 Etabs manual traditional steps beams zones sections trials PMM ratio for (Span = 35 m, Eave = 10 m)

	Beam zone 1										Beam zone 2				
Step	B1 hw (mm)	B1 tw (mm)	B1 bf (mm)	B1 tf (mm)	B2 hw (mm)	B2 tw (mm)	B2 bf (mm)	B2 tf (mm)	PMM	Step	B3 hw (mm)	B3 tw (mm)	B3 bf (mm)	B3 tf (mm)	PMM
Trial 1	600	8	225	15	700	8	225	15	0.885	Trial 1	700	8	225	14	0.972
Trial 2	700	14	250	20	700	14	250	20	0.933	Trial 2	700	12	250	20	0.792
Trial 3	700	14	250	20	700	14	250	20	0.935	Trial 3	700	10	250	16	0.991
Trial 4	700	14	250	20	700	14	250	20	0.932	Trial 4	700	10	250	16	0.987
Trial 5	700	14	250	20	700	14	250	20	0.923	Trial 5	700	10	250	16	0.976

of code of AISC python coding:

Tables 5-6 present the step-by-step traditional manual optimization of cross-sections to ensure maximum demand utilization of the provided sections, while also considering the relative stiffness of different structural elements. The generated dataset ensures safety under all critical aspects without serviceability conditions where B1 height at section 1 and B2, B3 respectively, as detailed in Tables 7-8 including drift and deflection serviceability limits, these constraints led to a slight reduction in Axial-Moment Interaction Ratio (PMM) ratios, as compliance with the allowable limits was prioritized

2.2.2 Load case definition and parameterization

The load cases were meticulously designed to reflect realistic loading conditions across 39000 scenarios, including a variety of parameters influencing structural behavior. Key variables included steel density, yield strength, unsupported length, and seismic and wind zone classifications, among others. This extensive parameter set enabled a comprehensive assessment of the frames' responses to various conditions, capturing the nuanced effects of both static and dynamic loads on the structure. Constraints were applied to the section dimensions, setting ratios for the flange and web sizes (e.g., tf = Hw/37.5, tw = Hw/75, bf = Hw/3), ensuring proportional scaling with height. This standardized approach allowed the frame to

Table 7 ETABS output steps for a 30 m span integrated with deflection and drift limits

Frame dimensions	Span (m)	30		30		30	
	Eave (m)	10		8		5	
Dimensions (mm) and strength ratio		C./B. Dim.	PMM	C./B. Dim.	PMM	C./B. Dim.	PMM
Column	Hw C**6**	474	0.967	474	0.972	422	0.9326
	Hw C**7**	821.6	0.**8**049	832.2	0.8675	790	0.851
Beam	Hw B1	884.8	0.**8**318	832.2	0.8185	822	0.8423
	Hw B**3**	620	0.**8**247	583	0.8287	576	0.8205
	Hw B**4**	682	0.**8**454	642	0.8896	634	0.8389
Deflection and drift		Act. Def.	All. Def.	Act. Def.	All. Def.	Act. Drift	All. Drift
Def. 1 (mm)		50.118	83.33333	50.57	83.33333	49.193	83.33333
Def. 2 (mm)		124.9	125	124.6	125	120.95	125
Drift (mm)		20.814	66.66667	11.29	53.33333	9.822	33.33333

Table 8 ETABS output steps for a 35 m span integrated with deflection and drift limits

Frame dimensions	Span (m)	35		35		35	
	Eave (m)	10		8		5	
Dimensions (mm) and strength ratio		C./B. Dim.	PMM	C./B. Dim.	PMM	C./B. Dim.	PMM
Column	632	0.9582	547.8	0.923	474	0.9156	0.9326
	948	0.8505	916.4	0.8692	895.4	0.8743	0.851
Beam	948	0.8658	916.4	0.8066	895.4	0.8899	0.8423
	664	0.8338	642	0.9593	627	0.8582	0.8205
	731	0.8049	707	0.9185	690	0.8038	0.8389
Deflection and drift		Act. Def.	All. Def.	Act. Def.	All. Def.	Act. Drift	All. Drift
Def. 1 (mm)		56.313	97.222	58.37	97.222	56.162	97.222
Def. 2 (mm)		142.31	145.833	146.4	145.833	140.282	145.833
Drift (mm)		12.59	66.667	8.9	53.333	8.577	33.333

maintain structural integrity while meeting design constraints for strength, deflection, and drift. constraints for strength, deflection, and drift.

2.2.3 Data generation and structural modeling integration using the ETABS API

To generate a comprehensive dataset for training machine learning models, a custom Python-based framework was developed using finite element analysis (FEA) techniques. The structural configurations were modeled programmatically with parametric variations in eave height, bay span, and frame spacing. This automated generation ensured consistent and diverse structural configurations across the defined design space.

The finite element modeling process utilized the anastruct library, a Python-based FEA tool that allows for defining structural elements, applying loads, and solving load cases. Modules such as System Elements, Load Case, and Load Combination were employed to simulate real-world load conditions including dead, live, wind, and seismic loads. This approach enabled rapid and scalable generation of structural behavior data across various configurations.

In other parts of the study, integration with the ETABS Application Programming Interface (API) was also employed to validate straining actions and assist in section design based on AISC more detailed simulation outputs as below Figs. 3(b)-(c) for a sample;

2.2.4 Segmentation strategy for beams and columns

The column in the given structure follows a tapering design, where the cross-section transitions from a larger section at the base to a smaller one at the top. Specifically, for the columns, we made three splits: one at the start, one in the middle, and one at the top end. This tapering design effectively distributes stresses, optimizing material usage while maintaining structural stability and minimizing weight.

For the beam, the span is divided into three distinct zones. In the first zone, which extends from the column to a length of 0.3 L (where L is half the total span), we made 6 segments and the beam height (Hw) is at its maximum. After 0.3 L, Hw decreases by 30% and remains constant throughout zone 2, where we made 11 segments along 0.55 L. Finally, in zone 3, which spans 0.15 L towards the center,

we made 3 segments and the beam height (Hw) increases by 10%, improving structural performance in areas of higher moment demand. This strategic zoning enhances the overall efficiency of the beam while balancing strength and weight across the span as shown in Figs. 1-2-19.

2.2.5 ML & DL algorithms application notes

Selecting the right machine learning (ML) algorithm is essential for achieving accurate and efficient structural optimization. This comparison covers five widely used models—Polynomial Regression, LightGBM, XGBoost, Random Forest, and ANNs—highlighting their strengths and usage considerations.

Polynomial Regression: A simple baseline model that assumes a linear relationship between input features and target outputs. While limited in its ability to capture complex, nonlinear patterns, Linear Regression offers a reference point for more advanced models.

LightGBM: A gradient boosting model designed for high efficiency on large datasets, LightGBM is suitable for high-dimensional data due to its ability to capture complex interactions. It works particularly well for structured datasets and offers fast training with robust performance.

XGBoost: Another gradient-based boosting model, XGBoost is known for its flexibility and performance improvements over traditional boosting. It is adept at handling outliers and noise, making it valuable for datasets where even slight prediction improvements are critical.

Random Forest: An ensemble of decision trees, Random Forest can capture nonlinear dependencies and complex interactions within the data. It is generally resilient to overfitting and provides highly accurate predictions by averaging results from multiple decision trees.

Artificial Neural Network (ANN): The ANN model introduces deep learning into the framework. It consists of multiple hidden layers, each enabling the model to learn intricate patterns in the data. The ANN can handle complex relationships that might not be captured effectively by traditional ML models, making it a valuable addition to this comparison Performance Metrics. Table 9 provides a concise overview to support informed decision-making in structural design applications.

2.2.6 FEM-based simulation, section dimensions & safety

Tables 7-8 summarize structural dimensions, deflections, and drift ratios for different scenarios under all loads, ensuring compliance with AISC codes. Each case provides section dimensions and corresponding PMM (axial-moment interaction) ratios, along with deflection and drift evaluations for sections frame divided to two column sections Lower Section (Base) C6 with smaller cross section and Upper Section (Top) large C7 and beam frame is divided as Figs. 1-2 to Hw for the three zones, Zone1 Starts with Hw by the end of this segment, Zone 2 and section remains uniform along this length to achieve maximum optimization B3 , and Zone 3 height B4.

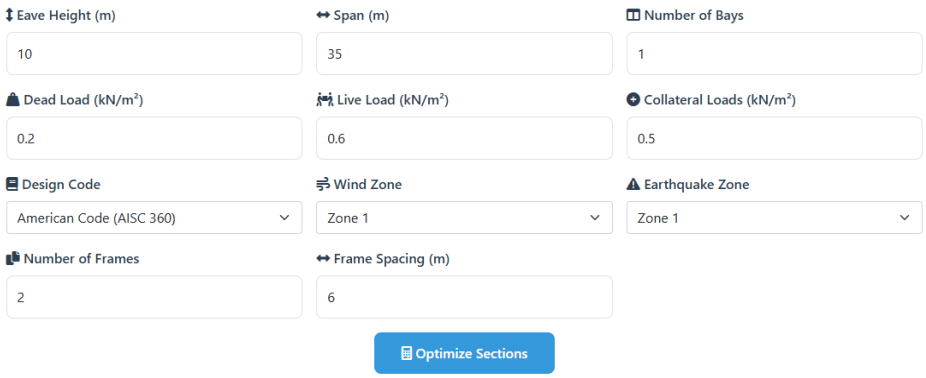
Deflection Type 1 (Def. 1) accounts for live load deflection, while Deflection Type 2 (Def. 2) considers all loads. Drift values measure lateral displacement, crucial for stability. For instance, at 10 m eave height, Def. 1 is 56.313 mm and Def. 2 is 142.31 mm, both within code limits. The PMM ratios (e.g., 0.9582 and 0.8505 for columns C6 and C7) indicate structural adequacy. The first row in each case details section dimensions, while the second assesses compliance with deflection and drift constraints. The results demonstrate adherence to AISC limits as rhe following flowchart in Fig. 3(e), maintaining structural safety and usability across varying heights and loads.

2.2.7 Graphical User Interface (GUI) development

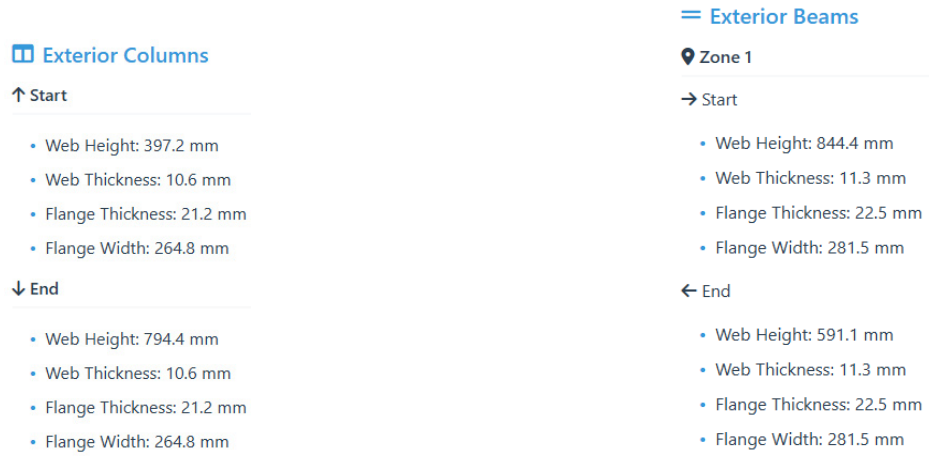
To facilitate practical application of the predictive framework, we developed a graphical user interface (GUI) that allows users to input specific design parameters and receive optimized section dimensions. The GUI aims to simplify the prediction process by enabling real-time, user-friendly interaction with the trained machine learning model. Although still in development, the interface is expected to enhance accessibility for engineers and designers, bridging the gap between complex machine learning outputs and practical, actionable design solutions as shown in Figs. 4(a)-(b).

Table 9 ML algorithms features and application notes

Algorithm	Advantages	Usage notes
Polynomial regression	Simple to implement and interpret; Good for modeling non-linear relationships (if degree is appropriate)	Requires careful selection of polynomial degree to avoid overfitting; best for data with clear mathematical trends.
LightGBM	Extremely fast training and prediction; Efficient with large datasets; Handles categorical features well; Built-in feature importance	Works best with proper hyperparameter tuning; highly effective for advanced structural applications.
XGBoost	Excellent accuracy and generalization; Robust to overfitting; Supports missing values; Handles complex nonlinearities	Performs best with well-tuned parameters; widely used in engineering for high predictive performance.
Random forest	Handles high-dimensional data; Robust to overfitting; Captures feature importance naturally; Less tuning than boosting models	Ideal for initial modeling phases; delivers reliable results, especially with many variables.
Artificial Neural Networks (ANNs)	Very powerful for complex nonlinear patterns; Flexible architecture; Scalable for large datasets; Can approximate any function	Needs sufficient training data and tuning for optimal performance; powerful tool in modern structural modeling.



(a) GUI user input



(b) GUI results sample of prediction (Output)

Fig. 4 User interface for parametric input and predicted output of optimal steel section dimensions

2.3 Distinction between predictive modeling and explicit structural optimization

It is important to highlight that the prediction process in this study was fundamentally based on a pre-generated dataset comprising carefully optimized sections—each designed to achieve the minimum possible dimensions and, consequently, the lowest feasible structural weight. This strategic data foundation ensures that the machine learning models do not merely replicate arbitrary design patterns but instead learn from high-efficiency structural configurations. As a result, the predicted sections inherently represent optimized outcomes.

Thus, the surrogate model serves as a practical and efficient alternative to conventional step-by-step structural design procedures, offering automatically optimized solutions without the need for repeated manual iterations.

3. Machine learning

3.1 ML methodology

In recent advancements in civil engineering, machine learning has demonstrated significant potential in enhancing structural engineering practices. understanding the core objectives and significance of this innovative approach. In the realm of structural engineering, the ability to accurately predict the ideal sections for beams and columns within steel frame structures is paramount.

Our machine learning models operate within predefined dataset input boundaries shaped by specific structural dimensions eave heights (5-10 m), bay spans (20-35 m), and spacing (6-10 m), also dead, live, collateral loads, in addition support condition fixed or hinged ensuring accuracy within typical commercial configurations. To further enhance model performance using 20% Testing and 80% training.

Structural components were labelled for clarity (e.g., ‘B’ for Beams, ‘C’ for Columns). Zones and splits were defined for detailed analysis. Graphical analyses focused on ‘B’ Z1 (Split1) height of section Hw’ as output.

3.2 Metrics explanation

Mean Absolute Error (MAE) is a measure of errors between paired observations expressing the same phenomenon. Essentially, it is the average of the absolute differences between the predicted values and the actual values without considering the direction of the error (i.e., no negative signs). It provides a straightforward indication of prediction accuracy where a value of 0 indicates no error, or perfect predictions. The formula for MAE is Eq. (3)

$$\mathrm{MAE} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}|\mathrm{x_i} - \mathrm{x}| \quad (3)$$

where: n = the number of errors, Σ = summation symbol (which means "add them all up"), |xi – x| = the absolute errors. MAE is particularly useful when you want to avoid large errors; it's less sensitive to outliers than MSE because it does not square the errors in its calculation. Mean Squared Error (MSE) is a measure that calculates the average squared difference between the estimated values and the actual value. By squaring the errors, MSE gives a larger weight to larger errors. This means that the presence of very large errors in the dataset will dominate the MSE, which can be both an advantage and a disadvantage depending on the context. The formula for MSE is Eq. (4)

$$\mathrm{MSE} = \frac{\sum(\mathrm{y_i} - \mathrm{y})^2}{\mathrm{n}} \quad (4)$$

where: yi is the ith observed value, ŷi is the corresponding predicted value., n = the number of observations. MSE is useful when large errors are particularly undesirable, and it is critically important to penalize these errors heavily. R-squared (Coefficient of Determination (R^2)) score, also known as the coefficient of determination, is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. It provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model, through the proportion of total variation of outcomes explained by the model. The formula for R^2 is Eqs. (5)-(7)

$$RSS = \sum_{I}^{n}\left(y_i - y_{pred_i}\right)^2 \quad (5)$$

$$TSS = \sum_{I}^{n}(y_i - y_{mean})^2 \quad (6)$$

$$R^2 = 1 - \frac{Residual\ sum\ of\ squares}{total\ sum\ of\ squares} = 1 - \frac{RSS}{TSS} \quad (7)$$

An R^2 score of 1 indicates perfect predictions with no error, while a score closer to 0 indicates that the model fails to accurately capture the target variable. Negative values of R^2 may occur when the chosen model does not follow the trend of the data, so the model fits worse than a horizontal line.

3.3 Dataset boundaries and limits

Our machine learning models operate within predefined dataset boundaries shaped by specific structural dimensions, including eave heights (5-10 meters), bay spans (20-35 meters), and spacing (6-10 meters). These dimensions are chosen based on typical configurations for steel frames in commercial and industrial buildings, aligning with prevailing construction practices. However, these constraints naturally limit the scope of the models' predictive accuracy to structures that fall within this defined range. While the models perform well within these parameters, they may exhibit bias or reduced accuracy when applied to structures with significantly different dimensions. Expanding the dataset to cover a broader range of structural dimensions could improve the models' adaptability, enabling optimization of a more extensive array of steel structures.

3.4 Simulated structural analysis

Each model was analyzed using the anastruct solver, which calculates displacements, internal forces, and moment interactions under the applied loads. The FEA setup ensured uniform application of materials, cross-sectional properties, and load combinations, following standard engineering assumptions. The results captured key performance indicators such as maximum deflections and stress responses for each frame.

The output from each simulation was recorded in a structured dataset, including the input geometric parameters and the resulting structural responses. This dataset formed the basis for training machine learning models that predict frame performance under different configurations.

3.5 Data structuring and quality verification

The simulation results were organized into tabular format to facilitate machine learning application. Each entry included dimensional inputs and corresponding outputs, such as displacement or stress, enabling algorithms to learn structural behavior patterns.

Verification of selected cases was conducted by comparing the simulation results to theoretical solutions based on structural engineering principles and code-based equations (e.g., AISC). This validation ensured the fidelity of the generated data and confirmed that the finite element simulations reflect realistic structural responses.

Post-prediction verification of design constraints has already been performed on randomly selected predicted frame sections. These were evaluated using a Python-based implementation of the AISC design code, integrated with straining actions extracted from ETABS. The calculated results were cross-validated and found to match the solver values

3.6 Data preparation and feature engineering

The first step in developing a reliable machine learning and deep learning framework was to prepare the dataset, which involved carefully curating input features and target variables. The features selected for the models included critical parameters such geometrical aspects (e.g., eave height, bay span, spacing), and environmental factors (e.g., wind velocity, seismic zone). These features were chosen to reflect the conditions under which steel structures operate, capturing both static and dynamic loads and the associated structural responses. Each frame was divided into multiple segments, with section dimensions (height of web Hw,

thickness of flange tft, thickness of web tw, and breadth of flange bf) as the target variables. This detailed segmentation enabled the models to capture localized variations within the frame under different loading conditions. The data was then normalized to ensure a uniform scale, enhancing the efficiency of the learning algorithms.

3.7 Feature importance analysis

To enhance the interpretability of the machine learning models, particularly the tree-based algorithms, a feature importance analysis was conducted. This analysis identified which input features (e.g., span, eave height, wind velocity) had the greatest impact on the predicted section dimensions, providing insights into the driving factors behind the model's decisions. Understanding feature importance not only validated the selection of input variables but also provided practical engineering insights, highlighting how specific parameters influence the optimization of steel frames. This interpretability is especially valuable for structural engineers who may use these models to inform design decisions.

Table 10 Metrics for the polynomial regression model

	MAE	MSE	Adjusted R^2 on training	Adjusted R^2 on testing
Polynomial regression model	0.63213	0.0613	80.639%	79.171%

Table 11 Metrics for the LightGBM mdel

	MAE	MSE	Adjusted R^2 on training	Adjusted R^2 on testing
LightGBM mdel	0.142	0.0569	90.581%	90.112%

Table 12 Metrics for the XGBoost model

	MAE	MSE	Adjusted R^2 on training	Adjusted R^2 on testing
XGBoost model	0.131	0.0520	91.391%	90.147%

Table 13 Metrics for the random forest model

	MAE	MSE	Adjusted R^2 on training	Adjusted R^2 on testing
Random forest model	0.336	0.01823	94.132%	92.728%

Table 14 Metrics for the ANN model

	MAE	MSE	Adjusted R^2 on training	Adjusted R^2 on testing
ANN model	0.212	0.0762	91.631%	90.541%

3.8 Models implementation

For the purpose of clarity and effective communication of the findings, the graphical analyses within this paper are concentrated on the variable 'B' Z1 (Split1) Hw'. Focusing on this specific segment allows for a more detailed and unobstructed visualization of data trends and model performances. By narrowing the scope to 'B' Z1 (Split1) Hw', the plots remain uncluttered, enabling a clearer interpretation of the relationships and residuals. This targeted approach ensures that the visualizations effectively illustrate the key aspects of the study, highlight key trends, and make the complex interactions between variables more comprehensible for the reader, thereby ensuring clear and effective data visualization, as illustrated in Tables 10-14, and Figs. 5-14.

3.8.1 Polynomial regression model degree 8

Table 10 presents the performance metrics of the Polynomial Regression model. The model recorded a Mean Absolute Error (MAE) of 0.63213 and a Mean Squared Error (MSE) of 0.0613. The adjusted R^2 values were 80.639% for the training set and 79.171% for the testing set,

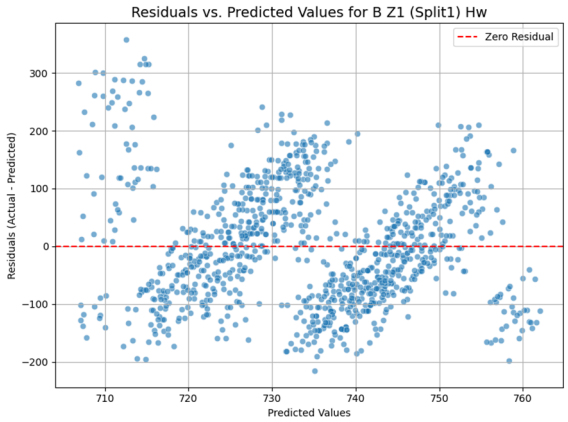


Fig. 5 Residual plotting for polynomial regression

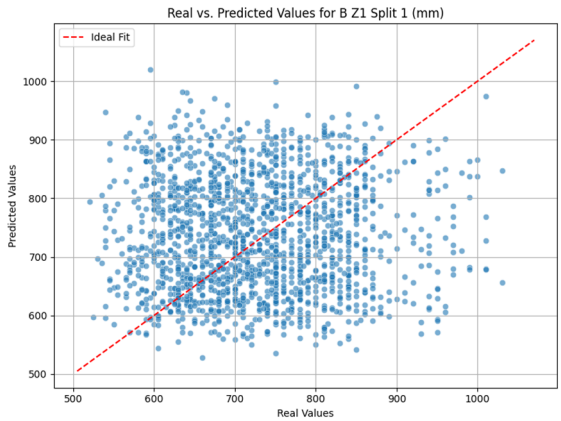


Fig. 6 Scatter plotting for LightGBM model

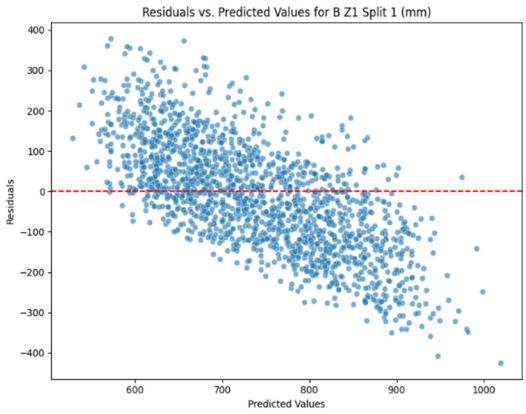


Fig. 7 Residual potting for LightGBM mModel

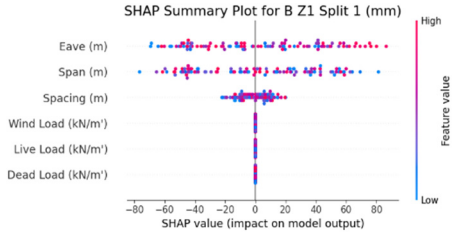


Fig. 8 SHAP potting for LightGBM model

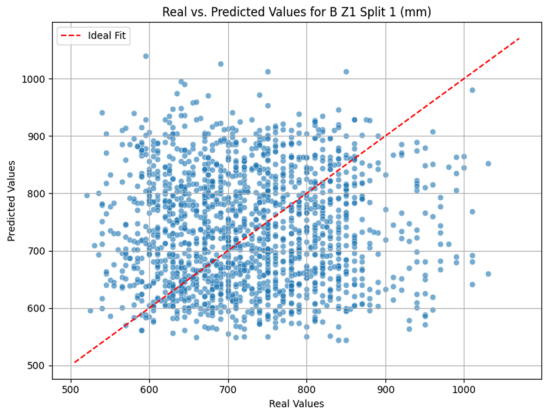


Fig. 9 Scatter plotting for XGBoost model

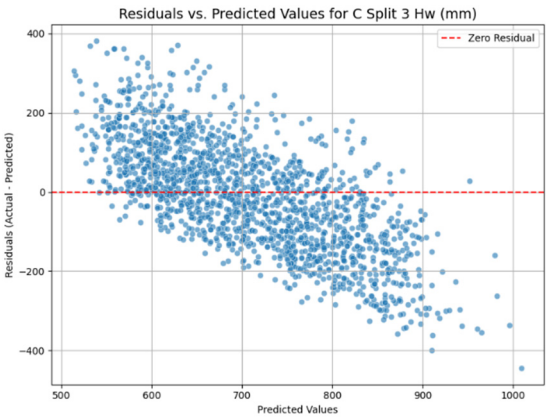


Fig. 10 Residual plotting for XGBoost model

indicating acceptable generalization performance with limited overfitting. However, the R^2 values fall short of the preferred threshold of 85%, suggesting that the model's accuracy is below expectations and may not sufficiently capture the complexity of the underlying relationships in the data.

3.8.2 LightGBM model

Table 11 reports the performance metrics of the LightGBM model, which yielded a low MAE of 0.142 and an MSE of 0.0569. The adjusted R^2 values reached 90.581% on the training set and 90.112% on the testing set, exceeding the typical threshold of 85%. These results confirm the model's high accuracy and its strong generalization capability. The performance meets expectations, indicating that the LightGBM model can be reliably adopted for predicting section dimensions.

The SHapley Additive exPlanations (SHAP) summary plot (Fig. 8) further confirms the model's robustness by identifying the most influential features affecting the prediction of cross-sectional dimensions—specifically, the height of the I-section, which directly impacts material cost. Among the input variables, spacing (m) exhibited the highest influence on the model's output, as indicated by its broad SHAP value distribution centered around zero and extending in both directions. This suggests that variations in spacing significantly affect the model's predictions. Spacing contributes most to the model's interpretability, followed by eave height (m) and span (m), emphasizing the engineering significance of these parameters in determining structural section behavior.

3.8.3 XGBoost model

Table 12 and Figs. 9-10 demonstrate similar behavior of the LightGBM model in terms of residual and scatter plotting. However, in the SHAP analysis in Fig. 11, span is identified as the most influential parameter affecting frame performance, rather than eave.

3.8.4 Random forest model

Table 13 presents the performance metrics of the Random Forest model, including MAE, MSE, and adjusted R^2 for both training and testing datasets. The model achieved a Mean Absolute Error (MAE) of 0.336 and a Mean Squared Error (MSE) of 0.01823, indicating high predictive precision. The adjusted R^2 values were 94.132% on training and 92.728% on testing, which demonstrates strong generalization with minimal overfitting.

Figs. 12-13 illustrates the scatter plot of predicted versus

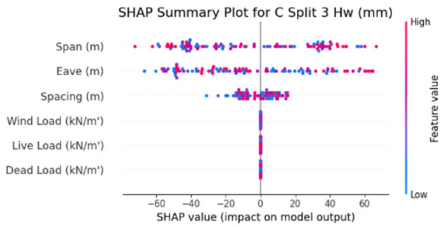


Fig. 11 SHAP plotting for XGBoost model

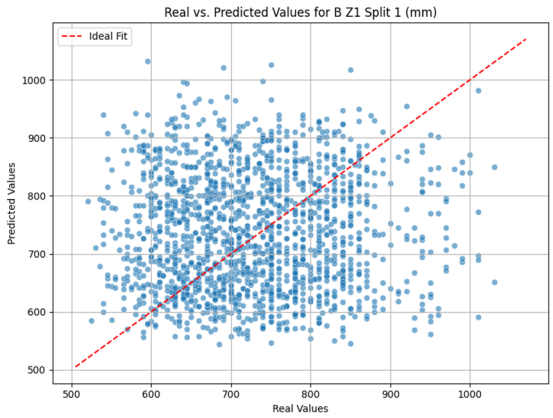


Fig. 12 Scatter plotting for random forest model

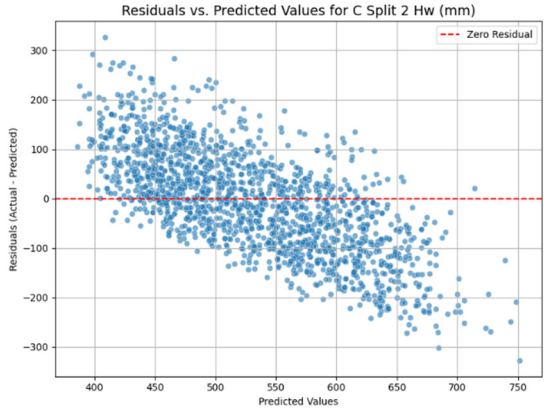


Fig. 13 Residual plotting for random forest model

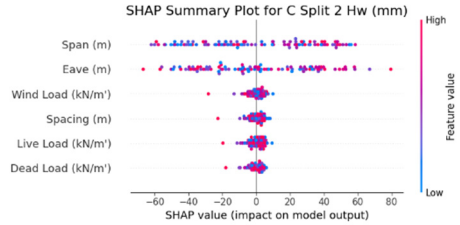


Fig. 14 SHAP plotting for random forest model

real values, showing moderate clustering around the ideal fit line, while Fig. 13 provides the residual distribution, indicating some bias at lower predicted values. Fig. 14 highlights SHAP values, confirming that frame height (eave) and span were the most influential input parameters in the model's prediction process.

The predicted values of random forest algorithm indicate that the height of section is mainly related to span, eave for geometry and wind for loading variables and figures indicate strong predictive capability with higher values of R^2.

3.9 Interpretability and engineering implications

The SHAP analysis conducted in this study provides valuable interpretability for the machine learning model's predictions. As illustrated in Figs. 8, 11 and 14, the span and eave height emerged as the most influential parameters, followed by spacing. Loads such as wind, live, and dead loads showed comparatively lower impact on the model's predictions. These findings do not only enhance the transparency of the machine learning process but also offer practical guidance for traditional optimization efforts.

Specifically, by highlighting the most sensitive design variables—such as the span and eave height—this analysis can help structural engineers prioritize which parameters to focus on when conducting rigorous mathematical optimization.

For instance, since the span significantly influences the predicted outputs, reducing the span or optimizing its configuration could lead to more efficient designs in traditional iterative workflows. Thus, the SHAP analysis serves as a useful complementary tool that can guide initial assumptions, parameter selection, and constraint prioritization in both predictive and optimization-based design strategies.

3.10 Feature importance comparison across models

Fig. 15 presents a comparative SHAP-based feature importance analysis across the three machine learning models utilized in this study: LightGBM, XGBoost, and Random Forest. The chart illustrates the estimated average SHAP values, reflecting the relative contribution of each input parameter to the model's predictions.

Across all models, the eave height and span consistently emerged as the most influential parameters, albeit with slight variations in ranking and magnitude among the models. Specifically, the LightGBM model exhibited the highest sensitivity to the eave height, while the XGBoost and Random Forest models showed a stronger association with the span. This reinforces the engineering understanding that both frame geometry parameters—span and eave—are critical drivers of section size and structural performance.

The spacing between frames also demonstrated a moderate impact across all models, whereas environmental loads, including wind load, live load, and dead load, showed comparatively lower influence. This trend suggests that geometric configurations have a more dominant effect

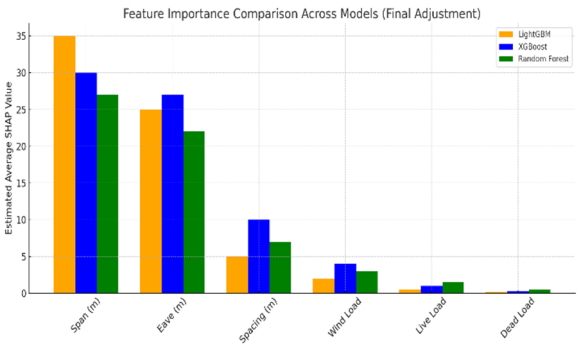


Fig. 15 Comparative feature importance based on SHAP values across all algorithms

on the predicted structural dimensions than applied loads within the analyzed range.

The alignment in feature importance rankings among the three models validates the robustness of the machine learning framework and confirms that the dominant design drivers remain consistent regardless of algorithm selection. This consistency enhances the interpretability and practical relevance of the proposed predictive system.

4. Deep learning

4.1 Introduction

In recent years, deep learning (DL) has emerged as a powerful tool in structural engineering, enabling the optimization of complex structures like steel frames by capturing intricate, nonlinear relationships in data (Marzouk *et al*. 2024b). Deep learning, particularly with neural network architectures, provides a robust framework for predictive modeling in scenarios where traditional engineering approaches face limitations due to complex patterns and high-dimensional datasets. Researchers like Pan *et al*. (2021) demonstrated the effectiveness of artificial neural networks (ANNs) in predicting structural response under variable load conditions, achieving notable improvements in accuracy compared to conventional techniques (Lagaros 2023). By leveraging deep learning, we can further enhance our capacity to analyze, predict, and optimize the performance of steel structures across varied configurations.

4.2 Model architecture and training

The core of our deep learning framework is an artificial neural network (ANN) designed to predict structural parameters and responses of steel frames under different loading conditions. Our model's architecture comprises an input layer, followed by multiple dense (fully connected) hidden layers with ReLU (Rectified Linear Unit) activations, and an output layer tailored for regression tasks. The selection of this architecture allows the model to capture complex patterns within the data and achieve high predictive accuracy.

The ANN model was trained using a dataset generated through simulated structural analyses, which included key structural features like dimensions, material properties, and load conditions. To further enhance model performance using 20% Testing and 80% training, the training strategy incorporated the following considerations:

Early Stopping and Validation Split: During training, a validation s (10% of the data) was used to monitor the model's performance on unseen data. The inclusion of early stopping, if applied, would halt training once validation loss ceases to improve, thus preventing overfitting.

Batch Size and Epochs: A batch size of 32 and 50 epochs were chosen as a balance between computational efficiency and the ability to capture complex patterns. These parameters were adjusted based on initial tests to ensure the model could generalize well without excessive training time. The chosen architecture consists of multiple layers arranged as follows:

4.2.1 Input layers

The model begins with an input layer, which takes a number of features based on the selected engineering parameters. For our structural dataset, this includes both categorical features (e.g., seismic zone, wind zone) and numerical features (e.g., bay span, eave height). Using TensorFlow's Input layer, we ensure that the model receives all required features in a format suitable for processing.

4.2.2 Hidden layers

The model includes Five dense (fully connected) hidden layers with progressively decreasing numbers of neurons: 128, 64, and 32, respectively. These layers allow the model to capture both high-level and finer details within the data.

Activation Function (ReLU): Each dense layer uses the ReLU (Rectified Linear Unit) activation function. ReLU is popular in deep learning due to its efficiency in handling nonlinear data, and it helps prevent issues like the vanishing gradient problem by allowing only positive values to pass through while zeroing out negative values. This approach is ideal for the varying relationships and dependencies among structural data.

4.2.3 Output layer

The final layer is a single neuron without an activation function (i.e., linear activation), which is typical in regression tasks, as it outputs continuous values corresponding to structural performance indicators. This value corresponds to the predicted structural response or parameter being optimized, such as deflection, PMM ratio, or a specific frame dimension.

The output layer consists of a single neuron with linear activation Ma *et al*. (2019), producing continuous predictions for parameters like deflection, PMM ratio, and frame dimensions.

4.2.4 Model architecture performance

Capacity to Capture Nonlinear Patterns: Structural engineering problems often involve nonlinear interactions between various design parameters, load effects, and structural responses. With multiple dense layers and ReLU activations, the model effectively learns these patterns.

Scalability: This architecture can be adapted for different types of structural data by adjusting the input shape and, if necessary, adding more layers or neurons to handle increased complexity or data volume.

Simplicity and Interpretability: Compared to more complex deep learning models (e.g., CNNs or RNNs), this fully connected ANN is straightforward yet powerful enough to provide accurate predictions. Its simplicity makes it easier to interpret, an advantage when communicating results to engineering professionals.

Figs. 16-17 represents results for Residual & scatter Plotting, and Figs. 18(a)-(b) represents results for losses, for Test & Training data Assessment for ANN Model. Table 14 presents the performance metrics of the Artificial Neural Network model, including MAE, MSE, and adjusted R^2 for both training and testing datasets. The model achieved a Mean Absolute Error (MAE) of 0.212 and a Mean

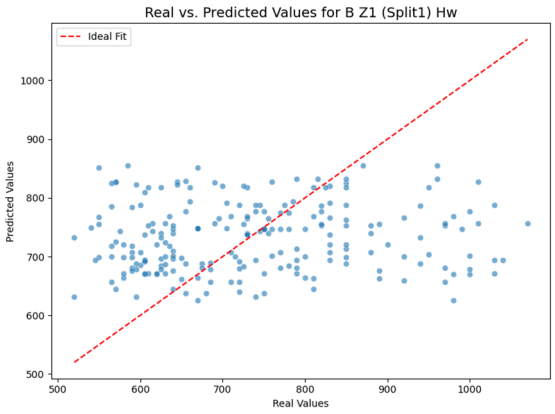


Fig. 16 Scatter plotting for ANN model

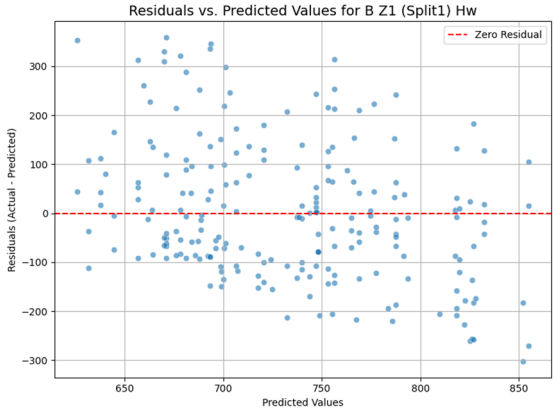


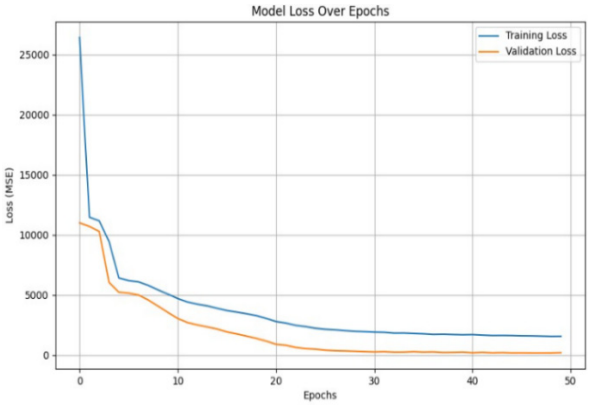
Fig. 17 Residual plotting for ANN model

Squared Error (MSE) of 0.0762, indicating high predictive precision. The adjusted R^2 values were 91.631% on training and 90.541% on testing, which demonstrates strong generalization with minimal overfitting.

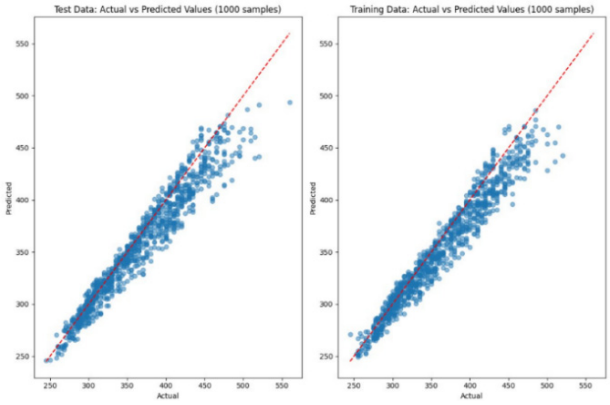
Fig. 18(a) shows a sharp initial decline in both training and validation losses during the first 30 epochs, indicating rapid learning in the early training phase. As training progresses, the losses gradually plateau, demonstrating model convergence and stability without signs of overfitting, as the validation loss consistently follows the training loss. In Fig. 18(b), the scatter plots for both training and test datasets reveal a strong linear correlation between the actual and predicted values, closely following the reference line (dashed red). This trend reflects the ANN model's robust performance, with minimal deviation across a wide range of output values, confirming high accuracy and excellent generalization across unseen data.

5. Problem statement, challenges and variables handling, study limits

The statement of the problem in this study was developing a methodology to effectively predict, design, and optimize steel sections that meet rigorous strength, deflection, and drift criteria under various geometric conditions in addition support condition fixed or hinged simultaneously. Balancing these parameters while ensuring structural safety and performance required a robust approach to both simulation and data handling, as each element of the steel frame design must meet specific engineering standards for strength and stability under diverse loading conditions.



(a) Model losses for ANN



(b) Test & training data assessment for ANN model

```python
# Train the XGBoost model
print("nTraining XGBoost model...")
xgb_model = XGBRegressor(
    n_estimators=100,
    learning_rate=0.1,
    max_depth=6,
    min_child_weight=1,
    subsample=0.8,
    colsample_bytree=0.8,
    random_state=42,
    n_jobs=-1 # Use all available c
)
```

(c) Regularization validation python code

Fig. 18 ANN model performance and regularization setup

Table 15 Prediction results sample of out of range trained data set

Section type	Parameter	Dim. (mm) for Eave 10 m	Dim. (mm) for Eave 11 m	Difference	Increasement %
Top column section	Hw_2	544.3	564.95	20.65	3.79
	tw_2	8	8.09	0.09	1.13
	bf_2	182.08	189.07	6.99	3.84
	tf_2	15.01	15.73	0.72	4.80
Section type	Parameter	Dim. (mm) for Eave 10 m	Dim. (mm) for Eave 11 m	Difference	Increasement %
Beam-column section	Hw_6	574.3	595.35	21.05	3.67
	tw_6	8	8.6	0.60	7.50
	bf_6	192.08	199.07	6.99	3.64
	tf_6	16	16.66	0.66	4.13

5.1 Integrated optimization approach

Addressing these challenges required an integrated approach that combines simulation tools with machine learning techniques to iteratively explore configurations that balance strength, Stability deflection, and drift performance. Finite element analysis using Python's *anastruct* library enabled rapid evaluation of diverse structural configurations. This framework allowed for dynamic refinement of section dimensions, material properties, and boundary conditions, ensuring that the final designs meet the required structural performance criteria while optimizing material efficiency, minimizing cost, and reducing design time

5.2 ML constraints and limitations

This setup allowed us to refine section dimensions, material choices, usage Málaga-Chuquitaype (2022), and support conditions dynamically, ensuring that the final designs meet the required performance standards while optimizing for material efficiency and maintaining structural integrity achieving prediction limits of R2 < 0.7, between 0.8 to 0.9 and more than 0.9 represents excellent and strong correlation with minimal unexpected unexplained variance. for mean squared error MSE <0.1 using normalized metrics if comparing across dataset.

ML constraints were embedded originally in the model and Machine learning (ML) models are typically trained on datasets that encompass specific ranges of structural parameters, such as loads, spans, geometries, and spacing. Consequently, their predictive accuracy may diminish when applied to conditions that fall outside these established ranges. This limitation underscores the necessity for additional studies and analyses to ensure the reliability of ML predictions in uncharted scenarios.

5.3 Integrating implicit regularization with ML

In this study, regularization was implicitly incorporated into the machine learning workflow through the configuration of the XGBoost model. Several key hyperparameters contributed to controlling model complexity and preventing overfitting. Specifically, the model was defined with max_depth = 6, min_child_weight = 1, subsample = 0.8, and colsample_bytree = 0.8, as shown in the code snippet below in Fig. 18(c).

These parameters act in Fig. 18(c) as structural and stochastic regularization mechanisms. For instance, max_depth limits the tree complexity, min_child_weight ensures splits occur only when sufficient data is available, while subsample and colsample_bytree introduce randomness by sampling data and features, respectively. Together, these contribute to the model's generalization capability, even without explicit regularization terms.

Table 16 Metrics comparison for all models

Model name	MAE	MSE	Adj. R^2 (Testing)	Adj. R^2 (Training)
Polynomial regression nodel	0.632	0.0613	80.639%	79.171%
Light GBM model	0.142	0.0569	90.581%	90.112%
XGBoost model	0.131	0.0520	91.391%	90.147%
Random forest model	0.336	0.0182	94.132%	92.728%
ANN model	0.212	0.0762	91.631%	90.541%

5.4 Sensitivity analysis and evaluation of model generalization capability

To further evaluate the generalization capability of the proposed machine learning-based predictive framework, a parametric sensitivity analysis was conducted using input scenarios that slightly exceeded the original training data boundaries. These scenarios included longer spans or eaves ranging about 10% about the data set, as well as increased load intensities beyond the range of the initial dataset, such as elevated live and wind loads.

The predicted section dimensions, PMM ratios, deflection, and drift values were compared against results obtained from high-fidelity finite element simulations. The

Table 17 Metrics comparison between ETABS & random forest algorithm App. results

Span-Eave (m)	BMD (%)	SFD (%)	NFD (%)	Deflection Etabs (mm)	Deflection App. (mm)	Deflection Etabs (mm)	Deflection App. (mm)	Drifting Etabs (mm)	Drifting App. (mm)	Tonnage Etabs (kg)	Tonnage App. (kg)
35-8	0.5	0.3	0.1	45.1	47.4	129.8	136.3	28.3	26.9	12674.3	13308
35-5	0.4	0.3	0.1	57.5	60.4	143.4	150.6	12.6	12.0	10833.6	11375
30-10	0.5	0.3	0.1	48.1	50.6	120.3	126.3	30.5	29.0	10666.3	11199
30-8	0.5	0.3	0.1	49.3	51.8	122.7	128.8	17.8	16.9	9289.3	9753.7
30-5	0.4	0.3	0.1	49.8	52.3	123.0	129.2	14.0	13.3	7878.6	8272.6
25-10	0.5	0.4	0.1	40.1	42.1	99.6	104.6	42.3	40.2	7534.8	7911.5
25-8	0.5	0.4	0.1	42.6	44.8	103.3	108.5	22.7	21.6	6486.4	6810.7
25-5	0.4	0.4	0.1	41.6	43.7	100.6	105.6	16.5	15.7	5381.5	5650.6
20-10	0.5	0.2	0.1	33.6	35.3	80.4	84.4	61.5	58.5	4561.7	4789.8
20-8	0.5	0.2	0.1	34.9	36.7	83.2	87.4	31.3	29.7	4301.4	4516.5
20-5	0.4	0.2	0.1	34.3	36.0	81.5	85.5	25.0	23.7	3463.7	3636.9

comparison revealed strong alignment, with weight differences remaining not exceed 5% and 4% for dimensions, such as the following study for eave equals 11 m instead of 10 m;

These findings in Table 15 demonstrate the robustness and adaptability of the predictive model, confirming its capacity to deliver reliable outputs even under conditions that differ moderately from the training domain. This supports the practical integration of the surrogate-driven prediction framework into early-stage structural design workflows, where rapid and code-compliant estimation of feasible section dimensions is required.

6. Results and discussion

6.1 Comparison of models

In structural engineering, traditional analytical approaches often struggle to fully capture the complex relationships between structural design parameters, such as eave heights, bay spans, and varied load distributions. Machine learning (ML) and deep learning algorithms offer significant advantages by identifying patterns and predictive insights that can help optimize steel frame designs more effectively. Marzouk *et al*. (2024b), Millán *et al*. (2021), Nguyen *et al*. (2020), and Nguyen and Tuan (2020).

This study compares the predictive performance of Linear Regression, LightGBM, XGBoost, Random Forest, and an Artificial Neural Network (ANN) as summarized in Table 17 in estimating structural responses. Each model was trained on a dataset incorporating structural design features and load scenarios, focusing on predicting beam and column dimensions across different zones.

6.2 Model performance matrix

Given the complexity and variability inherent in structural design parameters such as eave heights, bay spans, and load distributions, traditional modeling approaches can be enhanced by leveraging ML and DL algorithms that can handle multifaceted relationships and predict outcomes more effectively. This study investigates the applicability and effectiveness of several ML algorithms—Polynomial Regression, Linear Regression, and Random Forest—in predicting structural behaviors of steel frames under various loading conditions. These predictions focus on key structural components: columns and beams across multiple zones of a frame, each characterized by distinct spans and load-bearing requirements. By comparing the performance of these algorithms across different metrics—R^2 Score, Mean Absolute Error (MAE), and Mean Squared Error (MSE)—in Table 16 this research aims to determine the most reliable and accurate methods for predicting structural responses, thereby guiding more efficient and cost-effective engineering solutions.

Polynomial Regression, serving as a baseline, struggles with nonlinear relationships and is excluded from further comparison.

XGBoost achieves the highest testing accuracy (Adj. R^2 = 91.391%) with low MSE (0.0520), indicating strong predictive capability.

LightGBM follows closely, with slightly lower accuracy (Adj. R^2 = 90.581%), maintaining efficient training and robust performance.

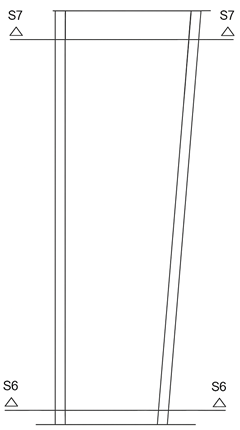


Fig. 19 Column section elevation

Table 18 Comparison between traditional (real) weights of columns and the best predicted values from the ML module

Span-Eave (m)	Hw6 (mm)	Tw6 (mm)	Bf6 (mm)	tf Exterior 6 (mm)	tf Interior 6 (mm)	Hw7 (mm)	Tw7 (mm)	Bf7 (mm)	tf Exterior 7 (mm)	tf Interior 7 (mm)	Traditional weight (kg)	Predicted weight results (kg)
20 – 6	250	5	175	8	8	500	5	175	8	8	220.193	209.183
20 – 8	250	5	175	8	8	500	5	250	8	8	331.270	319.013
20 – 10	400	6	200	10	8	600	6	225	10	9	582.863	512.919
25 – 6	300	7	175	10	9	300	8	175	10	10	310.860	270.448
25 – 8	400	7	175	11	9	500	9	200	11	10	508.680	432.378
25 – 10	500	7	200	13	12	500	10	250	13	10	812.475	739.352
30 – 6	400	8	200	11	10	400	8	200	11	10	388.575	359.432
30 – 8	500	10	200	13	12	500	10	250	13	12	569.910	530.016
30 – 10	500	10	200	13	12	500	10	200	12	12	1030.313	968.494
35 – 8	400	8	200	12	10	800	10	200	12	10	612.300	557.193
35 - 10	500	10	200	14	12	1000	10	200	14	12	996.950	797.560

Table 19 Comparison between real weights of beams in zone 1 and the best predicted values from the ML module

Span-Eave (m)	hw1 (mm)	tw1 (mm)	bf1 (mm)	tf1 (mm)	hw2 (mm)	tw2 (mm)	bf2 (mm)	tf2 (mm)	Traditional weight (kg)	Predicted weight results (kg)
20 - 6	400	5	175	10	300	5	175	10	61.82	61.20
20 - 8	400	5	175	9	300	5	175	9	60.64	59.43
20 - 10	400	6	175	9	300	6	175	9	65.35	62.08
25 - 6	650	12	200	12	500	12	200	15	184.35	179.74
25 - 8	600	8	200	12	500	8	200	12	141.30	138.47
25 - 10	600	8	200	12	500	8	200	12	141.30	139.89
30 - 6	600	7	200	12	500	7	200	12	158.96	157.37
30 - 8	600	7	200	12	500	7	200	12	158.96	157.37
30 - 10	600	7	200	12	500	7	200	12	158.96	152.60
35 - 6	600	13	250	18	500	13	250	18	386.37	367.05
35 - 8	600	8	225	15	500	8	225	15	246.24	224.08
35 - 10	700	14	250	20	700	14	250	20	408.00	395.76

Random Forest outperforms all models with an Adj. R^2 of 94.132%, demonstrating its strength in capturing nonlinear dependencies.

ANN performs well (Adj. R^2 = 91.631%) but slightly underperforms compared to Random Forest However, its ability to model high-dimensional relationships makes it valuable for structural predictions.

Polynomial Regression shows moderate performance (Adj. R^2 = 80.639%), indicating its limitations in handling complex structural interactions.

And for specific results compression between Random Forest algorithm application (App.) results and main models in ETABS as bending (BM), shear (SF), Normal forces (NF), deflection, drift, and tonnage for frame components) weight presents in Table 17.

Under identical geometric and loading conditions, the minimal differences observed in straining actions—bending moment (BM), shear force (SF), and normal force (NF)—as well as in deflection and drift values, result in material tonnage estimates with a variation not exceeding 5%, indicating negligible impact on overall steel cost.

6.3 Comparison for weights

This section compares the predicted structural weights with those obtained from traditional ETABS analysis to evaluate the accuracy of the machine learning model. Subsection 6.3.1 focuses on columns, and Subsection 6.3.2 focuses on Beams with results summarized in Tables 18-19 and illustrated in Figs. 20-21.

6.3.1 Columns study

Results from ETABS compared to the predicted results are presented in Table 18 and Fig. 20.

The results of Table 18 and Fig. 20 indicate a range of discrepancies between the predicted weights from the machine learning models and the actual weights calculated using ETABS software. Analyzing the data:

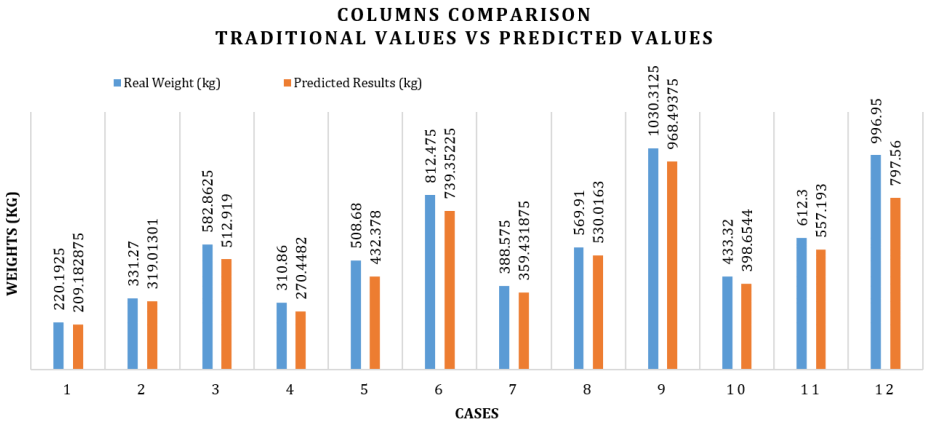


Fig. 20 Graphical representation between real weights of columns and the best predicted values from the ML module

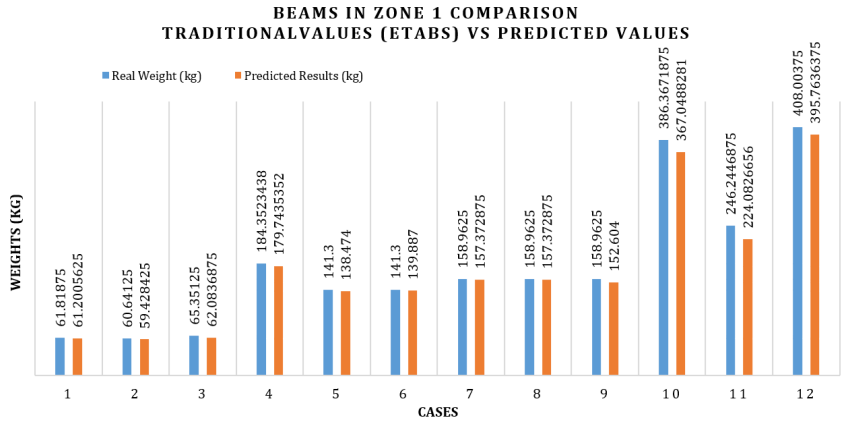


Fig. 21 Graphical representation between real weights of beams in zone 1 and the best predicted values from the ML module

For lower weight columns (around 220 kg to 330 kg), the predicted values are relatively close to the traditional ETABS values, with discrepancies ranging from approximately 5% to 10%.

Moderate weight columns (around 310 kg to 610 kg) show a larger deviation in predictions, with errors increasing up to about 20%.

The highest discrepancies are observed in columns with higher weights (greater than 800 kg), where prediction errors reach up to 20%, and in one instance, nearly 20%.

6.3.2 Beams in zone 1 study

Results from ETABS compared to the predicted results are presented in Table 19 and Fig. 21.

In assessing the machine learning predictions against the ETABS values for beams in Zone 1 in Table 21 and Fig. 21, a pattern of closely matched predictions emerges:

Light Beams: For beams with weights around 60 kg to 65 kg, the predictions are quite accurate, showing only minimal discrepancies ranging from 0.7% to approximately 5%. This suggests a high degree of model accuracy for lower weight categories.

Medium Beams: Weights in the mid-range (approximately 140 kg to 185 kg) also display good model performance, with predictions deviating by about 2% to 3%.

Notably, the model consistently handles beams within this weight spectrum effectively.

Heavier Beams: For beams weighing about 160 kg, the predictions exhibit slight variations; two predictions perfectly match the actual values, while one shows a deviation of about 4%. This variance could indicate potential issues in feature consistency or model sensitivity at slightly higher weights.

Heaviest Beams: Beams over 200 kg show greater prediction errors. For a 386 kg beam, the prediction undershoots by about 5%, and for a 246 kg beam, the error

Table 20 Feature importance

Feature importance	Parameter	Typical range (m)-(kN/m²)	Impact on structural behavior
1	Span (m)	20 – 35	Economy, secondary member design.
2	Frame height (m)	5 – 10	Influences lateral stability, wind loads
3	Spacing (m)	5 – 10	Influences lateral stability, wind loads
4	Live load (kN/m²)	0.6	Critical for serviceability, ultimate limits.
5	Dead load (kN/m²)	0.25	Permanent loads affecting long-term deflection

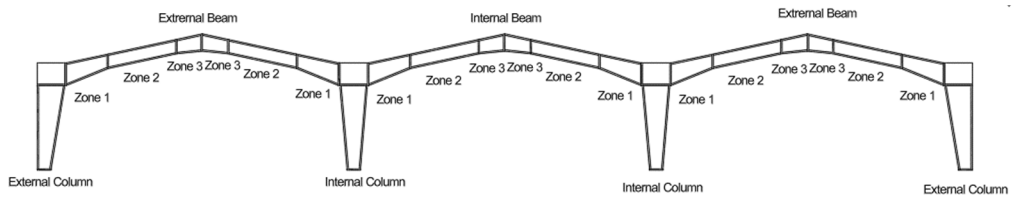


Fig. 22 Frame multi span steel structure layout

Table 21 Metrics comparison between ETABS & ML algorithm (App.) results for multi span frames

Span-Eave (m)	BMD (%)	SFD (%)	NFD (%)	Deflection Etabs (mm)	Deflection App. (mm)	Deflection Etabs (mm)	Deflection App. (mm)	Drifting Etabs (mm)	Drifting App. (mm)	Tonnage Etabs (kg)	Tonnage App. (kg)
35-8	0.5	0.3	0.1	45.1	47.4	129.8	136.3	28.3	26.9	12674.3	13308
35-5	0.4	0.3	0.1	57.5	60.4	143.4	150.6	12.6	12.0	10833.6	11375.3
30-10	0.5	0.3	0.1	48.1	50.6	120.3	126.3	30.5	29.0	10666.3	11199.7
30-8	0.5	0.3	0.1	49.3	51.8	122.7	128.8	17.8	16.9	9289.3	9753.7
30-5	0.4	0.3	0.1	49.8	52.3	123.0	129.2	14.0	13.3	7878.6	8272.6
25-10	0.5	0.4	0.1	40.1	42.1	99.6	104.6	42.3	40.2	7534.8	7911.5
25-8	0.5	0.4	0.1	42.6	44.8	103.3	108.5	22.7	21.6	6486.4	6810.7
25-5	0.4	0.4	0.1	41.6	43.7	100.6	105.6	16.5	15.7	5381.5	5650.6
20-10	0.5	0.2	0.1	33.6	35.3	80.4	84.4	61.5	58.5	4561.7	4789.8
20-8	0.5	0.2	0.1	34.9	36.7	83.2	87.4	31.3	29.7	4301.4	4516.5
20-5	0.4	0.2	0.1	34.3	36.0	81.5	85.5	25.0	23.7	3463.7	3636.9

approaches 9%. The largest beam, weighing 408 kg, has a prediction error of about 3%, indicating a slightly better model response to the upper weight range compared to the mid-range.

6.4 SHAP analysis discussion

A detailed feature importance analysis, as presented in Table 20, identified span (20-35 m) as the most influential parameter affecting the predictive performance of machine learning models. Span plays a pivotal role in distributing loads, determining structural economy, and guiding the design of secondary members. Following span, frame height (eave) (5-10 m) was the second most critical feature, significantly impacting lateral stability, wind resistance, and the natural frequency of the structure. The spacing between frames (5-10 m) ranked third in influence, contributing to variations in frame stiffness and lateral system behavior, albeit with a more moderate effect.

In contrast, live load (0.6 Kilonewtons per Square Meter (kN/m²)) and dead load (0.25 kN/m²) had a lesser impact on model outputs, which may be attributed to their relatively uniform application across the dataset. These findings align with prior studies emphasizing the dominant role of geometric parameters in structural optimization and machine learning-based prediction models Zhang and Wang (2023), Kim *et al*. (2022), Ahmed and Fekry (2024), and Gao and Li (2022).

In determining the cross-sectional dimensions, the height (Hw) of the I-section emerged as the most impactful feature, directly influencing both material cost and structural efficiency. This parameter is primarily sensitive to variations in geometric features. While frame span was identified as the most influential overall, the LightGBM model exhibited the highest sensitivity to frame height (eave).

7. Multi span study

The study of multi bays steel moment resisting steel frames, involves the development of a machine learning model designed to predict the optimal steel sections for frame elements in a structural system. The model takes into account various load cases and predicts the tonnage, section dimensions, and straining actions for different frame elements. The frame consists 3-bay configurations as Fig. 22 with standardized sectioning rules for beams and columns.

The overall differences in straining actions are less than 1%, while the discrepancies in predicted values using Random Forest algorithm for tonnage, deflection, and drift compared to traditional results fall within a range of approximately 5-7% in Table 21 as follows;

8. Conclusions

This study demonstrates the transformative potential of machine learning (ML) and deep learning (DL) in optimizing steel structure design, achieving high predictive accuracy and computational efficiency. The key findings are

summarized as follows:

- *Framework clarification*

This study explicitly positions its methodology as a *surrogate-driven prediction framework*, which leverages data-driven machine learning models to predict feasible structural dimensions. It is important to note that this approach is fundamentally distinct from *explicit mathematical optimization formulations* that solve strict objective functions under predefined constraints. Instead, the presented framework offers rapid, reliable predictions based on learned patterns from pre-generated datasets comprising optimized section designs.

- *Exceptional ML model performance*

Advanced machine learning models—XGBoost, LightGBM, and Random Forest—consistently delivered high accuracy, with adjusted R^2 scores exceeding 90% and low Mean Absolute Error (MAE) values. Notably, the Random Forest model achieved a robust R^2 of 94.132%, while XGBoost also performed well (R^2: 91.391%, MAE: 0.131, MSE: 0.0520), outperforming other algorithms in capturing nonlinear relationships among structural parameters, loads, and environmental factors. In contrast, Polynomial Regression is not recommended for this type of structural prediction due to its comparatively lower accuracy and limited generalization capability.

- *Deep learning performance*

The Artificial Neural Network (ANN) demonstrated strong capability in modeling high-dimensional interactions, achieving an R^2 of 91.631%, which is comparable to the performance of the XGBoost model. While ANN performed well, XGBoost maintained a slight advantage in terms of Mean Squared Error (MSE) and Mean Absolute Error (MAE).

- *Validation against traditional method*

The Random Forest model produced results that closely matched ETABS outputs for bending, shear, deflection, drift, and tonnage across 12 structural models, validating its reliability for real-world engineering applications.

All models—LightGBM, XGBoost, Random Forest, and ANN—achieved R^2 values greater than 0.9 and MSE values below 0.1, indicating strong predictive correlations with minimal unexplained variance, which is a critical benchmark for engineering-level accuracy.

- *Column dimensions and weights*

The comparison between predicted and actual ETABS values for column weights provides valuable insights into the performance of the applied machine learning models. The models showed strong alignment for lighter columns, and although slight deviations appeared with heavier columns, predictions remained within an acceptable range. These observations suggest opportunities for further enhancement—such as incorporating additional features or refined preprocessing—to improve accuracy across a broader spectrum of structural weights. Overall, the models demonstrated solid potential in capturing weight trends for varying.

- *Beam dimensions and weights*

The predictive analysis for beams in Zone 1 demonstrates that the employed machine learning models are particularly effective at handling lighter beams, with high accuracy noted in the lowest weight brackets. As the beam weight increases, the model exhibits varied performance, with a noticeable decrease in accuracy for medium to heavy beams, albeit still maintaining reasonable proximity to the actual values.

- *Practical integration*

The GUI enabled ML framework bridges theoretical analysis and practical design, enabling engineers to rapidly derive optimal sections with minimal computational effort.

- *Multi-span predictions*

Using XGBoost presented a specific predictive challenge but demonstrated strong alignment with both straining actions and tonnage, further validating its effectiveness in handling complex structural configurations. Notably, the prediction differences did not exceed 7%, indicating a reliable level of accuracy and supporting effective cost control practices.

- *Broader implications*

These data-driven methodologies complement traditional design approaches, enhancing efficiency, adaptability, and cost-effectiveness while adhering to AISC, American Society of Civil Engineers ASCE, and ECP standards.

- *Model generalization capability*

To validate the predictive robustness of the developed ML framework, a parametric sensitivity analysis was conducted using inputs beyond the original training range—such as longer spans and eaves. The model maintained reliable predictive performance under these conditions. These findings confirm that the surrogate-based framework not only performs accurately within the trained domain, but also generalizes with robustness to slightly extrapolated scenarios.

- *Sensitivity of frame parameters*

In determining the cross-sectional dimensions, the height of the I-section emerged as the most impactful feature, directly influencing both material cost and structural efficiency. This parameter is primarily sensitive to variations in geometric features. While frame span was identified as the most influential overall, the LightGBM model exhibited the highest sensitivity to frame height (eave).

- *Future work*

Could explore expanding the dataset to include a broader range of structural configurations—such as variations in span, spacing, and height—as well as additional load conditions, including crane loads. It may also involve incorporating other structural elements like connections, refining deep learning models to handle more complex scenarios, and conducting real-world validation to ensure practical applicability across diverse engineering

contexts.

References

Abarkan, I., Rabi, M., Vendramell Ferreira, F.P., Shamass, R., Limbachiya, V., Jweihan, Y.S. and Santos, L.F.P. (2024a), "Machine learning for optimal design of circular hollow section stai/nless steel stub columns: A comparative analysis with Eurocode 3 predictions", *Eng. Appl. Artif. Intell.*, **132**, p. 107952. https://doi.org/10.1016/j.engappai.2024.107952

Abarkan, I., Rabi, M., Vendramell Ferreira, F.P., Shamass, R., Limbachiya, V., Jweihan, Y.S. and Pinho Santos, L.F. (2024b), "Machine learning for optimal design of circular hollow section stainless steel stub columns: A comparative analysis with Eurocode 3 predictions", *Eng. Appl. Artif. Intell.*, **132**, p. 107952. https://doi.org/10.1016/j.engappai.2024.107952

Ahmed, A. and Fekry, M. (2024), "Comparative evaluation of machine learning algorithms for structural frame design optimization", *J. Build. Eng.*, **78**, p. 107527. https://doi.org/10.1016/j.jobe.2024.107527

American Institute of Steel Construction (2022), Specification for Structural Steel Buildings (AISC 360-16, 2022 ed.), American Institute of Steel Construction; Chicago, IL, USA.

Barakat, S., Alhalabi, M., Mostafa, O. and Haj Fattouh, I. (2022), "Size optimization of truss structures using calibrated shuffled complex evolution algorithm", *Jordan J. Civil Eng.*, **16**(4).

Basta, A., Serror, M.H. and Marzouk, M. (2020), "A BIM-based framework for quantitative assessment of steel structure deconstructability", *Autom. Constr.*, **111**, p. 103064. https://doi.org/10.1016/j.autcon.2019.103064

Bonafide Research (2025), Middle East & Africa Structural Steel Market Outlook, 2030; Bonafide Research Pvt. Ltd, Vadodara, India. https://www.bonafideresearch.com/product/250299676/middle-east-and-africa-structural-metal-market

Cicconi, P., Germani, M., Bondi, S., Zuliani, A. and Cagnacci, E. (2016), "A design methodology to support the optimization of steel structures", *Procedia Cirp*, **50**, 58-64. https://doi.org/10.1016/j.procir.2016.05.030

Dhiman, S., Thakur, N. and Sharma, N. (2019), "A review on behaviour of columns of steel framed structure with various steel sections", *Int. J. Eng. Technol.*, **6**, 587-590.

Gao, R. and Li, J. (2022), "Multi-objective optimization of pre-engineered buildings considering cost, weight, and constructability", *Structures*, **38**, 971-982. https://doi.org/10.1016/j.istruc.2022.05.027

Gorriz, J.M., Clemente, R.M., Segovia, F., Ramirez, J., Ortiz, A. and Suckling, J. (2024), "Is k-fold cross validation the best model selection method for machine learning?", arXiv preprint, arXiv:2401.16407. https://arxiv.org/abs/2401.16407

Kavya, B.R., Sureshchandra, H.S., Prashantha, S.J. and Shrikanth, A.S. (2022), "Prediction of mechanical properties of steel fiber-reinforced concrete using CNN", *Jordan J. Civil Eng.*, **16**(2).

Kim, H., Lee, J. and Park, S. (2022), "Deep learning-based prediction of optimal beam sections for modular structures", *Autom. Constr.*, **136**, p. 104191. https://doi.org/10.1016/j.autcon.2022.104191

Lagaros, N.D. (2023), "Artificial neural networks applied in civil engineering"., *Appl. Sci.*, **13**(2), p. 1131.

Ma, Y., Matta, E., Meissner, D., Schellenberg, H. and Hinkelmann, R. (2019), "Can machine learning improve the accuracy of water level forecasts for inland navigation? Case study: Rhine River Basin, Germany", In: *38th IAHR World Congress Panama City 2019*, *Water-Connecting the World*, pp. 1979-1989. https://doi.org/10.3850/38WC092019-0274

Málaga-Chuquitaype, C. (2022), "Machine learning in structural design: An opinionated review", *Front. Built Environ.*, **8**, Article 815717. https://doi.org/10.3389/fbuil.2022.815717

Marzouk, M., Elhakeem, A. and Adel, K. (2024a), "Artificial neural networks applications in construction and building engineering (1991–2021): Science mapping and visualization", *Appl. Soft Comput.*, **152**, p. 111174. https://doi.org/10.1016/j.asoc.2023.111174

Marzouk, M., El-Attar, M. and Youssef, M. (2024b), "Deep Learning-Based Prediction of Optimal Section Dimensions in Structural Design", *J. Struct. Eng. Applicat.*, **59**(2), 112-128. https://doi.org/10.1016/j.struceng.2024.01.005

Millán, M., Galindo-Aires, R. and Alencar, A. (2021), "Application of artificial neural networks for predicting the bearing capacity of shallow foundations on rock masses", *Rock Mech. Rock Eng.*, **54**. https://doi.org/10.1007/s00603-021-02549-1

Nguyen, T.-H. and Tuan, V.A. (2020), "Application of artificial intelligence for structural optimization", (Tien Khiem, N., Van Lien, T., Xuan Hung, N. eds.), *Modern Mechanics and Applications*, Lecture Notes in Mechanical Engineering. https://doi.org/10.1007/978-981-16-3239-6_82

Nguyen, M.T., Thai, D.K. and Kim, S.E. (2020), "Predicting the axial compressive capacity of circular concrete filled steel tube columns using an artificial neural network", *Steel Compos. Struct., Int. J.*, **35**(3), 415-437. https://doi.org/10.12989/scs.2020.35.3.415

Noori, F. and Varaee, H. (2022), "Nonlinear seismic response approximation of steel moment frames using artificial neural networks", *Jordan J. Civil Eng.*, **16**(1).

Pan, X., Wen, Z. and Yang, T. (2021), "Dynamic Analysis of Nonlinear Civil Engineering Structures Using Artificial Neural Network with Adaptive Training".

Saleh, Y.N., Mourad, S.A. and Ibrahim, A.M. (2024). "Topology optimization of vertical shear links in eccentrically braced frames", In: *Structures*, **66**, p. 106821. https://doi.org/10.1016/j.istruc.2024.106821

Salehi, H. and Burgueño, R. (2018), "Emerging artificial intelligence methods in structural engineering", *Eng. Struct.*, **171**, 170-189. https://doi.org/10.1016/j.engstruct.2018.05.084

Senatore, G. and Wang, Y. (2024), "Topology optimization of adaptive structures: New limits of material economy", *Comput. Methods Appl. Mech. Eng.*, **422**, p. 116710. https://doi.org/10.1016/j.cma.2023.116710

Senatore, G., Wang, Y., Virgili, F. and Blandini, L. (2025), "Global optimal actuator placement for adaptive structures: New formulation and benchmarking", *J. Intell. Mater. Syst. Struct.*, **36**(3), 151-173. https://doi.org/10.1177/1045389X241293861

Torky, A.A. and Aburawwash, A.A. (2018), "A deep learning approach to automated structural engineering of prestressed members", *Int. J. Struct. Civil Eng. Res.*, **7**(4), 347-352. https://doi.org/10.18178/ijscer.7.4.347-352

Zakhera, A., Salem, S. and Torky, A.A. (2024), "Optimizing column location in flat slab system using Particle Swarm Optimization vs Genetic Algorithms", In: *IOP Conference Series: Earth and Environmental Science*, Vol. 1396, No. 1, p. 012008. https://doi.org/10.1088/1755-1315/1396/1/012008

Zhang, Y. and Wang, D. (2023), "Optimization of steel frame structures under combined loading using machine learning approaches", *Eng. Struct.*, **292**, p. 115567. https://doi.org/10.1016/j.engstruct.2023.115567

Abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
ASCE	American Society of Civil Engineers
AISC	American Institute of Steel Construction
bf	Breadth of Flange
CNN	Convolutional Neural Network
DL	Deep Learning
ETABS	Extended 3D Analysis of Building Systems
FEM	Finite Element Method
GUI	Graphical User Interface
Hw	Height of Web
kN/m²	Kilonewtons per Square Meter
LightGBM	Light Gradient Boosting Machine
ML	Machine Learning
MAE	Mean Absolute Error
MSE	Mean Squared Error
PMM	Axial-Moment Interaction Ratio
PSO	Particle Swarm Optimization
R^2	Coefficient of Determination
ReLU	Rectified Linear Unit
SHAP	SHapley Additive exPlanations
tf	Thickness of Flange
tw	Thickness of Web