ДГКС ASES | E-12
Kongres 2026
Congress 2026

doi: 10.67563/ASESCONF2026069H

Amina Hajdarević[1], Martin Vild[2], Filip Ljubinković[3], Luís Simões da Silva[3]

COMPARISON AND TUNING OF MACHINE LEARNING MODELS FOR COLUMN WEB PANEL IN TRANSVERSE COMPRESSION

Summary: The component method enables efficient design of steel joints by decomposing them into individual components characterized by their resistance and stiffness. This paper presents a machine learning-based approach for resistance prediction of the column web panel in transverse compression, based on a numerical dataset generated using IDEA StatiCa Connection. Three machine learning models are investigated and compared: a deep neural network, Random Forest, and XGBoost. Particular attention is given to the influence of model tuning on predictive performance, where the deep neural network is tuned using a manual sensitivity-based procedure, while Random Forest and XGBoost are tuned using grid search. The results highlight differences in model behavior, sensitivity to tuning, and overall prediction accuracy. The study emphasizes the importance of appropriate model selection and tuning when applying machine learning models to resistance prediction problems in steel joint design.

Keywords: machine learning, numerical design calculations, component method, column web in transverse compression

POREĐENJE I PODEŠAVANJE MODELA MAŠINSKOG UČENJA ZA PANEL REBRA STUBA USLED POPREČNOG PRITISKA

Rezime: Metoda komponenti omogućava efikasno projektovanje čeličnih spojeva njihovim razlaganjem na pojedinačne komponente, koje se opisuju pomoću otpornosti i krutosti. U ovom radu predstavljen je pristup zasnovan na mašinskom učenju za predviđanje otpornosti rebra stuba pri poprečnom pritisku, na osnovu numeričkog skupa podataka generisanog pomoću programa IDEA StatiCa Connection. Razmatrana su i upoređena tri modela mašinskog učenja: duboka neuronska mreža, Random Forest i XGBoost. Posebna pažnja posvećena je uticaju podešavanja modela na tačnost predviđanja, pri čemu je duboka neuronska mreža podešena pomoću ručne analize osetljivosti, dok su Random Forest i XGBoost podešeni primenom grid search metode. Rezultati ukazuju na razlike u ponašanju modela, njihovoj osetljivosti na podešavanje i ukupnoj tačnosti predviđanja. Studija naglašava značaj odgovarajućeg izbora i podešavanja modela pri primeni mašinskog učenja za predviđanje otpornosti u projektovanju čeličnih spojeva.

Ključne reči: mašinsko učenje, numerički proračun nosivosti, metoda komponenti, rebro stuba izložen poprečnom pritisku.

[1] Brno University of Technology, Institute of Steel and Timber Structures, Czech Republic, amina.hajdarevic@vutbr.cz
[2] Brno University of Technology, Institute of Steel and Timber Structures, Czech Republic
[3] University of Coimbra, ISISE, ARISE, Department of Civil Engineering, Portugal

1. INTRODUCTION

The design of steel joints increasingly relies on numerical approaches that extend beyond idealised code-based design expressions in EN 1993-1-8 [1]. Finite-element-based tools are now widely used for member and connection verification, reflecting the complexity of real joint behaviour and the dispersion often observed when simplified formulas are compared with experiments or high-fidelity simulations. In parallel, machine learning (ML) has been investigated as a complementary approach to traditional codified design methods for steel structures, particularly for resistance prediction problems [2,3].

Within this context, the component method remains an efficient and transparent framework for steel joint design. However, several components, including the column web panel in transverse compression, exhibit behavior that is difficult to capture fully using empirical parameters calibrated on limited experimental datasets and simplified boundary conditions [4,5].

In this study, a dataset generated using IDEA StatiCa Connection is used to train and evaluate machine learning models for resistance prediction of the column web panel in transverse compression. Although machine learning models can capture nonlinear relationships in numerically generated datasets, their performance depends strongly on model selection and hyperparameter tuning.

Therefore, this paper compares selected machine learning models and evaluates the influence of selected hyperparameter configurations on their predictive performance. Particular attention is given to prediction accuracy, bias, coefficient of variation, and the practical aspects of model tuning. A manual sensitivity-based tuning procedure is used to study key neural-network hyperparameters, while Random Forest and XGBoost models are tuned using grid search over selected hyperparameter ranges. The aim is to determine which of the investigated machine learning models provides the most accurate and consistent resistance predictions for the column web panel in transverse compression.

2. DATASET AND INPUT FEATURES

The numerical dataset was generated in IDEA StatiCa Connection using an automated Python-based workflow through the application REST API.

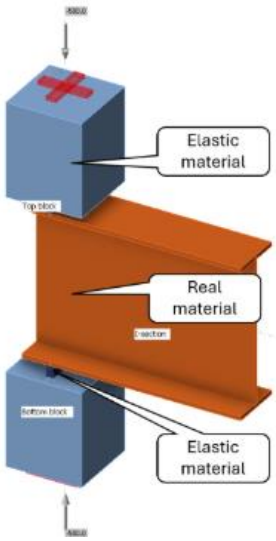


Figure 1: IDEA StatiCa parametrized model, description provided in [6]

Standard hot-rolled steel sections were used, including European IPE, HEA and HEB profiles, as well as American W sections.The inputs consist of section depth (D), flange width (B), flange thickness (t_f), web thickness (t_w), web–flange fillet radius (r_w), relative end distance ($x = \frac{e}{D}$), where (e) is the distance from the section end to the applied transverse load, relative bearing plate width ($a = \frac{t_p}{t_f}$), where (t_p) is the bearing plate thickness, limiting strain ε_{lim} and the yield strength (f_y). The final filtered dataset consisted of 9006 numerical models, and the target variable was the plastic resistance $F_{pl,Rd}$.

To describe the structure of the input space, a correlation matrix of the input and output variables was calculated, as shown in Figure 2.

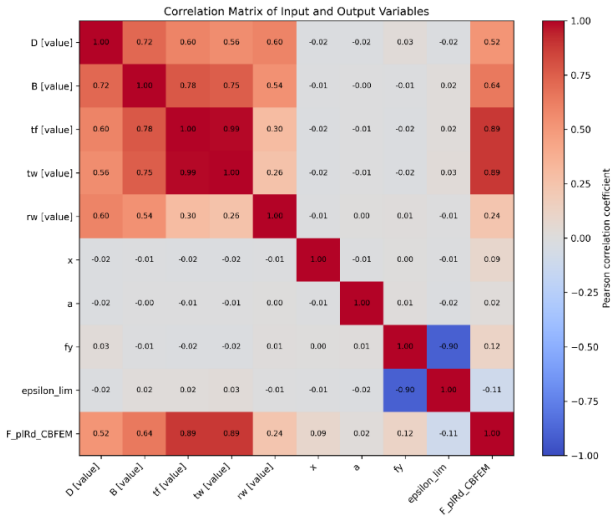


Figure 2: Correlation matrix of input and output features

The correlation matrix shows that the strongest linear relationship with the target resistance is observed for the flange thickness (t_f) and web thickness (t_w), both with a correlation coefficient of 0.89. At the same time, these two input variables are almost perfectly correlated with each other, with a correlation coefficient of 0.99. These were followed by the flange width (B), section depth (D), web–flange fillet radius (r_w), and yield strength (f_y), which also showed relevant, but lower, correlation with the target resistance.

3. MACHINE LEARNING MODELS

Three machine learning models were investigated and compared using the same dataset, input variables, target resistance, and evaluation metrics: Random Forest [7], and XGBoost [8] and a deep neural network (DNN) [9]. These models were selected because they represent different regression strategies commonly applied to structured engineering datasets. The DNN was chosen for its ability to learn complex patterns directly from the data. Random Forest was selected as a bagging-based ensemble method, while XGBoost was selected as a boosting-based ensemble method.

To ensure a consistent final comparison, the filtered dataset was divided into a training pool and an independent test set using an 80:20 split. Five-fold cross-validation (CV) was applied within the training pool for hyperparameter tuning and model selection. The CoV of the prediction-to-reference resistance ratio was adopted as the primary selection criterion, while (R^2) and bias were used as supporting performance metrics. After selecting the final hyperparameter

configuration, each model was retrained using the full training pool and evaluated on the independent test set.

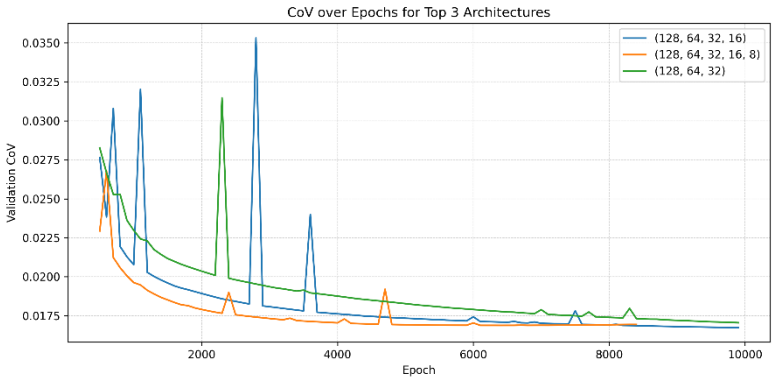
3.1. MANUAL SENSITIVITY STUDY OF THE DNN

For the deep neural network, a manual sensitivity study was carried out in order to select a suitable configuration before the final model comparison. This preliminary tuning stage was performed using a fixed 70:15:15 split into training, validation, and test subsets.

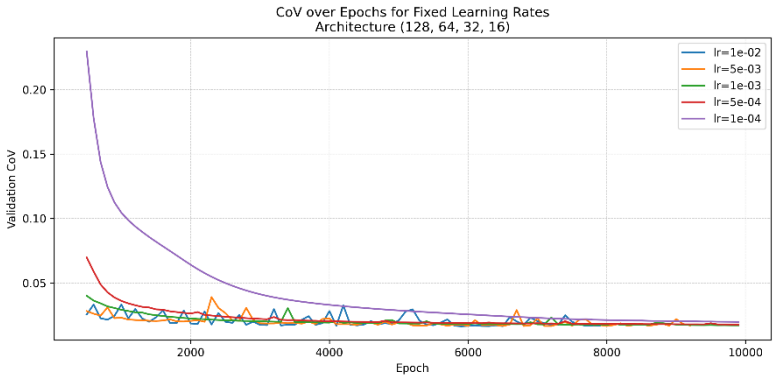
During each sensitivity study, one group of hyperparameters was varied while the remaining settings were kept fixed. The validation coefficient of variation of the prediction-to-reference resistance ratio was used as the main criterion for selecting the most suitable configuration.

The DNN models were implemented using the PyTorch deep learning framework [10]. Several network architectures were investigated, including architectures with different numbers of hidden layers and neurons. The investigated optimizers included SGD, SGD with momentum, Adagrad, RMSprop, Adam, and AdamW, while the tested activation functions included ReLU, LeakyReLU, ELU, GELU, SiLU, and tanh. Different weight decay values and learning rates were also examined to evaluate their influence on regularization, convergence speed, and the stability of the validation CoV during training.

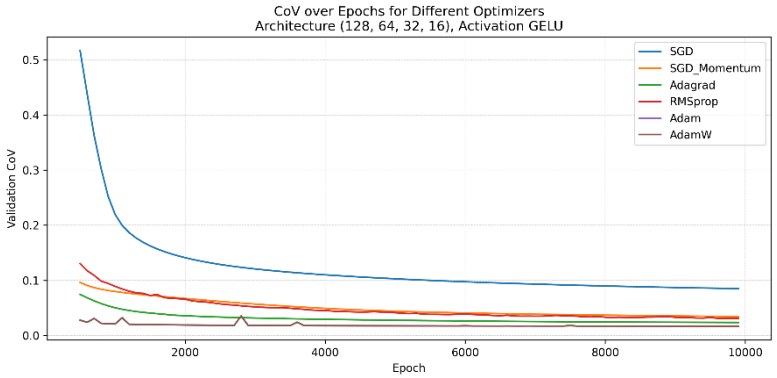
The results of the manual sensitivity study are shown in Figure 3.



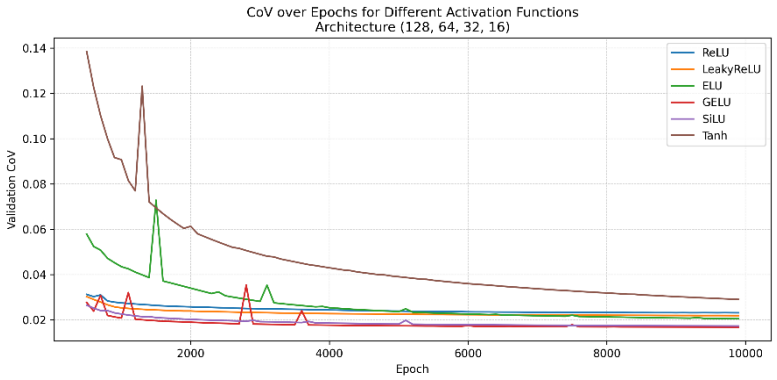
a)



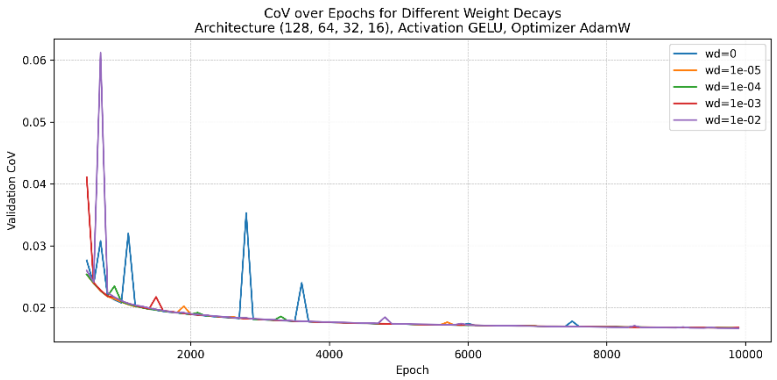
b)



c)



d)



e)

Figure 3: Manual sensitivity study of the neural network hyperparameters: influence of a) architecture, b) learning rate, c) optimizer, d) activation function, and e) weight decay on the validation CoV, over training epochs

The final selected DNN configuration consisted of the architecture (128, 64, 32, 16), GELU activation function, AdamW optimizer, learning rate (10^{-2}), and weight decay (10^{-2}). After this selection step, the DNN was retrained and evaluated using the same final evaluation procedure as the Random Forest and XGBoost models.

3.2. RANDOM FOREST

For the Random Forest model, a grid-based hyperparameter search was carried out, followed by additional checks around the best-performing parameter region until a performance plateau was observed. The tested values and the selected best configuration are summarized in Table 1.

Table 1: Hyperparameters of the Random Forest model search

Hyperparameter	Tested values	Best value
Number of estimators	100, 200, 400, 800, 1000, 2000	2000
Maximum tree depth	None, 10, 20, 30	None
Minimum samples for split	2, 5, 10	2
Minimum samples per leaf	1, 2, 4	1
Maximum number of features	sqrt, log2, all features (1.0)	1.0
Bootstrap sampling	True, False	True

3.3. XGBOOST

For the XGBoost model, a grid-based hyperparameter search was carried out, followed by additional checks around the best-performing parameter region until a performance plateau was observed. The tested values and the selected best configuration are summarized in Table 2.

Table 2: Hyperparameters of the XGBoost model search

Hyperparameter	Tested values	Best value
Number of estimators	800, 2000, 3000, 4000, 6000, 8000	8000
Maximum tree depth	3, 5, 7, 9	7
Learning rate	0.01, 0.02, 0.03, 0.05, 0.10	0.01
Subsample ratio	0.8, 1.0	0.8
Column sample ratio	0.8, 1.0	0.8
Minimum child weight	1, 3	3
L1 regularization	0.0, 0.01	0.0
L2 regularization	1.0, 5.0	1.0

4. RESULTS AND DISCUSSION

The final comparison of the selected models is summarized in Table 3. Among the investigated models, Random Forest showed the highest prediction scatter, with a mean CV CoV of 0.0654 and a test CoV of 0.0617. XGBoost improved the prediction accuracy, reducing the mean CV CoV and test CoV to 0.0295 and 0.0258, respectively. The best performance was obtained by the selected DNN configuration, which achieved the lowest mean CV CoV of 0.0179 and the lowest test CoV of 0.0184.

The corresponding test (R^2) values were 0.9933, 0.9981, and 0.9995 for Random Forest, XGBoost, and DNN, respectively. These values indicate that all three models captured the general relationship between the input parameters and the reference resistance with high accuracy. In addition, all models produced bias values close to 1.0, showing that no strong systematic overestimation or underestimation was observed. However, clear differences were found in the scatter of the prediction-to-reference resistance ratio. The DNN showed the lowest dispersion, followed by XGBoost, while Random Forest produced the largest scatter.

Table 3: Comparison of the three selected models' performance

Model	Mean CV Bias	Mean CV CoV	Std CV CoV	Mean CV (R^2)	Test Bias	Test CoV	Test (R^2)
Random Forest	1.0103	0.0654	0.0023	0.9908	1.0077	0.0617	0.9933
XGBoost	1.0040	0.0295	0.0018	0.9969	1.0034	0.0258	0.9981
DNN	0.9973	0.0179	0.0006	0.9992	0.9934	0.0184	0.9995

The predicted-to-reference resistance plots are shown in Figure 4. The plots confirm the differences in prediction scatter observed in Table 3. Random Forest shows the widest scatter, particularly at higher resistance values. XGBoost produces a more compact distribution around the ideal prediction line, while the DNN gives the narrowest scatter and the closest overall agreement with the reference values.

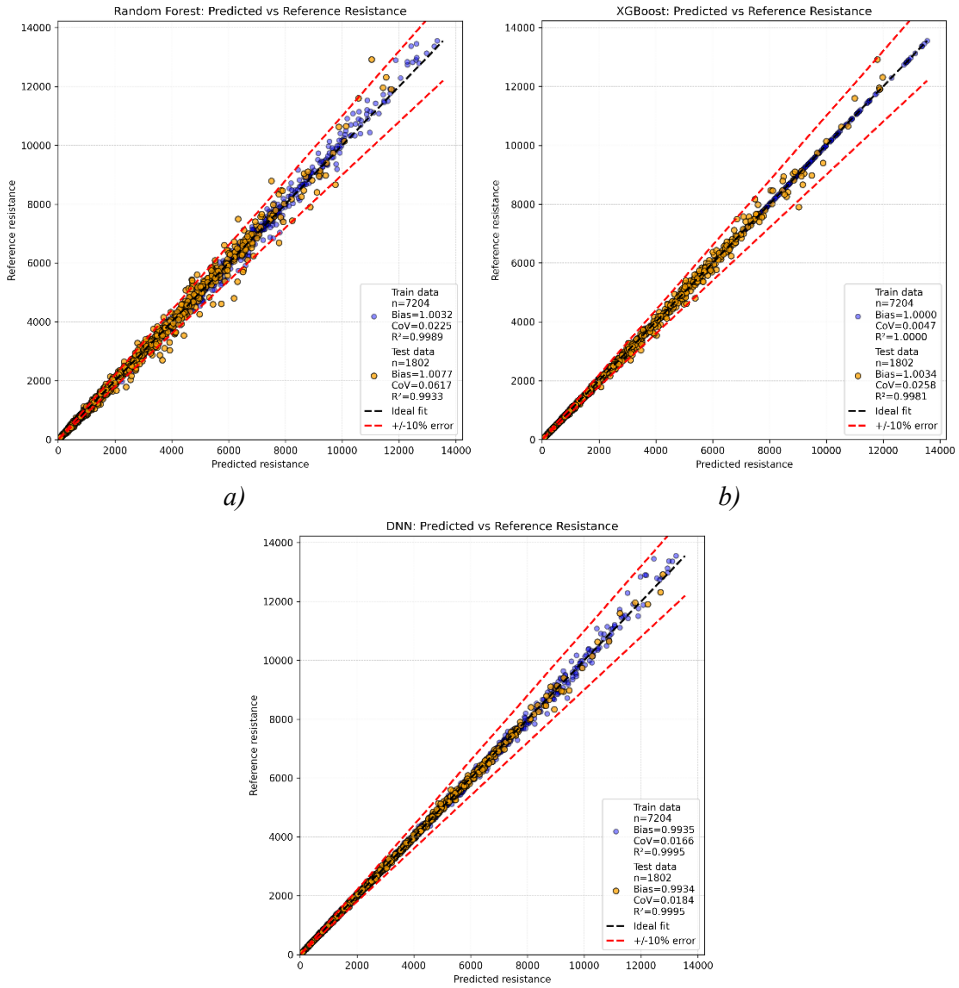


Figure 4: Predicted versus reference resistance for the selected models: a) Random Forest; b) XGBoost; c) DNN

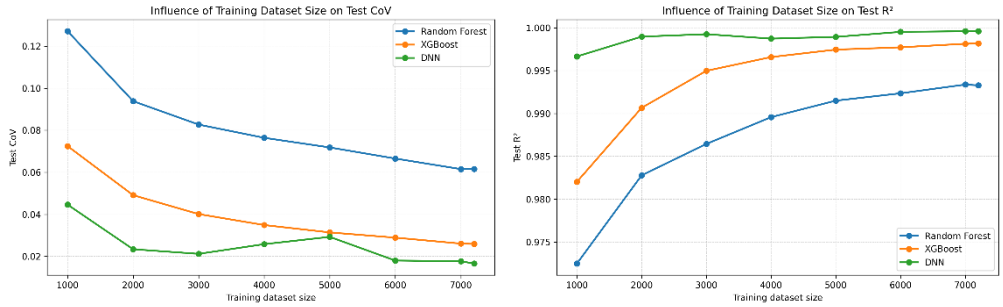
In addition to the main performance metrics, the ratio between test CoV and training CoV was evaluated for the final selected models as an additional indicator of possible overfitting. The minimum and maximum values of the prediction-to-reference resistance ratio (F_{pred}/F_{ref}) were also calculated on the independent test set to identify possible outlier predictions. These results are summarized in Table 4.

Table 4: Generalization and outlier diagnostics for the selected models

Model	Train CoV	Test CoV	Test/Train CoV	Test ratio range (F_{pred}/F_{ref})
Random Forest	0.0225	0.0617	2.75	0.8039–1.3655
XGBoost	0.0047	0.0258	5.44	0.9130–1.1574
DNN	0.0166	0.0184	1.11	0.9090–1.0766

The generalization analysis shows that Random Forest had the largest test scatter and the widest test ratio range, indicating the largest extreme deviations among the investigated models. XGBoost achieved a low test CoV and a relatively narrow test ratio range, although its test/train CoV ratio was higher due to the very small training CoV. In contrast, the DNN showed only a small increase from training CoV to test CoV and had the narrowest overall test ratio range. This indicates good generalization and no strong evidence of overfitting for the selected DNN configuration.

Since the generation of numerical data is computationally demanding, an additional dataset-size sensitivity analysis was performed.



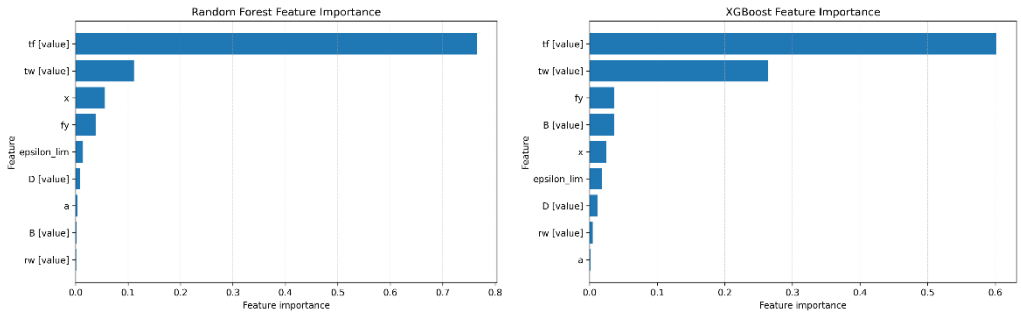
a) *b)*

Figure 5: Influence of training dataset size on model performance: a) test CoV; b) test (R^2)

As shown in Figure 5, the test CoV generally decreased as the training dataset size increased. Random Forest showed the highest scatter for all investigated training sizes, while XGBoost achieved substantially lower CoV values. However, the DNN consistently provided the lowest test CoV, even for the smallest subset of 1000 samples. DNN requires the least amount of samples. Adding more than 3000 samples does not significantly increase the performance of DNN anymore.

The corresponding (R^2) values increased with training dataset size for all models. The DNN maintained the highest (R^2) values over the full investigated range, followed by XGBoost and Random Forest. These results confirm that the selected DNN configuration remained the most accurate model even when reduced training datasets were used.

As a supplementary analysis, the built-in feature-importance measures of the selected tree-based models were evaluated, as shown in Figure 6. Since Random Forest and XGBoost are based on ensembles of decision trees, they provide built-in measures that estimate the relative importance of input variables from the trained model structure [7,8]. In this study, these values were used only to examine the behaviour of the tree-based models and to support the discussion of the input-variable relationships.



a) *b)*

Figure 6: Built-in feature importance of the selected tree-based models: a) Random Forest; b) XGBoost

The feature-importance results reflect the behaviour of the trained tree-based models within the investigated dataset. As shown in Figure 6, both Random Forest and XGBoost assigned the highest importance to the flange thickness (t_f), followed by the web thickness (t_w). This is consistent with the correlation matrix shown in Figure 2, where both thickness parameters exhibited the strongest linear correlation with the target resistance. However, some differences were observed for the remaining variables. In the Random Forest model, the next most relevant parameters were the relative end distance (x) and yield strength (f_y), while in XGBoost, the next most relevant parameters were (f_y) and flange width (B).

Overall, the results suggest that section thickness parameters contain a large part of the predictive information used by the tree-based models. In addition, the high importance assigned to (t_f) can be explained by the role of the column flange in load spreading: as the load is transferred through the flange, the flange thickness affects the effective width of the column web resisting transverse compression.

5. CONCLUSIONS

This paper compared a deep neural network, Random Forest, and XGBoost for predicting the resistance of the column web panel in transverse compression using 9,006 numerical models generated in IDEA StatiCa Connection. All models were evaluated using the same input variables, 80:20 training–test split, five-fold cross-validation procedure, and performance metrics, with the coefficient of variation of the predicted-to-reference resistance ratio adopted as the primary model-selection criterion.

All three selected models achieved high predictive accuracy, with test (R^2) values of 0.9933, 0.9981, and 0.9995 for Random Forest, XGBoost, and the DNN, respectively. However, substantially greater differences were observed in the prediction scatter. Random Forest obtained a test CoV of 0.0617, XGBoost reduced it to 0.0258, and the DNN achieved the lowest value of 0.0184. Thus, compared with Random Forest, XGBoost reduced the test CoV by approximately 58%, while the DNN reduced it by approximately 70%. The DNN also achieved a test CoV approximately 29% lower than that of XGBoost. The same ranking was obtained during cross-validation, where the mean CoV values were 0.0654, 0.0295, and 0.0179, respectively. All test bias values remained close to unity, ranging from 0.9934 to 1.0077, indicating no substantial systematic overestimation or underestimation.

The generalization analysis further supported the selection of the DNN. Its CoV increased only from 0.0166 on the training set to 0.0184 on the test set. It also produced the narrowest test prediction-ratio range, 0.9090–1.0766, compared with 0.9130–1.1574 for XGBoost and 0.8039–1.3655 for Random Forest. These results indicate that the selected DNN provided the most accurate and consistent predictions without strong evidence of overfitting.

The dataset-size sensitivity analysis showed that the DNN retained the lowest test CoV and the highest test (R^2) for all investigated training-set sizes, including the smallest subset of 1,000 samples. Its performance improved as the dataset increased to approximately 3,000 samples, after which the additional improvement was limited. This indicates that, for the investigated parameter space, the selected DNN can achieve high predictive performance using a substantially reduced portion of the complete numerical dataset.

The correlation and feature-importance analyses identified flange thickness (t_f) and web thickness (t_w) as the dominant input variables, both having a correlation coefficient of 0.89 with the target resistance. Their importance is consistent with the mechanical role of the flange and web in transferring and resisting transverse compression. However, because (t_f) and (t_w) were highly correlated with each other, their individual importance values should be interpreted with caution.

Overall, among the three investigated and tuned models, the DNN provided the best balance of accuracy, low prediction scatter, and generalization. Future work will extend the numerical dataset, validate the models against experimental results, and use a systematic ablation study to determine whether the number of input variables can be reduced without a relevant loss of predictive accuracy.

ACKNOWLEDGEMENT

The financial support of project FAST-S-26-9017 is gratefully acknowledged.

REFERENCES

[1] CEN. EN 1993-1-8: Eurocode 3: Design of steel structures, Part 1-8: Design of joints, European Committee for Standardization, Brussels, 2005.

[2] Thai H.-T. Machine learning for structural engineering: A state-of-the-art review, Structures, 38, 2022, 448–491. (no change needed)

[3] Ljubinković F., Carvalho A., Freire F., Conde J., Simões da Silva L. Towards stochastic characterization of welded beam-to-column steel joints using artificial neural networks, Structural Engineering International, 35 (4), 2025, 617–633.

[4] Conde J., Freire F., Ljubinković F., Vild M., Simões da Silva L. Welded beam-to-column steel joints: Assessment of European design rules, Thin-Walled Structures, 199, 2024, 111844.

[5] Balázs I., Pešek O., Horáček M., Vild M. Numerical and experimental verification of column web in transverse compression, Thin-Walled Structures, 203, 2024, 112267.

[6] Hajdarević A., Vild M., Müller A., Çakır T. Machine learning for column web panel in transverse compression component, Eurosteel 2026, Kraków, Poland, 2026, in print.

[7] Breiman L. Random forests, Machine Learning, 45 (1), 2001, 5–32.

[8] Chen T., Guestrin C. XGBoost: A scalable tree boosting system, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, 785–794.

[9] LeCun Y., Bengio Y., Hinton G. Deep learning, Nature, 521 (7553), 2015, 436–444.

[10] Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., et al. PyTorch: An imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems, 32, 2019.