Rakenteiden Mekaniikka (Journal of Structural Mechanics)
Vol. 57, No. 2, 2024, pp. 43–64
http://rakenteidenmekaniikka.journal.fi
https://doi.org/10.23998/rm.144743

VERTAISARVIOITU
KOLLEGIALT GRANSKAD
PEER-REVIEWED
www.tsv.fi/tunnus

Evaluation of machine learning techniques for capacity prediction of cold-formed steel beams subjected to bending

Ayman R. Hamdallah[1], Antti H. Niemi and Ahmad Abdullah

Summary Stiffened Cold-Formed Steel (CFS) sections often exhibit intricate nonlinear behaviors attributable to factors such as flexure effects and excessive slenderness. Traditional design methodologies, including the direct stiffness method, may inadequately capture these subtleties, potentially resulting in conservative or suboptimal designs. This study aimed to evaluate the performance of various machine learning algorithms, including simple and ensemble models, to predict the bending capacity of stiffened and unstiffened cold-formed beams in pure bending. A parametric study was conducted based on verified finite element analysis, and the machine learning algorithms were utilized to develop a unified capacity prediction method. The performance of six classical machine learning algorithms and four ensemble models were compared. The findings demonstrate that ensemble models, including AdaBoost, Gradient Boosting, Random Forest, and Extra Trees, outperform simple machine learning models in predicting the bending capacity of CFS beams. Moreover, introducing the stacking ensemble technique, using six different base models selectively, resulted in better performance than the individual baseline models. The approach addressed the nonlinearity pattern in the dataset caused by the flexure effect and excessive slenderness. The study suggests that adopting the proposed numerical and machine learning techniques could be a reliable method for predicting the structural behaviour and conducting cost-effective design of CFS beams, compared to the traditional analytical methods.

Keywords: cold-formed section, stiffeners, finite element analyses, machine learning, bending

Received: 5 April 2024. Accepted: 31 July 2024. Published online: 9 August 2024.

Introduction

Cold-Formed Steel (CFS) cross sections are used effectively and extensively in the construction industry and many structural applications as secondary load-carrying elements, such as roof purlins sections and transmission line towers. Most commonly, CFS cross sections are C-sections and Z-sections. These sections are manufactured by

[1]Corresponding author: ayman.hamdallah@oulu.fi

bending flat sheets with a thickness range from 0.4mm to 6.4mm, according to Eurocode 3 [1] and North America's typical thickness ranges [2]. CFS has attractive advantages such as ease of installation and prefabrication, high strength-to-weight ratios, and high structural efficiency. Furthermore, the inherent versatility of the manufacturing process facilitates the creation of diverse geometries, thereby offering significant potential for the optimization of CFS sections to align with specific structural objectives. This adaptability holds promise for enhancing manufacturing practices and realizing optimized structural designs.

CFS sections typically consist of plate elements with a significant width-to-thickness ratio. As a result, local buckling and distortional buckling are the primary failure modes for cold-formed steel members. These geometric failure modes hinder the efficient utilization of material, which can be resisted with numerous techniques. In plate mechanics, incorporating stiffeners can enhance the section's strength by providing out-of-plane support to the flat plate elements. Recently, the structural behavior of CFS stiffened sections has been investigated. For instance, complex edge stiffener, simple lips, perpendicular or inclined to flanges profile, have been used to improve the structural behavior of the channel-section columns against the expected buckling failure modes [3-5], and Z-purlins under combined bending and shear [6]. Columns with built-up sections have also been enhanced with intermediate stiffeners [7, 8]. Furthermore, efforts have been made to study the effectiveness of introducing edge or compression flange stiffeners [9-12], and intermediate web stiffeners [13] to C- and Z-sections subjected to bending.

There are two analytical methods adopted in the specifications to design the CFS sections [1, 14]: the Direct Strength Method (DSM) and the Effective Width Method (EWM). The former method is easy to apply if the studied sections are within the geometric boundaries specified in the North American standard [14]. However, in case new section geometry is studied, the semi-empirical strength curves used to determine the ultimate strength have to be calibrated against the test results of such sections [13, 15]. The second approach could be reliable but tedious, as an effective width of the stiffened element must be calculated iteratively, especially for complex stiffened sections that design codes have not adequately covered.

Machine Learning Methods (MLMs) have recently been employed to overcome such calculation difficulties and speed up the design process of CFS sections. The applications included the loading capacity prediction of beams experiencing local buckling [16, 17], distortional buckling [18], lateral-torsional buckling [19, 20], yielding [21], designing concrete-filled cold-formed steel columns [22], and the development of moment-rotation curves of semi-rigid joints using linear genetic programming [23]. Classical machine learning methods were also used for capacity prediction of CFS compression members, such as circular hollow sections [24-26], built-up sections [27], and for prediction of the axial load capacity of cold-formed lipped channels [28]. In some cases where CFS sections are subjected to combined axial-bending straining actions, a deep learning procedure was used to evaluate the beams' structural performance [29] and to predict the axial capacity of CFS channel [30]. Usually, data collected from the literature is used. However, in studying new problems, validated numerical data may be considered a cost-effective alternative [12, 31, 32].

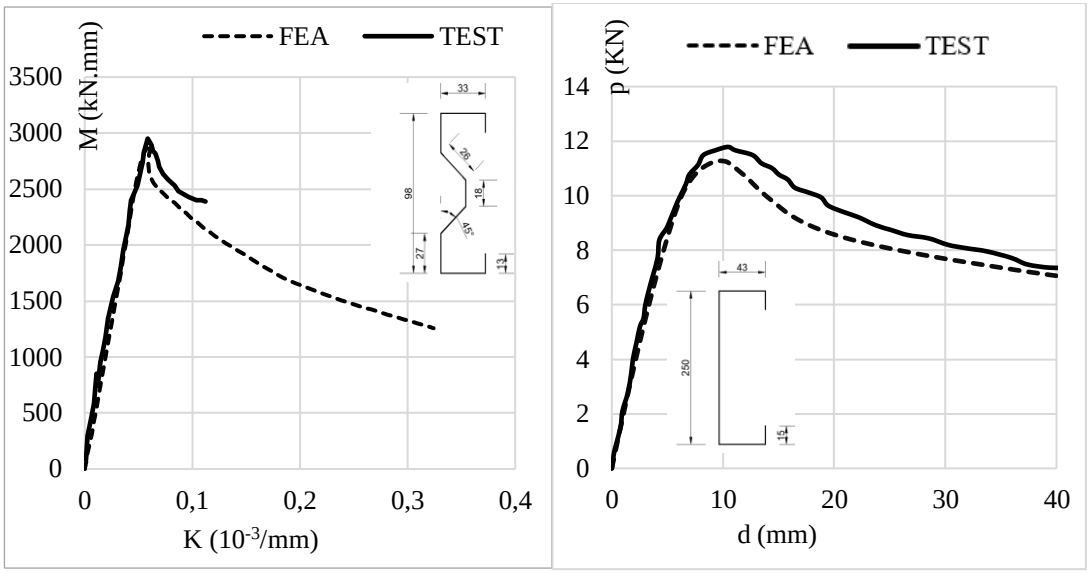
So far, the implementation of machine learning in designing CFS sections subjected to bending has not been comprehensively addressed. A challenging issue with this technique is comparing MLMs with varying complexity levels caused by the geometric parameters affecting the mechanical behavior of CFS sections to identify a viable approach for predicting bending strength based on geometric parameter data points [33, 34]. This paper evaluates ten different machine learning algorithms for designing stiffened and unstiffened C-channel beams subjected to flexural loads. A three-dimensional finite element model [35] was developed to identify the most significant parameters influencing the behavior of stiffened C-channel beams. These parameters were subsequently utilized as inputs for machine learning algorithms. A dataset comprising 140 points, generated using Central Composite Design (CCD), was employed to select the most critical factors for maximizing bending capacity. The best-performing techniques were then selected for a stacking approach to predict the bending capacity of CFS channels. The developed numerical and machine learning methods provide a systematic approach for estimating the capacity of stiffened CFS channels subjected to flexural loads. Moreover, it can contribute to more efficient and cost-effective construction projects by reducing material waste and overestimation in steel constructions while meeting long-term design criteria.

Dataset

Finite element results compare reliably to experimental testing in structural behavior investigations [31, 36-39]. In our study, a nonlinear Finite Element Analysis (FEA) was used to record the response of stiffened and unstiffened beam sections subjected to flexural loading by utilizing the finite element package ABAQUS [35]. Different geometric parameters were investigated, including the size and position of the proposed V-shape stiffener based on CCD, which is a popular experimental design method used in Response Surface Methodology (RSM) to build a quadratic model for the response variable without needing to test all possible combinations of factors. A total of 140 data points were generated for training the machine learning models using an elasto-plastic finite element method validated against 32 CFS specimens of test data taken from the existing literature [13, 40]. The comprehensive framework of the finite element model had been previously verified and validated, incorporating loading conditions, material modeling, and comparison with experimental data [41].

Figure 1(a) shows a moment-curvature curve of the CFS-stiffened channel beam (C-1-B4) tested by Wang and Young [13] under bending about its major x axis. The figure also shows one of the verified test results of Laim et al. [40] (Figure 1(b)). The test was to evaluate and study the flexural behavior of different CFS cross section profiles. As shown in Figure 1, the model was able to follow the flexural behavior of tests up to failure with acceptable accuracy. Thus, it could be concluded that the model could reliably simulate the structural response of beams in terms of ultimate load and failure mode, ensuring reliable results for further parametric studies. Notably, the structural analysis was carried out based on the center line dimensions and the metallic base thickness of the cross sections. The meticulous measurements conducted on the cross-sectional

dimensions of all specimens and the reported material properties ascertained through tensile coupon tests mitigated the impact of geometric imperfections [13, 40].



(a) (b)

Figure 1: Comparison of (a) FEA moment-curvature curve with the experimental results [13] (b) FEA load-displacement curve with the experimental results [40]

Parameters

As a result of this study, the moment capacity of a channel beam is represented by M_{ult}, which represents the ultimate pure bending moment at the midspan of the test specimens. The test specimens were cut to a specified length of 1400 mm, resulting in a final clear test span of 1260 mm. This relatively short span was chosen to investigate the local and distortional buckling of the stiffened sections. The design parameters for moment capacity and web stiffening of CFS beams were determined, with a total of five parameters: beam height h, flange width b, thickness t, the angle of the inclined part measured from the vertical axis θ and the inclined length of the stiffener w, see Figure 2. Values of those factors were constrained by AISI specification limitations [2] and summarized in Table 1. The ultimate moment capacity M_{ult} was then determined numerically for the developed models, according to the loading and boundary condition of the test beam (C-1-B4) [13], to train the MLMs.

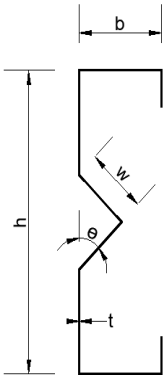


Figure 2: Schematic diagram of stiffened CFS channel section

Table 1: Variables of machine learning algorithm and their limits

Variable	Units	Min.	Max.
b	mm	6	128
h	mm	60	320
t	mm	0.4	2.4
$\boldsymbol{\theta}$	degree	15	60
w	mm	3	80

Pearson correlation analysis is a statistical technique used to quantify the strength and direction of the linear relationship between two continuous variables. To find the correlations and dependencies between the different design parameters, the Pearson correlation coefficient was calculated. It varies between –1 and 1 and is defined as

$$\rho_{ij} = \frac{n(\sum x_i y_i) - (\sum x_i)(\sum y_i)}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}, \quad (1)$$

where x_i and y_i are individual sample points indexed with i and n is sample size. Figure 3 shows that the variables are not highly correlated, except for b and h with Pearson's correlation coefficients of 0.67. It is notable that b/h value ranges from 0.1 to 0.4 to ensure the applicability of these sections and to facilitate the comparison of the effects of utilizing the stiffener at a constant cross-sectional length.

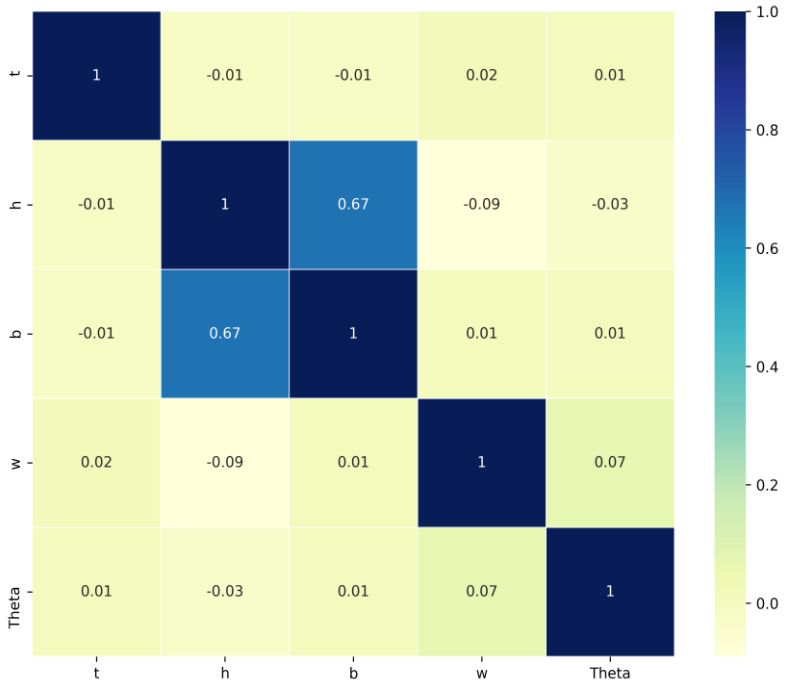


Figure 3: Pearson correlation matrix

Feature importance study

When dealing with input data containing different features, identifying each feature's importance can significantly contribute to improving the performance of MLMs. Moreover, it can better understand how each feature contributes to overall performance prediction. The feature importance is calculated according to the performance deterioration caused by the removal of each feature. In the context of Random Forests, which are ensembles of decision trees, Mean Decrease in Impurity (MDI) is a measure of feature importance calculated as the average (mean) decrease in impurity across all trees in the forest resulting from splits on that feature [42]. Based on the random forest regression, a sequence of capacity predictions is performed, each excluding one feature. Figure 4 shows the feature importance based on the mean and standard deviation (black line) of the impurity decrease related to each variable. The results indicate that plate thickness and the flange width are major contributors to the model's predictive accuracy. The thickness is prominently important due to its contribution to the overall stiffness of the section, While the flange width outperforms the section's height due to the absence of a stiffener in the flange. Including the stiffener on the web would divide its height into short and stiff parts, diminishing its effect. Other variables also contribute, but their impact is relatively similar and less pronounced.

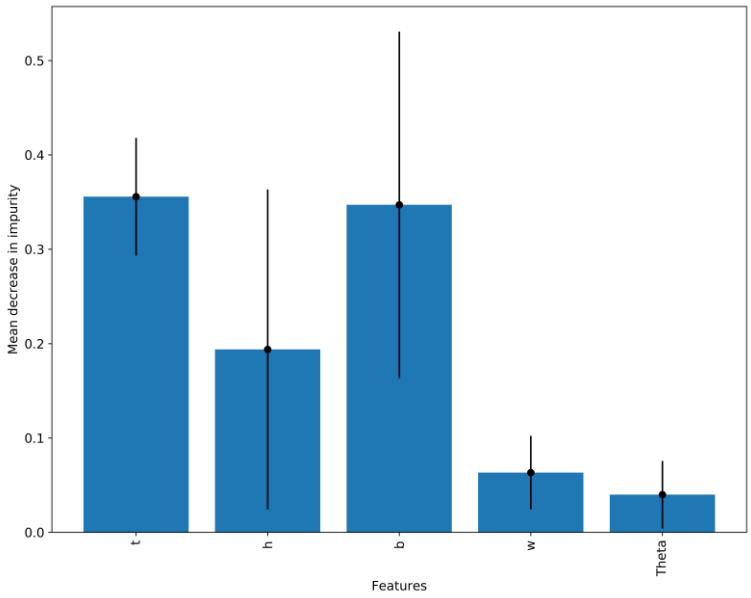


Figure 4: Feature importances using MDI

Machine learning methods

In this section, MLMs used to predict the bending capacity of CFS channels subject to pure bending are introduced. This is a regression task, where the correlations between the bending capacity (response or dependent variable) and some related features like geometric properties (predictors or independent variables) are first learned from the training set and then applied to make inferences for new samples (testing set). The ten most effective models are then selected to perform the bending prediction.

Linear Regression

Linear Regression (LR) is perhaps the simplest approach for modeling the relationship between a dependent variable y and the independent variables $x_0, x_1,\ldots, x_n$. The objective of linear regression is to find the hyperplane that best fits the data points and can be used to predict the values of the dependent variable for new observations as

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n, \tag{2}$$

with some coefficients $w_0, w_1,\ldots, w_n$. The ordinary least squares method is a simple and easy-to-implement LR algorithm. Still, it has some limitations, such as overfitting and multicollinearity. In cases where there are many independent variables, multicollinearity can lead to instability in the estimates of the regression coefficients. Regularization methods such as Lasso and Elastic Net can overcome these limitations by adding a penalty term to the loss function to control the coefficients' size. These methods can help prevent overfitting and improve the model's generalization performance. The following sub-sections will discuss these regularization methods in more detail [42].

Lasso

Least Absolute Shrinkage and Selection Operator (Lasso) is a regularization technique used in linear regression to address the issue of overfitting by adding a penalty term to the loss function. The penalty term, the absolute value of the sum of the coefficients multiplied by a hyperparameter alpha, encourages the coefficients of less important features to be shrunk towards zero, resulting in a sparse model where only the most important features are retained. Lasso has been shown to be effective in improving the model's generalization performance and reducing the estimates' variance, see [43] for more details.

Elastic-Net

Elastic-Net (EN) is a regularization technique that combines Lasso and so-called Ridge regression. It overcomes the limitations of each method by adding a penalty term to the loss function that includes both the L_1 and L_2 norms of the coefficients. The L_1 norm helps to achieve sparsity by shrinking some of the coefficients to zero, while the L_2 norm shrinks the remaining coefficients towards zero. The ratio between the two penalty terms can be tuned using a hyperparameter, allowing the user to control the trade-off between Lasso and Ridge regression [44].

K-Nearest Neighbors Regression

K-Nearest Neighbors regression (KNN) is a non-parametric algorithm for regression problems. Unlike LR, KNN regression makes no assumptions about the underlying data distribution. Instead, it uses the K nearest data points in the training set to predict the value of a new data point. The predicted value averages the K nearest neighbors' target values. The number of neighbors, K, is a hyperparameter that needs to be selected by the user [45].

KNN regression has several advantages, such as capturing nonlinear relationships and being easy to understand and implement. However, it also has some drawbacks, such as being sensitive to outliers and being inefficient when working with large datasets. Additionally, selecting the appropriate value for K can be challenging, and it may require some trial and error to find the optimal value for a given problem.

Decision Trees Regression

CART, which stands for Classification and Regression Trees, is a machine learning algorithm that can be used for regression tasks. It works by creating a decision tree based on the training data, where each internal node of the tree represents a decision based on a certain feature, and each leaf node represents a predicted value for the target variable. The algorithm splits the data at each node based on the feature that provides the most information gain, which is determined using a criterion such as mean squared error [46].

Once the decision tree is constructed, predictions for new data points are made by traversing down the tree based on the values of the features until a leaf node is reached, and the predicted value at that node is returned as the prediction for the new data point. CART can be useful for identifying nonlinear relationships between the features and target variables, as it can capture complex interactions between features. However, it can

be prone to overfitting, especially if the tree is allowed to grow too deep. Regularization techniques such as pruning or limiting the tree's depth can help mitigate this issue.

Support Vector Regression

Support Vector Regression (SVR) is a powerful machine learning algorithm that is used for regression tasks. SVR is a popular choice for machine learning tasks because it can handle high-dimensional data and can be used to model complex relationships between variables. SVR is a type of support vector machine (SVM) algorithm that can be used to solve linear and nonlinear problems. In contrast to linear regression, which aims to find a hyperplane that fits the data as well as possible, SVR tries to find a nonlinear function that can best fit the data while minimizing the error. The goal of SVR is to maximize the margin between the predicted values and the actual values. This algorithm uses a kernel function to map the input data into a high-dimensional feature space where finding a linear boundary can best separate the data points is possible. A kernel allows finding a hyperplane in a higher dimension space without increasing the computational cost.

Ensemble Methods

Ensemble Methods are machine learning techniques that combine multiple models to improve the overall performance and reduce the risk of overfitting. Ensemble methods are effective in a wide range of applications, including classification, regression, and clustering. Typically, ensemble methods are divided into two main categories:

- Averaging methods: such as Bagging and Random Forest, involve building multiple independent models and averaging their predictions. This technique reduces variance and results in a more stable and accurate model.
- Boosting methods: such as AdaBoost and Gradient Boosting, involve building models sequentially, where each subsequent model is built to correct the errors of the previous one. This technique reduces bias and results in a more robust and accurate model.

Forest of Randomized Trees

Two algorithms in the scikit-learn-ensemble module that use randomized decision trees are the Random Forest (RF) and Extra-Trees (ET) algorithms [42]. Both algorithms use perturb-and-combine techniques that are specifically designed for trees. This means a diverse set of regressors is created by introducing randomness in the regressor construction. The prediction of the ensemble is given as the averaged prediction of the individual regressors [46].

In RF, many decision trees are grown independently, each using a random subset of the training data and a random subset of the features. To make a prediction, the test data is run through each of the decision trees in the forest, and the final prediction is obtained by aggregating the individual predictions of all the trees. The most common way of aggregating the predictions is to take the average for regression problems of all the trees. Random Forest has several advantages over single decision trees, including better accuracy, robustness against overfitting, and the ability to handle high-dimensional data

with complex interactions between variables. It has been used in a variety of applications, including finance, healthcare, and remote sensing, among others.

ET is particularly useful for high-dimensional datasets, where the number of features is much larger than the number of samples. In extra randomized trees, the splitting points of the nodes are chosen randomly instead of being chosen based on the best split. This randomness leads to a reduction in the variance of the model, making it less sensitive to the noise in the data. The algorithm also uses a bagging technique, where each tree is trained on a randomly selected subset of the training data. The final prediction is made by averaging the predictions of all the trees in the forest.

Gradient Boosting Regression

Gradient Boosting regression (GBM) is a machine learning algorithm that sequentially builds an ensemble of decision trees. In this approach, each decision tree is built to correct the errors made by the previous one, thus minimizing the overall error. The algorithm starts with a single decision tree, and the subsequent trees are trained on the previous tree's residuals (the difference between the predicted and actual values). The final prediction is made by combining the predictions of all the trees in the ensemble.

GBM has become a popular algorithm for regression tasks due to its high accuracy and flexibility. It can handle various data types, including continuous and categorical variables, and perform feature selection and handle missing data. Additionally, it has built-in regularization techniques to prevent overfitting. However, it can be sensitive to hyperparameters, and tuning these parameters can be time-consuming.

AdaBoost (AB) is a boosting technique used for classification and regression problems. It is an iterative algorithm that focuses more on samples that the model previously misclassified. In AB, base estimators are built sequentially, and each subsequent estimator tries to correct the errors made by the previous one. The final prediction is made by taking a weighted average of the predictions from all the individual base estimators. The effectiveness of AB relies heavily on the choice of the base estimator and the learning rate. The learning rate determines how much each estimator contributes to the final prediction; a lower learning rate can lead to better results but with a larger number of estimators needed.

Stacking

Stacking is an ensemble learning technique combining multiple models to improve predictive performance. It involves training several base models and using their predictions as input features for a final model, often called a meta-model [42]. The stacking process involves several steps:

- Splitting the data: The dataset is divided into training and testing sets.
- Building base models: Several base models are trained on the training set.
- Generating predictions: The trained base models are used to make predictions on the test set.
- Building a meta-model: The predictions from the base models are used as input features for a meta-model, which is trained on the training set.
- Making final predictions: The meta-model is used to make final predictions on the test set.

One advantage of stacking over other ensemble methods is its flexibility in combining different types of models. So, linear and nonlinear models can be combined in a stacking ensemble to leverage the strengths of both models.

Results and discussion

This section provides a detailed comparison of ten different MLMs that serve as baseline models. The baseline models are categorized into two types: simple models and ensemble models. The discussion begins with the results of the simple models, followed by the findings of the ensemble methods. Lastly, the results of the stacking technique are presented.

The Root Mean Squared Error (RMSE) is used for evaluating the performance of the regression models. It measures the difference between the predicted and actual values to estimate how much the predictions of the model deviate from the true values on average and is defined as

$$RMSE = \sqrt{\frac{1}{n}\sum(\tilde{x}_i - x_i)^2}, \tag{3}$$

where n is the total number of test data, and $\tilde{x}_i$ and x_i are the predicted bending capacity coefficient and the test value, respectively. In the context of moment capacity prediction, a lower RMSE indicates that the model is more accurate in predicting the bending capacity of CFS channels.

Simple models

The performance of six MLMs using the RMSE and relative errors, as presented in Table 2 and Table 3, revealed distinct patterns in their predictive performance. Based on random splitting results, KNN and SVR emerged as the best performers among the models. KNN achieved the lowest test RMSE of 0.0992 and the lowest average relative error of 2.0895, with a maximum relative error of 23.3500, making it the most reliable model in terms of both consistency and accuracy. SVR also performed strongly, with a test RMSE of 0.0793, the second-lowest among all models, an average relative error of 6.5373, and a maximum relative error of 72.1928, balancing generalizability and predictive accuracy. In contrast, Lasso and EN were the worst performers. Lasso exhibited a test RMSE of 0.2571, a maximum relative error of 272.4345, and an average relative error of 20.0468, while EN had a test RMSE of 0.2469 and a maximum relative error of 234.8743 and an average relative error of 17.9576, indicating their inefficacy for this dataset. LR demonstrated moderate performance with a test RMSE of 0.1420 but high relative errors, with a maximum relative error of 207.5478 and an average relative error of 14.7019, suggesting significant inaccuracies in some predictions. DTR displayed a notable degree of overfitting, with a very low train RMSE of 0.0213 but a significantly higher test RMSE of 0.2086. Despite this overfitting, DTR maintained low relative errors, with a maximum

of 8.5694 and an average of 0.8040, suggesting good performance on certain samples but a lack of generalizability.

Stratified shuffle split is a variation of the stratified cross validation technique, where the dataset is randomly shuffled before it is split into train and test sets. This technique ensures that the class proportions are maintained in both the training and testing sets. The results of the comparison between the same MLMs using stratified shuffle split show that the performance of some models could be improved. In contrast, others have deteriorated compared to the previous random split results. In this phase, the performance of KNN remained strong with a test RMSE of 0.1466 and an improved average relative error of 1.6521, albeit with a slight increase in RMSE compared to random splitting. DTR demonstrated notable improvement with a test RMSE of 0.1401 and a significantly lower average relative error of 1.6364, indicating better generalizability and accuracy than in the random split evaluation. LR showed a considerable increase in test RMSE from 0.1420 in the random split to 1.2836 in the stratified sampling, reflecting worse performance and poor generalization. Lasso and EN also performed poorly, with test RMSEs of 0.2629 and 0.2385 and high average relative errors of 8.5706 and 7.1050, respectively, underscoring their inefficacy for this dataset in both evaluations. SVR maintained competitive performance with a test RMSE of 0.1534 and an average relative error of 5.3519, showing consistent accuracy. Thus, the stratified shuffle split technique could improve the performance of some MLMs and help address class imbalance issues. However, the results may vary depending on the dataset's specific characteristics and the modeling approach.

The boxplot of RMSE values, shown in Figure 5, can provide a useful visual summary of the performance of different MLMs, helping to identify patterns and outliers within the data. Comparing the boxplots of RMSE values for different models, the models with the lowest median and the minor variability could be identified, indicating better overall performance.

Table 2: RMSE of train and test set of the simple models:

Model	RMSE			
	Random split		Stratified shuffle split	
	Train	Test	Train	Test
LR	0.1871	0.1420	0.1323	0.1915
LASSO	0.2999	0.2571	0.2381	0.2808
EN	0.2873	0.2469	0.2294	0.2735
KNN	0.1349	0.0992	0.0897	0.1745
DTR	0.0213	0.2086	0.0347	0.1320
SVR	0.1252	0.0793	0.0762	0.1646

Table 3: Relative errors for the simple models

Model	Random split		Stratified shuffle split	
	Max relative error	Avg relative error	Max relative error	Avg relative error
LR	207.5478	14.7019	33.8230	8.7587
LASSO	272.4345	20.0468	40.6717	8.5706
EN	234.8743	17.9576	28.4777	7.1050
KNN	30.2741	2.0895	23.3500	1.6521
DTR	8.5694	0.8040	20.6480	1.6364
SVR	72.1928	6.5373	32.2793	5.3519

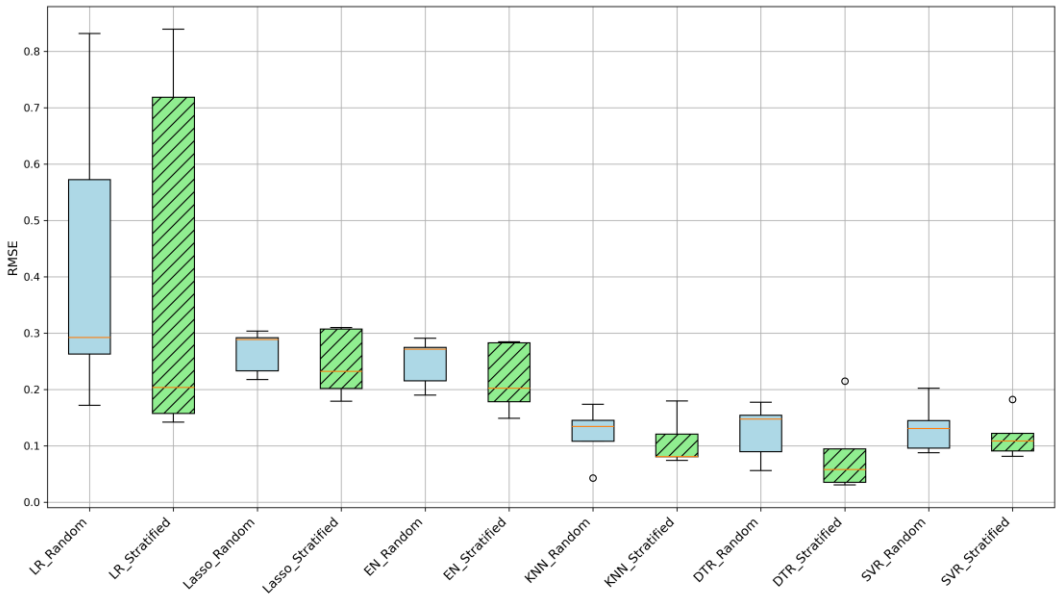


Figure 5: Boxplot of RMSE values for the simple models

Ensemble models

The ensemble models evaluated using stratified shuffle splitting demonstrated notable performance differences, as listed in Table 4. Among the ensemble models, GBM achieved the lowest Train RMSE of 0.0101, indicating high accuracy on the training set, but its Test RMSE increased to 0.1348, suggesting some degree of overfitting. Despite this, GBM maintained a relatively low Test Avg. Relative Error of 1.5376, highlighting its potential for better generalization. Similarly, ET, while perfectly fitting the training data with a Train RMSE of 0.0000, showed a Test RMSE of 0.1365. ET's Test Avg.

Relative Error was 1.6123, slightly higher than that of GBM, but it was still indicative of controlled errors on the test set.

In contrast, the RF model performed the worst among the evaluated models. RF had a Train RMSE of 0.0331 and a higher Test RMSE of 0.1396, indicating significant overfitting. Additionally, RF's Test Avg. Relative Error of 1.6984 was the highest among the models, suggesting less reliable performance on the test data. These results emphasize the superior balance in test set performance of GBM and ET compared to RF, underscoring the need for further model tuning and regularization to enhance robustness and generalization.

Table 4: Performance metrics for ensemble models using stratified shuffle splitting

Model	RMSE		Relative Errors	
	Train	Test	Max. Relative Error	Avg. Relative Error
AB	0.0465	0.1231	22.0745	2.9650
GBM	0.0101	0.1348	21.1912	1.5376
RF	0.0331	0.1396	22.9964	1.6984
ET	0	0.1365	20.6480	1.6123

Stacking

In machine learning, ensemble methods are a common approach to improving the predictive performance of models. In this context, the stacking of the dataset using six different base models was applied, performing more accurately than baseline models. From the previous findings, the stacking included LR, KNN, DTR, RF, and GBM. In addition, a multilayer perceptron (MLP) was added to enhance the stacking process. RF is then used as a higher model to make the final prediction. It should be mentioned that the stacking model was created using a combination of both linear and nonlinear models. The nonlinear models (MLP, RF, and GBM) contributed more to the improved performance than the linear models (LR and KNN).

In addressing the trade-off between simplicity and accuracy for the studied models, it is crucial to consider both the complexity of the model and its predictive performance. Simpler models, such as LR and KNN, offer transparency and ease of interpretation but may lack the capacity to capture intricate patterns in the data. In contrast, ensemble and stacking models, like RF, GBM, and MLP, are inherently more complex but can achieve higher accuracy by combining the strengths of multiple base learners. One effective strategy is to use model interpretation techniques to balance transparency and accuracy. For instance, feature importance metrics can be derived from ensemble models to understand the contribution of each feature to the predictions. Additionally, partial dependence plots can illustrate the relationship between key features and the target variable, providing insights into the model's behavior.

The results of the stacking approach show that it was able to improve the predictive performance of the individual base models. In Table 5, the RMSE of the stacked model is 0.057, much lower than any of the individual base models, including the RF model, which

had the lowest RMSE on the test set (0.080). This suggests combining different models has led to a more accurate overall prediction. Among the individual base models, the RF model had the lowest RMSE on both the training and test sets, indicating its effectiveness in predicting the target variable. The MLP and GBM models also performed well, with low RMSE on the test set. However, the LR and KNN models had comparatively higher RMSE on the test set, indicating their limitations in accurately predicting the target variable.

Table 5: RMSE of train and test set of the stacking process

Model	RMSE			
	Random Split		Stratified Shuffle Split	
	Train	Test	Train	Test
LR	0.184	0.142	0.176	0.155
MLP	0.162	0.117	0.153	0.126
KNN	0.164	0.120	0.134	0.133
DTR	0	0.132	0.0001	0.036
RF	0.062	0.080	0.056	0.053
GBM	0.089	0.093	0.040	0.067
Stacked	0.115	0.057	0.079	0.036

Interestingly, the DTR model achieved zero RMSE on the training set but did not contribute much to the stacked model's performance. This may be because the decision tree model overfitted the training data and did not generalize well to the test data.

The results of stratified splitting show that the stacking ensemble model outperformed all the individual models on the test set with an RMSE of 0.036, the lowest error among all the models. The individual models also performed well on the test set, with the MLP and random forest models having the lowest RMSE of 0.126 and 0.053, respectively. The DTR model achieved the lowest RMSE on the training set, which could indicate overfitting, as it had a high RMSE on the test set.

It's worth noting that the stacked model's performance on the training set was not as good as some of the individual models, such as the MLP and random forest models. This suggests that the stacked model combined the strengths of the individual models to generalize well to the test set rather than simply memorizing the training data. Overall, the stacking approach using the selected base models improved the models' predictive performance, with the stacked model achieving the lowest RMSE on the test set. This highlights the effectiveness of ensemble methods in improving the accuracy of predictions, mainly when using a combination of both linear and nonlinear models.

Comparison of FEA and machine learning prediction to DSM

Wang et al. [13] investigated the structural behavior and evaluated the appropriateness of the current direct strength method on the design of CFS stiffened cross sections subjected to bending. The nominal moment capacities (M_{DSM}) of CFS stiffened sections were calculated using the modified direct strength formulae [13]. The flexural strength (M_{DSM}) is the minimum of nominal flexural strength for local buckling (M_{nl}) and Flexural strength for distortional buckling (M_{nd}):

$$M_{\mathrm{DSM}} = \min(M_{\mathrm{nl}}, M_{\mathrm{nd}}). \tag{4}$$

The modified formulae for calculating the nominal flexural strength (M_{nl}) subjected to local buckling for sections symmetric about the axis of bending are as follows:

$$M_{\mathrm{nl}} = \left[1 + (\eta - 1)\left(1 - {}^{1}\!/_{C_{\mathrm{yl}}^{2}}\right)\right]M_{\mathrm{y}} \qquad \text{for } \lambda_{\mathrm{l}} \le 0.880, \text{ and} \tag{5}$$

$$M_{\mathrm{nl}} = \left[1 - 0.06\left(\frac{M_{\mathrm{crl}}}{M_{\mathrm{y}}}\right)^{0.26}\right]\left({}^{M_{\mathrm{crl}}}\!/_{M_{\mathrm{y}}}\right)^{0.26} M_{\mathrm{y}} \qquad \text{for } \lambda_{\mathrm{l}} > 0.880, \tag{6}$$

where $C_{\mathrm{yl}} = \sqrt{{}^{0.880}\!/_{\lambda_{\mathrm{l}}}} \le 3$, $\lambda_l = \sqrt{M_{\mathrm{y}}\,/M_{\mathrm{crl}}}$, η is the shape factor depends on the shape of the cross section ($\eta = {}^{Z_f}\!/_{S_f}$), Z_f is the plastic section modulus, and S_f is the gross section modulus referenced to the extreme fiber at first yield. The f_y is the yield stress, which is the 0.2% proof stress and M_{crl} is the critical elastic local buckling moment.

The modified formulae for calculating the nominal flexural strength (M_{nd}) subjected to distortional buckling for sections symmetric about the axis of bending are as follows:

$$M_{nd} = \left[1 + (\eta - 1)\left(1 - {}^{1}\!/_{C_{yd}^{2}}\right)\right]M_{y} \qquad \text{for } \lambda_{\mathrm{l}} \le 0.857, \text{ and} \tag{7}$$

$$M_{\mathrm{nd}} = \left[1 - 0.13\left(\frac{M_{\mathrm{crd}}}{M_{\mathrm{y}}}\right)^{0.54}\right]\left({}^{M_{\mathrm{crl}}}\!/_{M_{\mathrm{y}}}\right)^{0.54} M_{\mathrm{y}} \qquad \text{for } \lambda_{\mathrm{l}} > 0.857, \tag{8}$$

where $C_{\mathrm{yd}} = \sqrt{{}^{0.857}\!/_{\lambda_{\mathrm{d}}}} \le 3$, $\lambda_l = \sqrt{M_{\mathrm{y}}\,/M_{\mathrm{crd}}}$, and M_{crd} is the critical distortional buckling moment.

The comparison of FEM and the nominal values predicted by machine learning and the modified DSM is shown in Table 6. Different geometric ranges of CFS sections have been included in the comparison within the geometric limits of AISI [14] and EC3 [47].

Table 6: Comparison of the FEA results and machine learning predictions with DSM

Section (web height-flange width)	thickness (mm)	Stiffener Dim.		FEM (kNmm)	Failure mode	DSM* (kNmm)	Stacking prediction MLM (kNmm)	comparison	
		w (mm)	θ					$\frac{\text{DSM}}{\text{FEM}}$	$\frac{\text{MLM}}{\text{FEM}}$
120-30	3.6	33.94	45	14706	F	13888	15627	0.94	1.06
120-30	1.9	33.8	45	6690	L+F	6683	6567	1.00	0.98
150-52	2.4	41.58	45	14785	L+F	14156	14008	0.96	0.95
150-52	2.4	41.58	30	14633	L+F	14037	15144	0.96	1.03
150-52	2.4	21.21	45	13890	L+F	13915	14365	1.00	1.03
82.5-52	1.9	23.4	45	4666	L+F	4691	5366	1.01	1.15
82.5-52	1.9	12.66	45	4005	L+F	4631	5374	1.16	1.34
120-52	1.9	17.68	45	7333	L+F	7386	6366	1.01	0.87
120-30	1	33.94	45	2955	L+F	2583	2903	0.87	0.98
120-30	1	16.97	45	2498	L+F	2533	2413	1.01	0.97
82.5-52	1.5	35	45	3087	L+F	3363	3988	1.09	1.29
82.5-52	1.5	22.27	45	3413	L+F	3317	3560	0.97	1.04
82.5-52	1.5	35	30	3335	L+F	3294	4228	0.99	1.27
150-52	1.5	42.43	45	8627	L+F	7039	8723	0.82	1.01
82.5-52	1.5	11.66	45	2893	L+F	3265	3607	1.13	1.25
150-52	1.5	42.43	30	8254	L+F	6995	8163	0.85	0.99
150-52	1.5	21.21	45	6724	L+F	6925	6728	1.03	1.00
82.5-52	1	34.29	45	2026	L+F	1857	2063	0.92	1.02
82.5-52	1	34.29	30	1859	L+F	1819	2034	0.98	1.09
150-52	1	42.43	30	4534	L+F	3821	4180	0.84	0.92
120-52	1	16.97	45	2749	L+F	2842	2462	1.03	0.90
120-52	0.6	33.49	45	1447	L+F	1342	1710	0.93	1.18
120-52	1	33.94	45	3168	L+F	2886	2618	0.91	0.83
82.5-52	1	11.67	45	1969	L+F	1792	1968	0.91	1.00
150-52	1	42.43	45	4433	L+F	3838	4248	0.87	0.96
Average								0.97	1.04
Standard deviation								0.09	0.13

* Modified direct strength method

L=local buckling; D=distortional buckling; and F=global flexural buckling

The results show that machine learning was able to predict reasonable predictions for the selected CFS sections. However, it sometimes tends selectively to overestimate the bending capacity of thick lower height profiles, with maximum predictions reaching values of 29 % and 34 % in some cases, compared to the FEM predictions. The average ratio of the bending capacities predicted using MLM to the determined FEM values was 1.04, with a standard deviation of 0.13. So, machine learning could be considered, to a large extent, a reliable approach for predicting CFS sections' bending capacity incorporating complex stiffeners.

Conclusions

This study employed machine learning algorithms to create a unified capacity prediction method for stiffened and unstiffened CFS beams under pure bending. Verified finite element results have been used to feed the MLMs. The performance of the classical algorithms and ensemble techniques was then evaluated. A stacking technique was used with the selective algorithms to outperform the performance of individual models. The study suggests that the proposed numerical machine learning approaches could be reliable in predicting the structural behavior of CFS beams, as it predicted the complexities relevant to geometric parameters with acceptable accuracy. The following conclusions can be drawn:

- The performance of six different simple MLMs based on random splitting results was assessed. The results show that the KNN and SVR models performed the best, with the lowest test RMSE values of 0.1037 and 0.0787, respectively. Both models can handle nonlinear data relationships, which are present in the dataset. Additionally, these models can handle complex and high-dimensional data, which also exist in this study.
- A comparison was made among the performance of several MLMs using the stratified shuffle split technique and the previously used random split technique. The results indicated that the performance of some models was improved, while others deteriorated when stratified shuffle split was used. However, the KNN and DTR models still performed well. The results also showed that using a stratified shuffle split technique could improve the performance of some MLMs and help address class imbalance issues.
- The study's findings indicated that the ensemble MLMs, including AB, GBM, RF, and ET, could provide improved predictive performance compared to simple MLMs evaluated earlier. The ET model, in particular, perfectly fit the training data but had a higher test RMSE value than the RF model, which had slightly higher training RMSE. This suggests that the Random Forest model may be better suited for generalizing to new data.
- The applied stacking technique to the dataset used six base models to improve the approach performance. It combined both linear and nonlinear models. The nonlinear models (MLP, RF, and GBM) contributed more to the improved performance than the linear models (LR and KNN). In addition, the introduced MLP further enhanced the stacking technique. The results indicated that the

stacking approach significantly improved the performance of individual MLMs. The stacked model achieved an RMSE of 0.036, lower than any individual model. This suggests that the stacked model can capture the underlying patterns in the data, making more accurate predictions.

- Generally, the introduced machine learning approach tends to overestimate the ultimate capacity prediction of the thick lower height profiles of CFS beam sections. Conversely, the DSM slightly underestimates the capacity prediction of sections since it includes reduction factors for design purposes. The average ratio of the bending capacities predicted using ML to the FEM values was 1.04, with a standard deviation of 0.13. Such results point to the potential of ML as a reliable and cost-effective design approach for CFS sections.

References

[1] CEN. Eurocode 3: Design of steel structures, Part 1-3: General rules, supplementary rules for cold-formed members and sheeting, 2005: European Committee for Standardization, Brussels, .

[2] Hancock, G.J., T. Murray, and D.S. Ellifrit. *Cold-formed steel structures to the AISI specification*. 2001, New York: Marcel Dekker, Inc.

[3] Young, B. and J. Chen. Column tests of cold-formed steel non-symmetric lipped angle sections. *Journal of Constructional Steel Research*, 2008. 64(7):808-815.

[4] Young, B. and J. Yan. Design of cold-formed steel channel columns with complex edge stiffeners by direct strength method. *Journal of Structural Engineering*, 2004. 130(11):1756-1763.

[5] Zhang, Y., C. Wang, and Z. Zhang. Tests and finite element analysis of pin-ended channel columns with inclined simple edge stiffeners. *Journal of Constructional Steel Research*, 2007. 63(3):383-395.

[6] Pham, C.H., A.F. Davis, and B.R. Emmett. Numerical investigation of cold-formed lapped Z purlins under combined bending and shear. *Journal of Constructional Steel Research*, 2014. 95:116-125. https://doi.org/10.1016/j.jcsr.2013.11.015.

[7] Young, B. and J. Chen. Design of cold-formed steel built-up closed sections with intermediate stiffeners. *Journal of Structural Engineering*, 2008. 134(5):727-737. doi:10.1061/(ASCE)0733-9445(2008)134:5(727).

[8] Zhang, J.-H. and B. Young. Compression tests of cold-formed steel I-shaped open sections with edge and web stiffeners. *Thin-Walled Structures*, 2012. 52:1-11. https://doi.org/10.1016/j.tws.2011.11.006.

[9] Wang, H. and Y. Zhang. Experimental and numerical investigation on cold-formed steel C-section flexural members. *Journal of Constructional Steel Research*, 2009. 65(5):1225-1235.

[10] Haidarali, M.R. and D.A. Nethercot. Local and distortional buckling of cold-formed steel beams with both edge and intermediate stiffeners in their compression flanges. *Thin-Walled Structures*, 2012. 54:106-112.

[11] Li, H. and Q. Wen. Study on distortional buckling performance of cold-formed thin-walled steel flexural Members with stiffeners in the flange. *Thin-Walled Structures*, 2015. 95:161-169.

[12] JunYe, et al. Development of more efficient cold-formed steel channel sections in bending. *Thin-Walled Structures*, 2016. 101:1-13.
[13] Wang, L. and B. Young. Design of cold-formed steel channels with stiffened webs subjected to bending. *Thin-Walled Structures*, 2014. 85:81-92.
[14] AISI. North American Specification for the Design of Cold-Formed Steel Structural Members, in AISI S100, 2016: Washington, D.C.
[15] Schafer, B., A. Sarawit, and T. Peköz. Complex edge stiffeners for thin-walled members. *Journal of Structural Engineering*, 2006. 132(2):212-226.
[16] Gholizadeh, S., A. Pirmoz, and R. Attarnejad. Assessment of load carrying capacity of castellated steel beams by neural networks. *Journal of Constructional Steel Research*, 2011. 67:770-779.
[17] Fonseca, E.T., et al. A neuro-fuzzy evaluation of steel beams patch load behaviour. *Advances in Engineering Software*, 2008. 39(7):558-572. https://doi.org/10.1016/j.advengsoft.2007.07.005.
[18] Tohidi, S. and Y. Sharifi. Neural networks for inelastic distortional buckling capacity assessment of steel I-beams. *Thin-Walled Structures*, 2015. 94:359-371. https://doi.org/10.1016/j.tws.2015.04.023.
[19] Sharifi, Y., et al. Study of neural network models for the ultimate capacities of cellular steel beams. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 2020. 44(2):579-589. 10.1007/s40996-019-00281-z.
[20] Ferreira, F.P.V., et al. Lateral–torsional buckling resistance prediction model for steel cellular beams generated by Artificial Neural Networks (ANN). *Thin-Walled Structures*, 2022. 170:108592. https://doi.org/10.1016/j.tws.2021.108592.
[21] Gandomi, A.H., et al. A new prediction model for the load capacity of castellated steel beams. *Journal of Constructional Steel Research*, 2011. 67(7):1096-1105. https://doi.org/10.1016/j.jcsr.2011.01.014.
[22] Zarringol, M., H.-T. Thai, and M.Z. Naser. Application of machine learning models for designing CFCFST columns. *Journal of Constructional Steel Research*, 2021. 185:106856. https://doi.org/10.1016/j.jcsr.2021.106856.
[23] Gandomi, A.H., et al. Behavior appraisal of steel semi-rigid joints using Linear Genetic Programming. *Journal of Constructional Steel Research*, 2009. 65(8):1738-1750. https://doi.org/10.1016/j.jcsr.2009.04.010.
[24] Xu, Y., B. Zheng, and M. Zhang. Capacity prediction of cold-formed stainless steel tubular columns using machine learning methods. *Journal of Constructional Steel Research*, 2021. 182:106682. https://doi.org/10.1016/j.jcsr.2021.106682.
[25] Xu, Y., M. Zhang, and B. Zheng. Design of cold-formed stainless steel circular hollow section columns using machine learning methods. *Structures*, 2021. 33:2755-2770. https://doi.org/10.1016/j.istruc.2021.06.030.
[26] Rabi, M., et al. Prediction of the cross-sectional capacity of cold-formed CHS using numerical modelling and machine learning. *Results in Engineering*, 2023. 17:100902. https://doi.org/10.1016/j.rineng.2023.100902.
[27] Xiao, L., et al. Loading capacity prediction and optimization of cold-formed steel built-up section columns based on machine learning methods. *Thin-Walled Structures*, 2022. 180:109826. https://doi.org/10.1016/j.tws.2022.109826.

[28] Rajneesh, K., et al. Prediction of axial load capacity of cold formed lipped channel section using machine learning. *Structures*, 2023. 50:1429-1446. https://doi.org/10.1016/j.istruc.2023.02.102.
[29] Fang, Z., et al. Deep learning-based procedure for structural design of cold-formed steel channel sections with edge-stiffened and un-stiffened holes under axial compression. *Thin-Walled Structures*, 2021. 166:108076. https://doi.org/10.1016/j.tws.2021.108076.
[30] Fang, Z., et al. Deep learning-based axial capacity prediction for cold-formed steel channel sections using Deep Belief Network. *Structures*, 2021. 33:2792-2802. https://doi.org/10.1016/j.istruc.2021.05.096.
[31] Dai, Y., et al. A novel machine learning model to predict the moment capacity of cold-formed steel channel beams with edge-stiffened and un-stiffened web holes. *Journal of Building Engineering*, 2022. 53:104592. https://doi.org/10.1016/j.jobe.2022.104592.
[32] Leng, J., *Optimization techniques for structural design of cold-formed steel structures*, in *Recent Trends in Cold-Formed Steel Construction*. 2016, Elsevier. p. 129-151.
[33] Kumar, N. and D.R. Sahoo. Optimization of lip length and aspect ratio of thin channel sections under minor axes bending. *Thin-Walled Structures*, 2016. 100:158-169.
[34] Lee, J., et al. Optimum design of cold-formed steel channel beams using micro Genetic Algorithm. *Engineering Structures*, 2005. 27(1):17-24. https://doi.org/10.1016/j.engstruct.2004.08.008.
[35] Smith, M. ABAQUS/Standard User's Manual, Version 6.14. , 2014, Simulia.
[36] Dinis, P.B., D. Camotim, and N. Silvestre. FEM-based analysis of the local-plate/distortional mode interaction in cold-formed steel lipped channel columns. *Computers & Structures*, 2007. 85(19):1461-1474.
[37] Martínez-Martínez, J. and L. Xu. Simplified nonlinear finite element analysis of buildings with CFS shear wall panels. *Journal of Constructional Steel Research*, 2011. 67(4):565-575. https://doi.org/10.1016/j.jcsr.2010.12.005.
[38] Anbarasu, M. and M. Adil Dar. Axial capacity of CFS built-up columns comprising of lipped channels with spacers: Nonlinear response and design. *Engineering Structures*, 2020. 213:110559. https://doi.org/10.1016/j.engstruct.2020.110559.
[39] Dar, M.A., D.R. Sahoo, and A.K. Jain. Interaction between chord compactness and lacing slenderness in CFS built-up columns. *Structures*, 2021. 30:985-995. https://doi.org/10.1016/j.istruc.2021.01.065.
[40] Laím, L., J.P.C. Rodrigues, and L.S. da Silva. Experimental and numerical analysis on the structural behaviour of cold-formed steel beams. *Thin-walled structures*, 2013. 72:1-13.
[41] Hamdallah, A.R. and A. Abdullah. Numerical investigation of the efficient use of stiffeners in cold-formed channels subjected to bending. *JES. Journal of Engineering Sciences*, 2020. 48(No 3):383-395. 10.21608/jesaun.2020.107706.
[42] Pedregosa, F., et al. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 2011. 12:2825-2830.

[43] Friedman, J.H., T. Hastie, and R. Tibshirani. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 2010. 33(1):1 - 22. 10.18637/jss.v033.i01.
[44] Kim, S.J., et al. An interior-point method for large-scale L1-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing*, 2007. 1(4):606-617. 10.1109/JSTSP.2007.910971.
[45] Goldberger, J., et al. Neighbourhood Components Analysis. in *Advances in Neural Information Processing Systems 17*. 2004. NIPS.
[46] Hastie, T., R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Vol. 2, 2009, New York: Springer.
[47] CEN. Eurocode 3: Design of steel structures, Part 1-5: Plated structural elements, in European Committee for Standardization, Brussels, 2005.

Ayman R. Hamdallah, Antti H. Niemi
Civil Engineering Research Unit, University of Oulu
Oulu 90570, Finland
first name.last name@oulu.fi

Ahmed Abdullah
Department of Civil Engineering, Aswan University
Aswan 81542, Egypt
Civil Engineering Department, University of Bisha
Bisha 61922, Saudi Arabia
ahmed.abdalla@aswu.edu.eg