Data-driven modeling of cold-formed steel bolted angle connections under axial tension using Artificial Neural Networks

Ruan A. Carvalho[1], Francisco Evangelista Jr.[1]

[1]*Department of Civil and Environmental engineering, University of Brasilia*
Darcy Ribeiro,70910-900, Brasília - DF, Brazil
runcarvalho@gmail.com, fejr.unb@gmail.com

Abstract. L-shaped laminated steel elements, known as angles, are commonly bolted to other elements to make connections in steel structures and, when connected by only one of its legs and subjected to axial tension, their resistance efficiency is reduced due to complex phenomena, such as shear lag. Aiming to obtain better understanding of this problem, the present work uses data-driven modeling with Artificial Neural Networks (ANN) as an approach to the study of load capacity of cold-formed steel bolted angles connected by one leg and under axial tension. A dataset is built with samples from numerical and experimental works, collecting steel angles' geometric parameters, tensile strength of the material and ultimate capacity regarding net section failure. Different sets of input variables are used in the data-driven modeling, implementing feature selection and importance algorithms, while performing 5-fold cross-validation and hyperparameter tuning with Bayesian Optimization. The predictive accuracy of the Machine Learning (ML) models is compared with the one given by equations from European, Brazilian, Australasian and North American standards. The comparison shows much superior results in the ANN models, proving the competitiveness and effectiveness of ML techniques and data-driven modeling.

Keywords: Data-Driven Modeling, Bolted connections, Cold-formed steel, Machine Learning, Artificial Intelligence.

1 Introduction

Steel is one of the main structural materials in civil engineering, because of that it is important to understand its mechanical properties and failure mechanisms. Angles are L-shaped laminated steel elements that, when bolted and subjected to axial tension, may present net section failure, i.e., caused by rupture of the section where the holes are located. When these elements are connected by only one of its legs, there is influence of complex phenomena, such as shear lag, because of that, in steel structures standards across the world, a reduction factor is applied to the resistance of the net section. This factor and general formulae proposed by these standards have proved to be inaccurate, limited and inconsistent, as shown by Sarothi *et al.* [1] and Može [2].

Fleitas *et al.* [3] conducted a parametric study of the net section resistance in bolted cold-formed steel angles under tension and proposed two similar equations with different coefficients, for angles with one and two bolt lines. Sarothi *et al.* [1] used eleven Machine Learning (ML) techniques as an approach to studying the bearing capacity of double shear bolted connections and performed grid search tuning, an exhaustive and expensive technique, with 10-fold cross-validation to select optimal hyperparameters values. Xiao *et al.* [4] also used ML methods, but to study loading capacity prediction and optimization of cold-formed steel built-up section columns. The authors used Shapley Additive Explanations (SHAP) to investigate feature importance and dependency, with no feature selection methods to reduce input space dimensions.

The main objective of the current work is to use data-driven modeling with Artificial Neural Networks (ANN) as an approach to the study of net section resistance of cold-formed steel bolted angles connected by one leg and

under axial tension. The results are compared with the ones given by equations of standards from 4 regions of the world, aiming to verify the general effectiveness of ML techniques and data-driven modeling. As a secondary objective, different models were created using only the most important features, hoping to identify most relevant variables for predicting steel angles net section resistance and build cheaper but accurate enough models. A dataset of 314 instances was built collecting numerical and experimental data from five works, comprehending data of equal and unequal single steel angles with one and two lines of bolts. The data-driven models were trained with 80% of the data, tested with 20%, validated with 5-fold cross-validation and their hyperparameters tuned with the efficient Bayesian Optimization technique. The equations analyzed in this work are from European, Brazilian, Australasian and North American standards.

Initially, it is presented some background of the ML related procedures executed in the study, explaining fundamentals of ML regression, Multilayer Perceptron architecture, optimizers, regularization techniques and hyperparameters tuning. Then, methodology is explained, focusing on the collection and preprocessing of the built dataset and on the data-driven modeling. Results are shown and discussed, comparing performance metrics of the developed models with the ones from some standards, with a final graphical comparison between best performing model and best performing standard. At last, final conclusions of the work are gathered and presented.

2 Background

This section uses references of the literature of Goodfellow, Bengio and Courville [5]. Artificial Neural Networks (ANN) are computational models with structure inspired by animal brains, presenting neurons disposed in layers connected to each other. The Multilayer Perceptron (MLP) is a fully connected feedforward ANN, where there are at least three layers (Fig. 1), one input layer, one or more hidden layers and one output layer. These layers are densely connected, with all its neurons connected with all neurons of the next layer, opposing sparsely connected layers. In feedforward networks, the information goes through the model in one direction only, from input to output, with no cycle among the neurons. The output $\widehat{\boldsymbol{y}}$ of the input layer, i.e., the input of the first hidden layer, is given by

$$\widehat{\boldsymbol{y}} = \phi(\boldsymbol{X}\boldsymbol{W} + \boldsymbol{b}), \tag{1}$$

where $\boldsymbol{X}$ is a matrix with each row representing an instance and each column a feature, ϕ is the nonlinear activation function, $\boldsymbol{W}$ is the weight matrix and $\boldsymbol{b}$ the bias vector.

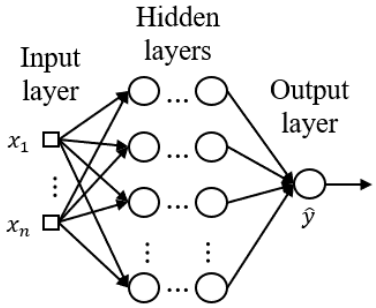

Figure 1. Multilayer Perceptron architecture

Every connection in the network has associated weight and bias, which form the parameters of the model, calculated with gradient-based optimizers in the training phase. These optimization algorithms refer to minimizing the loss function, usually Mean Squared Error (MSE) when approaching regression problems, by changing parameters' values in small steps in the opposite direction of gradient. The size of these steps is defined by a hyperparameter α called learning rate, that can be fixed or, in modern algorithms, adaptive. This process is only viable because of the backpropagation algorithm, introduced by Rumelhart, Hinton and Williams [6], that presents an efficient technique of passing information forward through the network to produce a scalar loss and then backward to compute inexpensively the gradients, applying recursively the chain rule of calculus.

Regarding optimization algorithms with adaptive learning rates, Adam (derived from Adaptive moment

estimation), introduced by Kingma and Ba [7], is a method for efficient stochastic optimization. It computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. The algorithm keeps track of an Exponentially Weighted Moving Averages (EWMA) of the gradient and an EWMA of the squared gradients. Reddi, Kale and Kumar [8] have shown that Adam does not always converge to the optimal solution and proposed AMSGrad, a variant of Adam with guaranteed convergence. The variant provides a new exponential moving average, taking the maximum of all second moment estimates until present time step and normalizing the running average of the gradient.

In validation phase, the objective is to adjust model's hyperparameters and parameters in order to guarantee good generalization capacity. The adjusts are done iteratively, based on model training, measurement of model's predictive accuracy on the validation set and new adjusts to the hyperparameters, until reaching good enough accuracy. A possible approach to the validation stage is the use of automatic hyperparameters tuning techniques, such as Bayesian Optimization (BO). The technique performs well on computationally expensive function because it creates a surrogate model of the objective function with, usually, Gaussian Processes, allowing for cheaper evaluations. The surrogate model is fit to a few initial random points, then each new point of evaluation is decided based on an acquisition function. BO performs the search for optimal hyperparameters values in a user-defined space and for a given number of iterations, or until convergence.

When data is limited, cross-validation techniques present much better use of data, as they allow the usage of subsets of the training set as validation set, instead of requiring a separated set exclusive to the validation task. In this context, k-fold cross-validation divides the data in a number k of folds, with one of them being selected as the validation set and the rest as training set, this process is repeated k times always selecting a different fold for validation.

Regularization techniques are methods that assist the development of models of great generalization capability, preventing overfitting, situation where there is excessive learning of the patterns in training set, capturing also noise and outliers, thus performing poorly on unseen data. Some of these techniques are:

- Early Stopping, consists in stopping model training when validation error is not improving after a given number of consecutive iterations, also called patience, thus preventing the model to perform excessive training;
- Weight Decay, modifying the loss function in training by adding a criterion that penalizes large weight values based on squared L^2 norm and on a hyperparameter λ that controls the strength of this preference for smaller weights;
- Batch Normalization, proposed by Ioffe and Szegedy [9], is a method of adaptive reparametrization that speeds up training and improves model performance. It normalizes the inputs of each layer to have mean 0 and variance 1, then, in order to maintain representation capacity of the model, it learns parameters γ and β in training to respectively scale and shift the result.

After training and validation, models go through test phase, where their final performance metrics are evaluated, predicting target variable values based on new unseen observations. In this final stage, with the fitted model, feature importance technique may be performed, like Permutation Importance (PI), in order to answer what features present the biggest impact on predictions. Based on Breiman [10] and Fisher, Rudin, and Dominici [11], PI algorithm randomly shuffles the instances of a single feature column and measures the difference between model error before and after the permutation, the process may be done n times for more robust results.

3 Methodology

All programming activities were executed by the authors of the present work in Python 3.12.0 and with assist of some libraries: matplotlib and seaborn for plotting and data visualization; pandas and NumPy for data manipulation and analysis; scikit-learn for inputs normalization, training-test and training-validation splitting and performance measurement; scikit-optimize for hyperparameters tuning and optimization; joblib for parallelization; and PyTorch for initialization, training, validation, testing and other activities related to the ANN modeling.

3.1 Data collection and preprocessing

The dataset used in the present work is composed of 314 samples and was collected from five studies, four

of which were experimental, composing 154 samples, and one numerical with 160 samples. The size of the sample collected from each work and corresponding authorship are listed in Tab. 1.

Table 1. Source and number of instances of sampled steel angles data

Authors	Sample size
LaBoube and Yu [12]	12
Yip and Cheng [13]	15
Paula *et al.* [14]	66
Teh and Gilbert [15]	61
Fleitas *et al.* [3]	160
TOTAL	314

Each sample presents ten features, nine of which were used as inputs to predict the target variable T_{exp}, the resistance of the steel angle net section. Input features consist of the ultimate tensile strength of the steel material (F_u), and eight geometric parameters of the steel angles: width of connected leg (b_c), width of not connected leg (b_d), thickness (t), number of bolt lines (n_lines), number of bolts per line (n_holes), length of connection (L) and two connection eccentricities, from shear plane to centroid of cross-section ($\bar{x}$) and from centroid of the connection to centroid of cross-section ($\bar{y}$). In order to stabilize and improve model training, min-max normalization was applied to each input feature, transforming minimum value to 0 and maximum to 1.

The 314 instances were divided in a 4:1 proportion, with 80% (251 instances) composing the training set and 20% (63 instances) the test set. K-fold cross-validation was done with $k = 5$ folds, so each iteration of the algorithm split the original training set in 80% for training (200 or 201 instances) and 20% for validation (51 or 50 instances), thus following same proportion of the training-test division.

3.2 Data-driven modeling

The data was fed through a feedforward fully connected Artificial Neural Network (ANN), also known as a Multilayer Perceptron (MLP) architecture. Every hidden layer of the MLP was built sequentially with a linear transformation (applying weights and biases), a batch normalization layer and a ReLU (Rectified Linear Unit) activation function. Training was performed with AMSGrad variant of the Adam optimizer, batch size equal to the number of instances in the training set (200 or 201) and for a maximum of 10,000 epochs. This high number of maximum epochs was used to guarantee convergence during training, while the implementation of early stopping with patience of 200 prevented overfitting, stopping training much before 10,000 epochs.

Bayesian Optimization (BO) was used to optimize hyperparameters listed in Tab. 2 with their respective search space. The technique was implemented with parallelization, where each CPU core performed an independent BO, starting in different random initial points and proceeding with the sequential algorithm for a given number n of calls, that was arbitrarily chosen as $n = 10$. Since the CPU used in the research has 8 cores, a total of 80 iterations were performed, where the desired optimal hyperparameters values were given by the iteration with the lowest objective function value. The objective function used was the mean of the Mean Squared Errors (MSE) of the 5 validation folds.

Table 2. Optimized hyperparameters and respective search space

Hyperparameter	Range
Learning rate (α)	$[10^{-4}, 10^{-1}]$
Weight decay (λ)	$[10^{-9}, 1]$
Hidden layers	[1, 4]
Neurons in 1st hidden layer	[10, 1000]
Neurons in 2nd hidden layer	[10, 1000]
Neurons in 3rd hidden layer	[10, 1000]
Neurons in 4th hidden layer	[10, 1000]

Parallelization was also implemented for model training and inference on GPU with PyTorch's support to CUDA, a parallel computing platform and programming model created by NVIDIA that allows for straightforward GPU-accelerated data-driven modeling. The GPU used is a NVIDIA GeForce MX110 with 2 GB of dedicated memory.

A baseline model, called model 1, was created with the original set of 9 inputs. Then, based on Permutation Importance (PI) of model 1 on test set, models 2 and 3 were created with respectively two and one thirds of the most important features from the 9 used, that is, 6 and 3 inputs. This feature selection step is performed aiming to obtain sufficiently accurate cheaper models, requiring the acquisition of less physical input variables data.

4 Results and discussion

After fitting model 1, Permutation Importance was implemented with $n = 1000$ and the mean and Standard Deviation (SD) of these permutations are shown in Tab. 3, ordered from most to least important variable. So, model 2 is composed of t, b_c, n_lines, L, F_u and $\bar{y}$ and model 3 of t, b_c and n_lines. Also, it can be said that steel angle thickness t and width of the connected leg b_c are clearly the most important features for net section resistance prediction.

Table 3. Mean and SD of increase in MSE after permutation

Input features	Mean	SD
t	590.88	2.88
b_c	530.50	3.12
n_lines	148.72	10.52
L	135.61	11.02
F_u	85.35	16.83
$\bar{y}$	48.19	22.54
b_d	40.95	38.28
$\bar{x}$	10.38	95.62
n_holes	6.90	96.01

To compare models 1 through 3, some regression evaluation metrics were calculated in training and testing stages: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), coefficient of determination R^2, Mean Absolute Percentage Error (MAPE) and Standard Deviation (SD) of Absolute Percentage Error (APE). In validation, the metrics analyzed were mean and SD of the MSE from the 5 folds. All the mentioned metrics results are shown in Tab. 4.

Table 4. Regression evaluation metrics of models 1 through 3 in validation, training and test

Performance metrics	Model 1	Model 2	Model 3
Mean MSE - Validation	31.78	32.59	163.39
SD MSE - Validation	12.55	7.59	38.25
MSE - Training	13.39	16.97	142.89
MSE - Test	28.60	38.40	135.00
RMSE - Training	3.66	4.12	11.95
RMSE - Test	5.35	6.20	11.62
R^2 - Training	0.98	0.98	0.83
R^2 - Test	0.97	0.96	0.85
MAPE - Training	2.52%	2.85%	10.20%
SD of APE - Training	2.67%	2.74%	8.48%
MAPE - Test	5.09%	5.48%	11.35%
SD of APE - Test	4.37%	4.25%	10.01%

As observed in Tab. 4, model 1, using the original set of 9 features, presents very good performance and great generalization capability, showing R^2 greater than 0.95 and MAPE of almost 5.10% in test set. Model 2, in comparison to model 1, provides worse results, with increases of 16% in RMSE and 34% in MSE on test set, but still provides high accuracy, with R^2 greater than 0.95 and MAPE of approximately 5.50%. It can be said that model 2 represents an interesting alternative of lower predictive accuracy but with 3 less required input features. On the other hand, model 3 shows much higher error measures, thus loosing too much predictive capability in exchange for its reduced number of inputs.

Model 1 predictions were then compared to the ones given by equations from: American Iron and Steel Institute (AISI) standards, AISI S100-16 [16]; Eurocode 3, EN 1993-1-1:2005 [17] and EN 1993-1-8:2005 [18]; Brazilian ABNT NBR 14762:2010 [19]; and Australasian AS/NZS 4600:2018 [20], AS 4100:2020 [21] and NZS 3404:Part 1:1997 [22]. The comparison was done through some performance metrics evaluated on test set, MSE, RMSE, R^2, MAPE and SD of APE, and its results are shown in Tab. 5.

Table 5. Regression evaluation metrics of model 1 and equations from different standards

Performance metrics	**Model 1**	**AISI**	**Eurocode 3**	**NBR**	**AS/NZS**
MSE - Test	28.60	340.40	1789.14	1488.52	1603.50
RMSE - Test	5.35	18.45	42.30	38.58	40.04
R^2 - Test	0.97	0.62	-1.01	-0.67	-0.80
MAPE - Test	5.09%	15.70%	39.93%	35.85%	33.40%
STD of APE - Test	4.37%	9.65%	21.01%	12.81%	32.01%

As observed in Tab. 5, the data-driven model is very superior to the equations given by all the analyzed standards, with much lower error metrics and higher coefficient of determination R^2. Since AISI presented the best performing equation among the standards, a graphical comparison between model 1 and AISI Standard was made (Fig. 2), plotting predictions of the target variable against the observed values of test set. The plots support that the Machine Learning model provides much more accurate predictions than all of the standards analyzed.

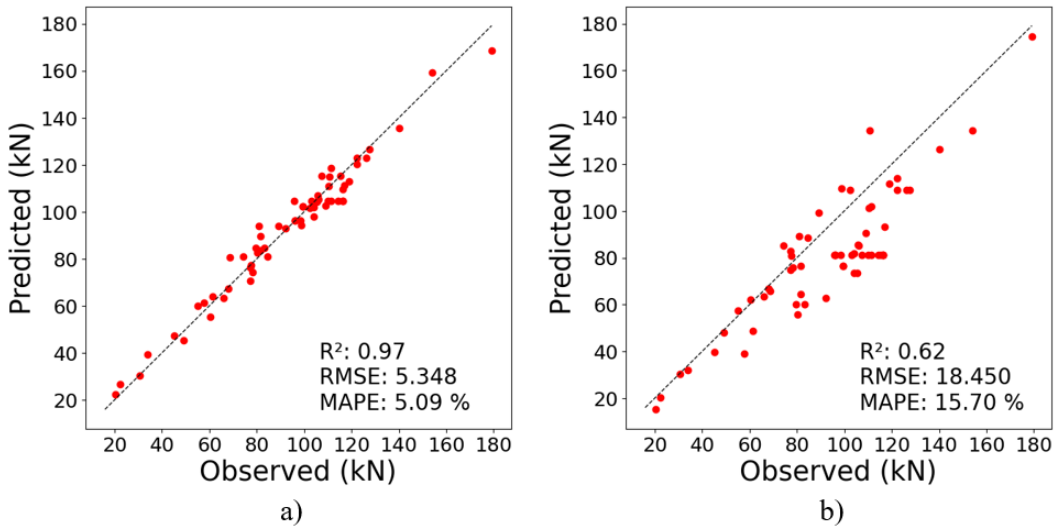

Figure 2. Predicted against observed values of test set in: a) Model 1; b) AISI Standard

5 Conclusions

In the present work, GPU-accelerated data-driven models were created with different sets of input features, executing feature selection based on permutation feature importance. The model created from the original set of 9 inputs provided greatly accurate predictions. Also, a slightly less accurate model was developed, providing an alternate cheaper model with two thirds of the inputs. When comparing the best performing model with standards

from 4 regions of the world, the ANN showed outstanding results, with much more accurate predictions.

In conclusion, Machine Learning techniques and data-driven modeling proved to be powerful tools for predicting net section resistance of cold-formed steel angles under axial tension, thus providing important knowledge and insight on the theme. These tools proved to be very efficient even when compared with some of the most known standards in use across the world.

Acknowledgements. The authors gratefully acknowledge the financial support granted by the Brazilian research agency CNPq (National Council for Scientific and Technological Development) and the infrastructure provided by the University of Brasília (UnB), specially by the Laboratory of Computational Methods and Artificial Intelligence (LAMCIA) located at PECC-UnB (Postgraduate Program in Structures and Civil Construction).

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] S. Z. Sarothi, K. S. Ahmed, N. I. Khan, A. Ahmed and M. L. Nehdi. "Predicting bearing capacity of double shear bolted connections using machine learning". *Engineering Structures*, vol. 251, 2022.
[2] P. Može. "Statistical evaluation of bearing resistance and related strength functions for bolted connections". *Journal of Constructional Steel Research*, vol. 171, 2020.
[3] I. Fleitas, J. Bonilla, L. M. Bezerra and E. Mirambell. "Net section resistance in bolted cold-formed steel angles under tension". *Journal of Constructional Steel Research*, vol. 167, 2020.
[4] L. Xiao, Q. Y. Li, H. Li and Q. Ren. "Loading capacity prediction and optimization of cold-formed steel built-up section columns based on machine learning methods". *Thin-Walled Structures*, vol. 180, 2022.
[5] I. Goodfellow, Y. Bengio and A. Courville. *Deep Learning*. Massachusetts: MIT Press, 2016.
[6] D. E. Rumelhart, G. E. Hinton and R. J. Williams. "Learning internal representations by error propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. Cambridge, MA, USA: MIT Press, pp. 318-362, 1986.
[7] D. P. Kingma and J. Ba. "Adam: A method for Stochastic Optimization". arXiv preprint arXiv:1412.6980, 2014.
[8] S. J. Reddi, S. Kale and S. Kumar. "On the convergence of Adam and beyond". arXiv preprint arXiv:1904.09237, 2019.
[9] S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*, pp. 448-456, 2015.
[10] L. Breiman. "Random Forests". *Machine Learning* 45, 3-32, 2001.
[11] A. Fisher, C. Rudin and F. Dominici. "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously". *Journal of Machine Learning Research*, vol. 20, n. 177, p. 1-81, 2019.
[12] R. A. LaBoube and W. W. Yu. "Tensile and Bearing Capacities of Bolted Connections". Final Summary Report, University of Missouri-Rolla, Rolla, Missouri, 1995.
[13] A. S. M. Yip and J. J. R. Cheng. "Shear Lag in Bolted Cold-Formed Steel Angles and Channels in Tension". Structural Engineering Report No. 233, University of Alberta, Edmonton, Canada, 2000.
[14] V. F. Paula, L. M. Bezerra and W. T. Matias. "Efficiency reduction due to shear lag on bolted cold-formed steel angles", *Journal of Constructional Steel Research*, vol. 64, n. 5, pp. 571-583, 2008.
[15] L. H. The and B. P. Gilbert. "Net section tension capacity of cold-reduced sheet steel angle braces bolted at one leg", *Journal of Structural Engineering*, vol. 139, n. 3, pp. 328-337, 2013.
[16] American Iron and Steel Institute (AISI), "AISI S100-16, North American Specification for the Design of Cold-Formed Steel Structural Members", AISI, 2016.
[17] European Committee for Standardization (CEN), "EN 1993-1-1:2005, Eurocode 3: Design of steel structures - Part 1-1: General rules and rules for buildings", CEN, 2005.
[18] European Committee for Standardization (CEN), "EN 1993-1-8:2005, Eurocode 3: Design of steel structures - Part 1-8: Design of joints", CEN, 2005.
[19] Brazilian National Standards Organization (ABNT), "ABNT NBR 14762:2010, Design of cold-formed steel structures", ABNT, 2010.
[20] Standards Australia/Standards New Zealand (SA/SNZ). "AS/NZS 4600:2018, Cold-formed steel structures." SA/SNZ, 2018.
[21] Standards Australia (SA). "AS 4100:2020, Steel structures." SA, 2020.
[22] Standards New Zealand (SNZ). "NZS 3404:Part 1:1997, Steel Structures Standard." SNZ, 1997.