Neural network-based analytical model to predict the shear strength of steel girders with a trapezoidal corrugated web

Miguel Abambres [1,1] and He J [2]

[1]Num3ros
[2]Affiliation not available

March 19, 2024

Abstract

Corrugated webs are used to increase the shear stability of steel webs of beam-like members and to eliminate the need of transverse stiffeners. Previously developed formulas for predicting the shear strength of trapezoidal corrugated steel webs, along with the corresponding theory, are summarized. An artificial neural network (ANN)-based model is proposed to estimate the shear strength of steel girders with a trapezoidal corrugated web, and under a concentrated load. 210 test results from previous published research were collected into a database according to relevant test specimen parameters in order to feed the simulated ANNs. Seven (geometrical and material) parameters were identified as input variables and the ultimate shear stress at failure was considered the output variable. The proposed ANN-based analytical model yielded maximum and mean relative errors of 0.0% for the 210 points from the database. Moreover, still based on those points, it was illustrated that the ANN-based model clearly outperforms the other existing analytical models, which yield mean errors larger than 13%.

Hosted file

`Software Bugs Found on March 2021.docx` available at https://authorea.com/users/594089/articles/675909-neural-network-based-analytical-model-to-predict-the-shear-strength-of-steel-girders-with-a-trapezoidal-corrugated-web

Neural network-based analytical model to predict the shear strength of steel girders with a trapezoidal corrugated web

Miguel Abambres [a], Jun He [b]

[a] *R&D, Abambres' Lab, 1600-275 Lisbon, Portugal*
abambres@netcabo.pt

[b] *School of Civil Eng, Changsha University of Science and Technology, 410114, Changsha, PR China*
hejun@csust.edu.cn

Abstract

Corrugated webs are used to increase the shear stability of steel webs of beam-like members and to eliminate the need of transverse stiffeners. Previously developed formulas for predicting the shear strength of trapezoidal corrugated steel webs, along with the corresponding theory, are summarized. An artificial neural network (ANN)-based model is proposed to estimate the shear strength of steel girders with a trapezoidal corrugated web, and under a concentrated load. 210 test results from previous published research were collected into a database according to relevant test specimen parameters in order to feed the simulated ANNs. Seven (geometrical and material) parameters were identified as input variables and the ultimate shear stress at failure was considered the output variable. The proposed ANN-based analytical model yielded maximum and mean relative errors of 0.0% for the 210 points from the database. Moreover, still based on those points, it was illustrated that the ANN-based model clearly outperforms the other existing analytical models, which yield mean errors larger than 13%.

Keywords: shear strength; corrugated webs; trapezoidal corrugation; steel girders; artificial neural networks; analytical model

1. Introduction

Corrugated steel plates without additional stiffeners are characterized by high shear buckling strength and out-of-plane flexural stiffness, having been widely used in structural engineering applications, such as large span roof, steel plate shear walls, and bridge girders (Wu et al. 2006, He et al. 2012, Emami et al. 2013, Jiang et al. 2015). The concept of replacing

1

flat webs with corrugated webs in bridge structures was first proposed in France at 1986, then successfully applied worldwide in the past 30 years (Combaut 1988, Kondo et al. 1994). Typically, the corrugations in the web are trapezoidal, but forms like sinusoidal, triangular, and rectangular have also been considered. Corrugations are used to increase the shear stability of webs in beam-like members, thus eliminating the need of transverse stiffeners. This paper focuses on the shear strength of steel webs with trapezoidal corrugations, a topic that has been extensively studied since the end of 20th century, covering both buckling and plasticity phenomena (e.g., Bergfelt & Leiva-Aravena, 1984, Leiva-Aravena & Edlund 1987, Johnson & Cafolla 1997, Metwally 1998, Sayed-Ahmed 2001, 2007, Abbas 2003). Shimada (1965) was the first to study the shear strength of steel girders made of folded-plate webs. Easley and McFarland (1969) proposed the global shear buckling equation of corrugated webs by treating them as flat and orthotropic. Lindner & Aschinger (1988) carried out experimental tests to assess the shear strength of trapezoidal corrugated steel webs, and suggested using 70% of the elastic shear buckling stress as the nominal strength for design purposes. Luo & Edlund (1994, 1996) analysed the buckling of trapezoidal corrugated panels under in-plane loading by spline finite strip and finite element (FE) methods. The influence on the elastic buckling load of various parameters, such as geometry, loading patterns and boundary conditions, was assessed. Elgaaly et al. (1996) presented experimental and analytical results for steel beams with trapezoidal corrugated webs loaded predominantly in shear, and proposed bucking formulas based on (i) the local deformation of the corrugation folds modelled as isotropic flat plates, or (ii) the global deformation of the entire web panel modelled as an orthotropic plate. Metwally

2

(1998) investigated the behavior of steel girders with trapezoidal corrugated webs, and proposed a formula for predicting their nominal shear strength. Yamazaki (2001) proposed formulae for the computation of the buckling strength of corrugated webs, based on results from 6 full-scale models of steel bridge girder webs. Driver et al. (2006) tested full-scale corrugated web girders made of HPS 485W steel, assessed the effect of web initial geometric imperfections through measurements of the out-of-plane displacements, and proposed a lower bound design equation that accounts for both local and global buckling of the web in the elastic and inelastic domains. Watanabe et al. (2007) presented test results for the shear capacity of steel girders using four different trapezoidal corrugation shapes. Yi et al. (2008) studied the nature of the interactive shear buckling of corrugated webs, and concluded that the first order interactive shear buckling equation not accounting for material inelasticity provides a good estimation of the shear strength of corrugated steel webs by comparison with 15 tests and finite element analysis (FEA) results. Moon et al. (2009) presented 3 test results, described the shear strength formula previously presented by Yi et al. (2008), and compared the proposed formula and several other formulas with results from 17 tests. Sause & Braxtan (2011) summarized previously developed formulas for predicting the shear strength of trapezoidal corrugated steel webs, along with the corresponding theory, and proposed a novel formula. Nie et al. (2013) conducted an experimental and analytical study to investigate the shear strength of trapezoidal corrugated steel webs. The experimental program involved testing eight H-shaped steel girders with trapezoidal webs. Hassanein and Kharoob (2013a, b) carried out geometrically and materially nonlinear imperfect analyses (GMNIA) of full-scale bridge girders with corrugated

3

steel webs failing by shear, and found that when the ratio of flange's thickness to corrugated web's thickness was greater than 3, the boundary conditions between flange and web were approximately fixed; a formula for computing the interactive shear buckling of corrugated webs under fixed boundary conditions was proposed. Leblouba et al. (2017a, b) conducted laboratory tests on a series of corrugated steel web beams to investigate their shear behaviors; three typical failure modes were observed and the failure mechanisms on the post-buckling phase were assessed; besides, five analytical models for the estimation of the critical shear buckling stress based on FE analysis results were proposed and validated against test data.

Despite all the research done on the shear behavior of steel members with corrugated webs, design codes or guidelines, with the Eurocode (2005) and JSCE (1998) as exceptions, are still lacking specifications for this type of structural elements (Papangelis et al. 2017). Moreover, there are still many uncertainties and discrepancies associated with test data and proposed models due to many factors, including geometric imperfections of the web, material properties, shear bucking modes, and inconsistencies between the assumed test conditions and the theoretical models. In order to effectively (accurately and efficiently) estimate the shear capacity of corrugated web steel girders, this paper proposes the use of artificial neural networks (also referred in this manuscript as ANN or neural nets), a popular machine learning technique. Machine learning, one of the six disciplines of Artificial Intelligence (AI) without which the task of having machines acting humanly could not be accomplished, allows us to 'teach' computers how to perform tasks by providing examples of how they should be done (Hertzmann and Fleet 2012). When there is abundant data (also called examples or patterns)

explaining a certain phenomenon, but its theory richness is poor, machine learning can be a perfect tool; as such, its application to the problem of shear in corrugated steel web is suitable and timely. The artificial neural network is the (i) oldest (McCulloch and Pitts 1943) and (ii) most powerful (Hern 2016) technique of machine learning. ANNs also lead the number of practical applications, virtually covering any field of knowledge (Wilamowski and Irwin 2011, Prieto et al. 2016). In its most general form, an ANN is a mathematical model designed to perform a particular task, based in the way the human brain processes information, i.e. with the help of its processing units (the neurons). ANNs have been employed to perform several types of real-world basic tasks. Concerning functional approximation, ANN-based solutions are frequently more accurate than those provided by traditional approaches, such as multi-variate nonlinear regression, besides not requiring a good knowledge of the function shape being modelled (Flood 2008). The proposed ANN was designed based on 210 experimental results available to date in the literature (see section 2). The focus of this study was not to understand the mechanics underlying the shear behavior of corrugated steel webs, but parametric studies by means of accurate and robust ANN-based models make it possible to evaluate and improve existing mechanical models.

.

2. Data Gathering

Many shear strength tests of I-shaped beams and girders with trapezoidal corrugated webs have been conducted. The 210-point dataset (available in Developer 2018a) used to feed the

ANN software employed in this work, was assembled from the following experimental results:

Lindner & Aschinge (1988), Elgaaly et al. (1996), Peil (1998), Yamazaki (2001), Sause et al.

(2003), Abbas (2003), Sause and Clarke (2003), Gil et al. (2005), Watanabe et al. (2007), Moon

et al. (2009), Sause & Braxtan (2011), He (2011), Nie et al.(2013), Leblouba et al. (2017 a,b).

Seven independent variables were adopted as inputs in ANN simulations, as described and

illustrated in Table 1 and Fig. 1, respectively. In Fig.1(a), a simply supported girder with

corrugated steel web (height h_w) is subjected to a concentrated load (Q) distanced a (shear span)

from the left support, until (web) shear failure occurs. Fig.1(b) depicts details of the corrugated

web mid surface (top-view of a single "wavelength"), namely the widths of parallel and

inclined folds (b and c, respectively), the projected width of the inclined fold (d), the

corrugation depth (h_r), and the thickness of the corrugated plate (t_w). The maximum shear stress

(assumed uniform over web's height) when failure occurs (τ_e) is the target/output (dependent)

variable considered in all assessed ANNs. Target values were either reported by the authors of

the experimental tests or calculated as $\tau_e = V_e / (h_w t_w)$, being V_e the maximum shear force carried

by the test specimen.

Tab. 1. Variables (and some stats on their values) considered for ANN simulations.

Input variables			ANN inputs	Values		
				min	max	average
Geometry	a (mm)	shear span	1	287	4500	990
	h_w (mm)	web height	2	260	2005	744
	t_w (mm)	web thickness	3	0.1	8	1.6
	b (mm)	width of the parallel fold	4	19.8	300	112
	d (mm)	projected width of the inclined fold	5	11.9	220	71.8
	h_r (mm)	corrugation depth	6	12	150.2	66.9

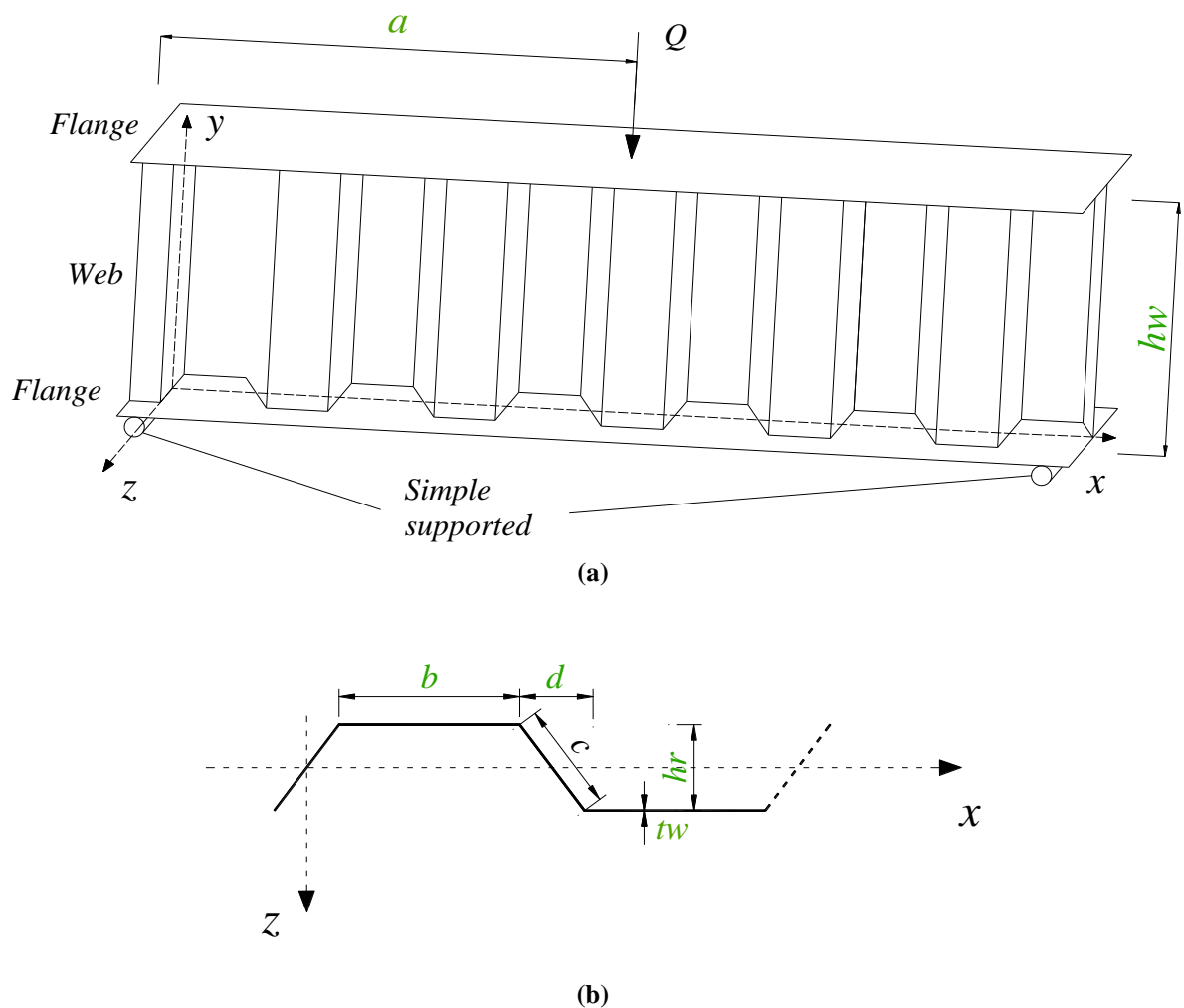| Material	f_y (MPa)	steel nominal yield stress	**7**	171	714	377.69
Target variable			**ANN output**	**Values**		
				min	max	average
Strength	τ_e (MPa)	ultimate shear stress	**1**	19.0	375.9	160.1

(a)

(b)

Fig. 1. (a) Simple supported corrugated web steel girder under concentrated load, and **(b)** mid surface profile of a single "wavelength" of the trapezoidal corrugated web.

3. Artificial Neural Networks

3.1   Introduction

The general ANN structure consists of several nodes disposed in L vertical layers (input layer, hidden layers, and output layer) and connected between them, as depicted in Fig. 2. Associated to each node in layers 2 to L, also called neuron, is a linear or nonlinear transfer (also called activation) function, which receives the so-called net input and transmits an output. All ANNs implemented in this work are called feedforward, since data presented in the input layer flows in the forward direction only, i.e. every node only connects to nodes belonging to layers located at the right-hand-side of its layer, as shown in Fig. 2. ANN's computing power makes them suitable to efficiently solve small to large-scale complex problems, which can be attributed to their (i) massively parallel distributed structure and (ii) ability to learn and generalize, i.e, produce reasonably accurate outputs for inputs not used during the learning (also called training) phase.
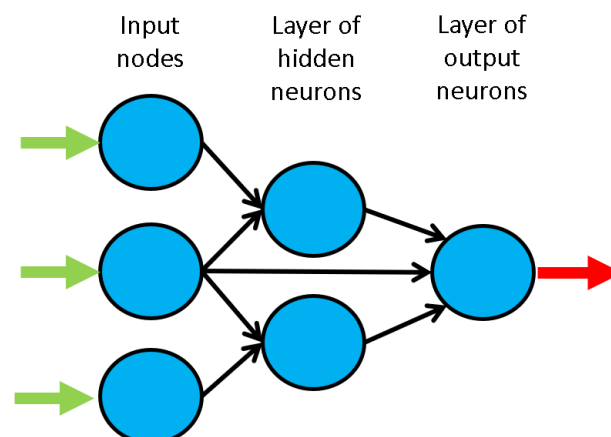

Fig. 2. Example of a feedforward neural network.

3.2   Learning

Each connection between 2 nodes is associated to a synaptic weight (real value), which, together with each neuron's bias (also a real value), are the most common types of neural net unknown parameters that will be determined through learning. Learning is nothing else than determining network unknown parameters through some algorithm in order to minimize network's performance measure, typically a function of the difference between predicted and target (desired) outputs. When ANN learning has an iterative nature, it consists of three phases: (i) training, (ii) validation, and (iii) testing. From previous knowledge, examples or data points are selected to train the neural net, grouped in the so-called training dataset. Those examples are said to be 'labelled' or 'unlabeled', whether they consist of inputs paired with their targets, or just of the inputs themselves – learning is called supervised (e.g., functional approximation, classification) or unsupervised (e.g., clustering), whether data used is labelled or unlabeled, respectively. During an iterative learning, while the training dataset is used to tune network unknowns, a process of cross-validation takes place by using a set of data completely distinct from the training counterpart (the validation dataset), so that the generalization performance of the network can be attested. Once 'optimum' network parameters are determined, typically associated to a minimum of the validation performance curve (called early stop – see Fig. 3), many authors still perform a final assessment of model's accuracy, by presenting to it a third fully distinct dataset called 'testing'. Heuristics suggests that early stopping avoids overfitting, i.e. the loss of ANN's generalization ability. One of the causes of overfitting might be learning

too many input-target examples suffering from data noise, since the network might learn some

of its features, which do not belong to the underlying function being modelled (Haykin 2009).
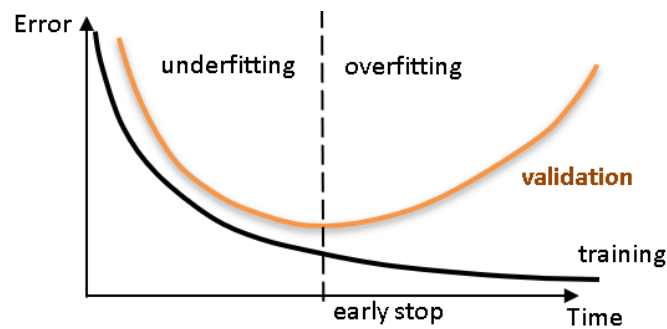

Fig. 3. Cross-validation - assessing network's generalization ability.

3.3   Implemented ANN features

The 'behavior' of any ANN depends on many 'features', having been implemented 15 ANN

features in this work (including data pre/post processing ones). For those features, it is important

to bear in mind that no ANN guarantees good approximations via extrapolation (either in functional

approximation or classification problems), i.e. the implemented ANNs should not be applied

outside the input variable ranges used for network training. Since there are no objective rules

dictating which method per feature guarantees the best network performance for a specific

problem, an extensive parametric analysis (composed of nine parametric sub-analyses) was carried

out to find 'the optimum' net design. A description of all methods/formulations implemented

for each ANN feature (see Tabs. 2-4) – they are a selection from state of art literature on ANNs,

including both traditional and promising modern techniques, can be found in previous

published works (e.g., Abambres et al. 2018) – the reader might need to go through it to fully

understand the meaning of all variables reported in this manuscript. The whole work was coded

in MATLAB (The Mathworks, Inc. 2017), making use of its neural network toolbox when dealing

with popular learning algorithms (1-3 in Tab. 4). Each parametric sub-analysis (SA) consists of

running all feasible combinations (also called 'combos') of pre-selected methods for each ANN

feature, in order to get performance results for each designed net, thus allowing the selection of the

best ANN according to a certain criterion. The best network in each parametric SA is the one

exhibiting the smallest average relative error (called performance) for all learning data.

Tab. 2. Implemented ANN features (F) 1-5.

FEATURE METHOD	F1 Qualitative Var Represent	F2 Dimensional Analysis	F3 Input Dimensionality Reduction	F4 % Train-Valid-Test	F5 Input Normalization
1	Boolean Vectors	Yes	Linear Correlation	80-10-10	Linear Max Abs
2	Eq Spaced in]0,1]	No	Auto-Encoder	70-15-15	Linear [0, 1]
3	-	-	-	60-20-20	Linear [-1, 1]
4	-	-	Ortho Rand Proj	50-25-25	Nonlinear
5	-	-	Sparse Rand Proj	-	Lin Mean Std
6	-	-	No	-	No

Tab. 3. Implemented ANN features (F) 6-10.

FEATURE METHOD	F6 Output Transfer	F7 Output Normalization	F8 Net Architecture	F9 Hidden Layers	F10 Connectivity
1	Logistic	Lin [a, b] = 0.7[φ_{min}, φ_{max}]	MLPN	1 HL	Adjacent Layers
2	-	Lin [a, b] = 0.6[φ_{min}, φ_{max}]	RBFN	2 HL	Adj Layers + In-Out
3	Hyperbolic Tang	Lin [a, b] = 0.5[φ_{min}, φ_{max}]	-	3 HL	Fully-Connected
4	-	Linear Mean Std	-	-	-
5	Bilinear	No	-	-	-
6	Compet	-	-	-	-
7	Identity	-	-	-	-

Tab. 4. Implemented ANN features (F) 11-15.

FEATURE METHOD	F11	F12	F13	F14	F15
	Hidden Transfer	Parameter Initialization	Learning Algorithm	Performance Improvement	Training Mode
1	Logistic	Midpoint (W) + Rands (b)	BP	NNC	Batch
2	Identity-Logistic	Rands	BPA	-	Mini-Batch
3	Hyperbolic Tang	Randnc (W) + Rands (b)	LM	-	Online
4	Bipolar	Randnr (W) + Rands (b)	ELM	-	-
5	Bilinear	Randsmall	mb ELM	-	-
6	Positive Sat Linear	Rand [-Δ, Δ]	I-ELM	-	-
7	Sinusoid	SVD	CI-ELM	-	-
8	Thin-Plate Spline	MB SVD	-	-	-
9	Gaussian	-	-	-	-
10	Multiquadratic	-	-	-	-
11	Radbas	-	-	-	-

3.4 Network Performance Assessment

Several types of results were computed to assess network outputs, namely (i) maximum error, (ii) % errors greater than 3%, and (iii) performance, which are defined next. All abovementioned errors are relative errors (expressed in %) based on the following definition, concerning a single output variable and data pattern,

$$e_{qp} = 100 \left| \frac{d_{qp} - y_{qLp}}{d_{qp}} \right|$$, (1)

where (i) d_{qp} is the qth desired (or target) output when pattern p within iteration i ($p=1,\ldots, P_i$) is presented to the network, and (ii) y_{qLp} is net's q^{th} output for the same data pattern. Moreover, denominator in eq. (1) is replaced by 1 whenever $|d_{qp}| < 0.05 - d_{qp}$ in the nominator keeps its

real value. This exception to eq. (1) aims to reduce the apparent negative effect of large relative errors associated to target values close to zero. Even so, this trick may still lead to (relatively) large solution errors while groundbreaking results are depicted as regression plots (target vs. predicted outputs).

3.4.1 Maximum Error

This variable measures the maximum relative error, as defined by eq. (1), among all output variables and learning patterns.

3.4.2 Percentage of Errors > 3%

This variable measures the percentage of relative errors, as defined by eq. (1), among all output variables and learning patterns, that are greater than 3%.

3.4.3 Performance

In functional approximation problems, network performance is defined as the average relative error, as defined in eq. (1), among all output variables and data patterns being evaluated (e.g., training, all data).

3.5 Software Validation

Several benchmark datasets/functions were used to validate the developed software, involving low- to high-dimensional problems and small to large volumes of data. Due to paper length limit, validation results are not presented herein but they were made public online (Researcher 2018b).

3.6   Parametric Analysis Results

Aiming to reduce the computing time by cutting in the number of combos to be run – note that all features combined lead to hundreds of millions of combos, the whole parametric simulation was divided into nine parametric SAs, where in each one feature 7 only takes a single value. This measure aims to make the performance ranking of all combos within each 'small' analysis more 'reliable', since results used for comparison are based on target and output datasets as used in ANN training and yielded by the designed network, respectively (they are free of any postprocessing that eliminates output normalization effects on relative error values). Whereas (i) the 1st and 2nd SAs aimed to select the best methods from features 1, 2, 5, 8 and 13 (all combined), while adopting a single popular method for each of the remaining features (F_3: 6, F_4: 2, F_6: {1 or 7}, F_7: 1, F_9: 1, F_{10}: 1, F_{11}: {3, 9 or 11}, F_{12}: 2, F_{14}: 1, F_{15}: 1 – see Tabs. 2-4) – SA 1 involved learning algorithms 1-3 and SA 2 involved the ELM-based counterpart, (ii) the $3^{rd} – 7^{th}$ SAs combined all possible methods from features 3, 4, 6 and 7, and concerning all other features, adopted the methods integrating the best combination from the aforementioned first SA, (iii) the 8th SA combined all possible methods from features 11, 12 and 14, and concerning all other features, adopted the methods integrating the best combination (results compared after postprocessing) among the previous five sub-analyses, and lastly (iv) the 9th SA combined all possible methods from features 9, 10 and 15, and concerning all other features, adopted the methods integrating the best combination from the previous analysis. Summing up the ANN feature combinations for all parametric SAs, a total of 475 combos were run for this work.

ANN feature methods used in the best combo from each of the abovementioned nine parametric sub-analyses, are specified in Tab. 5 (the numbers represent the method number as in Tabs 2-4). Tab. 6 shows the corresponding relevant results for those combos, namely (i) maximum error, (ii) % errors > 3%, (iii) performance (all described in section 3, and evaluated for all learning data), (iv) total number of hidden nodes in the model, and (v) average computing time per example (including data pre- and post-processing). All results shown in Tab. 6 (i) were obtained for single ANNs only, since no NNC networks yielded better results for this particular problem, and (ii) are based on target and output datasets computed in their original format, i.e. free of any transformations due to output normalization and/or dimensional analysis. The microprocessors used in this work have the following features: OS: Win10Home 64bits, RAMs: 48 GB, Local Disk Memory: 1 TB, CPUs: Intel® Core™ i7 8700K @ 3.70-4.70 GHz.

Tab. 5. ANN feature (F) methods used in the best combo from each parametric sub-analysis (SA).

SA	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
1	1	2	6	2	5	1	1	1	1	1	3	2	3	1	3
2	1	2	6	2	2	7	1	2	1	1	9	2	5	1	3
3	1	2	6	3	5	1	1	1	1	1	3	2	3	1	3
4	1	2	1	1	5	1	2	1	1	1	3	2	3	1	3
5	1	2	1	2	5	1	3	1	1	1	3	2	3	1	3
6	1	2	6	4	5	7	4	1	1	1	3	2	3	1	3
7	1	2	6	3	5	5	5	1	1	1	3	2	3	1	3
8	1	2	6	3	5	5	5	1	1	1	3	5	3	1	3
9	1	2	6	3	5	5	5	1	2	3	3	5	3	1	3

Tab. 6. Performance results for the best design from each parametric sub-analysis.

SA	ANN				
	Max Error (%)	Performance All Data (%)	Errors > 3% (%)	Total Hidden Nodes	Running Time / Data Point (s)
1	0.7	0.0	0.0	28	1.93E-04
2	265.3	15.6	72.4	90	9.80E-05
3	0.8	0.0	0.0	28	1.51E-04
4	0.9	0.0	0.0	28	1.29E-04
5	0.8	0.0	0.0	28	1.44E-04
6	0.1	0.0	0.0	28	1.01E-04
7	26.6	1.5	14.3	28	1.26E-04
8	17.0	1.2	12.4	28	1.41E-04
9	0.0	0.0	0.0	28	1.35E-04

3.7 Proposed ANN-Based Model

The proposed model is the one, among the best ones from all parametric SAs, exhibiting the lowest maximum error (SA 9). That model is characterized by the ANN feature methods {1, 2, 6, 3, 5, 5, 5, 1, 2, 3, 3, 5, 3, 1, 3} in Tabs. 2-4. Aiming to allow implementation of this model by any user, all variables/equations required for (i) data preprocessing, (ii) ANN simulation, and (iii) data postprocessing, are presented in 3.7.1-3.7.3, respectively. The proposed model is a single MLPN with 4 layers and a distribution of nodes/layer of 7-14-14-1. Concerning connectivity, the network is fully-connected, and the hidden and output transfer functions are all Hyperbolic Tangent and Bilinear, respectively. The network was trained using the Levenberg-Marquardt algorithm (978 epochs). After design, the average network computing time concerning the presentation of a single example (including data pre/postprocessing) is 1.35E-04 s – Fig. 4 depicts a simplified scheme of some of network key features. Lastly, all relevant performance results concerning the proposed

ANN are illustrated in 3.7.4. The obtained ANN solution for every data point can be found in Developer (2018a).
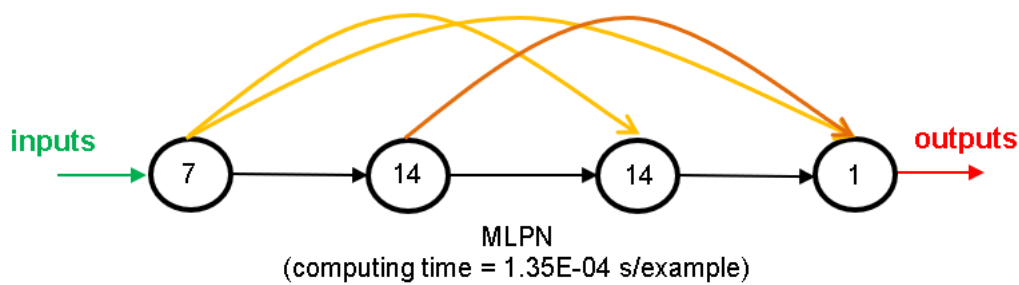

Fig. 4 Proposed 7-14-14-1 fully-connected MLPN – simplified scheme.

It is worth recalling that, in this manuscript, whenever a vector is added to a matrix, it means the former is to be added to all columns of the latter (valid in MATLAB).

3.7.1   Input Data Preprocessing

For future use of the proposed ANN to simulate new data $Y_{1,sim}$ (7 x P_{sim} matrix), concerning P_{sim} patterns, the same data preprocessing (if any) performed before training must be applied to the input dataset. That preprocessing is defined by the methods used for ANN features 2, 3 and 5 (respectively 2, 6 and 5 – see Tab. 2), which should be applied after all (eventual) qualitative variables in the input dataset are converted to numerical (using feature 1's method). Next, the necessary preprocessing to be applied to $Y_{1,sim}$, concerning features 2, 3 and 5, is fully described.

Dimensional Analysis and Dimensionality Reduction

Since dimensional analysis (*d.a.*) and dimensionality reduction (*d.r.*) were not carried out,

one has

$$\left\{Y_{1,sim}\right\}_{d.r.}^{after} = \left\{Y_{1,sim}\right\}_{d.a.}^{after} = Y_{1,sim} \tag{2}$$

Input Normalization

After input normalization, the new input dataset $\{Y_{1,sim}\}_n^{after}$ is defined as function of the

previously determined $\{Y_{1,sim}\}_{d.r}^{after}$, and they have the same size, reading

$$\left\{Y_{1,sim}\right\}_n^{after} = \left(\left\{Y_{1,sim}\right\}_{d.r}^{after} - \text{INP}(:,1)\right) ./\ \text{INP}(:,2)$$

$$\text{INP} = \begin{bmatrix} 990.400346190476 & 668.905673826269 \\ 744.012857142857 & 396.481809107959 \\ 1.55576190476190 & 1.28849686190348 \\ 111.986666666667 & 71.5898412598522 \\ 71.8395776190476 & 52.3896551822015 \\ 66.8513071428572 & 47.7496285409415 \\ 377.690000000000 & 133.759804523706 \end{bmatrix} \tag{3}$$

where one recalls that operator './' divides row *i* in the numerator by INP(*i*, 2).

3.7.2   ANN-Based Analytical Model

Once determined the preprocessed input dataset $\{Y_{1,sim}\}_n^{after}$(7 x P_{sim} matrix), the next step is

to present it to the proposed ANN to obtain the predicted output dataset $\{Y_{4,sim}\}_n^{after}$ (1 x P_{sim}

vector), which will be given in the same preprocessed format of the target dataset used in learning. In order to convert the predicted outputs to their 'original format' (i.e., without any transformation due to normalization or dimensional analysis – the only transformation visible will be the (eventual) qualitative variables written in their numeric representation), some postprocessing is needed, as described in detail in 3.7.3. Next, the mathematical representation of the proposed ANN is given, so that any user can implement it to determine $\{Y_{4,sim}\}_n{}^{after}$, thus eliminating all rumors that ANNs are 'black boxes'.

$$Y_2 = \varphi_2\left(W_{1-2}^T\left\{Y_{1,sim}\right\}_n^{after} + b_2\right)$$

$$Y_3 = \varphi_3\left(W_{1-3}^T\left\{Y_{1,sim}\right\}_n^{after} + W_{2-3}^TY_2 + b_3\right)$$

$$\left\{Y_{4,sim}\right\}_n^{after} = \varphi_4\left(W_{1-4}^T\left\{Y_{1,sim}\right\}_n^{after} + W_{2-4}^TY_2 + W_{3-4}^TY_3 + b_4\right) \qquad , (4)$$

where

$$\varphi_2 = \varphi_3 = \varphi(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$$

$$\varphi_4 = \varphi_4(s) = \begin{cases} s, & s \geq 0 \\ 0, & s < 0 \end{cases} \qquad . (5)$$

Arrays W_{j-s} and b_s are stored online in Developer (2018b), aiming to avoid an overlong article and ease model's implementation by any interested reader.

3.7.3   Output Data Postprocessing

In order to transform the output dataset obtained by the proposed ANN, $\{Y_{4,sim}\}_n^{after}$ (1 x P_{sim}

vector), to its original format ($Y_{4,sim}$), i.e. without the effects of dimensional analysis and/or output

normalization (possibly) taken in target dataset preprocessing prior training, one has

$$Y_{4,sim} = \left\{Y_{4,sim}\right\}_n^{after}$$

, (6)

since no output normalization nor dimensional analysis were carried out.

3.7.4   Performance Results

Finally, results yielded by the proposed ANN, in terms of performance variables defined in

sub-section 3.4, are presented in this section in the form of several graphs: (i) a regression plot

(Fig. 5) where network target and output data are plotted, for each data point, as x- and y-

coordinates respectively – a measure of linear correlation is given by the Pearson Correlation

Coefficient (R); (ii) a performance plot (Fig. 6), where performance (average error) values are

displayed for several learning datasets; and (iii) an error plot (Fig. 7), where values concern all

data (iii$_1$) maximum error and (iii$_2$) % of errors greater than 3%.
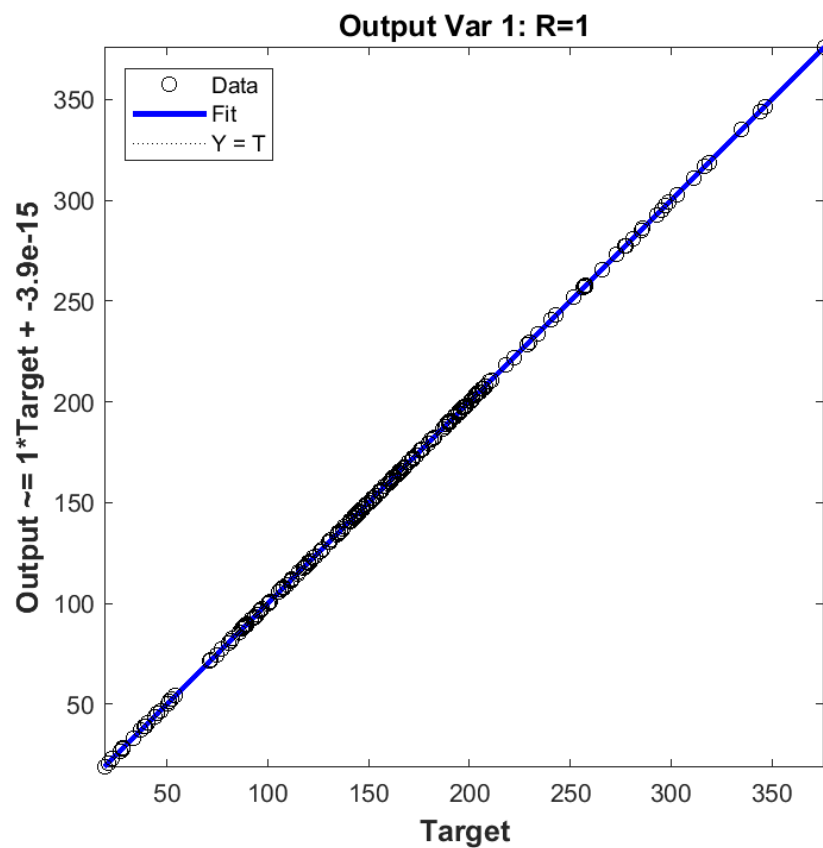
Fig. 5. Regression plot for the proposed ANN (see output variable in Fig. 1).
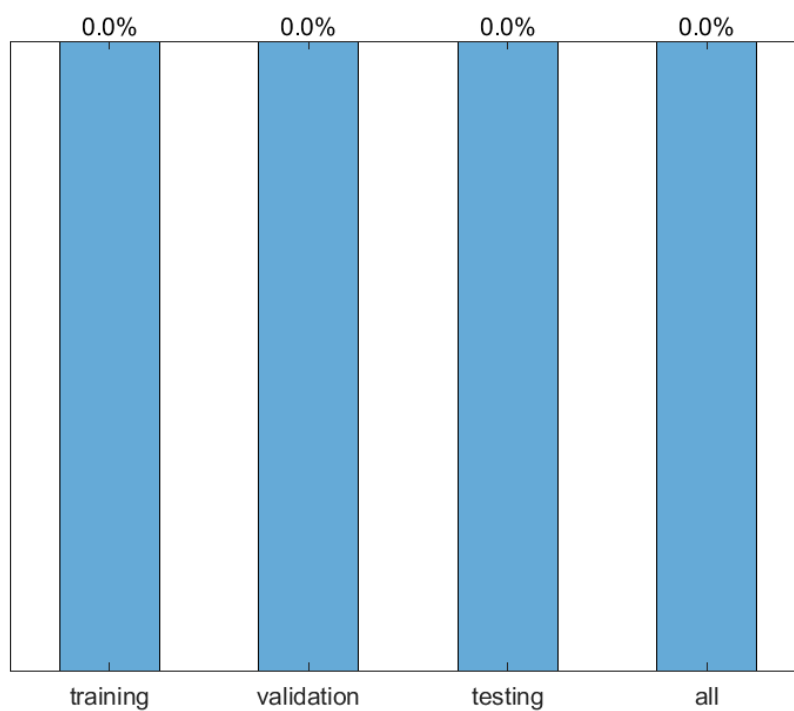
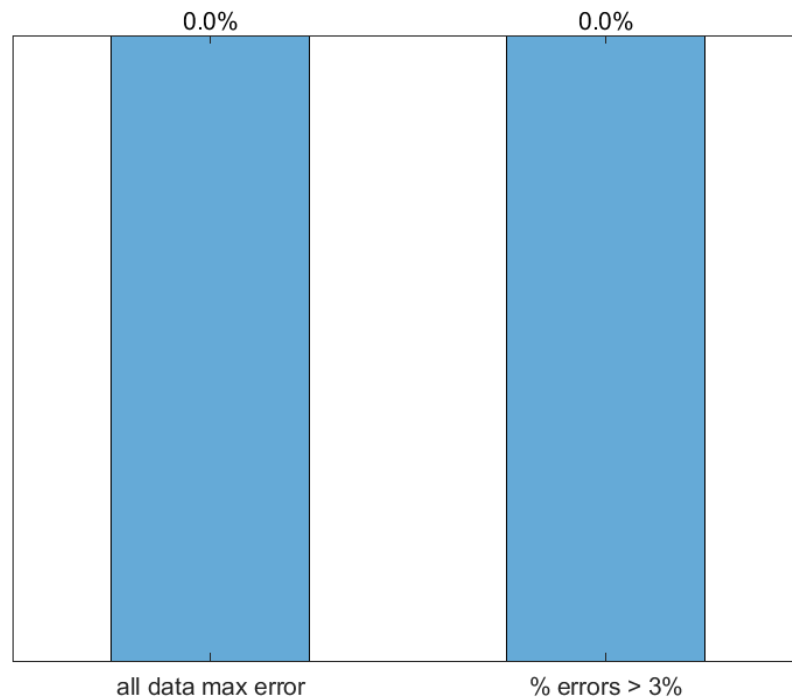**Fig. 6.** Performance plot (mean errors) for the proposed ANN.

Fig. 7. Error plot for the proposed ANN.

4. ANN-based vs. Existing Models

Shear strength of steel I-girders is controlled by buckling and/or shear yielding of the corrugated web. Shear buckling of corrugated webs is often classified as local buckling, global buckling, and interactive buckling, as exemplified in Fig.8 via the FEA-based displacement contours for those elastic buckling modes. Global buckling involves multiple folds and the buckled shape extends diagonally over the height of the web. Local buckling is controlled by deformations within a single "fold" of the web. The interactive shear buckling mode is attributed to the interaction between local and global shear buckling modes.
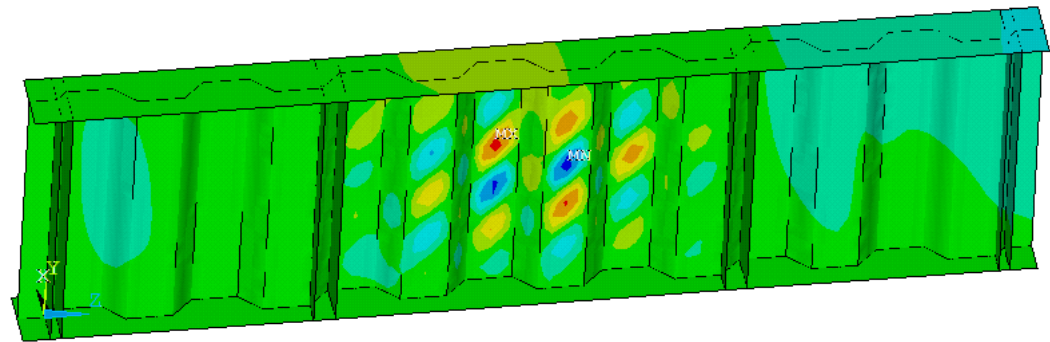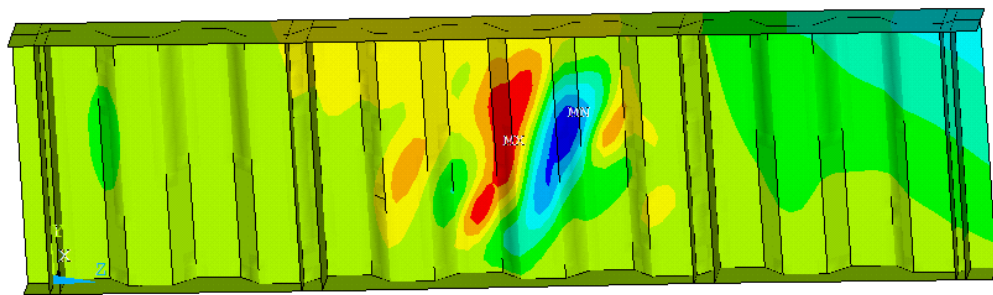
(a) Local buckling
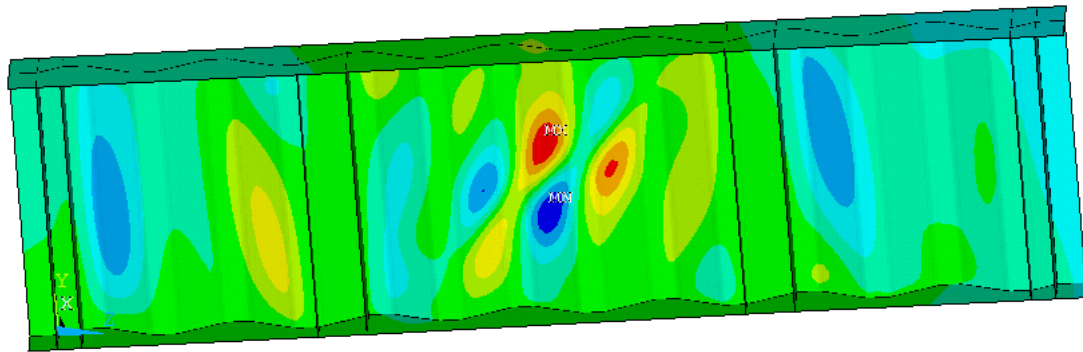
(b) Interactive buckling

(c) Global buckling

Fig.8. Shear buckling modes via FEA.

4.1 Local shear buckling

The local elastic shear buckling stress of a corrugated web can be predicted using classic

plate buckling theory (Timoshenko and Gere 1961). A single parallel or inclined fold is

assumed to be supported by the adjacent folds and steel flanges. The corresponding local elastic

shear buckling stress, $\tau^e_{cr,L}$ is

$$\tau^e_{cr,L} = k_L \frac{\pi^2 E}{12\left(1-\upsilon^2\right)} \left(\frac{t_w}{w}\right)^2$$
, (7)

where (i) w is the maximum fold width, max(b, c), and (ii) k_L is the local shear buckling

coefficient, which depends on the boundary conditions and the fold aspect ratio – k_L lies

between 5.34 (assuming simply supported edges) and 8.98 (assuming fixed edges). For

practical design purposes, k_L=5.34 is recommend by Moon *et al*. (2009).

4.2 Global shear buckling

An expression for the global elastic shear buckling stress of a corrugated steel plate ($\tau^e_{cr,G}$)

was developed by Easley and McFarland (1969) using orthotropic plate theory, reading

$$\tau^e_{cr,G} = k_G \frac{D_x^{3/4} D_y^{1/4}}{t_w h_w^2}$$
, (8)

being (i) k_G, the global shear buckling coefficient, and (ii) D_x and D_y, the bending stiffnesses

per unit length of the corrugated web with respect to its central principal axes x and y,

respectively. Easley (1975) proposed that k_G varies between 36 (assuming the web pin-ended

by the flanges) and 68.4 (assuming the web fix-ended by the flanges). Elgaaly et al (1996)

suggested that k_G should be taken as 31.6 for simply supported boundaries and 59.2 for the

clamped counterpart. D_x and D_y for trapezoidal corrugated webs can be determined as:

$$D_x = \frac{EI_x}{b + h_r} \tag{9}$$

$$D_y = \frac{Et_w^3}{12} \frac{b+d}{b+c} \tag{10}$$

where I_x is the moment of inertia about the x-axis, $I_x = 2bt_w(h_r/2)^2 + h_r^3 t_w/(6\sin\alpha)$.

4.3 Interactive shear buckling

The interactive shear buckling mode is attributed to the interaction between local and global

modes and governs shear buckling strength. Lindner & Aschinger (1988) first proposed the

corresponding elastic shear buckling stress formula ($\tau^e_{cr,I}$)

$$\tau^e_{cr.I} = \frac{\tau^e_{cr,L} \cdot \tau^e_{cr,G}}{\left(\left(\tau^e_{cr,G}\right)^n + \left(\tau^e_{cr,L}\right)^n\right)^{1/n}} \tag{11}$$

being exponent n an integer – several researchers have proposed distinct values (Yi et al. 2008,

Sause & Braxtan 2011, Hassanein & Kharoob 2013a, Leblouba et al. 2017b).

4.4 Shear strength

Previous studies (Yi et al. 2008, Sause & Braxtan 2011) have shown that shear strength of

corrugated steel webs was generally controlled by interactive shear buckling. In this context,

the shear buckling non-dimensional slenderness ($\lambda_{I,n}$) of a corrugated steel web is defined as

$$\lambda_I = \sqrt{\tau_y / \tau^e_{cr,I}} \quad , \quad (12)$$

where τ_y is the tangential yield stress (typically $f_y / \sqrt{3}$, being f_y the normal yield stress).

Driver et al. (2006) proposed eq. (11) to calculate the shear capacity of corrugated steel webs for all types of shear failures, reading (n=2)

$$\tau_D = \sqrt{\frac{\left(\tau^e_{cr,L} \cdot \tau^e_{cr,G}\right)^2}{\left(\tau^e_{cr,G}\right)^2 + \left(\tau^e_{cr,L}\right)^2}} \quad . \quad (13)$$

In case any elastic shear buckling stress ($\tau^e_{cr,\,G}$ or $\tau^e_{cr,\,L}$) exceeds 80% of the shear yield stress τ_y, its value in eq. (13) should be replaced by an inelastic counterpart given by (Elgaaly et al. 1996)

$$\tau^{in}_{cr} = \sqrt{0.8\tau_y \tau^e_{cr}} \leq \tau_y \quad . \quad (14)$$

El-Metwally (1998) proposed the following equation to calculate the shear capacity of corrugated steel webs (using n=2 in eq. (11))

$$\tau_{EL} = \tau_y \times \left(1 / \left(\left(\lambda_I\right)^4 + 1\right)\right)^{1/2} \quad . \quad (15)$$

Sauce and Braxtan (2011) summarized a large number of previous experimental data, and selected 22 groups of results to fit eq. (16), for the prediction of the shear capacity of corrugated steel webs (using n=3 in eq. (11))

$$\tau_{Sauce} = \tau_y \times \left(1 / \left(\left(\lambda_I\right)^6 + 2\right)\right)^{1/3} \quad . \quad (16)$$

Leblouba et al. (2017a), based on 113 test results collected from the literature, and 12 tests

carried out by themselves, developed the following analytical model (based on the hyperbolic

Richards equation) to calculate the shear strength of corrugated steel webs (using $n=4$ in eq. (11))

$$\tau_{\text{Leblouba}} = \tau_y \times \left(1 / \left(\left(\lambda_I / 1.58 \right)^{1.6} + 1 \right) \right)^{1.15} \qquad . \quad (17)$$

Fig.9 compares the shear strengths yielded by the analytical models presented before (τ_D,

τ_{EL}, τ_{Sauce}, $\tau_{Leblouba}$, τ_{ANN}) to those obtained experimentally (τ_e), for the 210 steel girders assessed

in this work (dataset in Developer 2018a). The average ratios τ_D/τ_e, τ_{EL}/τ_e, τ_{Sauce}/τ_e, $\tau_{Leblouba}/\tau_e$

are 0.78, 0.87, 0.82 and 0.83, with standard deviations of 0.13, 0.16, 0.13 and 0.11, respectively.

It can be found that all those models underestimate the shear strength of trapezoidal corrugated

web girders. For comparison, the average value of τ_{ANN}/τ_e is 1.00, with a standard deviation of

0.00. The major improvement of the proposed ANN-based analytical model (see sub-section

3.7), as compared to the existing calculation methods, is quite clear in Fig. 9, where x-axis

shows the predicted shear capacity τ_{model} (τ_{ANN}, τ_D, τ_{EL}, τ_{Sauce}, $\tau_{Leblouba}$) and the y-axis shows the
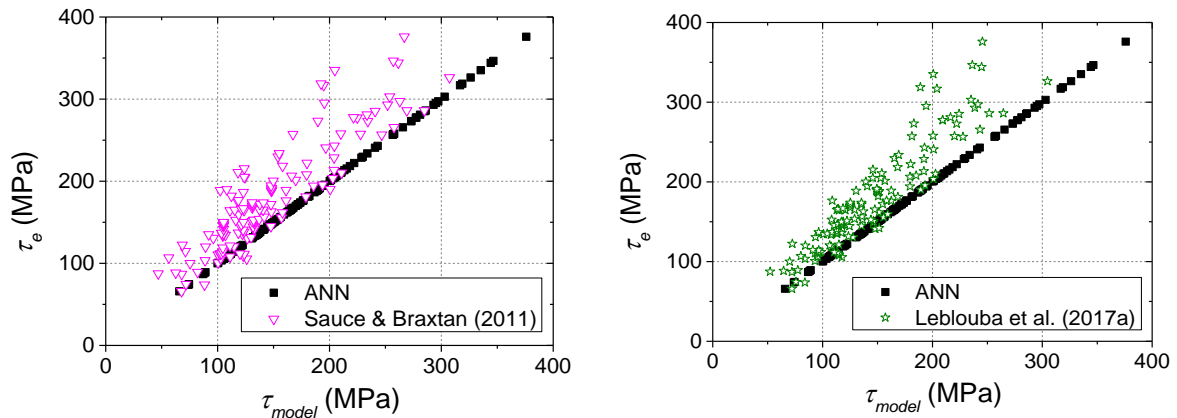
experimental counterpart τ_e.
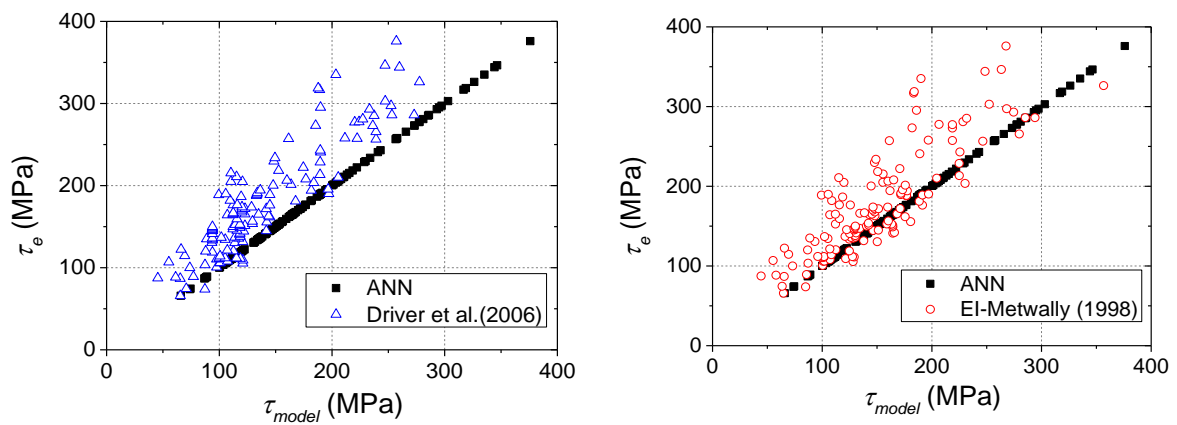
Fig. 9. Comparison between tested and predicted shear strength for several proposed models.

5. Discussion

Regardless the high quality of the predictions yielded by the proposed model, the reader should not **blindly** accept it as accurate for any other instances falling inside the input domain of the design dataset. Any analytical approximation model must undergo extensive validation before it can be taken as reliable (the more inputs, the larger the validation process). Models proposed meanwhile are part of a learning process towards excellence.

6.   Conclusions

This paper describes how artificial neural networks (ANN) can be used to predict the shear capacity of steel girders with a trapezoidal corrugated web, and proposes an analytical model for that purpose. The developed model was designed from a 210-point database of test results available in the literature. Seven governing (geometrical and material) parameters were identified as input variables, and the ultimate shear stress (assumed uniform along web's

height) at failure was considered as the target/output variable for the ANN simulations. The proposed ANN-based analytical model yielded maximum and mean relative errors of 0.0% concerning all the 210 test results previously collected. Fig. 9 shows that the ANN-based approach clearly outperforms the existing calculation models assessed in this work, for the dataset considered (made available at Developer 2018a) – latter models exhibit mean errors greater than 13%.

The focus of this study was not to assess the mechanics underlying the behavior of corrugated web steel girders, but parametric studies by means of accurate and robust ANN-based models make it possible to evaluate and improve existing mechanical models.

Contributions

He J. developed sections 1, 2 and 4; Abambres M. developed sections 3 and 5 (ANN-related); Remaining sections had equal contributions from both authors.

Acknowledgements

There are no conflicts of interest to disclose.

References

Abambres M, Rajana K, Tsavdaridis K, Ribeiro T (2018). Neural Network-based formula for the buckling load prediction of I-section cellular steel beams, hal-02074835

Abbas HH (2003). Analysis and design of corrugated web I-girders for bridges using high performance steel. Ph.D. dissertation. Bethlehem (PA): Dept. of Civil and Environmental Engineering, Lehigh University.

Bergfelt A, Leiva-Aravena L (1984). Shear Buckling of Trapezoidal Corrugated Girder Webs, Division of Steel and Timber Structures, Chalmers University of Technology, Gothenburg, Publication S 84:2, Sweden, 64p.

Combault J (1988). The Maupre Viaduct Near Charolles, France. Proceedings, AISC Engineering Conference, 12.1-12.22.

Developer (2018a). dataset + target vs output [Data set], downloadable

Developer (2018b). W and b arrays [Data set], downloadable

Driver RG, Abbas HH, Sause R (2006). Shear behavior of corrugated web bridge girders. Journal of Structural Engineering, ASCE,32(2), 195–203.

Easley JT (1975). Buckling formulas for corrugated metal shear diaphragms, J. Struct. Div. ASCE. 101, 1403–1417.

Easley JT, McFarland DE (1969). Buckling of light-gage corrugated metal shear diaphragms. Journal of the Structural Division, ASCE, 95(ST7), 1497-516.

Elgaaly M, Hamilton RW, Seshadri A (1996). Shear strength of beams with corrugated webs. Journal of Structural Engineering, ASCE, 122(4), 390-8.

Emami F, Mofid M, Vafai A (2013). Experimental study on cyclic behavior of trapezoidally corrugated steel shear walls, Eng. Struct., 48, 750–762.

EN 1993-1-5: 2005; Eurocode 3: Design of steel structures, Part 1-5: Plated structural elements.

Flood I (2008). Towards the next generation of artificial neural networks for civil engineering, Advanced Engineering Informatics, 228(1), 4-14.

Gil H, Lee S, Lee J, Lee H (2005). Shear buckling strength of trapezoidally corrugated steel webs for bridges, Journal of Transportation Research Board, 473-480

Hassanein MF, Kharoob OF (2013a). Behavior of bridge girders with corrugated webs: (I) Real boundary condition at the juncture of the web and flanges, Eng. Struct. 57, 554–564.

Hassanein MF, Kharoob OF (2013b). Behavior of bridge girders with corrugated webs: (II) Shear strength and design, Eng. Struct. 57,544–553.

Haykin SS (2009). Neural networks and learning machines, Prentice Hall/Pearson, New York.

He J (2011). Mechanical performance and design method of composite bridge with corrugated steel webs. Ph.D. dissertation, Tongji University, Shanghai, China.

He J, Liu Y, Chen A, Yoda T (2012). Mechanical behavior and analysis of composite bridges with corrugated steel webs: State-of-the-art, Int. J. Steel Struct. 12,321–338.

Hern A (2016). Google says machine learning is the future. So I tried it myself. Available at: www.theguardian.com/technology/2016/jun/28/all (Accessed: 2 November 2016).

Hertzmann A, Fleet D (2012). Machine Learning and Data Mining, Lecture Notes CSC 411/D11, Computer Science Department, University of Toronto, Canada.

Jiang RJ, Au FTK, Xiao YF (2015). Prestressed Concrete Girder Bridges with Corrugated Steel Webs : Review, J. Struct. Eng. ASCE, 141,1–9.

Johnson RP, Cafolla J (1997). Local flange buckling in plate girders with corrugated steel webs. Proceedings, Instn Civ Engrs Structs&Bldgs, Vol. 123, May, 148-156.

JSCE (1998). Design manual for PC bridges with corrugated steel webs, Research committee for hybrid structures with corrugated steel webs, Japan. (in Japanese).

Kondo M, Shimizu Y, Kobayashi K, Hattori M (1994). Design and construction of the Shinkai Bridge-Prestressed concrete bridge using corrugated steel webs. Bridge Found. Eng.,13-20. (in Japanese).

Leblouba M, Barakat S, Altoubat S, Junaid MT, M. Maalej (2017a). Normalized shear strength of trapezoidal corrugated steel webs, J. Constr. Steel Res. 136, 75–90.

Leblouba M, Junaid MT, Barakat S, Altoubat S, Maalej M (2017b). Shear buckling and stress distribution in trapezoidal web corrugated steel beams, Thin-Walled Struct. 113, 13–26.

Leiva-Aravena L, Edlund B (1987). Buckling of trapezoidally corrugated webs. In: ECCS colloquium on stability of plates and shells. Belgium: Ghent University.

Lindner J, Aschinger R(1988). Grenzschubtragfähigkeit von I-Tägern mit trapezförmig profilierten Stegen. Stahlbau, 57(12), 377-80. (in German).

Luo R, Edlund B (1994). Buckling analysis of trapezoidally corrugated panels using spline finite strip method. Thin-walled structures, 18, 209-224.

Luo R, Edlund B (1996). Shear capacity of plate girders with trapezoidally corrugated webs. Thin-walled structures, 26, 19-44.

McCulloch WS, Pitts W (1943). A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics, 5(4), 115–133.

Metwally AE (1998). Prestressed composite girders with corrugated steel webs. M.S. thesis. Calgary (AB): Dept. of Civil Engineering, University of Calgary.

Moon J, Yi J, Choi BH, Lee HE (2009). Shear strength and design of trapezoidally corrugated steel webs. Journal of Constructional Steel Research, 65(5), 1198-205.

Nie JG, Zhu L, Tao MX, Tang L (2013). Shear strength of trapezoidal corrugated steel webs, Journal of Constructional Steel Research, 85, 105–115.

Papangelis J, Trahair N, Hancock G (2017). Direct strength method for shear capacity of beams with corrugated webs, Journal of Constructional Steel Research 137,152–160.

Peil U (1998). Statische versuche an trapezstegtr̈agern untersuchung der querkraftbeanspruchbarkeit, Institut f̈ur Stahlbau. Braunschweig (Germany): Technischen Universiẗat Braunschweig.

Prieto A, Prieto B, Ortigosa EM, Ros E, Pelayo F, Ortega J, Rojas I (2016). Neural networks: An overview of early research, current frameworks and new challenges, Neurocomputing, 214(November), 242-268.

Researcher, The (2018b). "Ann software validation-report. pdf", figshare, doi: 10.6084/m9.figshare.6962873.

Sause R, Clarke T (2003). Bearing Stiffeners and Field Splices for Corrugated Web Girders, Technical Report.

Sause R, Abbas HH, Wassef W, Driver RG, Elgaaly M (2003). Corrugated web girder shape and strength criteria, Technical Report.

Sause R, Braxtan TN (2011). Shear strength of trapezoidal corrugated steel webs. Journal of Constructional Steel Research, 67, 223-236.

Sayed-Ahmed EY (2001). Behaviour of Steel and/or Composite Girders with Corrugated Steel Webs, Canadian Journal of Civil Engineering, Vol. 28, No. 4, 656-672.

Sayed-Ahmed EY (2007). Design aspects of steel I-girders with corrugated steel webs. Electronic Journal of Structural Engineering, 7, 27-40.

Shimada S (1965). Shear strength of steel plate girders with folded web plate (Ripple Web Girders). Proceeding of JSCE, 124, 1-10. (in Japanese).

The Mathworks, Inc (2017). MATLAB R2017a, User's Guide, Natick, USA.

Timoshenko SP, Gere JM (1961). Theory of elastic stability, McGraw-Hill Publishing Co., NewYork.

Watanabe K, Uchida S, Kubo M (2007). Shear buckling capacity of steel girders with corrugated webs. Journal of Structural Engineering, JSCE, 53A, 13-24. (in Japanese).

Wilamowski BM, Irwin JD (2011). The industrial electronics handbook: Intelligent Systems, CRC Press, Boca Raton.

Wu LL, Gao XN, Shi YJ, Wang YQ (2006). Theoretical and experimental study on interactive local buckling of arch-shaped corrugated steel roof, Int. J. Steel Struct, 6 45–54.

Yamazaki M (2001). Buckling strength of corrugated webs. Journal of Structural Engineering, JSCE, 47A, 19-26. (in Japanese).

Yi J, Gil H, Youm K, Lee H (2008). Interactive shear buckling behavior of trapezoidally corrugated steel webs. Engineering Structures, 30(6), 1659-66.

Software bugs in previous papers – sources and solutions

On **March 2021**, Abambres found some bugs on his own ANN software, which had been used to yield ANN results for several papers published until that date, including this one. A not-so-severe bug was found in the definition of the 8th parametric sub-analysis, since it was not defined exactly as it was described in section "Parametric Analysis Results". The critical bugs were found in non-ELM (non-Extreme Learning Machine) learning algorithms for "mini-batch" and "online" training modes, and unintendedly caused the "% Train - Valid - Test" ANN feature to become "100 - 0 - 0" during neural net design, thus affecting the generalization potential of the proposed ANN. The bug sources were the following:

- "trainlm", "traingd" and "traingda" training functions do not allow incremental (online or mini-batch) training when using MATLAB neural network toolbox – regardless the data format employed in *train(…)* arguments, only batch training will be used.
- if *net.divideMode* is not specified when using MATLAB neural net toolbox, the default value for feedforward ANNs is *'sample'*. However, the implemented data division (train / valid / test) for incremental training uses timestep (instead of sample) indexes, which means the assignment *net.divideMode = 'time'* was missing right before *net.divideFcn = 'divideind'*.

In order to overcome the above cited issues, Abambres first explored MATLAB's training functions that are said to allow incremental training, but found several limitations (reported in [here](https://archive.ph/r6Yw3) https://archive.ph/r6Yw3 and [here](https://archive.ph/HsaC6) https://archive.ph/HsaC6). Thus, Abambres' final decision was to redefine the ANN parametric analysis in order to (i) remove all incremental non-ELM learning algorithms, and (ii) increase the ELM-based simulations while improving their cross-validation.

Miguel Abambres

Final Note

For personal and professional reasons, I have decided not to write/publish (pro-bono) the new version of this paper for the results obtained with the debugged ANN software, even though the former were pretty satisfactory.

Miguel Abambres