Proceedings of the
Annual Stability Conference
Structural Stability Research Council
Denver, Colorado, March 22-25 2022

Prediction of the load-displacement and local buckling behavior of hollow structural sections using Deep Neural Networks (DNN)

Andreas Müller[1], Andreas Taras[2]

Abstract

The general design approach to account for geometrical and material nonlinearities within steel structures is made partially during the structural analysis and partially through subsequent cross-section and member verifications. Typically, structural analysis of steel frames is performed by using beam finite elements, which are capable of taking into account geometrical imperfections to account for second order effects but not able to capture the local buckling phenomena explicitly. Its assessment is therefore made through the concept of cross-section classification, creating artificial steps within the capacity verification of structural members by placing class-specific restrictions on the analysis. This simplified assumption particularly affect structural steel members benefiting from strain hardening and members which would need to be classified in slender regions.

To overcome some of this code related shortcomings more advanced design by analysis approaches are developed, combining the computational efficiency of beam-element models with the ability to account for slenderness-dependent deformation capacities (CSM-approach). Although, this approach imitates the effects of local buckling without the actual need to perform complicated shell based GMNIA calculations, it is only valid in the pre-buckling range. The proposed paper herein, presents a method to carry out beam-element analysis that accounts for the nonlinear load-displacement behavior in the pre- and post-buckling range of sections with various local slenderness. The DNN-DSM, which makes use of machine learning techniques (deep-neural-networks – DNN) to predict the nonlinear stiffness matrix terms in a beam-element formulation within the Direct-Stiffness-Method (DSM). Based on DNN-models trained on the results from an extensive pool of nonlinear (GMNIA) shell element simulations, first outcomes of this method are able to describe the nonlinear load-displacement behavior of various SHS and RHS sections loaded by a normal force N acting in the cross-sectional center of gravity.

1. Introduction and Motivation

1.1 Problem Definition

Structural hollow sections are particularly suitable in a wide field of engineering applications, especially when the need for structural efficiency and optimization demands arise. In this context the overall structure, the profile geometry itself or the steel grades are optimized to a level where the

[1] PhD Student, Steel and Composite Structures, ETH Zurich <andreas.mueller@ibk.baug.ethz.ch>
[2] Professor, Steel and Composite Structures, ETH Zurich <taras@ibk.baug.ethz.ch>

desired design optimum is reached. Due to the increasingly better and simpler possibilities of parameterizing the structural models by means of suitable software, changes can be carried out in a fraction of time that would have been required decades ago. Nevertheless, the calculations and design checks done within the software are mostly based on code provisions of EC3 or AISC 360-16 (EN 1993-1-1 2006, AISC 360-16 2016) which imply simplifications and assumptions derived and designed with the goal to be suited in the framework for hand calculations. The traditional separation of analysis and verification, whereby both are dependent on the cross-sectional slenderness and corresponding classification into categories ranging from stocky to slender, affects high strength steel (HSS) sections in particular. These often would need to be classified as slender cross-sections, forcing designers to take local buckling phenomena into account (EN 1993-1-5 2006). Available deformation capacities (Toffolon et.al. 2019a, Toffolon et.al. 2019b, Toffolon and Taras 2019c, Toffolon and Taras 2019d, Meng et.al. 2019, Müller and Taras 2019), different material properties between hot rolled and cold-formed structural profiles (EN 10210-2 2006; EN 10219-2 2006) including strain hardening and more precise material laws, proposed by (Yun and Gardner 2017, Yun and Gardner 2018) and partially adopted in the current draft of prEN1993-1-14, are neglected. In reality, in spite of their lower hardening capacity and ultimate strains, even slender high-strength steel hollow sections display a non-negligible plastic strength component and rotational capacity that exceeds these code restrictions. Thus, new formulations and methods are needed to do justice to the advantages of high-strength steel sections and slender sections in general, particularly for the case of hollow structural sections, developing new design methodologies that make increasing use of numerical simulations (design by analysis).

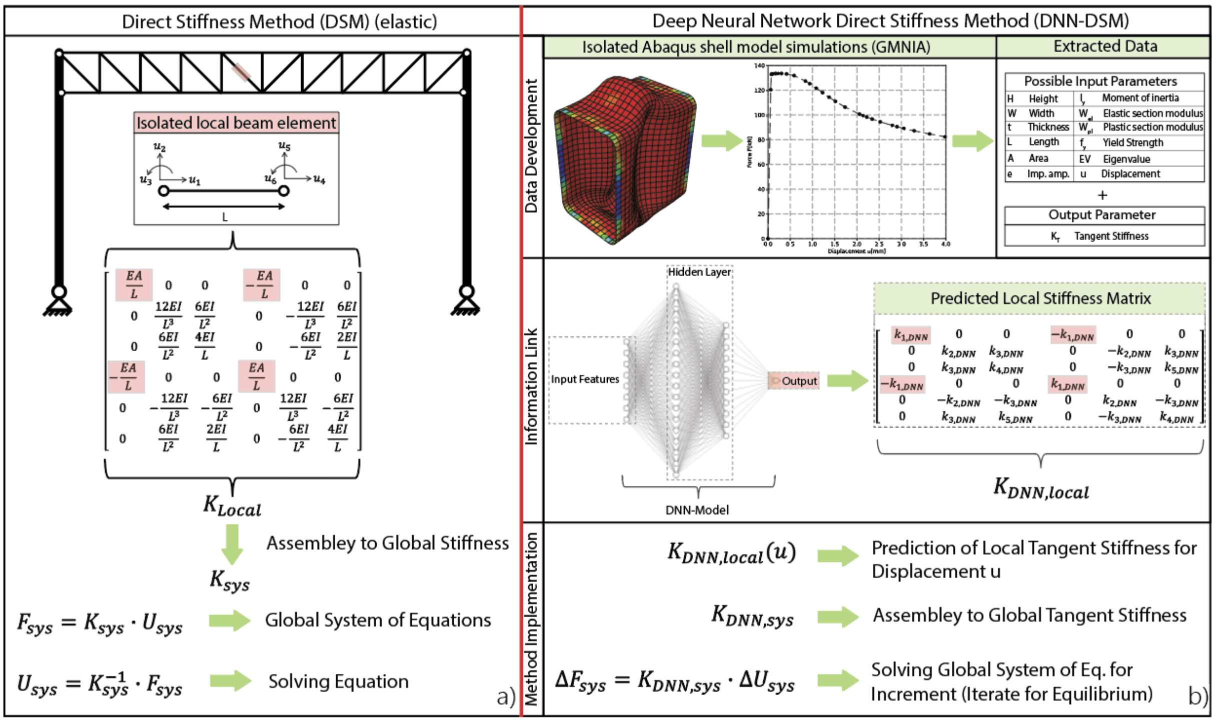
It is possible to carry out advanced finite element GMNIA (**g**eometrically **m**aterial **n**onlinear **a**nalysis with **i**mperfections) simulations using shell elements in structural design, as these provide realistic and accurate solutions depending on the modelling effort and problem knowledge. Nevertheless, these methods are computationally time intensive, due to the model size and complexity, and are not generally suitable for use in a design. For this reason, currently various efforts are being undertaken to combine the computational efficiency of beam-element models (which are already commonly used in design practice) with the ability to account for slenderness-dependent deformation capacities and nonlinear redistribution of internal forces in a structural truss or frame. Existing methods make use of beam finite element based GMNIA simulations with slenderness dependent strain limits imitating the effects of local buckling (CSM approach (Gardner 2008, Fieber 2019, Walport 2019)) or expand the beam element formulation to local buckling modes (GBT approach (Schard 1989, Silvestre 2005))

This paper presents the initial steps towards a novel approach to carry out a beam-element analysis that accounts for the nonlinear load-displacement behaviour of hot rolled and cold-formed SHS and RHS sections (DIN EN 10210-2 2006, DIN EN 10219-2 2006) of different local slenderness. The key challenge hereby is the accurate representation of the different load-displacement behaviour of elements of various local slenderness in the beam stiffness matrix, i.e. the question of how to include local buckling and plasticity in the beam structural analysis.

Fig 1 a) shows the general implementation within the elastic DSM formulation using the example of a truss frame, dominated by normal forces. An isolated beam element represented by the local elastic stiffness matrix K_{local} is constructed, assembled to a global system of equations and solved by calculating the inverse global stiffness matrix and subsequently the associated displacements

U_{sys}. Here only the terms dominated by the normal force are highlighted in red, since the first implementation steps presented within this paper will focus exclusively on this load case. The novel method denominated as DNN-DSM (deep neural network direct stiffness method) makes use of machine and deep learning techniques (ML and DL) to predict the nonlinear stiffness matrix of a beam element under different deformations and rotations acting in plane.

A general overview, starting with data development up to the method implementation, is presented within Fig. 1 b). The developed deep neural network (DNN) models are based on data sets derived from a pool of numerical (LBA and GMNIA) shell elements simulations, designed in such a way that only local buckling is the driving instability phenomena for the investigated cross-sections (s. Sec. 2.1). Therefore the local length of the elements was set to the maximum of whether the height or the width of the cross-section. This assumption was made within a first feasibility study. Thorough investigations on cross-section and load dependent buckling lengths are under way, following up the research carried out by (Fieber 2019). The extracted data includes geometrical and mechanical parameters based on (DIN EN 10210-2 2006, DIN EN 10219-2 2006) as well as directly simulation related outputs including the cross-section dependent displacement u and the tangent stiffness K_T. Subsequently, the resulting database is used for the training of the DNN models, with the tangent stiffness as the governing output parameter (s. Sec. 2.2). The DNN based prediction of the tangent stiffness $K_{DNN,local}$ is thereby evaluated for every discretized beam element within a global structure, assembled to a global stiffness matrix $K_{DNN,sys}$ and used to evaluate a differential force ΔF_{sys}, which is calculated under the consideration of incremental displacement steps ΔU_{sys}.



Direct Stiffness Method (DSM) (elastic)
Isolated local beam element
u_2
u_3
u_1
u_5
u_6
u_4
L
K_{Local}
Assembley to Global Stiffness
K_{sys}
$F_{sys} = K_{sys} \cdot U_{sys}$
Global System of Equations
$U_{sys} = K_{sys}^{-1} \cdot F_{sys}$
Solving Equation
a)
Deep Neural Network Direct Stiffness Method (DNN-DSM)
Data Development
Isolated Abaqus shell model simulations (GMNIA)
Extracted Data
Possible Input Parameters
H Height
W Width
t Thickness
L Length
A Area
e Imp. amp.
I_y Moment of inertia
W_{el} Elastic section modulus
W_{pl} Plastic section modulus
f_y Yield Strength
EV Eigenvalue
u Displacement
+
Output Parameter
K_T Tangent Stiffness
Information Link
Input Features
Hidden Layer
Output
DNN-Model
Predicted Local Stiffness Matrix
$K_{DNN,local}$
Method Implementation
$K_{DNN,local}(u)$
Prediction of Local Tangent Stiffness for Displacement u
$K_{DNN,sys}$
Assembley to Global Tangent Stiffness
$\Delta F_{sys} = K_{DNN,sys} \cdot \Delta U_{sys}$
Solving Global System of Eq. for Increment (Iterate for Equilibrium)
b)

$$K_{Local} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}$$

$$K_{DNN,local} = \begin{bmatrix} k_{1,DNN} & 0 & 0 & -k_{1,DNN} & 0 & 0 \\ 0 & k_{2,DNN} & k_{3,DNN} & 0 & -k_{2,DNN} & k_{3,DNN} \\ 0 & k_{3,DNN} & k_{4,DNN} & 0 & -k_{3,DNN} & k_{5,DNN} \\ -k_{1,DNN} & 0 & 0 & k_{1,DNN} & 0 & 0 \\ 0 & -k_{2,DNN} & -k_{3,DNN} & 0 & k_{2,DNN} & -k_{3,DNN} \\ 0 & k_{3,DNN} & k_{5,DNN} & 0 & -k_{3,DNN} & k_{4,DNN} \end{bmatrix}$$

Figure 1: a) Direct Stiffness Method (elastic formulation), b) Initial formulation of the Deep Neural Network Direct Stiffness Method (DNN-DSM)

1.2 Introduction to Deep Neural Networks

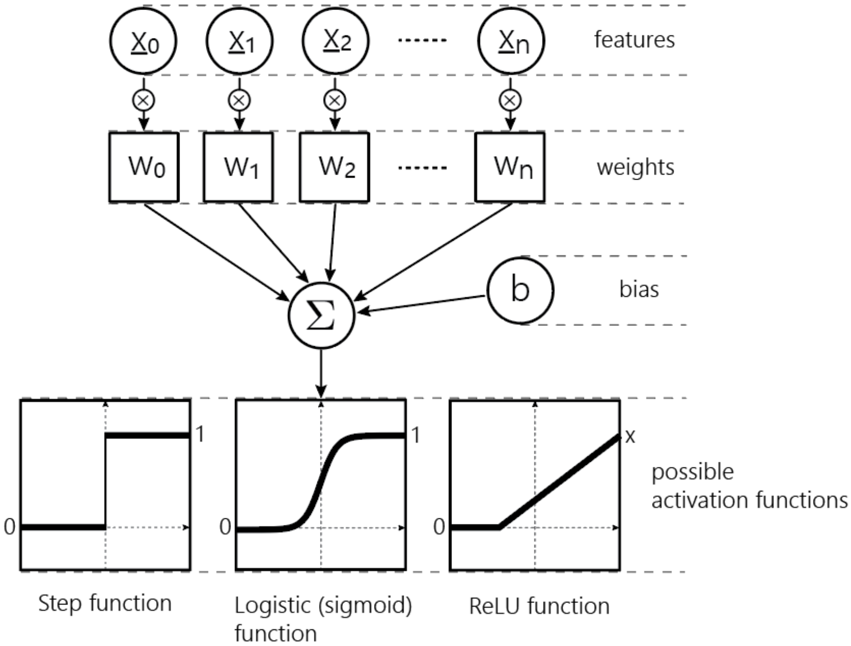


Figure 2: General representation of a neuron

The concept of deep neural networks is based on findings from (McCulloch and Pitts 1943, Hebb 1949, Minsky and Seymour 1969), in the period between the 1940s to the end of the 1960s. Nowadays its increasing popularity (of machine learning in a general sense) is fuelled by the access to large amounts of data, the availability of Graphics Processing Units (GPUs), algorithmic developments and an easier accessibility to the field of machine learning due to the development of high-level libraries/APIs. Apart from the rapid spread of ML and DL related methods within the automotive industry or applications of facial recognition, the first applications are also being developed in the fields of civil engineering. In (D´Aniello et.al. 2014, Güneyisi et.al. 2014) DNN models are used to obtain the available rotation capacity and the flexural overstrength factor for steel beams. Fonseca implemented and evolved throughout several publications (Fonseca et.al. 2001, Fonseca et.al. 2003a, Fonseca et.al. 2003b) NN models to predict and estimate data for the patch load behaviour. In (Fonseca et.al. 2008) an additional neuro fuzzy system was implemented to consider the difference in the beam structural collapse behaviour (web and flange yielding, web buckling and web crippling).

A common representation of an artificial neuron (Frochte 2018) is shown in Fig. 2 and can be written as:

$$y(x) = a \cdot \left(\left(\sum W_n \cdot x_n \right) + b \right) \tag{1}$$

It consists mainly of three parameters:

(i) The weights W, which are updated during the training of the DNN model throughout a preset amount of epochs (optimization steps), being also an indicator for the strength

of a connection within a network. A low weight value represents a weak connection, a high value vice versa a strong connection.

(ii) A bias b as an additional trainable nonzero value which is added to the summation of weighted inputs of a neuron

(iii) a represents the activation function, with an inherent predefined threshold used to capture a linear behaviour (Step function, s. Eq.(2)) or more complex nonlinear multiclass or regression problems (sigmoid function s. Eq.(3) or ReLU (Rectified Linear Unit) s. Eq.(4)).

$$\text{Step Function}: f(x) = \begin{Bmatrix} 0, & x < 0 \\ 1, & x \geq 0 \end{Bmatrix} \tag{2}$$

$$\text{Logistic (sigmoid)}: f(x) = \frac{1}{1+e^{-x}} \tag{3}$$

$$\text{ReLU}: f(x) = \max\begin{Bmatrix} 0, & x < 0 \\ x, & x \geq 0 \end{Bmatrix} \tag{4}$$

The optimization process in a neural network uses backpropagation as a technique to update the weights within a training procedure. Therefore, the chain rule is used to calculate the gradients of all nodes within a network. This enables the application of a gradient descent rule, expressed by a learning rate η, which controls the update step size and the derivative of the function within the Nabla-Operator.

$$x^{(j+1)} = x^{(j)} - \eta \cdot \nabla f\left(x^{(j)}\right) \tag{5}$$

The overall estimated accuracy of a neural network is highly dependent on the quality and distribution of the input parameters. In many cases it is therefore necessary to transform or scale these values, using different methods like normalization (s. Eq.(6)) or standardization (s. Eq.(7)) as follows (Frochte 2018):

$$\hat{x}^{(i)} = \frac{x^{(i)} - x^{(i)}_{\min}}{x^{(i)}_{\max} - x^{(i)}_{\min}} \tag{6}$$

$$\tilde{x}^{(i)} = \frac{x^{(i)} - \bar{x}^{(i)}}{\sigma} \tag{7}$$

with:

$\bar{x}^{(i)}$ Mean value of an input feature

σ Standard deviation of an input feature

Data transformation eliminates the major problem of multiple features having different magnitudes, ranges and units by scaling them down. Therefore, data normalization is used to scale the magnitudes of available features between the values of 0 and 1 (or -1 and 1), corresponding to the lowest and highest values. Standardizing the date means rescaling it, while the mean value is set to 0 and the standard deviation to 1. In a lot of engineering applications data standardization shows better performance evaluations, since outliers are taken better into account. A closer look on both concepts and its effects on the learning behaviour is carried out and presented in Section 3 of this paper.

2. Data Development

Every ML model requires data to calculate some target values. Nevertheless, the data quantity, structure, density, appropriate feature set or the data transformation strategy affect the accuracy of the output and is always dependent on the goals and specific demands. The principle applies, however, that poor data quality will lead to inferior results, in general independent on the computational level of the used method. It is therefore necessary to first investigate the data before choosing an appropriate ML model. This procedure is widely known under the term of “feature engineering” and includes different methods, starting with investigations on data quality and distribution followed by unsupervised methods like principle component analysis (PCA) to reduce the feature dimensionality or the evaluation of the feature importance using decision tree based algorithms.

Table 1: Investigated profiles (DIN EN 10210-2 2006, DIN EN 10219-2 2006) and applied parameters

Used Profiles	Number of Sections	Dimension Range c/t
SHS hot rolled	88	8.0 – 47.62
SHS cold-formed	88	8.0 – 47.62
RHS hot rolled	93	9.52 – 56.25
RHS cold-formed	92	12.5 – 55.55
Used Parameters	**Number of Parameters**	**Values**
Steel grade f_y	3	S355, S460, S700
Imperfection amplitude e_0	3	B/200, B/300, B/400

The developed data sets for the SHS and RHS profiles are based on the geometric dimension properties from DIN EN 10210-2 (DIN EN 10210-2 2006) and DIN EN 10219-2 (DIN EN 10219-2 2006) for hot rolled and cold-formed structural hollow sections, respectively. Thus, a total of 361 European profiles, three different steel grades ranging from mild to high-strength (S355, S460 and S700) and three different imperfection amplitudes (B/200, B/300 and B/400) were taken into account, see Tab. 1. This parameters form the basis for further LBA and GMNIA simulations conducted in Abaqus (Abaqus, 2016). The description of the finite element models is summarized within Section 2.1. The subsequent data extraction is discussed and presented in Section 2.2.

2.1 Finite Element Modelling

The developed Abaqus models are making use of isoparametric shell elements with reduced integration of type S4R, with a mesh density of around 60 elements in circumferential and (depending on the total member length) 50 – 100 elements per meter in longitudinal direction. The geometry of the profiles is based on code provisions of (DIN EN 10210-2 2006, DIN EN 10219-2 2006) with a local length L (longitudinal direction) set as the bigger value of ether the width W or the height H of the cross-section. Therefore, the loads and deformations are applied through defined reference points (RF-Points) which are located at the upper and lower edge of the cross-section (s.

Fig. 3). These are connected through multiple point constraints (MPC-Beam formulation) to associated node sets along the upper and lower profile outer edge (s. Fig. 3 a) shown exemplary for the quarter range of the cross-section). This definition implies a rigid connection between the nodes at the extremity and a reference node at the centroid of the respective sections. All boundary conditions were set as fixed, except for the deformation in the longitudinal direction. Throughout this study an elastic-ideal plastic material model was used, with an infinite yield plateau assumed at a stress level $\sigma_{\text{von-Mises}} = f_y$, without an explicit consideration of residual stresses within the Abaqus models. The validation of the Abaqus model is based on an extensive analytical, numerical and experimental campaign, conducted between the years of 2017 and 2019 at the University of Bundeswehr Munich, in the context of the EU-funded (RFCS) project HOLLOSSTAB (Grand Nr. 2015-709892). The reader is referred to the references of the project for further details in (Toffolon et.al. 2019a, Toffolon et.al. 2019b, Toffolon and Taras 2019c, Toffolon and Taras 2019d, Meng et.al. 2019).

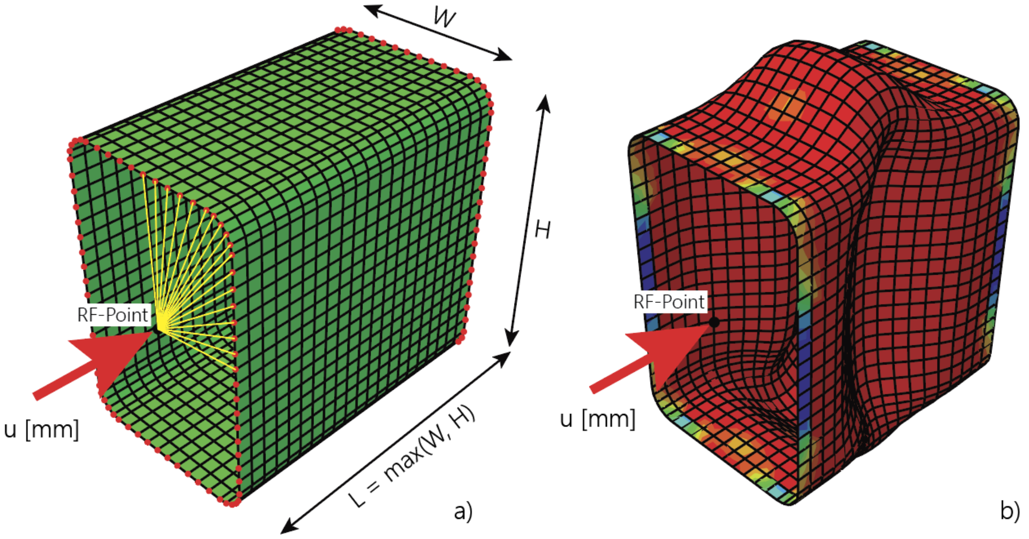


Figure 3: a) Finite element model of an RHS section; b) Deformation in the direction of the longitudinal axis

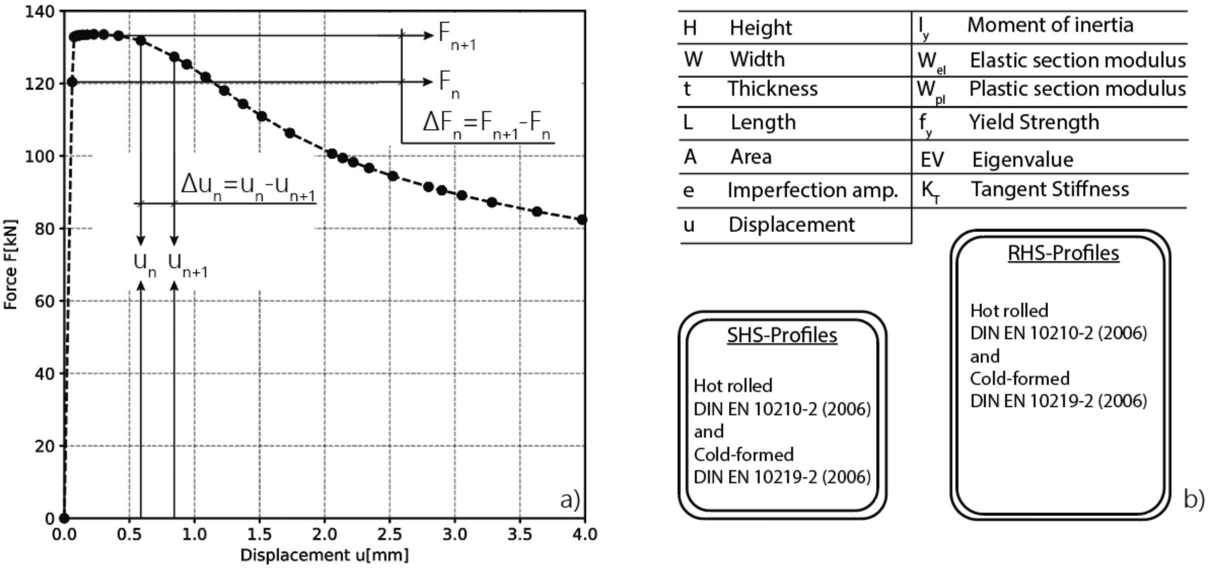
The basic simulation process for the generation of the required data sets is always performed in two steps. In a first step an LBA analysis is carried out in order to identify the elastic critical buckling resistance of the cross-section and the eigenshape as the critical imperfection form. In a second step a GMNIA simulation is performed to determine an elasto-plastic buckling load – the realistic buckling resistance that considers both material and geometric nonlinearities – of the cross-section as well as the courses of the pre- and post-buckling range. The nonlinear calculations in Abaqus were performed using the static general stress analysis.

2.2 Data Extraction

The parameters from Tab.1 lead to a total amount of 361 LBA and 3249 GMNIA simulations conducted in Abaqus. This data basis is subsequently used for the extraction of the input features used for the training of the DNN models. Therefore, the LBA analysis output from Abaqus is used to extract the cross-section dependent elastic critical buckling load. The GMNIA analysis results, on the other hand, are used to determine the incremental deformation steps Δu_n and an associated differential force ΔF_n, see Fig. 4 a). These values are subsequently used to calculate an incremental tangent stiffness $K_{T,n}$, see Eq.(8) for the whole displacement range u of a cross-section in the pre- and post-buckling range.

$$K_{T,n} = \frac{\Delta F_n}{\Delta u_n} \tag{8}$$

This procedure leads to a first estimated amount of 145377 data rows, including all investigated parameters from Tab.1, summarized within the feature representation in Fig. 4 b). Additional information on feature importance, data structure and distribution, as well as their effect on the model behaviour and accuracy (feature engineering) are presented within the following Section 3 in more detail.



H	Height	I_y	Moment of inertia
W	Width	W_{el}	Elastic section modulus
t	Thickness	W_{pl}	Plastic section modulus
L	Length	f_y	Yield Strength
A	Area	EV	Eigenvalue
e	Imperfection amp.	K_T	Tangent Stiffness
u	Displacement		

SHS-Profiles

Hot rolled
DIN EN 10210-2 (2006)
and
Cold-formed
DIN EN 10219-2 (2006)

RHS-Profiles

Hot rolled
DIN EN 10210-2 (2006)
and
Cold-formed
DIN EN 10219-2 (2006)

Figure 4: a) Data extraction from Abaqus simulations, b) Overview of selected features

3. Feature Engineering

The process of data analysis and interpretation is not structurally defined and does not necessarily follow a fixed sequence of operations. Since this process is highly iterative and many parameters are constantly changed, no clear hierarchy can be named within the individual modifications. Therefore, according to the principle of an ascending complexity, the most important adjustments are pointed out and summarized in the following subsections.

3.1 *Data Expansion and Splitting*

Based on own investigations carried out to analyze the data density, structure and its influence on the performance on different DNN models, a clear outcome is obtained and presented in Fig. 4. The left side shows two load-displacement diagrams and the right side an associated frequency distribution of the tangent stiffness. The Abaqus GMNIA simulation is described by the black dotted curve and is considered here as the benchmark resistance. Each dot represents one increment of the numerical simulation. For the purpose of a better representation, the points were connected through interpolated lines. The red dotted line from Fig. 4 a) symbolizes the predicted resistance, calculated with one DNN model by using the raw data set. This means that the displacement u had the same incremental distribution as provided by the Abaqus simulations. A profound difference between the black and the red curve can be obtained, especially in areas of small displacement in the pre-buckling range, i.e. high tangent stiffness. Due to this premature softening behaviour, the maximum load is predicted very inaccurately though. On the other hand, the course obtained in the post-buckling range is more precise, although the overall resistance is predicted too low. Fig. 4 b) shows the frequency distribution of the tangent stiffness of the whole data set, including all investigated cross-sections.

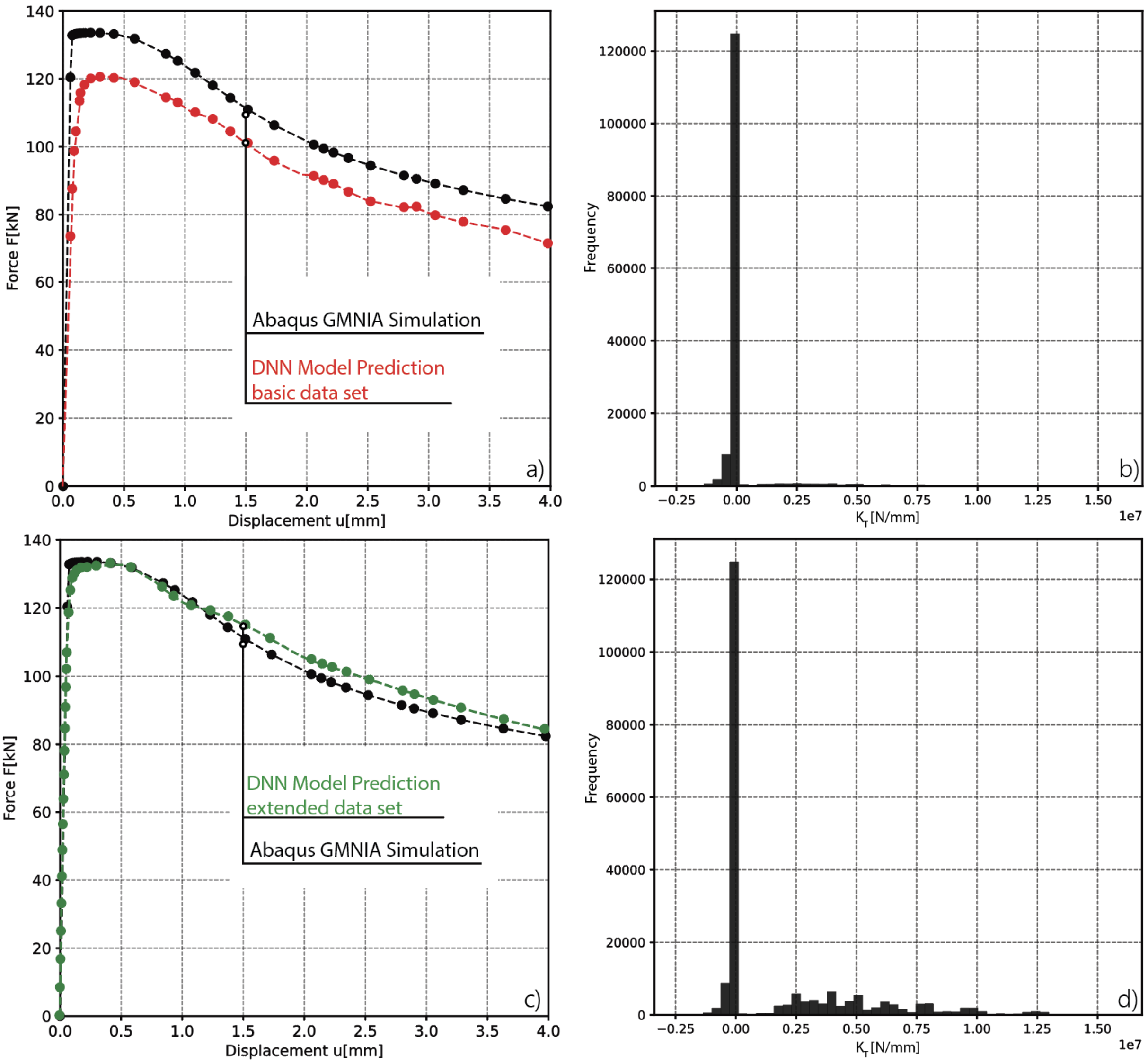


Figure 4: a) Prediction based on raw data set, b) Histogram showing K_T based on raw data set, c) Prediction based on extended data set, d) Histogram showing K_T based on extended data set

The highest frequency is clearly reached around the value of zero. This range is attributed to the area near the maximum force of the load-displacement diagram exemplary represented in Fig. 4 a). The peak can be explained by the smaller increments chosen within the Abaqus simulation to achieve equilibrium in non-linear areas. On the other hand, relatively large incremental steps are made in the elastic range at the beginning of the calculation. Therefore, the different incremental step sizes along the load-displacement curve lead to an uneven distribution of data, resulting in predictions that are more accurate for regions with higher density. For this reason, high values of the tangent stiffness $K_T \approx K_e$, extracted from the start of every Abaqus simulation, are seen as outliers by the DNN models and are therefore difficult to predict precisely. Fig. 4 c) shows an updated prediction, which is more accurate in representing the resistance within the pre-buckling range. An additional data enlargement was carried out by interpolating linear between the increments in the elastic range, i.e. the very first increments. Thus, leading to an artificially increased data density for the tangent stiffness of $K_T >> 0$, as shown in the histogram of Fig. 4 d).

Based on the outcomes of further DNN model performance evaluations, additional adjustments were made with regard to the data structure. It was obtained that a further splitting between hot rolled and cold-formed SHS and RHS profiles and the pre- and post-buckling range led to a significantly better performance. Especially the prediction of the cross-section dependent maximum load F_{max} was met with higher accuracy as further presented in Section 4 of this paper. Subsequently a general data expansion between all increments in the pre- and post-buckling range lead to better accuracies and prediction results. In addition, a deformation limit u_{max} of six times the deformation at the point of reaching the maximum load F_{max} was specified. This assumptions and adjustments lead to a separation of data sets as summarized in Tab. 2.

Table 2: Updated data subsets

Profiles	Data set size (rows)
SHS hot rolled, pre-buckling	97394
SHS hot rolled, post-buckling	177721
RHS hot rolled, pre-buckling	105811
RHS hot rolled, post-buckling	201969
SHS cold-formed, pre-buckling	103360
SHS cold-formed, post-buckling	183217
RHS cold-formed, pre-buckling	86792
RHS cold-formed, post-buckling	207561
Total amount:	1163825

3.2 *Data Transformation*

The performance of a neural network (gradient descent solver) is highly dependent on the dimension of the input parameters, its magnitude and distribution. It is therefore necessary, but always dependent on the data structure, to transform and scale the input parameters. This can be done for example by normalization (s. Eq.(6)) or standardization (s. Eq.(7)) (Frochte 2018) in general cases. Fig. 5 shows the difference between the performances of several DNN models based on normalized and standardized inputs. In each case six randomly chosen feature combinations were taken into account. These will not be commented on further, as they serve for demonstration purpose only. The x-axis represents the number of epochs. These are in a general sense the number of repetitions in the training procedure. The y-axis describes the network accuracy using the measure r-squared (s. Eq.(9)), a statistical measure of the unexplained and the total variation.

$$r^2 = 1 - \frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n} \left(y_i - \overline{y}_i\right)^2} \tag{9}$$

with:

$y_i, \hat{y}_i, \overline{y}_i$ Actual output, predicted output, standard deviation

The performance of DNN models with standardized feature combinations are summarized in Fig. 5 a). First, it can be noted that the r^2 value increases steeply straight from the beginning of the optimization process. Already after less than 50 epochs the value of $r^2 = 0.90$ is exceeded. After approximately 200 epochs, even the value of $r^2 = 0.95$ is passed for most feature combinations. On the other hand, Fig. 5 b) shows the DNN model performance using normalized data sets. A direct comparison shows that the overall model accuracy is lower. Due to a flatter course between the first 200 epochs from the beginning, an $r^2 = 0.95$ can only be reached by the end of the optimization process around 1500 epochs. This difference underlines the choice of data standardization within further investigations and results presented in Section 4.

An additional influence on the model performance is attributed to the chosen format of the predicted value within the output layer. Depending on its distribution within the data set, the accuracy can vary greatly within the predicted values. The frequency distribution of the raw values of the tangent stiffness K_T, for hot rolled SHS profiles in the pre-buckling range, is shown in Fig. 6 a).

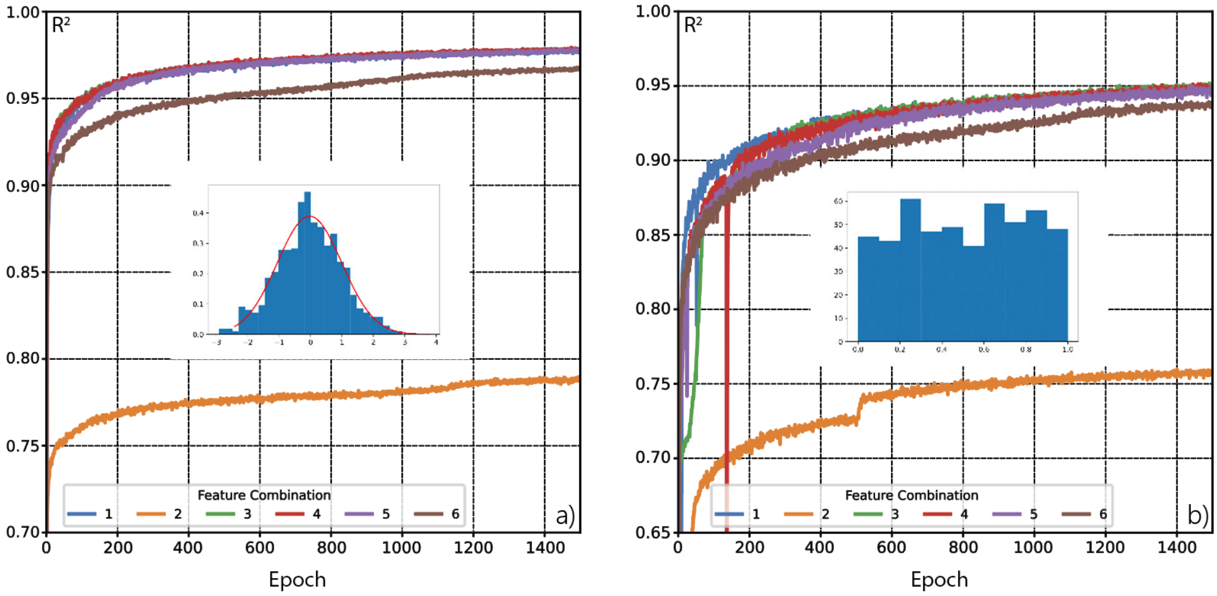


Figure 5: Training results using a) standardized features; b) normalized features

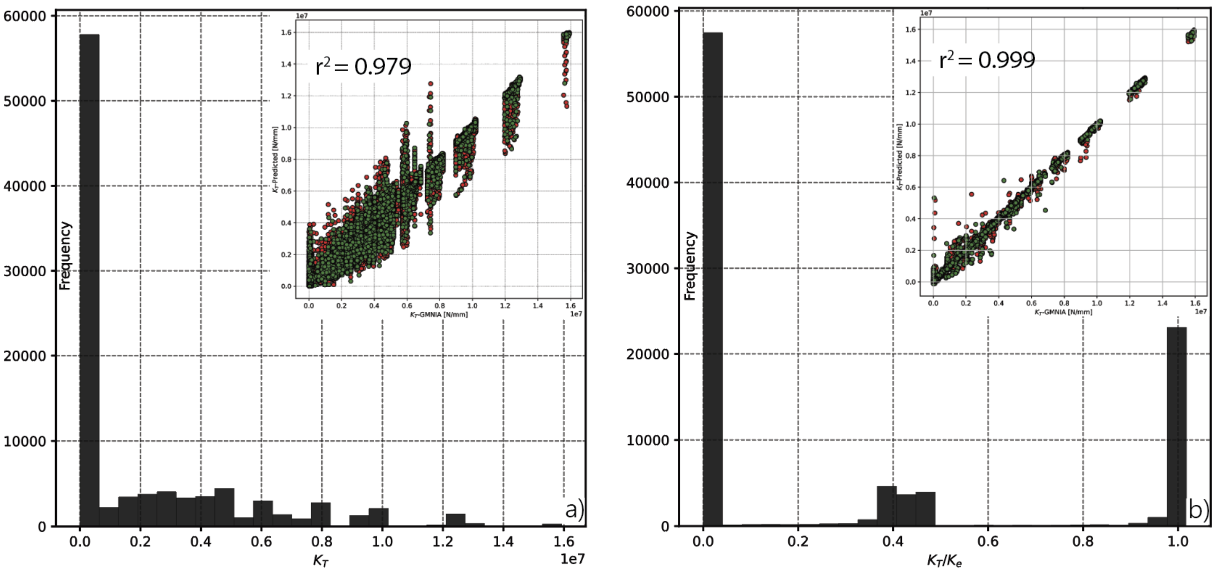


Figure 6: Output distribution for SHS hot rolled profiles a) raw values of the tangent stiffness K_T; b) transformed values of the tangent stiffness divided by the elastic stiffness K_T/K_e

As already mentioned briefly in Section 3.1, values concentrated around zero represent the tangent stiffness ($K_T << K_e$) near the maximum reached resistance. However, all other values can be a combination of $K_T \approx K_e$ or $K_T < K_e$, attributed to the elastic initial range or an intermediate range with a propagating nonlinear behaviour. This mixed tangent stiffnesses on the one side and the big difference in its magnitude (ranging between 0 and 1e7) on the other side for the direct prediction of K_T (s. Fig. 6 a)), significantly affect the optimization process of the DNN model. Thus, in the investigations carried out, an $r^2 = 0.95$ was reached after approximately 1000 and an $r^2 = 0.98$ after approximately 6000 epochs. Fig. 6 b) shows the frequency histogram of the tangent stiffness K_T divided by the elastic stiffness K_e. The resulting distribution in this case is completely different from before, being clustered in mainly three parts due to the transformation. Values around zero are still attributed to the area near the maximum resistance. However, the cross-section dependent tangent stiffness in the elastic range ($K_T \approx K_e$) is now located close to the value of one. All other values of $K_T < K_e$ are in between these boundary ranges and are assigned to an increasing nonlinear local behaviour. The prediction accuracy in this case is significantly higher. Thus an $r^2 = 0.99$ is already reached after approximately 200 epochs using the same DNN model architecture. Although the difference between the estimated accuracies is less than 2%, a much higher scatter is obtained within the predicted stiffnesses using the raw output data in Fig. 6 a). It must also be pointed out that the optimization process, measured in terms of the required epochs and the estimated associated model accuracy, was slower compared to the performance of DNN models based on the transformed data from Fig. 6 b).

3.3 Feature Importance

ML and DL methods offer several approaches to detect features that have the most influence on the learning process. This has the advantage that input parameters that have no significant impact on the predictions are eliminated. Therefore, the calculation effort is reduced while the optimization process depends on fewer parameters. The most common way to seek for these parameters is the application of tree based algorithms. Decision trees are based on the logical structure of a tree, predicting the values through a sum of individual choices. Starting with the input values in a so called root, the data is split within different possible decisions (categorical decision for classification and quantitative decisions for regression problems). Following this principle to the end nodes of the model, a condition or value is proposed. This procedure is highly powerful but prone to problems connected to data quality and tree complexity leading to an effect called overfitting. It occurs when the algorithm starts to capture noise, being too accurate on the training data due to a very deep tree structure. Therefore, the general approach goes lost, which leads to poor predictions on the base of unseen test data.

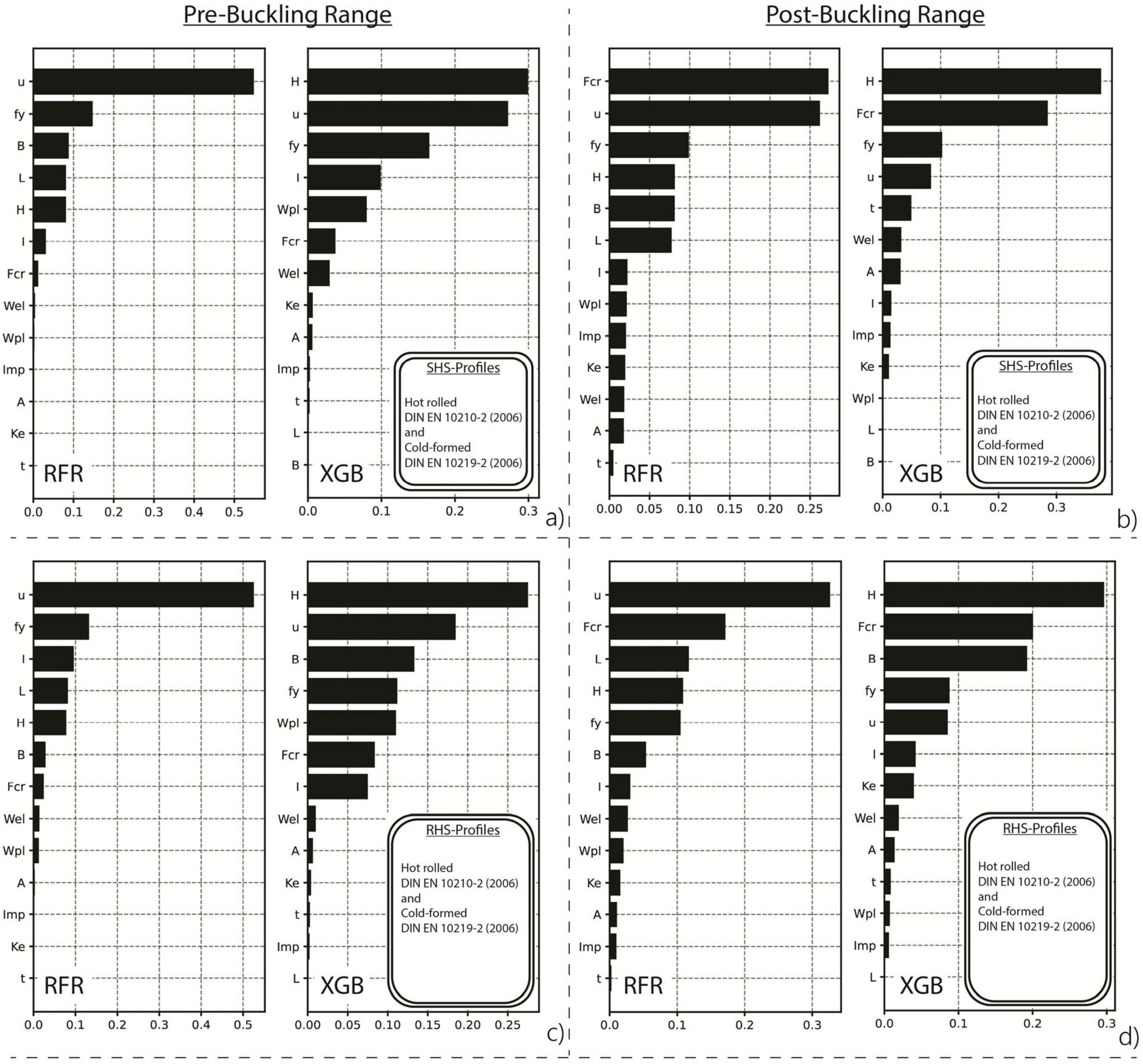


Figure 7: Evaluation of feature importance and comparison between the Random Forest model and the XGBoost model a) pre-buckling range of SHS profiles; b) post-buckling range of SHS profiles; c) pre-buckling range of RHS profiles; d) post-buckling range of RHS profiles

For this reason, two methodologies (Random Forest Regressor (Breiman 2001) and XGBoost Regressor (Chen and Guestrin 2016)) were used in the following, which proved to be successful in the case of the present data sets. The random forest algorithm is based on the general idea of a decision tree with the main difference of using not only one big tree to represent the problem, but rather a high number of simpler trees which are more general and therefore not as much affected by overfitting. The whole dataset is first subsampled (bootstrap sampling), meaning that a random number of features and data rows is used to build smaller data sets. This leads to a higher variety and a different data density. In the following, individual trees are build parallel leading to a large number of different predictions, which are gathered and averaged (bootstrap aggregating). In contrast, the XGBoost algorithm builds the trees not parallel to each other but rather uses a boosting technique. Therefore, the trees (weak learners) are built sequentially so that each new tree corrects the error of the previous one (boosting). Both algorithms were used to evaluate and plot the importance of the features of the SHS and RHS profiles in the pre- and post-buckling range, see Fig. 7 and Tab. 3. The abbreviation RFR is attributed to the Random Forest algorithm, while XGB is attributed to the XGBoost algorithm.

The output of the feature importance is exemplary explained by Fig. 7 a), using the data set of the SHS profiles (hot rolled and cold-formed) in the pre-buckling range. All features are sorted according to their importance and displayed on the y-axis. The importance itself is shown along the x-axis with the mean decrease impurity (also known as Gini index) computed for the Random Forest (RFR) and XGBoost structure (XGB). It is calculated as the total reduction of the impurity within a split, made by one selected feature on average over all trees within the forest. Apart from their order, the three most important features are initially the same, represented by the displacement u, the steel grade f_y and the height H or width B. In terms of SHS profiles the values for B, H and L are identical. This also explains one main difference between these methods based on their structure. Since the Random Forest algorithm uses many individual trees, linear dependent features are not necessarily eliminated, as they are used independently within reduced data sets. For this reason, features B, H and L are listed with approximately the same importance (Fig. 7 a) RFR). On the other hand, XGBoost (Fig. 7 a) XGB) uses one sequential build of trees. Therefore, equal features are eliminated throughout the process.

Table 3: Estimated feature combinations

	Pre-buckling data sets	Post-buckling data sets
SHS hot rolled and cold-formed	H, W_{el}, W_{pl}, I_y, f_y, u, F_{cr}	H, t, W_{el}, A, I_y, Imp, f_y, u, F_{cr}
RHS hot rolled and cold-formed	H, B, W_{pl}, I_y, f_y, u, F_{cr}	H, B, W_{el}, I_y, Imp, f_y, u, K_e, F_{cr}

Subsequently, the procedure of feature engineering requires the use of different methods, since the output can vary strongly and only the sum of reasonable results lead to a general tendency for the choice of important features and their combination. Furthermore, additional combinations of the selected features within the DNN models should be tested to determine differences in the performance and overall accuracy. Tab. 3 summarizes the chosen feature variations based on the output of Fig. 7 and additional accuracy evaluations from DNN models.

4. DNN Model Development and Results

Table 4: Estimated hyperparameters

Model Parameters	Selection
Hidden layer **1** (neurons)	27
Hidden layer **2** (neurons)	27
Hidden layer **3** (neurons)	18
Hidden layer **4** (neurons)	9
Activation function	ReLU
Optimizer	Adam
Learning rate	0.0005

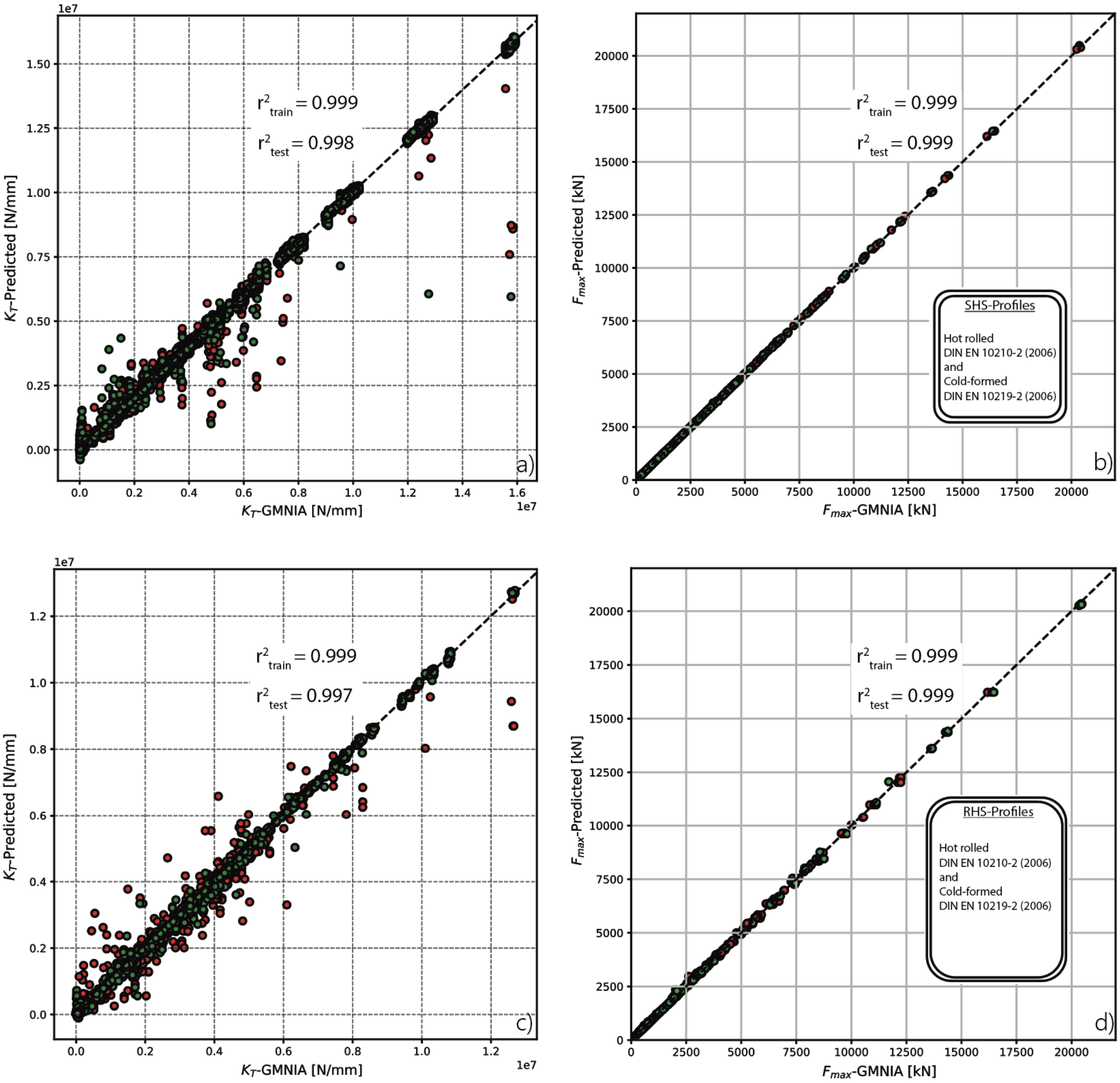


Figure 8: a) Simulation ($K_{T\text{-GMNIA}}$) vs .prediction ($K_{T\text{-Predicted}}$) for SHS profiles; b) Simulation ($F_{max\text{-GMNIA}}$) vs .prediction ($F_{max\text{-Predicted}}$) for SHS profiles; c) Simulation ($K_{T\text{-GMNIA}}$) vs .prediction ($K_{T\text{-Predicted}}$) for RHS profiles; d) Simulation ($F_{max\text{-GMNIA}}$) vs .prediction ($F_{max\text{-Predicted}}$) for RHS profiles

In general two types of parameters are distinguished within a deep neural network (feed forward network). The trainable parameters, which are changed throughout the optimization process, including the weights and the biases. The preset hyperparameters, which do not change throughout the optimization process. This includes the general model architecture (amount of neurons and hidden layer), the chosen optimizer, an associated learning rate and the activation functions within the hidden layer, the batch size and additional regularization techniques. Without appropriate information on the DNN model performance, the computational effort needed for the estimation of possible hyperparameters is immense. For this reason, the problem was considered systematically by first exploring the parameters in a coarse pattern to come up with possible, still rough ranges of values. Consequently, these values were adjusted in finer intervals until suitable values were found. Therefore, a total of 193 individual combination was tested within the framework of preliminary investigations using the Random Search Method. Random Search is in terms of ML one method to estimate decisive parameters, which can be suitable but do not necessary have to, since not all possible combinations are taken into account. The opposite would be the Grid Search Method were each parameter combination is tested. However, as there is not only one specific solution but rather a potential solution space of possible combinations of hyperparameters, this workflow is suitable in order to detect the overall tendencies within the DNN architecture. All calculations were performed on the basis of a train and test philosophy, meaning that a specific data amount was used for the training (80%) and an additional independent amount for the validation process (20%). This procedure is crucial to detects problems like overfitting and the interpretation of the overall behaviour, assessing transferability to “unseen” data. Tab. 4 summarizes the selected hyperparameters from these initial investigations.

First results of the accuracy of the estimated DNN models are presented within Fig. 8, where the x-axis is representing the simulated values and the y-axis the predicted values. Fig. 8 a) and c) show exemplarily the overall prediction of the tangent stiffness K_T for hot rolled and cold formed SHS as well as RHS profiles in the pre-buckling range. This values are used as initial inputs for the calculation of a force accumulated over the deformation (s. Eq.(10)):

$$F_n = F_{n-1} + K_{T,n} \cdot \Delta u_n \tag{10}$$

Despite the high r^2 values shown in Fig. 8 a) and c), a certain scatter is present which might imply inaccuracies in the calculation of the forces (s. Eq(10)). Therefore, the load-displacement curves of the considered SHS and RHS profiles were calculated within the pre-buckling range by using the trained DNN models. Fig. 8 b) and d) show a summary of these results by isolating the maximum load F_{max}, which is calculated from an accumulation including all predicted values of the tangent stiffness from the steps before. Thus, it was assumed that the overall error is most present by the end of every cross-section dependent load-displacement curve. It can be seen that the outliers has in general no influence on the estimated maximum force F_{max}. The associated r^2 value lies in both cases around 0.999 for training and testing data. This can be attributed to the fact that the predicted values of K_T, originating from the data development using Eq(8), are based on the incremental distribution from Abaqus simulations. The data sets were additionally expanded further (s. Tab. 2) reducing the incremental step size. This results in the fact that deviations within the tangent stiffness lead to very small differences in the accumulated force, as long as the step size remains small. Nevertheless, this initial outcome underlines specifically two conclusions. The data structure can be processed by the model well enough and used for predictions i.e. a predictable problem suited for ML applications. The trained DNN model is "general" enough and does not tend to overfitting problems meaning that $r^2_{train} \approx r^2_{test}$.

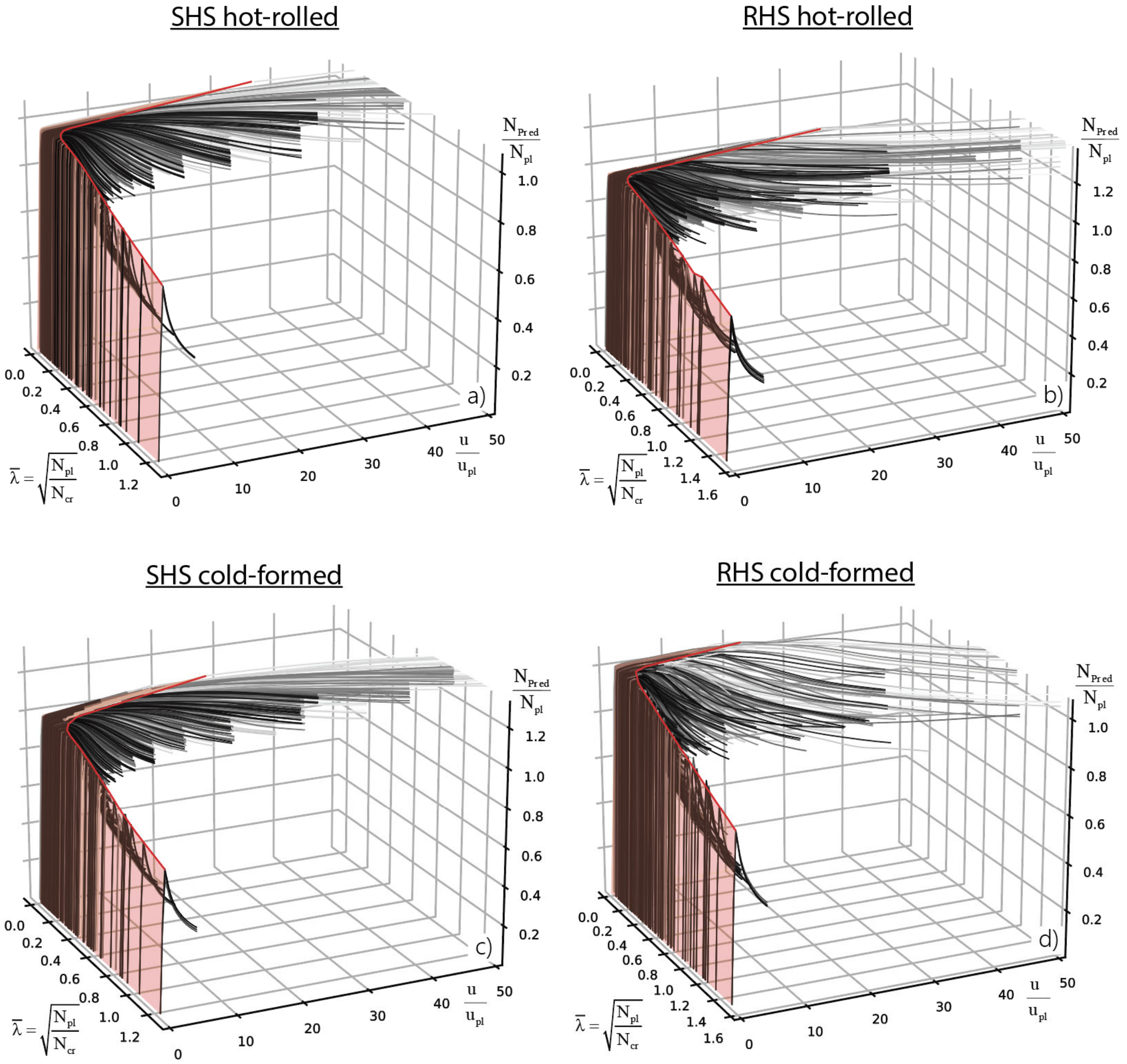


Figure 9: Predicted pre- and post-buckling range for a) hot-rolled SHS profiles; b) hot-rolled RHS profiles; c) cold-formed SHS profiles; d) cold-formed RHS profiles

Fig. 9 shows additionally the prediction of 792 hot-formed SHS, 792 cold-formed SHS, 837 hot-formed RHS and 828 cold-formed RHS profiles. A normalized representation was chosen within the axes. This definition has the considerable advantage that all curves are displayed and compared in a scale independent manner. Therefore, the pre- and post-buckling range is plotted along its local slenderness, the predicted force N_{Pred} divided by the plastic cross-section capacity N_{pl} and the corresponding displacement u divided by u_{pl}, the theoretical plastic deformation. A solid red line along the reached maximum loads split the ascending from the descending branch. Thus, Fig. 9 illustrates in a first step (by taking into account only the longitudinal deformation) that the pre- and post-buckling range of various SHS as well as RHS profiles of different slenderness, steel grade and manufacturing process can be predicted with high accuracy. Subsequently, this three-dimensional information forms the basis for the implementation into a DSM based beam formulation. Further research on the implementation of local redistribution capacities within the beam finite element formulation is under way.

4. Conclusions and Outlook

The presented paper describes first findings and results towards a novel method (DNN-DSM), combining the computational advantages of beam-element models with the accuracy of numerical shell-element based simulations. The connection is made by using predictive models based on the concepts and techniques from machine and deep learning, trained and tested on data sets derived from a pool of Abaqus shell element simulations. For this reason, this paper first introduces the finite element modeling and the subsequent extraction and compilation of the required data sets. Basic principles of feature engineering are presented and applied to the presented data sets. Since a large part of the actual work lies in the preparation (post processing) of the data, the focus within this paper is placed on this in particular. For this purpose, three processing steps, the data expansion, the data transformation and the concepts of feature importance are presented and their impact on the DNN model performance is described. Subsequently, further steps towards the development of appropriate DNN models, including hyperparameter tuning, are described and summarized. Finally, the accuracy of the DNN models is demonstrated by first the prediction of the main output parameter K_T (tangent stiffness) and second by the resulting forces (s. Eq.(10)) calculated from it. In both cases very precise results are achieved, reaching values of r^2_{train} and r^2_{test} of 0.99 and 0.98 for training and testing data sets, respectively (s. Fig. 8). Fig. 9 underlines subsequently the potential of this approximated solution methods, predicting all investigated 3249 hot rolled and cold-formed SHS and RHS profiles. However, this computational procedure and implementation is under development it will be shown on a larger scale in future publications.

The focus of further research is laid on the implementation of current results into the beam finite element (DSM) formulation, being able to predict the nonlinear behaviour and possible slenderness dependent redistributions within a truss structure, subjected exclusively to normal forces. The validation of these calculations is to be carried out with the help of developed benchmark models from own investigations and literature. In a next step the data sets will be further developed to account for additional transverse deformations v_z and rotations φ_y, enabling the prediction of a full beam structure. This would enhance the full range of deformation and rotation combinations within a beam formulated DSM approach. Additional feature engineering approaches need to be investigated to narrow the used features and reduce the overall dimensionality of the data using unsupervised approaches like principle component analysis (PCA: a method to remove redundant or highly correlated features) or autoencoder (an unsupervised artificial neural network that compresses and reconstructs the data in terms of dimensionality reduction).

References

Abaqus. Refernce manual, version 6.16. (2016). *Simulia, Dassault Systéms*. France.

AISC 360-16. (2016). “Specification for structural steel building.” *American Institute of Steel Constructions,* Chicago.

Breiman, L. (2001). “Random Forest.” *Machine Learning,* 45, 5-32.

Chen, T., Guestrin, C. (2016). “XGBoost: A scalable tree boosting system.” *In: Proc ACM SIGKDD Int Conf Knowl Discov Data Min*, 785-94.

DIN EN 1993-1-1. (2006). “Eurocode 3. Design of steel structures – Part 1-1: General rules and rules for buildings.” *CEN – European Committee for Standardization,* Brussels.

DIN EN 1993-1-5. (2006). “Eurocode 3. Design of steel structures – Part 1-5: Plated structural elements.” *CEN – European Committee for Standardization*, Brussels.

DIN EN 10210-2. German version EN 10210-2: (2006). “Hot finished structural hollow sections of non-alloy and fine grain steels – Part 2: Tolerance, dimensions and sectional properties.”

DIN EN 10219-2. German version EN 10219-2: (2006). “Cold formed welded structural hollow sections of non-alloy and fine grain steels – Part 2: Tolerances, dimensions and sectional properties.”

D´Aniello, M., Güneyisi, E. M., Landolfo, R., Mermerdaş, K. (2014). “Analytical prediction of available rotation capacity of cold-formed rectangular and square hollow section beams.” *Thin-Walled Structures,* 77, 141-152.

Fieber, A.C. (2019). “Structural Steel Design using Advanced Analysis with Strain Limits.” *Ph.D. Thesis, Department of Civil and Environmental Engineering,* Imperial College London.

Fonseca E.T., Vellasco M.M.B.R., Vellasco P.C.G. da S., de Andrade S.A.L., Pacheco M.A.C. (2001). “A neural network system for patch load prediction.” *J Intell Robot Syst*,31(1/3),185-200.

Fonseca ET, Vellasco PCGdaS, de Andrade SAL, Vellasco MMBR. (2003a). “A patch load parametric analysis using neural networks.” *J Const Steel Res,* 59(2), 251-267

Fonseca ET, Vellasco PCGdaS, de Andrade SAL, Vellasco MMBR. (2003b). “Neural network evaluation of steel beam patch load capacity.” *Adv Eng Software*, 34(11-12), 763-772.

Fonseca, E.T., Vellasco P.C.G. da S., Vellasco M.M.B.R., de Andrade S.A.L. (2008). “A neuro-fuzzy evaluation of steel beams patch load behaviour.” Advances in Engineering Software, 39, Issue 7, 558-572.

Frochte, J. (2018). „Maschinelles Lernen. Grundlagen und Algorithmen in Python.“ Carl Hanser Verlag München.

Gardner, L. (2008). “The continuous strength method.” *Proceedings of the Institution of Civil Engineers – Structures and Buildings*, 161(3), 127-133.

Güneyisi, E. M., D´Aniello, M., Landolfo, R., Mermerdaş, K. (2014). “Prediction of the flexural overstrength factor for steel beams using artificial neural networks.” *Steel and Composite Structures,* Vol.17, No.3, 215-236.

Hebb, D. (1949). “The organization of behaviour.” Wiley, New York.

McCulloch, W. S., Pitts, W. (1943). “A logical calculus of the ideas immanent in nervous activity.” *The bulletin of mathematical biophysics 5*, Nr. 4, 115-133.

Meng, X., Toffolon, A., Gardner, L., Taras, A. (2019). “The generalised slenderness-based resistance method for the design of CHS and EHS.” *Steel Construction,* Vol. 12, Issue 4, Wiley.

Minsky, M. L., Seymour, P. A. (1969). “Perceptrons. An Introduction to Computational Geometry.” Massachusetts Institute of Technology.

Müller, A., Taras A. (2019). “On the post-buckling rotational capacity of squared hollow sections in uniform bending. An initial study on the impact of initial imperfections on deformation paths.” *Steel Construction – Design and Research 12,* No. 3 (8).

Schard, R. (1989). „Verallgemeinerte Technische Biegetheorie.“ Springer-Verlag, Berlin.

Silvestre, N. (2005). “Generalized Beam Theory: New Formulations, Numerical Implementation and Applications.” *Ph.D. Thesis,* IST-Technical University of Lisbon, Portugal.

Toffolon, A., Müller, A., Nico, I. Taras, A. (2019a). “Experimental and numerical analysis of the local and interactive buckling behaviour of hollow sections.” *Proceedings of SDSS 2019,* Prague.

Toffolon A., Meng X., Taras A., Gardner L. (2019b). “The generalised slenderness-based resistance method for the design of SHS and RHS.” *Steel Construction,* Vol. 12, Issue 4, Wiley.

Toffolon, A., Taras, A. (2019c). “Development of an OIC-Type local buckling design approach for cold-formed unstiffened and groove-stiffened hollow sections.” *Thin-Walled Structures,* 144 (6) Elsevier.

Toffolon A., Taras A. (2019d) “Proposal of a design curve for the overall resistance of cold formed RHS and SHS members”, *Conference Paper, Nordic Steel Conference 2019,* Copenhagen.

Walport, F. (2019). “Design of steel and stainless Steel Structures by Advanced Inelastic Analysis.” *Ph.D. Thesis, Department of Civil and Environmental Engineering*, Imperial College London.

Yun, X., Gardner, L. (2017). “Stress-Strain curves for hot-rolled steels.” *Journal of Constructional Steel Research* 133 (6) 36-46.

Yun, X., Gardner, L. (2018). “Description of stress-strain curves for cold-formed steels.” *Construction and Building Material* 189 (11) 527-538.