Neural network representation of mechanical fasteners in large-scale analyses

Victor André[1], David Morin[1], Miguel Costas[1], Magnus Langseth[1]

[1] Structural Impact Laboratory (SIMLab) and Centre for Advanced Structural Analysis (CASA), Department of Structural Engineering, Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway

1 Abstract

This paper presents an artificial neural network (NN) modeling approach for representing mechanical fasteners in large-scale finite element crash simulations for explicit analysis using LS-DYNA version R9.3.1. The NN-model is established to describe the local force-deformation response of point-connectors in automotive applications like self-piercing-rivets and flow-drill-screws. The behaviour from initial loading until failure or unloading is covered. Various architectures and complexities of feedforward NNs were evaluated and trained based on synthetic experiments generated from the constraint model proposed by Hanssen et al. [1]. The constraint model is available as ****CONSTRAINED_SPR2*** *but was used in form of a cohesive element (8-noded, 4-point cohesive element with offsets for use with shells).*

The NN consists of three hidden layers each having 100 nodes. Weights and bias where trained using the Python/Keras package on the basis of synthetic stretch paths calculated by the constraint model. The NN is represented by a sequence of matrix multiplication inside a ****MAT_USER_DEFINED_MATERIAL_MODELS*** *where the joints force-components and damage are computed each time step.*

This forms a proof of concept for implementing such a machine learning modeling technique not based on physics-motivated equations. Both the impact of network complexity and training data diversity was investigated. The numerical results are compared to physical tests and the ****CONSTRAINED_SPR2****-model for five different joint configurations including self-piercing rivets and flow-drill screws. Experimental data was obtained by Sønstabø et. al [2, 3, 4, 5]. All joint configurations show variation in their loading and failure behaviour which gives a wide selection for validating a NN-design that fits all applications. It can be shown that a rather basic machine learning technique like a feedforward NN is able to reproduce path-dependent force-deformation relation for the application in the explicit* LS-DYNA *solver.*

2 Introduction

Structural joining of parts made from different materials requires appropriate techniques and fastener types. The mechanical behaviour of a joint in terms of stiffness, ductility and failure is unique for different material combinations and part geometries. Modelling the failure behaviour of large joined structures requires the model for each individual joint to capture complex failure modes and sequences. There exist macroscopic models for representing a joint in large-scale analysis using shell meshes. Nevertheless, these models are usually designed for a specific joint type and even with a larger set of parameters and an optimal fit, the shape of a force-displacement function is pre-defined. To find a model, which would fit multiple joint types, numerical techniques like NNs could be applied to gain the needed flexibility and degree of freedom.

While a phenomenological model only needs a few physical tests to be fitted, the creation of a database for a NN based model is more tedious since the joint must be challenged in various loading and failure modes. Detailed FEM models of a joint could be used to generate a large dataset of loading-scenarios and could, with the help of a NN, be translated into large scale models. That approach would allow a more flexible joint model which would almost copy the force-displacement behavior of a joint without deviation due to the unsharpness of a phenomenological model. Also, the high order of a NN function would give the flexibility to model various joint behaviours if there can be enough loading-scenarios obtained to fit the problem.

As a proof of concept for using a NN model inside an explicit FE-solver, various NN designs were evaluated and a reasonable one was implemented as a user-subroutine in LS-DYNA. The training data for the NN model was generated synthetically with the constraint model proposed by Hanssen et. al [1].

3 Methodology

3.1 Constraint model introduction

For large scale analysis, the joined sheets are represented by master and slave nodes. Those nodes are tied to a cohesive element, sandwiched between both sheet meshes as seen in Fig.1:. The constraint model proposed by Hanssen et. al. [1] computes the joint force from the stretch of the cohesive element, decomposed in a normal and tangential component. The joint force is computed as normal and tangential component and is sent back to the cohesive element as tractions. A damage variable is calculated as function of loading angle and element stretch. The governing equations can also be found in ***CONSTRAINED_SPR2**, which is the LS-DYNA implementation of the model proposed by Hanssen.

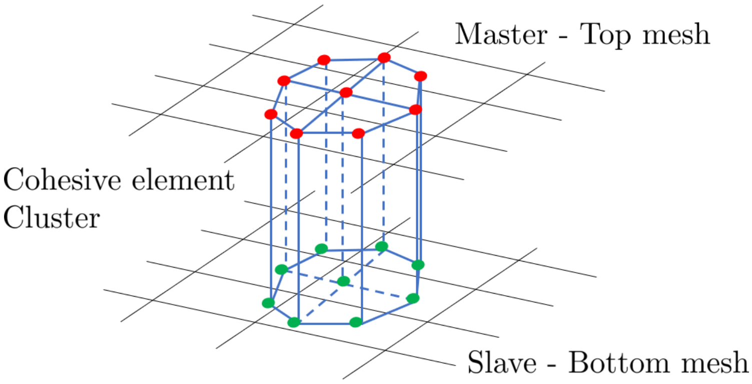

Fig.1: Large scale mesh with cluster of four cohesive elements representing joint

3.2 Neural network modelling

The joint model will be represented by a feedforward NN. It takes four input variables, which are the normal and tangential cohesive element stretch δ_n, δ_t and a respective history measure of the stretches $\bar{\delta}_n, \bar{\delta}_t$. The NN predicts both normal and tangential tractions f_n, f_t as well as a damage variable η. The history measures of both stretch components are realised as running mean and are needed to increase the NNs dimensionality to be able to model path dependent behaviour as seen during loading/unloading. The NN architecture is displayed in Fig.2:. The number of nodes in the input and output layer is fixed given by the present problem. However, the number of hidden layers and their hidden nodes will be changed and the effect investigated. There will be no more than four hidden layers and not more than 500 nodes per hidden layer.

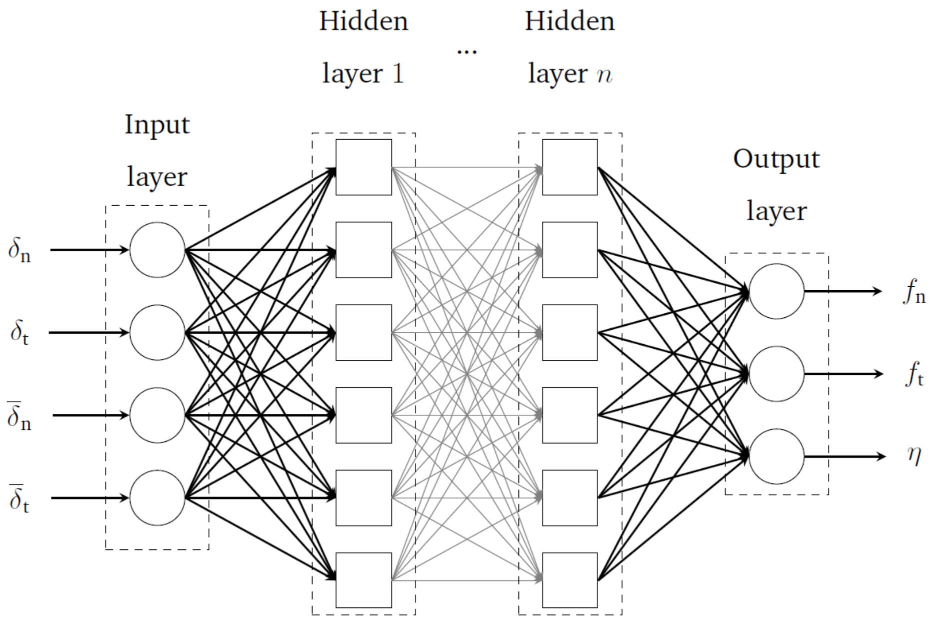

Fig.2: Fully connected feedforward NN; four input variables (stretches, stretch histories) and three output variables (forces/tractions, damage)

All hidden layers are activated with the LeakyReLU (Leaky rectified linear unit) function whereas the output layer applies the ReLU function to give strictly positive results. The joint model only calculates tractions for positive stretches while compression of the joint area is constrained by the sheet contact formulation. The Nadam (Nesterov-accelerated Adaptive Moment Estimation) gradient method was used for training with the minimization goal of the mean squared error (MSE) function. Network training was performed with Python/Keras and the optimized weight and bias values were saved for later implementation in the user-subroutine. The network was trained on scaled input and target values to have similar magnitude of gradients during optimization not to exclude an output variable from training.

Besides the NN architecture, the quality and amount of training data passed to the network is vital to achieve a reasonable problem fit. Various sequences of stretch paths and their histories were created and the corresponding joint force- and damage-response was calculated by the model proposed by Hanssen. The training scenarios consisted of proportional loading (loading a joint under constant angle) until failure, partial loading and following unloading as well as loading with a change of loading angle. A set with more complex loading sequences was kept aside from training for later NN validation.

3.3 NN training results

To find a reliable NN architecture, which would replicate the joint behaviour, a full factorial design with different combination of hidden layer and hidden node number was executed. Each designs performance is shown with the final achieved MSE for training and validation in Fig.3:.

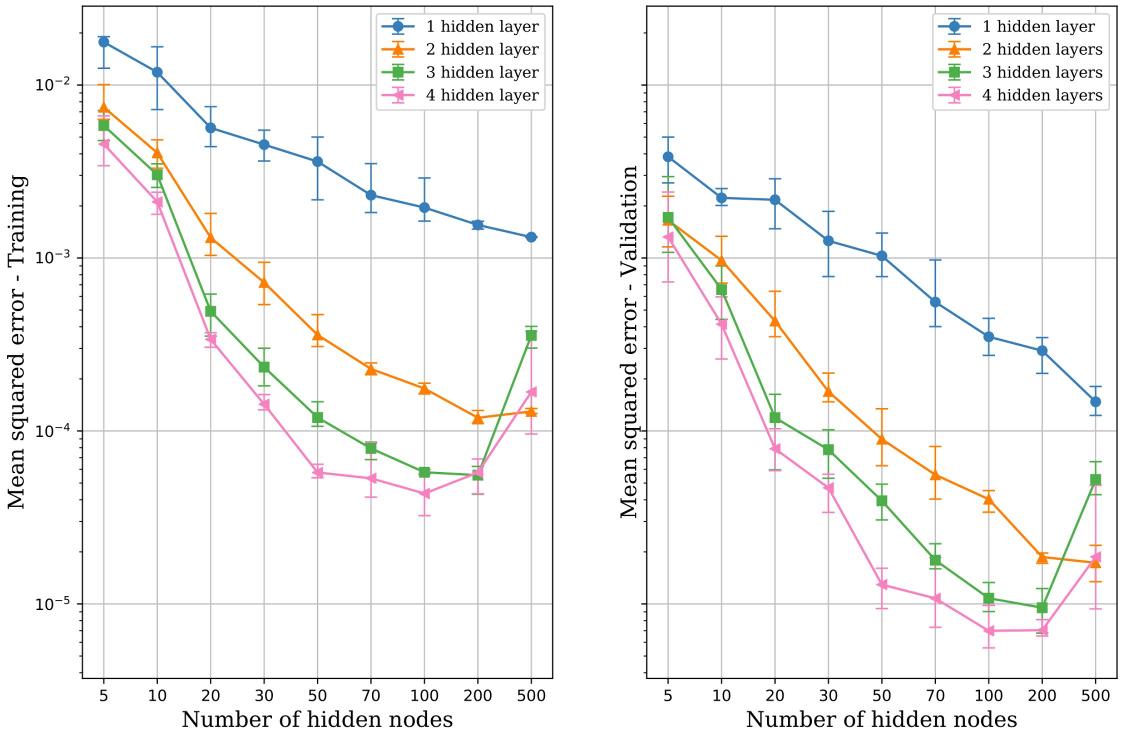

Fig.3: Final training (left) and validation (right) MSE: mean, lowest and highest achieved value for five randomly initialised trainings

The trained proportional loading predictions for a selection of different complex NN designs are shown in Fig.4:. The contour plot shows different levels [0.7, 0.8, 0.9, 0.95, 1.0=failure] of joint damage and the achieved final MSE for the whole training set. From Fig.3: and Fig.4: it can be seen, that a design with three hidden layers each containing 100 nodes, gives a good training error and performs well under proportional loading. That design was kept constant for the following simulations and user-subroutine implementation. A NN of this size contains 21.003 trainable parameters (weights + bias), which need to be trained and imported to the subroutine.

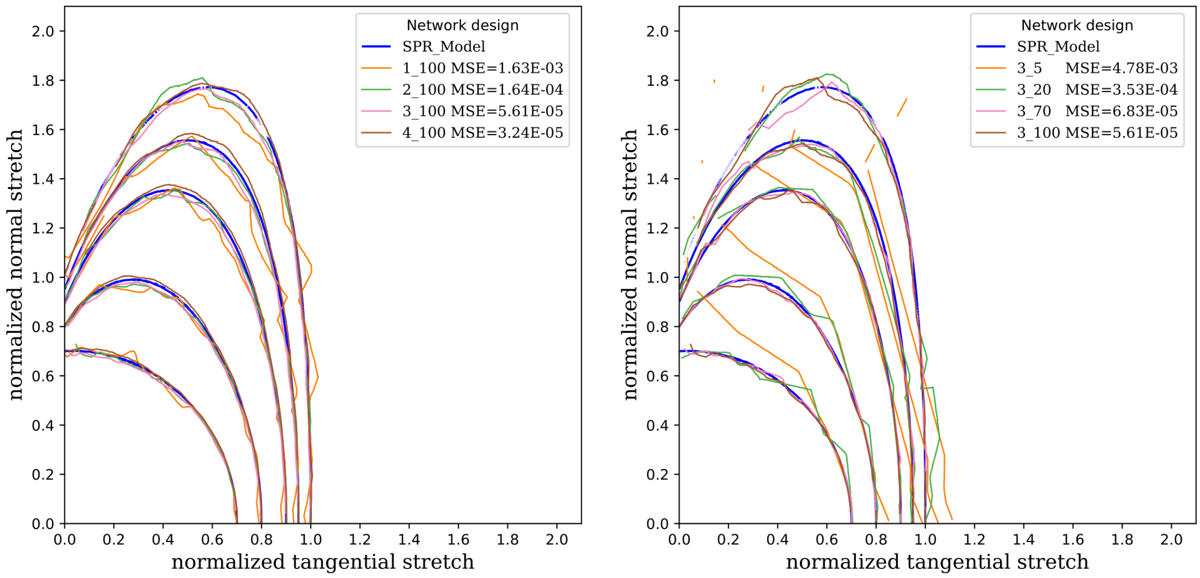

Fig.4: Contour plots with different levels of damage: 100 nodes and one to four hidden layers (left), three hidden layers and 5, 20, 70, 100 nodes (right)

4 Simulation and experimental comparison

Both the ***CONSTRAINED_SPR2**-model (realised as cohesive element) and it's NN representation were challenged in single-unit joint tests and a component test. The single-unit tests consist of two 40 by 120 mm rectangular shaped sheets, overlapped as cross and joined in the center (crosstest). The component test is made from a U-shaped channel covered with a flat plat and joined with total 14 flow-drill-screws along the seam. The resulting crashbox structure was loaded axially. The plates were modeled by a shell mesh using Belytshko-Tsay elements with reduced integration. They had a quadrilateral shape and an approximate size of 2 mm. Friction between plates was modeled by surface-to-surface contact formulation and a static friction coefficient of 0.2.

The base material of the sheets was aluminium AA6016 in T4 condition. An isotropic plasticity model and the Hershey-Hosford yield criterion with Voce-hardening was used, applied by the ***MAT_258** material model proposed by Costas et al. [6]. Both material and joint model parameters were adopted from Sønstabø et. al. [2].

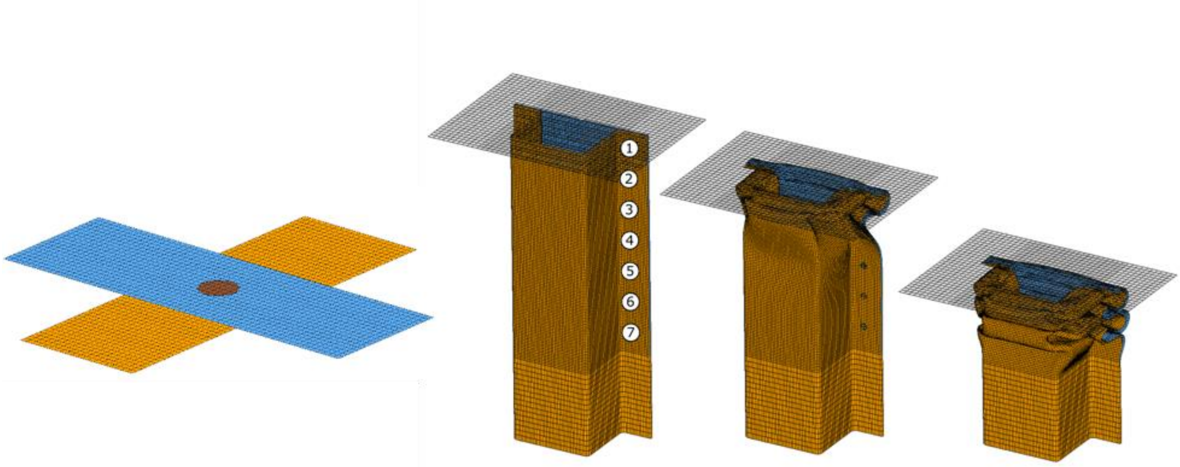

Fig.5: Crosstest for single unit test (left) and crashbox with joint numbering (right)

The results from the crosstest simulation are shown in Fig.6:. It can be seen, that both simulations with cohesive elements are in good coincidence. The NN is able to copy the model proposed by Hanssen. There is a deviation in force for the mixed mode test with delayed failure by 0.5 mm. The ***CONSTRAINED_SPR2**-model coincides with the other two shear simulations but deviates in the mixed mode and in the tensile mode. Global force-displacement results from the component simulation are shown in Fig.7: where the NN model shows almost no deviation from the ***CONSTRAINED_SPR2**-model. The component deformed and failed in a similar way.

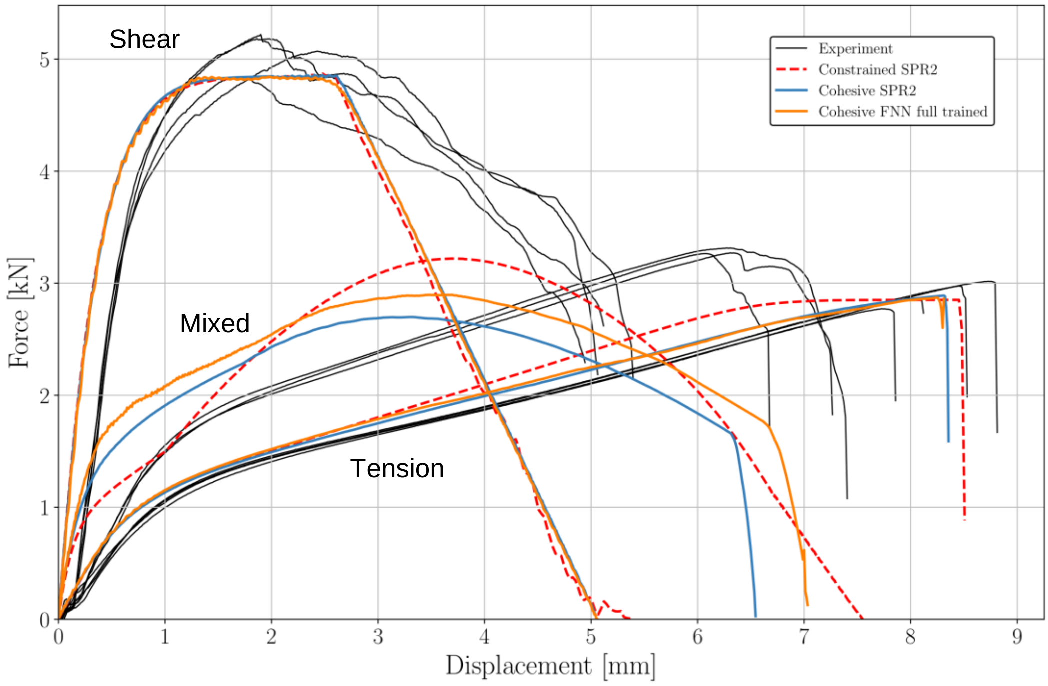

Fig.6: Force-displacement response from crosstest under tensile, mixed and shear loading; simulation with ****CONSTRAINED_SPR2****, SPR2 model as cohesive element and NN model as cohesive element*

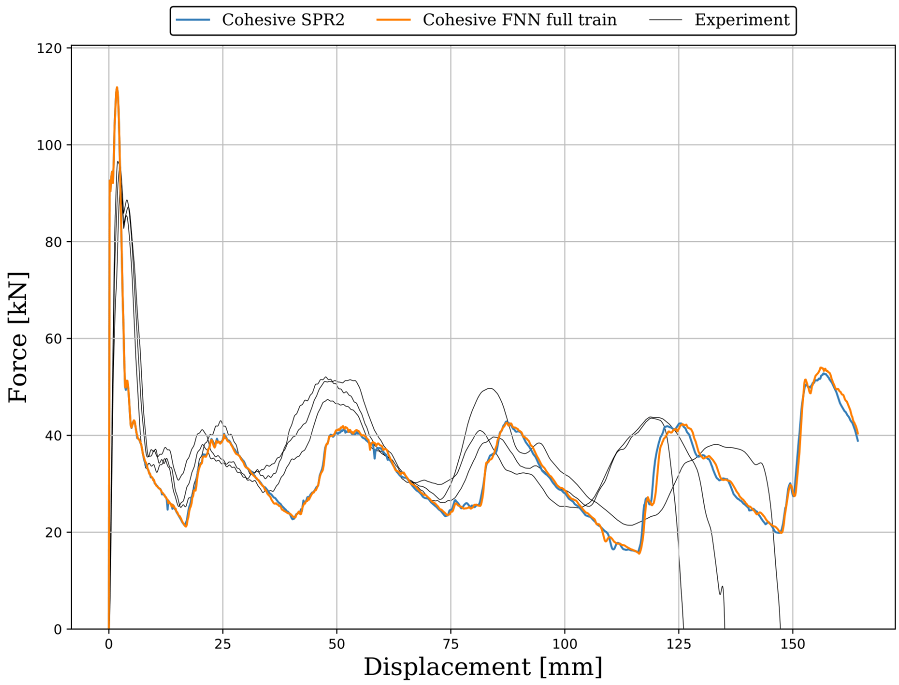

Fig.7: Global force-displacement response component simulation

The global response from the NN model was similar to the ***CONSTRAINED_SPR2**-model and in addition is the local response from the cohesive elements shown in Fig.8:. Complex and non-proportional loading can be seen here. The NN model overall replicates the model proposed by Hanssen and captures unloading where the traction component is reduced while the damage stays constant. There can be a slight deviation in traction level seen when the joint is about to reach failure.

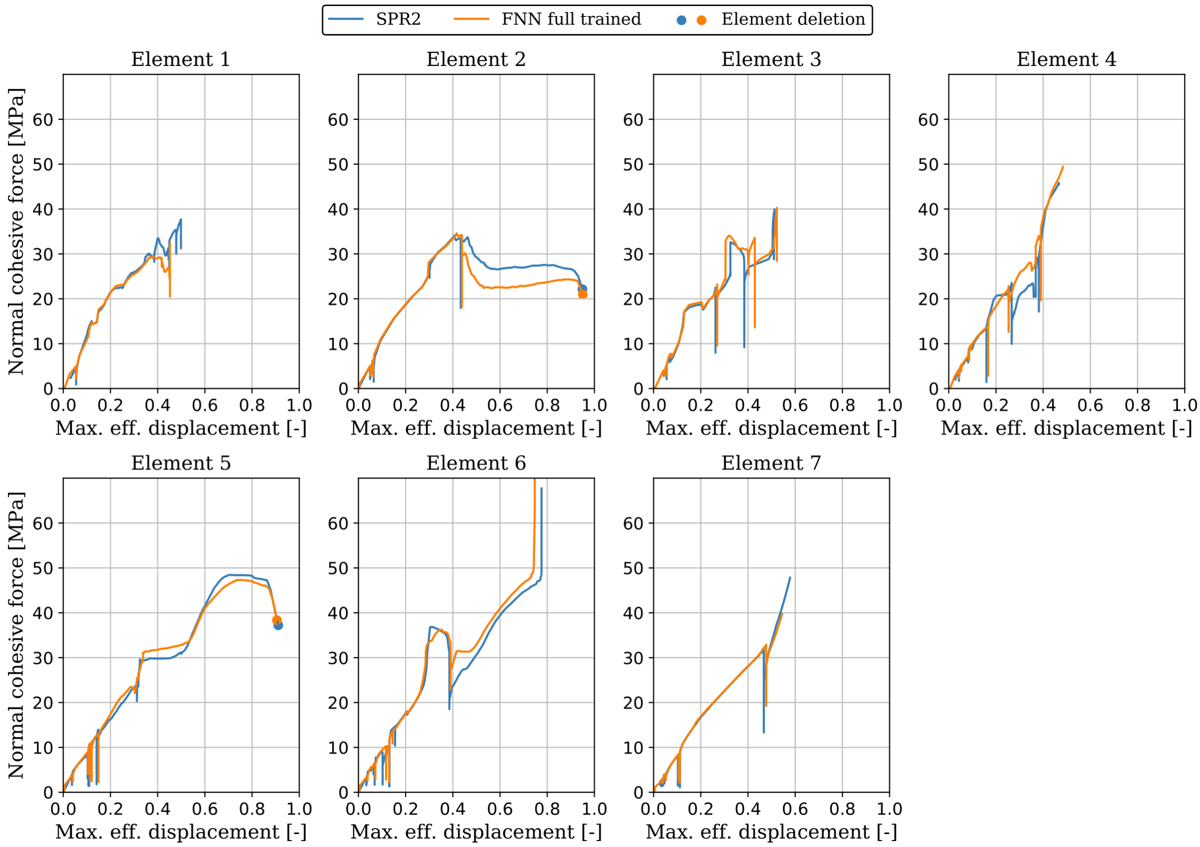

Fig.8: Local response cohesive element 1-7; normal traction component over damage measure; failure/deletion of element 2 and 5

5 Conclusion

A phenomenological model for joints represented by ***CONSTRAINED_SPR2** was replaced by a feedforward NN and successfully applied in a user subroutine to run in the explicit LS-DYNA solver. The constraint model was used as training basis for the NN and both were compared in simulations. The global force-displacements results were compared to physical tests and locally in the cohesive elements between both models. Following conclusions are made:

- A common feedforward NN architecture is found which replicates the ***CONSTRAINED_SPR2** behaviour, which serves as proof of concept for application in an explicit FE routine
- The NN works purely by matrix multiplication using the Intel Math Kernel Library and can take the input variables in vectorized form.
- It efficiently computes force and damage components each time step without oscillations or numeric instabilities
- The NN was able to capture unloading and the point of joint failure, so that the local behavior matched the model proposed by Hanssen

6 Literature

[1] Hanssen, A. G.; Olovsson L.; Porcaro R.; Langseth, M: "A large-scale finite element point-connector model for self-piercing rivet connections", European Journal of Mechanics, A/Solids, 2010, 29(4):484-495

[2] Sønstabø, J. K.; Holmstrøm P. H.; Morin, D.; Langseth, M.; "Macroscopic strength and failure properties of flow-drill screw connections", Journal of Materials Processing Technology, 2015, 222:1-12

[3] Sønstabø, J. K.; Morin, D.; Langseth, M.; " Macroscopic modelling off flow-drill screw connections in thin-walled aluminium structures", Thin Walled Structures, 2016, 105:185-206

[4] Sønstabø, J. K.; Morin, D.; Langseth, M.; " Testing and modelling of flow-drill screw connections under quasi-static loadings", Journal of Materials Processing Technology, 2018, 255:724-738

[5] Sønstabø, J. K.; Morin, D.; Langseth, M.; " Static and dynamic testing and modelling of aluminium joints with flow-drill screw connections", International Journal of Impact Engineering, 2018, 115:58-75

[6] Costas, M.; Morin, D.; Hopperstad, O. S.; Børvik, T.; Langseth, M.: "A through-thickness damage regularisation scheme for shell elements subjected to severe bending and membrane deformations", Journal of the Mechanics and Physics of Solids, 2019, 123:190-206