Szyndler, Joanna; Härtel, Sebastian; Bambach, Markus

Machine learning of the dynamics of strain hardening based on contact transformations

Journal of Intelligent Manufacturing

Journal of Intelligent Manufacturing (2026) 37:933–954
https://doi.org/10.1007/s10845-025-02577-6

Machine learning of the dynamics of strain hardening based on contact transformations

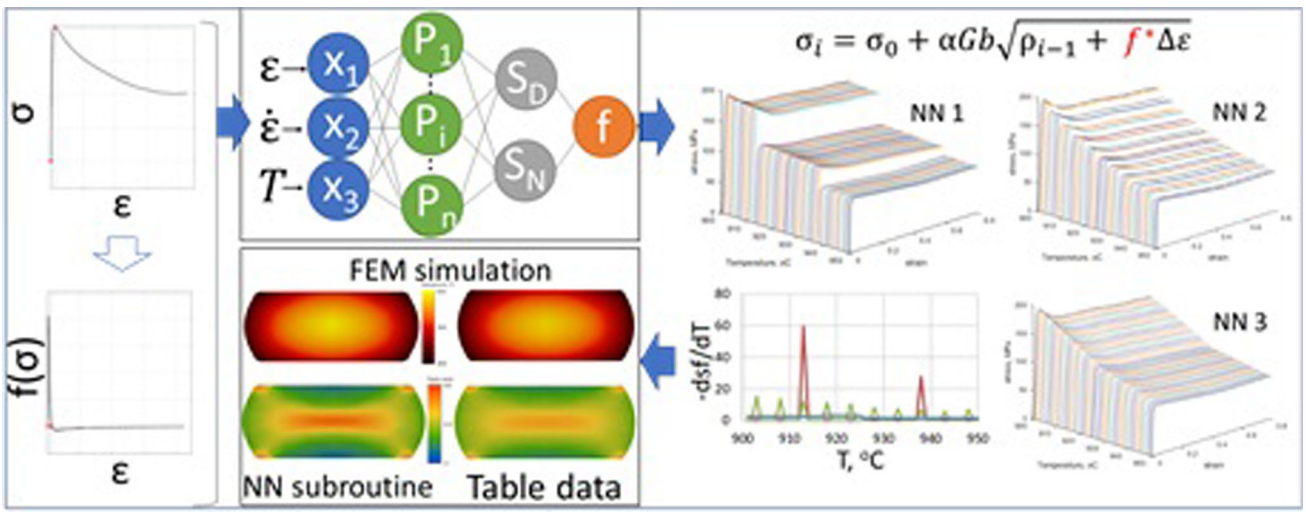
Joanna Szyndler[1] · **Sebastian Härtel**[1] · **Markus Bambach**[2]

Received: 15 May 2024 / Accepted: 21 January 2025 / Published online: 5 February 2025

Abstract

Dislocation density-based models offer a physically grounded approach to modeling strain hardening in metal forming. Since these models are typically defined by Ordinary Differential Equations (ODEs), their accuracy is constrained by both, the model formulation and the parameter identification process. Machine Learning (ML) provides an alternative by allowing models to be constructed directly from experimental data, bypassing the accuracy limitations of explicitly defined models. However, applying ML to ODEs introduces the need for novel training techniques. This work presents a new approach for developing neural ODE models for flow curve description, utilizing a contact transformation to simplify the problem of learning an ODE into a learning a multivariate function.

Graphical abstract



Keywords Machine learning · Flow curve · Forming · Finite element model · Dislocation density

Introduction

When metals and alloys are deformed plastically, the stress necessary to keep up plastic deformation typically increases. This phenomenon is referred to as strain hardening. Due to the importance of metal forming, many flow stress models have been developed in the past. The particular model used in the simulation of a process or even a process chain has

✉ Joanna Szyndler
szyndler@b-tu.de

[1] Chair of Hybrid Manufacturing, Brandenburg University of Technology Cottbus–Senftenberg, Konrad–Wachsmann–Allee 17, 03046 Cottbus, Germany

[2] Advanced Manufacturing Laboratory, ETH Zürich, Technoparkstrasse 1, 8005 Zurich, Switzerland

a decisive influence on the accuracy of process simulations. According to the famous quote, "All models are wrong, but some are useful", the following paragraphs divide flow stress models into four classes, analyze their limits, and propose a new data-driven approach to overcome them.

Empirical models

Flow curve models are based on algebraic equations that link the macroscopic flow stress to the process variables strain, strain rate, and temperature. They are primarily "wrong" in the sense that they cannot account for any mechanism related to the underlying microstructure, e.g., the relaxation of the microstructure that occurs when temperature or strain rate changes, and thus have limited accuracy, especially in forming operations at elevated temperature (i.e., Norton-Hoff model described by Norton (1929)). They are also wrong because they can only approximate the experimental data within the functional space spanned by the model functions.

Semi-empirical models

This type of model is based on experimental results and casts some experimental evidence into constitutive equations that draw upon physical quantities such as activation energies while modeling other aspects using regression models (i.e., Beynon-Sellars described by Beynon and Sellars (1992) or Zerilli-Armstrong model presented by Zerilli and Armstrong (1987)). One of the salient features is that these models are missing state variables that represent the thermo-mechanical history. Their usage is thus limited to deformation conditions covered by the lab-scale experiments used to parametrize these models, i.e., they cannot extrapolate.

Physics-based models

In most metals and alloys, plastic deformation is mediated by dislocations that move under shear stresses acting on glide planes. New dislocations need to be generated to maintain a certain strain rate. According to Taylor (1934), the critical shear stress rises with the square root of dislocation density:

$$\sigma = \sigma_0 + \alpha G b \sqrt{\rho} \tag{1}$$

Evolution laws for the dislocation density are needed to predict the dependence of flow stress on strain, strain rate, and temperature. Kocks and Mecking (2003) proposed a minimal viable model, and a multitude of more sophisticated models exist. Some common models are summarized in Table 1.

These models are based on ODEs and can trace the evolution of dislocation density (and other internal variables representing microstructure) under varying deformation conditions. Recent work has linked the physics-based theory of flow stress to ab initio calculations presented, i.e., by Bambach et al. (2021a) and Li et al. (2019). However, Kocks and Mecking (2003), the pioneers of dislocation-based models for flow stress, gave the rather pessimistic forecast that "an ab initio theory of strain hardening, with a quantitative prediction of the numerical constants, is unlikely to everbe derived even for a specific case, and impossible with any generality."

Table 1 Common dislocation density models

References	Evolution equation
Kocks and Mecking (2003)	$\frac{d\rho}{d\varepsilon} = k_1\sqrt{\rho} - k_2\rho$
Bergström (1983)	$\frac{d\rho}{d\varepsilon} = U_0\sqrt{\rho} - \left[\Omega_0 + Cexp\left(-\frac{mQ_v}{RT}\right)\dot{\varepsilon}^{-m}\right]\rho$
	$\frac{d\rho}{d\varepsilon^p} = U(\varepsilon^p) - \Omega\ \rho - A$
Estrin and Mecking (1984)	$\frac{d\rho}{d\varepsilon^p} = M\left(k + k_1\sqrt{\rho} + k_2\rho\right)$

Physics-based models often require simplifications related to microstructure evolution, which necessitates the introduction of assumptions. As a result, their accuracy is constrained by the explicit model formulation and numerical challenges associated with parameter identification. How can a strain-hardening model be constructed without the restrictions of empirical and physics-based approaches? One possibility is to resort to data-driven models.

Data-driven models

With the recent rise of artificial intelligence and deep learning methods in manufacturing, flow stress models based on Neural Networks (NNs) were developed. Deep learning algorithms are excellent at approximating functions defined when a sufficiently large number of input–output datasets is available (see i.e. Rihi et al., (2022; , 2024)). The mathematical theory proves that NNs approximate smooth functions on closed intervals arbitrarily well, provided that enough data are available, cf. Barron (1993). Thus, they can be shown to outperform classical empirical and physics-based models on the training and validation data provided, if the task is to predict flow curves for constant temperatures and strain rates. An example of such a model for flow stress prediction is described by Phaniraj and Lahiri (2003), where a simple 2-layer NN is proposed, which takes the strain, strain rate, temperature, and the carbon equivalent as input data and predicts flow stress behavior for a range of carbon steels under specific experimental conditions. Similarly, Rao and Prasad (1995) used a back-propagation neural network to describe the evolution of flow stress during the hot deformation process of medium carbon steel. Reddy et al. (2008) also attempted to develop a back-propagation NN, in this case, to

predict the behavior of Ti6Al4V under any given processing conditions across a wide phase range, including the α + β and the β phase region. Further work was presented by Wang et al. (2016), where experimental data are compared with results of Johnson–Cook (JC), Khan-Huang-Liang (KHL), and modified Zerilli-Amstrong (ZA) constitutive models, as well as with artificial neural networks that predict the flow stress behavior of Ti6Al4V in the temperature range from 20 to 600 °C and a few discrete strain rate values (10^{-6}, 10^{-5}, 10^{-4} s-1). This time, a three-layer NN model with a back-propagation training algorithm was developed, where strain, stress, and temperature were used as input data and flow stress was an output of the trained NN. Another example of such investigation is made by Siewior and Madej (2021), who tested the capabilities of two types of artificial neural networks to predict flow stress data of magnesium ZE20. Similar research work was performed by Stendal et al. (2019), where the application of three different machine learning methods to the phenomenological flow stress modeling of the TiAl alloy was analyzed. Also, Kusiak and Pietrzyk (1999) applied a neural network strategy to predict the yield stress in the hot forming of metals.

However, in actual metal-forming processes, the flow stress typically evolves under varying temperatures and strain rates. Thus, it should be described by evolution equations for internal variables, such as the dislocation density, rather than by feed-forward NNs. Up to now, limited work deals with neural networks for dynamical systems, i.e., systems described by ordinary differential equations (ODE). One of the earliest examples was presented by Chen et al. (2018), where the use of a black-box ODE solver as a model component was explored, with a focus on developing new models for time-series modeling development, supervised learning, and density estimation. Further work was put forward by Habiba and Pearlmutter (2020), where the mathematical foundation of ODEs was used to design two new ODE-based Recurrent Neural Network (RNN) models. The described solution significantly reduced the computation overhead of the hidden and cell states. Also, the required training time was pointed out to be less than the latent ODEs and conventional neural ODE solutions. In wire-arc additive manufacturing, Bambach et al. (2022) presented a structured approach to learning the semi-discrete system equation of a finite volume model, which represents a high-dimensional ODE system, from thermal imaging data. In deformation processing, Tac et al. (2022) applied neural ODEs (N-ODEs) to estimate the derivatives of the strain energy function with respect to deformation invariants, using both synthetic and experimental stress-stretch data from porcine skin biaxial experiments. The study also demonstrated the applicability of the N-ODE-based material model in finite element simulations. In a follow-up study, Tac et al. (2023) developed a fully data-driven model of anisotropic finite viscoelasticity using N-ODEs, introducing N-ODE-based functions for the Helmholtz free energy function, the creep potential, and an evolution equation for fiber stress. More recently, Tac et al. (2024) extended the N-ODE approach to develop a fully data-driven continuum damage mechanics model, incorporating polyconvexity in the strain energy function and dissipation related to damage accumulation. While these models target plasticity in general terms, there is still limited work on flow stress modeling for hot forming. Notably, Bambach et al. (2021b) present a physics-based model framework, replace the model functions typically used in Eq. (1) above with long short-term memory (LSTM) networks, and propose a data assimilation method to update the model equations whenever new data has been measured. This approach focuses on process control, where the response of the workpiece is measured regularly and does not intend to set up a data-driven model for process simulation systematically.

Alternatively, the evolution of flow stress could be modeled with Recurrent Neural Networks (RNNs), which are capable of creating most of the dynamics, which are, for instance, discussed in a review publication by Liu et al. (2021). Also, Liu et al. (2021) focus on the history, current development, and promising applications of machine learning, especially in fluid mechanics. While Mei et al. (2020) used a long short-term memory RNN to characterize the constitutive relationship of Inconel 718 with excellent results in comparison to the experimental data, Chen (2021) developed an RNN model that was applied to learn the underlying physical law of viscoelasticity. Also, Gorji et al. (2020) used RNNs to demonstrate their ability to replicate the predictions for arbitrary loading paths of one of the most complex anisotropic plasticity models, Yld2000-2d, with homogeneous anisotropic hardening. Moreover, the deformation response of a two-dimensional metallic foam has been successfully described through an RNN.

RNNs are designed to recognize the sequential nature of input data and use detected patterns to predict future outputs. Thus, the output from the previous steps is fed into the input of the current state. This type of neural network is designed to retain information over time, meaning that RNNs use data from the previous time step to predict the outputs of the current time step. As explained by Rius et al. (1998), this approach requires fixed time step values. In summary, RNNs are capable of modeling time series, but they proceed at constant time step values, which makes them difficult to apply in finite element (FE) simulations. Hence, RNNs do not seem suitable for direct implementation into FE models, where the time increment size is dynamically adjusted throughout the analysis so that the convergence in implicit time integration is achieved or stability limits are fulfilled in explicit time integration, as described by Belytschko et al. (2014).

While empirical, semi-empirical, and physics-based models are limited by their constrained ability to accurately represent true material behavior across a wide range of forming conditions, data-driven models based on ANN and RNN are limited either by their inability to dynamically synchronize with the time steps defined by a finite element solver. Except for Bambach et al. (2021b), no prior attempts were made to devise neural ODE models for strain hardening. That work, however, did not aim at combining the model with an FE solver.

Therefore, the present work aims at deriving a framework for generating accurate dynamic models of strain hardening from experimental data that is applicable in FEM simulations. The authors hypothesize that the dynamics of strain hardening can be effectively modeled using a neural ODE approach. To test this hypothesis, the objective of this study is to train a neural network with carefully selected parameters and implement it as a user subroutine within an FE simulation program. The performance of this neural ODE-based model will be validated against results obtained using conventional methods.

An ML framework for the dynamics of strain hardening

The first approach taken in this paper is to apply a contact transformation to Eq. (1), shifting the problem from stress–strain space (σ, ε) into the space defined by the strain hardening rate θ and stress σ (θ, σ):

$$\theta = \frac{d\sigma}{d\varepsilon} = \alpha G b \frac{d\rho/d\varepsilon}{2\sqrt{\rho}} = \underbrace{(\alpha G b)^2 \frac{d\rho/d\varepsilon}{2(\sigma - \sigma_0)}}_{g(\sigma(\varepsilon), \dot{\varepsilon}, T)} \tag{2}$$

here, G, b, ρ, ε, α are the shear modulus, burgers vector, dislocation density, strain rate, and the Taylor constant, respectively.

When the true stress–true strain curves recorded experimentally are transformed into θ–σ-space, the resulting right-hand side of Eq. (2) can be represented by an interpolant g*. Back transformation is then easily achieved by integrating the strain hardening rate along the deformation path. ODE solvers capable of achieving the desired precision can be employed. This approach does not allow for a physical interpretation. Solving Eq. (2) for $d\rho/d\varepsilon$, in contrast, yields an explicit data-driven evolution equation for the dislocation density:

$$\frac{d\rho}{d\varepsilon} = \underbrace{2\theta(\sigma - \sigma_0)(\alpha G b)^{-2}}_{f(\rho(\varepsilon), \dot{\varepsilon}, T)} = 2\theta\sqrt{\rho}(\alpha G b)^{-1} \tag{3}$$

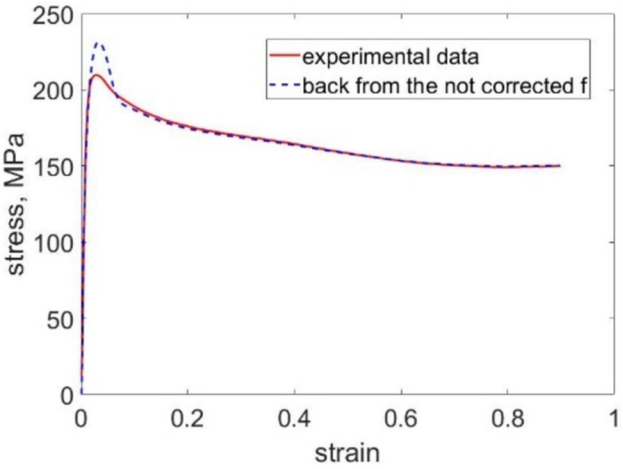


Fig. 1 A plot of the stress–strain curves is calculated from the uncorrected (blue) interpolant f* and the reference experimental data (red) (Color figure online)

Here, σ, ε, θ are the stress, strain, and strain hardening rate, respectively.

The right-hand side of Eq. (3) needs to be computed from flow stress data and then represented by a suitable interpolant f*. This allows for integrating the internal variable ρ along the deformation path and hence, computing the flow stress σ for arbitrary load paths using Eq. (1).

It is worth mentioning that the interpolants f* and g* used in both approaches will not yield the original stress–strain curve. The reason is that the interpolants f* and g* are affected by errors stemming from numerical differentiation, and by interpolation errors and noise. Thus, in both cases, the right-hand side needs to be corrected so that their integration yields the original flow stress data. An example is shown in Fig. 1.

This task can be solved in various ways. When a neural network is applied to solve this problem, backpropagation in the training phase has to be performed through the ODE solver. Recent work in computer science proposes to solve an adjoint problem to identify the right-hand side, which is then represented by a neural network. If the right-hand side is represented by a neural network, this network needs to predict the dynamics for various strain rates, temperature, and maybe also microstructural states, which makes training very demanding. Thus, a different approach is followed here: the right-hand side is first identified for each sample, and then a neural network is used to construct the right-hand side for arbitrary input values. In this case, for each stress–strain curve recorded at constant temperature and strain rate, an optimization problem is solved to identify the right-hand side. It is worth noting that for a stress–strain curve with a single peak, the function f has a characteristic shape. It starts at $f(\sigma_0) = 0$, reaches a maximum, and has a second root at the peak stress ($f(\sigma_{peak}) = 0$). Afterward, it passes through a

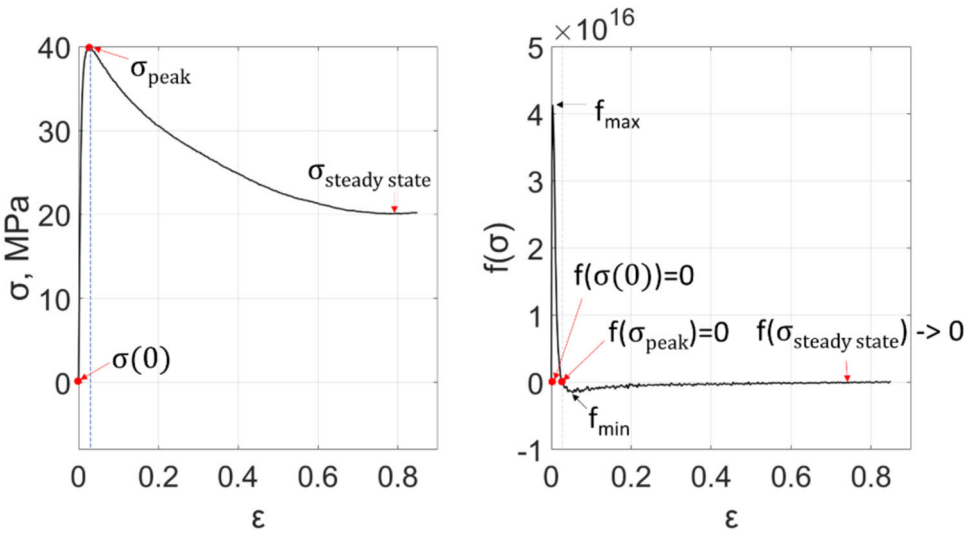


Fig. 2 Graphical explanation of the shape of the f-function with its characteristic points (Color figure online)

minimum and gradually approaches zero as it nears steady-state deformation (Fig. 2).

Thus, an interpolant can be constructed based on these characteristic points, and additional support points can be added as needed. An example of the plot of the interpolant f* and the flow stress calculated from it, with and without the mentioned error correction, is presented in Fig. 3.

Data prepared in the form of temperature, strain rate, strain, and interpolant f*, can then be used for training of the neural network. Figure 4 presents a flowchart summarizing the proposed framework.

Before the final neural network type was selected, different types of neural networks were tested, and various training methods were explored, e.g., the Levenberg–Marquardt backpropagation method, Bayesian regularization backpropagation, Resilient backpropagation, and more. However, most of them revealed convergence or training speed problems. The tested Levenberg–Marquardt backpropagation algorithm generally has a fast convergence speed when the input values are given appropriately. Unfortunately, its solution largely depends on the quality of initial values and the shape, including the complexity of the search space. Thus, it easily converges to only a local minimum, which may not represent the global minimum, which was a common problem with the data applied in this research. A similar situation was observed when using a Bayesian regularization backpropagation NN. The training process is repeated, and learning stops when either the target error threshold or the maximum number of iterations is reached. However, since the BP learning algorithm is a first-order gradient method, its convergence speed is slow, making it prone to getting trapped in local minima. A similar issue was observed with the resilient backpropagation solution, which also can tend to skip the extremum due to the discontinuity at the minimum of the local approximation. The most promising results were obtained with the Generalized Regression Neural Network (GRNN) available in Matlab®, a radial basis function network using a feedforward training solution. Its key advantages include high accuracy and the need for only single-pass learning, eliminating the need for backpropagation. It also handles noisy input data effectively, which minimizes the risk of falling into local minima. A general scheme of the architecture of the applied GRNN model is shown in Fig. 5.

The GRNN generally includes input, pattern, summation, and output layers. The number of neurons in the input layer is determined by the dimension of the input vector from the training data samples. In this case, a $3 \times Q$ matrix is used as input data, representing strain, strain rate, and temperature. These neurons pass the input variables directly to the pattern layer, which is part of the hidden layer in the network. The number of neurons in this layer corresponds to the number of training samples Q, with each neuron representing one sample. The user is free to adjust the *spread* parameter, which affects the quality of the NN's training results. In the summation layer, which is also part of the hidden layer of this network, two types of neurons are used: one that performs a summation of the weighted outputs (S_{N1}) and another that operates as a summation-only weight (S_D). The final output layer produces the prediction results of the GRNN model by calculating the ratio of the S_{N1} neuron to the S_D neuron. A detailed diagram illustrating the GRNN architecture, adapted from Demuth and Beale (1998), is presented in Fig. 6.

The ||dist|| operation in Fig. 6 takes the input vector (p) and the input weight matrix ($\mathbf{IW^{1,1}}$), producing a vector of Q elements. These elements represent the distances between the input vector ($\mathbf{p}$) and the vectors formed from the rows of the input weight matrix. The bias ($\mathbf{b^1}$), set to a column vector of 0.8326/spread, is combined with the output of ||dist|| using element-wise multiplication. A radial basis transfer function ($radbas(n) = e^{-n^2}$) is applied to calculate the output of

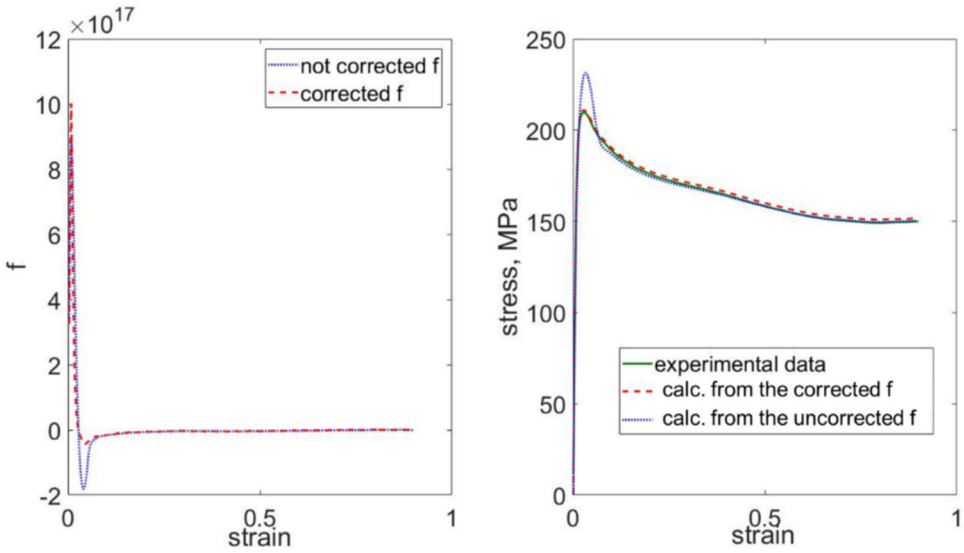


Fig. 3 Comparison between the flow stress calculated based on corrected (red) and uncorrected (blue) interpolant f*. The flow stress obtained from the experiment is presented as a green line

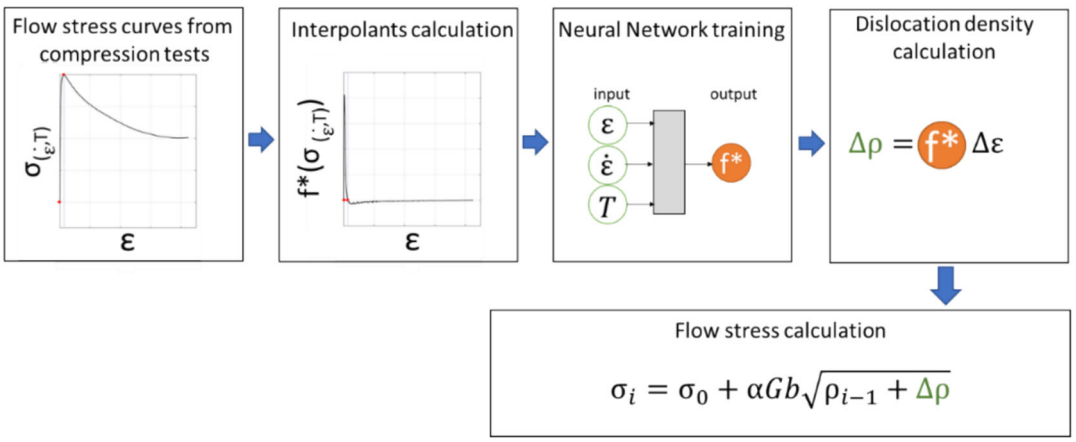


Fig. 4 Flowchart of the proposed framework summarizing the overall concept

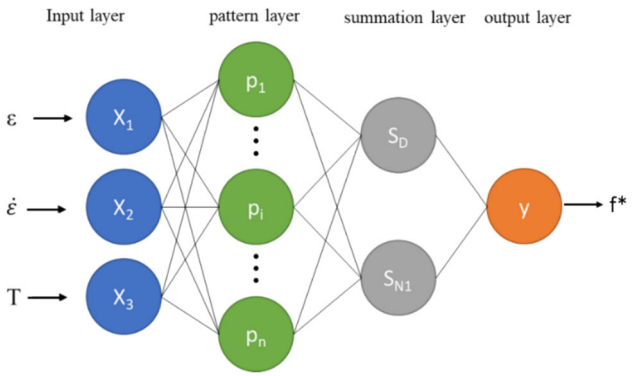


Fig. 5 General scheme of the generalized regression neural network

the hidden layer. In the final layer, the network produces Q elements in the vector $\mathbf{n^2}$ using the function *nprod*, which calculates the dot product of each row of $\mathbf{LW^{2,1}}$ with the input vector $\mathbf{a^1}$. The output is normalized by the sum of the elements in $\mathbf{a^1}$, and the result ($\mathbf{n^2}$) is passed through the linear transfer function (*purelin*) to compute the final network output ($\mathbf{n^2}$). During the training process, the number of input vectors determines the number of radial basis neurons and linear neurons. After training, the GRNN generates the appropriate weight matrices ($\mathbf{IW^{1,1}}$ and $\mathbf{LW^{2,1}}$) and the threshold vector ($\mathbf{b^1}$).

Estimating the value of the spread is crucial because it influences the quality of the neural network's training process. The larger the *spread*, the smoother the function approximation. A smaller *spread* value should be used to fit data very closely, whereas a larger *spread* value is recommended to fit the data more smoothly.

GRNN—spread analysis

As mentioned earlier, the value of the spread parameter can significantly influence the prediction accuracy of the trained

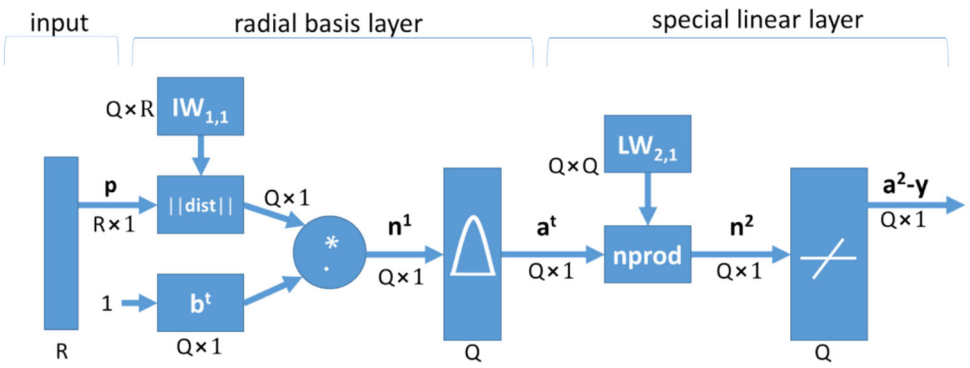


Fig. 6 Detailed diagram of the GRNN, where R represents the number of elements in the input vector, Q represents the number of neurons in layers 1 and 2, and input/target pairs. The diagram also includes the Euclidean distance function (||dist||), the input weight matrix ($\mathbf{IW^{1,1}}$) of the first layer, the hidden layer threshold ($\mathbf{b^1}$), and the weight vector of the linear layer ($\mathbf{LW^{2,1}}$)

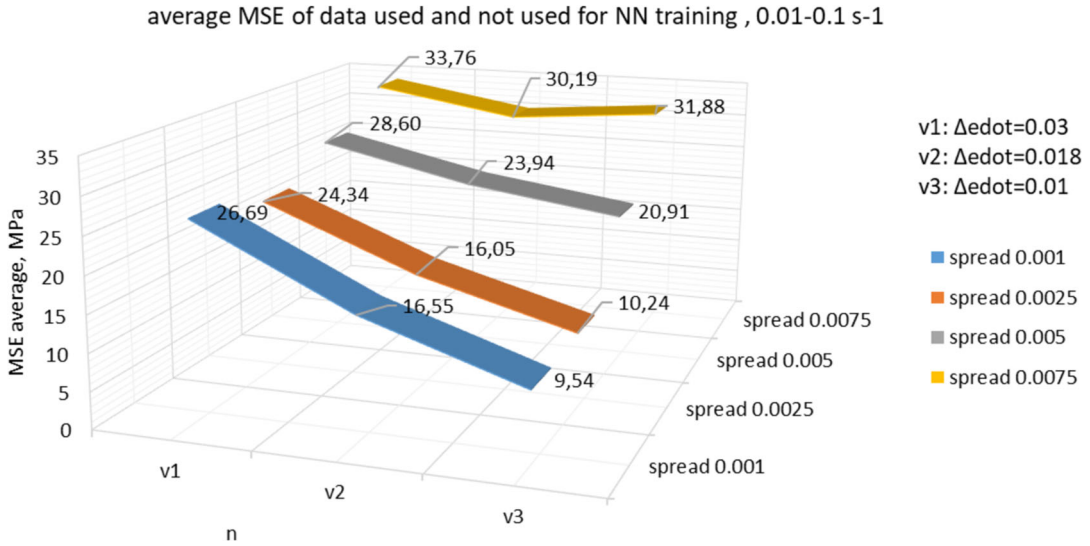


Fig. 7 A plot of the average MSE values for the strain rate training range 0.01–0.1 s^{-1}

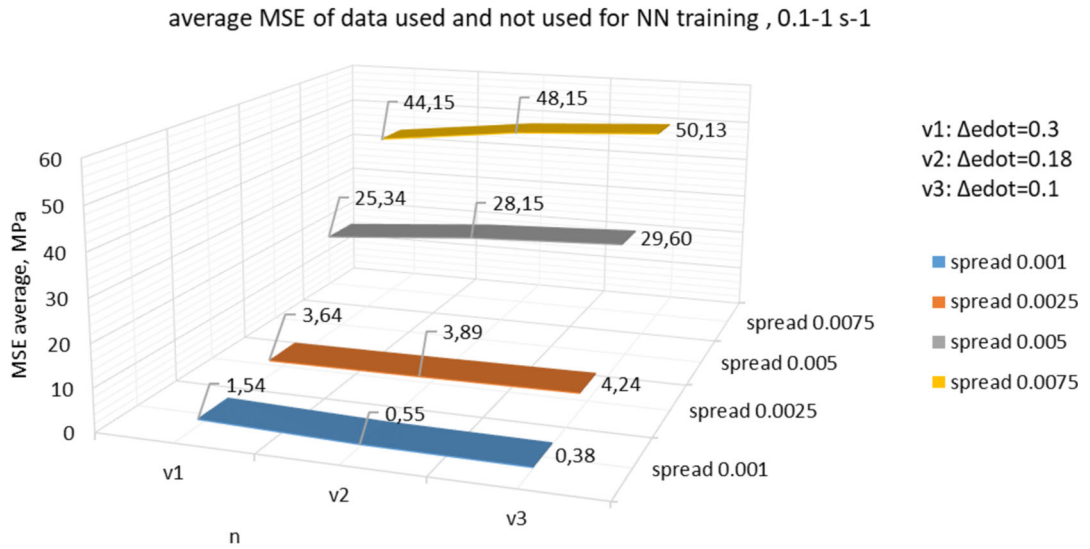


Fig. 8 A plot of the average MSE values for the strain rate training range 0.1–1 s^{-1}

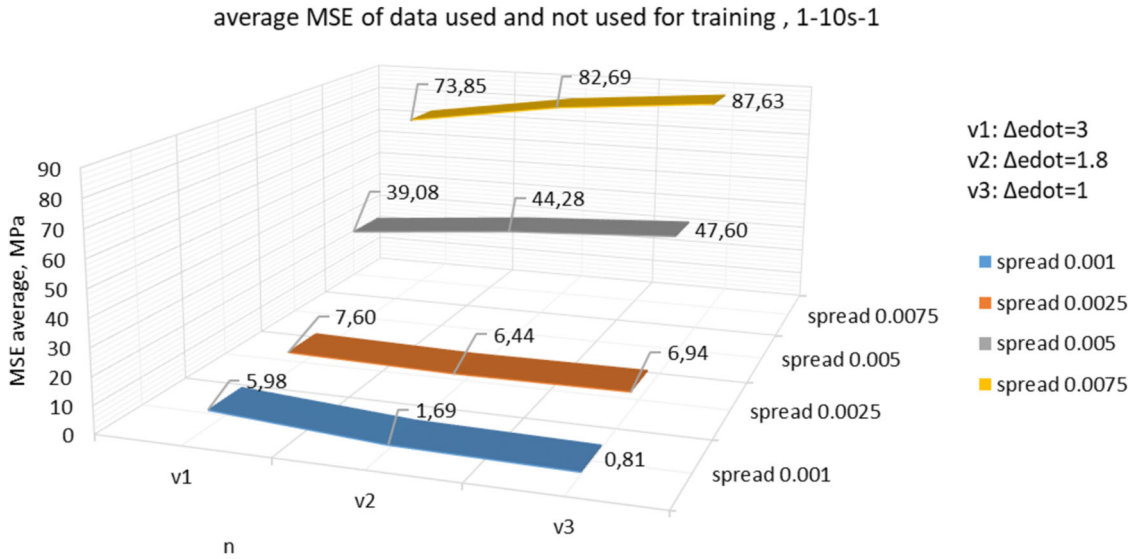


Fig. 9 A plot of the average MSE values for the strain rate training range 1–10 s^{-1}

Table 2 Parameters used for NN training

$\dot{\varepsilon}$ range	0.01–0.1 s^{-1}	0.1–1 s^{-1}	1–10 s^{-1}
	$\Delta\dot{\varepsilon}$, s^{-1}		
V_1 ($n = 4$)	0.03	0.3	3
V_2 ($n = 6$)	0.018	0.18	1.8
V_3 ($n = 10$)	0.01	0.1	1

Table 3 Parameters used for NN testing

$\dot{\varepsilon}$range	0.01–0.1 s^{-1}	0.1–1 s^{-1}	1–10 s^{-1}
	$\dot{\varepsilon}$, s^{-1}		
V_1 ($n = 3$)	0.025, 0.055, 0.085	0.25, 0.55, 0.85	2.5, 5.5, 8.5
V_2 ($n = 5$)	0.019, 0.037, 0.055, 0.073, 0.091	0.19, 0.37, 0.55, 0.73, 0.91	1.9, 3.7, 5.5, 7.3, 9.1
V_3 ($n = 9$)	0.015, 0.025, 0.035, 0.045, 0.055, 0.065, 0.075, 0.085, 0.095	0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95	1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5

neural network. Therefore, it was essential to analyze this factor. Four spread values were selected for investigation: 0.001, 0.0025, 0.005, and 0.0075. The neural network was trained and tested for each of these values, using different numbers of flow stress curves to assess the impact of the spread on the prediction quality. The training and testing data were generated using a flow stress model developed in earlier work by Bambach et al. (2021a), which describes the behavior of additively manufactured Ti6Al4V material at 900 °C during compression tests. All test parameters are summarized in Table 2, where *n* represents the number of the f* curves used for the NN training. To evaluate the reliability of the network's prediction, a different set of parameters—those not used during training—was tested, as shown in Table 3. After training the neural network, the Mean Squared Error (MSE) was calculated for each case, along with the average MSE for each spread value Vx V_x (Figs. 7, 8, and 9).

The results presented in Figs. 7, 8, and 9 clearly demonstrate that the smallest MSE value is achieved with the smallest spread value (0.001) and the highest number of training data simultaneously. However, preparing the training data for the neural network requires a series of experiments, which can be costly and time-consuming. Therefore, a balance must be struck between time, cost, and quality of results.

It is worth noting that even when compression tests are conducted at a constant strain rate, the strain rate distribution varies across the sample. Given that strain rate significantly affects stress, it is essential to provide a sufficiently wide range of training data to ensure the NN can predict material behavior across different strain rates. Thus, based on the presented results, the authors decided to select, for further analysis, stress–strain curves obtained for the following strain rate values: 0.01, 0.04, 0.07, 0.1, 0.4, 0.7, 1, 4, 7, and 10 s^{-1}, and to train the NN with a spread value equal to 0.001. This way, only ten flow stress curves were required to train the neural network, which predicts the interpolant f* that is later used for the flow stress calculation. Figure 10 presents the results of the NN training, showing a comparison between expected flow stress curves (exp) and calculated from the NN- predicted interpolant f* (NN). The MSE

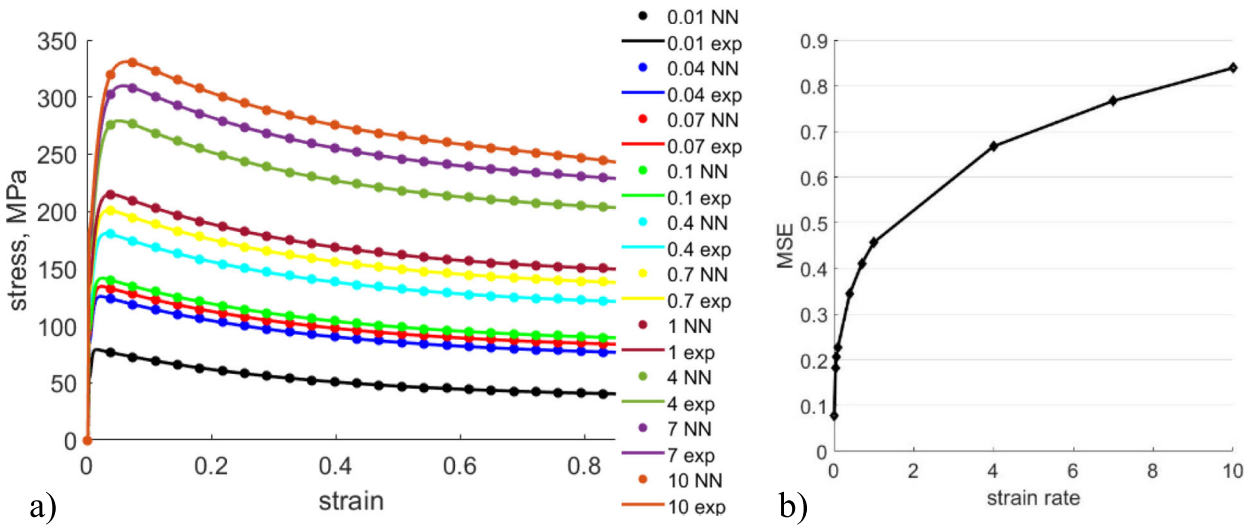


Fig. 10 **a** Expected and calculated stress–strain plots for the strain rates used for NN training; **b** MSE between expected and calculated stress–strain plots

between expected and obtained stress values is shown as a function of the strain rate. The results demonstrate excellent prediction accuracy, with the maximum MSE being lower than 0.9 MPa for the data set used in training.

The following nine strain rate values were selected to test the reliability of the neural network's prediction for input data not used in its training: 0.025, 0.055, 0.085, 0.25, 0.55, 0.85, 2.5, 5.5, and 8.5 s^{-1}. In Fig. 11a, the dotted lines represent the neural network's predictions, while the corresponding continuous colored lines show the reference data used for validation. The results demonstrate a very good match between the predicted and expected values in most cases. Figure 11b presents the MSE between the expected and predicted flow stress values, revealing that only the stress values at strain rates of 0.025 and 2.5 s^{-1} are noticeably underestimated. The maximum deviation between the expected and calculated stress values is 16.58 MPa, corresponding to a 10.84% difference. This deviation may be due to a slightly too large gap between the data points selected for the neural network training process (e.g., strain rates of 1, 4, 0.01, and 0.04 s^{-1}).

The analysis of the influence of neural network training parameters on the quality of the results enabled us to define optimal learning parameters for reliable predictions. The next step was to plan and conduct real experiments to obtain the flow stress curves necessary for the neural network training process, specifically for the Ti6Al4V alloy.

Experimental data for Ti6Al4V

Cylindrical samples with a diameter of 5 mm and a height of 8 mm were machined from the Ti6Al4V samples produced by Laser Metal Deposition (LMD). The chemical composition of the specimens was analyzed on a GDA 650 HR analyzer (Spectrum Analytic GmbH) (Table 4). The compositions differ but fall within the allowable range according to the DIN 17851:1990-11 specification of Ti6Al4V.

Based on the above analysis of the NN training process, a set of uniaxial compression tests was proposed and performed for the Ti6Al4V alloy, delivering accurate data for the NN training. The samples were deformed in a dilatometer DIL805A/D/T (TA Instruments). The samples were heated in an argon atmosphere with a heating rate of 10 K/s, dwelled for 3 min, and tested at three temperatures 900, 925, and 950 °C, at strain rates: 0.0013, 0.005, 0.01, 0.04, 0.07, 0.1, 0.4, 0.7, 1, 4, 7, and 10 s^{-1}; to a total strain 0.9. Molybdenum plates of ~ 1 mm thickness were placed between the specimen and the Si3N4 punches. The plates were heated to the deformation temperature to reduce thermal gradients in the specimen. Immediately after the compression test, each specimen was quenched using argon at a cooling rate of 100 K/s. Figure 12 presents a sample geometry, the respective finite element simulation model, and a sample after the compression test. The obtained flow stress curves are plotted in Fig. 13.

The obtained flow stress plots reveal no clear pattern in the case of the maximum recorded stress values, especially for the strain rates below 1 s^{-1} at temperatures 925 and 950 °C, which is noticeable in Fig. 13d. As expected, the peak stress value reduces with increasing deformation temperature. Interestingly, for strain rates of 4 s^{-1} and higher, the flow curves begin to lose their parallel behavior in the later stages of deformation, with some curves crossing each other, such as in Fig. 13a at strain rates of 1 and 4 s^{-1}. This is likely influenced by microstructural evolution during deformation. The results indicate that the material's behavior becomes more complex and intriguing at higher strain rates.

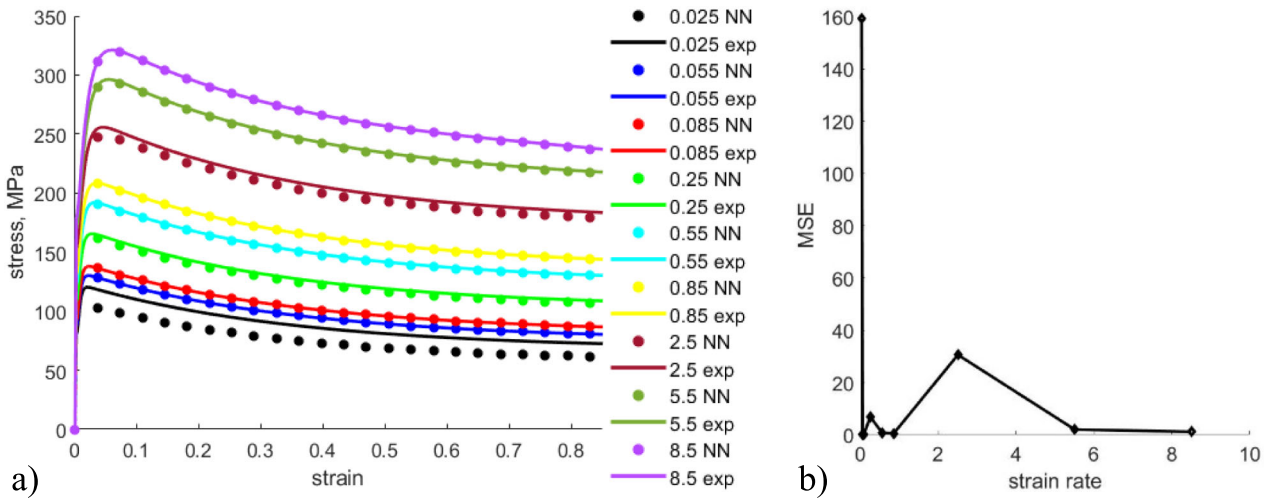


Fig. 11 **a** Expected and calculated stress–strain plots for the strain rates not used for the NN training; **b** MSE between expected and calculated stress–strain plots

Table 4 Chemical composition of specimens

Composition	Ti	Al	V	Cr	Fe	Mn	Ni	Sn	Si	C	B
Wt (%)	89.306	6.447	3.867	0.008	1.191	0.018	0.012	0.023	0.105	0.013	0.001

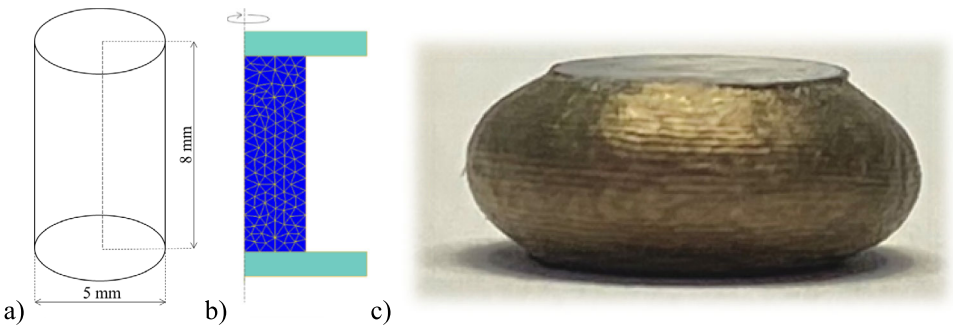


Fig. 12 **a** Specimen geometry for hot compression tests, **b** an axisymmetric finite element model for validation, and **c** sample after the compression test at a strain rate 1 s^{-1} and 900 °C

Modelling high-temperature flow curves of Ti6Al4V

Based on the experimental data obtained, interpolants f* were identified and adjusted for each temperature and strain rate, and the entire neural network training process was repeated using the determined spread value of 0.001. As shown in Fig. 14, excellent results were achieved, with a comparison between the experimentally obtained (dashed lines) and neural network-predicted (continuous lines) stress–strain plots. In the worst case, the calculated MSE between the expected and predicted flow stress plots was less than 2 MPa (Fig. 14d), with an average MSE of just 0.177 MPa.

To investigate the quality of the NN prediction for other strain rate values that were not considered during the training process, more flow stress curves were generated with the NN for the following strain rates: 0.003, 0.0075, 0.025, 0.055, 0.085, 0.25, 0.55, 0.85, 2.5, 5.5, and 8.5 s^{-1} (Fig. 15). It is evident that the predicted results (continuous lines) fit well between the curves used in the training process (dashed lines). Therefore, it can be concluded that the developed neural network is also reliable for predicting strain rate values that fall between those used during training.

To examine the NN predictions for temperature values that were not considered during the NN training process, an NN was used to determine flow stress plots for all strain rate values used for training, and 51 flow curves were generated for a temperature range of 900–950 °C. Figure 16 presents an example of the results obtained for the strain rates 0.01, 0.1, 1, and 7 s^{-1}.

As observed, three temperature values with a 25 °C difference are insufficient to maintain smooth predictions between them, with two significant drops in stress occurring between 912 and 913 °C, and between 937 and 938 °C in all cases. To

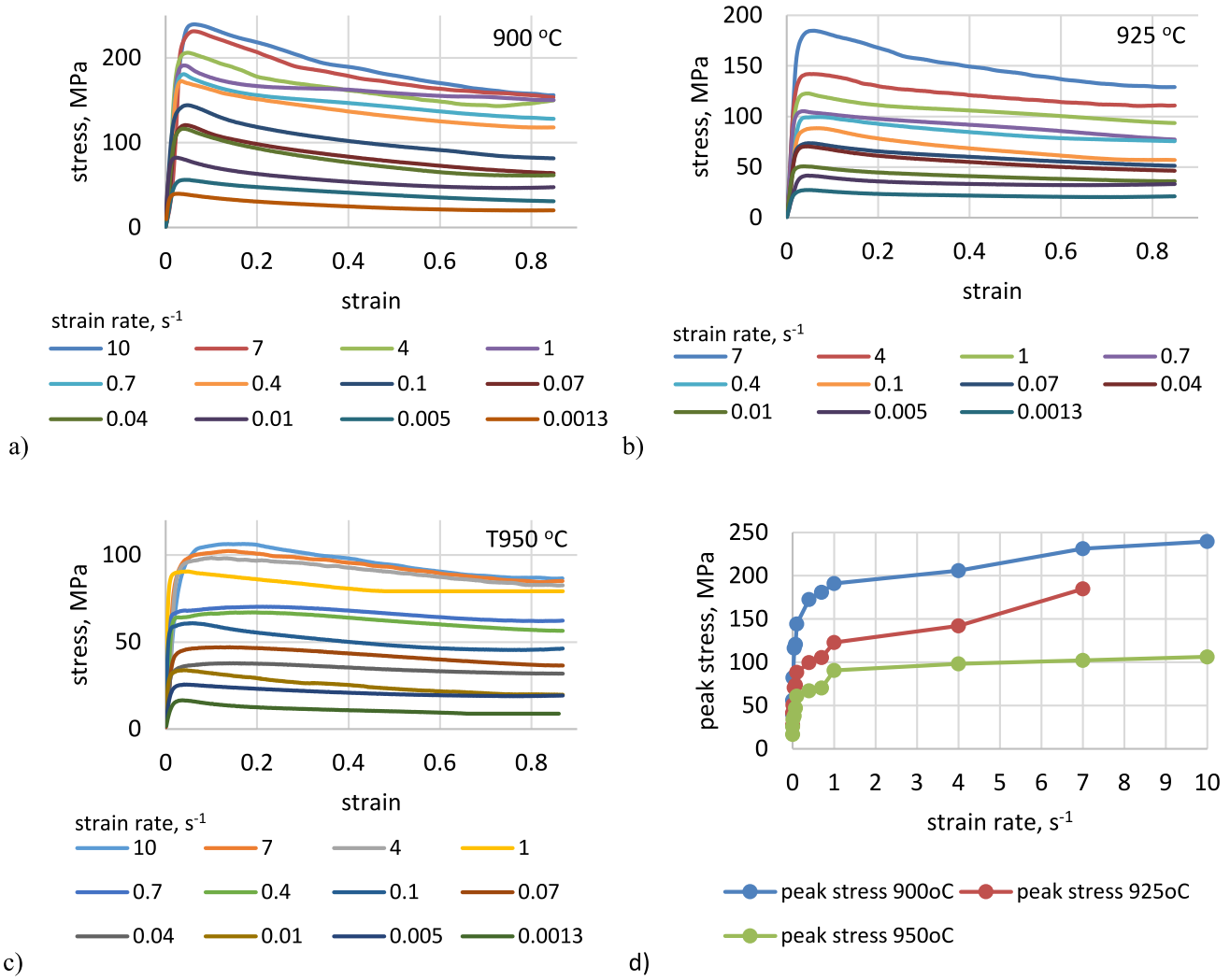


Fig. 13 Flow stress curves obtained from the experimental procedure for **a** 900, **b** 925, and **c** 950 °C. **d** Maximum recorded stress value of the flow curves obtained for different strain rates

address this issue, and based on the experimentally obtained flow stress curves (Fig. 13), a quadratic interpolation was applied to generate additional training data for the neural network at more temperature values: 900, 905, 910, 915, 920, 925, 930, 935, 940, 945, and 950 °C, at strain rates of 0.0013, 0.005, 0.01, 0.04, 0.07, 0.1, 0.4, 0.7, 1, 4, 7, and 10 s^{-1}. Examples of these interpolated stress–strain curves for strain rates of 0.01, 0.1, 1, and 7 s^{-1} are shown in Fig. 17.

The interpolants f* were then recalculated for each flow stress curve, and the entire neural network training process was repeated using the same training parameters. Once again, 51 flow curves were generated for each strain rate value over the temperature range of 900–950 °C to assess the improvement in the neural network's predictions. Figure 18 shows an example of the obtained result for strain rates of 0.01, 0.1, 1, and 7 s^{-1}.

An improvement in the neural network's predictions is already visible; however, some discontinuities for temperature values not included in the training process remain noticeable. To better assess the quality of the predictions, the derivatives of stress with respect to temperature and strain were calculated and are shown in Fig. 19a and b, respectively. The example presented in Fig. 19a was obtained at a strain of 0.04, where peak values of the flow stress curves are observed, and Fig. 19b corresponds to a temperature of 911 °C, which was not part of the training data.

As shown in Fig. 19a, the more training data is used, the smaller the differences between consecutive temperature results. When only the experimental data (three temperature values) was used for training, the maximum calculated difference was 60 MPa (red line). In contrast, when the amount of training data was artificially increased (to 11 temperature

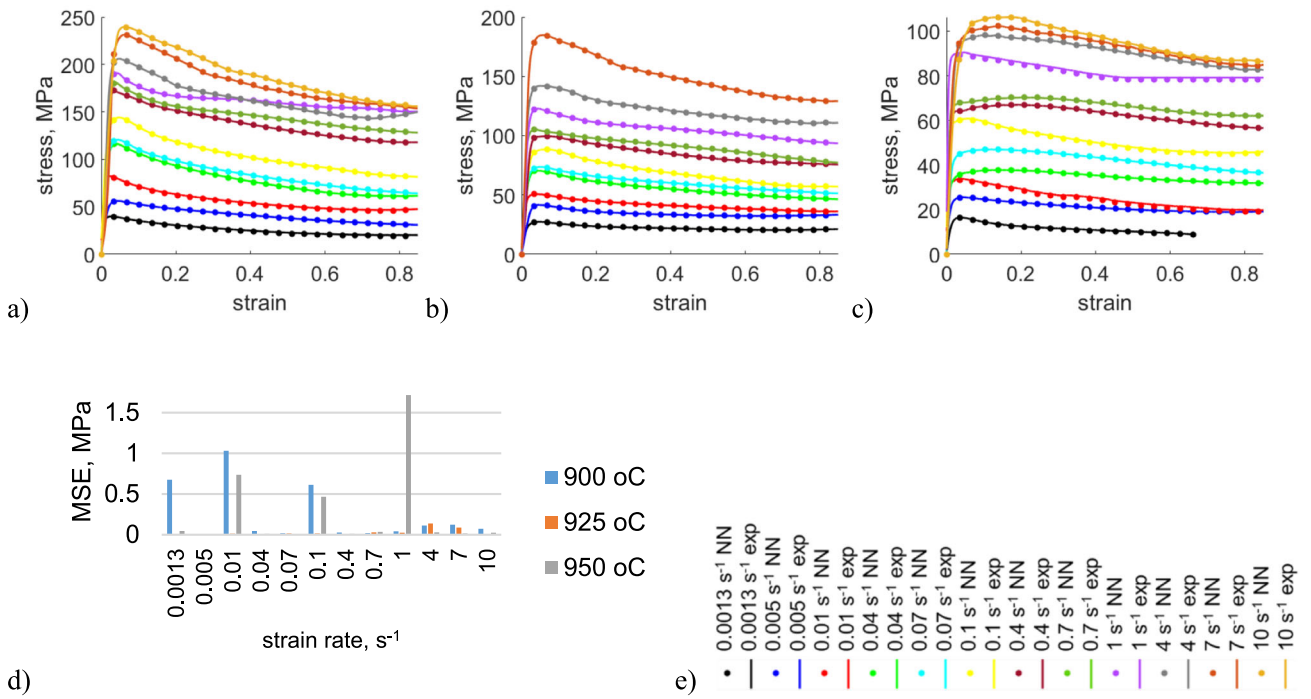


Fig. 14 Flow stress plots obtained by training the neural network for **a** 900 °C, **b** 925 °C, **c** 950 °C, and **d** MSE for all temperatures and strain rates according to the scale (**e**)

values), the maximum difference was reduced by a factor of four, to just 15 MPa (green line).

Regarding the derivatives of stress with respect to strain, a smooth prediction is observed across subsequent strain values, with only an initial sharp drop, which corresponds to the shape of the flow stress curve. These findings show that with more training data, a smoother transition between stress values is achieved. However, the results also indicate that increasing the amount of training data may still not be sufficient to achieve fully satisfactory and smooth predictions for cases not included in the training process.

Given the high cost and time required for experimental testing, another solution for improving the obtained results was analyzed. This time, an already trained neural network was used to predict the interpolant f* values for temperatures in the ranges of 900–925 °C and 925–950 °C. A linear interpolation was then applied to calculate the missing f* values at specific temperature limits. This approach enabled a smoother prediction of stress–strain curves, as shown in Fig. 20.

A significant improvement in the results can be seen in Fig. 21, where the calculated derivatives of stress with respect to temperature and strain are compared (blue lines). In this case, the maximum recorded difference in stress over a temperature step is 3 °C, which is 20 times smaller than that of the neural network trained with three temperature values (red line) and five times smaller than the neural network trained with 11 temperatures (green line). Additionally, an improvement in the stress derivatives with respect to strain is noticeable at low strain values, even though no additional strain data was introduced during training. As a result, this final solution was implemented and applied to finite element model calculations to assess its applicability in numerical simulations.

Finite element model implementation

For a strain increment $\Delta\varepsilon$, the explicit Euler integration scheme reads:

$$\sigma_{i+1} = \sigma_i + \theta_i \Delta\varepsilon \tag{4}$$

which can be implemented into constitutive models for metal plasticity, i.e., entered into the consistent material tangent.

By applying the interpolant f* that is calculated with the NN presented in Eq. (3), the dislocation density value can be determined, which is then used in Eq. (1) to calculate the flow stress value using $\alpha = 0.5$, $b = 2.95 \times 10^{-10}$ [m] and $G = (49.02 - 5.821/(\exp(181/(T + 273.15)) - 1))$ [GPa]. These values were taken from Varshni (1970).

The described flow stress model was implemented as a Lua subroutine in QForm, a software designed for numerical simulations of bulk-forming processes. Compression test simulations were conducted, comparing the neural network predictions with material data imported in tabular form.

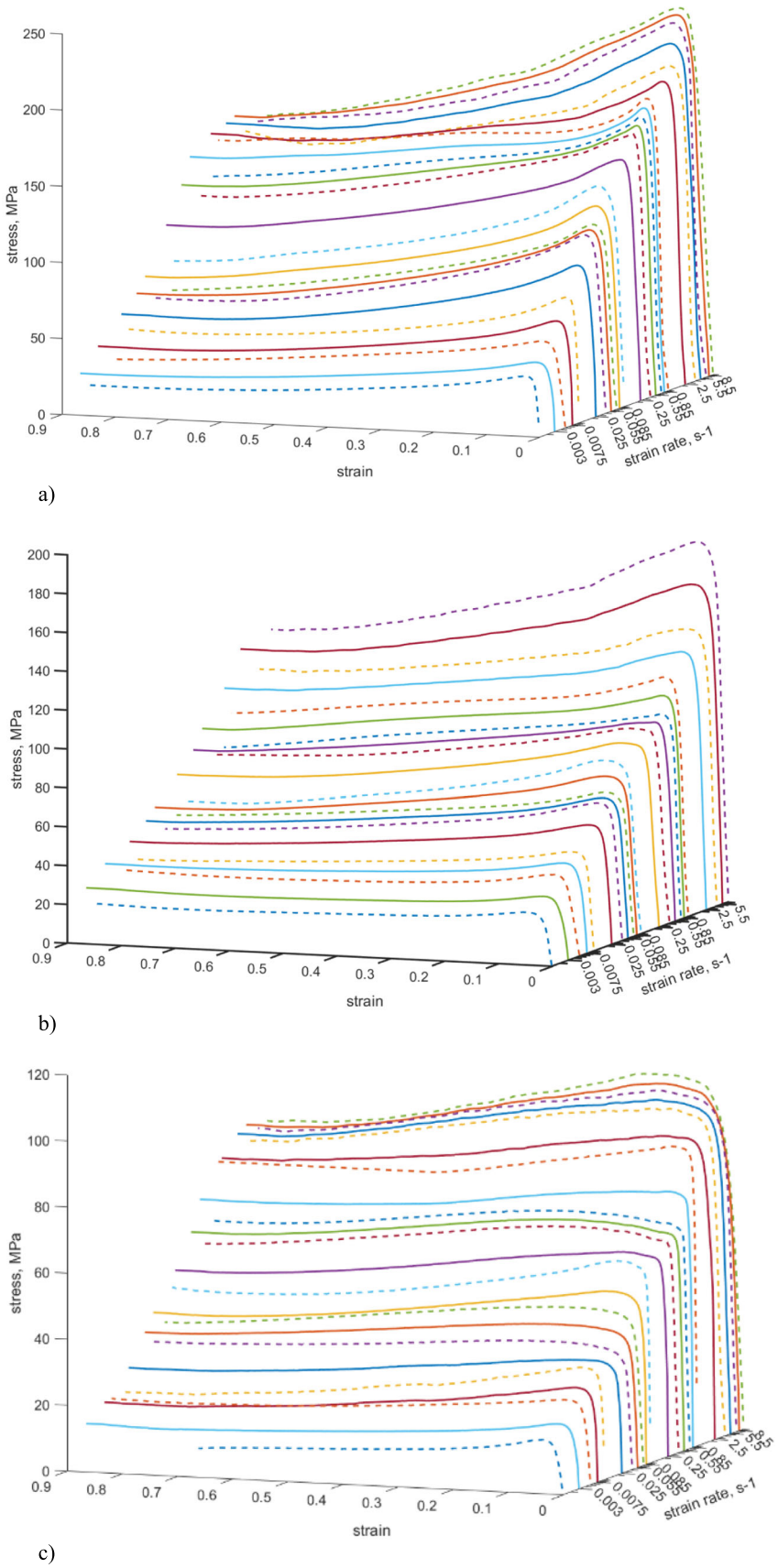


Fig. 15 3D flow stress plots were obtained from the trained neural network at **a** 900, **b** 925, and **c** 950 °C. Continuous lines represent the prediction of the NN for input data that was not used for the NN training. Dashed lines represent NN prediction for the input data used for its training process

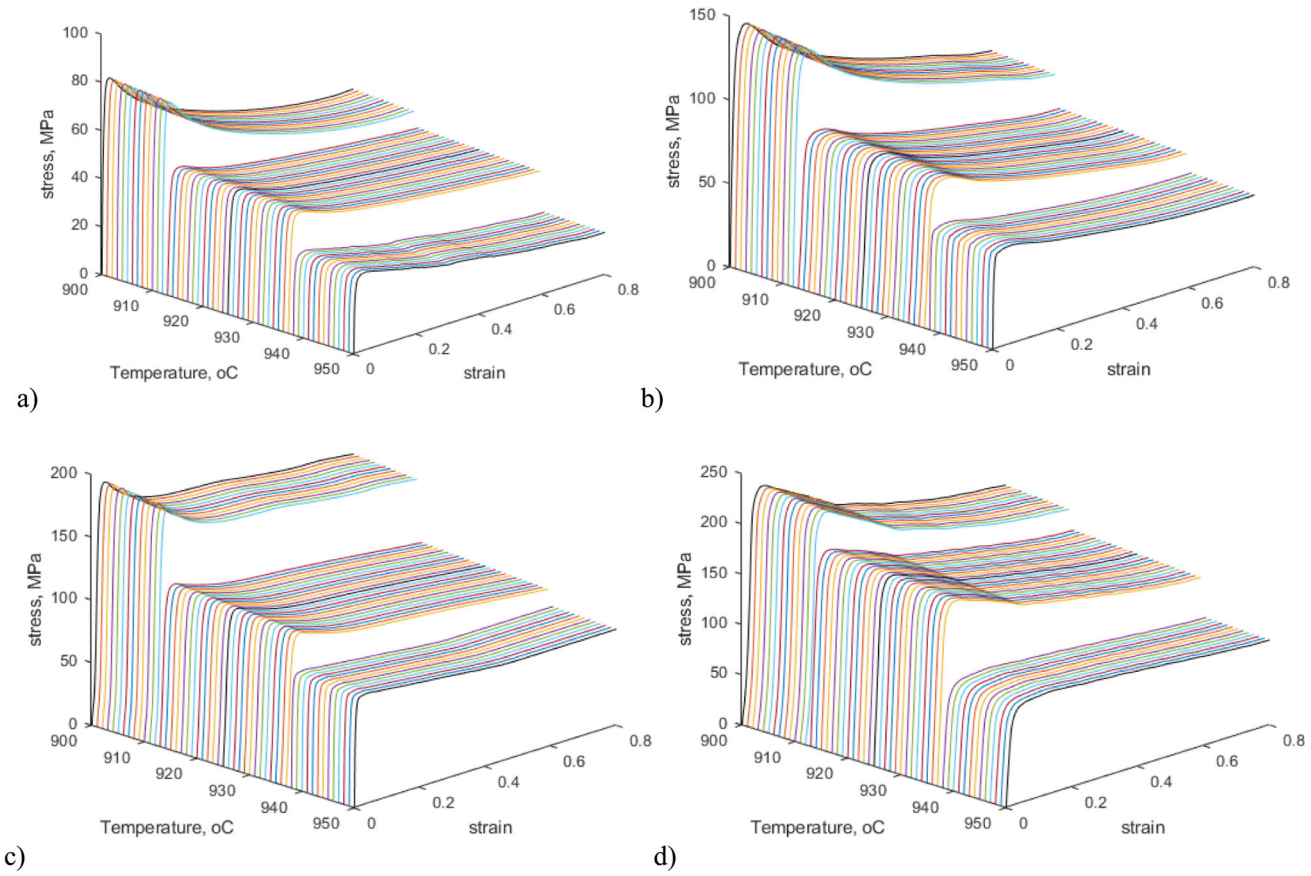


Fig. 16 3D flow stress plots at strain rates **a** 0.01, **b** 0.1, **c** 1 **d** 7 s^{-1}, and a wide range of temperatures, predicted with the neural network that was trained with only 3 temperature values

The simulations were performed under isothermal conditions (900, 925, and 950 °C) and a constant strain rate (0.0013, 0.005, 0.01, 0.04, 0.07, 0.1, 0.4, 0.7, 1, 4, 7, and 10 s^{-1}) during deformation, with velocity and time as variables. The resulting load–displacement plots (Fig. 22) demonstrate a satisfying and comparable material response. Minor deviations may be attributed to oscillations in strain rate values or the quality of the finite element mesh. Additionally, the average mean-squared error (MSE) is 0.0858, indicating very good predictive accuracy of the implemented neural network model. A detailed plot of the MSE for each simulation pair is shown in Fig. 23, where the maximum MSE is 0.61 kN (900 °C, 0.4 s^{-1}) and the smallest is less than 0.00005 kN (900 °C, 0.4 s^{-1}).

To investigate the influence of non-isothermal conditions on the quality of the results, two compression test simulations were performed, incorporating heat exchange between the sample and the surrounding air. The initial temperature of the sample was set to 925 °C, while the air temperature was set to 20 °C. The Coulomb friction coefficient between the sample and the tools was set to 0.09. The compression movement of the upper tool was controlled by a velocity–time function to maintain a strain rate of 1 s^{-1}. As shown in Fig. 24, both the neural network model and the tabular data-based model exhibit very similar temperature and strain distributions.

The internal temperature at the center of the sample increased from 925 to 933 °C, while the outer region cooled down to 919 °C. In terms of strain distribution, the neural network model predicted a slightly higher strain in the middle of the sample compared to the conventional tabular data model. When analyzing the stress–strain curve for a single point at the center of the sample, a good match between the two approaches is evident (Fig. 25a). The only notable difference is in the maximum strain value, which is 1.56 for the neural network solution and 1.38 for the tabular data approach. Furthermore, the load values recorded over time show an almost perfect match (Fig. 25b).

The implemented neural network solution within the FEM approach delivers highly accurate and reliable results, and it has the potential to be further enhanced by incorporating more advanced material characteristics. A well-developed neural network that provides information on flow stress and dislocation density distribution could be expanded to include, for example, a grain size evolution model that accounts for

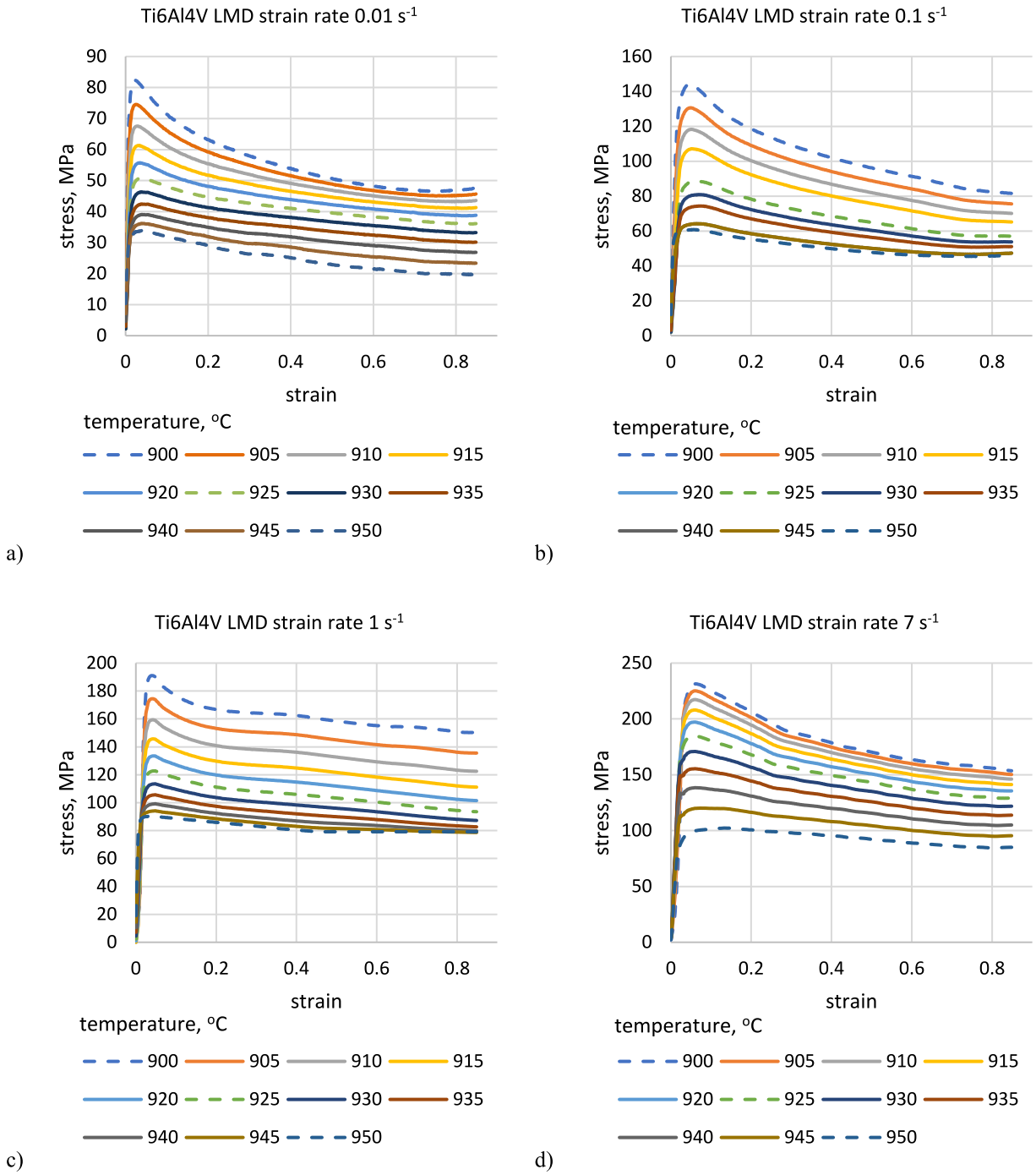


Fig. 17 Examples of flow stress plots obtained with the quadratic interpolation method at strain rate **a** 0.01, **b** 0.1, **c** 1 **d** 7 s^{-1} and a wide range of temperatures; continuous lines—original experimentally obtained curves, dotted lines—interpolated curves

the phenomenon of dynamic recrystallization. A correlation between dislocation density, stress, and grain size evolution is the subject of much research, as explored by Narutani and Takamura (1991), Song et al. (2024), Li et al. (2020), and Morris (2001). Incorporating the relevant equations into our NN-based model would enable the calculation of grain size based on dislocation density during the simulation, which could further be used to predict material properties such as hardness distribution.

However, it must be noted that from a computational time perspective, the neural network solution is significantly slower than the tabular data model. Under the same simulation parameters (FE mesh quality, time step size, etc.), the

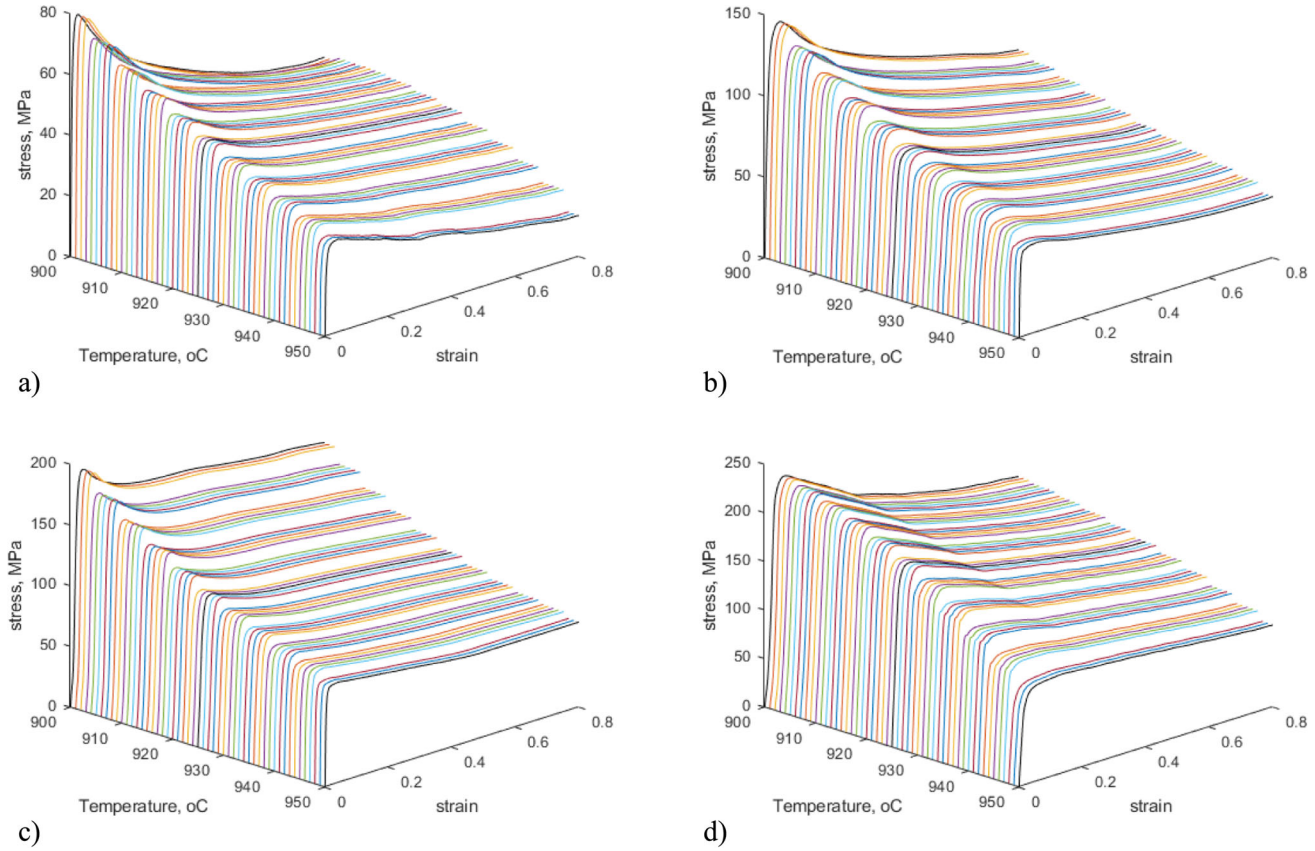


Fig. 18 3D flow stress plots at strain rates **a** 0.01, **b** 0.1, **c** 1 **d** 7 s^{-1}, and a wide range of temperatures, predicted with the neural network that was trained with 11 temperature values

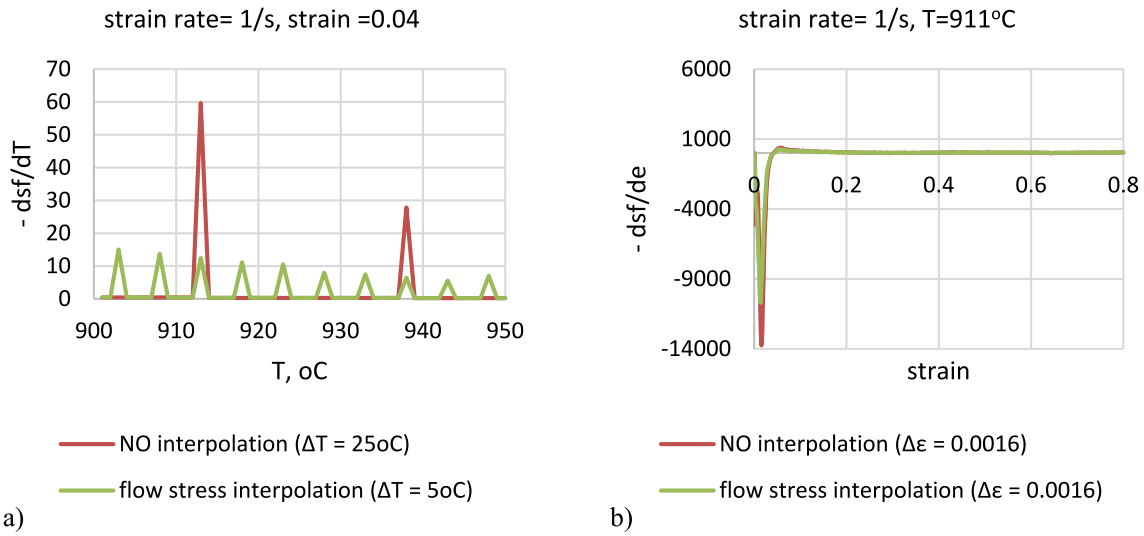


Fig. 19 Derivatives of stress over **a** temperature, and **b** strain after the NN training at 3 and 11 temperature values

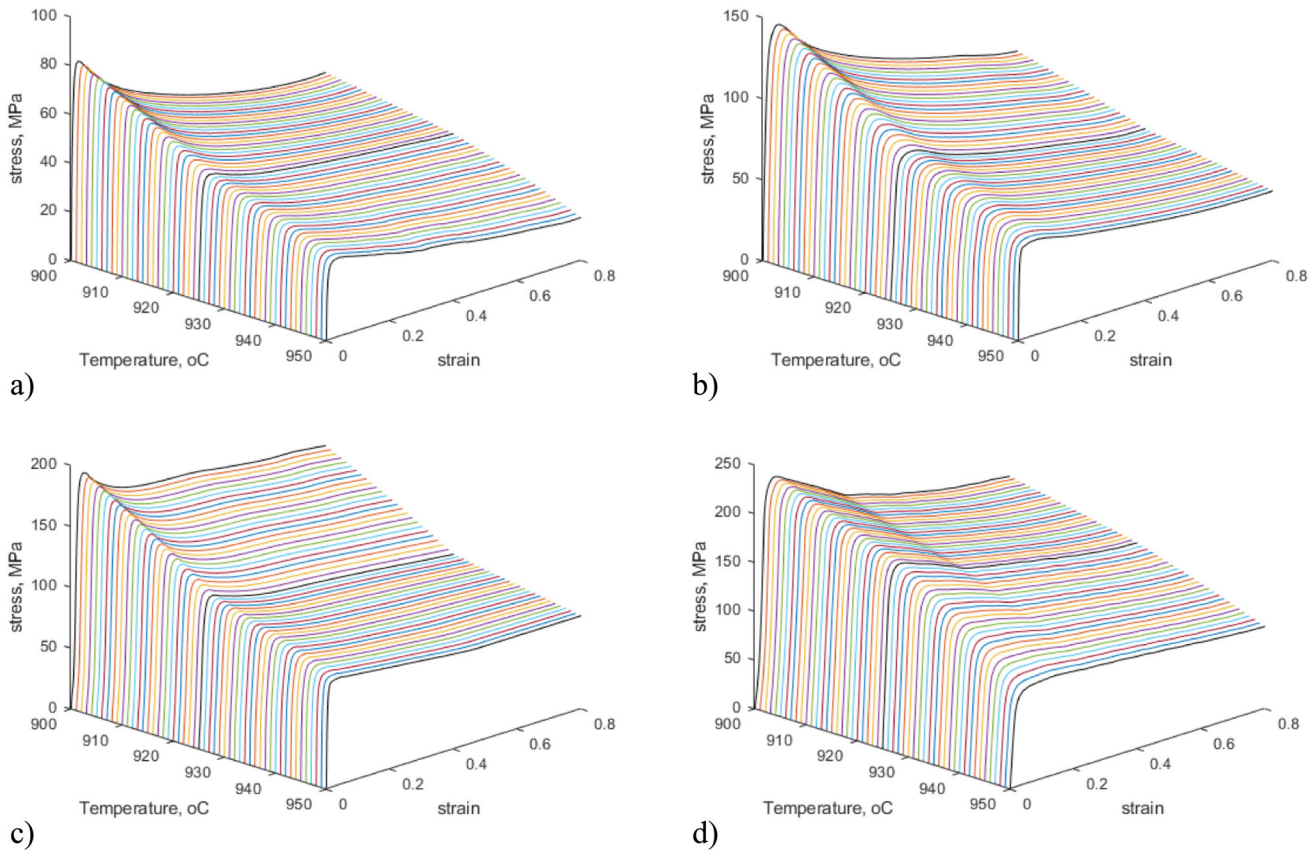


Fig. 20 3D flow stress plots were predicted by the neural network (trained with 3 temperature values) and applied linear interpolation method at **a** 0.01, **b** 0.1, **c** 1 **d** 7 s^{-1}, presented for a wide range of temperatures

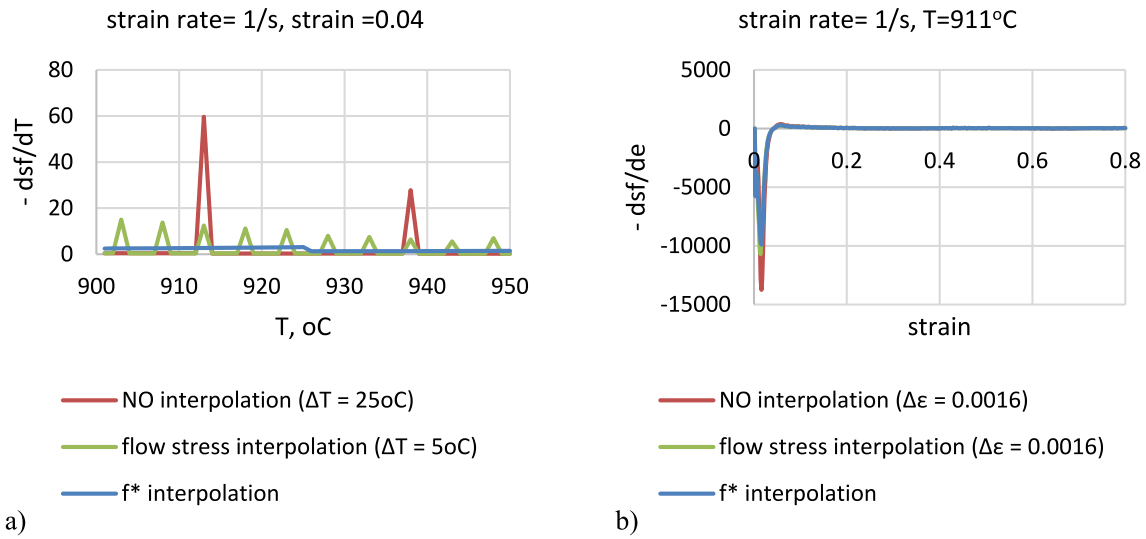


Fig. 21 Derivatives of stress over **a** temperature, **b** strain after the NN training at 3 and 11 temperature values, and interpolated f*

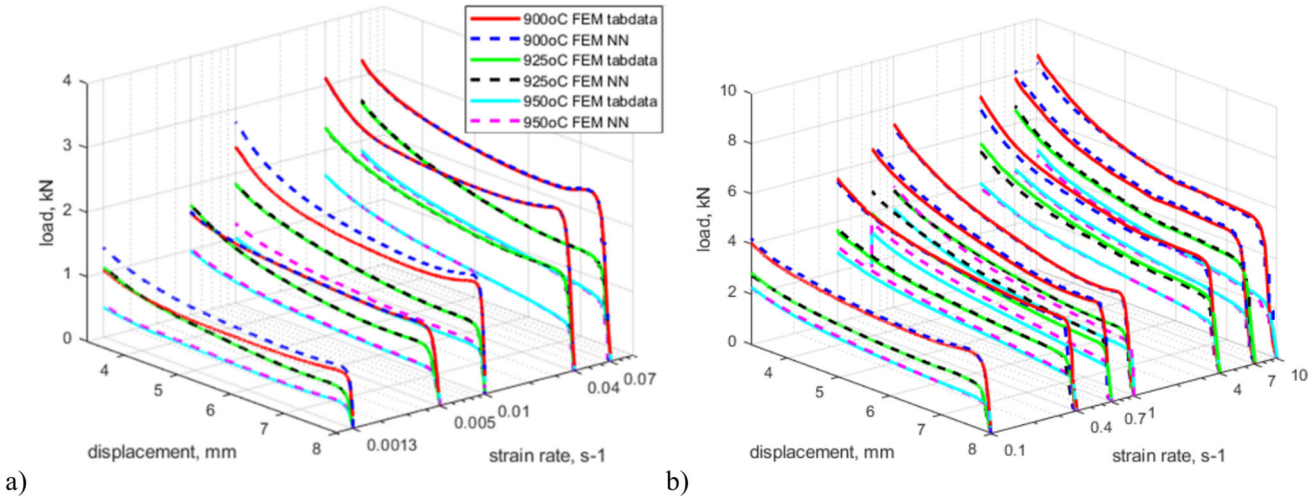


Fig. 22 Load–displacement plots after FEM simulation (tabular data—continuous lines, NN—dotted lines)

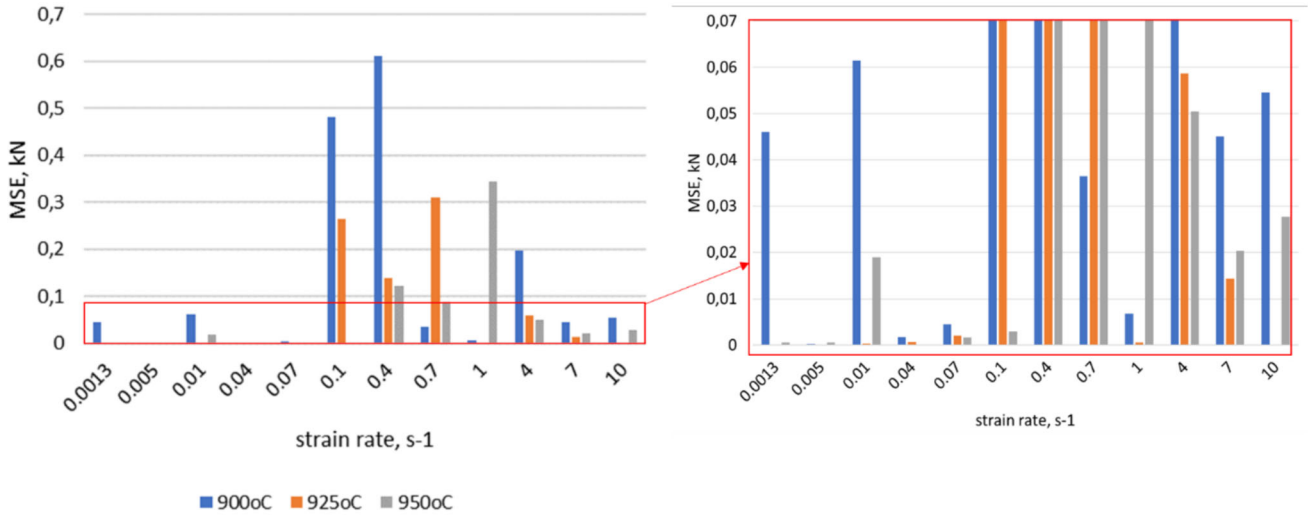


Fig. 23 MSE between NN and tabular data approach applied in FEM

neural network solution took 8 h, 8 min, and 42 s to complete the simulation, whereas the tabular data model took only 3 min and 7 s. This result raises concerns about the feasibility of using the neural network solution in combination with FEM simulations for industrial applications.

First, implementing the NN model in a more performant programming language instead of Lua would help speed up the simulation. This cannot be done on the user level in Qform, but could be accomplished by the software developers. In the current setting, the high computational time can be mitigated by training a Generalized Regression Neural Network with fewer data points—particularly by reducing the number of strain values considered. This approach would decrease the size of the matrices involved in the NN function, which is called whenever the finite element solver needs to update the stress values. However, a smaller dataset risks overfitting, i.e., the model may perform well on training data but poorly on unseen data.

An alternative strategy could involve implementing an algorithm that dynamically adjusts the hidden layer by pruning unnecessary neurons based on factors such as the identification error, as explored by Al-Mahasneh et al. (2017). Another possible improvement is leveraging more powerful computing hardware with greater memory capacity. This would allow for more efficient matrix operations, particularly when large datasets—sometimes generated via data augmentation—are necessary, which remains an area of active research.

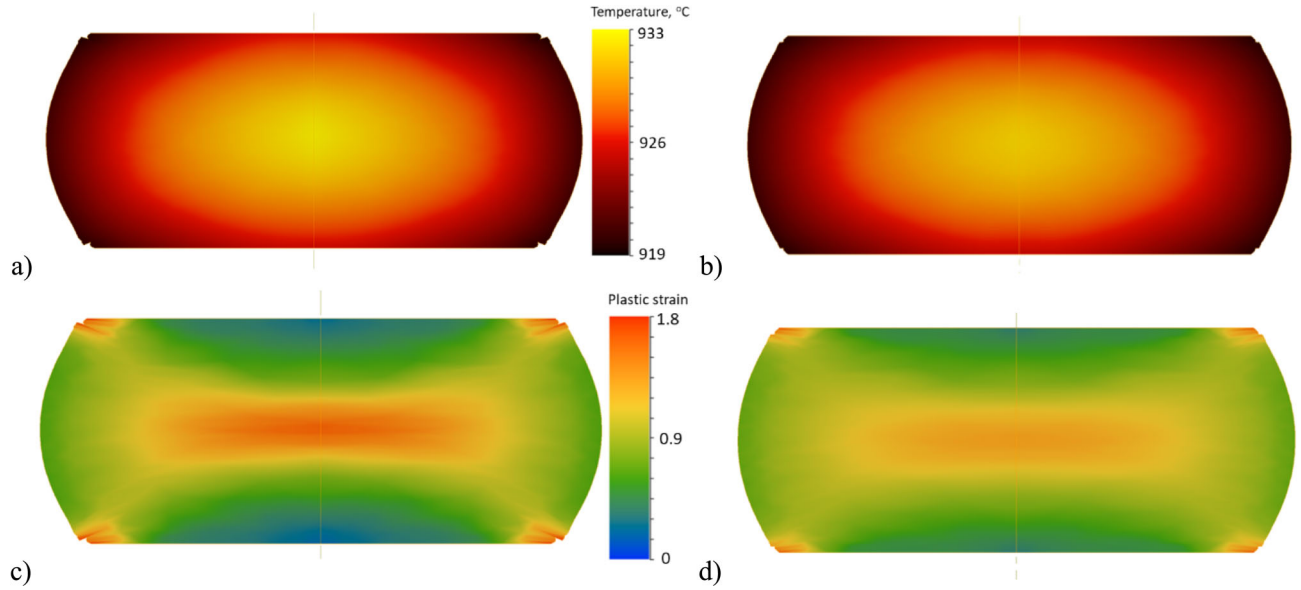


Fig. 24 Temperature and strain distributions are based on the NN approach (**a**, **c**) and tabular data (**b**, **d**)

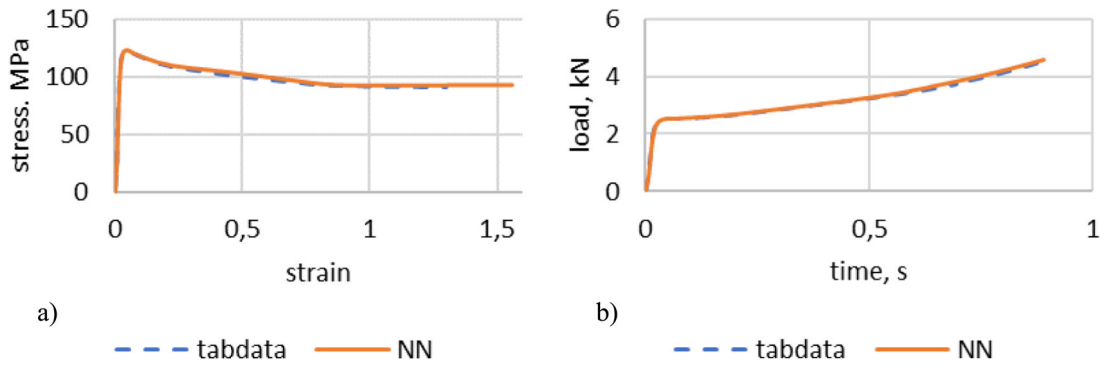


Fig. 25 FEM simulation results in the form of **a** a stress–strain plot at the selected point, and **b** a load-time plot

Discussion

The current research on neural ODE approaches for flow stress modeling faces several critical challenges. One major issue is data quality and quantity. Neural networks typically require large, high-quality datasets to accurately learn complex relationships. If the data used to train the model contains noise or is limited in scope (e.g., limited temperature or strain rate ranges), the model might struggle to generalize effectively beyond these conditions. Generating such datasets can be expensive and time-consuming.

Additionally, extrapolation poses a challenge, as neural networks, while effective at interpolating within the training range, often struggle with extrapolation. In flow stress modeling, this could result in poor model performance under extreme conditions of strain rate, temperature, or material state that are not well-represented in the training data. Another challenge is physical interpretability. Neural networks, especially in machine learning models, tend to function as "black boxes," which complicates the interpretation or validation of model predictions, particularly for physically significant quantities like dislocation density evolution. While the model framework and the contact transform used in this work introduce a layer of physics-based structure, the neural network component can obscure the physical meaning of predictions.

Computational efficiency is also a concern. The complexity of the neural network architecture and the number of parameters involved can make training and running the model computationally expensive, particularly when real-time predictions are required for applications such as process control. Especially the use of Neural Networks in conjunction with Finite Element Method simulations poses several challenges, particularly in industrial applications, as highlighted by Tufail et al. (2023), Barbierato and Gatti (2024), Khoei et al. (2023), and Mobarak et al. (2023).

Furthermore, the model's parameter sensitivity can be problematic. Its performance may be highly sensitive to the choice of tuning parameters in the neural network. Identifying and fine-tuning these parameters can be labor-intensive, and even small changes can lead to drastically different outcomes. Another issue is generalizability across materials. If the model is trained on data from specific alloys, such as Ti6Al4V, its ability to generalize to other materials may be limited. Retraining or significant adjustments might be necessary for different materials, reducing the model's broad applicability unless sufficient multi-material training data is available.

Finally, the convergence and stability of neural ODEs pose another challenge. Neural networks may introduce stability or convergence issues in dynamic systems like ODEs. The learned right-hand side function might lead to numerical instability, particularly in long-term simulations or highly sensitive systems.

Despite these challenges, neural ODE approaches with a GRNN have the potential to outperform traditional physics-based flow stress models, particularly in simulating complex material behavior. Unlike traditional physics-based models that rely on predetermined equations based on physical laws (e.g., constitutive equations), the GRNN-based neural ODE approach learns directly from data, enabling it to capture complex, nonlinear material behaviors that are often difficult or impossible to describe using standard equations. For example, in cases where flow stress depends on multiple, highly interdependent variables such as temperature, strain rate, and microstructural evolution, the neural ODE can offer a more flexible and adaptive modeling approach.

While physics-based models often require modifications or recalibration when applied to different material states or loading conditions (e.g., the initial microstructure of Ti6Al4V), a GRNN-based neural ODE, once trained on a comprehensive dataset, has the potential to generalize across a wider range of material states and operating conditions without the need for extensive re-parameterization. This leads to more efficient and adaptable modeling in real-world applications, where material behavior can vary under different process conditions. Additionally, physics-based models may struggle to incorporate complex dependencies between variables—such as strain hardening, dynamic recrystallization, or temperature effects—while a neural ODE can learn these relationships directly from data. The GRNN's ability to handle noisy data and approximate complex functional relationships makes it particularly suitable for modeling material behavior under nonlinear and multi-dimensional loading conditions.

Conclusions and outlook

A novel data-driven approach to strain hardening modeling has been presented in this work, utilizing a neural ODE framework. The key conclusions are as follows:

- The dynamics of strain hardening, including temperature and strain rate-dependent dislocation density evolution, can be effectively modeled using a neural ODE approach. This approach integrates a physics-based framework for dislocation density evolution with a contact transformation and data-driven learning of the function f* responsible for the evolution. Unlike classical models, the accuracy is not constrained by explicit mathematical formulations but rather by the quantity and quality of available data.
- In the case studied, 12 strain rate values were sufficient to train a regression neural network at a constant temperature, achieving a maximum MSE of less than 2 MPa between the predicted and experimental flow stress curves.
- For generalization across varying temperatures, a 25 °C difference between training data points was insufficient to achieve smooth predictions between two temperature ranges. Therefore, either additional training data or interpolation should be applied to predict f* values for untrained temperature ranges.
- Significant improvements in prediction accuracy were achieved by applying interpolation to the neural network-predicted f* values for temperature points not included in the training set. This method reduces the need for extensive temperature-specific data, limiting the required training dataset to just three temperature values.
- Combining a large dataset of experimental data with interpolation between predicted f* values for temperatures spaced 25 °C apart resulted in highly accurate predictions.
- The integration of the trained neural network into FEM software enabled reliable simulations of forming processes, demonstrating the potential of the neural ODE approach in practical applications.
- However, FEM simulations using the neural network model implemented in Lua were over 150 times slower than those using tabular material data. This indicates a need for more efficient implementation, and reduced-order versions of the neural ODE model to improve computational efficiency and enable practical industrial use.
- While the current model demonstrates the viability of neural ODE approaches for flow stress modeling in hot forming, it does not yet outperform conventional models. Its key advantage lies in its flexibility, as it is not limited to specific behaviors as physics-based models are. This flexibility can be exploited in industrial applications that require high accuracy by training the model on a representative dataset tailored to specific conditions.

The proposed approach can be further enhanced by incorporating a grain size evolution model that accounts for dynamic recrystallization. This extension would offer a valuable tool for industrial applications, enabling more accurate predictions of the final properties of formed parts and contributing to better process optimization and improved product quality in manufacturing. Future work will focus on parameter identification for more complex stress states, extending the model to other damage mechanisms, and closed-loop control of the process variables.

Acknowledgements The authors gratefully acknowledge the financial support provided by the Federal Ministry for Economic Affairs and Energy (BMWi) for the LUFO SAMT64 Project "Forging and additive manufacturing as a process combination for the resource-efficient production of aerospace structural components made of TiAl6V4 on flexible production scales" (20W1719D). M. Bambach acknowledges funding received by the German Research Foundation through the grant BA 4253/13-1. The authors gratefully thank Dr. Irina Sviridova from the Brandenburg University of Technology Cottbus-Senftenberg for performing a set of experimental tests.

Funding Open Access funding enabled and organized by Projekt DEAL. This work is supported by the Bundesministerium für Wirtschaft und Energie, 20W1719D.

Declarations

Conflict of interest The authors declare no conflicts of interest in this work.

References

Al-Mahasneh, A. J., Anavatti, S. G., & Garratt, M. A. (2017). Altitude identification and intelligent control of a flapping wing micro aerial vehicle using modified generalized regression neural networks. In *2017 IEEE symposium series on computational intelligence (SSCI)* (pp. 1–6). Honolulu. https://doi.org/10.1109/SSCI.2017.8280951

Bambach, M., Gerster, S., & Herty, M. (2021b). Online data assimilation of hybrid flow stress model by particle filtering. *CIRP Annals, 70*(1), 255–260. https://doi.org/10.1016/j.cirp.2021.04.002

Bambach, M., Sideris, I., Fabbri, M., & Wegener, K. (2022). Faster than real-time path-sensitive temperature modeling of wire-arc additive manufacturing by a data-driven finite volume method. *CIRP Annals, 71*(1), 189–192. https://doi.org/10.1016/j.cirp.2022.04.024

Bambach, M., Sizova, I., Szyndler, J., Bennett, J., Hayatt, G., Cao, J., Papke, T., & Merklein, M. (2021a). On the hot deformation behavior of Ti-6Al-4V made by additive manufacturing. *Journal of Materials Processing Technology, 288*, 116840. https://doi.org/10.1016/j.jmatprotec.2020.116840

Barbierato, E., & Gatti, A. (2024). The challenges of machine learning: A critical review. *Electronics, 13*(2), 416. https://doi.org/10.3390/electronics13020416

Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions of Information Theory, 39*(3), 930–945. https://doi.org/10.1109/18.256500

Belytschko, T., Liu, W. K., Moran, B., & Elkhodary, K. (2014). *Nonlinear finite elements for continua and structures*. John Wiley & Sons.

Bergström, Y. (1983). The plastic deformation of metals: A dislocation model and its applicability. *Reviews on Powder Metallurgy and Physical Ceramics, 2*, 79–265.

Beynon, J. H., & Sellars, M. C. (1992). Modeling microstructure and its effects during multipass hot rolling. *ISIJ International, 32*(3), 359–367. https://doi.org/10.2355/isijinternational.32.359

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2018). Neural ordinary differential equations. In *32nd conference on NeurIPS*. Montreal. https://doi.org/10.48550/arXiv.1806.07366

Chen, G. (2021). Recurrent neural networks (RNNs) learn the constitutive law of viscoelasticity. *Computational Mechanics, 67*, 1009–1019. https://doi.org/10.1007/978-0-387-75934-0

Demuth, H., & Beale, M. (1998). Neural network toolbox for use in MATLAB-user's guide. *The MathsWork, 4*, 7–9.

Estrin, Y., & Mecking, H. (1984). A unified phenomenological description of work hardening and creep based on one-parameter models. *Acta Metallurgica, 32*(1), 57–70. https://doi.org/10.1016/0001-6160(84)90202-5

Gorji, M. B., Mozaffar, M., Heidenreich, J. N., Cao, J., & Mohr, D. (2020). On the potential of recurrent neural networks for modeling path-dependent plasticity. *Journal of the Mechanics and Physics of Solids*. https://doi.org/10.1016/j.jmps.2020.103972

Habiba, M., & Pearlmutter, B. A. (2020). Neural ordinary differential equation-based recurrent neural network model. In *31st ISSC* (pp. 1–6). Letterkenny. https://doi.org/10.1109/ISSC49989.2020.9180182

Khoei, T. T., Slimane, H. O., & Kaabouch, N. (2023). Deep learning: Systematic review, models, challenges, and research directions. *Neural Computing and Applications, 35*, 23103–23124. https://doi.org/10.1007/s00521-023-08957-4

Kocks, U. F., & Mecking, H. (2003). Physics and phenomenology of strain hardening: The FCC case. *Progress in Materials Science, 48*, 171–273. https://doi.org/10.1016/S0079-6425(02)00003-8

Kusiak, J., & Pietrzyk, M. (1999). Artificial intelligence approach to the variable-based rheological model for steels. In *IEEE proceedings*, 2nd IPMM'99. Honolulu. https://doi.org/10.1109/IPMM.1999.791484

Li, H., Zhao, Z., Guo, H., Ning, Y., & Yao, Z. (2019). Dislocation density-based model for flow behavior of a near-a titanium alloy considering effects of initial lamellar thickness. *Journal of Materials Engineering and Performance, 28*, 2477–2487. https://doi.org/10.1007/s11665-019-04006-1

Li, W., Vittorietti, M., Jongbloed, G., & Sietsma, J. (2020). The combined influence of grain size distribution and dislocation density on hardness of interstitial free steel. *Journal of Materials Science and Technology, 45*, 35–43. https://doi.org/10.1016/j.jmst.2019.11.025

Liu, X., Tian, S., Tao, F., & Yu, W. (2021). A review of artificial neural networks in the constitutive modeling of composite materials. *Composites, Part b: Engineering*. https://doi.org/10.1016/j.compositesb.2021.109152

Mei, H., Lang, L., Yang, X., Liu, Z., & Li, X. (2020). Study on constitutive relation of nickel-base superalloy Inconel 718 based on long short term memory recurrent neural network. *Metals, 10*, 1588. https://doi.org/10.3390/met10121588

Mobarak, Md. H., Mimona, M. A., Islam, Md. A., Hossain, N., Zohura, F. T., Imtiay, I., & Rimon, Md. I. H. (2023). Scope of machine learning in materials research—A review. *Applied Surface Science Advances, 18*, 100523. https://doi.org/10.1016/j.apsadv.2023.100523

Morris, J. W. (2001). The influence of grain size on the mechanical properties of steel. Report, Berkeley. University of North Texas Libraries, UNT Digital Library; crediting UNT Libraries Government Documents Department. https://doi.org/10.2172/861397; Retrieved from https://digital.library.unt.edu

Narutani, T., & Takamura, J. (1991). Grain-size strengthening in terms of dislocation density measured by resistivity. *Acta Metallurgica Et Materialia, 39*(8), 2037–2049. https://doi.org/10.1016/0956-7151(91)90173-X

Norton, F. H. (1929). *Creep of steel at high temperatures*. McGraw-Hill Book Co.

Phaniraj, M. P., & Lahiri, A. K. (2003). The applicability of neural network model to predict flow stress for carbon steels. *Journal of Materials Processing Technology, 141*, 219–227. https://doi.org/10.1016/S0924-0136(02)01123-8

Rao, K. P., & Prasad, Z. K. D. V. (1995). Neural network approach to flow stress evaluation in hot deformation. *Journal of Materials Processing Technology, 53*, 55–566. https://doi.org/10.1016/0924-0136(94)01744-L

Reddy, N. S., Lee, Z. H., Park, C. H., & Lee, C. S. (2008). Prediction of flow stress in Ti-6Al-4V alloy with an equiaxed α+β microstructure by artificial neural networks. *Materials Science and Engineering A, 492*, 276–282. https://doi.org/10.1016/j.msea.2008.03.030

Rihi, A., Baina, S., Mhada, F.-Z., Bachari, E. E., Tagemouati, H., Guerboub, M., Benzakour, I., Baina, K., & Abdelwahed, E. H. (2024). Innovative predictive maintenance for mining grinding mills: From LSTM-based vibration forecasting to pixel-based MFCC image and CNN. *The International Journal of Advanced Manufacturing Technology, 135*, 1271–1289. https://doi.org/10.1007/s00170-024-14588-3

Rihi, A., Baina, S., Mhada, F.-Z., Elbachari, E., Tagemouati, H., Guerboub, M., & Benzakour, I. (2022). Predictive maintenance in mining industry: Grinding mill case study. *Procedia Computer Science, 207*, 2483–2492. https://doi.org/10.1016/j.procs.2022.09.306

Rius, A., Ruisánchez, I., Callao, M. P., & Rius, F. X. (1998). Reliability of analytical systems: Use of control charts, time series models and recurrent neural networks (RNN). *Chemometrics and Intelligent Laboratory Systems, 40*, 1–18. https://doi.org/10.1016/S0169-7439(97)00085-3

Siewior, H., & Madej, L. (2021). A repeatability study of artificial neural network predictions in flow stress model development for magnesium alloy. *Computer Methods in Materials Science, 21*(4), 209–218. https://doi.org/10.7494/cmms.2021.4.0769

Song, E., Andani, M. T., & Misra, A. (2024). Investigation of grain size and geometrically necessary dislocation density dependence of flow stress in Mg-4Al by using nanoindentation. *Acta Materialia, 265*, 119633. https://doi.org/10.1016/j.actamat.2023.119633

Stendal, J. A., Bambach, M., Eisentraut, M., Sizova, I., & Weiβ, S. (2019). Applying machine learning to the phenomenological flow stress modeling of TNM-B1. *Metals, 9*(2), 220. https://doi.org/10.3390/met9020220

Tac, V., Costabal, F. S., & Tepole, A. B. (2022). Data-driven tissue mechanics with polyconvex neural ordinary differential equations. *Computer Methods in Applied Mechanics and Engineering, 398*, 115248. https://doi.org/10.1016/j.cma.2022.115248

Tac, V., Kuhl, E., & Tepole, A. B. (2024). Data-driven continuum damage mechanics with built-in physics. *Extreme Mechanics Letters, 71*, 102220. https://doi.org/10.1016/j.eml.2024.102220

Tac, V., Rausch, M. K., Costabal, F. S., & Tepole, A. B. (2023). Data-driven anisotropic finite viscoelasticity using neural ordinary differential equations. *Computer Methods in Applied Mechanics and Engineering, 411*, 116046. https://doi.org/10.1016/j.cma.2023.116046

Taylor, G. (1934). The mechanism of plastic deformation of crystals: Part I—Theoretical. *Proceedings of the Royal Society of London: Series a, Containing Papers of a Mathematical and Physical Character, 145*, 362–387.

Tufail, S., Riggs, H., Tariq, M., & Sarwat, A. I. (2023). Advancements and challenges in machine learning: A comprehensive review of models, libraries, applications, and algorithms. *Electronics, 12*(8), 1789. https://doi.org/10.3390/electronics12081789

Varshni, Y. P. (1970). Temperature dependence of the elastic constants. *Physical Review b., 2*, 3952–3958. https://doi.org/10.1103/PhysRevB.2.3952

Wang, F., Zhao, J., & Zhu, N. (2016). Constitutive equations and ANN approach to predict the flow stress of Ti-6Al-4V alloy based on ABI tests. *Journal of Materials Engineering and Performance, 25*, 4875–4884. https://doi.org/10.1007/s11665-016-2337-4

Zerilli, F. J., & Armstrong, R. W. (1987). Dislocation-mechanics-based constitutive relations for material dynamics calculations. *Journal of Applied Physics, 61*, 1816–1825. https://doi.org/10.1063/1.338024