Machine Learning Assisted Optimization of Hardening Parameters

Crista Ekblom, F. San Koktas, Akın Erdem

October 2023

1 Introduction

This section gives the motivation for the research conducted and presented in this report (Section 1.1). Following this the specific research question is presented and elaborated in Section 1.2.

1.1 Motivation

The development of new materials for the utilization of different industries is of great importance. These materials need to have certain kinds of properties to be of use. For instance, the automotive industry requires lightweight metals. Knowing the properties of the materials is essential. As an example, in the context of the automotive industry knowing the properties of the metals used for the building of the car is important when calculating how durable the car is in a crash scenario. One component of knowing the material properties is to find the parameters of so-called constitutive equations. There are different kinds of constitutive equations used depending on e.g., what kind of hardening of the material is desired to be investigated.

Traditionally, these parameters have been identified with methods that require significant computational time. Examples of these are Constitutive Equation Gap Method (CEGM), Virtual Fields Method (VFM) and Finite Element Model Updating (FEMU). To make the parameter identification process faster, applying machine learning methods to solve the inverse problem has become more prominent in recent research. Machine learning has developed fast in the past decade and there are many algorithms suitable for the solving of the inverse problem that calibration of the parameters is.

Therefore, it is of interest to develop a machine learning algorithm for the purpose of the identification of the hardening parameters, which will be the focus of the research conducted and presented in this report. The following section goes in-depth about the specific research question that is aimed to be answered.

1.2 Research Question

The aim of the research presented in this report is to develop a machine learning-based model that can solve the inverse problem and calibrate the hardening parameters. The purpose is to use experimental data generated with software Abaqus for the training of the machine learning model. The dataset was provided and consisted of four different kinds of geometries for the steel material DP1000 with force and displacement. Therefore, the research questions this report aims to seek an answer to:

- Can a machine learning model be used to calibrate the hardening parameters for steel DP1000?
- How do the results of the developed machine learning model compare with a more traditional method of solving an inverse problem?

2 Theoretical Background

This section reviews the theoretical background to the research conducted and presented in this report. Section 3.1 discusses the hardening laws utilized in the research for this paper, following a brief introduction to what inverse methods are in Section 3.2. Section 3.3 gives a short account of what machine learning entails. Finally, Section 3.3 reviews the literature related to the machine learning assisted optimization of hardening parameters.

2.1 Hardening Laws

A hardening law (or rule) "describes the relationship of stress–strain to increase further the amount of stress to produce additional strain after having reached the elastic limit" [1]. There are different kinds of hardening rules. These differ based on what kind of hardening is desired to be investigated. Examples are kinematic and strain hardening. The hardening laws utilized in the research conducted for this paper are the Swift, Voce and Swift-Voce hardening laws. These laws can be used to describe the expansion of yield surface or in other words isotropic hardening [2]. These are each explained briefly in the following subsections.

2.1.1 Swift

The Swift hardening law is one of the most used models to describe flow stress-strain behavior in sheet metal forming simulations [3]. The law has three parameters and can be described by the below equation:

$$\sigma(\varepsilon) = C_1 \cdot (C_2 + \varepsilon)^{C_3}$$

2.1.2 Voce

The Voce hardening law differs from the Swift hardening law in that it has four parameters. The law can be described with the below equation:

$$\sigma(\varepsilon) = C_1 + (C_2 - C_1) \cdot \exp\left(-C_3 \cdot \varepsilon^{C_4}\right)$$

2.1.3 Swift-Voce

The Swift-Voce hardening law is an extension of both the Swift and Voce. This hardening law combines both laws into one. The law has in total of seven parameters. This law can be seen described as below:

$$\sigma(\varepsilon) = C_1 \cdot \sigma_{\text{Swift}}(\varepsilon) + (1 - C_2) \cdot \sigma_{\text{Voce}}(\varepsilon)$$

2.2 Inverse Method

Inverse method is a type of technique where the input parameters for a model are estimated (with uncertainty) from comparing the model output magnitudes with experimental data [4]. This is the opposite of a direct or forward problem, where the parameters are known, and the predictions of observations can be predicted through the help of e.g., physical laws. The inverse methods can be utilized for instance for the calibration of identifying parameters for flow curves of materials. This is also what is the aim of the research conducted in this paper. Other applications where inverse problems are present are for example computerized tomography, seismic imaging, electron microscopy and magnetic resonance imaging [5]. Figure 1 displays the difference between the forward and the inverse problem and clearly shows how these two types of problems are opposites.

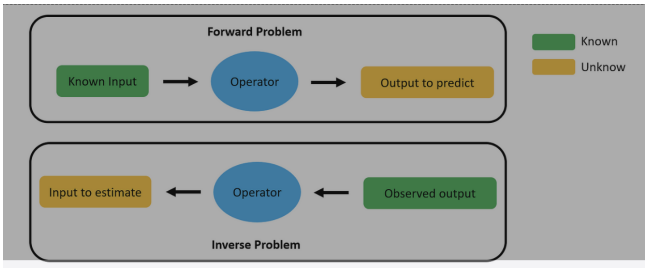

Figure 1: Inverse Problem vs Forward Problem [6].

2.3 Machine Learning

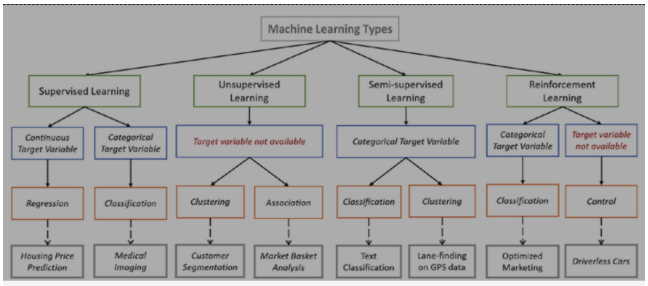

Figure 2: Machine Learning Categorized [8].

In recent years machine learning (ML) has become something of a hot topic and is currently being utilized and applied in many fields of research [7]. Terms like artificial intelligence and data mining have been used interchangeably with it. The concept entails the development of a computer system that learns and

improves from data without having to be explicitly programmed.
Machine learning algorithms can be classified in different types based on the learning process of the model. The four major categories are unsupervised learning, semi-supervised learning, supervised learning, and reinforcement learning [7]. Figure 2 displays these categories and what type of machine learning algorithms fall under each category with examples of applications.

2.4 Literature Review

Knowing the mechanical properties of materials is of increased importance – especially with the development of new materials for the use of e.g., automotive and aerospace industries [9]. To identify for example the hardening parameters of a material methods like Constitutive Equation Gap Method (CEGM), Virtual Fields Method (VFM) and Finite Element Model Updating (FEMU) have been utilized to solve the inverse problem where the displacement field has been obtained and is used as input [10]. The problem with these methods is that they cannot handle complex heterogeneity or thickness variations in the material and the cost of computations can become very high [10]. This is why with the fast development of machine learning, applying ML methods for the calibration of hardening parameters has become more prominent in research in recent years. The motivation for this has to do with decreased cost and ability to have the calibration done faster than with other types of methods used for solving inverse problems [10].

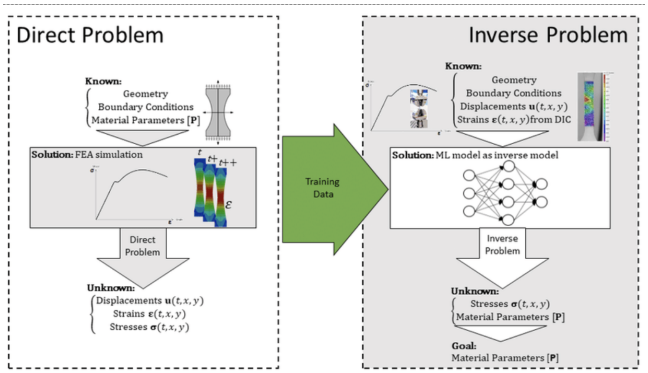

Figure 3: ML as inverse problem for calibration of parameters [10]

Lourenço et al. [11] describe different scenarios in metal forming applications where machine learning algorithms can be utilized. One of the applications they discuss is the application of machine learning for parameter identification with inverse modelling. A case study is conducted where an ANN-based algorithm is developed to find the parameters for Swift's hardening law and Hill48 anisotropic criterion. What is remarked by the authors is that the reliability of the developed model is highly dependent on the quantity and quality of the

training data used. The process of this case study conducted by Lourenço et al. can be seen illustrated in Figure 3.

Yao et al. [12] conduct a similar kind of process with identifying the parameters for the Swift hardening law for 6061 aluminum alloy with the help of backpropagation neural network algorithm. Data from different specimens with different geometries of the researched aluminum alloy were used for the developed neural network model. It was found that the model could predict quite well the parameters for the material in comparison to an experimental-numerical hybrid model. The significant benefit was the decrease in required computational time where CPU time for the machine learning model was 192.5 h in comparison to the experimental-numerical hybrid method where this was 296.7 h.

Zhang et al. [10] develop a machine learning model for the mechanical characterization of heterogenous membranes. They compare their results with the results obtained by traditional FEMU. A light weighted Fully Connected Neural Network (FCNN) is utilized to solve the inverse problem where the goal is to identify the Young's moduli for the material. It is found that the FCNN model can predict the value as accurately as the FEMU methodology, but the computational time is 6 orders of magnitude less than for the FEMU. The prediction ability is seen to improve in the future by the authors with the improvement of the structure to the used neural network algorithm making the use of this kind of methodology for the calibration of the material parameters even more compelling.

Neural network-based algorithms are often favored in research as the choice of machine learning algorithm for the parameter identification as demonstrated by the already mentioned examples. Others that have used similar type of methodology with developing a neural network model are Lin et al. [13], Guo et al. [14], Wang et al. [15], Chen et al. [16] and Hajari et al. [17]. Liu et al. [18] reviews the use of ANN in constitutive modelling for composite materials and state that the reasons why this type of machine learning algorithms are used widely in research for material behavior have to do with the fact that these algorithms perform well with growing data, are able to approximate complex nonlinear relations and there are many advanced open-source libraries for this type of algorithms (e.g. Tensorflow and PyTorch).

An alternative machine learning algorithm to an ANN-based one for material parameter identification is presented by Bastos et al. [19]. They use Extreme Gradient Boosting (XGBoost) model for the parameter calibration to find the Swift law parameters. The model is tested with multiple different sizes of datasets. It is found that the results improve with the increased number of samples in the training dataset, but that 1500 sample set already gives good results from the model. To improve the computational time of the model since the time required for training was huge, a so-called Principal Component Analysis (PCA) was conducted to reduce the dimensionality of the dataset used for the training.

Utilizing machine learning in material parameter identification is still a relatively new development. Improvements to the models are to be expected with

the further development of machine learning methods. At current ANN-based models are dominant within the field of parameter identification for constitutive modelling, although as Bastos et al. [19] demonstrate alternative methods can be deployed. The focus for the research presented in this paper is to develop an ANN-based algorithm to identify the hardening parameters for DP1000 steel sheet with different geometries. The following section details the methodology for the research conducted.

3 Methodology

This section presents first in Section 4.1 the workflow for the research conducted. Following this Section 4.2 describes the data utilized and finally Section 4.3 discusses the configuration of the used machine learning models.

3.1 Data

We were provided with initial estimates for hardening parameters, iteration-based guesses for hardening parameters, and anticipated force-displacement profiles at both 25 degrees Celsius and 400 degrees Celsius. The data organization was structured as follows:

```
data for training/
  (hardening parameters) initial guesses/
  (hardening parameters) iteration guesses (400C)/
  25C expected force-displacement curves/
  400C expected force-displacement curves/
```

Each of the four subfolders contained data for CHD6, NDBR6, NDBR20, and NDBR50 geometries, primarily comprising extensive .npy (NumPy) and .csv files. Cleaning the data, particularly the .npy files, posed certain challenges.

In order to gain insight into the .npy file structures, we employed a process of printing and inspecting arrays. These arrays followed a consistent pattern, starting with c1, c2, c3, c4, c5, c6, and c7 values representing hardening parameters, followed by dictionaries for force and displacement. This pattern recurred consistently throughout the data.

Working with the data stored in both .csv and .npy files presented challenges due to discrepancies in the number of columns and rows across the files we processed. To address this, we undertook the following steps:

1. For .npy files, we standardized them to contain 7 columns, representing c1 to c7. For .csv files, we ensured they had 2 columns, one for force and one for displacement.

2. We balanced the number of rows across both file types by duplicating data points within the force-displacement datasets.

3. We converted the standardized .npy files into .csv format to enhance compatibility with our local environment, such as Visual Studio. .csv files provide a more accessible means of visualization and analysis.

With an equal number of rows in both datasets, we proceeded with our data analysis. Our intended output focused on the hardening parameters (c1 to c7), which we placed on the Y-axis. Simultaneously, the input features encompassing force and displacement values were plotted on the X-axis within our neural network model. This approach allowed us to perform effective analysis and modeling of the data.

3.2 Configuration of the Model

We followed a structured, iterative process, with each phase building upon the lessons learned from the preceding one.

Phase 0: Basic Neural Network Model

- In the initial phase, the primary objective was to grasp the fundamentals of neural network architecture.
- We established a simple neural network model, designed not for actual parameter prediction, but rather to acquire foundational knowledge about constructing neural networks.
- Random data was used, and our focus lay on recording training data, predictions, and loss dynamics throughout 1000 epochs.

Phase 1: Simple Dummy Model for Hardening Parameters

- With a foundational understanding of neural networks, we embarked on creating a simple dummy model with a specific aim: predicting hardening parameters within different geometries.
- The complexity of operating on more intricate datasets led us to continue using randomized data.
- Our analytical focus encompassed recording training loss, comparing hardening parameters across diverse geometries, investigating the impact of learning rates, examining loss convergence dynamics, and evaluating the accuracy of predicted values against actual data.

Phase 2: Optimization of the Dummy Model

- Armed with insights from the initial dummy model, we diligently optimized its architecture and performance.
- By refining the model, we observed an improvement in training loss, enhancing its predictive accuracy.

- This optimization process underlined the capacity of neural networks to continually enhance their performance.

Phase 3: Transition to Real Data

- In the subsequent phase, we took a significant leap from synthetic to authentic data.
- For a uniform geometry, we compared actual data to model predictions for individual parameters, adopting a multi-output regression approach.
- To enhance visualization, Savitzky-Golay smoothing was applied to the data curves.
- Training loss based on real data was closely monitored.

Phase 4: Integration of Real Data and Geometries

- Building upon the real-data model, we extended our approach to accommodate various geometries, each with its distinct material properties.
- Our endeavor involved tracking training loss over 1000 epochs and predicting the hardening parameters for each geometry.
- We explored learning rates and loss convergence behavior for different geometries.
- Verification of predicted values was conducted, initially focusing on 'CHD6' geometry. This framework can be duplicated for other geometries.

3.2.1 Phase 0: Basic Neural Network Model

Data Generation In our work, we began by generating the dataset that our neural network model would learn from. The 'axis' variable represents our input data, which is a column vector consisting of 500 points along the x-axis. We carefully defined the range from 0 to 10, ensuring a wide spectrum of input values. To create the corresponding output data, we applied a specific mathematical function:

$$f(x) = x \cdot \cos(x)$$

This function takes each 'x' value and computes the corresponding 'y' value by multiplying it with its cosine. These 'ordinate' values serve as our expected outputs, which the neural network aims to predict.

Neural Network Architecture In the architecture of our neural network, we made several key design choices. Our objective was to build a model that could capture the underlying relationship between input and output. Here's how we structured it:

- Input Layer: The first layer of our network is the input layer, representing the single input feature, 'x.' It uses the Rectified Linear Unit (ReLU) activation function. This layer serves as the entry point for our data.
- Hidden Layers: To enable the network to capture complex patterns, we included two hidden layers, each consisting of 64 neurons. These layers use the ReLU activation function, which introduces non-linearity and flexibility into the model, enabling it to learn intricate relationships within the data.
- Output Layer: The output layer is critical for regression tasks. It comprises a single neuron, reflecting the one-dimensional nature of our output. Importantly, this output neuron does not employ any activation function, making it suitable for regression.

Model Compilation To train the neural network effectively, we compiled it with specific settings tailored to our problem. These settings govern how the network learns from the data:

- Optimizer: We selected the 'adam' optimizer, which is a widely used choice for gradient-based optimization. The 'adam' optimizer adapts the learning rate during training, making it effective for a wide range of problems.
- Loss Function: For our regression problem, we chose the 'mean squared error' loss function. This function quantifies the difference between the model's predictions and the actual output by calculating the mean of the squared errors. Minimizing this loss is the primary goal during training.

Training With the neural network architecture and settings in place, we initiated the training process. This involved presenting the network with our 'axis' data as input and the corresponding 'ordinate' values as the target output. We ran the training for 100 epochs, allowing the model to iteratively adjust its internal parameters (weights and biases) to minimize the mean squared error.

Predictions After completing the training process, we put our model to the test by making predictions. Using the same 'axis' data, we employed the trained model to predict the 'ordinate' values. These predictions represent the network's estimation of the function $f(x) = x \cdot \cos(x)$ based on the patterns it has learned from the training data.

Visualization To gain a visual understanding of our model's performance, we created a plot. This plot showcases both the original function $f(x) = x \cdot \cos(x)$ (in purple) and the model's predictions (in blue). By comparing these two curves, we can assess how effectively our neural network has approximated the underlying function across the range of 'x' values. This visualization provides valuable insights into the model's accuracy and its ability to capture the complex relationship within the data.

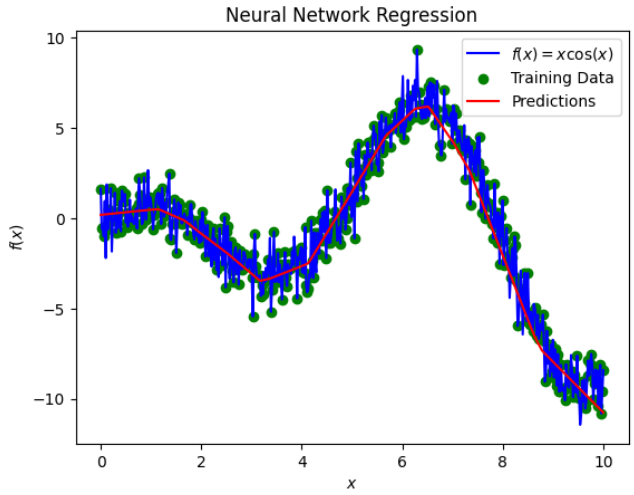

Figure 4: Neural Network Regression of Phase 0 Model

Figure 5: Training Loss of Phase 0 Model

3.2.2 Phase 1: Simple Dummy Model for Hardening Parameters

In Phase 1 of our project, we maintained continuity with the neural network architecture, data generation process, optimizer, loss function, training methodology, and prediction procedure used in Phase 0. However, we introduced different geometries and incorporated the consideration of hardening parameters.

To identify the hardening parameters, we created a paramConfig dictionary that included information such as parameter names, permissible lower and upper bounds, exponents, units, and types. Then, we introduced an array of diverse geometries, each of which was associated with a specific yielding index. This approach allowed us to tailor our efforts to the unique characteristics of each ge-

ometry. For every geometry, we executed backpropagation, computing gradients and calibrating the neural network's parameters. We recorded the loss values for each epoch, and saw the evolution of the loss function during the training phase. After training, we visualized the loss values across epochs. Then, we used our neural network to predict hardening parameters for each geometry. By inputting random data, we extracted predictions for these parameters. As the final step, we used a bar chart to visualize the hardening parameters "c1" to "c7" across our array of geometries.

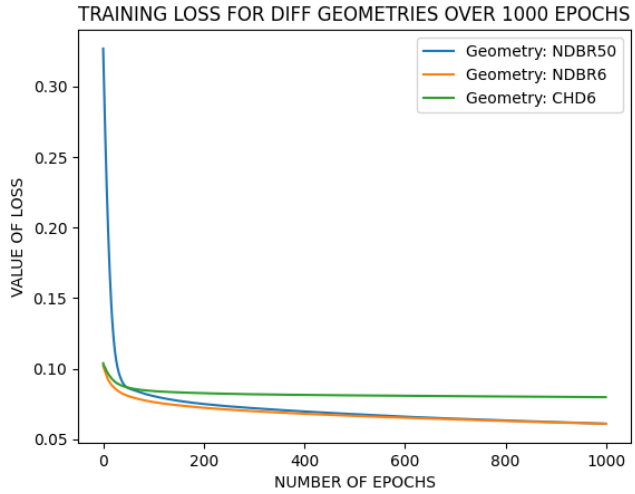

Figure 6: Training Loss for Different Geometries based on Phase 1 Model

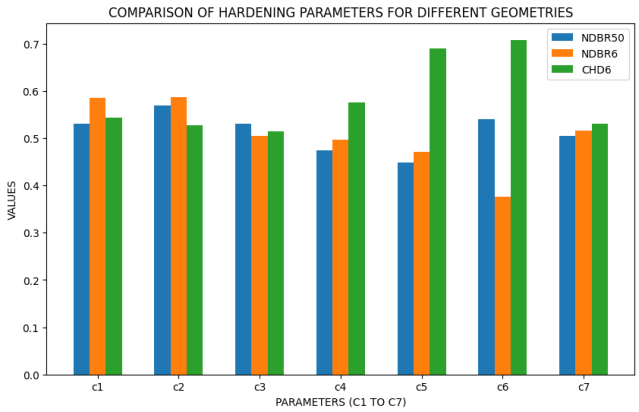

Figure 7: Comparison of Hardening Parameters for Different Geometries based on Phase 1 Model

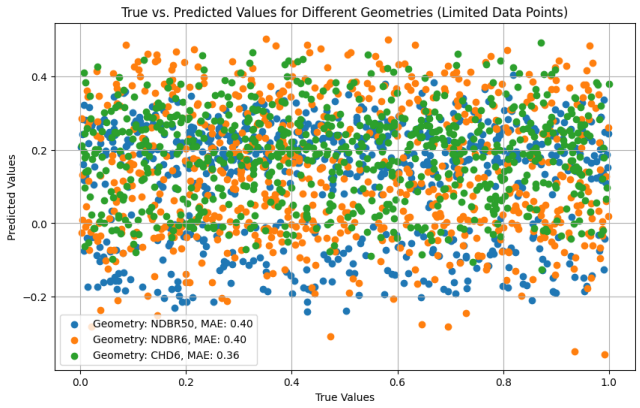

Figure 8: True vs. Predicted Values for Different Geometries based on Phase 1 Model

3.2.3 Phase 2: Optimization of the Dummy Model

We optimized the neural network by iteratively training it using a supervised learning approach. The optimization process involves adjusting the model's parameters to minimize the Mean Squared Error loss. After successful training, the model can then make predictions, which can be compared and visualized to assess its performance. This iterative optimization process is key to the machine learning model's ability to make accurate predictions.

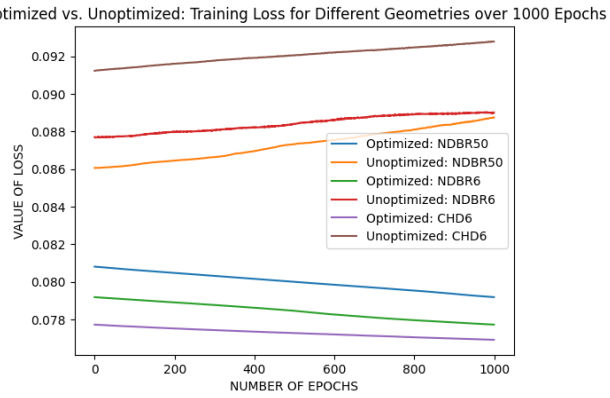

Figure 9: Training Loss Comparison Between Optimized and Unoptimized Models (Phase 2 Model)

3.2.4 Phase 3: Transition to Real Data

In contrast to our previous dummy model, which operated on synthetic data, this model is built to process real data sourced from Zinan for a single uniform geometry. Our primary goal with this model is to predict and understand the hardening parameters, specifically c1 to c7, placing them on the Y-axis of our analysis. Meanwhile, the X-axis is populated with input features that encompass crucial force and displacement values. This arrangement is the cornerstone of our neural network model, enabling us to conduct a comprehensive and accurate analysis of the provided data, making it a valuable tool for data-driven insights and modeling.

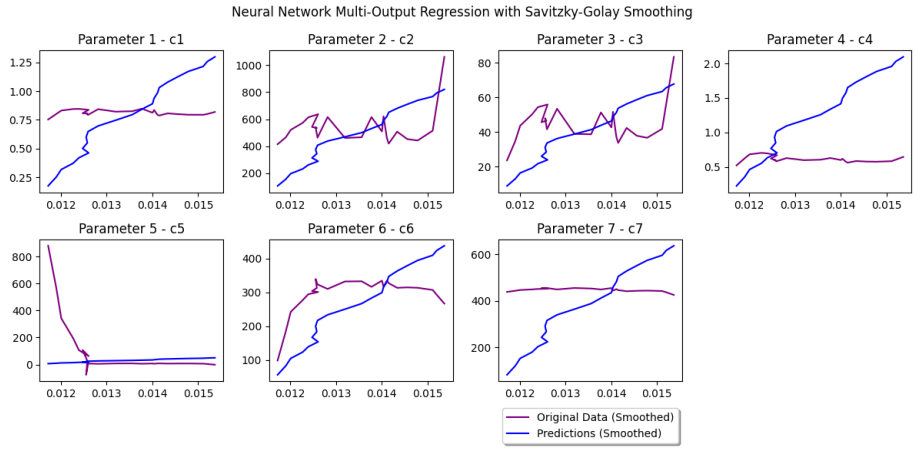

Figure 10: Hardening Parameter Prediction for a Uniform Geometry with Phase 3 Neural Network Model

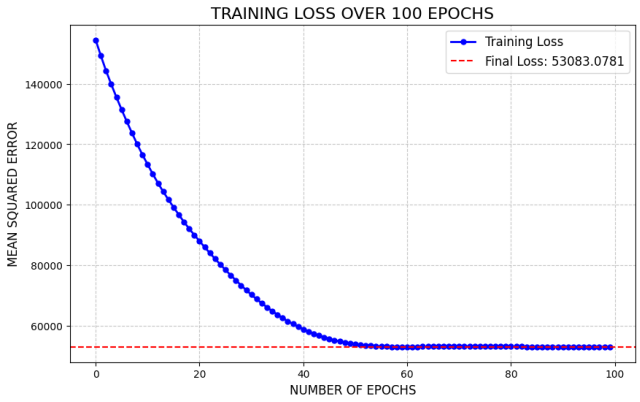

Figure 11: Training Loss of Phase 3 Model

3.2.5 Phase 4: Integration of Real Data and Geometries

While the Phase 3 operated on real data, it had a limitation in that it did not take into account the variations across different geometries. In contrast, Phase 4 model has been designed to address this gap. It not only considers the diverse range of geometries but also takes into consideration their specific yielding indices. This enhancement allows us to tailor our analysis and predictions to the unique characteristics of each geometry, making our model more robust and adaptable to real-world scenarios.

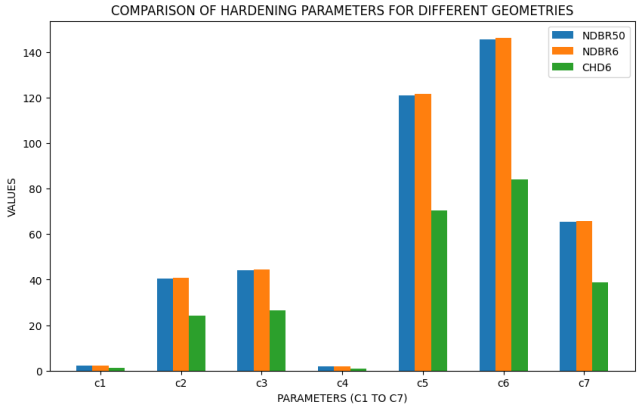

Figure 12: Comparison of Hardening Parameters for Different Geometries based on Phase 4 Model

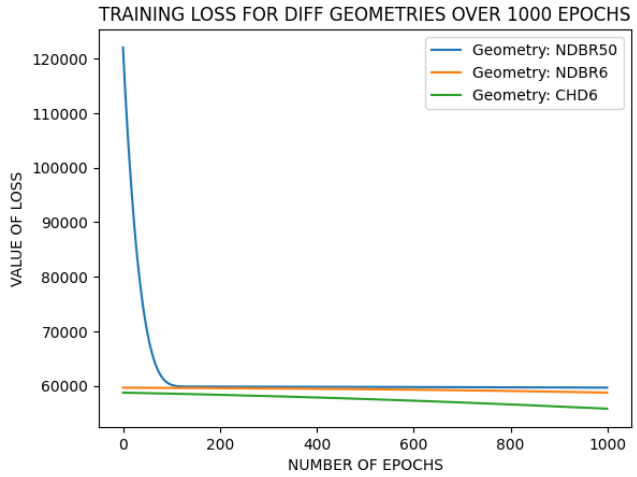

Figure 13: Training Loss of Phase 4 Model

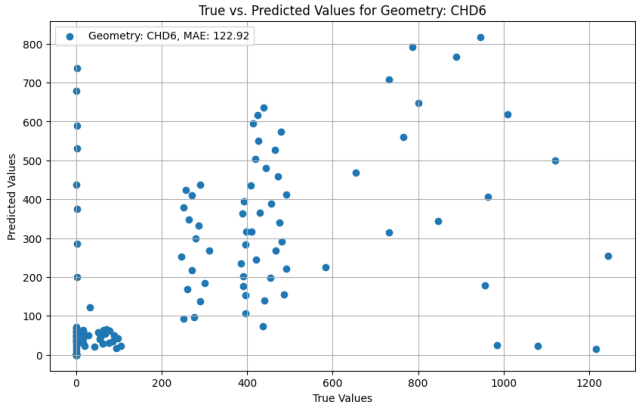

Figure 14: True vs. Predicted Values for CHD6 based on Phase 4 Model

Implementation of Different Learning Rates During phase 4, We experimented with varying learning rates for each geometry, and we visualized how these different learning rates impacted the training loss for these geometries.

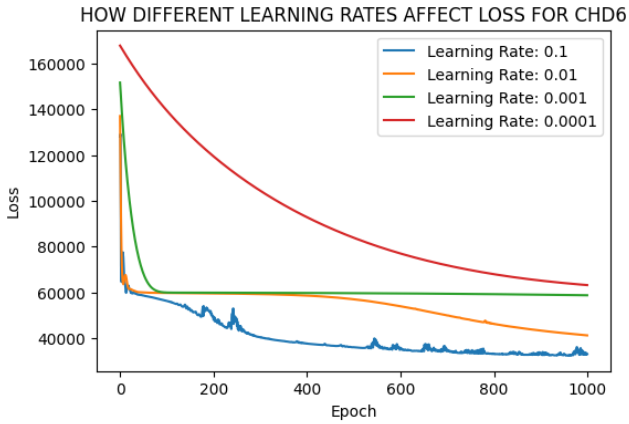

Figure 15: Impact of Different Learning Rates on CHD6 based on Phase 4 Model

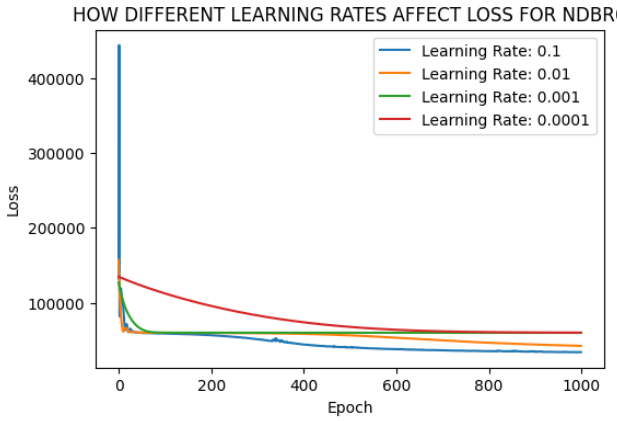

Figure 16: Impact of Different Learning Rates on NDBR6 based on Phase 4 Model

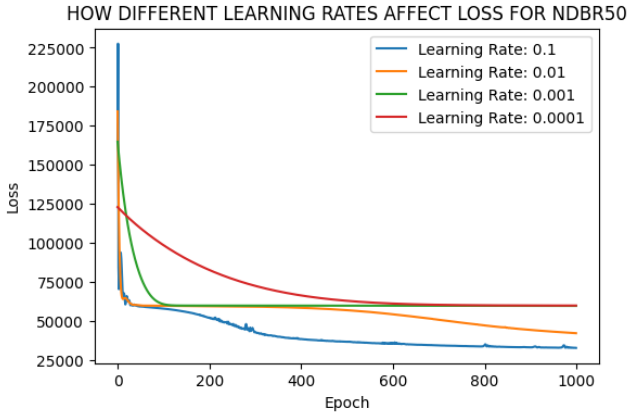

Figure 17: Impact of Different Learning Rates on NDBR50 based on Phase 4 Model

4 Results

5 Discussion

6 Conclusions

7 Reflections

8 References

Bibliography

[1] M. Peksen, "Numerical thermomechanical modelling of solid oxide fuel cells," *Progress in Energy and Combustion Science,* vol. 48, pp. 1-20, 2015.

[2] Y. Li, J. He, B. Gu and S. Li, "Identification of advanced constitutive model parameters through global optimization approach for DP780 steel sheet," in *International Conference on the Technology of Plasticity, ICPT*, Cambridge, 2017.

[3] D. K. Banerjee, K. Prasad, D. Kumar and K. Hariharan, "Uncertainties in Swift hardening law parameters and their influence on the flow stress and the hole expansion behavior of dual phase (DP600) steel specimens," *Journal of Materials Engineering and Performance,* 2023.

[4] F. Reventos, "Chaper 3: Parameters and concepts," in *Thermal-Hydraulics of Water Cooled Nuclear Reactors*, Duxford, Woodhead Publishing, 2017, pp. 89-141.

[5] GIP, "What are Inverse Problems?," GIP, 9 October 2023. [Online]. Available: http://inverseprobleme.de/?page_id=285&lang=en. [Accessed 2023 October 2023].

[6] S. Guo, "Solving Inverse Problems With Physics-Informed DeepONet: A Practical Guide With Code Implementation," Towards Data Science, 17 July 2023. [Online]. Available: https://towardsdatascience.com/solving-inverse-problems-with-physics-informed-deeponet-a-practical-guide-with-code-implementation-27795eb4f502. [Accessed 9 October 2023].

[7] R. Pugliese, S. Regondi and R. Marini, "Machine learning-based approach: global trends, research directions, and regulatory standpoints," *Data Science and Management,* vol. 4, pp. 19-29, 2021.

[8] J. Fumo, "Types of Machine Learning Algorithms You Should Know," Towards Data Science, 15 June 2017. [Online]. Available: https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861. [Accessed 9 October 2023].

[9] W. Liu, J. Lian, S. Münstermann, C. Zeng and X. Fang, "Predicition of crack formation in the progressive folding of square tubes during dynamic axial crushing," *International Journal of Mechanical Sciences,* no. 176, p. 105534, 2020.

[10] Y. Zhang, L. Guo, C. Brousse, C.-H. Lee, A. Azoug, H. Lu and S. Wang, "Machine learning based inverse modelling of full-field strain distribution for mechanical characterization of a linear elastic and heterogeneous membrane," *Mechanics of Materials,* vol. 165, no. 104134, 2022.

[11] R. Lourenço, A. Andrade-Campos and P. Georgieva, "The Use of Machine-Learning Techniques in Material Constitutive Modelling for Metal Forming Processes," *Metals,* vol. 12, no. 3, p. 427, 2022.

[12] D. Yao, S. Pu, M. Li, Y. Guan and Y. Duan, "Parameter identification method of the semi-coupled fracture model for 6061 aluminium alloy sheet based on machine learning assistance," *International Journal of Solids and Structures,* no. 111823, 2022.

[13] Y. Lin and J. Z. J. Zhang, "Application of neural networks to predict the elevated temperature flow behavior of a low alloy steel," *Computational Materials Science,* vol. 43, no. 4, pp. 752-758, 2008.

[14] L.-F. L. B.-C. Guo and Z.-M. Zhang, "Constitutive relationship model of TC21 alloy based on artificial neural network," *Transactions of Nonferrous Metals Society of China,* vol. 23, pp. 1761-1765, 2013.

[15] J. Wang, B. Zhu, C.-Y. Hui and A. T. Zehnder, "Determination of material parameters in constitutive models using adaptive neural network machine learning," *Journal of the Mechanics and Physics of Solids,* vol. 177, p. 105324, 2023.

[16] D. Chen, Y. Y. X. Li, J. Wei and L. Guan, "Efficient parameters identification of a modified GTN model of ductile fracture using machine learning," *Engineering Fracture Mechanics,* vol. 245, p. 107535, 2021.

[17] A. Hajari, M. Morakabati, S. Mahdi Abbasi and H. Badri, "Constitutive modeling for high-temperature flow behavior of Ti-6242S alloy," *Materials Science & Engineering A,* no. 681, pp. 103-113, 2017.

[18] X. Liu, S. Tian, F. Tao and W. Yu, "A review of artificial neural networks in the constitutive modeling of composite materials," *Composites Part B: Engineering,* vol. 224, p. 109152, 2021.

[19] N. Bastos, P. Prates and A. Andrade-Campos, "Material Parameter Identification of Elastoplastic Constitutive Models Using Machine Learning Approaches," *Key Engineering Materials,* vol. 926, pp. 2193-2200, 2022.