Cycle-domain plasticity modeling using neural networks and symbolic regression

Computers and Structures 321 (2026) 108086

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/cas

Cycle-domain plasticity modeling using neural networks and symbolic regression

Nasrin Talebi*, Knut Andreas Meyer, Magnus Ekh

Division of Material and Computational Mechanics, Department of Industrial and Materials Science, Chalmers University of Technology, SE-412 96, Gothenburg, Sweden

ARTICLE INFO

Keywords:
Cycle-domain plasticity modeling
Machine learning
Data-driven modeling
Neural networks
Symbolic regression

ABSTRACT

Simulation of many loading cycles with traditional time-domain material models, requiring discretization of each cycle with several time steps, can result in high computational cost. One effective approach to speed up cyclic simulations is employing cycle-domain material models. Finite element simulations of rails subjected to many wheel passages are a relevant application of such models. Proposing a per-cycle evolution equation for plastic strains in cycle-domain models is, however, a challenge. To address this, we investigate the feasibility and accuracy of using machine learning models as tools for formulating such an equation. Specifically, we enforce our knowledge from constitutive modeling for elasticity and formulate the evolution law by employing feed-forward neural networks with different inputs, as well as symbolic regression to discover an interpretable expression. Training, validation, and test data have been generated using a cyclic time-domain plasticity model considering pulsating uniaxial stress loadings with constant and variable strain ranges. The obtained results demonstrate the potential of cycle-domain plasticity modeling using both uninterpretable and interpretable data-driven machine learning as an alternative to time-domain material modeling. Furthermore, both approaches have revealed reasonably good extrapolation performance beyond the training regime.

1. Introduction

For structures subjected to cyclic loading where no stabilized behavior is obtained, it is time-consuming to apply a non-linear constitutive model and perform time steps in Finite Element (FE) simulations [1]. One approach to speed up the simulation of many loading cycles is to extrapolate the response by using load sequence extrapolation with error control, as developed by Johansson and Ekh [2]. In this approach, state variables and global responses obtained from FE simulations at cycle N are extrapolated using a Taylor series expansion. Alternatively, Brommesson et al. [3] proposed scaling the number of cycles in experiments and recalibrating the material model parameters accordingly. Although this reduces computational time linearly, calibrating multiple sets of material parameters remains time-consuming.

An alternative approach is to use cycle-domain modeling as proposed by Suiker and de Borst [4]. The model framework is based on standard plasticity theory and is formulated as a viscoplastic model to simulate the evolution of plastic deformations in ballasted tracks subjected to many loading cycles. Based on that framework, Li et al. [5] developed a three-dimensional FE simulation tool to predict long-term differential settlement in ballasted tracks to enable fast simulations of many wheel passages. Cycle-domain material models have the advantage that the FE algorithm is not modified, as only the applied material model is changed, making the method suitable for industrial implementations. In addition to geomechanical applications, for modeling the stiffness degradation of short-fiber reinforced plastics, Köbler et al. [6] formulated their fatigue damage model in the cycle domain. In the same context, Magino et al. [7] proposed a hybrid log-cycle fatigue damage model that combines time-domain accuracy in initial cycles with the computational efficiency of cycle-domain models in high-cycle fatigue regimes. The key challenge with formulating cycle-domain models is to find suitable per-cycle evolution laws for the internal variables. In this paper, we propose to do this by using Machine Learning (ML) techniques.

The use of ML to enhance time-domain material modeling has been a rapidly growing area of research in recent years. As reviewed by Fung et al. [8], ML-based constitutive models can be categorized as either uninterpretable or interpretable. Focusing on the former, Ali et al. [9] trained a Feed-Forward Neural Network (FFNN) to predict the stress-strain response and texture evolution of single crystal and polycrystalline materials under uniaxial tension and simple shear.

* Corresponding author.
Email address: nasrin.talebi@chalmers.se (N. Talebi).

https://doi.org/10.1016/j.compstruc.2025.108086
Received 12 August 2025; Accepted 20 December 2025

For modeling anisotropic plane-stress plasticity under multiaxial loading, Gorji et al. [10] demonstrated the potential of Recurrent Neural Networks (RNNs) with the use of a large amount of training data. Linka et al. [11] introduced a new approach by incorporating constitutive modeling knowledge directly into the architecture of FFNNs, allowing for efficient training with a moderate amount of data and resulting in improved extrapolation performance.

The inclusion of physical principles into neural networks has been shown to noticeably improve the ability to generalize outside the training regime in material modeling [12]. Raissi et al. [13] introduced Physics-Informed Neural Networks (PINNs) to solve forward and inverse problems, while imposing physical constraints on the loss function. Following this, Abueidda et al. [14] employed PINNs to solve partial differential equations for displacement fields, and Haghighat et al. [15] developed a PINN-based framework for constitutive modeling by embedding elastoplastic inequality constraints into the loss function. Some ML-based material modeling techniques aim to satisfy physical constraints by construction. For instance, Meyer and Ekre [16] proposed embedding FFNNs directly into the evolution laws of internal variables. The framework inherently fulfills thermodynamics and material objectivity. Fuhg et al. [17] suggested a thermodynamically consistent framework, which is instead based on isotropic and nonlinear kinematic hardening potentials. Following this approach, Jadoon et al. [18] developed an NN-based framework for finite strain elastoplastic material modeling.

Despite the satisfactory results of the NN-based constitutive models, their black-box nature remains a limitation [8]. In the context of interpretable ML, Flaschel et al. [19] introduced a method based on sparse regression to discover constitutive laws for isotropic hyperelastic materials using displacements and global force data, and later extended the work to automatically discover constitutive laws for isotropic materials with unknown categories of material behavior [20]. Combining sparse regression and binomial searching, Meyer and Ekre [16] discovered analytical expressions from NN-generated data for evolution equations of internal variables. Alternatively, Symbolic Regression (SR), referring to methods based on genetic algorithms [8], was applied by Versino et al. [21] to discover flow stress equations for copper. Working with a limited amount of experimental data, they addressed this constraint by incorporating their expert knowledge to enhance the predictive accuracy of the models. Based on micromechanical finite element simulations, Bomarito et al. [22] demonstrated the potential of SR in discovering the von Mises yield function and the plastic strain evolution equation. Using the same methodology, Abdusalamov et al. [23] proposed an SR-based approach to identify the strain energy density function for hyperelastic materials. Bahmani et al. [24] suggested a hybrid framework to formulate the yield functions of perfectly plastic materials by expressing them as polynomials composed of symbolic equations that replace trained univariate neural networks.

In contrast to available studies on ML-based time-domain constitutive modeling (e.g., [11,15–17]), this study proposes ML-based cycle-domain plasticity modeling. Incorporating established knowledge from material modeling for elasticity, we obtain the per-cycle evolution law for plastic strains using either FFNNs (uninterpretable ML models) or SR (an interpretable alternative). To focus on the differences between these approaches, we study one-dimensional problems. It should be noted that, although cycle-domain models can be employed in different applications involving cyclic loadings, some assumptions in this work are motivated from a railway mechanics point of view. The paper is organized as follows: In Section 2, we explain the standard formulation of a time-domain plasticity model, while the cycle-domain model formulation for uniaxial stress loading is presented in Section 3. In Section 4, the time-domain plasticity model adopted to generate artificial training, validation, and test data, together with the corresponding data generation procedure, is described. In Section 5, we explain the considered FFNNs and SR models, and finally, in Sections 6 and 7, we discuss and summarize our findings.

2. Standard formulation of time-domain plasticity models

In this section, we present the standard formulation of a time-domain plasticity model in a small strain setting. Note that, second-order tensors are written in boldface, e.g. $\boldsymbol{t}$, while fourth-order tensors are written in capitalized, boldface, and upright form, e.g. $\mathbf{T}$.

The total strain, $\boldsymbol{\epsilon}$, is additively decomposed into an elastic strain, $\boldsymbol{\epsilon}^{\mathrm{e}}$, and a plastic strain, $\boldsymbol{\epsilon}^{\mathrm{p}}$,

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon}^{\mathrm{e}} + \boldsymbol{\epsilon}^{\mathrm{p}} \;\rightarrow\; \boldsymbol{\epsilon}^{\mathrm{e}}(\boldsymbol{\epsilon}, \boldsymbol{\epsilon}^{\mathrm{p}}) = \boldsymbol{\epsilon} - \boldsymbol{\epsilon}^{\mathrm{p}} \tag{1}$$

The stress, $\boldsymbol{\sigma}$, can then be calculated using Hooke's law

$$\boldsymbol{\sigma} = \mathbf{E}^{\mathrm{e}} : \left(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^{\mathrm{p}}\right) \tag{2}$$

where $\mathbf{E}^{\mathrm{e}}$ is the elasticity tensor. The yield function $\Phi(\boldsymbol{\sigma}, \mathbb{A})$ defines the elastic domain ($\Phi < 0$) and the plastic domain ($\Phi = 0$), where $\mathbb{A} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$ is a set of hardening stresses. The evolution of plastic strains, $\boldsymbol{\epsilon}^{\mathrm{p}}$, is typically assumed to follow an associative flow rule

$$\dot{\boldsymbol{\epsilon}}^{\mathrm{p}} = \dot{\lambda}\,\frac{\partial \Phi}{\partial \boldsymbol{\sigma}} \tag{3}$$

whereas for the hardening stresses, $\boldsymbol{a}_i$, non-associative evolution laws provide better agreements with experimentally observed material behavior [25]

$$\dot{\boldsymbol{a}}_i = \dot{\lambda}\, g_i(\boldsymbol{\sigma}, \mathbb{A}) \tag{4}$$

$\dot{\lambda}$ is the plastic multiplier (rate of accumulated equivalent plastic strain) and can be determined from the Karush-Kuhn-Tucker (KKT) loading/unloading conditions

$$\Phi \le 0, \quad \dot{\lambda} \ge 0, \quad \Phi\,\dot{\lambda} = 0 \tag{5}$$

To determine whether the material response is elastic or plastic, the Karush-Kuhn-Tucker conditions given in Eq. (5) are used. Considering the implicit backward Euler time integration scheme, we assume that, at the time step ${}^{n}t$, the material response is elastic, i.e., ${}^{n}\lambda = 0$, thereby $\Delta\lambda = {}^{n}\lambda - {}^{n-1}\lambda = 0$. Accordingly, ${}^{n}\boldsymbol{\sigma}$ is equal to the calculated trial stress ${}^{n}\boldsymbol{\sigma}^{\mathrm{tr}}$ as: ${}^{n}\boldsymbol{\sigma} = {}^{n}\boldsymbol{\sigma}^{\mathrm{tr}} = \mathbf{E}^{\mathrm{e}} : \left({}^{n}\boldsymbol{\epsilon} - {}^{n-1}\boldsymbol{\epsilon}^{\mathrm{p}}\right)$, and the hardening stresses do not change: ${}^{n}\boldsymbol{a}_i = {}^{n-1}\boldsymbol{a}_i$. This assumption holds if $\Phi \le 0$; otherwise, a system of nonlinear equations must be solved. For further details, the reader is referred, e.g., to [26].

The formulation of the cycle-domain material model introduced in Section 3 is independent of the choice of a time-domain material model. In this study, to train, validate, and test the ML-based cycle-domain model, we have chosen a reference time-domain plasticity model with a von Mises yield function, an associative evolution law for the plastic strain, and a nonlinear evolution law for the hardening stress; see Section 4.1 for further details.

3. Cycle-domain material model

To enhance the computational efficiency of cyclic simulations involving many cycles, we propose replacing a standard time-domain model with a cycle-domain model for pulsating uniaxial loading, inspired by Suiker and de Borst [4]. In the cycle-domain model, we predict the peak stress and the corresponding plastic strain in each loading cycle N, as shown in Fig. 1. This choice is motivated by the application of railway mechanics, where rails are subjected to purely pulsating loading [5,27]. The notation ${}^{N}(\cdot)$ refers to cycle N, and ${}^{N-1}(\cdot)$ refers to cycle $N-1$. Assuming small strains and Hooke's law (Eq. (2)) for uniaxial loading, we replace the time derivatives of quantities with their change per unit cycle, N, as

$$\frac{\mathrm{d}\sigma_{11}}{\mathrm{d}N} = E\,\frac{\mathrm{d}\epsilon^{\mathrm{e}}_{11}}{\mathrm{d}N} = E\left(\frac{\mathrm{d}\epsilon_{11}}{\mathrm{d}N} - \frac{\mathrm{d}\epsilon^{\mathrm{p}}_{11}}{\mathrm{d}N}\right) \tag{6}$$

where E is Young's modulus. The challenge with cycle-domain models is to propose a suitable per-cycle evolution equation for plastic strains.

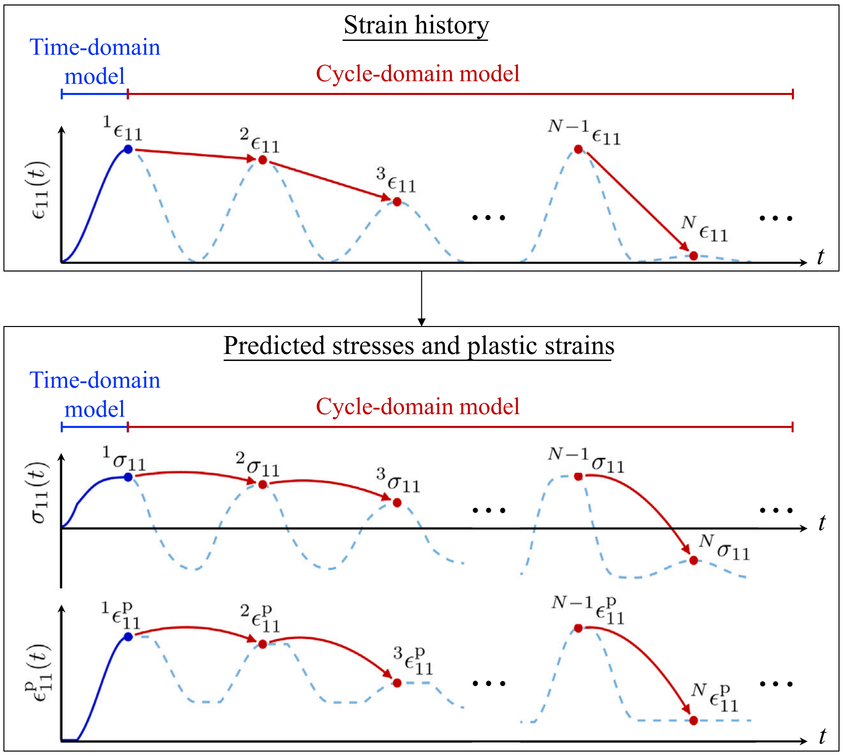


Fig. 1. Schematic illustration of the cycle-domain material model in cyclic simulations. The solid blue lines represent the input strain history to the time-domain model and its simulated first half-cycle. The red circles show the peak strain ${}^{N}\epsilon_{11}$, as well as the peak normal stress ${}^{N}\sigma_{11}$ and the corresponding plastic strain ${}^{N}\epsilon^{\mathrm{p}}_{11}$ predicted by the cycle-domain model for the loading cycle N. The dashed light blue lines denote the strain histories and responses from the time-domain model.

We aim to discover this equation with machine learning. Specifically, we will train a model, f_{ml}, with inputs $\boldsymbol{x}$

$$\frac{\mathrm{d}\,{}^{N}\epsilon^{\mathrm{p}}_{11}}{\mathrm{d}N} = \mathcal{H}_{\Phi}\left({}^{N-1}\epsilon^{\mathrm{p}}_{11}, {}^{N}\sigma^{\mathrm{tr}}_{11}\right) f_{\mathrm{ml}}(\boldsymbol{x}) \qquad \text{where} \tag{7}$$

$$\mathcal{H}_{\Phi}\left({}^{N-1}\epsilon^{\mathrm{p}}_{11}, {}^{N}\sigma^{\mathrm{tr}}_{11}\right) = \begin{cases} 0, & \text{if} \quad {}^{N-1}\epsilon^{\mathrm{p}}_{11} = 0 \quad \text{and} \quad |{}^{N}\sigma^{\mathrm{tr}}_{11}| < Y_0 \\ 1, & \text{otherwise} \end{cases} \qquad \text{and}$$

$${}^{N}\sigma^{\mathrm{tr}}_{11} = E\left({}^{N}\epsilon_{11} - {}^{N-1}\epsilon^{\mathrm{p}}_{11}\right)$$

$\mathcal{H}_{\Phi}(\cdot)$ is introduced to ensure that no plastic strain increment occurs when the material response is elastic (i.e., $|{}^{N}\sigma^{\mathrm{tr}}_{11}|$ is smaller than the initial yield stress Y_0) with no previous plasticity (${}^{N-1}\epsilon^{\mathrm{p}}_{11} = 0$). Accordingly, the cycle-domain model will behave equivalently to the time-domain model for elastic-only loading (corresponding to high cycle fatigue loading). This has been found to facilitate the training of f_{ml} later in the ML procedure described in Section 5. Note that we use a semi-implicit discretization scheme between the loading cycles. When using the model, the first half-cycle is simulated with the time-domain model, before applying the cycle-domain model to predict the subsequent cycles, as illustrated in Fig. 1.

4. Data generation

This section presents the reference time-domain plasticity model adopted to generate artificial data and the corresponding data generation procedure. The data will be used for training, validation, and testing of the cycle-domain model with different ML-based evolution equations for plastic strains.

4.1. Reference time-domain cyclic plasticity model

Based on the general small strain plasticity model formulation described in Section 2, a specific reference model is described in the following. The elastic behavior is linear isotropic, i.e.,

$$\mathbf{E}^{\mathrm{e}} = 2\,G\,\mathbf{I}^{\mathrm{dev}} + K_{\mathrm{b}}\,\boldsymbol{I}\otimes\boldsymbol{I} \rightarrow \boldsymbol{\sigma} = 2\,G\,\boldsymbol{\epsilon}^{\mathrm{e}}_{\mathrm{dev}} + K_{\mathrm{b}}\,\mathrm{tr}(\boldsymbol{\epsilon}^{\mathrm{e}})\,\boldsymbol{I}, \quad K_{\mathrm{b}} = \frac{E\,G}{3\,(3\,G - E)} \tag{8}$$

where G, K_{b}, and E are the shear, bulk, and Young's moduli, respectively. $\boldsymbol{I}$ is the second-order identity tensor, and $\boldsymbol{\epsilon}^{\mathrm{e}}_{\mathrm{dev}}$ is the deviatoric elastic strain. The fourth-order deviatoric identity tensor is defined as $\mathbf{I}^{\mathrm{dev}} = \boldsymbol{I}\overline{\otimes}\boldsymbol{I} - \boldsymbol{I}\otimes\boldsymbol{I}/3$, where the non-standard open product $\overline{\otimes}$ between two second-order tensors is given by $\boldsymbol{a}\overline{\otimes}\boldsymbol{b} = a_{ik}\,b_{jl}\,\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l$. The model considers kinematic hardening and has the hardening stress $\mathbb{A} = \{\boldsymbol{\beta}\}$, where $\boldsymbol{\beta}$ is the back-stress. Using the von Mises effective stress, the yield function is then defined as

$$\Phi = \sqrt{\frac{3}{2}\,\boldsymbol{\sigma}^{\mathrm{red}}_{\mathrm{dev}} : \boldsymbol{\sigma}^{\mathrm{red}}_{\mathrm{dev}}} - Y_0 \le 0 \quad \text{with} \quad \boldsymbol{\sigma}^{\mathrm{red}}_{\mathrm{dev}} = \boldsymbol{\sigma}_{\mathrm{dev}} - \boldsymbol{\beta} \tag{9}$$

where $\boldsymbol{\sigma}_{\mathrm{dev}}$ is the deviatoric stress. The evolution of the plastic strain, $\boldsymbol{\epsilon}^{\mathrm{p}}$, follows the associative flow rule

$$\dot{\boldsymbol{\epsilon}}^{\mathrm{p}} = \dot{\lambda}\,\frac{\partial\Phi}{\partial\boldsymbol{\sigma}} = \dot{\lambda}\,\boldsymbol{\nu} \quad \text{with} \quad \boldsymbol{\nu} = \sqrt{\frac{3}{2}}\,\frac{\boldsymbol{\sigma}^{\mathrm{red}}_{\mathrm{dev}}}{|\boldsymbol{\sigma}^{\mathrm{red}}_{\mathrm{dev}}|} \tag{10}$$

where the plastic multiplier, $\dot{\lambda}$, can be determined from the KKT loading/unloading conditions, as described in Section 2. The evolution of the back-stress, $\boldsymbol{\beta}$, is assumed to be of the Armstrong-Frederick type [28]

$$\dot{\boldsymbol{\beta}} = -\frac{2}{3}\,H_{\mathrm{kin}}\,\dot{\lambda}\left[-\boldsymbol{\nu} + \frac{3}{2}\,\frac{\boldsymbol{\beta}}{\beta_{\infty}}\right] \tag{11}$$

Table 1
Material model parameter values for the reference time-domain cyclic plasticity model.

Material parameters	E	G	Y_0	H_{kin}	β_∞
Unit	200 GPa	75 GPa	400 MPa	150 GPa	500 MPa

H_{kin} is the kinematic hardening modulus, and β_∞ is the saturation value of the back-stress. The values of the material model parameters, inspired by those calibrated based on experimental data for isotropic R260 pearlitic rail steel [29], are listed in Table 1.

4.2. Training, validation, and test data generation

In this study, data required for training, validation, and testing of the considered ML models are generated artificially using the time-domain cyclic plasticity model described in Section 4.1. The data are employed to train an NN and to discover a symbolic equation representing the per-cycle evolution equation for plastic strains in the cycle-domain material model. They are also used for model validation and testing. Data generation is performed through material point simulations. Each dataset consists of 100 loading cycles, with each loading cycle discretized into 400 time steps. Cyclic pulsating strain-controlled uniaxial stress loadings with both constant and variable strain ranges are applied. For constant strain range loadings, 400 logarithmically spaced strain range values, $\Delta\epsilon_{11}$, spanning from 0.2 % (corresponding to the elastic limit) to 3.0 % are considered. The lower limit of the strain range is chosen to avoid elastic material responses without previous plasticity. Logarithmic spacing is chosen to generate a higher density of datasets in the transition region from elastic shakedown to plastic shakedown behavior, where small changes in strain ranges significantly affect the material response. Variable strain range values are randomly generated between 0.0 % and 3.0 % 400 times, following a uniform distribution. These procedures produce a total of 800 artificial datasets, covering a wide range of material behavior.

Some results from the time-domain material model under loadings with constant $\Delta\epsilon_{11}$ are presented in Fig. 2(a)–(c). For the lower $\Delta\epsilon_{11}$ of 0.39 % (Fig. 2(a)), the material response shows elastic shakedown behavior following the first loading cycle. However, a higher strain range value of 0.77 % (Fig. 2(b)) causes a reduction in the peak stresses before stabilizing into a plastic shakedown state. Under more severe loading conditions, with $\Delta\epsilon_{11}$ of 3.0 %, the material reaches a plastic shakedown state from the second loading cycle, see Fig. 2(c). In contrast to the loadings with constant strain ranges, from Fig. 2(d), no stabilized behavior after 100 cycles can be observed.

Training, validation, and test data are prepared in the form of $\{{}^N\sigma_{11}, {}^{N-1}\epsilon^{\text{p}}_{11}, \text{d}^N\epsilon^{\text{p}}_{11}/\text{d}N\}$ for the two-input NN and SR models. For the three-input models, the data are structured as $\{{}^N\sigma_{11}, {}^{N-1}\epsilon^{\text{p}}_{11}, {}^{N-1}\sigma_{11}, \text{d}^N\epsilon^{\text{p}}_{11}/\text{d}N\}$; see Section 5 for more details regarding the models and their inputs. For completeness, the relevant variables are reintroduced in this section, even though some of them are defined in Section 3. ${}^N\sigma_{11}$ denotes the peak stress in loading cycle N, ${}^{N-1}\sigma_{11}$, as well as ${}^{N-1}\epsilon^{\text{p}}_{11}$, are the peak stress and plastic strain values in loading cycle $N-1$, and $\text{d}^N\epsilon^{\text{p}}_{11}/\text{d}N$ is the plastic strain increment from cycle $N-1$ to N. As an example, Fig. 3 illustrates ${}^N\sigma_{11}$, ${}^{N-1}\epsilon^{\text{p}}_{11}$, and $\text{d}^N\epsilon^{\text{p}}_{11}/\text{d}N$ corresponding to a dataset obtained from loading with variable strain ranges. One observation is the occurrence of negative $\text{d}^N\epsilon^{\text{p}}_{11}/\text{d}N$, resulting from a cycle with a high $\Delta\epsilon_{11}$ followed by a cycle with a much lower $\Delta\epsilon_{11}$.

5. Machine Learning (ML) models

In this work, we investigate two approaches to formulate the per-cycle evolution equation for plastic strains, $\text{d}\epsilon^{\text{p}}_{11}/\text{d}N$, in the cycle-domain material model: training an NN and SR. The artificially generated datasets are split into 80 % for training, 10 % for validation, and 10 % for testing. Considering the datasets with variable strain range loading, a few data points corresponding to elastic material behavior with no previous plasticity are filtered out according to the $\mathcal{H}_\Phi$ function (see Eq. (7)) in the cycle-domain model formulation.

In both alternative approaches, the loss function, minimizing the difference between the predictions of the models and the target values, is chosen to be of the mean-squared error type and is formulated as

$$\hat{L}(\boldsymbol{x}) = \frac{1}{k}\sum_{i=1}^{k}\left(\left(\frac{\text{d}^N\hat{\epsilon}^{\text{p}}_{11}}{\text{d}N}\right)_i - f_{\text{ml}}(\boldsymbol{x}_i)\right)^2 \quad \text{where } f_{\text{ml}}(\boldsymbol{x}) = f_{\text{nn}}(\boldsymbol{x}) \text{ or } f_{\text{sr}}(\boldsymbol{x}) \tag{12}$$

where k denotes the total number of data points included in the training, validation, or test sets, and $\frac{\text{d}^N\hat{\epsilon}^{\text{p}}_{11}}{\text{d}N}$ is the min-max normalized $\frac{\text{d}^N\epsilon^{\text{p}}_{11}}{\text{d}N}$. Since there is a significant difference in the orders of magnitude of the input variables, all the input and output variables of the studied models are scaled to the interval $[0,1]$ (min-max normalized). $f_{\text{nn}}(\boldsymbol{x})$ and $f_{\text{sr}}(\boldsymbol{x})$ represent an NN and SR model, respectively, with input variables $\boldsymbol{x}$. We first evaluate the performance of the models with two input variables: ${}^N\sigma_{11}$ and ${}^{N-1}\epsilon^{\text{p}}_{11}$. The corresponding training data is illustrated in Fig. 4(a), showing a surface-like distribution of data with slight dispersion. The data spans a range of $\frac{\text{d}^N\epsilon^{\text{p}}_{11}}{\text{d}N}$, transitioning from large positive values, through zero, to negative values. Further, the distribution is inhomogeneous, with a noticeably higher sparsity in the region approximately defined by $0.0\ \% \leq {}^{N-1}\epsilon^{\text{p}}_{11} \leq 0.3\ \%$. In this region, some data points also deviate from the general surface trend, as shown by the red circle. Further analysis of the training data distribution for different intervals of ${}^{N-1}\epsilon^{\text{p}}_{11}$ shows that nearly the same inputs correspond to multiple outputs. This behavior is exemplified in Fig. 4(b) for $0.500\ \% \leq {}^{N-1}\epsilon^{\text{p}}_{11} \leq 0.541\ \%$. Due to the non-unique input-output mapping, the NN or SR model can only learn an approximate solution based on the provided input information. Thus, we introduce ${}^{N-1}\sigma_{11}$ as an additional input, forming three-input models, to assess the extent of performance improvement over the two-input models.

5.1. Feed-Forward Neural Network (FFNN)

One investigated approach to obtain the per-cycle evolution equation for plastic strains is using a fully connected FFNN. Although RNNs have been used to model history-dependent behavior, see e.g. [10,30], we have chosen to base our ML-based cycle-domain model on a time-domain plasticity model and have employed an FFNN, as one component of the model, to formulate the per-cycle evolution law. It should be noted that the material model formulation can be considered a recurrent architecture, as each step depends on the previous state.

In a fully connected FFNN, each neuron in a given layer is connected to all neurons in the subsequent layer [31]. The network establishes a mapping from the input vector $\boldsymbol{x}_0$ to the output vector $\boldsymbol{y}$, denoted as $\boldsymbol{y} = f_{\text{nn}}(\boldsymbol{x}_0)$. The information propagates sequentially through m hidden layers before reaching the output layer. Each hidden layer l computes its output using a nonlinear function, referred to as the activation function $a_l(\cdot)$

$$\boldsymbol{x}_l = a_l\left(\underline{\boldsymbol{W}}_l\,\boldsymbol{x}_{l-1} + \boldsymbol{b}_l\right) \quad \text{where} \quad a_l(x) = \begin{cases} \dfrac{1}{1+e^{-x}} & l = 1,\ldots,m-1 \\ \max(0,x) & l = m \end{cases} \tag{13}$$

$\underline{\boldsymbol{W}}_l$ and $\boldsymbol{b}_l$ are trainable weight matrices and bias vectors associated with layer l, respectively. The output vector $\boldsymbol{y}$ is computed as

$$\boldsymbol{y} = a_{m+1}\left(\underline{\boldsymbol{W}}_{m+1}\,\boldsymbol{x}_m + \boldsymbol{b}_{m+1}\right) \quad \text{where} \quad a_{m+1}(x) = \tanh(x) \tag{14}$$

$\tanh(x)$ is chosen for the output layer to allow for both positive and negative plastic strain increments, as stated in Section 4.2.

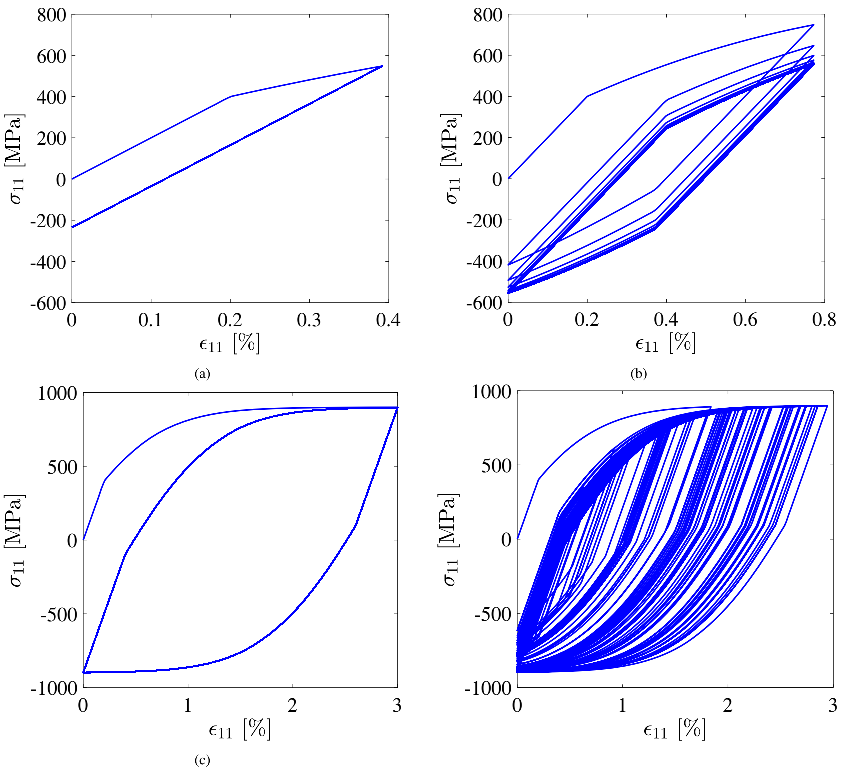


Fig. 2. The stress-strain responses from the time-domain material model for loadings under constant strain range of (a) 0.39 % (elastic shakedown), (b) 0.77 % (mean stress relaxation), (c) 3.0 % (plastic shakedown) and under variable strain ranges over 100 cycles with maximum value of (d) 2.94 %.

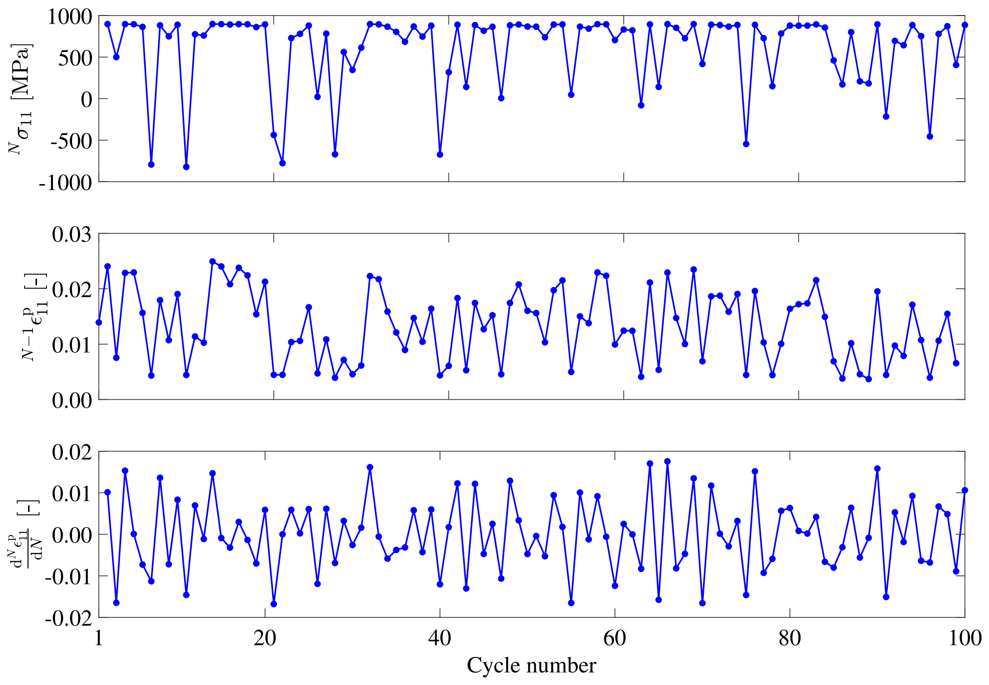
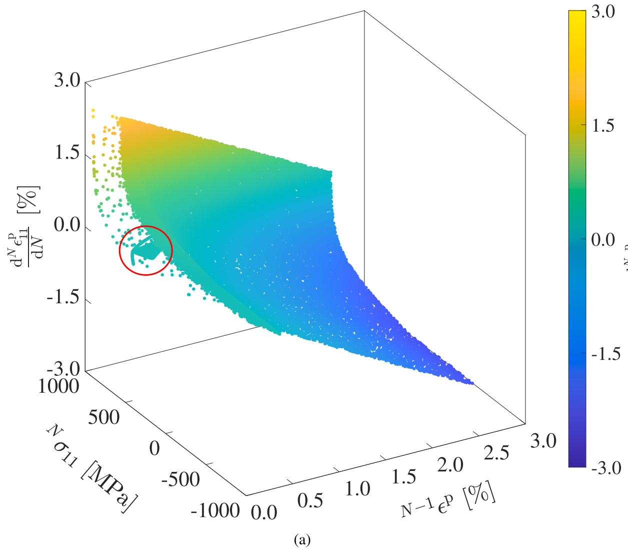


Fig. 3. Training data from a dataset with variable strain ranges over 100 loading cycles.

In this paper, an NN comprising $m = 5$ hidden layers with 6 neurons per layer is considered. Expanding the network architecture ($m = 10$ with 50 neurons per layer) did not significantly improve the training loss compared to the smaller network (from 1.81×10^{-5} to 1.22×10^{-5}). Therefore, we chose the former network architecture for its comparable accuracy and higher computational efficiency. Starting from five distinct random seeds, the NN is trained over four runs, with each consisting of 50,000 epochs. For a given seed, the first training run begins with randomly initialized network parameters, i.e., weights and biases. The identified parameters with the lowest training loss are then selected as



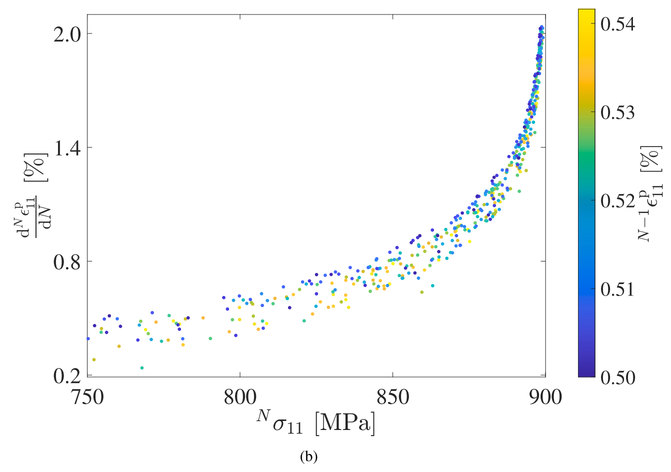


Fig. 4. (a) Training data for the two-input models and (b) training data for an interval of $0.500\,\% \leq {}^{N-1}\epsilon^{\mathrm{p}}_{11} \leq 0.541\,\%$.

initial guesses for the subsequent training run. The goal of this approach is to improve the chance that the optimization algorithm finds the global optimum. The neural network model corresponding to the seed that gives the lowest training loss is selected as the best model. In each training run, the network parameters are identified using the gradient-based Rectified Adam optimization algorithm [32] (RAdam) in the open-source PyTorch library [33] with a learning rate of 0.002, and the gradients are computed using the backpropagation algorithm. Although these hyperparameters, obtained using trial-and-error, gave satisfactory training performance, more efficient training may be achievable via automatic hyperparameter tuning; see [34].

5.2. Symbolic Regression (SR)

To circumvent the black-box nature of NN-based constitutive models, we investigate whether the neural network can be replaced by a human-interpretable mathematical expression that best fits the given data using SR. This method does not require a prior assumption of functional forms, in contrast to conventional regression methods adopted in constitutive modeling. Instead, only building blocks such as basic mathematical operators, functions, and numerical constants need to be defined [35]. In addition, constraints to regulate the complexity of the resulting expressions can be specified. In this contribution, we employ the open-source PySR library [36], which implements a multi-population evolutionary algorithm based on the stochastic optimization method of genetic programming.

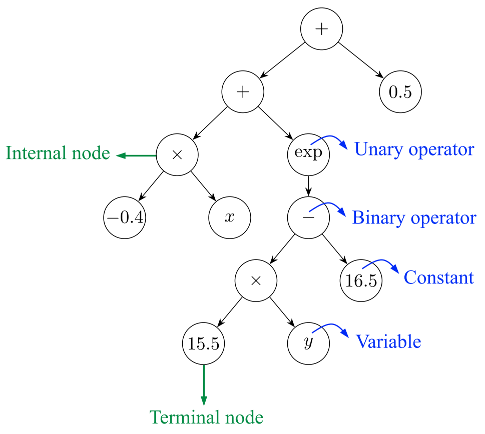


Fig. 5. Expression tree of the symbolic equation $(-0.4\,x + \exp(15.5\,y - 16.5)) + 0.5$.

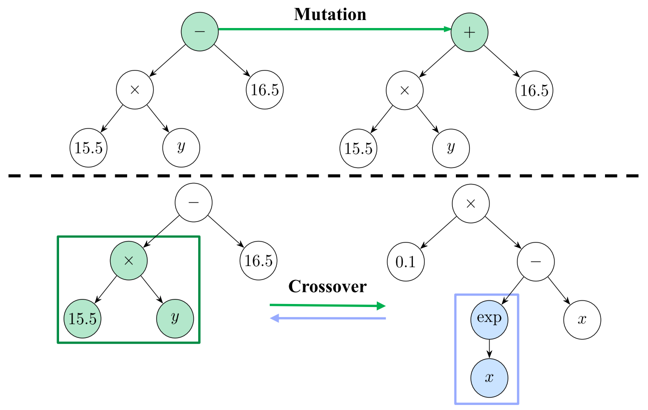


Fig. 6. Mutation and cross-over operations applied to the candidate solutions.

The genetic-programming-based SR [36] starts with populations of randomly generated candidate solutions, constructed using predefined building blocks. The candidate solutions are represented as expression trees with internal nodes (mathematical operations or functions) and terminal nodes (variables or constants) [24,35]. Fig. 5 shows an example of the expression tree of the candidate solution $(-0.4\,x + \exp(15.5\,y - 16.5)) + 0.5$. The expression tree consists of binary operators $(+, -, \times)$, a unary operator $(\exp)$, variables (x, y), and numerical constants. The performance of each candidate solution in a population is evaluated according to how well it fits the target values. These populations evolve in parallel through a series of operations: mutation (Fig. 6), which randomly changes a candidate solution by, e.g., adding a node or mutating an operator, crossover (Fig. 6), which combines some information from two candidate solutions, simplification of mathematical expressions, and optimization of constants. These operations generate new sets of candidate solutions with improved fitness, and the process continues until a defined stopping criterion is satisfied.

In the present study, the considered binary operators are addition, subtraction, and multiplication, and the unary operators are chosen to be similar to the activation functions in the NN models, namely exponential $(\exp)$, hyperbolic tangent $(\tanh)$, and the Macaulay bracket $(\langle\bullet\rangle)$. For parameter identification within a given expression, the gradient-based BFGS optimization algorithm is employed. A total of 24 populations are considered, each with 5000 iterations. The SR process terminates when the total number of iterations across all populations is reached. The result of the SR is a Pareto front of analytical expressions with different complexities and resulting training losses. We assume the default expression complexity measure in PySR, the number of nodes in a symbolic expression tree, with a maximum complexity of 30.

6. Results and discussions

In this section, we present the results of the two-input and three-input NN, as well as SR models. In particular, we compare the corresponding training and validation losses and evaluate the performance of the cycle-domain material model when embedding the NN and SR models as the per-cycle evolution equations for plastic strains in the material model formulation.

6.1. Neural Networks (NN)

6.1.1. Two-input NN

The training procedure described in Section 5.1 resulted in the training and validation loss evolutions for the two-input NN presented in Fig. 7 (left). The minimum normalized training loss, $\hat{L}_{\text{train}}$, decreased from 9.6×10^{-5} in the first training run to 5.4×10^{-5} in the last run; however, after the second run, no noticeable improvement can be observed, with the loss slightly decreasing from 5.9×10^{-5} in the third run to 5.4×10^{-5} in the last run, see Fig. 7 (right). In contrast, the minimum $\hat{L}_{\text{train}}$ for the three-input NN continued decreasing until the last run, see Section 6.1.2. The corresponding normalized validation losses, $\hat{L}_{\text{valid}}$, are 7.3×10^{-5} in the first run and 4.4×10^{-5} in the last run. This indicates no sign of overfitting, as can be observed in Fig. 7 (left). Additionally, the normalized test loss, $\hat{L}_{\text{test}}$, of 5.7×10^{-5} in the last training run, shows a good performance of the model on unseen data. Fig. 8 illustrates the values of $\mathrm{d}\epsilon^{\mathrm{p}}_{11}/\mathrm{d}N$ along the z−axis obtained by the trained NN, and the colors represent relative errors with respect to the maximum value of all output features considering training, validation, and test data. Overall, there is good agreement between the surface-like distribution shown in Fig. 4(a) and the one predicted by the NN. However, in regions with lower data density and higher deviation from the surface-like trend, the relative errors in $\mathrm{d}\epsilon^{\mathrm{p}}_{11}/\mathrm{d}N$ are higher. As discussed in Section 5 and exemplified in Fig. 4(b), the NN model with two inputs results in an approximate solution due to the presence of one-to-many mappings. Therefore, we investigate, in the next section, whether informing the network with the third input ${}^{N-1}\sigma_{11}$ can lead to an enhancement of the NN performance.

6.1.2. Three-input NN

Fig. 9 (left) presents the evolution of the training and validation losses for the three-input NN. Retraining the NN reduced the minimum $\hat{L}_{\text{train}}$ from 3.6×10^{-5} in the first training run, to 3.0×10^{-5} in the second, 2.4×10^{-5} in the third, and 1.8×10^{-5} in the last run, see Fig. 9 (right). The

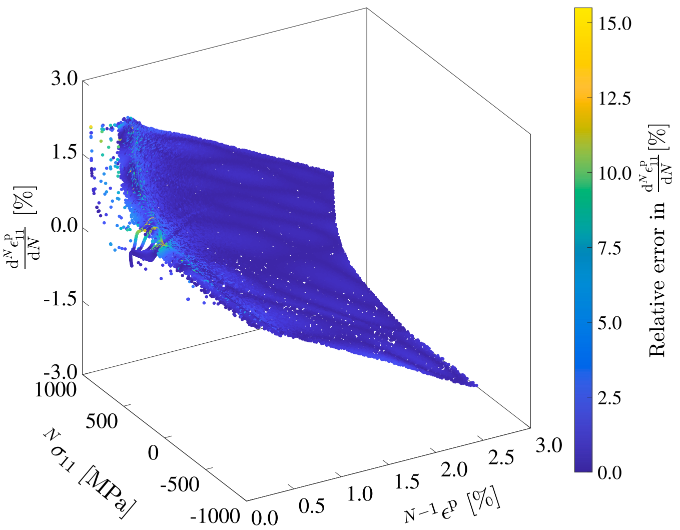


Fig. 8. Relative error in $\mathrm{d}\epsilon^{\mathrm{p}}_{11}/\mathrm{d}N$ from the two-input NN model considering training data.

corresponding $\hat{L}_{\text{valid}}$ and $\hat{L}_{\text{test}}$ in the last run are 1.7×10^{-5} and 2.1×10^{-5}, respectively. A comparison between the three-input and two-input NNs in the last training run shows that inclusion of the additional input feature, ${}^{N-1}\sigma_{11}$, has reduced the training loss by approximately 66 %, the validation loss by 61 %, and the test loss by 63 %. Moreover, a comparison of the validation losses indicates that expanding the input features of the network has not resulted in overfitting, and the test loss comparison implies improved generalizability.

To assess the performance of the cycle-domain material model, we embed the trained NN, f_{nn}, as the per-cycle evolution equation in the formulation of the cycle-domain model; see Eqs. (6) and (7) in Section 3. As mentioned in Section 3, we simulate the first half loading cycle with the time-domain material model, after which the cycle-domain model predicts σ_{11} and $\epsilon^{\mathrm{p}}_{11}$ for the following loading cycles. Fig. 10(a) and (b) present predicted stress and plastic strain values for a case selected among test datasets with variable strain ranges over 100 cycles, using the two- and three-input NNs. Overall, the predictions from the cycle-domain model show good agreement with the original data generated by the accurate time-domain material model, considering both neural network models. The mean absolute error in σ_{11} is 15.7 MPa and 10.2 MPa for the two- and three-input NN models, respectively. The corresponding relative errors in $\epsilon^{\mathrm{p}}_{11}$ (with respect to the reference value from the time-domain model) after 100 cycles are 0.007 % and 0.002 %.

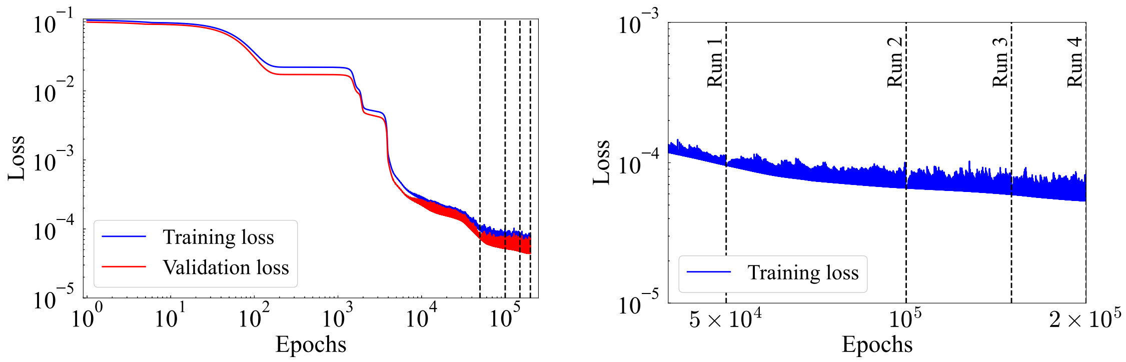


Fig. 7. Training and validation loss evolutions for the two-input NN model (left). A zoomed-in view of the former between epochs 4×10^4 and 2×10^5 is shown in the right figure. The vertical dashed lines correspond to the epoch number after each training run.

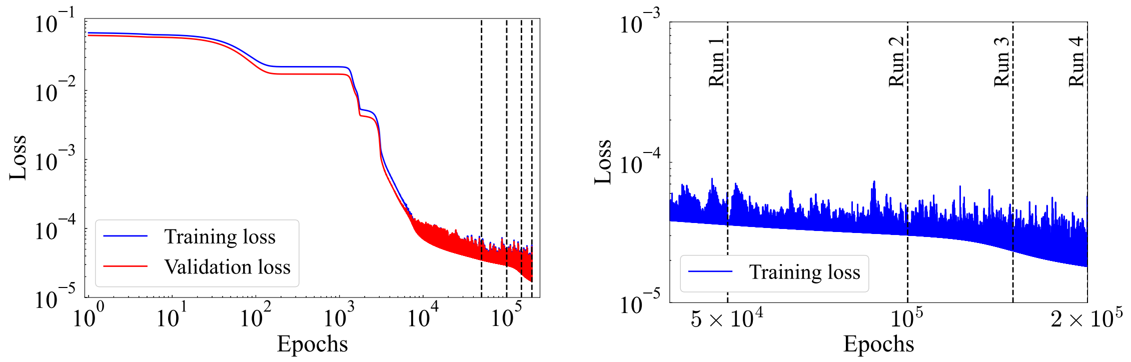
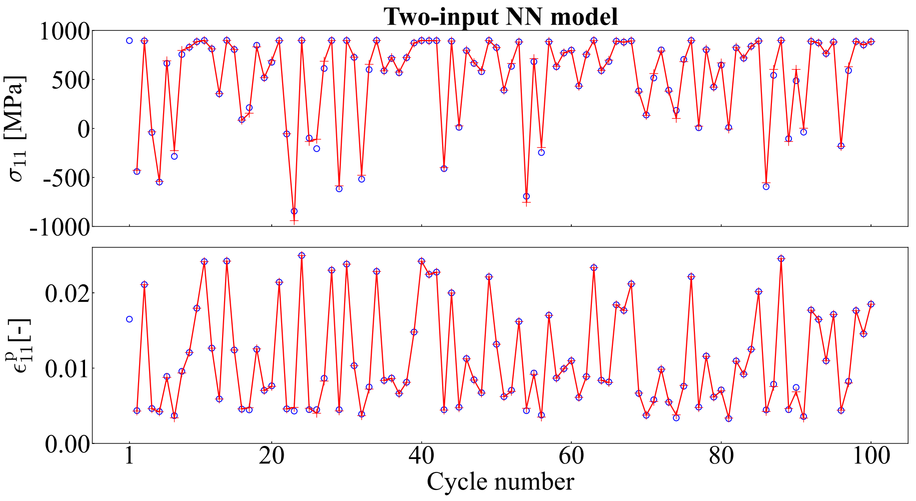
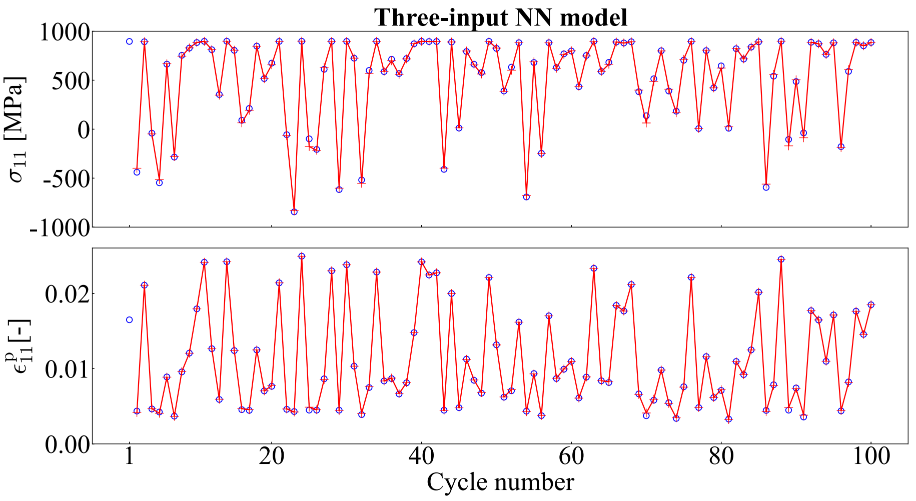


Fig. 9. Training and validation loss evolutions for the three-input NN model (left). A zoomed-in view of the former between epochs 4×10^4 and 2×10^5 is shown in the right figure. The vertical dashed lines correspond to the epoch number after each training run.



(a)



(b)

Fig. 10. Stress versus cycle number and plastic strain versus cycle number for a test dataset under loading with variable strain ranges over 100 loading cycles, when using (a) the two-input NN model and (b) the three-input NN model in the cycle-domain model formulation. Circles and red plus signs correspond to the results from the time-domain model and the cycle-domain model, respectively.

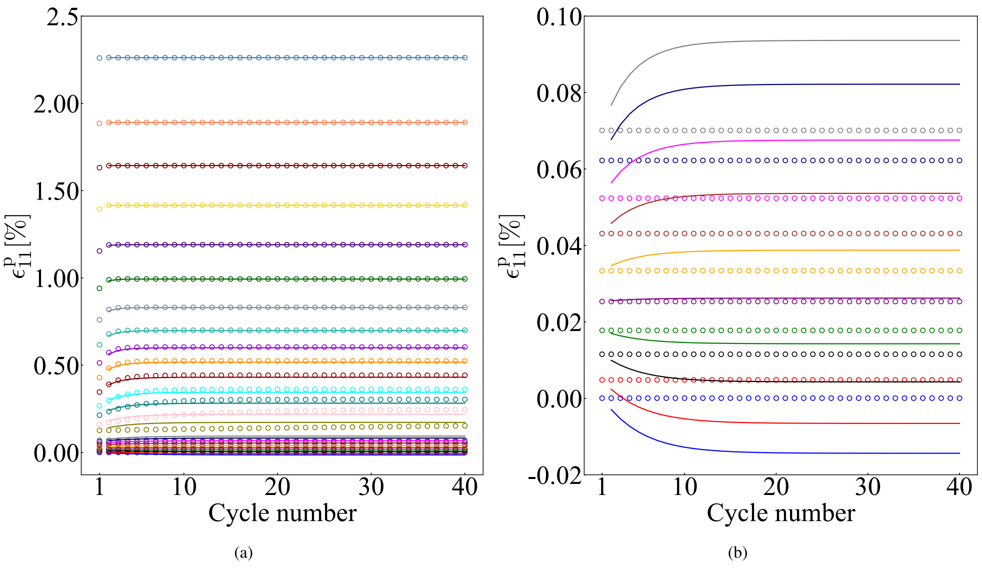


Fig. 11. Plastic strain evolutions for (a) 25 training datasets under constant strain range loadings and (b) a subset of 10 datasets from (a) that exhibit elastic shakedown from the second loading cycle. Circles and solid lines correspond to the results from the time-domain model and the cycle-domain model employing the three-input NN, respectively. Only the first 40 loading cycles are shown.

Although there is overestimation or underestimation in σ_{11} in some loading cycles, these errors have not led to accumulated error in $\epsilon_{11}^{\mathrm{p}}$ over 100 cycles. Additional results for selected cases among the training and validation datasets are provided in Appendix A. It is worth mentioning that the number of cycle increments in the cycle-domain model over 99 loading cycles is 98, in contrast to 39,600 time increments required by the time-domain model. A comparative assessment of the different ML models adopted in this study is presented in Section 6.3.

Considering the datasets under constant strain ranges, Fig. 11(a) presents the evolution of plastic strains for 25 training datasets with strain ranges from 0.2 % to 2.7 %. Since plastic strain saturates after a few cycles, for illustration purposes, only the first 40 cycles are shown. The cycle-domain model performs very well in predicting plastic strain evolutions for cases with higher strain ranges (that lead to higher plastic strains). However, when focusing on those datasets where elastic shakedown occurs after the first loading cycle, as shown in Fig. 11(b), the model's accuracy in predicting the corresponding plastic strain evolutions decreases. Nevertheless, in all cases, the plastic strains stabilize after a few cycles. These findings highlight the importance of considering a broad range of material responses in training datasets when using ML-based approaches in constitutive modeling to better understand and address the limitations of such models. Despite the limitation of the model under certain loading scenarios with constant strain ranges, it has shown satisfactory performance for variable strain range loadings. These are important in applications such as railway rails, as they are subjected to a wide range of vehicles with different axle loads, nominal/worn wheel profiles, and speeds [37].

To evaluate the extrapolation capability of the NN-based cycle-domain material model, we have simulated 5000 loading cycles with randomly generated variable strain ranges. Fig. 12, which presents only the last 100 cycles, demonstrates good agreement between the results from the reference time-domain and cycle-domain material model. The mean absolute error in σ_{11} after 5000 cycles is 10.1 MPa, showing satisfactory extrapolation behavior of the cycle-domain model employing the three-input NN as the per-cycle evolution law.

6.2. Symbolic Regression (SR)

In this section, we present and discuss the results of using SR to find the evolution equation for plastic strains in the cycle-domain material model. As shown in Section 6.1, the performance of the three-input NN was better than that of the two-input NN. Thus, we will only present results for SR using three inputs: ${}^{N}\sigma_{11}$, ${}^{N-1}\epsilon_{11}^{\mathrm{p}}$, and ${}^{N-1}\sigma_{11}$. For this regression task, the unary operators have been introduced sequentially:

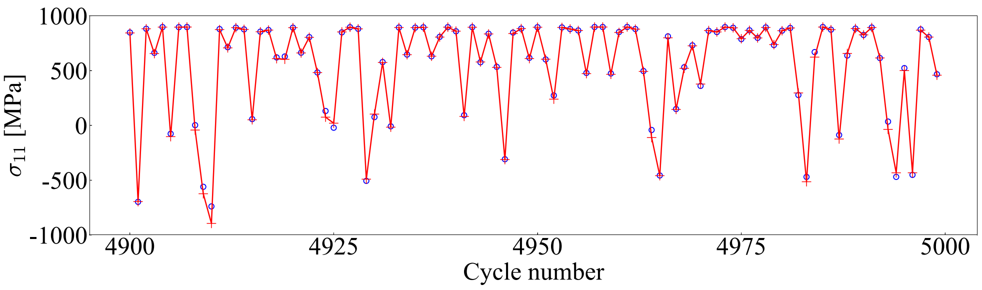


Fig. 12. Stress versus cycle number over 5000 loading cycles when employing the three-input NN in the cycle-domain model formulation. Only the last 100 loading cycles are shown. Circles and red plus signs correspond to the results from the time-domain model and the cycle-domain model, respectively.

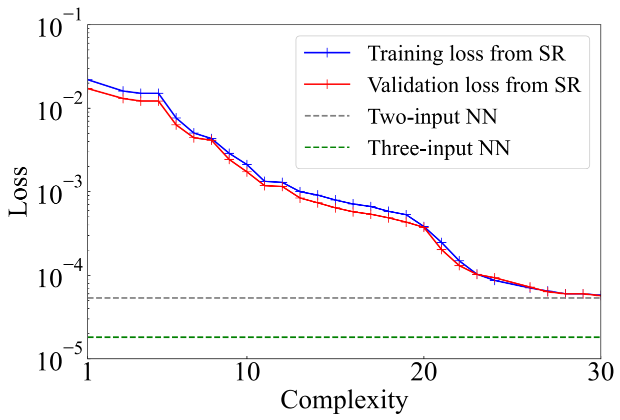


Fig. 13. Pareto front from PySR. The dashed gray and green lines show the minimum normalized training losses in the last training runs from the two- and three-input NN models, respectively.

first the exponential function, followed by the hyperbolic tangent, and finally the Macaulay bracket. Although the first and the third cases resulted in symbolic expressions with relatively low normalized training losses of 6.68×10^{-5} and 6.01×10^{-5}, respectively (considering the most complex equations), the second case with a slightly lower loss of 5.76×10^{-5} has been selected for further analysis and discussion in this section.

Following the procedure described in Section 5.2, Fig. 13 presents normalized training and validation losses ($\hat{L}_{\text{train}}$ and $\hat{L}_{\text{valid}}$, respectively) against the complexity of the symbolic expressions discovered by PySR. As expected, increasing equation complexity leads to improvements in both losses. However, beyond the complexity level of 28, a plateau appears to be emerging. Furthermore, there is no indication of overfitting, even at higher levels of complexity. It can be observed that the performance of the three-input SR model approaches that of the two-input NN, but not the three-input NN.

Among the discovered analytical expressions, five equations with different complexities are selected for further analysis and presented in Table 2. The first three equations with complexity levels of 6, 12, and 20 do not include the third input variable ${}^{N-1}\hat{\sigma}_{11}$, in contrast to the more complex equations that achieve lower values of $\hat{L}_{\text{train}}$ and $\hat{L}_{\text{valid}}$.

A comparison between the results of the NN models and symbolic expressions shows that the training and validation losses of the equations are higher than those of the three-input NN model (1.8×10^{-5} and 1.7×10^{-5}, respectively), while they approximately converge to those of the two-input NN model (5.4×10^{-5} and 4.4×10^{-5}).

We have embedded the equations presented in Table 2 into the formulation of the cycle-domain material model following the procedure described in Section 6.1. Fig. 14 presents a comparison of the predicted σ_{11} by the four symbolic expressions, considering a training dataset. The mean absolute errors in σ_{11} for the equations with complexity levels of 12, 20, 26, and 30 are 44.4 MPa, 26.8 MPa, 19.1 MPa, and 10.9 MPa, respectively, showing more accurate predictions with increasing complexity at the expense of reduced interpretability. Note that the corresponding result from the simplest equation is excluded from Fig. 14, due to its high error of 193.1 MPa in σ_{11}.

We have selected the most accurate symbolic expression (with a complexity level of 30) as the per-cycle evolution equation for plastic strains in the cycle-domain material model and presented its predictions for σ_{11} and ϵ^{p}_{11} for the test dataset in Fig. 15. A reasonably good fit to the results from the time-domain model can be observed. The mean absolute error in σ_{11} is 16.6 MPa, and the relative error in ϵ^{p}_{11} after 100 cycles is 0.018 %. A similar conclusion to that in Section 6.1.2 can be drawn that the errors in σ_{11} did not noticeably influence the predicted ϵ^{p}_{11} values after 100 cycles. Table 3 summarizes the errors in σ_{11} and ϵ^{p}_{11} from the different ML models for the considered test dataset, showing a reasonably good performance from the NN-based and the SR-based cycle-domain models. Further discussion regarding the performance of the ML models is presented in Section 6.3. Supplementary results for the selected training and validation datasets are provided in Appendix A.

Regarding the extrapolation performance of the SR-based cycle-domain model, the simulation of 5000 cycles has resulted in a mean absolute error of 14.8 MPa in σ_{11}. This error is comparable to that of the NN model and shows that the SR-based cycle-domain model extrapolates similarly well beyond the training regime.

6.3. Comparison of different ML models

According to Table 4, presenting the normalized errors for the ML models evaluated in this paper, the three-input NN model, with 199 parameters, has the lowest training, validation, and test losses. Moreover, the SR model with complexity 30 performs similarly to the two-input NN. A comparison of the errors in the predicted σ_{11} for the

Table 2
Discovered symbolic expressions, along with their corresponding complexity levels, normalized training losses ($\hat{L}_{\text{train}}$), and normalized validation losses ($\hat{L}_{\text{valid}}$). ${}^{N}\hat{\sigma}_{11}$, ${}^{N-1}\hat{\epsilon}^{\text{p}}_{11}$, and ${}^{N-1}\hat{\sigma}_{11}$ represent the min-max normalized ${}^{N}\sigma_{11}$, ${}^{N-1}\epsilon^{\text{p}}_{11}$, and ${}^{N-1}\sigma_{11}$, respectively.

Symbolic expression	Complexity	$\hat{L}_{\text{train}}$	$\hat{L}_{\text{valid}}$
$0.24\left(\exp({}^{N}\hat{\sigma}_{11}) - {}^{N-1}\hat{\epsilon}^{\text{p}}_{11}\right)$	6	7.7×10^{-3}	6.3×10^{-3}
$\left(-0.46\,{}^{N-1}\hat{\epsilon}^{\text{p}}_{11} + \exp\left(15.52\,{}^{N}\hat{\sigma}_{11} - 16.52\right)\right) + 0.50$	12	1.3×10^{-3}	1.2×10^{-3}
$\exp\left(167.37\,{}^{N}\hat{\sigma}_{11} - 168.90\right) - \exp\left(-8.75 \times 10^{-5} \exp\left(8.02\,{}^{N}\hat{\sigma}_{11}\right)\right) - 0.59 \tanh\left({}^{N-1}\hat{\epsilon}^{\text{p}}_{11} - 1.00\right)$	20	3.8×10^{-4}	3.8×10^{-4}
$\exp\left(167.38\,{}^{N}\hat{\sigma}_{11} - 168.90\right) + 1.93 \times 10^{-4} \exp\left(7.16\,{}^{N}\hat{\sigma}_{11}\right) + 0.47 - 0.09\,{}^{N-1}\hat{\sigma}_{11} \left({}^{N-1}\hat{\sigma}_{11} + {}^{N-1}\hat{\epsilon}^{\text{p}}_{11}\right) \exp\left({}^{N-1}\hat{\epsilon}^{\text{p}}_{11}\right)$	26	7.1×10^{-5}	7.2×10^{-5}
$\left(\exp\left(167.36\,{}^{N}\hat{\sigma}_{11} - 168.91\right) + 0.46 - \tanh\left(\tanh\left(\tanh\left(0.08 \exp\left(2\,{}^{N-1}\hat{\epsilon}^{\text{p}}_{11}\right)\left({}^{N-1}\hat{\sigma}_{11}\right)^{2}\right)\right)\right)\right) + 1.23 \times 10^{-4} \exp\left(\exp\left({}^{N}\hat{\sigma}_{11} + 1.04\right)\right)$	30	5.8×10^{-5}	5.7×10^{-5}

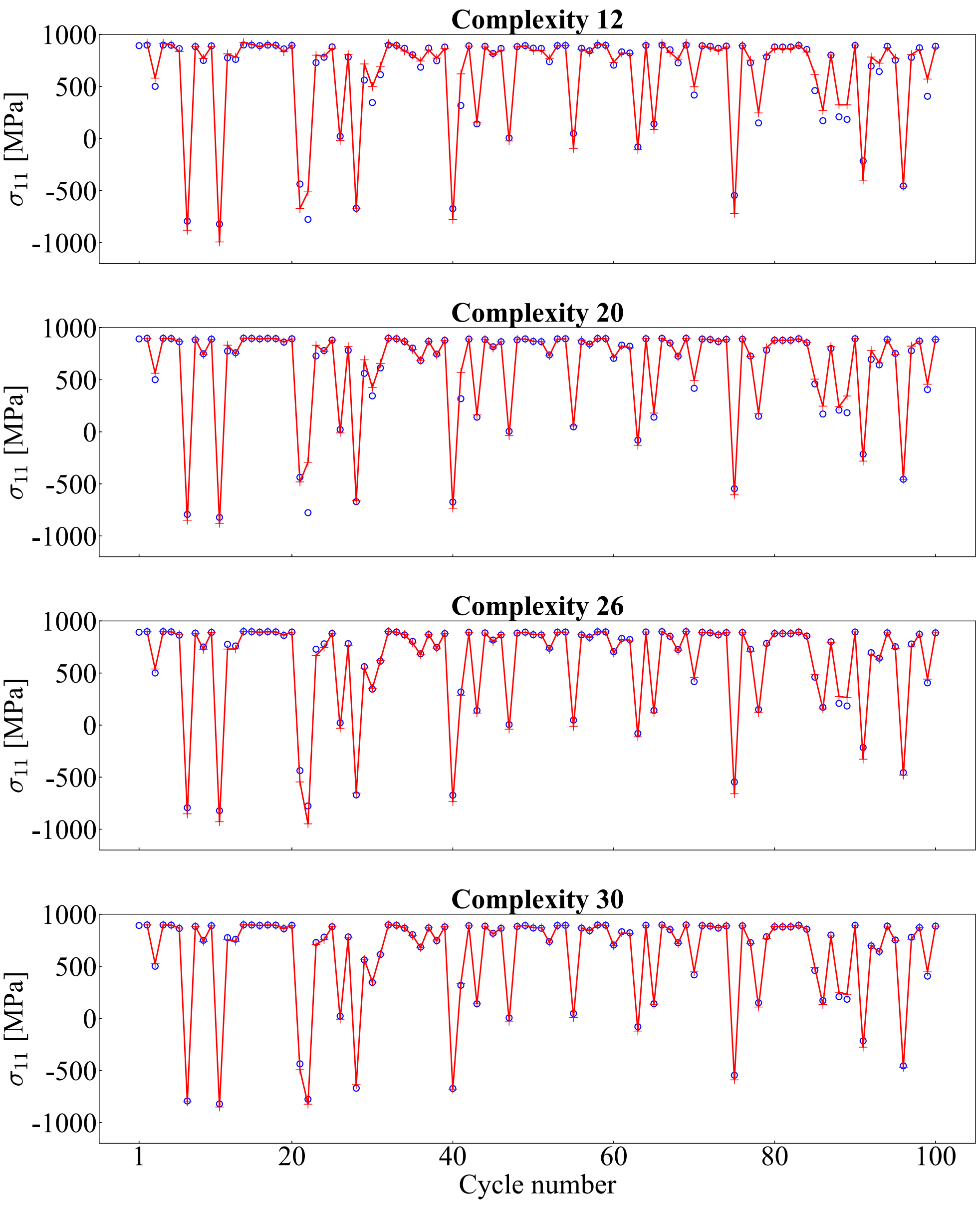


Fig. 14. Stress versus cycle number from symbolic expressions with different complexities according to Table 2, considering a training dataset. Circles and red plus signs correspond to the results from the time-domain model and the cycle-domain model, respectively.

ML models is provided in Table 5. The average mean absolute errors over 100 loading cycles, considering all the training, validation, and test datasets under variable strain range loadings, are noticeably reduced, from the two-input to the three-input NN model. Additionally, the three-input symbolic expression shows similar errors to those from the two-input NN, but with higher minimum values of the mean absolute errors.

In general, the results show the potential of the evaluated ML models, i.e., FFNNs and SR, as tools for formulating the per-cycle evolution law of plastic strains in the cycle-domain plasticity model. However, each approach has its limitations. In particular, as mentioned in Section 1, FFNNs lead to uninterpretable models, where the influence of input features or model parameters on the outputs can become highly difficult to explain in networks with multivariate inputs [24] and complex architectures. Conversely, SR achieves explicit analytical equations, which can allow for analyzing the model behavior. However, there is a trade-off between interpretability and accuracy, but the resulting symbolic expressions are more computationally efficient compared to FFNNs by requiring fewer floating-point operations. Obtaining a well-performing symbolic expression, however, requires providing suitable unary and/or binary operators (and/or constraints on them) based on users' prior knowledge or based on the distribution of training data, despite the advantage of not requiring a predefined functional form. Further, the discovered expressions should be rigorously analyzed to ensure that they lead to reliable results when employed in material point or FE simulations.

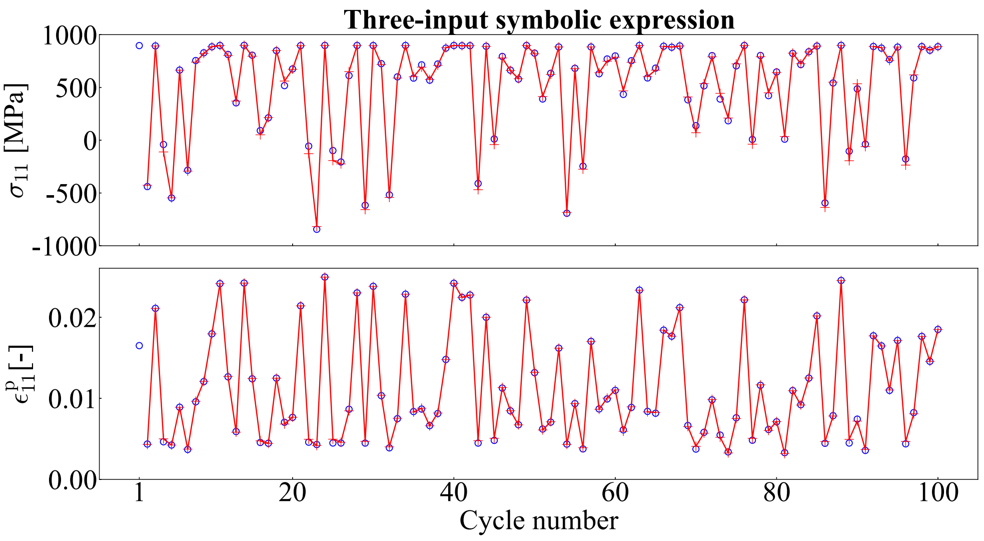


Fig. 15. Stress versus cycle number and plastic strain versus cycle number for a test dataset under loading with variable strain ranges over 100 loading cycles, when using the three-input symbolic expression with complexity level of 30 in the cycle-domain model formulation. Circles and red plus signs correspond to the results from the time-domain model and the cycle-domain model, respectively.

Table 3
Mean absolute error (MAE) in σ_{11} over 100 loading cycles and relative error (RE) in ϵ_{11}^{p} in the last loading cycle for the considered test dataset.

Model	MAE in σ_{11} [MPa]	RE in ϵ_{11}^{p} [%] ($\times 10^{-2}$)
f_{nn} (2 inputs)	15.7	0.7
f_{nn} (3 inputs)	10.2	0.2
f_{sr} (3 inputs, complexity 30)	16.6	1.8

Table 4
Normalized training, validation, and test losses ($\hat{L}_{train}$, $\hat{L}_{valid}$, and $\hat{L}_{test}$, respectively) for different ML models.

Model	$\hat{L}_{train}$ ($\times 10^{-5}$)	$\hat{L}_{valid}$ ($\times 10^{-5}$)	$\hat{L}_{test}$ ($\times 10^{-5}$)
f_{nn} (2 inputs)	5.4	4.4	5.7
f_{nn} (3 inputs)	1.8	1.7	2.1
f_{sr} (3 inputs, complexity 30)	5.8	5.7	6.1

Table 5
Error in σ_{11} for different ML models used in the cycle-domain material model, considering all training, validation, and test datasets with variable strain range loadings over 100 loading cycles. The values denote the mean (minimum, maximum) of the mean absolute errors computed for each dataset.

Model	Train [MPa]	Validation [MPa]	Test [MPa]
f_{nn} (2 inputs)	15.1 (5.6, 36.9)	15.6 (7.4, 24.9)	14.9 (7.7, 27.0)
f_{nn} (3 inputs)	10.2 (5.1, 19.5)	10.5 (6.1, 20.6)	10.5 (6.7, 18.0)
f_{sr} (3 inputs, complexity 30)	15.0 (9.0, 23.0)	15.1 (11.3, 21.1)	16.2 (11.2, 25.5)

7. Concluding remarks

In this contribution, we have investigated the potential of using ML to formulate the per-cycle evolution equation for plastic strains in cycle-domain material models. Compared to traditional time-domain models, these models aim to reduce the computational time of cyclic simulations when involving many loading cycles, such as FE simulations of many wheel over-rollings in railway mechanics. In the proposed framework, we have enforced our knowledge from constitutive modeling for elasticity and have employed FFNNs with different inputs to formulate the evolution equation, in addition to SR as an ML tool for equation discovery. To generate training, validation, and test data, cyclic pulsating uniaxial stress loadings with constant and variable strain ranges over a load sequence of 100 cycles using a standard time-domain material model have been considered. The proposed ML-based cycle-domain model predicts the peak stress and the corresponding plastic strain in each loading cycle.

The NN model with more input features resulted in lower training, validation, and test losses, although all the ML tools demonstrated good capability. Moreover, when embedded in the cycle-domain model formulation as the evolution equation, the extended-input NN provided better agreement with the reference time-domain model. Although the discovered symbolic expressions are human-readable compared to the NN models, their interpretability decreases with increasing accuracy. Despite this, both ML tools performed effectively when extrapolated to 5000 loading cycles, which is more than 50 times the number of cycles in the training data.

CRediT authorship contribution statement

Nasrin Talebi: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Knut Andreas Meyer:** Writing – review & editing, Supervision, Methodology, Data curation, Conceptualization. **Magnus Ekh:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

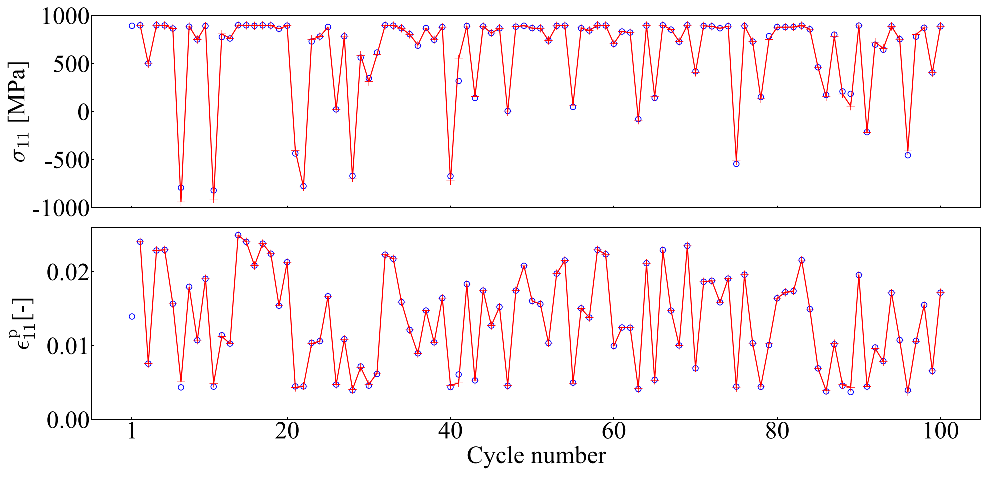
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

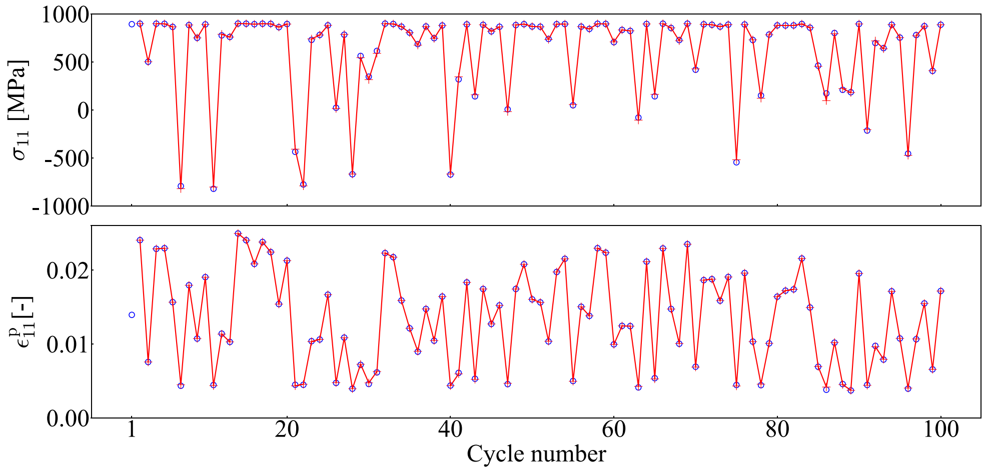
This work is part of the ongoing activities within the National Center of Excellence CHARMEC (www.chalmers.se/charmec). Parts of the study have been funded by Europe's Rail project IAM4RAIL under grant agreement No. 101101966. The majority of the numerical simulations in this study were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at Chalmers Center for Computational Science and Engineering (C3SE) partially funded by the Swedish Research Council through a grant agreement No. 2022-06725.

Appendix A. Cycle-domain performance evaluation considering training and validation datasets

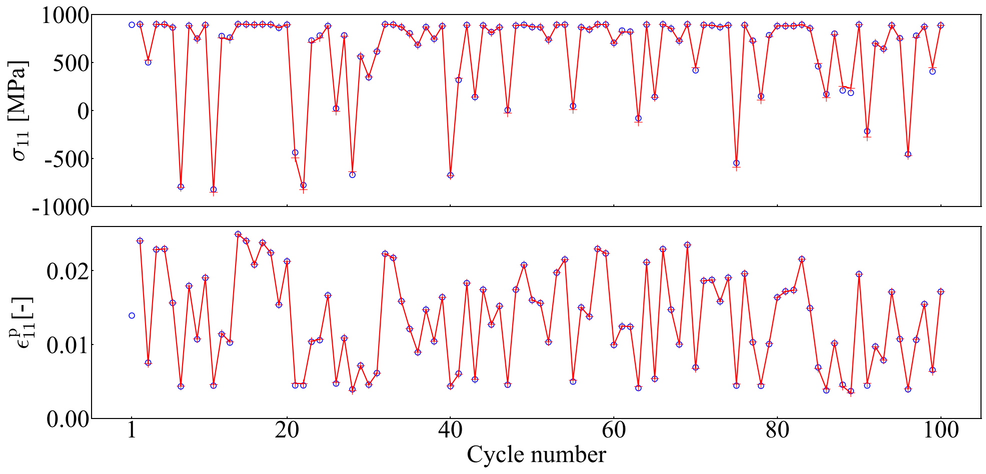
This section presents the predicted σ_{11} and $\epsilon_{11}^{\mathrm{p}}$ from the evaluated ML-based cycle-domain models for cases selected among the training datasets (Fig. A.16) and validation datasets (Fig. A.17).



(a) Results from the two-input NN model

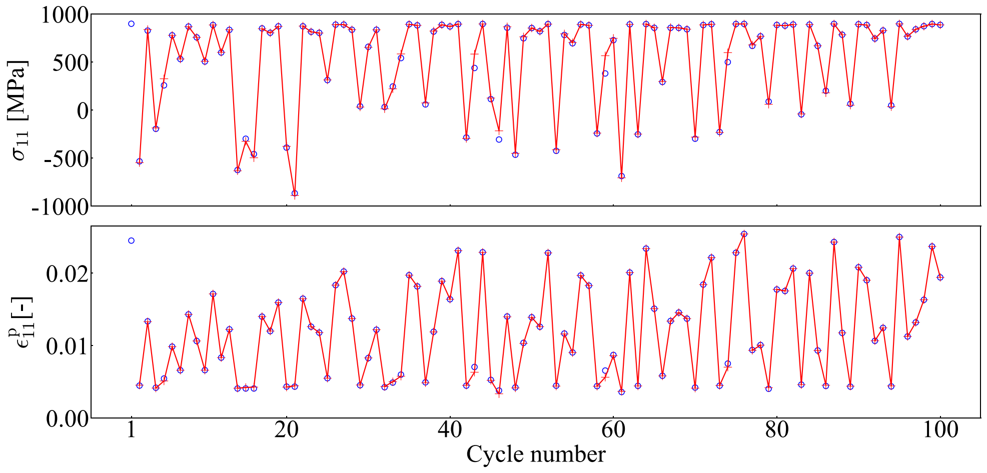


(b) Results from the three-input NN model

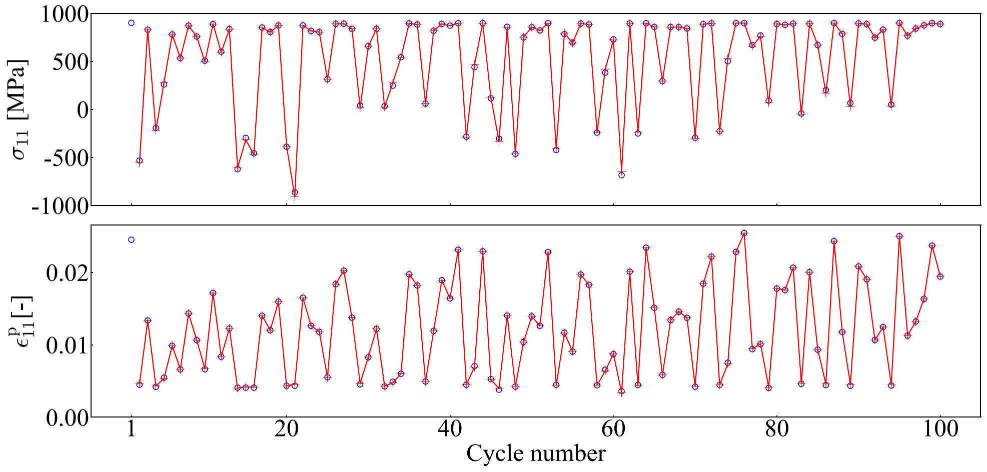


(c) Results from the three-input symbolic expression with complexity level of 30

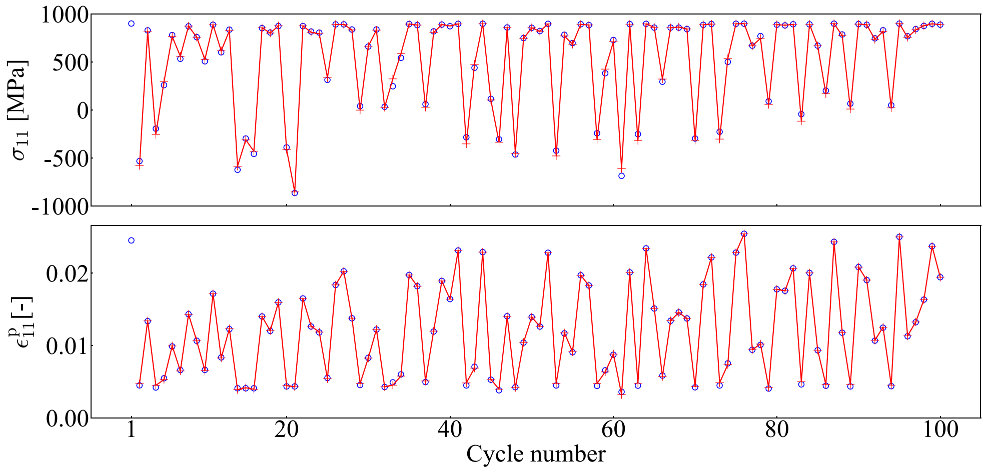
Fig. A.16. Stress versus cycle number and plastic strain versus cycle number for a training dataset under loading with variable strain ranges over 100 loading cycles, when using different ML models in the cycle-domain model formulation. Circles and red plus signs correspond to the results from the time-domain model and the cycle-domain model, respectively.



(a) Results from the two-input NN model



(b) Results from the three-input NN model



(c) Results from the three-input symbolic expression with complexity level of 30

Fig. A.17. Stress versus cycle number and plastic strain versus cycle number for a validation dataset under loading with variable strain ranges over 100 loading cycles, when using different ML models in the cycle-domain model formulation. Circles and red plus signs correspond to the results from the time-domain model and the cycle-domain model, respectively.

Data availability

Data will be made available on request.

References

[1] Leidermark D, Simonsson K. Procedures for handling computationally heavy cyclic load cases with application to a disc alloy material. Mater High Temp 2019;36(5):447–58. https://doi.org/10.1080/09603409.2019.1631587

[2] Johansson G, Ekh M. On the modeling of large ratcheting strains with large time increments. Eng Comput 2007;24(3):221–36. https://doi.org/10.1108/02644400710734945

[3] Brommesson R, Ekh M, Hörnqvist M. Correlation between crack length and load drop for low-cycle fatigue crack growth in Ti-6242. Int J Fatigue 2015;81:1–9. https://doi.org/10.1016/j.ijfatigue.2015.07.006

[4] Suiker ASJ, de Borst R. A numerical model for the cyclic deterioration of railway tracks. Int J Numer Methods Eng 2003;57(4):441–70. https://doi.org/10.1002/nme.683

[5] Li X, Ekh M, Nielsen JCO. Three-dimensional modelling of differential railway track settlement using a cycle domain constitutive model. Int J Numer Anal Methods Geomech 2016;40(12):1758–70. https://doi.org/10.1002/nag.2515

[6] Köbler J, Magino N, Andrä H, Welschinger F, Müller R, Schneider M. A computational multi-scale model for the stiffness degradation of short-fiber reinforced plastics subjected to fatigue loading. Comput Methods Appl Mech Eng 2021;373:113522. https://doi.org/10.1016/j.cma.2020.113522

[7] Magino N, Köbler J, Andrä H, Welschinger F, Müller R, Schneider M. A space-time upscaling technique for modeling high-cycle fatigue-damage of short-fiber reinforced composites. Compos Sci Technol 2022;222:109340. https://doi.org/10.1016/j.compscitech.2022.109340

[8] Fuhg JN, Anantha Padmanabha G, Bouklas N, Bahmani B, Sun W, Vlassis NN, Flaschel M, Carrara P, De Lorenzis L. A review on data-driven constitutive laws for solids. Arch Comput Methods Eng 2024:1–43. https://doi.org/10.1007/s11831-024-10196-2

[9] Ali U, Muhammad W, Brahme A, Skiba O, Inal K. Application of artificial neural networks in micromechanics for polycrystalline metals. Int J Plast 2019;120:205–19. https://doi.org/10.1016/j.ijplas.2019.05.001

[10] Gorji MB, Mozaffar M, Heidenreich JN, Cao J, Mohr D. On the potential of recurrent neural networks for modeling path dependent plasticity. J Mech Phys Solids 2020;143:103972. https://doi.org/10.1016/j.jmps.2020.103972

[11] Linka K, Hillgärtner M, Abdolazizi KP, Aydin RC, Itskov M, Cyron CJ. Constitutive artificial neural networks: a fast and general approach to predictive data-driven constitutive modeling by deep learning. J Comput Phys 2021;429:110010. https://doi.org/10.1016/j.jcp.2020.110010

[12] Rosenkranz M, Kalina KA, Brummund J, Kästner M. A comparative study on different neural network architectures to model inelasticity. Int J Numer Methods Eng 2023;124(21):4802–40. https://doi.org/10.1002/nme.7319

[13] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707. https://doi.org/10.1016/j.jcp.2018.10.045

[14] Abueidda DW, Lu Q, Koric S. Meshless physics-informed deep learning method for three-dimensional solid mechanics. Int J Numer Methods Eng 2021;122(23):7182–201. https://doi.org/10.1002/nme.6828

[15] Haghighat E, Abouali S, Vaziri R. Constitutive model characterization and discovery using physics-informed deep learning. Eng Appl Artif Intell 2023;120:105828. https://doi.org/10.1016/j.engappai.2023.105828

[16] Meyer KA, Ekre F. Thermodynamically consistent neural network plasticity modeling and discovery of evolution laws. J Mech Phys Solids 2023;180:105416. https://doi.org/10.1016/j.jmps.2023.105416

[17] Fuhg JN, Hamel CM, Johnson K, Jones R, Bouklas N. Modular machine learning-based elastoplasticity: generalization in the context of limited data. Comput Methods Appl Mech Eng 2023;407:115930. https://doi.org/10.1016/j.cma.2023.115930

[18] Jadoon AA, Meyer KA, Fuhg JN. Automated model discovery of finite strain elastoplasticity from uniaxial experiments. Comput Methods Appl Mech Eng 2025;435:117653. https://doi.org/10.1016/j.cma.2024.117653

[19] Flaschel M, Kumar S, De Lorenzis L. Unsupervised discovery of interpretable hyperelastic constitutive laws. Comput Methods Appl Mech Eng 2021;381:113852. https://doi.org/10.1016/j.cma.2021.113852

[20] Flaschel M, Kumar S, De Lorenzis L. Automated discovery of generalized standard material models with Euclid. Comput Methods Appl Mech Eng 2023;405:115867. https://doi.org/10.1016/j.cma.2022.115867

[21] Versino D, Tonda A, Bronkhorst CA. Data driven modeling of plastic deformation. Comput Methods Appl Mech Eng 2017;318:981–1004. https://doi.org/10.1016/j.cma.2017.02.016

[22] Bomarito GF, Townsend TS, Stewart KM, Esham KV, Emery JM, Hochhalter JD. Development of interpretable, data-driven plasticity models with symbolic regression. Comput Struct 2021;252:106557. https://doi.org/10.1016/j.compstruc.2021.106557

[23] Abdusalamov R, Hillgärtner M, Itskov M. Automatic generation of interpretable hyperelastic material models by symbolic regression. Int J Numer Methods Eng 2023;124(9):2093–104. https://doi.org/10.1002/nme.7203

[24] Bahmani B, Suh HS, Sun W. Discovering interpretable elastoplasticity models via the neural polynomial method enabled symbolic regressions. Comput Methods Appl Mech Eng 2024;422:116827. https://doi.org/10.1016/j.cma.2024.116827

[25] Bari S, Hassan T. An advancement in cyclic plasticity modeling for multiaxial ratcheting simulation. Int J Plast 2002;18(7):873–94. https://doi.org/10.1016/S0749-6419(01)00012-2

[26] Simo JC, Hughes TJR. Computational inelasticity. Springer; 1998. https://doi.org/10.1007/b98904

[27] Bernasconi A, Filippini M, Foletti S, Vaudo D. Multiaxial fatigue of a railway wheel steel under non-proportional loading. Int J Fatigue 2006;28(5–6):663–72. https://doi.org/10.1016/j.ijfatigue.2005.07.045

[28] Armstrong PJ, Frederick CO, et al. A mathematical representation of the multiaxial Bauschinger effect, vol. 731. Berkeley Nuclear Laboratories Berkeley, CA; 1966. https://doi.org/10.3184/096034007x207589

[29] Talebi N, Andersson B, Ekh M, Meyer KA. Influence of a highly deformed surface layer on RCF predictions for rails in service. Wear 2025:206173. https://doi.org/10.1016/j.wear.2025.206173

[30] Mozaffar M, Bostanabad R, Chen W, Ehmann K, Cao J, Bessa MA. Deep learning predicts path-dependent plasticity. Proc Natl Acad Sci 2019;116(52):26414–20. https://doi.org/10.1073/pnas.1911815116

[31] Brunton SL, Kutz JN. Neural networks and deep learning. Cambridge University Press; 2019. pp. 195–226. https://doi.org/10.1017/9781108380690.007

[32] Liu L, Jiang H, He P, Chen W, Liu X, Gao J, Han J. On the variance of the adaptive learning rate and beyond [arXiv preprint] arXiv:1908.03265. 2019.

[33] Paszke A. PyTorch: an imperative style, high-performance deep learning library arXiv:1912.01703.[arXiv preprint] 2019.

[34] Feurer M, Hutter F. Hyperparameter optimization. In: Automated machine learning: methods, systems, challenges. Springer International Publishing Cham; 2019. pp. 3–33. https://doi.org/10.1007/978-3-030-05318-5

[35] Wang Y, Wagner N, Rondinelli JM. Symbolic regression in materials science. MRS Commun 2019;9(3):793–805. https://doi.org/10.1557/mrc.2019.85

[36] Cranmer M. Interpretable machine learning for science with PySR and SymbolicRegression.jl arXiv:2305.01582.[arXiv preprint] 2023.

[37] Magel EE. Rolling contact fatigue: a comprehensive review, Tech. rep., US Department of Transportation, Federal Railroad Administration, 132 p. (2011).