Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Data-Driven Modeling of 1D Cyclic Plasticity: A Comparative Study of Recurrent and Feedforward Neural Networks

Mohammad Ashraf-Ganjouei
Shahrood University of Technology
Seyed Hadi Ghaderi
s.h.ghaderi@gmail.com

Shahrood University of Technology
Seyed Mojtaba Varedi-Koulaei
Shahrood University of Technology

Research Article

Keywords:

Posted Date: November 19th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11626771/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Data-Driven Modeling of 1D Cyclic Plasticity: A Comparative Study of Recurrent and Feedforward Neural Networks

Mohammad Ashraf-Ganjouei[1], Seyed Hadi Ghaderi[1*], Seyed Mojtaba Varedi-Koulaei[1]

[1*]Faculty of Mechanical Engineering, Shahrood University of Technology, Shahrood, Iran.

*Corresponding author(s). E-mail(s): s.h.ghaderi@gmail.com;
Contributing authors: Mohammad.Ganjouei@shahroodut.ac.ir; varedi@shahroodut.ac.ir;

Abstract

Constitutive modeling of materials under cyclic loading remains computationally demanding in solid mechanics. Classical combined isotropic–kinematic hardening models, such as Voce and Armstrong–Frederick formulations, require an iterative return-mapping algorithm at each integration point, which is costly in large-scale finite element (FE) simulations. Data-driven machine learning (ML) surrogates offer an alternative, but standard memoryless feedforward networks (multilayer perceptrons, MLPs) cannot represent path-dependent elastoplastic behavior unless internal state variables (e.g., backstress and accumulated plastic strain) are supplied as inputs; these variables are often unavailable in experiments. This work compares recurrent neural network architectures—long short-term memory (LSTM) networks and gated recurrent units (GRUs)—with feedforward MLPs for history-dependent cyclic plasticity. Synthetic datasets are generated with a custom Abaqus UMAT implementation of combined isotropic–kinematic hardening under non-monotonic, multi-amplitude strain paths. A sliding-window formulation lets the recurrent hidden states act as implicit surrogates for internal variables. Hyperparameter studies show that recurrent models reduce validation mean squared error (MSE) by more than an order of magnitude relative to the MLP baseline (from $\mathbf{3.960 \times 10^{-3}}$ to $\mathbf{2.410 \times 10^{-4}}$ for a two-layer LSTM with $\boldsymbol{W = 15}$). The recurrent models reproduce hysteresis loops and sharp reversal vertices without the path-dependent drift seen in the MLP. Among the tested configurations, the single-layer GRU ($\boldsymbol{W = 15}$) gives the lowest mean absolute error and reversal error, while the two-layer LSTM ($\boldsymbol{W = 15}$) gives the lowest validation MSE. The results provide practical guidance for using history-aware neural surrogates in nonlinear FE solvers.

Keywords: Cyclic Plasticity, Combined isotropic-kinematic hardening, Return-Mapping Algorithm, Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Data-Driven Constitutive Modeling

1 Introduction

1.1 Background and Classical Cyclic Plasticity

Predicting inelastic material response under cyclic loading is central to structural integrity assessment, low-cycle fatigue prediction, and metal forming analysis [1, 2]. Under cyclic deformation, metallic alloys exhibit pronounced history-dependent effects, including the Bauschinger effect, cyclic hardening or softening, mean-stress relaxation, and ratcheting [3, 4]. Phenomenological continuum plasticity captures these mechanisms by decomposing hardening into an isotropic part, which expands or contracts the yield surface uniformly in

stress space, and a kinematic part, which translates the yield-surface centre through the backstress tensor $\boldsymbol{\alpha}$ [5, 6].

Among classical formulations, the nonlinear kinematic hardening rule of Armstrong and Frederick [7], together with its multi-component extensions by Chaboche [8, 9], provides a physically grounded representation of transient dynamic recovery during plastic flow. Coupled with the exponential isotropic hardening law of Voce [10], the one-dimensional yield criterion takes the form

$$f(\sigma, \alpha, \bar{\varepsilon}^p) = |\sigma - \alpha| - \sigma_y(\bar{\varepsilon}^p) \leq 0, \tag{1}$$

where σ is the Cauchy stress, α is the scalar backstress, and $\sigma_y(\bar{\varepsilon}^p)$ is the evolving yield stress expressed as a saturation function of the accumulated equivalent plastic strain $\bar{\varepsilon}^p$.

In implicit finite element (FE) formulations, integrating these rate-independent elastoplastic equations over a discrete time increment $[t_n, t_{n+1}]$ requires robust return-mapping algorithms [2, 11, 12]. Each local update involves a Newton–Raphson iteration for the plastic consistency parameter $\Delta\gamma$, followed by evaluation of the algorithmic consistent tangent operator $\mathbb{C}^{\mathrm{ep}} = \mathrm{d}\sigma_{n+1}/\mathrm{d}\varepsilon_{n+1}$ to preserve the asymptotic quadratic convergence of the global equilibrium iterations [12]. Although mathematically rigorous, this nested procedure is executed at millions of material integration points and dominates the cost of large-scale cyclic-plasticity simulations; calibrating the underlying internal variables further requires nonlinear parameter identification [13, 14]. These bottlenecks have motivated the search for efficient, data-driven surrogate constitutive models.

1.2 Data-Driven and Machine Learning Approaches in Plasticity

Over the past three decades, scientific machine learning (SciML) has been increasingly integrated into solid mechanics [15–17]. Early studies showed that feedforward neural networks (multilayer perceptrons, MLPs) could learn nonlinear stress–strain relationships directly. However, standard MLPs are memoryless mappings $\boldsymbol{\sigma}_{n+1} = \mathcal{N}(\boldsymbol{\varepsilon}_{n+1})$: in elastoplasticity they fail unless augmented with the underlying internal state variables ($\boldsymbol{\alpha}$, $\bar{\varepsilon}^p$) as explicit inputs. Huang et al. [18] demonstrated this by requiring accumulated absolute strain as an explicit history variable inside a proper-orthogonal-decomposition feedforward network, and Stoffel et al. [19] used convolutional networks for viscoplastic structural response, again with explicit history features. In practice, extracting or measuring such internal variables under arbitrary loading remains a major obstacle.

A widely explored remedy is to embed the neural network *inside* the return-mapping algorithm. Zhang and Mohr [20] used MLPs to map von Mises yield surfaces while retaining an explicit algorithmic structure; Teranishi [21] developed neural-network constitutive models for uniaxial cyclic plasticity in which a feedforward network replaces the local Newton iteration of a Chaboche-type return mapping, reporting accurate stress histories at roughly 70% of the original computational cost. While accurate, these hybrid models preserve the local iterative architecture and the explicit state-variable dependency of the classical solver; they accelerate the return mapping but do not eliminate it.

A second branch avoids the return mapping altogether and treats the constitutive update as a sequence-to-sequence learning problem. Dettmer et al. [22] proposed a generic recurrent stress-update framework for rate-independent elastic and inelastic materials, trained on stress–strain sequences, and showed that

piecewise-linear elastoplasticity can be represented exactly; their study also compared against GRU and LSTM baselines and proposed a thermodynamic-consistency verification criterion. Mozaffar et al. [23] and Gorji et al. [24] demonstrated that LSTM and GRU cells can reproduce path-dependent crystal-plasticity responses directly from raw strain sequences. Recent extensions include transfer learning for woven composites [25], neural controlled differential equations for path-dependent behaviour [26], self-consistency-reinforced gated recurrent units for homogenised responses [27], and temperature-dependent cyclic-plasticity surrogates built on feedforward networks [28].

A third, physics-constrained branch enforces thermodynamic or yield-surface restrictions directly in the training objective. Physics-Informed Neural Networks (PINNs) and thermodynamics-based models impose such constraints through the loss [29]; Hildebrand and Klinge [30, 31] extended this to 3D cyclic plasticity with backstress-aware physics regularisation and showed cyclic stability with peak-value deviations below 2%. Shoghi and Hartmaier [32] avoided explicit recurrence by coupling global history-dependent features with support-vector classification to capture mixed hardening in a single decision step, and physics-informed formulations for cyclic crystal plasticity [33] and thermodynamically constrained neural models for path-dependent geomaterials [34] have recently appeared. Despite this progress, standard LSTMs [35] and GRUs [36] remain the most natural candidates for learning path dependence directly from raw strain sequences, because their hidden states accumulate history without any externally supplied internal variables. What is still missing is a controlled comparison of how window length, recurrent depth, and gating topology (MLP vs. LSTM vs. GRU) affect the sharp reversal vertices and Bauschinger loops of a 1D cyclic-plasticity benchmark when *no* internal state variable is provided as input.

1.3 Knowledge Gaps, Novelty, and Research Objectives

Despite the growing literature on machine-learning surrogates, four technical gaps remain:

1. **Implicit internal-state representation.** Feedforward models require pre-computed internal variables $(\alpha_n, \bar{\varepsilon}^p_n)$ to reproduce cyclic behaviour [18, 19], and hybrid return-mapping networks [21, 22] inherit this dependency. A rigorous demonstration is needed of how recurrent cell states $(\boldsymbol{c}_t)$ in gated architectures function as implicit history representations driven solely by observable strain sequences.
2. **Comparative robustness at reversal vertices.** The ability of different architectures (MLP vs. LSTM vs. GRU) to avoid spurious path drift and capture localised reversal extrema—without cumulative error propagation over successive cycles—has not been benchmarked quantitatively against a common return-mapping reference.
3. **Optimisation of sliding-window parameters.** The trade-off between the temporal receptive field (window size W) and model performance has received limited attention; most recurrent works fix the window a priori [22, 23], leaving the joint effect of W, depth, and gating topology unexplored.
4. **Separation of global and reversal accuracy.** Existing studies report aggregated error metrics that mask failure at load-reversal points, where the material switches from plastic flow to elastic unloading and errors accumulate across cycles.

To address these gaps, this study establishes an end-to-end framework for modelling 1D cyclic plasticity (coupled Voce/Armstrong–Frederick hardening) using recurrent neural networks. The main contributions are:

- **High-fidelity dataset generation.** An implicit Newton–Raphson return-mapping algorithm implemented via an Abaqus UMAT generates stress–strain trajectories under complex, non-monotonic cyclic protocols.
- **Systematic deep-architecture benchmarking.** We perform an exhaustive hyperparameter study comparing memoryless feedforward networks, multi-layer LSTMs, and GRUs across temporal window lengths ($W \in \{15, 25\}$), isolating the effects of depth and gating topology.
- **Reversal-focused mechanical evaluation.** We show that stacked LSTM and GRU architectures substantially outperform MLPs, accurately capturing hysteresis-loop stabilisation and peak reversal points—including Bauschinger-type reverse yielding—without explicitly tracking internal thermodynamic variables, and we quantify the residual error at reversal vertices separately from the global error.

The remainder of this paper is structured as follows. Section 2 establishes the governing continuum plasticity formulation, including the backward-Euler return mapping and the consistent tangent operator used in the reference UMAT. Section 3 presents the finite element setup, dataset generation protocols, and the architectures of the MLP, LSTM, and GRU models. Section 4 presents the quantitative hyperparameter optimisation results and detailed evaluations of reversal-vertex tracking. Section 5 discusses the mechanical implications and limitations, and Section 6 provides concluding remarks.

2 Plasticity Theory and Computational Formulation

This section summarizes the one-dimensional elastoplastic model used to generate the reference data. The formulation combines Voce isotropic hardening with Armstrong–Frederick nonlinear kinematic hardening. The same equations are used in the Abaqus UMAT to produce the stress–strain histories employed in the subsequent machine-learning study.

2.1 Yield Criteria and Evolution Equations

The yield function defines the boundary between the elastic and plastic domains. In one dimension it is written as

$$f(\sigma, \alpha, \bar{\varepsilon}^p) = |\sigma - \alpha| - \sigma_y(\bar{\varepsilon}^p) \leq 0, \tag{2}$$

where σ is the uniaxial Cauchy stress, α is the backstress that locates the yield-surface center, and $\sigma_y(\bar{\varepsilon}^p)$ is the evolving yield stress. The accumulated equivalent plastic strain is denoted by $\bar{\varepsilon}^p$.

Isotropic hardening follows the Voce exponential saturation law:

$$\sigma_y(\bar{\varepsilon}^p) = \sigma_{y0} + (\sigma_u - \sigma_{y0}) \left[1 - \exp\left(-A\bar{\varepsilon}^p\right)\right], \tag{3}$$

where σ_{y0} is the initial yield stress, σ_u is the saturation yield stress, and A controls the saturation rate.

Kinematic hardening is described by the Armstrong–Frederick rule. In rate form,

$$\mathrm{d}\alpha = C\ \mathrm{sgn}(\sigma - \alpha)\,\mathrm{d}\bar{\varepsilon}^p - \gamma\alpha\,\mathrm{d}\bar{\varepsilon}^p, \tag{4}$$

where C is the initial kinematic hardening modulus and γ controls dynamic recovery. The first term on the right-hand side is the linear hardening contribution, and the second term limits backstress growth under large plastic strains.

The UMAT receives the elastic and hardening constants through the property array. No specific ordering is repeated here because the ordering must match the actual subroutine definition.

2.2 Flow Rule and Incremental Formulation

Under infinitesimal deformations, the total strain increment is split into elastic and plastic parts:

$$\Delta\varepsilon = \Delta\varepsilon^e + \Delta\varepsilon^p. \tag{5}$$

The plastic strain increment follows the associated flow rule:

$$\mathrm{d}\varepsilon^p = \mathrm{d}\lambda\frac{\partial f}{\partial\sigma} = \mathrm{d}\lambda\ \mathrm{sgn}(\sigma - \alpha), \tag{6}$$

where $\mathrm{d}\lambda \geq 0$ is the plastic multiplier. The equivalent plastic strain increment is defined as the absolute value of the plastic strain increment,

$$\mathrm{d}\bar{\varepsilon}^p = |\mathrm{d}\varepsilon^p|.$$

Substituting Eq. (6) gives

$$\mathrm{d}\bar{\varepsilon}^p = |\mathrm{d}\lambda\ \mathrm{sgn}(\sigma - \alpha)| = \mathrm{d}\lambda. \tag{7}$$

Thus, for this one-dimensional model,

$$\Delta\bar{\varepsilon}^p = \Delta\lambda.$$

During plastic loading the stress state must remain on the yield surface. The Kuhn–Tucker conditions are

$$f \leq 0, \qquad \Delta\lambda \geq 0, \qquad f\Delta\lambda = 0, \tag{8}$$

together with the consistency condition

$$\dot{f} = 0. \tag{9}$$

2.3 Backward-Euler Return Mapping Algorithm and Consistent Tangent

The rate equations are integrated with a fully implicit backward-Euler return-mapping scheme. The algorithm first assumes an elastic trial state. If the trial state lies outside the yield surface, a plastic corrector returns the stress to the updated yield surface.

Let the converged state at increment n be σ_n, α_n, and $\bar{\varepsilon}_n^p$. Given a total strain increment $\Delta\varepsilon$, the trial stress and trial relative stress are

$$\sigma^{\text{trial}} = \sigma_n + E\Delta\varepsilon, \tag{10}$$

$$\alpha^{\text{trial}} = \alpha_n, \tag{11}$$

$$\eta^{\text{trial}} = \sigma^{\text{trial}} - \alpha_n. \tag{12}$$

The trial yield function is

$$f^{\text{trial}} = |\eta^{\text{trial}}| - \sigma_{y,n}, \tag{13}$$

where

$$\sigma_{y,n} = \sigma_{y0} + (\sigma_u - \sigma_{y0})\left[1 - \exp\left(-A\bar{\varepsilon}_n^p\right)\right].$$

If $f^{\text{trial}} \leq 0$, the step is elastic.

If $f^{\text{trial}} > 0$, plastic flow occurs. Let

$$\Delta p = \Delta\lambda = \Delta\bar{\varepsilon}^p,$$

and let

$$s = \operatorname{sgn}(\eta^{\text{trial}}).$$

The backward-Euler update for the backstress is

$$\alpha_{n+1} = \frac{\alpha_n + Cs\Delta p}{1 + \gamma\Delta p}. \tag{14}$$

The corresponding increment is

$$\Delta\alpha = \alpha_{n+1} - \alpha_n = \frac{Cs\Delta p - \gamma\alpha_n\Delta p}{1 + \gamma\Delta p}.$$

The local residual for the plastic multiplier is

$$\Phi(\Delta p) = f^{\text{trial}} - E\Delta p - s\Delta\alpha - \left[\sigma_y(\bar{\varepsilon}_n^p + \Delta p) - \sigma_{y,n}\right]. \tag{15}$$

The Newton–Raphson derivative is

$$\Phi'(\Delta p) = -E - H^{\text{iso}} - H^{\text{kin}}, \tag{16}$$

with the algorithmic hardening moduli

$$H^{\text{iso}} = A(\sigma_u - \sigma_{y0})\exp\left[-A(\bar{\varepsilon}_n^p + \Delta p)\right], \tag{17}$$

$$H^{\text{kin}} = \frac{C - \gamma\alpha_n s}{(1 + \gamma\Delta p)^2}. \tag{18}$$

The plastic multiplier is updated until $|\Phi|$ falls below the tolerance.

After convergence, the state variables are updated as

$$\bar{\varepsilon}^p_{n+1} = \bar{\varepsilon}^p_n + \Delta p, \tag{19}$$

$$\varepsilon^p_{n+1} = \varepsilon^p_n + \Delta p\, s, \tag{20}$$

$$\sigma_{n+1} = \sigma^{\text{trial}} - E\Delta p\, s, \tag{21}$$

$$\alpha_{n+1} = \frac{\alpha_n + Cs\Delta p}{1 + \gamma\Delta p}. \tag{22}$$

To preserve quadratic convergence of the global Newton iterations, the consistent tangent modulus is

$$C^{ep} = E\left(1 - \frac{E}{E + H^{\text{iso}} + H^{\text{kin}}}\right) = \frac{E(H^{\text{iso}} + H^{\text{kin}})}{E + H^{\text{iso}} + H^{\text{kin}}}. \tag{23}$$

The shortened return-mapping procedure is summarized in Algorithm 1.

Algorithm 1 Return mapping for 1D combined hardening plasticity.

Input: $\varepsilon_{n+1}, \varepsilon^p_n, \bar{\varepsilon}^p_n, \alpha_n;\ E, \sigma_{y0}, \sigma_u, A, C, \gamma$
$\sigma^{\text{trial}} \leftarrow E(\varepsilon_{n+1} - \varepsilon^p_n)$
$\eta^{\text{trial}} \leftarrow \sigma^{\text{trial}} - \alpha_n,\ s \leftarrow \operatorname{sgn}(\eta^{\text{trial}})$
$\sigma_{y,n} \leftarrow \sigma_{y0} + (\sigma_u - \sigma_{y0})[1 - \exp(-A\bar{\varepsilon}^p_n)]$
$f^{\text{trial}} \leftarrow |\eta^{\text{trial}}| - \sigma_{y,n}$
if $f^{\text{trial}} \le \text{TOL}\,\sigma_{y,n}$ **then**
    $\sigma_{n+1} \leftarrow \sigma^{\text{trial}},\ \alpha_{n+1} \leftarrow \alpha_n$
    $\bar{\varepsilon}^p_{n+1} \leftarrow \bar{\varepsilon}^p_n,\ \varepsilon^p_{n+1} \leftarrow \varepsilon^p_n,\ C^{ep} \leftarrow E$
else
    $\Delta p \leftarrow 0,\ \Phi \leftarrow f^{\text{trial}}$
    while $|\Phi| > \text{TOL}\,\sigma_y$ **and** $k \le k_{\max}$ **do**
        $H^{\text{iso}} \leftarrow A(\sigma_u - \sigma_{y0})\exp[-A(\bar{\varepsilon}^p_n + \Delta p)]$
        $H^{\text{kin}} \leftarrow (C - \gamma\alpha_n s)/(1 + \gamma\Delta p)^2$
        $\Delta p \leftarrow \Delta p - \Phi/(-E - H^{\text{iso}} - H^{\text{kin}})$
        $\alpha_{n+1} \leftarrow (\alpha_n + Cs\Delta p)/(1 + \gamma\Delta p)$
        $\sigma_{y,n+1} \leftarrow \sigma_{y0} + (\sigma_u - \sigma_{y0})[1 - \exp(-A(\bar{\varepsilon}^p_n + \Delta p))]$
        $\Phi \leftarrow f^{\text{trial}} - E\Delta p - s(\alpha_{n+1} - \alpha_n) - (\sigma_{y,n+1} - \sigma_{y,n})$
        $k \leftarrow k + 1$
    end while
    $\bar{\varepsilon}^p_{n+1} \leftarrow \bar{\varepsilon}^p_n + \Delta p,\ \varepsilon^p_{n+1} \leftarrow \varepsilon^p_n + \Delta p\, s$
    $\sigma_{n+1} \leftarrow \sigma^{\text{trial}} - E\Delta p\, s$
    $C^{ep} \leftarrow E(H^{\text{iso}} + H^{\text{kin}})/(E + H^{\text{iso}} + H^{\text{kin}})$
end if
Output: $\sigma_{n+1}, \alpha_{n+1}, \bar{\varepsilon}^p_{n+1}, \varepsilon^p_{n+1}, C^{ep}$

3 Problem Setup, Dataset Generation, and Machine Learning Framework

3.1 Finite Element Model and Cyclic Loading Protocol

A one-dimensional cyclic plasticity problem was solved in Abaqus to generate the reference dataset. The model used a single T3D2 truss element with length $L = 1000$ mm and cross-sectional area 1.0 mm^2. The left node was fixed in all translational directions, and the right node was subjected to a prescribed axial displacement $u(t)$. A schematic of the computational model is shown in Figure 1. The material response

was computed with a user material subroutine (UMAT) implementing the combined Voce isotropic and Armstrong–Frederick kinematic hardening model described in Section 2. The material parameters were Young's modulus $E = 200{,}000$ MPa, Poisson's ratio $\nu = 0.3$, initial yield stress $\sigma_{y0} = 200$ MPa, ultimate yield stress $\sigma_u = 400$ MPa, isotropic hardening exponent $A = 2$, initial kinematic hardening modulus $C = 1000$ MPa, and recall parameter in nonlinear kinematic hardening $\gamma = 100$. These values are used as material inputs to the UMAT described in Section 2.

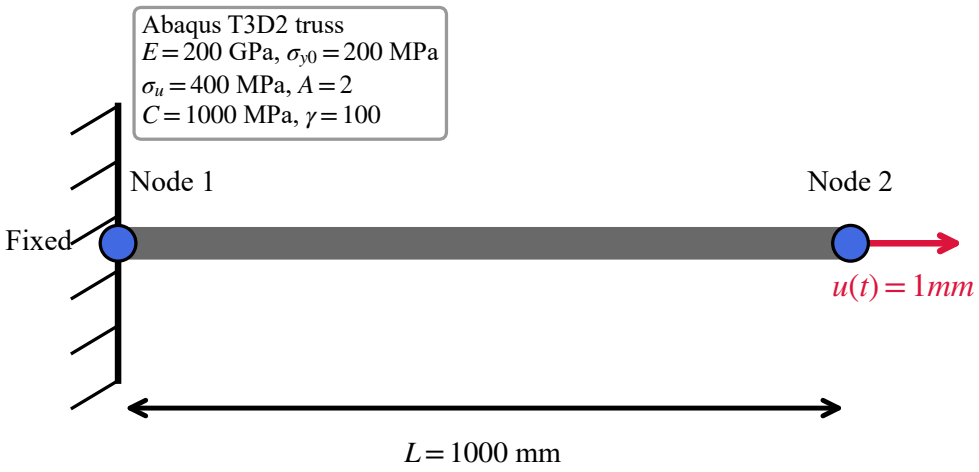


Fig. 1: Schematic of the one-dimensional truss model. The left node is fixed, and the right node is subjected to a prescribed axial displacement $u(t)$. The element is a single Abaqus T3D2 truss with length $L = 1000$ mm and cross-sectional area 1.0 mm^2.

The applied displacement history is non-monotonic and symmetric. It starts at $(0, 0)$, then follows the reversal points $(1, 1)$, $(2, -2)$, $(3, 3)$, $(4, -4)$, …, $(20, -20)$. Here the first value is the sequence step and the second is the applied displacement in mm. The amplitude increases progressively, producing repeated elastic–plastic transitions.

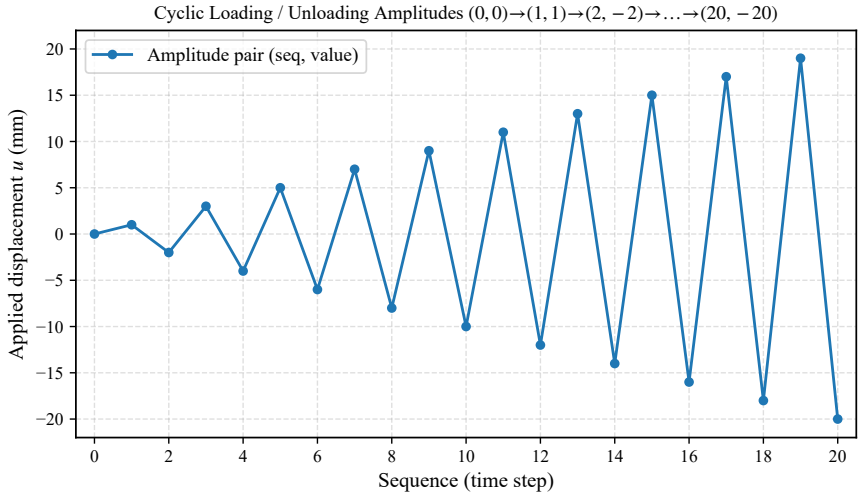


Fig. 2: Applied cyclic displacement protocol. The sequence lists (time step, displacement amplitude in mm): $(0, 0)$, $(1, 1)$, $(2, -2)$, …, $(20, -20)$. The amplitude increases progressively and includes multiple loading and unloading reversals.

3.2 Dataset Preprocessing and Sequential Feature Engineering

The input features are the current total strain ε_{n+1} and the strain increment $\Delta\varepsilon_{n+1}$. The target is the current stress σ_{n+1}. Thus,

$$X = [\varepsilon_{n+1}, \Delta\varepsilon_{n+1}], \qquad y = \sigma_{n+1}.$$

No internal state variables, such as the backstress α or the equivalent plastic strain $\bar{\varepsilon}^p$, are used as inputs.

The data are split in time order: the first 70% for training, the next 15% for validation, and the final 15% for testing. This gives 14,022 training samples, 3,005 validation samples, and 3,005 test samples. The chronological split prevents future states from entering training. A `StandardScaler` is fitted only on the training set. The same scaler is then applied to the validation and test sets. The target stress is also scaled for training and validation; the test target remains in MPa for final error metrics.

For the recurrent models, a sliding window transforms the tabular data into sequences. A window of size W contains W consecutive feature vectors. The resulting tensor has shape (N, W, d), where $d = 2$. The target for each sequence is the stress at the last time step in the window. The optimized recurrent configurations use $W = 15$. A window of $W = 25$ was also tested. Windows are generated independently within the training, validation, and test sets to avoid crossing split boundaries.

3.3 Deep Learning Architectures

3.3.1 Multi-Layer Perceptron (MLP)

The MLP is a memoryless baseline. It maps $[\varepsilon_{n+1}, \Delta\varepsilon_{n+1}]$ directly to σ_{n+1}. The selected configuration, `MLP_3L_64N_relu`, has three dense hidden layers of sizes $[64, 64, 32]$ with ReLU activation, followed by a linear output unit. The input dimension is 2. This model does not receive any history information beyond the current increment.

3.3.2 Sequential Windowing Formulation

For recurrent models, the input is a sequence of length W. The sequence contains the strain and strain increment for W consecutive increments. The hidden states are expected to carry the history needed for path-dependent plasticity. The optimized models use $W = 15$, which corresponds to a short strain history around the current increment.

3.3.3 Recurrent Architectures: LSTM and GRU

The LSTM model `LSTM_2L_64N_W15` has two stacked LSTM layers with sizes $[64, 32]$. The first layer returns the full sequence; the second returns the final hidden state. The GRU model `GRU_1L_64N_W15` has one GRU layer with 64 units. Both models then use a dense layer of 32 units with tanh activation, followed by a linear output unit for σ_{n+1}.

The LSTM gates follow the standard form:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \tag{24}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \tag{25}$$

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C), \tag{26}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \tag{27}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{28}$$

$$h_t = o_t \odot \tanh(C_t). \tag{29}$$

Here σ is the sigmoid function, tanh is the hyperbolic tangent, and $\odot$ is element-wise multiplication. The GRU follows the Keras implementation:

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z), \tag{30}$$

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r), \tag{31}$$

$$\tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, x_t] + b_h), \tag{32}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t. \tag{33}$$

In this notation z_t weights the previous hidden state, and $(1 - z_t)$ weights the candidate state, matching the Keras GRU update.

3.3.4 Training Strategy

All models were trained with the Adam optimizer and mean squared error loss. The initial learning rate was 10^{-3}. The MLP used a batch size of 128 and up to 200 epochs. The LSTM and GRU models used a batch size of 128 and up to 150 epochs. Early stopping monitored validation loss with patience 20 and restored the best weights. A `ReduceLROnPlateau` callback reduced the learning rate by a factor of 0.5 after 10 epochs without improvement, with a minimum learning rate of 10^{-6}. The validation set was used only for these callbacks. The test set was held out until final evaluation.

4 Results and Performance Analysis

4.1 Hyperparameter Optimization and Architecture Selection

Hyperparameters were selected using the chronological training and validation sets only. The test set was held out until final evaluation. All models used the Adam optimizer with an initial learning rate of 0.001 and mean squared error (MSE) loss. The MLP was trained for up to 200 epochs; the LSTM and GRU models were trained for up to 150 epochs. A batch size of 128 was used. Early stopping monitored validation loss with patience 20 and restored the best weights. ReduceLROnPlateau monitored validation loss with factor 0.5, patience 10, and minimum learning rate 10^{-6}. Validation MSE values in Tables 1–3 are computed on the standardized target, so they are used for model selection rather than as physical error measures.

4.1.1 Multilayer Perceptron

The MLP grid covered two- and three-layer networks, widths of 64 and 128, and tanh and ReLU activations. Table 1 lists the results. ReLU gave a lower validation MSE than tanh for every matched architecture. The lowest validation MSE was obtained by `MLP_3L_64N_relu`, with hidden layers $[64, 64, 32]$ and ReLU activation. Its validation MSE was 0.003960, and its training time was 51.114 s. This configuration was selected for the final MLP evaluation.

Table 1: MLP hyperparameter search. Validation MSE is computed on the standardized target.

Configuration	Hidden layers	Activation	Validation MSE (scaled)	Training time (s)
`MLP_2L_64N_tanh`	[64, 64]	`tanh`	0.022732	14.455
`MLP_2L_64N_relu`	[64, 64]	`relu`	0.005030	75.228
`MLP_2L_128N_tanh`	[128, 128]	`tanh`	0.025086	13.835
`MLP_2L_128N_relu`	[128, 128]	`relu`	0.006321	54.002
`MLP_3L_64N_tanh`	[64, 64, 32]	`tanh`	0.008222	85.200
`MLP_3L_64N_relu`	[64, 64, 32]	`relu`	0.003960	51.114

4.1.2 Long Short-Term Memory

The LSTM grid varied the look-back window $W \in \{15, 25\}$ and the recurrent depth: one layer [64] or two layers [64, 32]. The recurrent activation was fixed to tanh. Each model also used a Dense layer with 32 units and tanh activation, followed by a linear output layer. Table 2 gives the validation results. The best configuration was `LSTM_2L_64N_W15`, with $W = 15$, recurrent layers [64, 32], validation MSE 0.000241, and training time 129.771 s. For both depths, $W = 15$ had a lower validation MSE than $W = 25$. At $W = 15$ and $W = 25$, the two-layer LSTM had a lower validation MSE than the one-layer LSTM. The selected model was therefore the two-layer LSTM with $W = 15$.

Table 2: LSTM hyperparameter search. Validation MSE is computed on the standardized target. Recurrent activation was tanh for all configurations.

Configuration	Window	Recurrent layers	Validation MSE (scaled)	Training time (s)
`LSTM_1L_64N_W15`	15	[64]	0.000356	110.901
`LSTM_2L_64N_W15`	15	[64, 32]	0.000241	129.771
`LSTM_1L_64N_W25`	25	[64]	0.000664	158.437
`LSTM_2L_64N_W25`	25	[64, 32]	0.000442	183.769

4.1.3 Gated Recurrent Unit

The GRU search used the same window sizes and recurrent depths as the LSTM search. Recurrent activation was again tanh, followed by a Dense layer with 32 units and tanh activation and a linear output layer. Table 3 reports the results. The best configuration was `GRU_1L_64N_W15`, with $W = 15$, one recurrent layer [64], validation MSE 0.000477, and training time 98.137 s. For the one-layer GRU, $W = 15$ was better than $W = 25$. For the two-layer GRU, $W = 25$ was slightly better than $W = 15$ (0.000501 versus 0.000624), but this two-layer model still had a higher validation MSE than the best one-layer model. The selected GRU was therefore the one-layer model with $W = 15$.

Table 3: GRU hyperparameter search. Validation MSE is computed on the standardized target. Recurrent activation was tanh for all configurations.

Configuration	Window	Recurrent layers	Validation MSE (scaled)	Training time (s)
`GRU_1L_64N_W15`	15	[64]	0.000477	98.137
`GRU_2L_64N_W15`	15	[64, 32]	0.000624	199.204
`GRU_1L_64N_W25`	25	[64]	0.003368	56.429
`GRU_2L_64N_W25`	25	[64, 32]	0.000501	331.097

4.1.4 Selection Summary

Among the selected configurations, the LSTM had the lowest validation MSE (0.000241), followed by the GRU (0.000477) and the MLP (0.003960). The selected MLP trained fastest (51.114 s), while the selected GRU trained faster than the selected LSTM (98.137 s versus 129.771 s). These validation results determined the architectures used for final test-set evaluation.

4.2 Individual Model Evaluation on Cyclic Loading

The selected models were evaluated on the held-out test set. The loading history is non-monotonic and multi-amplitude, so it exercises both kinematic translation and isotropic expansion of the yield surface.

4.2.1 Multi-Layer Perceptron (MLP) Response

Figure 3 shows the MLP prediction. The model is memoryless and does not receive backstress or equivalent plastic strain as input. Because the stress response is history dependent, identical current strain and strain-increment pairs can correspond to different stress states. Figure 3(a) shows the prediction over the entire dataset; the deviation along the cyclic branches is visible in the training, validation and test regions. The time-history plot in Figure 3(b) shows phase lag and clipping at sharp unloading events. Figure 3(c) shows that the MLP does not reproduce the cyclic stress–strain loop shape or reverse plastic yielding. The scatter plot in Figure 3(d) gives a high global $R^2 = 0.9790$ because stresses span about ± 400 MPa, but local errors are large. The MLP gives RMSE = 55.818 MPa, MAE = 49.339 MPa, and reversal MAPE = 62.873%.

4.2.2 Long Short-Term Memory (LSTM) Response

Figure 4 shows the two-layer `LSTM_2L_64N_W15` prediction. The LSTM tracks the cyclic response closely, with RMSE = 14.339 MPa, MAE = 9.735 MPa, and $R^2 = 0.9986$. Figure 4(a) shows the prediction over the entire dataset, confirming that the accuracy is maintained across the training, validation and test regions. The transient stress tracking in Figure 4(b) follows the reference without visible phase shift in the plotted interval. Figure 4(c) shows that the model follows the nonlinear kinematic hardening curvature, the transition from elastic stiffness to plastic tangent modulus, and the cyclic loop shape over the tested strain ranges. The cell state C_t can be interpreted as an implicit representation of the backstress, while the input and forget gates control how much history is retained or discarded, consistent with plastic memory and dynamic recovery. The correlation scatter in Figure 4(d) is tightly clustered around the ideal-fit line.

4.2.3 Gated Recurrent Unit (GRU) Response

Figure 5 shows the single-layer `GRU_1L_64N_W15` prediction. The GRU gives similar global precision to the LSTM but lower local error. It gives RMSE = 14.455 MPa, MAE = 6.200 MPa (36.3% lower than the LSTM), $R^2 = 0.9986$, and a training time of 98.14 s (24.4% faster than the LSTM). Figure 5(a) shows the prediction over the entire dataset, and the accuracy is maintained in all three regions. The transient trajectory in Figure 5(b) closely follows the reference over the cycles. Figure 5(c) shows clear elastic unloading slopes and smooth reverse yielding. The reset gate can discard obsolete directional history at strain reversal, and the update gate interpolates between elastic and plastic response. The scatter plot in Figure 5(d) confirms the low local error.

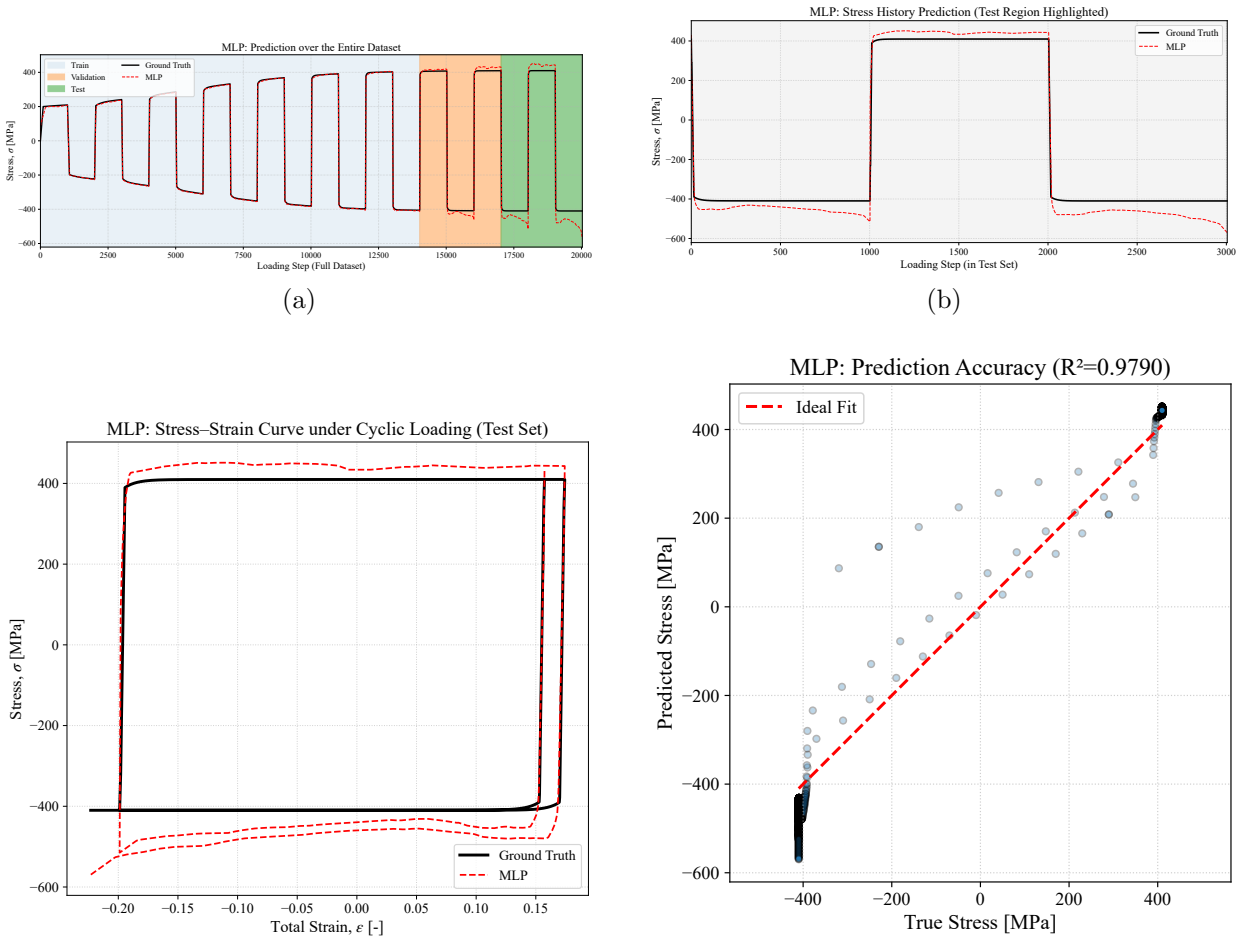


Fig. 3: Predictive performance of the optimized `MLP_3L_64N_relu` surrogate under cyclic loading. (a) Prediction over the entire dataset, with chronological train, validation and test regions shaded. (b) Transient stress time-history on the test subset, compared with the FE/UMAT reference. (c) Cyclic stress–strain response on the test subset, showing path-dependent distortions. (d) Correlation scatter between reference and predicted stress values on the test subset.

4.3 Comparative Benchmark and Hysteresis Tracking

Table 4 gives the final test-set metrics for the optimized MLP, LSTM, and GRU models. The MLP uses the current total strain and strain increment only, while the recurrent models use a sequence window. The best MLP is a three-layer network with 64–64–32 units and ReLU activations. The best LSTM is a two-layer model with 64 and 32 units and a window of 15 steps. The best GRU is a single-layer model with 64 units and the same window. All models were trained with the same early-stopping and learning-rate-reduction callbacks.

The MLP obtains a high coefficient of determination ($R^2 = 0.9790$), but its stress errors are much larger than those of the recurrent models: RMSE = 55.818 MPa and MAE = 49.339 MPa. Its reversal MAPE is 62.873%. The high R^2 therefore reflects the overall stress range rather than accurate path tracking. The MLP poorly resolves the hysteresis loop and the sharp changes that occur after load reversals. This is consistent with its lack of explicit sequence memory. The limitation is visible in Fig. 6(a), where the MLP prediction deviates from the UMAT reference along the cyclic branches over the entire dataset. Figures 6(b) and 6(c) show that the LSTM and GRU follow the reference cyclic response much more closely.

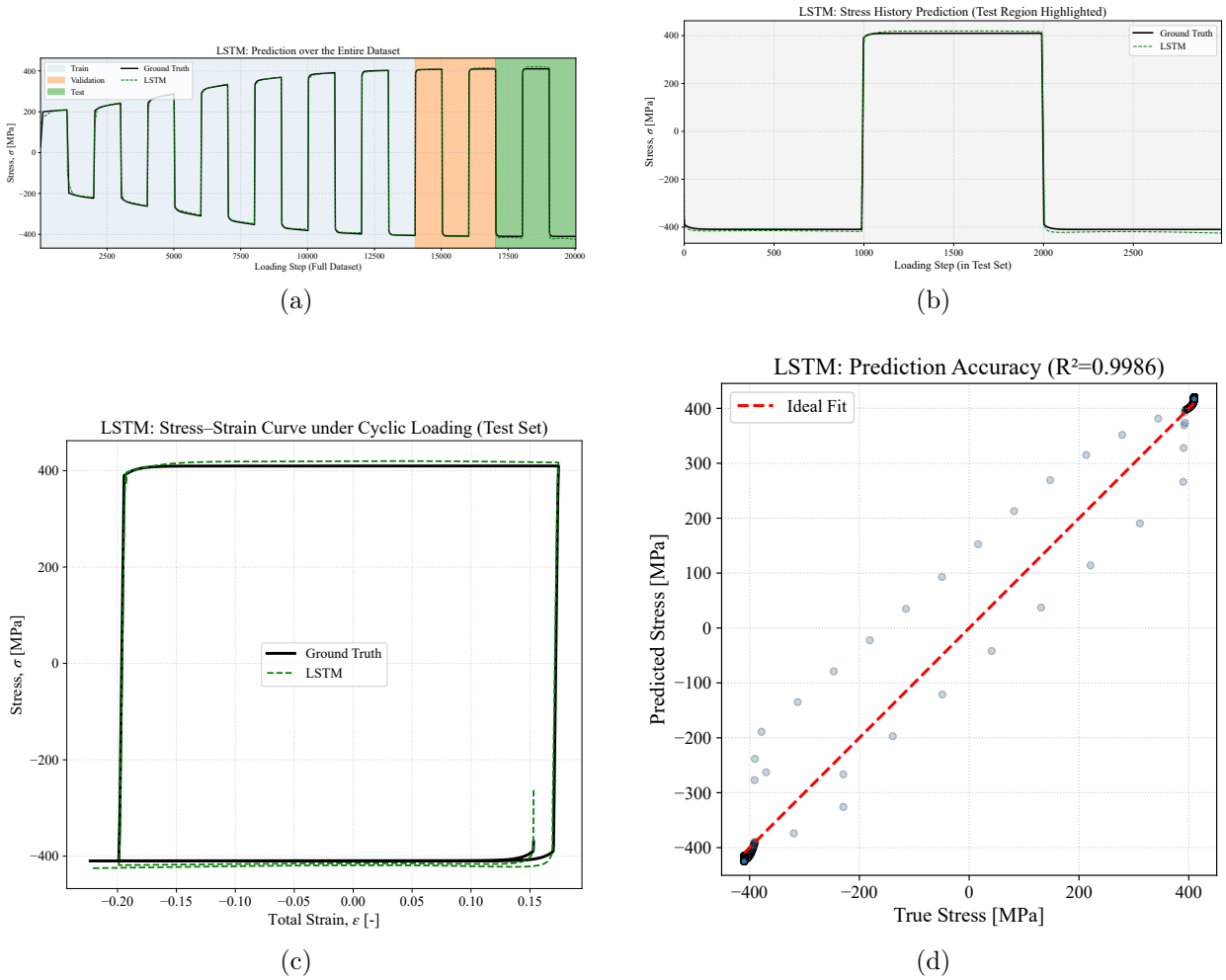


Fig. 4: Predictive performance of the optimized `LSTM_2L_64N_W15` surrogate under cyclic loading. (a) Prediction over the entire dataset, with chronological train, validation and test regions shaded. (b) Transient stress time-history on the test subset, compared with the FE/UMAT reference. (c) Cyclic stress–strain response on the test subset, following the nonlinear kinematic hardening curvature and the elastic-to-plastic transition. (d) Correlation scatter between reference and predicted stress values on the test subset.

The LSTM and GRU reduce the MLP errors by a large margin. The LSTM gives RMSE = 14.339 MPa, MAE = 9.735 MPa, $R^2 = 0.9986$, and reversal MAPE = 15.819%. The GRU gives RMSE = 14.455 MPa, MAE = 6.200 MPa, $R^2 = 0.9986$, and reversal MAPE = 9.901%. The LSTM has the slightly lower RMSE and the slightly higher R^2, but the differences are small. The GRU has the lower MAE and reversal MAPE. The GRU also trains faster: 98.137 s versus 129.771 s for the LSTM, a reduction of about 24% under the same setup. Thus, neither recurrent model dominates on all metrics. The LSTM is marginally better in global RMSE and R^2, while the GRU is preferable when reversal accuracy and training cost are the main criteria.

The reversal MAPE is also relevant to the Bauschinger effect. Reverse loading produces a directional change in the stress–strain response, and the GRU tracks this change more closely than the LSTM and much more closely than the MLP. The GRU reversal MAPE of 9.901% supports this observation, but it is a data-driven error measure, not a direct constitutive validation. The claim that the GRU captures the Bauschinger effect should therefore be limited to improved tracking of reverse loading in the test data.

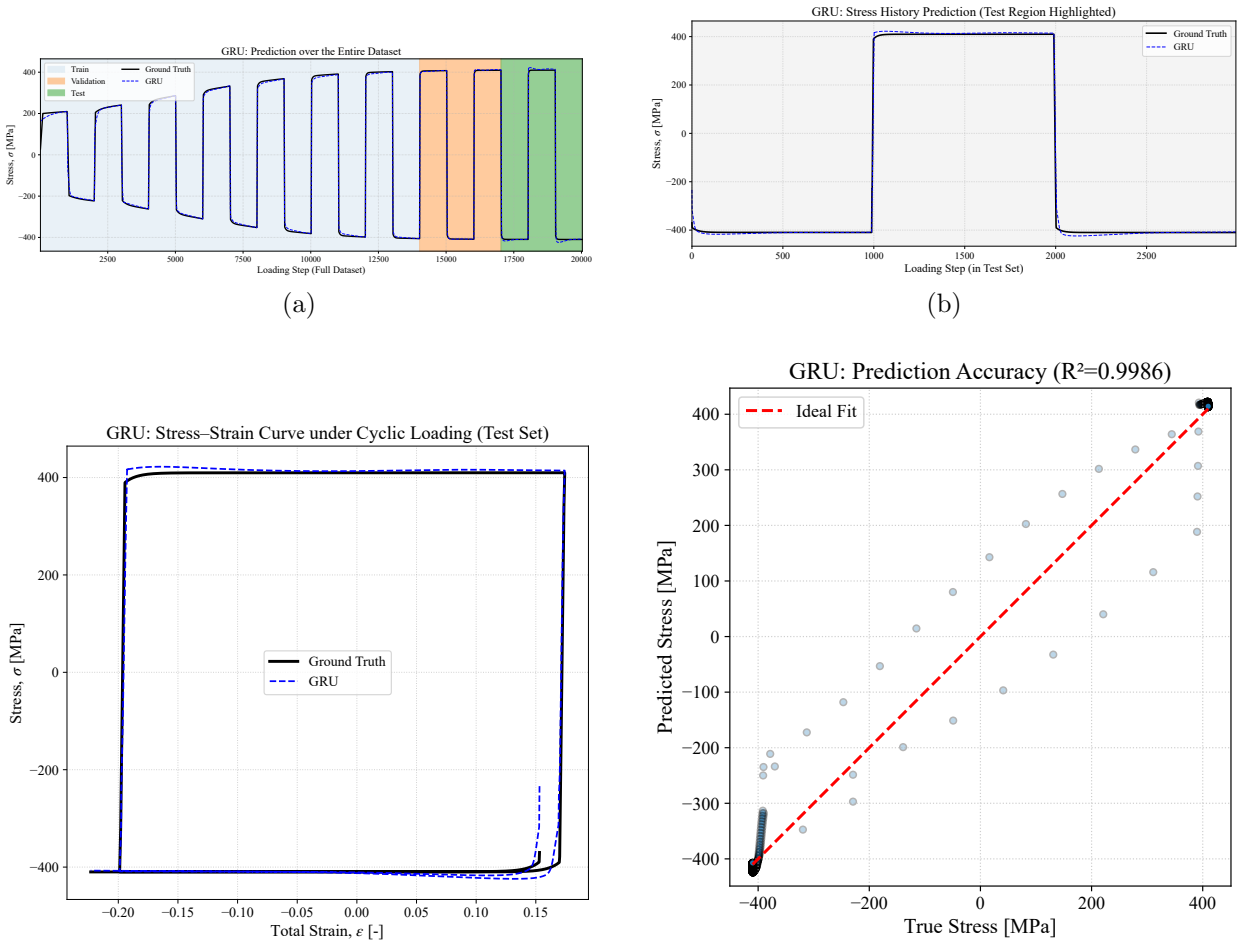


Fig. 5: Predictive performance of the optimized `GRU_1L_64N_W15` surrogate under cyclic loading. (a) Prediction over the entire dataset, with chronological train, validation and test regions shaded. (b) Transient stress time-history on the test subset, compared with the FE/UMAT reference. (c) Cyclic stress–strain response on the test subset, with clear elastic unloading slopes and smooth reverse yielding. (d) Correlation scatter between reference and predicted stress values on the test subset.

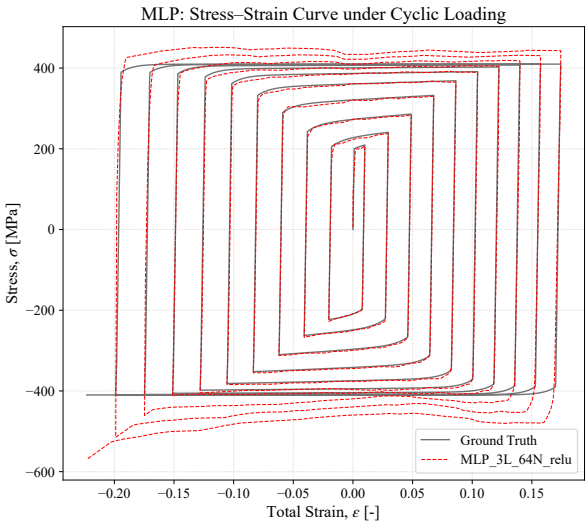
Table 4: Final test-set performance of the optimized MLP, LSTM, and GRU models. Training times include early stopping and learning-rate reduction.

Model	Best configuration	RMSE (MPa)	MAE (MPa)	R^2	Reversal MAPE (%)	Training time (s)
MLP	64–64–32, ReLU	55.818	49.339	0.9790	62.873	51.114
LSTM	64–32, window 15	14.339	9.735	0.9986	15.819	129.771
GRU	64, window 15	14.455	6.200	0.9986	9.901	98.137

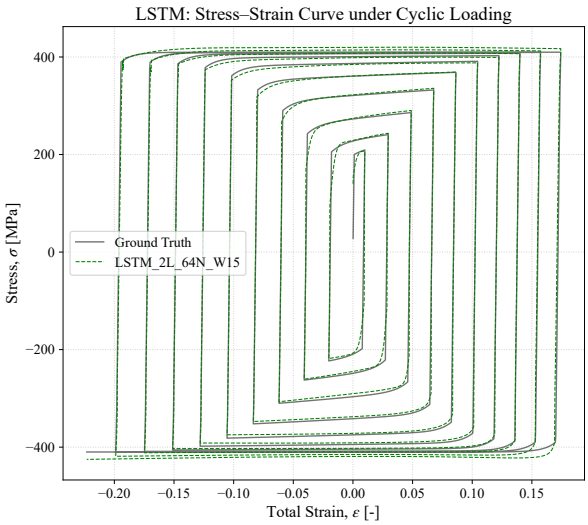
For the sequence models, the first $w-1$ test samples are used only as context. With $w = 15$, the metrics for the LSTM and GRU are computed on 2991 test points, whereas the MLP metrics use 3005 points. This difference does not change the ranking, but it should be stated when the results are compared.

4.4 Accuracy at Critical Loading Reversal Points

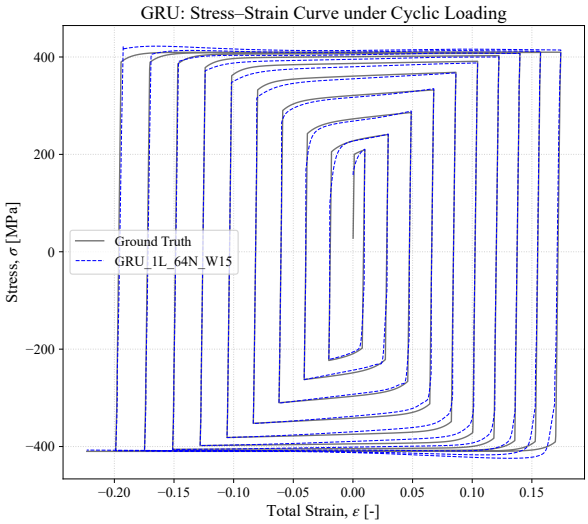
Load reversal points are the most demanding part of a cyclic loading history. At these points, the material switches from plastic flow to elastic unloading, and the stress–strain response follows the elastic slope $d\sigma = E\, d\varepsilon$. If a surrogate model does not resolve this transition, the error at each reversal accumulates over



(a)



(b)



(c)

Fig. 6: Comparison of the three optimized surrogates over the entire dataset under cyclic loading. (a) Optimized MLP (`MLP_3L_64N_relu`). (b) Optimized LSTM (`LSTM_2L_64N_W15`). (c) Optimized GRU (`GRU_1L_64N_W15`). The MLP prediction deviates noticeably from the reference along the cyclic branches, whereas the recurrent models follow the reference response more closely.

subsequent cycles. Reversal accuracy is therefore a useful check on whether the model has learned path dependence rather than just fitting the overall stress range.

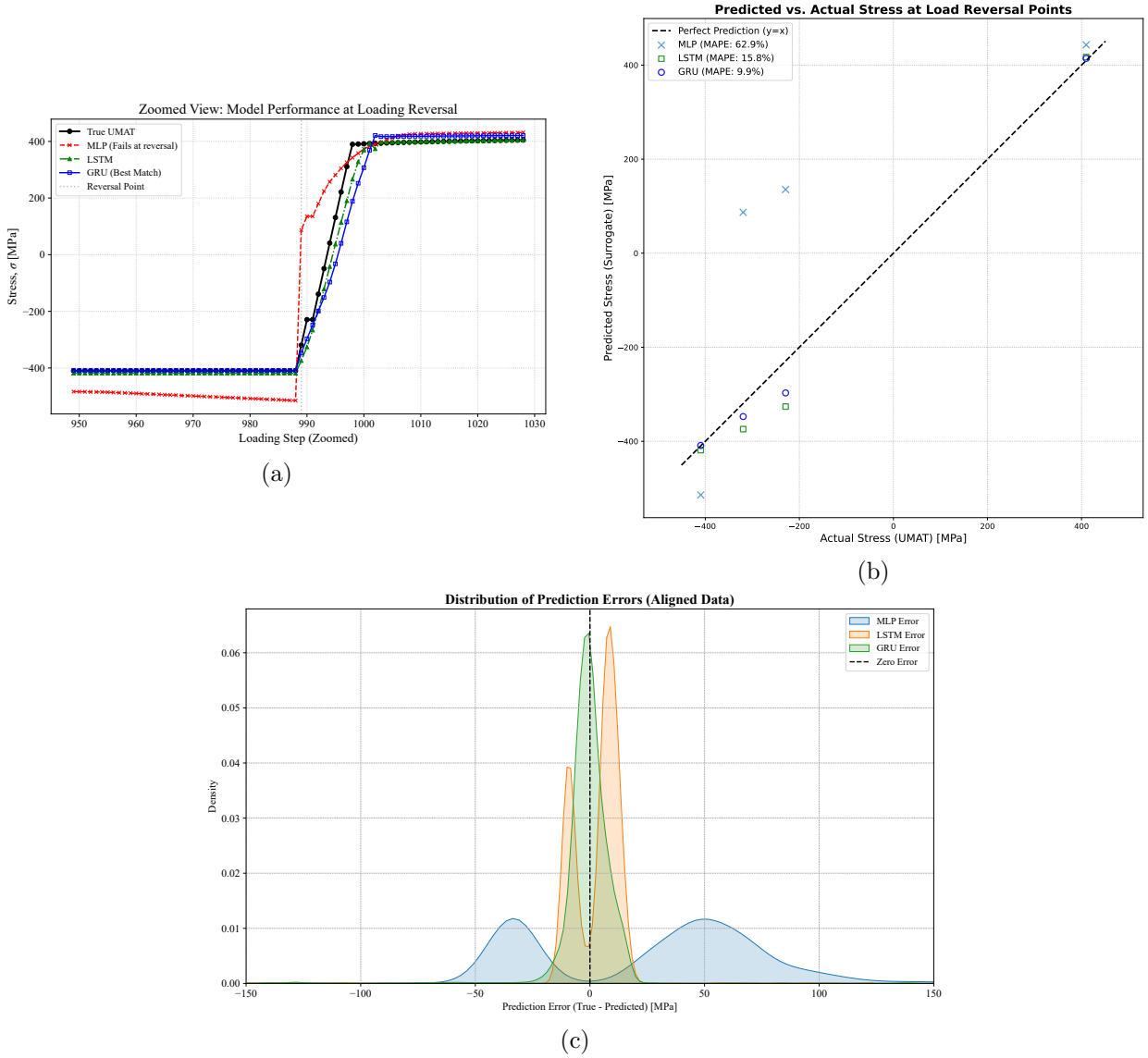


Fig. 7: Accuracy at load reversal points: (a) magnified hysteresis vertices showing the elastic unloading slope; (b) predicted vs. true stress at reversal increments; (c) distribution of residuals across the test set.

Figure 7 compares the three models at reversal points. The magnified vertices in Figure 7(a) show that the MLP rounds off the corners at reversal rather than reproducing the sharp transition seen in the reference solution. The MLP sees only the current strain and strain increment, so it cannot tell whether the load is increasing or decreasing from one increment alone. The LSTM and GRU produce sharper vertices that follow the elastic unloading slope more closely, although small deviations remain.

The reversal MAPE values in Table 4 confirm this ranking. The GRU gives the lowest reversal error (9.901%), followed by the LSTM (15.819%), and the MLP (62.873%). In Figure 7(b), the LSTM and GRU points at reversal increments lie close to the $y = x$ line. The MLP points are more scattered and tend to sit below the line, which means the model underestimates the peak stress at reversals.

Figure 7(c) shows the distribution of residuals over the test set. The GRU residuals form a narrow peak near zero. The MLP residuals are wider, and the LSTM falls between the two. These distributions agree with the error metrics in Table 4: the recurrent models track reversal loading more consistently than the MLP, and the GRU does so with the smallest spread. Note that the LSTM and GRU metrics are computed on 2991 test points (after windowing), while the MLP metrics use 3005 points, as stated in Section 4.3.

5 Discussion and Mechanical Implications

5.1 Interpretation of the Surrogate Predictions

The results show a clear difference between the feed-forward and recurrent surrogates. The optimized MLP reaches $R^2 = 0.978972$, but its RMSE is 55.818 MPa, its MAE is 49.339 MPa, and its reversal MAPE is 62.873%. These values indicate that the MLP captures the general stress level but not the unloading and reloading branches accurately. In cyclic plasticity, this is a serious limitation because reversal points control plastic strain accumulation and hysteresis shape.

The LSTM and GRU reduce the RMSE to 14.339 MPa and 14.455 MPa, respectively. Their R^2 values are 0.998615 and 0.998592. Both models therefore reproduce the hysteresis response much better than the MLP. The GRU gives the lower MAE, 6.200 MPa versus 9.735 MPa for the LSTM, and the lower reversal MAPE, 9.901% versus 15.819%. The LSTM has a marginal advantage in RMSE and R^2. Under the tested metrics, the two recurrent models are close, but the GRU is more consistent at reversals.

The optimal window size was 15 for both LSTM and GRU. Increasing the window to 25 raised the validation MSE in most tested recurrent configurations; the only exception was the two-layer GRU, where $W = 25$ gave a slightly lower validation MSE than $W = 15$ (0.000501 versus 0.000624). That two-layer variant, however, still failed to beat the selected single-layer GRU at $W = 15$. For this dataset, a short strain history is therefore sufficient to capture the loading path, and a longer window adds parameters without improving generalization.

5.2 Internal State Variables and Thermodynamic Consistency

The surrogate is trained only on the current total strain and the strain increment, $[\epsilon_t, \Delta\epsilon_t]$. It does not receive the backstress $\boldsymbol{\alpha}$, the plastic strain $\boldsymbol{\varepsilon}^p$, the accumulated plastic strain p, or the isotropic hardening variable R. The hidden state of the LSTM or GRU is a learned memory of the input sequence. It is not a physical internal variable and has no direct relation to the Armstrong–Frederick evolution law used in the UMAT.

Because of this input structure, the network does not evaluate the yield function, does not enforce the consistency condition, and does not impose the Clausius–Duhem inequality. The UMAT does all of these during the return mapping. The surrogate predicts stress directly, so it can violate the yield condition and thermodynamic restrictions, especially outside the training domain. Good agreement on the test set should therefore be interpreted as interpolation within the sampled loading range. Extrapolation to other histories, materials, or temperatures is not justified by the present results.

5.3 Computational Trade-offs

The training times are 51.1 s for the MLP, 129.8 s for the LSTM, and 98.1 s for the GRU. These are single-run wall-clock times, not a controlled benchmark. The MLP is the fastest, but its reversal error is too large for cyclic loading. The GRU is about 24% faster than the LSTM and gives better MAE and reversal MAPE. The LSTM gives slightly lower RMSE and slightly higher R^2. The choice therefore depends on the target quantity. If reversal accuracy and training cost matter, the GRU is the better balance. If the smallest global RMSE is the priority, the LSTM is marginally better.

5.4 Limitations and Future Perspectives

The main limitation is the training domain. The data come from one constitutive model and one loading protocol. Multiaxial loading, non-proportional paths, ratcheting, temperature effects, and strain-rate effects are not represented. The network also does not output a consistent tangent stiffness, so it cannot replace the UMAT inside an implicit finite element loop without a numerical approximation.

The reversal errors are still non-negligible. The GRU reaches a reversal MAPE of 9.901%, and the LSTM reaches 15.819%. These values are much better than the MLP result, but they are not zero. For path-dependent plasticity, this residual error can accumulate over many cycles.

Future work should include internal variables such as backstress, plastic strain, accumulated plastic strain, and isotropic hardening as additional inputs. A physics-informed loss could penalize yield-function violation and enforce the consistency condition. A hybrid formulation could use the surrogate only where it is reliable and fall back to the UMAT near reversals or outside the training range. Validation on multiaxial and thermomechanical loading is also needed before the model can be used in general finite element analysis.

6 Conclusion

This study compared feedforward and recurrent neural network surrogates for one-dimensional cyclic plasticity with combined Voce isotropic and Armstrong–Frederick kinematic hardening. The models were trained on strain sequences only, with no internal state variables as inputs. The main results are as follows.

1. **The MLP did not capture the hysteresis loop.** The memoryless MLP gave RMSE = 55.818 MPa, MAE = 49.339 MPa, and reversal MAPE = 62.873%. Its $R^2 = 0.9790$ reflects the wide stress range rather than accurate path tracking. The MLP does not reproduce the Bauschinger effect or the sharp stress changes at reversal.
2. **Recurrent models learned the path-dependent response from strain alone.** The LSTM and GRU reduced RMSE to about 14.3–14.5 MPa and reproduced the hysteresis loops, the Bauschinger effect, and the reversal vertices. The validation MSE dropped from 3.960×10^{-3} (MLP) to 2.410×10^{-4} (two-layer LSTM). The optimal window size was 15 for both architectures. The recurrent hidden states act as a learned memory of the strain sequence; they are not explicit representations of backstress or accumulated plastic strain.
3. **The LSTM and GRU traded off differently.** The LSTM gave the lowest RMSE (14.339 MPa) and a slightly higher R^2 (0.998615 vs. 0.998592). The GRU gave lower MAE (6.200 MPa vs. 9.735 MPa), lower reversal MAPE (9.901% vs. 15.819%), and trained 24.4% faster (98.14 s vs. 129.77 s).

4. **The GRU is the better default for this problem.** The single-layer GRU with $W = 15$ gave the best reversal accuracy and the lowest training cost. The LSTM is preferable only when global RMSE is the main criterion.

These findings apply to the tested loading protocol, material parameters, and network configurations. The models do not output a consistent tangent stiffness, and reversal errors remain nonzero (GRU: 9.901%, LSTM: 15.819%). Future work should test multiaxial and non-proportional loading, add physical constraints to the loss function, and evaluate whether the surrogate can replace a UMAT inside an implicit finite element solver.

Supplementary information. Not applicable.

Acknowledgements. Not applicable.

Declarations

- **Funding:** The authors received no financial support for the research, authorship, and publication of this article.

- **Conflict of interest/Competing interests:** The authors declare no competing interests.
- **Ethics approval and consent to participate:** Not applicable.
- **Consent for publication:** Not applicable.
- **Data availability:** The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.
- **Materials availability:** Not applicable.
- **Code availability:** The custom Abaqus UMAT and Python notebooks used in this study are available from the corresponding author on reasonable request.
- **Author contribution:** All authors contributed to the study conception and design. Mohammad Ashraf-Ganjouei implemented the UMAT, generated the reference data, and developed the machine learning models. Seyed Hadi Ghaderi and S. Mojtaba Varedi-Koulaei supervised the work and reviewed the manuscript. All authors read and approved the final manuscript.

References

[1] Lemaitre, J., Chaboche, J.-L.: Mechanics of solid materials. Cambridge University Press (1990) https://doi.org/10.1017/CBO9781139167970

[2] Simo, J.C., Hughes, T.J.R.: Computational Inelasticity. Interdisciplinary Applied Mathematics, vol. 7, pp. 1–70. Springer, New York (1998). https://doi.org/10.1007/b98904

[3] Chaboche, J.-L.: Time-independent constitutive theories for cyclic plasticity. International Journal of Plasticity **2**(2), 149–188 (1986) https://doi.org/10.1016/0749-6419(86)90010-0

[4] Chaboche, J.-L.: A review of some plasticity and viscoplasticity constitutive theories. International Journal of Plasticity **24**(10), 1642–1693 (2008) https://doi.org/10.1016/j.ijplas.2008.03.009

[5] Prager, W.: The theory of plasticity: a survey of recent achievements. Proceedings of the Institution of Mechanical Engineers **169**(1), 41–57 (1955) https://doi.org/10.1243/PIME_PROC_1955_169_015_02

[6] Ziegler, H.: A modification of Prager's hardening rule. Quarterly of Applied Mathematics **17**(1), 55–65 (1959) https://doi.org/10.1090/qam/104405

[7] Armstrong, P.J., Frederick, C.O.: A mathematical representation of the multiaxial Bauschinger effect. Technical Report RD/B/N 731, Central Electricity Generating Board (1966)

[8] Chaboche, J.-L.: Constitutive equations for cyclic plasticity and cyclic viscoplasticity. International Journal of Plasticity **5**(3), 247–302 (1989) https://doi.org/10.1016/0749-6419(89)90015-6

[9] Chaboche, J.-L.: On some modifications of kinematic hardening to improve the description of ratchetting effects. International Journal of Plasticity **7**(7), 661–678 (1991) https://doi.org/10.1016/0749-6419(91)90050-9

[10] Voce, E.: The relationship between stress and strain for homogeneous deformation. Journal of the Institute of Metals **74**, 537–562 (1948)

[11] Krieg, R.D., Krieg, D.B.: Accuracies of numerical solution methods for the elastic-perfectly plastic model. Journal of Pressure Vessel Technology **99**(4), 510–515 (1977) https://doi.org/10.1115/1.3454568

[12] Simo, J.C., Taylor, R.L.: Consistent tangent operators for rate-independent elastoplasticity. Computer Methods in Applied Mechanics and Engineering **48**(1), 101–118 (1985) https://doi.org/10.1016/0045-7825(85)90070-2

[13] Mahbadi, H., Eslami, M.R.: Cyclic loading of beams based on the Prager and Chaboche-type kinematic hardening rules. International Journal of Mechanical Sciences **47**(9), 1440–1456 (2005) https://doi.org/10.1016/j.ijmecsci.2005.04.004

[14] Dunne, F., Petrinic, N.: Introduction to Computational Plasticity. Oxford University Press, Oxford (2007). https://doi.org/10.1093/acprof:oso/9780198568261.001.0001

[15] Ghaboussi, J., Garrett, J.H., Wu, X.: Knowledge-based modeling of material behavior with neural networks. Journal of Engineering Mechanics **117**(1), 132–153 (1991) https://doi.org/10.1061/(ASCE)0733-9399(1991)117:1(132)

[16] Furukawa, T., Yagawa, G.: Implicit constitutive modelling for viscoplasticity using neural networks. International Journal for Numerical Methods in Engineering **43**(2), 195–219 (1998) https://doi.org/10.1002/(SICI)1097-0207(19980930)43:2⟨195::AID-NME419⟩3.0.CO;2-D

[17] Bock, F.E., Aydin, R.C., Cyron, C.J., Huber, N., Kalidindi, S.R., Klusemann, B.: A review of the application of machine learning and data mining approaches in continuum materials mechanics. Frontiers in Materials **Volume 6 - 2019** (2019) https://doi.org/10.3389/fmats.2019.00110

[18] Huang, D., Fuhg, J.N., Weißenfels, C., Wriggers, P.: A machine learning based plasticity model using

proper orthogonal decomposition. Computer Methods in Applied Mechanics and Engineering **365**, 113008 (2020) https://doi.org/10.1016/j.cma.2020.113008

[19] Stoffel, M., Gulakala, R., Bamer, F., Markert, B.: Artificial neural networks in structural dynamics: A new modular radial basis function approach vs. convolutional and feedforward topologies. Computer Methods in Applied Mechanics and Engineering **364**, 112989 (2020) https://doi.org/10.1016/j.cma.2020.112989

[20] Zhang, A., Mohr, D.: Using neural networks to represent von Mises plasticity with isotropic hardening. International Journal of Plasticity **132**, 102732 (2020) https://doi.org/10.1016/j.ijplas.2020.102732

[21] Teranishi, M.: Neural network constitutive model for uniaxial cyclic plasticity based on return mapping algorithm. Mechanics Research Communications **119**, 103815 (2022) https://doi.org/10.1016/j.mechrescom.2021.103815

[22] Dettmer, W.G., Muttio, E.J., Alhayki, R., Perić, D.: A framework for neural network based constitutive modelling of inelastic materials. Computer Methods in Applied Mechanics and Engineering **420**, 116672 (2024) https://doi.org/10.1016/j.cma.2023.116672

[23] Mozaffar, M., Bostanabad, R., Chen, W., Ehmann, K., Cao, J., Bessa, M.A.: Deep learning predicts path-dependent plasticity. Proceedings of the National Academy of Sciences **116**(52), 26414–26420 (2019) https://doi.org/10.1073/pnas.1911815116

[24] Gorji, M.B., Mozaffar, M., Heidenreich, J.N., Cao, J., Mohr, D.: On the potential of recurrent neural networks for modeling path dependent plasticity. Journal of the Mechanics and Physics of Solids **143**, 103972 (2020) https://doi.org/10.1016/j.jmps.2020.103972

[25] Ghane, E., Fagerström, M., Mirkhalaf, M.: Recurrent neural networks and transfer learning for predicting elasto-plasticity in woven composites. European Journal of Mechanics - A/Solids **107**, 105378 (2024) https://doi.org/10.1016/j.euromechsol.2024.105378

[26] He, Y., Semnani, S.J.: Incremental neural controlled differential equations for modeling of path-dependent material behavior. Computer Methods in Applied Mechanics and Engineering **422**, 116789 (2024) https://doi.org/10.1016/j.cma.2024.116789

[27] Wu, L., Noels, L.: Self-consistency reinforced minimal gated recurrent unit for surrogate modeling of history-dependent non-linear problems: Application to history-dependent homogenized response of heterogeneous materials. Computer Methods in Applied Mechanics and Engineering **424**, 116881 (2024) https://doi.org/10.1016/j.cma.2024.116881

[28] Sverdrup, F., Pellegrino, A., Tagarielli, V., Tasdemir, B.: A machine learning approach to modelling temperature-dependent cyclic behaviour. Materials Today Communications **45**, 112369 (2025) https://doi.org/10.1016/j.mtcomm.2025.112369

[29] Masi, F., Stefanou, I., Vannucci, P., Maffi-Berthier, V.: Thermodynamics-based artificial neural networks for constitutive modeling. Journal of the Mechanics and Physics of Solids **147**, 104277 (2021) https://doi.org/10.1016/j.jmps.2020.104277

[30] Hildebrand, S., Klinge, S.: Hybrid data-driven and physics-informed regularized learning of cyclic plasticity with neural networks. Machine Learning: Science and Technology **5**(4), 045053 (2024) https://doi.org/10.1088/2632-2153/ad95da

[31] Hildebrand, S., Klinge, S.: Physics-informed neural network modeling of cyclic plasticity for steel alloy 4130. Procedia Structural Integrity **72**, 520–528 (2025) https://doi.org/10.1016/j.prostr.2025.08.134

[32] Shoghi, R., Hartmaier, A.: A machine learning-based constitutive model incorporating history-dependent features for cyclic plasticity. Computer Methods in Applied Mechanics and Engineering **448**, 118524 (2026) https://doi.org/10.1016/j.cma.2025.118524

[33] Weng, H., Bamer, F., Luo, C., et al.: Physics-informed neural network for constitutive modeling of cyclic crystal plasticity considering deformation mechanism. International Journal of Mechanical Sciences **302** (2025) https://doi.org/10.1016/j.ijmecsci.2025.108857

[34] Zhu, Y., Chen, S., Li, X.: Thermodynamic-constrained neural modeling for path-dependent geomaterials via dynamic skeleton states. Theoretical and Applied Mechanics Letters, 100740 (2026) https://doi.org/10.1016/j.taml.2026.100740

[35] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997) https://doi.org/10.1162/neco.1997.9.8.1735

[36] Cho, K., Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)