Recurrent neural networks and transfer learning for predicting elasto-plasticity in woven composites

Downloaded from: https://research.chalmers.se, 2026-10-16 03:02 UTC

Citation for the original published paper (version of record):
Ghane, E., Fagerström, M., Mirkhalaf, M. (2024). Recurrent neural networks and transfer learning for predicting elasto-plasticity in woven
composites. European Journal of Mechanics, A/Solids, 107.
http://dx.doi.org/10.1016/j.euromechsol.2024.105378

N.B. When citing this work, cite the original published paper.

research.chalmers.se offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all kind of research output: articles, dissertations, conference papers, reports etc. since 2004. research.chalmers.se is administrated and maintained by Chalmers Library

(article starts on next page)

European Journal of Mechanics / A Solids 107 (2024) 105378

Contents lists available at ScienceDirect

European Journal of Mechanics / A Solids

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

Full length article

Recurrent neural networks and transfer learning for predicting elasto-plasticity in woven composites

Ehsan Ghane [a], Martin Fagerström [b], Mohsen Mirkhalaf [a,*]

[a] Department of Physics, University of Gothenburg, Gothenburg, Sweden
[b] Department of Industrial and Materials Science, Chalmers University of Technology, Gothenburg, Sweden

ARTICLE INFO

Keywords:
Woven composites
Elasto-plasticity
Computational modeling
Recurrent Neural Networks
Transfer-learning

ABSTRACT

Woven composites exhibit complex meso-scale behavior depending on meso- and micro-structural parameters. Accurately modeling their mechanical response is challenging and computationally demanding, especially for inelastic behavior. To address the computational burden, we have developed a Recurrent Neural Network (RNN) model as a surrogate for meso-scale simulations. As a basis for RNN training, a mean-field model generates a comprehensive data set representing elasto-plastic behavior. Arbitrary six-dimensional time histories of strain are used to generate multiaxial stress–strain histories under random walking and cyclic loading conditions as the source and target tasks, respectively. First, the RNN model is trained for the source task. The same model is trained leveraging *transfer learning* for the target task, containing fewer data and sparse features because only some strain components are non-zero. The candidate model is successfully trained and validated through a grid search exploration of over 220 different RNN configurations and demonstrates accurate predictions for both source and target tasks. The results demonstrate that transfer learning could be used to train the RNN effectively under varying strain conditions and arbitrary constituents' material properties, suggesting its potential as an appropriate tool for modeling path-dependent responses in woven composites.

1. Introduction

Woven composite laminates are commonly used in structural applications due to their automated and cost-effective manufacturing processes. However, modeling of woven composites presents significant challenges due to the presence of two heterogeneous sub-scales, the meso-scale, and the micro-scale, and the intricate interlacing of yarns, resulting in the development of complex stress states (Doitrand et al., 2017).

In order to predict the complex behavior of woven composites governed by the heterogeneous sub-scales configuration, different full-field micro-mechanical and meso-scale models have been developed (*e.g.* (Ma et al., 2021; Doitrand et al., 2015, 2017)). However, one of the major challenges of using meso-scale models is their high computational cost, which hinders the usage of these models for engineering applications (Spilker et al., 2023). As a remedy, mean-field models have been proposed and used (*e.g.* (Wu et al., 2021b,a)). In these models, average stress and average strain are considered for each sub-scale constituent. A better computational performance (compared to full-field models) is obtained, although at the expense of lower fidelity and accuracy.

Recently, data-driven approaches have gained considerable interest in developing surrogate models for different composites (*e.g.* (Calleja Vázquez et al., 2023; Rocha et al., 2021; Mentges et al., 2021; Maia et al., 2023; El Said, 2023; Bessa et al., 2017; Dekhovich et al., 2023)). Different kinds of Artificial Neural Networks (ANNs) have been used to develop remarkably efficient and highly accurate surrogate models. A feed-forward architecture is typically good enough to develop an ANN-enhanced model in the linear elastic regime (Ghane et al., 2023). However, for inelastic path-dependent behavior, it is required to use more advanced ANN architectures (Rosenkranz et al., 2023). In recent years, different kinds of Recurrent Neural Networks (RNNs), such as Gated Recurrent Units (GRU) (Cho et al., 2014b) and Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), have been employed for the inelastic path-dependent behavior of different composite materials (see *e.g.* (Mozaffar et al., 2019; Wu and Noels, 2022; Friemann et al., 2023; Maia et al., 2023; Li and Zhuang, 2020)). As a result, a remarkable computational enhancement and a high level of accuracy were obtained. Contrary to the current study, in such works, the constitutive properties of the materials remain constant at a lower scale, and the generalization ability of the model is determined by loading conditions and micro-structural

* Corresponding author.
E-mail address: mohsen.mirkhalaf@physics.gu.se (M. Mirkhalaf).

https://doi.org/10.1016/j.euromechsol.2024.105378
Received 23 February 2024; Received in revised form 13 May 2024; Accepted 20 June 2024
Available online 28 June 2024
0997-7538/© 2024 The Author(s). Published by Elsevier Masson SAS. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

morphologies (Mentges et al., 2021; Cheung and Mirkhalaf, 2024). An interested reader in an overview of recent developments, challenges, and potential future developments is referred to Mirkhalaf and Rocha (2024).

Training RNNs is sensitive to initializing the training parameters (weights and thresholds) due to the risk of vanishing or exploding gradients (Glorot and Bengio, 2010). Vanishing gradients result in slow learning, while exploding gradients lead to unstable training. Some methods have been suggested for initializing layer weights and biases (He et al., 2015; Glorot and Bengio, 2010). These methods can greatly impact how well the deep network trains. However, they are still based on randomly initializing the training parameters.

Furthermore, RNNs encounter challenges when dealing with sparse feature data sets where inputs have many zero values. For instance, in the case of a stress prediction task using strain tensor components as inputs, when we have pure-shear loading, only one component of the input feature is non-zero. This results in data sparsity that can interfere with learning and hinder the network's ability to capture meaningful patterns.

This paper employs a physics-guided initialization (Benady et al., 2023) of weights and thresholds using *transfer learning* (Yang et al., 2020). Using transfer learning, RNNs are able to overcome initialization challenges by leveraging knowledge derived from previously trained models. The network is initialized with the pre-trained model containing the expected material parameters, known as the source task. The network is then fine-tuned in accordance with the target task. This study's source task network is to predict the full stress history of plain woven composites with different constituent properties subjected to random multiaxial strain histories. The target task is to predict stress histories under conventional cyclic loading. Models that are trained on dense or diverse data sets can provide more robust representations of features that can be generalized well to sparse data sets. This approach accelerates training, enhances generalization over sparse feature samples, and facilitates effective learning.

This study formulates various GRU and LSTM models to predict the elasto-plastic behavior of plain woven composites. The specific nature of the task determines the choice between GRU and LSTM architectures. Moreover, the optimal performance of these models is closely linked to the number of training parameters. This facet is evaluated in depth within the framework of this study.

To generate the two required data sets, a mean-field model from Digimat-MF (Anon, 2016) is employed. The matrix and reinforcements are considered elastic–plastic and elastic, respectively. Notably deviating from prevailing trends observed in developing material-specific ANN surrogate models for composites (*e.g.* (Dornheim et al., 2023; Maia et al., 2023)), this study uniquely incorporates many matrix and reinforcement properties into its framework.

Six-dimensional arbitrary loading paths (for six independent strain components) are generated and applied to the meso-structural simulations. The simulation results serve as a data set for the source task. Bi-axial and pure shear cyclic load paths are generated for the target task data set, each containing a different peak strain, strain ratio, load ratio, and number of cycles. As a result, two comprehensive data sets (including stress–strain responses) for generic woven composites (varying matrix and reinforcement properties) subjected to randomly sampled and cyclic loading histories are generated.

When training a neural network, highly sensitive parameters and hyperparameters are present, such as learning rate, minibatch size, regularization strength, dropout rate, and network architecture (Pascanu et al., 2013). The identification of the best combination of hyperparameters requires a comprehensive study, which is frequently neglected throughout the literature due to the time-consuming nature of the process (Mozaffar et al., 2019; Abueidda et al., 2021; Huang et al., 2020; Vlassis and Sun, 2023; Jones et al., 2022; Friemann et al., 2023). We attempt to determine an optimal network by testing many possible combinations. The results show that an LSTM model is successfully trained and validated on the target task, enabling highly efficient elasto-plastic path-dependent simulations.

The remainder of this paper is structured as follows. Section 2 describes the data generation process, including the sub-scale mean-field modeling approach, material constituents, and design of computational experiments. Section 3 details the RNN model design and training and transfer-learning strategy. The obtained results and comparisons to micro-mechanical simulations are presented in Section 4, followed by a discussion of the developed RNN model. Concluding remarks are provided in Section 5.

2. Data generation

Every data sample contains (i) a particular set of constituent material properties, (ii) a 6D random strain loading path, and (iii) a 6D time history of homogenized stress components. The database is generated, considering various material parameters and loading conditions, to capture the complex behavior of various woven composites subjected to complex strain states. Two comprehensive data sets (one for the random walk[1] loading and one for the cyclic loads) are created by carefully controlling the input variables and using mean-field simulations as described below.

2.1. Homogenization of woven composite

Several scale-transition methods have been developed to compute the effective (homogenized) macroscale behavior of heterogeneous materials (Spilker et al., 2023), including computational homogenization (Xu et al., 2015; Espadas-Escalante and Isaksson, 2019), and mean-field homogenization (MFH) (Udhayaraman and Mulay, 2019). Computational homogenization of Representative Volume Elements (RVE) is commonly employed for woven composites. While this method offers general accuracy and detailed microscale fields, it presents notable drawbacks. Generating a suitable mesh for complex or realistic microstructures can be challenging and time-consuming. Additionally, computational costs, including CPU time and memory usage, can become excessive, particularly for nonlinear problems.

A mean-field model which uses the Mori–Tanaka theory (Mori and Tanaka, 1973) for homogenization, implemented in Digimat-MF (Anon, 2016), is used to conduct non-linear path-dependent elasto-plastic simulations of woven composites with varying properties. When employing the MFH method to homogenize woven composite behavior, neither the specific weave pattern nor the resulting crimp (or undulations) can be considered. The yarn region is not explicitly specified, and the reinforcement material properties are assumed to be transversely isotropic, aligned with the direction of the reinforcements. The composite is assumed to have a balanced weave, and the microstructure is solely determined by the assumed orthogonal orientations of the warp and weft directions. Notably, failure mechanisms are excluded from consideration. While the geometry of the meso-scale structure remains unchanged, the micro-structural constituent's properties change in each virtual sample (see more details in Section 2.2 below). As a consequence, the MFH procedure involves two steps. In the first step, the sub-scale of the composite material being studied is divided into smaller units called pseudo-grains (PGs). Each PG represents a localized region within the composite. Once the division into PGs is completed, the homogenization process begins. In the case of a woven composite, there are two PGs. One in the warp and one in the weft directions. Each PG is individually subjected to a homogenization procedure, where the

[1] It is worth noting that the term "random walk" is often associated with stochastic processes, where an unpredictable element determines the next step or state. Random walks in strain loading suggest that a strain value evolves over time due to random factors, resulting in a pattern that may resemble a walker's path.

Table 1
blueConstitutive material parameters for a sample carbon fiber/epoxy resin used in comparing MFH and FE..

	Parameter	Value
Fiber	Young's modulus E_F	385.5 GPa
	Poisson's ratio ν_F	0.37
	Fiber volume fraction V_f^{RVE}	0.16
Matrix	Young's modulus E_M	6 GPa
	Poisson's ratio ν_M	0.34
	Yield stress σ_y	48.5 MPa
	Linear hardening modulus H	100.5 MPa
	Hardening modulus H_∞	20 MPa
	Hardening exponent m	250.5

behavior and properties of the constituent materials within the PG are analyzed. In the second step, homogenization is extended to the entire sub-scale. The effective response of the entire sub-scale is computed by averaging over the collective behavior of all the homogenized pseudo-grains. An interested reader is referred to Doghri and Tinel (2005) for a more comprehensive understanding of the modeling approach.

To demonstrate a comparison, Fig. 1 illustrates shear stress–strain responses of RVEs with a particular set of constitutive material properties (given in Table 1) but with different yarn cross sections under an in-plain shear cycle. Accordingly, the MFH of the same constitutive material system is plotted. The balanced weave in the RVE homogenization approach has 15 yarns/cm in both warp and weft with a yarn spacing ratio equal to 0.1 and 0.5 crimp. The RVE overall fiber volume fraction (V_f^{RVE}), the amount of reinforcing fibers in a composite material relative to the total volume of the composite, is restricted to be constant and equal to the one in MFH by controlling the filament count and yarn density. Nevertheless, the yarn cross-section area varies in different realizations by changing spindle heights. Based on the comparison results, the adopted MFH serves as an approximation method and does not precisely capture the weaving pattern and behavior of the woven composite; however, the FE model is, on average, twelve times more time-consuming than the MFH model (up to 60 s). The main goal of this study is to develop a proof of concept for data-driven modeling of the complex non-linear behavior of woven composites. Hence, the trained neural network at the end is not designed to outperform the RVE computational homogenization nor MFH in terms of accuracy, but rather to demonstrate flexibility with respect to the material selection and speed. Therefore, we adopt the semi-analytical MFH model for the data-generation phase because it remains highly efficient compared to computational homogenization methods, which effectively serves the purpose of the current study.

The primary advantages of MFH models include their ease of use, low CPU time requirements, and reduced memory usage. The MFH model, however, provides only approximations of the volume averages of stresses and strains, either at the macro level or at the phase level. In order to improve homogenization accuracy, the framework is always open to adding more accurate data bases.

2.1.1. Constitutive behavior of sub-scale phases

Polymeric materials typically show a strain rate-dependent mechanical response (see, *e.g.* (Mirkhalaf et al., 2016, 2017)). However, an approximation of rate-independent behavior could be considered for most thermoset polymeric materials under quasi-static loading rates and at room temperature. Therefore, this study considers a rate-independent elasto-plastic response for the matrix material.

The matrix is assumed to obey J_2-plasticity with linear-exponential hardening (Simo and Hughes, 2006). The yield function is given by

$$\Phi(\sigma, \kappa) = \sigma_{vM} - (\sigma_y + \kappa) \leq 0, \tag{1}$$

where σ_y is the yield stress, and σ_{vM} is the von Mises stress defined by

$$\sigma_{vM} = \sqrt{\frac{3}{2}\boldsymbol{\sigma}_{dev} : \boldsymbol{\sigma}_{dev}}, \quad \boldsymbol{\sigma}_{dev} = \boldsymbol{\sigma} - \frac{1}{3}\mathrm{tr}(\boldsymbol{\sigma})\mathbf{I}. \tag{2}$$

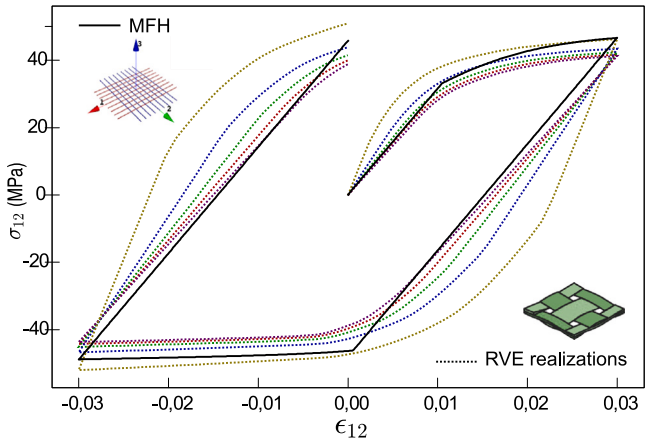

Fig. 1. Comparison of mean-field homogenization (solid black line) and finite-element homogenization (colored dashed lines) for four RVE realizations with different yarn cross sections under one in-plain shear cycle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Ranges of material parameters used to generate the required data set during simulations. All data samples in the training and testing sets are distinguished by a unique set of properties of their micro-structural constituents.

	Parameter	Value
Fiber	Young's modulus E_F	69–700 GPa
	Poisson's ratio ν_F	0.25–0.49
	Fiber volume fraction V_f	0.10–0.48
Matrix	Young's modulus E_M	2–10 GPa
	Poisson's ratio ν_M	0.2–0.49
	Yield stress σ_y	31–66 MPa
	Linear hardening modulus H	1–200 MPa
	Hardening modulus H_∞	10–30 MPa
	Hardening exponent m	1–500

In Eq. (2), $\boldsymbol{\sigma}_{dev}$ is the deviatoric stress tensor, and $\mathbf{I}$ is the second-order identity tensor. In Eq. (1), κ is the hardening stress which is given by

$$\kappa = H\bar{\varepsilon}^p + H_\infty\left(1 - e^{-m\bar{\varepsilon}^p}\right), \tag{3}$$

where H is the linear hardening modulus, H_∞ is referred to the hardening modulus, $m > 0$ is the hardening exponent, and $\bar{\varepsilon}^p \geq 0$ is the accumulated plastic strain.

Reinforcements are assumed to be isotropic and linearly elastic and obey Hooke's generalized law. Furthermore, the matrix and reinforcement phases are assumed to be perfectly bonded. Despite the fact that this assumption may not be true in all cases, it provides a reasonable basis for examining the overall behavior of the woven composite under study.

2.2. Design of computational experiments

This study involves two sets of input features for computational experiments: (i) static features representing fiber and matrix material properties and (ii) multi-dimensional sequential load path components (Section 2.2.2). A wide range of properties are considered for the static features, which are given in Table 2.

2.2.1. Sampling material features

Having uniformly distributed input features to train an ANN properly is beneficial. Regular grids of sample points can lead to coincident projections in different hyper-planes (Bessa et al., 2017), negatively impacting machine learning, especially in high-dimensional spaces (Bishop and Nasrabadi, 2006). Thus, using an effective sampling technique helps to achieve a random and uniform distribution while reducing simulation costs.

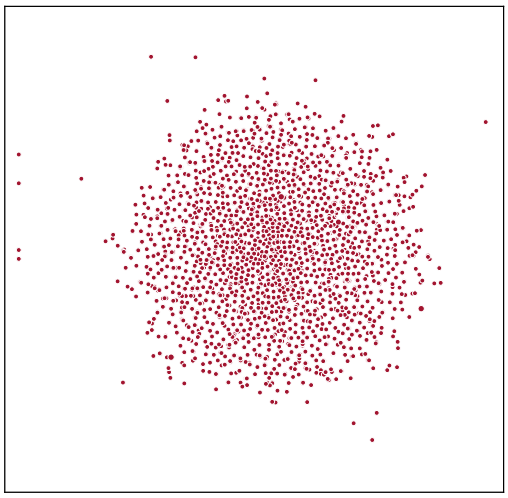

Fig. 2. t-SNE distribution (van der Maaten and Hinton, 2008) of 9-dimensional static feature space described in Table 2. Different micro-structural configurations are described by three fiber elasticity, three matrix elasticity, and three matrix plasticity feature parameters. The 2D scatter plot shows no clustering or pattern in the input space.

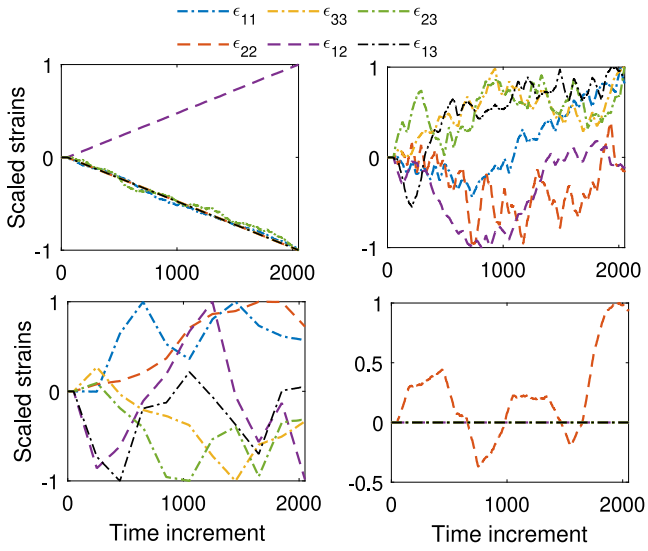

Fig. 3. Four samples of input strain loading paths from random walk data set (scaled between [−1,1]). Each graph contains six components of the strain tensor applied on a randomly chosen material set.

Random sampling and stratified sampling often result in clusters and gaps in a data set. Alternatively, Sobol sequence sampling (Saltelli et al., 2010), a quasi-random sampling technique, offers a solution. Unlike other pseudo-random algorithms, Sobol sequence sampling avoids clustering and gaps even in smaller data sets (Renardy et al., 2021). It aims to generate multiple parameters uniformly distributed across a multi-dimensional parameter space. Thus, in the current work, Sobol sequence sampling generates a comprehensive design space. 10,000 data samples are generated with different combinations of the static features given in Table 2. The resulting design space enables the exploration of various material and micro-structural configurations, providing valuable insights into the behavior of woven composite materials.

For visualizing high-dimensional data sets, t-SNE (t-Distributed Stochastic Neighbor Embedding) is utilized (van der Maaten and Hinton, 2008; Hinton and Roweis, 2002). It minimizes the discrepancy between high-dimensional and low-dimensional distributions using the Kullback–Leibler (KL) divergence (Mehlig, 2021) as a cost function. KL divergence quantifies how one probability distribution (from a multi-dimensional space) diverges from the expected (into a 2D space) probability distribution. The algorithm iteratively adjusts data point positions to minimize the cost function. Monitoring KL divergence during cost minimization indicates capturing data structure and relationships in the lower-dimensional domain successfully. t-SNE preserves local structures, patterns, and clusters. While it is primarily a visualization tool, it can effectively represent data distribution, as shown in Fig. 2. No pattern or gap is visible on the scattered 2D plot, indicating a good correlation between regular grids and random distributions in the static feature space.

2.2.2. Loading path generator

In order to sample representative strain paths, different approaches can be employed, *e.g.* (Heidenreich et al., 2023; Friemann et al., 2023). In the source task, a random walk representing long-term trends is combined with noises representing local variations. The algorithm (Friemann et al., 2023) utilizes a six-dimensional space for independent strain components. Components are sampled independently from a normal distribution to generate direction vectors, then normalized to unit vectors. The algorithm defines parameters: N_T as the total number of steps (constant and equal to 2000), n_1 as the number of drift directions (the number of major changes in the loading direction selected from $\{1, 2, 5, 10, 20, 25, 50, 100, 200\}$), n_2 as the noise vector with elements selected randomly from $[0, 1)$, and γ as the perturbation amplitude factor (chosen randomly from $[0, 1)$).

Initially, n_1 drift directions are chosen randomly, and each of these directions is iteratively repeated N/n_1 times to form a vector comprising a total of N_T elements. Subsequently, a noise vector, denoted as n_2, is generated with an equivalent number of elements as N/n_1 and is then scaled by a factor of γ. The vector of drift directions and the scaled noise vector are combined and scaled to a maximum of randomly selected 1% to 5% to complete one strain component of an input load sample. All components of a strain loading sample have the same number of n_1 but differ in other parameters. Fig. 3 shows four samples of the generated loading paths.

Some loading samples have sparsity in their input features, such as the fourth case in Fig. 3, where the input strain tensor only includes the transverse strain component, while all other components are zero. Since this rarely occurs, the trained network based on such a data set may need help to generalize the solution to cases with high feature sparsity, such as conventional cyclic loadings. Therefore, a second data set is generated based only on cyclic loads in shear and bi-axial load cases where plasticity is significant in woven composites. Fig. 4 shows four samples from the second data set. Indicating factors in the cyclic loads include the peak strain value, the strain ratio (ratio of shear strain to tensile strain in bi-axial scenarios), the load ratio (fraction of the maximum positive strain to the minimum negative strain value), and the number of cycles.

3. Recurrent neural networks

RNNs excel in handling data sequences of long lengths, making them suitable for language modeling, speech recognition (Chan et al., 2015), machine translation (Cho et al., 2014b; Sutskever et al., 2014), and time-series prediction (Mehlig, 2021). RNNs effectively model sequential data due to their internal memory, capturing temporal dependencies. Feedback loops in the RNN architecture enable information flow between input and output, maintaining an internal state for predictions based on current and past inputs.

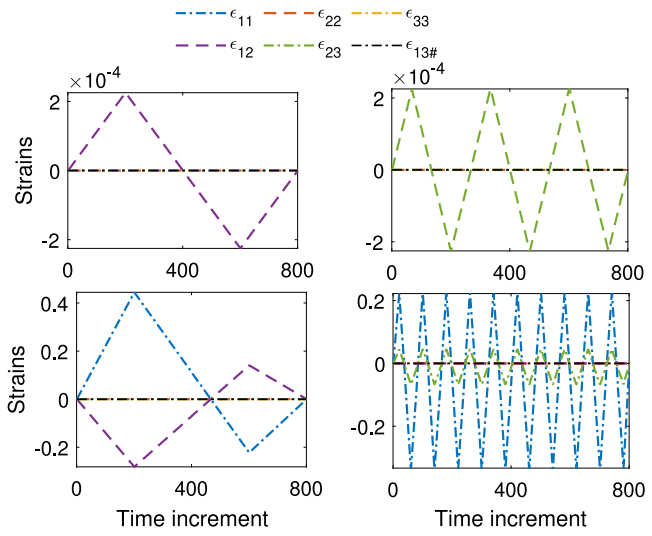

Fig. 4. Four input strain loading paths samples from a smooth cyclic path data set. Each graph contains six components of the strain tensor applied on a randomly chosen material set. The smooth cyclic loading paths include (from up-left to down-right) pure in-plain shear ϵ_{12}, out-of-plain shear ϵ_{23} and bi-axial $[\epsilon_{11}, \epsilon_{12}]$ and $[\epsilon_{12}, \epsilon_{23}]$ loading.

3.1. Transfer learning rescuing RNNs from initialization and sparsity hurdles

In the transfer learning paradigm, a model initially trained on a source task with abundant data serves as a foundational framework for a closely related target task. The primary objective is to leverage the pre-trained model's acquired knowledge of features, representations, and patterns to enhance the learning process on the target task (Yang et al., 2020). Furthermore, during the training phase for the source task, pre-trained models undergo regularization mechanisms, which mitigate overfitting risks. Transfer learning entails initializing the model with optimized weights specific to certain features, resulting in noteworthy acceleration of convergence and optimization improvements tailored for the target task. Transfer learning has been proven helpful for data fusion in elastic regime (Callaghan et al., 2023) and elasto-plastic behavior of short-fiber composites (Jung et al., 2022).

One specific strategy within transfer learning is fine-tuning (Heidenreich et al., 2023). This approach is employed when the source and target tasks are closely related, allowing for the adjustment of weights in a pre-trained model based on the data specific to the target task. In the context of this study, the source task involves predicting the six components of stress sequences derived from random loading simulations. Subsequently, during the original network training, the target task is formulated to predict stress components associated with cyclic loading paths. The RNN model employed for the source task integrates an a priori model, initially trained on extensive data sets, to adapt and enhance its performance on novel tasks. The weights and biases derived from the neural networks at the end of training serve as the initialization for the neural networks used in transfer learning.

3.2. Recurrent learning with GRU and LSTM units

Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) networks are two types of recurrent neural networks (RNNs) that address the vanishing and exploding gradient problems encountered in traditional RNNs (Pascanu et al., 2013; Lipton et al., 2015; Chung et al., 2014). GRUs use two so-called gating mechanisms, an update gate, and a reset gate, to selectively update and reset their hidden state. This enables them to gather information across time steps and retain long-term dependencies (Cho et al., 2014a). LSTM units, on the other hand, employ three gating mechanisms: the input gate, the forget gate, and the output gate. These gate mechanisms control the flow of information, select information to retain or discard, and filter relevant information for output, respectively (Hochreiter and Schmidhuber, 1997). An interested reader is referred to Author(s) of the Documentation (2021) for a more comprehensive understanding of LSTM and GRU units. Since GRUs have only two gates, they are computationally more efficient and easier to train with fewer parameters. However, they may not be as effective at capturing complex long-term dependencies as LSTMs. LSTMs with three gating mechanisms can model more complex relationships over longer sequences, but they have a higher computational cost and require more training data to prevent overfitting (Géron, 2022). The choice between GRUs and LSTMs depends on the specific task at hand, the availability of computational resources, and the desired trade-off between model complexity and performance. To better understand how data flows through an RNN network containing two layers of recurrent units, Fig. 5 is provided. Learning rate, minibatch size, regularization strength, dropout rate, and network architecture are crucial parameters during RNN training. This is especially critical before the transfer learning process on the target task. Regularization (also called weight decay) (Bishop and Nasrabadi, 2006) adds a penalty term to the loss function, promoting smaller weights in the network. This prevents the model from becoming overly sensitive to training data and enhances its generalization ability to unseen data. Additionally, the dropout layer (Srivastava et al., 2014) randomly sets a fraction of input units to zero during training, preventing co-adaptation of neurons and encouraging the network to learn more robust and independent representations. Combining L_2 regularization with dropout mitigates overfitting, reduces model complexity, and fosters more generalizable representations (Murphy, 2012). Gradient clipping, setting the absolute value to one or using the maximum of normalized features, helps eliminate exploding gradients (Pascanu et al., 2013).

3.3. Feature scaling

Differences in input variable magnitudes can bias neural networks and hinder network learning. Variables with larger values dominate and overshadow smaller ones, leading to unstable weight updates and sub-optimal performance. Standardization equalizes input feature scales, ensuring smoother convergence in the network training and preventing disproportionate variable influence. It also allows direct magnitude comparison and enhances interpretability, thereby promoting stable training, faster convergence, and improved neural network performance (Mehlig, 2021).

The standardization process involves normalizing sequential strain components to an absolute maximum value of one. Non-sequential material properties undergo min–max scaling to have values between zero and one. This enables neural networks to capture the underlying patterns better, resulting in improved performance and reliable predictions.

The network input consists of sequential data with 15 different features, concatenating 6 sequential strain tensor components and 9 static material properties. In each instance, the non-sequential features (the material properties) are repeated throughout the entire sequence of $N_T = 2000$ pseudo-time increments.

The output signals are components of the stress tensor. In Voigt notation, the symmetric stress tensor is represented as a six-component column vector as follows:

$$\tilde{\boldsymbol{\sigma}} = (\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23})^T \equiv (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6)^T \tag{4}$$

The mean squared error loss function is computed at the regression layer for training through back-propagation. The loss function for a batch of data is given by

$$L = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} L_i, \quad \text{with} \quad L_i = \frac{1}{2N_T} \sum_{t=1}^{N_T} \sum_{c=1}^{6} (\hat{\sigma}_c^{(t)} - \sigma_c^{(t)})^2. \tag{5}$$

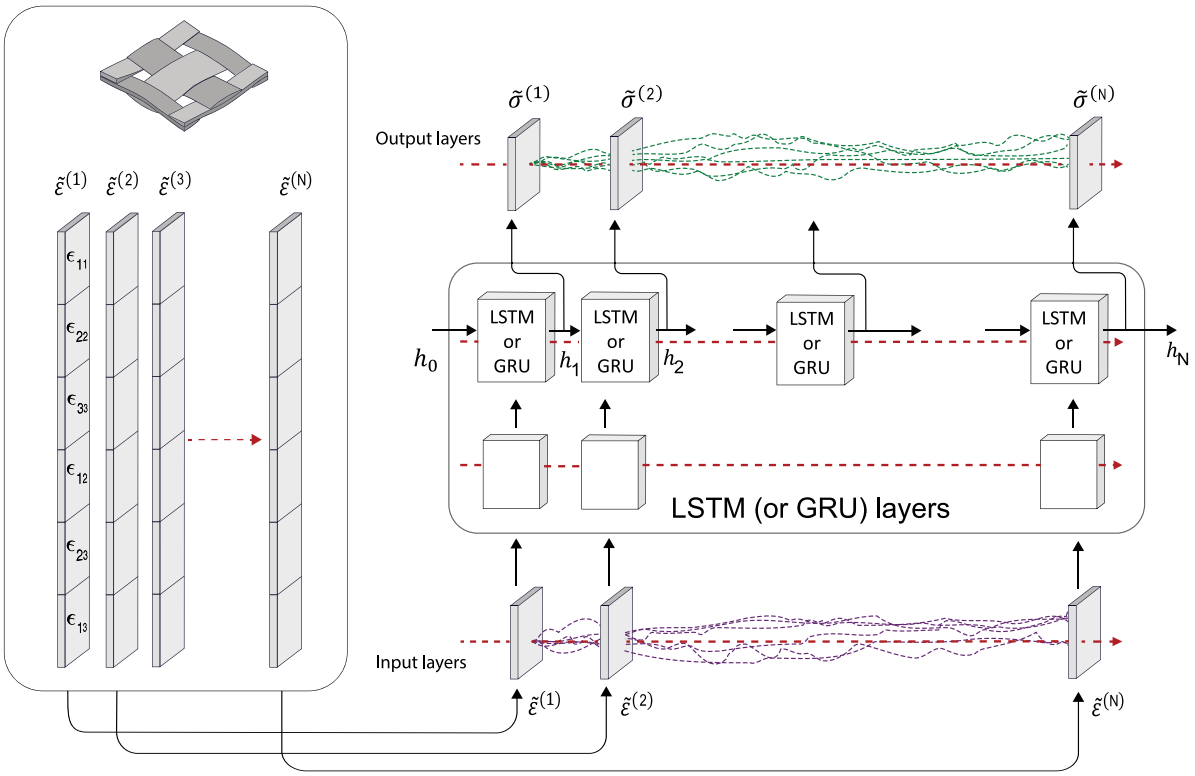

Fig. 5. A schematic representation of the data flow for feeding the RNN model with one multiaxial strain history. As shown on the left side of the figure, a woven composite and strain loading paths have been randomly sampled from the data set. Each stripe represents a time step of the strain tensor ($\tilde{\epsilon}^{(n)}$) containing six strain components (ϵ_{pq}). Red dashed lines indicate time increments in a load path. Six random loads are presented in the input layer in purple. Afterwards, the data propagates through the LSTM (or GRU) network layers. The network outputs are in green. The internal variables flow can also be seen in one of the network layers. A hidden (and/or cell) state is indicated by h_t. It should be noted that the RVE geometry in the input space is only representing the material type and not the actual type of data used in this study. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

As described in Eq. (4), six components of the stress tensor is considered in the output. $\hat{\sigma}_c^{(t)}$ and $\sigma_c^{(t)}$ are the predicted and desired normalized stress component at time step t in a training sample, respectively. N_T is the data sequence length, and N_{batch} is the batch size (the number of samples in each batch.) The reason for adding the $\frac{1}{2}$ is a common practice, primarily for computational convenience during the optimization process.

3.4. Metrics beyond loss

Network performance relies not only on architecture but also on hyperparameters like learning rate, regularization strength, batch size, etc. The choice of hyperparameters can significantly impact the validation loss, potentially overshadowing the effect of selecting the optimal network architecture based solely on validation loss described in Eq. (5). The effectiveness of loss as a performance metric after training is limited, mainly because it does not incorporate the effect of normalization. This limitation makes it difficult to effectively compare outputs across different test samples when they exhibit varying magnitudes of stress components.

This study considers additional evaluation metrics to find an optimal network configuration. For this purpose, we use the von Mises stress as defined in Eq. (2) as a scalar representative measure of the stress output. It is acknowledged that the von Mises stress measure is not necessarily appropriate to define the behavior of anisotropic materials such as woven composites, but it still serves as a convenient scalar measure of the magnitude of the stress.

Three statistical measures (Willmott and Matsuura, 2005) are used to assess and compare the predictive performance of multiple neural network configurations. In the following, $e_i^{(t)}$ represents an individual model-prediction (for one sample) error at time step t, defined as

$$e_i^{(t)} = \hat{\sigma}_{vM,i}^{(t)} - \sigma_{vM,i}^{(t)}, \tag{6}$$

where $\hat{\sigma}_{vM}^{(t)}$ is the predicted von Mises stress at time step t and $\sigma_{vM}^{(t)}$ is the desired von Mises stress at t.

Mean Absolute Error

The mean Absolute Error (MAE) measures the average magnitude of errors over the data sequence length (N_T) between the predicted and desired values. MAE is calculated by taking the average of the absolute differences between each predicted and desired value, normalized by the number of tested samples from the unseen data set (M):

$$\text{MAE} = \frac{1}{M N_T} \sum_{i=1}^{M} \sum_{t=1}^{N_T} \left| e_i^{(t)} \right|. \tag{7}$$

MAE provides an indication of the average size of errors produced by the model. There is no consideration for the direction of errors (overestimation or underestimation), and all errors are given equal weight. A lower MAE indicates a better performance.

Root Mean Square Error

Another commonly used measure of prediction error is the Root Mean Square Error (RMSE). It is obtained by taking the square root of the average of the squared differences between the predicted values and the desired values as

$$\text{RMSE} = \sqrt{\frac{1}{M N_T} \sum_{i=1}^{M} \sum_{t=1}^{N_T} (e_i^{(t)})^2}. \tag{8}$$

Due to the squaring operation, larger errors are penalized more heavily. RMSE gives a measure of the overall deviation or dispersion of errors.

Similarly to MAE, a lower RMSE indicates a better performance. Similarly to MAE, RMSE does not consider whether the model overestimates or underestimates a quantity at a given time instant.

Mean Bias Error

The bias or Mean Bias Error (MBE) refers to the systematic deviation between model predictions and the desired values in a data set. Bias can be positive or negative, indicating whether the predictions consistently overestimate or underestimate desired values. The scale and units of the predicted and desired values typically determine the bias range. MBE is computed by

$$\text{MBE} = \frac{1}{M N_T} \sum_{i=1}^{M} \sum_{t=1}^{N_T} e_i^{(t)}. \tag{9}$$

4. Results and discussion

For data consistency, only completed simulations are used.[2] The computational programming language MATLAB (Author(s) of the Documentation, 2021) is employed for implementing recurrent neural networks. Training for different network configurations is done on the Vera hardware (Anon, 2023) at Chalmers University of Technology.

Two data sets are generated, including 28,000 data samples for the source task on random walk loading paths and 10,000 samples for the target task on smooth cyclic loading paths. Each data set is randomly split into training (80%), validation (10%), and test (10%) sets. For each task, the training data set is iteratively passed through the neural network for multiple epochs, with shuffled data at each epoch. The validation set helps for tuning hyperparameters, such as batch size, learning rate, and regularization parameters, to achieve an optimal model performance. The test set serves as unseen data for final performance evaluation after the network training and validation.

Section 4.1 presents results related to the source task. The purpose of Section 4.2 is to emphasize the importance of tuning the network architecture and hyperparameters. Once the network's generalization ability is confirmed for a random load path, the transfer learning method is used in Section 4.3 for training a network to predict smooth cyclic loads (target task).

4.1. Prediction on the random strains test samples

A piece-wise learning rate decay strategy is employed, reducing the learning rate by 10% every ten epochs to enhance convergence. The training process uses ADAM optimizer (Kingma and Ba, 2014). In order to prevent overfitting, early stopping is used instead of a fixed number of epochs. Training stops when the model's performance on the validation set plateaus, while the loss on the training set continues to decrease. Fig. 6 illustrates the loss evolution for LSTM and GRU networks in accordance with Eq. (5). The plateau region on the validation set indicates convergence in the training of both cases. While the GRU network struggles more to reach the minimum loss value at the beginning, the minimum loss values are almost equal.

The three metrics, defined in Section 3.4, are computed based on the predictions made for each sample in the test set. The mean value over pseudo-time steps for three error metrics is obtained and presented in Tables 1 and 2 in supplementary materials for different network architectures and the number of training parameters. Various configurations of GRU and LSTM networks are presented in these tables in order to determine the best architecture. The network configuration is defined by the number of GRU (or LSTM) units and the specification of the subsequent dropout layer. For example, the "3GRU(128) dp(40%)" network consists of three GRU layers, each containing 128 units, followed by a 40% dropout layer. For detailed error evaluations on GRU and

[2] Some of the simulations with the highest number of loading drifts (200) are not converged in the Digimat-MF solver.

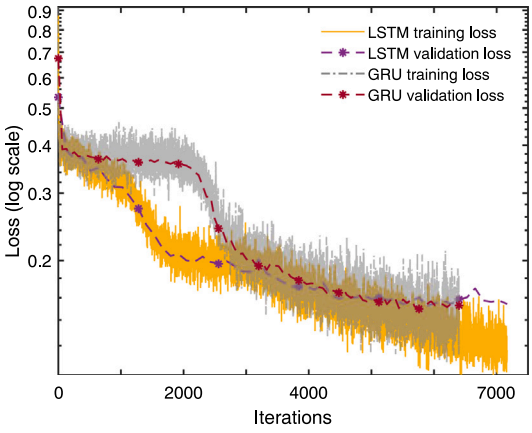

Fig. 6. Two examples of the training and validation loss evolutions on the random strain data set. Each network consists of three layers with 512 units of LSTM or GRU plus a 50% dropout after the first layer. In both cases, the learning rate is 0.001, the L_2 regularization is set to 0.001, and the minibatch size is 128. Yellow and gray lines indicate the training loss calculated at each iteration. The dashed lines in red and purple indicate the loss calculated at the end of each epoch for the validation set. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

LSTM networks, the reader is referred to the supplementary material of this study. Our objective is to determine the optimal architecture first; afterward, various hyperparameters are discussed using LSTM or GRU units in Section 4.2. Based on von Mises stress, Fig. 7 compares two best-performing LSTM and GRU networks in predicting two random walk loads from the unseen data set.

The test dataset was used to evaluate multiple neural network architectures and hyperparameter combinations. As a result of extensive experimentation, we identified the network with the lowest RMSE as the "candidate network". Although the smallest network, "GRU(32)", had the lowest validation loss, it cannot generalize well due to higher MAE and RMSE error ranges when testing unseen test data. "3LSTM(512) dp(50%)" displayed the lowest MAE and RMSE, indicating better prediction and generalization on unseen data than other models. However, note that the MBE of "3LSTM(512) dp(50%)" is not closest to zero, and its negative value suggests a systematic underestimation of von Mises stress.

Fig. 8 shows predictions for stress components on two randomly selected test samples for the candidate network based on error metrics "3LSTM(512) dp(50%)". Fig. 8(a) shows a strong correlation between the predictions and the desired stress values. However, stress prediction deviates for σ_{33}, where the desired value is close to zero throughout the loading increments. It is potentially related to the general underlying issue with neural networks, known as feature sparsity (Mehlig, 2021), which needs further research to handle sequential regression tasks. A network might have difficulty predicting features that are close to zero in a sample when there is a high number of input features (15 in this case).

In spite of a highly random loading path, in Fig. 8(b), the predicted values match well with the desired values from mean-field simulation. It can be seen that the predictions on normal components are better than those on shear components. This observation could be related to the higher level of non-linearity in shear components.

4.2. Discussion on hyperparameters

Various networks have undergone grid search training to find the best hyperparameter combination. The tested hyperparameters have been examined as follows: Minibatch Size = [16, 32, 64, 128], L_2 Regularization = [0.001, 0.01, 0.], Dropout Rate = [0.2, 0.5, 0.8], Learning Rate = [0.0001, 0.001, 0.01]. The grid search method leads to

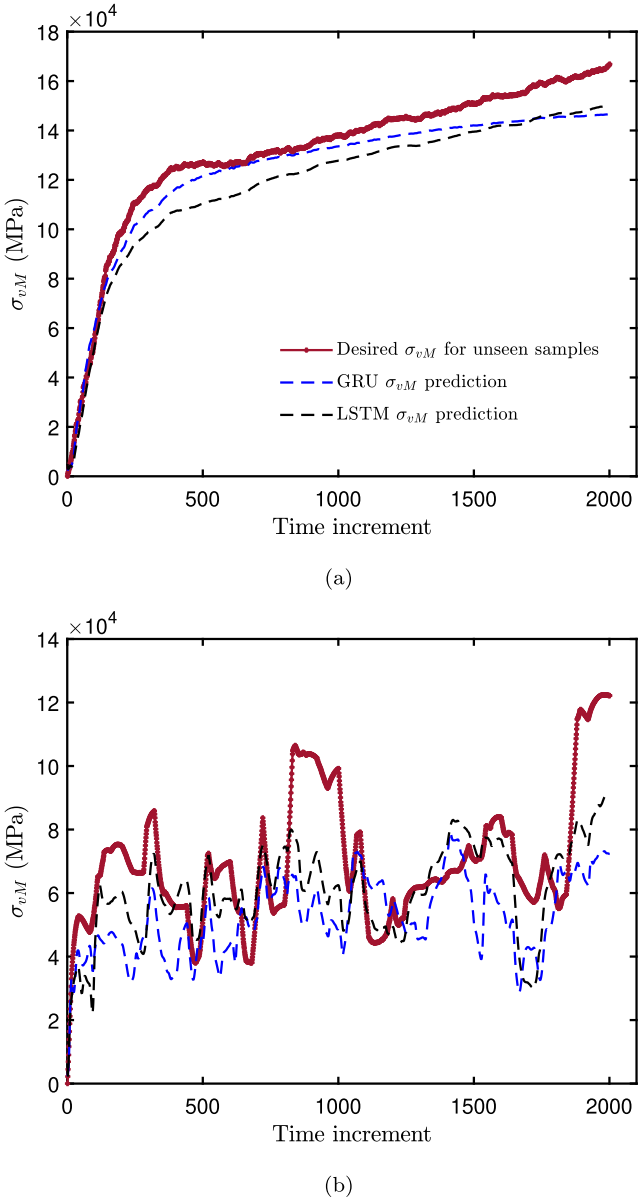

Fig. 7. Candidate LSTM (dashed black lines) and GRU (dashed blue lines) predictions on two samples with (a) $n_1 = 1$ and (b) 25 drift directions from the random walk paths of the unseen data set. The von Mises stresses are plotted on the real scale for two distinct sets of micro-structural constituents material properties. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

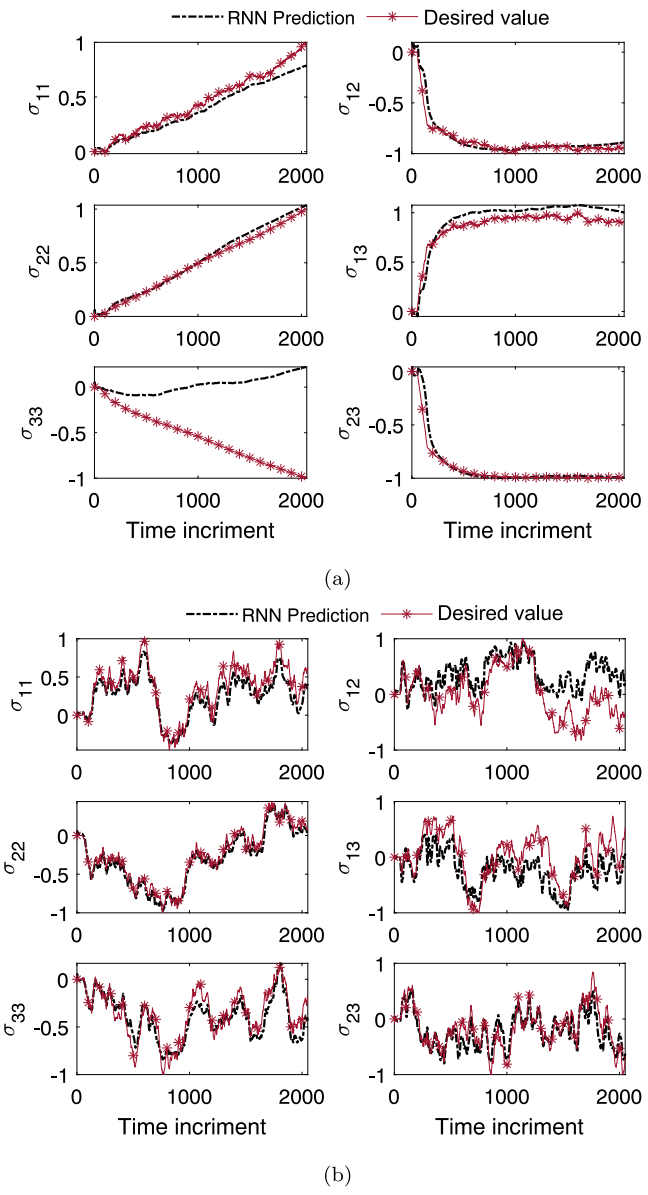

Fig. 8. Network (3LSTM(512) dp(50%)) predictions against micro-mechanical results for two samples from unseen random load cases. Stress values are scaled between $[-1, 1]$ (a) Six components of stresses in a case with a rather uniform load and (b) a case with high randomness in loading. The solid red line is the mean-field simulation output (desired values), and the dashed black lines are the network predictions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

training and evaluating more than 200 LSTM and GRU networks with different hyperparameter combinations. For detailed error evaluations, please refer to Table3 in the supplementary material of this study.

The candidate LSTM and GRU networks have three layers, each containing 512 units. As a result of the hyperparameters grid search, the LSTM network's optimum learning rate and minibatch size are 0.001 and 128, respectively, while the candidate GRU network optimum is 0.1 and 32. The regularization and the dropout rate are optimal in both networks at 0.001 and 50%, respectively. While LSTM networks outperform GRU networks in minimizing RMSE and MAE, they come with the cost of longer training times. Training and validating the tested GRU networks takes 20–100 min, and for LSTM networks, 120–300 min. Once trained, the network can predict the stress–strain response on 2800 random test sets, all in less than a second. In comparison, it takes between 18–60 s to compute each individual response to complex general 3D-loading (such as that found in the random test data) using DIGIMAT-MF.

4.3. Prediction on conventional load cycles

The initial task, referred to as the source task, involves predicting loading cases associated with random walk loading paths. Subsequently, the focus shifts to the target task, where the objective is to predict loading paths characterized by conventional cyclic patterns.

However, a notable challenge is associated with applying the trained network to the target task involving cyclic loads. The input features of cyclic loading samples consist of only one (for pure shear cases) or two (for bi-axial loading cases) active sequential features in addition to the static features (micro-structure). As a result, the cyclic loading

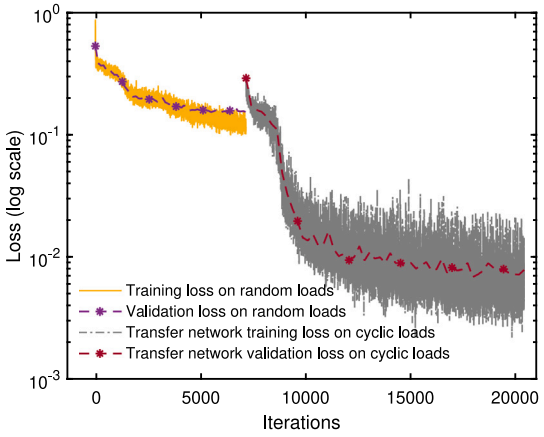

Fig. 9. On the source task, the original LSTM network training and validation losses are presented in yellow and dashed purple, respectively. Transfer learning is then applied to the target task (cyclic loads), and the loss is computed (gray for training and red for validation). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

features are sparse, and the ANN models have difficulty predicting stresses based on sparse samples. Therefore, the network undergoes fine-tuning with a data set specific to the target task. As seen in Fig. 9, the evolution of loss illustrates the convergence of the transfer learning process following the initial training of the LSTM network. As a result of the first 7000 iterations, the network loss function converged on both the training and test sets of the random load paths data set. The loss function jumps at the beginning of the fine-tuning process when samples from cyclic loads are fed to the network. The network is then trained using cyclic load samples through more iterations, and convergence occurs after 20,000 iterations.

The original network is trained on random walk paths and then fine-tuned on cyclic loading. There are two possible alternative scenarios: (1) the original network is limited to being trained on cyclic loads, or (2) the original network is initially trained on cyclic loads and fine-tuned with random walk paths. Three instances of cyclic loads are shown in Fig. 10, corresponding to a one-cycle test, a three-cycle test, and a ten-cycle test, respectively. It can be clearly seen that the original network trained based on the random walk data set (represented by the solid yellow line) cannot predict the cyclic loads despite its performance on the random walk data set. The dashed gray line indicates that the data set for the cyclic load is insufficient and that most features are too sparse to train a network to predict unseen test targets. The dashed blue line illustrates an alternative approach that involves training the network on cyclic loads and fine-tuning it using random walks. According to the results, transfer learning, in this case, cannot perform satisfactorily, referred to as negative transfer.

The success of learning a target task depends on its relation to the source task. In general, transfer learning between two tasks does not necessarily work in both directions. It has been demonstrated empirically by Rosenstein et al. (2005) that knowledge transfer across dissimilar domains results in performance degradation. However, addressing negative transfer remains an open question (Weiss et al., 2016) and is beyond the scope of this study. Cross-domain noise is increased when the source task has data sparsity or scarcity compared to the target task. Such noise reduces the ability to extract beneficial knowledge and may lead to negative transfer (Khan et al., 2024). In other words, transfer learning negatively affects the network performance on the target task.

The most promising results are obtained by training the model on random walk load paths (Section 4.1) and then fine-tuning it for cyclic loads, as illustrated by the dashed black line. With the transferred network, stress values can be predicted from sparse feature samples

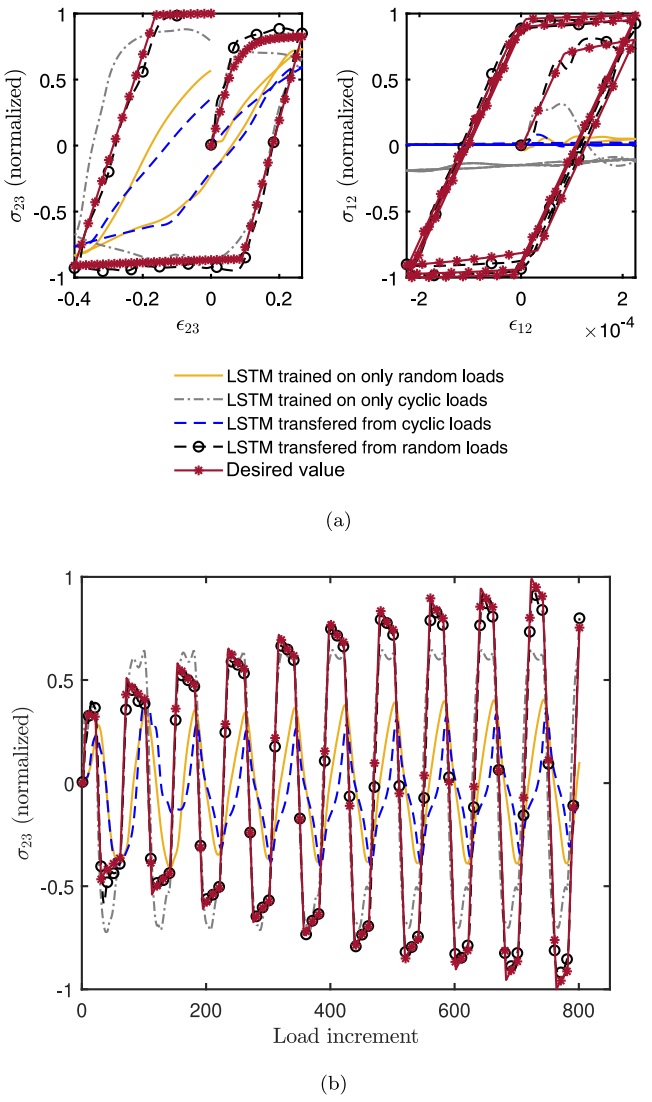

Fig. 10. LSTM network predictions on conventional cyclic loading compared with desired stress values (red solid lines with stars). Stress values are normalized between $[-1, 1]$. Black dashed lines indicate the candidate transferred model predictions. The blue dashed lines indicate predictions of a model trained initially on cyclic loads and then fine-tuned by random loads. The yellow and gray lines show the predictions of models that have been trained only on random and cyclic loads, respectively. (a-left) One cycle is pure out-of-plain shear, and (a-right) three cycles are pure in-plain shear. (b) ten cycles pure out-of-shear vs. load increments. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

through the entire loading increments not only for one-cycle tests (Fig. 10(a)-left) but also for multiple cycles (Figs. 10(a)-right and 10(b)). In some cases, the predictions deviate from the desired values at the beginning of loading, as shown in Fig. 10(a)-right. This behavior can be attributed to the nature of RNNs, which forget about the first part of a sequence when asked to predict the whole sequence at once, and not for forecasting and predicting one step ahead (Géron, 2022).

5. Conclusions

Woven composite laminates are used across a wide range of industries due to their cost-effective and automated manufacturing procedures. However, computationally efficient and industrially feasible models are still required to drive their use further. This study has investigated the capabilities of Recurrent Neural Networks (RNNs) combined with the transfer learning strategy in supervised learning of the

nonlinear behavior of woven composites, specifically when plasticity is present in the matrix phase. The present research has examined the predictive power of RNNs based on mean-field simulated data to be used as a fast and accurate surrogate model for mesoscale homogenization. A limitation of the present data-driven model stems from its reliance on low-fidelity MFH training data. Model predictions are also deterministic, unlike plain woven composite behavior in experiments, which is stochastic. To improve model accuracy, we intend to enhance future datasets with higher-fidelity homogenization techniques. Moreover, incorporating uncertainty sources into the model will also be a potential future development of the RNN model.

The RNN model incorporates two types of input features: (1) static input features sampled from a random design space for elastic fiber and elasto-plastic matrices with varying volume fractions; (2) sequential input features including six-dimensional time histories of the meso-scale strain tensor. The RNN model aims to predict six components of meso-scale homogenized stresses as outputs. Two distinct task data sets are considered. In the first one, named the source task, a random walk strategy ensures a diverse and comprehensive exploration of input strain path trajectories. Conventional cyclic strain loadings are considered in the second one, named the target task. The second task was found to be more challenging due to the presence of multiple zero-vectors in the sequential input features, known as feature sparsity. We have systematically examined a variety of GRU and LSTM architectures, along with several hyperparameters, in order to identify the most suitable model for predicting homogenized stresses in unseen test samples derived from random loads (source task). Having been exposed to random loads, the network has been retrained using a transfer learning paradigm and has demonstrated satisfactory performance in predicting stress components under conventional cyclic load conditions (target task).

In conclusion, this study provides evidence for transfer learning neural networks as an effective method for domain adaptation across diverse loading types and micro-structural constituent properties in material data sets. Our findings set a precedent for future investigations into knowledge transfer across varying-quality datasets. The results provide a robust foundation for future studies to tailor models for full-field simulations and even limited experimental data. Additionally, more relevant physics causing non-linearity, such as damage behavior, can be investigated by extending the data sets for training and validation. The insights gained from mean-field simulations can be used to develop models that are applicable in a wider range of settings and represent a significant opportunity for further advancement in this field.

CRediT authorship contribution statement

Ehsan Ghane: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Martin Fagerström:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Mohsen Mirkhalaf:** Writing – review & editing, Writing – original draft, Supervision, Resources, 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.

Data availability

Data will be made available on request.

Acknowledgments

Mohsen Mirkhalaf and Ehsan Ghane gratefully acknowledge financial support from the Swedish Research Council (VR grant: 2019-04715) and the University of Gothenburg, Sweden. Martin Fagerström is thankful for the support through Vinnova's strategic innovation programme LIGHTer, in particular via the project LIGHTer Academy Phase 3 (grant no. 2020-04526). Resources provided by Chalmers e-Commons enabled the computations. Ehsan Ghane is also grateful to Joel Magnusson, Iuri Rocha, and Hon Lam Cheung for their input and stimulating discussions, which enhanced this research.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.euromechsol.2024.105378.

References

Abueidda, Diab W., Koric, Seid, Sobh, Nahil A., Sehitoglu, Huseyin, 2021. Deep learning for plasticity and thermo-viscoplasticity. Int. J. Plast. 136, 102852.

Anon, 2016. Digimat-MF. https://www.e-xstream.com/products/digimat/tools?MF=1. (Accessed: 2 October 2016).

Anon, 2023. C3SE - About Vera. https://www.c3se.chalmers.se/documentation/intro-vera/slides/. (Accessed on 2 June 2023).

Author(s) of the Documentation, 2021. The MathWorks, inc. Deep learning toolbox: User's guide. MATLAB Documentation.

Benady, Antoine, Baranger, Emmanuel, Chamoin, Ludovic, 2023. NN-mCRE: a modified constitutive relation error framework for unsupervised learning of nonlinear state laws with physics-augmented neural networks.

Bessa, M.A., Bostanabad, R., Liu, Z., Hu, A., Apley, Daniel W., Brinson, C., Chen, W., Liu, Wing Kam, 2017. A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality. Comput. Methods Appl. Mech. Engrg. 320, 633–667.

Bishop, Christopher M., Nasrabadi, Nasser M., 2006. Pattern recognition and machine learning, vol. 4, Springer.

Callaghan, Jack S., Crump, Duncan, Nielsen, Anette S., Thomsen, O.T., Dulieu-Barton, Janice M., 2023. Quantitative full-field data fusion for evaluation of complex structures. Exp. Mech. 1–21.

Calleja Vázquez, Juan Manuel, Wu, Ling, Nguyen, Van-Dung, Noels, Ludovic, 2023. A micromechanical mean-field homogenization surrogate for the stochastic multiscale analysis of composite materials failure. Internat. J. Numer. Methods Engrg. 124 (23), 5200–5262.

Chan, William, Jaitly, Navdeep, Le, Quoc V., Vinyals, Oriol, 2015. Listen, attend and spell. arXiv preprint arXiv:1508.01211.

Cheung, Hon Lam, Mirkhalaf, Mohsen, 2024. A multi-fidelity data-driven model for highly accurate and computationally efficient modeling of short fiber composites. Compos. Sci. Technol. 246, 110359.

Cho, Kyunghyun, Van Merriënboer, Bart, Bahdanau, Dzmitry, Bengio, Yoshua, 2014a. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

Cho, Kyunghyun, Van Merriënboer, Bart, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, Bengio, Yoshua, 2014b. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Chung, Junyoung, Gulcehre, Caglar, Cho, KyungHyun, Bengio, Yoshua, 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Dekhovich, Aleksandr, Turan, O Taylan, Yi, Jiaxiang, Bessa, Miguel A, 2023. Cooperative data-driven modeling. Comput. Methods Appl. Mech. Engrg. 417, 116432.

Doghri, I., Tinel, L., 2005. Micromechanical modeling and computation of elasto-plastic materials reinforced with distributed-orientation fibers. Int. J. Plast. 21 (10), 1919–1940.

Doitrand, Aurélien, Fagiano, Christian, Hild, François, Chiaruttini, Vincent, Mavel, Anne, Hirsekorn, Martin, 2017. Mesoscale analysis of damage growth in woven composites. Composites A 96, 77–88.

Doitrand, Aurélien, Fagiano, Christian, Irisarri, F.-X., Hirsekorn, Martin, 2015. Comparison between voxel and consistent meso-scale models of woven composites. Composites A 73, 143–154.

Dornheim, Johannes, Morand, Lukas, Helm, Dirk, 2023. Neural networks for constitutive modeling–from universal function approximators to advanced models and the integration of physics. arXiv preprint arXiv:2302.14397.

El Said, Bassam, 2023. Predicting the non-linear response of composite materials using deep recurrent convolutional neural networks. Int. J. Solids Struct. 276, 112334.

Espadas-Escalante, Juan José, Isaksson, Per, 2019. Mesoscale analysis of the transverse cracking kinetics in woven composite laminates using a phase-field fracture theory. Eng. Fract. Mech. 216, 106523.

Friemann, J., Dashtbozorg, B., Fagerström, M., Mirkhalaf, S.M., 2023. A micromechanics-based recurrent neural networks model for path-dependent cyclic deformation of short fiber composites. Internat. J. Numer. Methods Engrg. 124 (10), 2292–2314.

Géron, Aurélien, 2022. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc..

Ghane, E., Fagerström, M., Mirkhalaf, S.M., 2023. A multiscale deep learning model for elastic properties of woven composites. Int. J. Solids Struct. 282, 112452.

Glorot, Xavier, Bengio, Yoshua, 2010. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, pp. 249–256.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian, 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1026–1034.

Heidenreich, Julian N., Bonatti, Colin, Mohr, Dirk, 2023. Transfer learning of recurrent neural network-based plasticity models. Internat. J. Numer. Methods Engrg..

Hinton, Geoffrey E., Roweis, Sam, 2002. Stochastic neighbor embedding. Adv. Neural Inf. Process. Syst. 15.

Hochreiter, Sepp, Schmidhuber, Jürgen, 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Huang, Dengpeng, Fuhg, Jan Niklas, Weißenfels, Christian, Wriggers, Peter, 2020. A machine learning based plasticity model using proper orthogonal decomposition. Comput. Methods Appl. Mech. Engrg. 365, 113008.

Jones, Reese E., Frankel, Ari L., Johnson, K.L., 2022. A neural ordinary differential equation framework for modeling inelastic stress response via internal state variables. J. Mach. Learn.r Model. Comput. 3 (3).

Jung, Jiyoung, Kim, Yongtae, Park, Jinkyoo, Ryu, Seunghwa, 2022. Transfer learning for enhancing the homogenization-theory-based prediction of elasto-plastic response of particle/short fiber-reinforced composites. Compos. Struct. 285, 115210.

Khan, Siraj, Yin, Pengshuai, Guo, Yuxin, Asim, Muhammad, Abd El-Latif, Ahmed A., 2024. Heterogeneous transfer learning: recent developments, applications, and challenges. Multimedia Tools Appl. 1–37.

Kingma, Diederik P., Ba, Jimmy, 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Li, Bin, Zhuang, Xiaoying, 2020. Multiscale computation on feedforward neural network and recurrent neural network. Front. Struct. Civil Eng. 14 (6), 1285–1298.

Lipton, Zachary C., Berkowitz, John, Elkan, Charles, 2015. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019.

Ma, Dayou, Giglio, Marco, Manes, Andrea, 2021. Analysis of mesoscale modelling strategies for woven composites. Mater. Des. Process. Commun. 3 (3), e145.

Maia, M.A., Rocha, I.B.C.M., Kerfriden, P., van der Meer, F.P., 2023. Physically recurrent neural networks for path-dependent heterogeneous materials: Embedding constitutive models in a data-driven surrogate. Comput. Methods Appl. Mech. Engrg. 407, 115934.

Mehlig, Bernhard, 2021. Machine Learning with Neural Networks: An Introduction for Scientists and Engineers, first ed. Cambridge University Press.

Mentges, N., Dashtbozorg, B., Mirkhalaf, S.M., 2021. A micromechanics-based artificial neural networks model for elastic properties of short fiber composites. Composites B 213, 108736.

Mirkhalaf, S.M., Pires, F.M. Andrade, Simoes, R., 2016. An elasto-viscoplastic constitutive model for polymers at finite strains: Formulation and computational aspects. Comput. Struct. 166, 60–74.

Mirkhalaf, S.M., Pires, F.M. Andrade, Simoes, Ricardo, 2017. Modelling of the post yield response of amorphous polymers under different stress states. Int. J. Plast. 88, 159–187.

Mirkhalaf, Mohsen, Rocha, Iuri, 2024. Micromechanics-based deep-learning for composites: Challenges and future perspectives. Eur. J. Mech. A Solids 105, 105242.

Mori, Tanaka, Tanaka, Kohichi, 1973. Average stress in matrix and average elastic energy of materials with misfitting inclusions. Acta Metall. 21 (5), 571–574.

Mozaffar, M., Bostanabad, R., Chen, W., Ehmann, K., Cao, J., Bessa, M.A., 2019. Deep learning predicts path-dependent plasticity. Proc. Natl. Acad. Sci. 116 (52), 26414–26420.

Murphy, Kevin P., 2012. Machine learning: a probabilistic perspective. MIT Press.

Pascanu, Razvan, Mikolov, Tomas, Bengio, Yoshua, 2013. On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning. Pmlr, pp. 1310–1318.

Renardy, Marissa, Joslyn, Louis R., Millar, Jess A., Kirschner, Denise E., 2021. To sobol or not to sobol? The effects of sampling schemes in systems biology applications. Math. Biosci. 337, 108593.

Rocha, I.B.C.M., Kerfriden, Pierre, van Der Meer, F.P., 2021. On-the-fly construction of surrogate constitutive models for concurrent multiscale mechanical analysis through probabilistic machine learning. J. Comput. Phys.: X 9, 100083.

Rosenkranz, Max, Kalina, Karl A., Brummund, Jörg, Kästner, Markus, 2023. A comparative study on different neural network architectures to model inelasticity. arXiv preprint arXiv:2303.03402.

Rosenstein, Michael T., Marx, Zvika, Kaelbling, Leslie Pack, Dietterich, Thomas G., 2005. To transfer or not to transfer. In: NIPS 2005 Workshop on Transfer Learning, Vol. 898, No. 3.

Saltelli, Andrea, Annoni, Paola, Azzini, Ivano, Campolongo, Francesca, Ratto, Marco, Tarantola, Stefano, 2010. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Comput. Phys. Comm. 181 (2), 259–270.

Simo, Juan C., Hughes, Thomas J.R., 2006. Computational inelasticity, vol. 7, Springer Science & Business Media.

Spilker, Kevin, Nguyen, Van-Dung, Wu, Ling, Noels, Ludovic, 2023. Three-scale bridging for woven composites using homogenization techniques. Eur. J. Mech. A Solids 100, 104974.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, Salakhutdinov, Ruslan, 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (56), 1929–1958.

Sutskever, Ilya, Vinyals, Oriol, Le, Quoc V., 2014. Sequence to sequence learning with neural networks. Adv. Neural Inf. Process. Syst. 27.

Udhayaraman, R., Mulay, Shantanu S., 2019. Multi-scale damage framework for textile composites: Application to plain woven composite. Eur. J. Mech. A Solids 77, 103809.

van der Maaten, Laurens, Hinton, Geoffrey, 2008. Visualizing data using t-SNE. J. Mach. Learn. Res. 9 (86), 2579–2605.

Vlassis, Nikolaos N., Sun, WaiChing, 2023. Geometric learning for computational mechanics part II: Graph embedding for interpretable multiscale plasticity. Comput. Methods Appl. Mech. Engrg. 404, 115768.

Weiss, Karl, Khoshgoftaar, Taghi M., Wang, DingDing, 2016. A survey of transfer learning. J. Big data 3, 1–40.

Willmott, Cort J., Matsuura, Kenji, 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Res. 30 (1), 79–82.

Wu, Ling, Adam, Laurent, Noels, Ludovic, 2021a. Micro-mechanics and data-driven based reduced order models for multi-scale analyses of woven composites. Compos. Struct. 270, 114058.

Wu, Ling, Noels, Ludovic, 2022. Recurrent neural networks (RNNs) with dimensionality reduction and break down in computational mechanics; application to multi-scale localization step. Comput. Methods Appl. Mech. Engrg. 390, 114476.

Wu, Ling, Zhang, Tianyu, Maillard, Etienne, Adam, Laurent, Martiny, Philippe, Noels, Ludovic, 2021b. Per-phase spatial correlated damage models of UD fibre reinforced composites using mean-field homogenisation; applications to notched laminate failure and yarn failure of plain woven composites. Comput. Struct. 257, 106650.

Xu, Jian, Lomov, Stepan Vladimirovitch, Verpoest, Ignace, Daggumati, Subbareddy, Van Paepegem, Wim, Degrieck, Joris, 2015. A progressive damage model of textile composites on meso-scale using finite element method: Fatigue damage analysis. Comput. Struct. 152, 96–112.

Yang, Qiang, Zhang, Yu, Dai, Wenyuan, Pan, Sinno Jialin, 2020. Transfer learning. Cambridge University Press.