

# 장단기 기억 신경망을 이용한 ABS 판재의 점탄성 모델링

도안 응웬 부<sup>1</sup>, 김지훈<sup>2, #</sup>

## Modeling Viscoelasticity of Acrylonitrile-butadiene Styrene Sheets using Long-short Term Memory Models

Nguyen Vu Doan and Ji Hoon Kim

(Received August 13, 2024 / Revised September 24, 2024 / Accepted September 24, 2024)

### Abstract

In this paper, the capabilities of recurrent neural networks (RNNs) to describe the viscoelastic properties of acrylonitrile-butadiene styrene (ABS) are investigated. The RNN model was trained using one-dimensional strains and corresponding stress data generated by the finite element method. The optimal model was then employed to predict the viscoelastic behavior of unseen test data. Furthermore, the viscoelastic-based RNN model was tested for extrapolation using other types of strain and corresponding stress data beyond the training set. The agreement between the predicted and actual stresses demonstrates the robust performance of the trained RNN model in predicting different types of strain inputs for larger strain tests, despite being trained only with step strain inputs. Therefore, the use of RNNs can be considered a viable alternative to conventional models for predicting viscoelastic behavior.

**Keywords** : Deep learning, Recurrent neural networks, History-dependent material, Viscoelasticity, Generalized Maxwell model

## 1. Introduction

Modeling history-dependent materials, which encompass a wide range of engineering materials exhibiting irreversible deformation behavior and energy dissipation, has been a significant challenge in computational mechanics. Conventional approaches have relied on the use of constitutive laws that incorporate internal state variables, such as strain and stress, to track the deformation history of materials, including viscoplasticity and viscoelasticity [1].

These classical methods, while successful, are complex and computationally intensive. However, the emergence of big data and advancements in computational power have enabled data-driven methods, particularly machine learning

and deep learning (DL), to emerge as powerful alternatives. Deep learning is a subfield of machine learning, with neural networks forming the foundational structure of deep learning algorithms. By leveraging deep neural networks to learn complex relationships from large datasets, DL has revolutionized numerous fields, including material discovery [2-4] and material joining processes [5, 6]. However, in tasks involving sequential data with long-term dependencies, such as natural language processing and music generation, conventional deep neural networks exhibit suboptimal performance. It is necessary to investigate techniques capable of handling this type of data.

Recurrent neural networks (RNNs) have been developed and implemented to address these challenges, demonstrating remarkable success in these domains [7]. RNNs have an intrinsic computational mechanism that uses hidden states to capture and transmit sequence information across time

1. 부산대학교 기계공학부, 박사과정

2. 부산대학교 기계공학부, 교수

#Corresponding Author: J.H Kim, School of Mechanical Engineering,  
Pusan National University, E-mail: kimjh@pusan.ac.kr,  
ORCID: 0000-0001-9334-0503

steps, where the output depends on preceding evaluations. This characteristic of RNNs closely aligns with the modeling requirements of history-dependent materials, making them naturally suited for modeling various history-dependent material models, including plasticity, viscoplasticity, and, more recently, viscoelastic materials, whose current state depends on their deformation history. Mozaffar et al. [8] conducted the pioneering study on employing RNNs for history-dependent plasticity. Subsequent research has also investigated the application of RNNs in plasticity or elastoplasticity modeling [9, 10] and viscoplasticity modeling [11]. Chen [12] demonstrated the effectiveness of RNN models in computing the viscoelastic response for three-dimensional models in the case of infinite strain and tested the ability to extrapolate the RNN-viscoelasticity model.

In this paper, the feasibility of using RNNs to describe viscoelasticity is investigated, with a particular focus on long short-term memory (LSTM) networks. A comprehensive dataset, including various strain inputs and their corresponding stresses derived from the Prony series configured from the relaxation test data of an acrylonitrile-butadiene styrene (ABS) sheet, is generated. The RNN-viscoelastic model is trained using this dataset, and its performance on both seen and unseen data is evaluated. Furthermore, the generalization capability of RNN models is presented for demonstrating their potential to extrapolate beyond the training data.

## 2. Modeling

### 2.1 Constitutive model of viscoelastic materials

The constitutive model for viscoelastic materials aims to capture this dual behavior by describing the relationship between stress ( $\sigma$ ), strain ( $\epsilon$ ), and time. Similar to other history-dependent materials, the dependence of current stress on previous state values, recursively linked to earlier states, is a key characteristic. This differentiation impacts the performance requirements in developing RNN models and their subsequent ability to generalize when addressing viscoelasticity issues. A constitutive relation for an initial strain input can be formulated as follows:

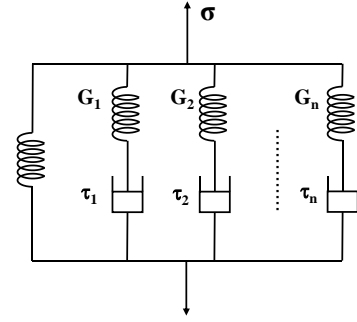


Fig. 1 Schematic of generalized Maxwell model

$$\sigma(t) = G(t) \cdot \epsilon_o \quad (1)$$

where  $\epsilon_o$  is the initial strain and  $G(t)$  is a relaxation function. In a generalized Maxwell solid (Fig. 1), the relaxation function is expressed using a Prony series as follows:

$$G(t) = G_0 \cdot \left(1 - \sum_{i=1}^n g_i \cdot (1 - e^{-t/\tau_i})\right) \quad (2)$$

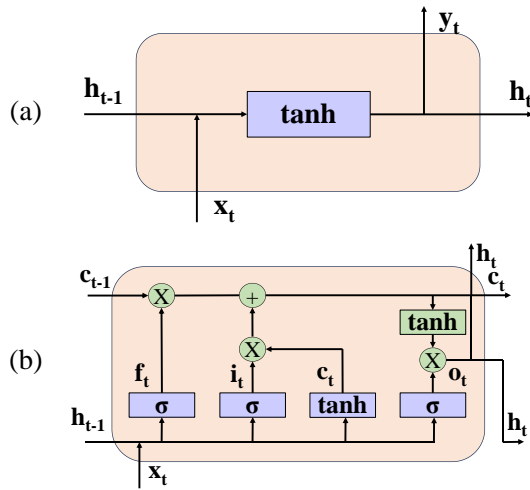
where,  $g_i$  is the  $i$ -th Prony constant ( $i = 1, 2, \dots$ ),  $\tau_i$  is the  $i$ -th relaxation time ( $i = 1, 2, \dots$ ), and  $G_0$  is the instantaneous modulus.

### 2.2 RNN

RNNs are designed to process sequential data with hidden state variables to store historical information. A basic RNN unit updates its hidden state and output as illustrated in Fig. 2(a):

$$\begin{aligned} h_{(t)} &= \sigma_a(W_{hh}x_{(t-1)} + W_{xh}x_{(t)} + b_h) \\ \hat{y} &= \sigma_a(W_0h_t + b_0) \end{aligned} \quad (3)$$

In Eq. (3),  $\sigma_a(\cdot)$  is the activation function (e.g., sigmoid or hyperbolic tangent). The weights  $W_{hh}$ ,  $W_{xh}$ ,  $W_0$  and bias  $b_h$ ,  $b_0$  are trainable parameters obtained during the training process. The loss function quantifies the discrepancy between the predictions and the actual outputs. The primary objective of the model is to iteratively adjust these weights and biases to minimize the loss function, thereby enhancing the accuracy of its predictions. For modeling history-dependent materials, the many-to-many architecture is used for external inputs (e.g., strains) and outputs (e.g., stress responses). The inherent similarity between the recursive



**Fig. 2 Structure of RNN: (a) basic RNN unit; (b) LSTM unit**

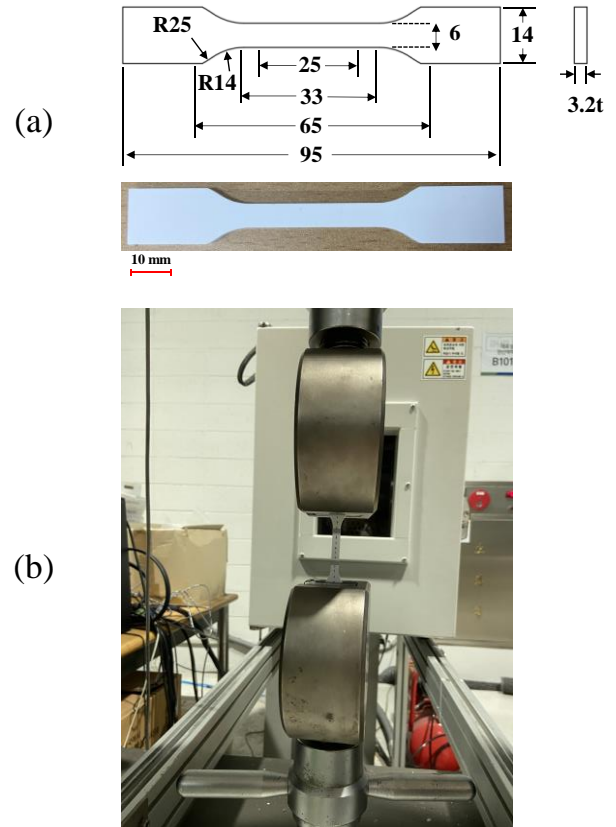
nature of stress update algorithms in viscoelastic material models and the sequential data processing of RNNs makes RNNs particularly suitable for this application.

However, the basic RNNs face challenges such as vanishing gradients, making them less effective for long-term dependencies. Therefore, advanced units like LSTM [13], illustrated in Fig. 2(b), and gated recurrent unit (GRU) [7, 10] have been developed to address these issues. LSTM networks have a more complex architecture and a greater number of model parameters compared to standard RNNs, allowing for more effective learning of the viscoelastic constitutive law.

The LSTM module enhances the basic RNN by adding a cell state vector  $c_{\langle t \rangle}$  and multiple gates to regulate the series of information. The operation in an LSTM can be described as follow:

$$\begin{aligned}
 f_{\langle t \rangle} &= \sigma_s(W_{xf}x_{\langle t \rangle} + W_{hf}h_{\langle t-1 \rangle} + b_f) \\
 i_{\langle t \rangle} &= \sigma_s(W_{xi}x_{\langle t \rangle} + W_{hi}h_{\langle t-1 \rangle} + b_i) \\
 o_{\langle t \rangle} &= \sigma_s(W_{xo}x_{\langle t \rangle} + W_{ho}h_{\langle t-1 \rangle} + b_o) \\
 \tilde{c}_{\langle t \rangle} &= \tanh(W_{xc}x_{\langle t \rangle} + W_{hc}h_{\langle t-1 \rangle} + b_c) \\
 c_{\langle t \rangle} &= f_{\langle t \rangle} * c_{\langle t-1 \rangle} + i_{\langle t \rangle} * \tilde{c}_{\langle t \rangle} \\
 h_{\langle t \rangle} &= o_{\langle t \rangle} * \tanh(c_{\langle t \rangle})
 \end{aligned} \quad (4)$$

where  $f_{\langle t \rangle}$ ,  $i_{\langle t \rangle}$ , and  $o_{\langle t \rangle}$  are the activated vectors for the forget, input/update, and output gates, respectively,  $\sigma_s(\cdot)$  is the sigmoid function,  $c_{\langle t \rangle}$  and  $c_{\langle t-1 \rangle}$  are the new candidate



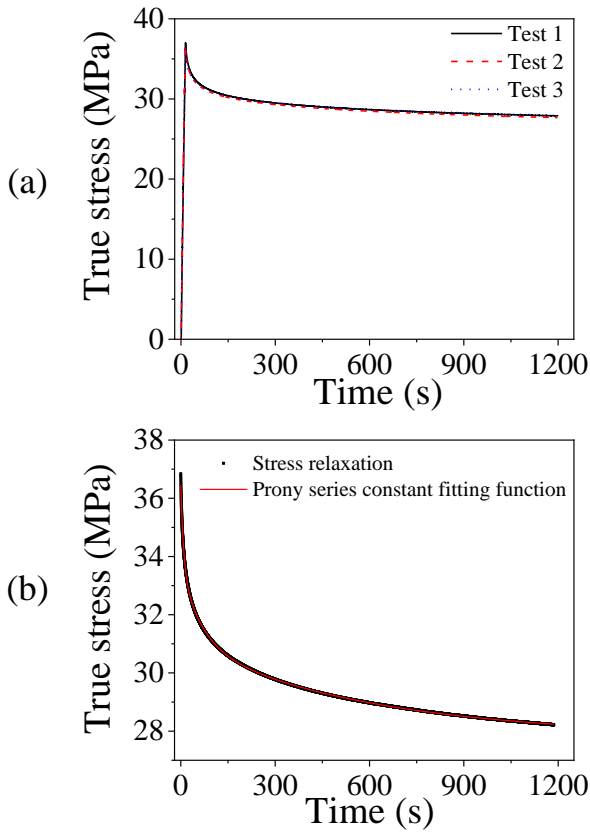
**Fig. 3 (a) ABS specimen; (b) stress relaxation experimental setup**

memory cell and cell state vectors, respectively. The variables  $W_{xf}$ ,  $W_{hf}$ ,  $W_{xi}$ ,  $W_{hi}$ ,  $W_{xo}$ ,  $W_{ho}$ ,  $W_{xc}$ ,  $W_{hc}$ ,  $b_f$ ,  $b_i$ ,  $b_o$ , and  $b_c$  are the trainable weights and biases. The symbol \* indicates element-wise multiplication. 1D strains and corresponding stresses are used as input sequences. The LSTM model was developed and implemented using Python. This implementation was conducted within the open-source web application Jupyter Notebook.

### 3. Experiment and Finite Element Model

#### 3.1 Stress relaxation test for ABS material

An ABS sheet was cut into specimens that followed ASTM D638 - type IV, as shown in Fig. 3(a). The specimen was then clamped in the grip of a universal testing machine (Unitech-T, RB 301, R&B Inc) for the stress relaxation test at room temperature. The specimen was elongated at a rate of 5 mm/min with a gauge length of 25 mm, corresponding



**Fig. 4 (a) Relaxation test result of the ABS material; (b) Curve fitting for stress relaxation data**

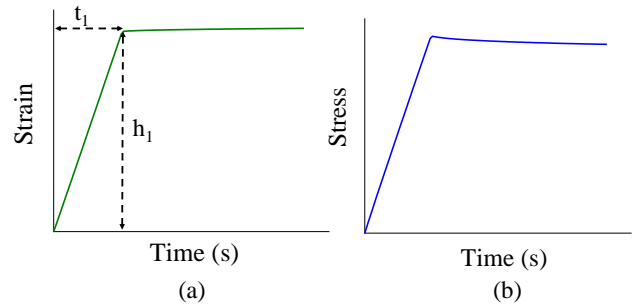
to a strain rate of  $0.00333 \text{ s}^{-1}$ , until it reached 80% of the initial yield strength (45 MPa) of the ABS material. This stress level was then maintained for a duration of 1200 s. The measured stress of the specimen is shown in Fig. 4(a). The loading phase response is ignored, while the data collected during the period of constant displacement or constant load was utilized to determine the Prony series parameters. To better capture the behavior of the material, the stress relaxation data were then fitted with 7 Prony constants:  $G_0$ ,  $g_1$ ,  $g_2$ ,  $g_3$ ,  $\tau_1$ ,  $\tau_2$ , and  $\tau_3$ , of the relaxation function (2) using non-linear regression. The fitting curve of the stress relaxation data is shown in Fig. 4(b), and the Prony series values are shown in Table 1.

### 3.2 Finite element model

The values of the Prony terms were imported into Abaqus commercial software for data generation. To obtain the stress-strain response for ABS material at room temperature, the deformation of a two-dimensional model

**Table 1 Prony series constant of ABS**

$G_0$ [GPa]	$g_i$	$\tau_i$
36.5	0.07706	8.65219
	0.07073	63.75375
	0.08918	577.58283



**Fig. 5 (a) Strain sequences; (b) corresponding stress under loading conditions of single element**

**Table 2 Configuration of parameters for strain data generation**

Variables	Specified range
$h_1$	$[-0.01, 0.01]$
$t_1$ [s]	$[1, 900]$

(1 mm x 1 mm) was captured over 1000 s. This model employed a 4-node shell element with reduced integration (S4R). The Poisson's ratio of ABS material used for simulation was 0.37. Subsequently, the logarithmic strain and corresponding stress values were extracted from Abaqus to serve as training data. The strain input for boundary conditions and corresponding stress are shown in Fig. 5, which can cover a broad range of viscoelastic responses.

## 4. Results

### 4.1 Data generation

The strain sequences were produced over the time interval from 0 to 1000 s with 100 discrete time steps in Abaqus. For each case of strain, the step height  $h_1$  and the time interval  $t_1$  were sampled within specified ranges to ensure a comprehensive dataset, as shown in Table 2.

**Table 3 MSE and MAE results for different RNN model during training, testing**

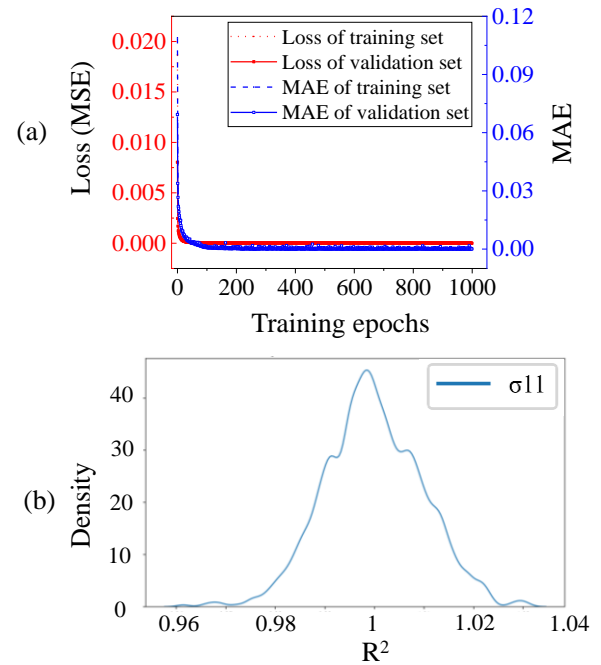
Number of LSTM layer	Number of hidden units (neurons)	MSE value	MAE value
1	5	$1.5304 \times 10^{-7}$	$3.6077 \times 10^{-4}$
1	10	$5.228 \times 10^{-9}$	$4.7247 \times 10^{-5}$
1	20	$7.1889 \times 10^{-9}$	$6.7089 \times 10^{-5}$
1	50	$9.8187 \times 10^{-9}$	$7.3158 \times 10^{-5}$
2	20	$1.2348 \times 10^{-7}$	$2.5506 \times 10^{-4}$

In total, 90 uniform points were sampled for each step height  $h_1$  and time interval  $t_1$ . This sampling process yielded 8100 strain samples. The associated stresses were calculated using stress update algorithms from the finite element model, culminating in 8100 stress-strain sequences.

#### 4.2 RNN model development

The generated dataset was divided into a training set (72%), a validation set (18%), and a test set (10%). During the training phase, the mean squared error (MSE) between predictions and actual values was used as the loss function. Additionally, the mean absolute error (MAE) was employed for evaluation.

In our work, various RNN models with different numbers of LSTM layers and hidden units were systematically assessed. The RNN models were trained using the training dataset with 1000 epochs and a batch size of 64 and the MSE and MAE were evaluated for an unseen test dataset to determine the optimal model. To obtain the best RNN model for predicting viscoelastic behavior, RNN models with different numbers of LSTM layers and hidden units were tested. In the case of one LSTM layer, different numbers of hidden units were used: 5 neurons (model 1), 10 neurons (current model), 20 neurons (model 2), and 50 neurons (model 3). A model with 2 LSTM layers and 20 hidden units (model 4) was also tested. The MSE and MAE values obtained for each model are illustrated in Table 3. As shown in Table 3, the model with 1 LSTM layer and 10 neurons yielded the lowest MSE and MAE values,  $5.228 \times 10^{-9}$  and  $4.7247 \times 10^{-5}$ , respectively. Therefore, this model was used for the subsequent analysis.



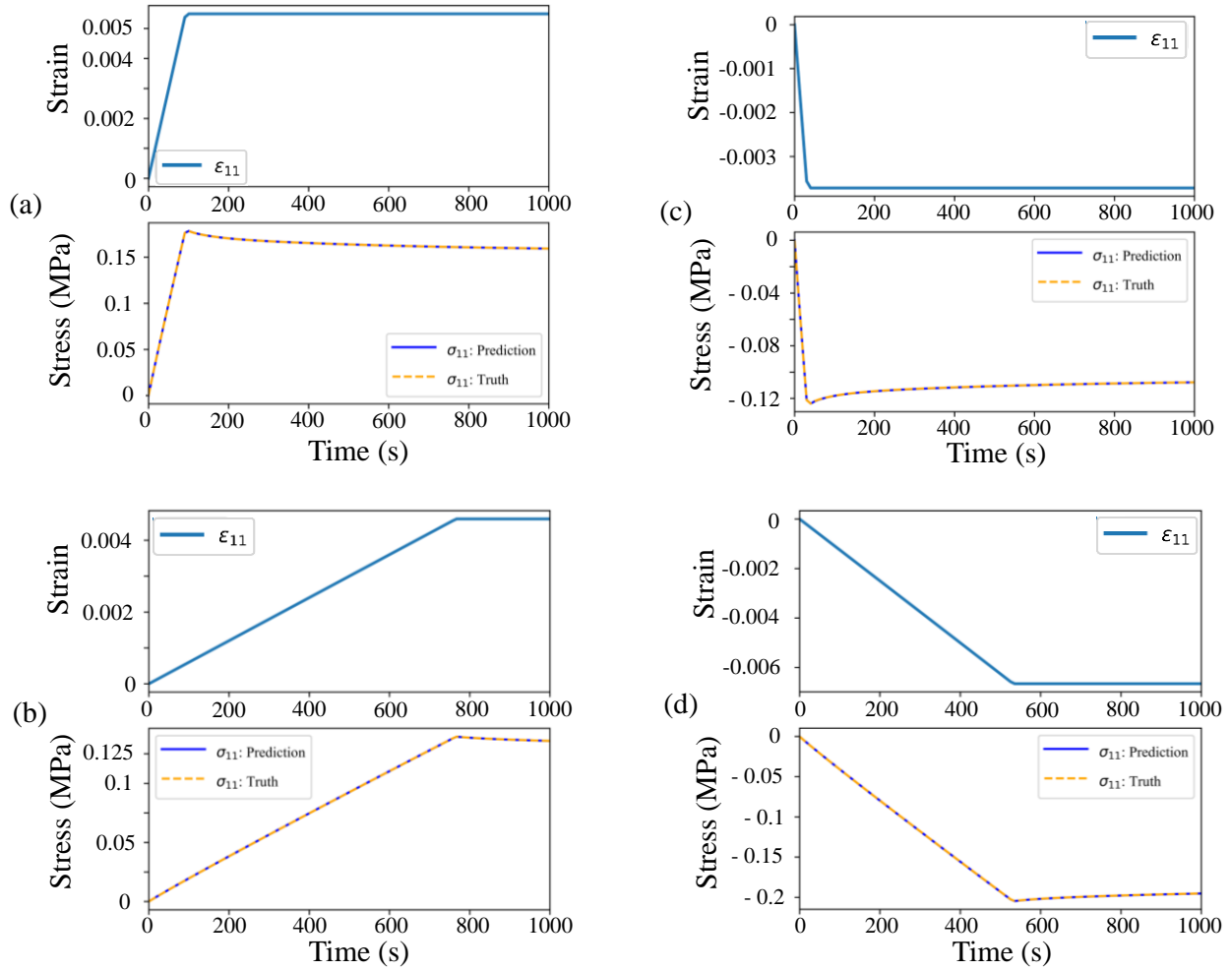
**Fig. 6 Evaluation of the constructed model: (a) the progression of loss function and mean absolute values throughout the training phase; (b) the distribution of the  $R^2$  correlation score for stress prediction on the test dataset**

#### 4.3 Training–testing

Fig. 6(a) illustrates the progression of loss function and MAE throughout the training process of the training and validation datasets. The training process shows a stable and robust model, as evidenced by the agreement between the training and validation performance of the dataset.

After the model was verified, its predictive ability was tested using the unseen test dataset. A total of 810 strain and corresponding stress sequences were applied to make predictions. Consequently, 810 unidirectional stress pairs of true and predicted values were obtained. An  $R^2$  correlation score distribution analysis for these stress pairs was conducted. Testing the model on unseen data revealed a high correlation between predicted and true stress values, with scores close to 1 indicating high accuracy.

Four random cases were generated from constructed RNN model to predict the response and compared with the corresponding true viscoelastic response, as shown in Fig. 7. As can be seen, the random cases from the test set demonstrated the model's accuracy in predicting



**Fig. 7** Evaluation of the RNN model for four randomly chosen cases, (a), (b), (c), and (d), from the test dataset. In each case, the upper subfigure displays the strain, while the lower subfigure presents the predicted and actual stress values

viscoelastic responses across a broad range of strain.

#### 4.4 Extrapolation tests

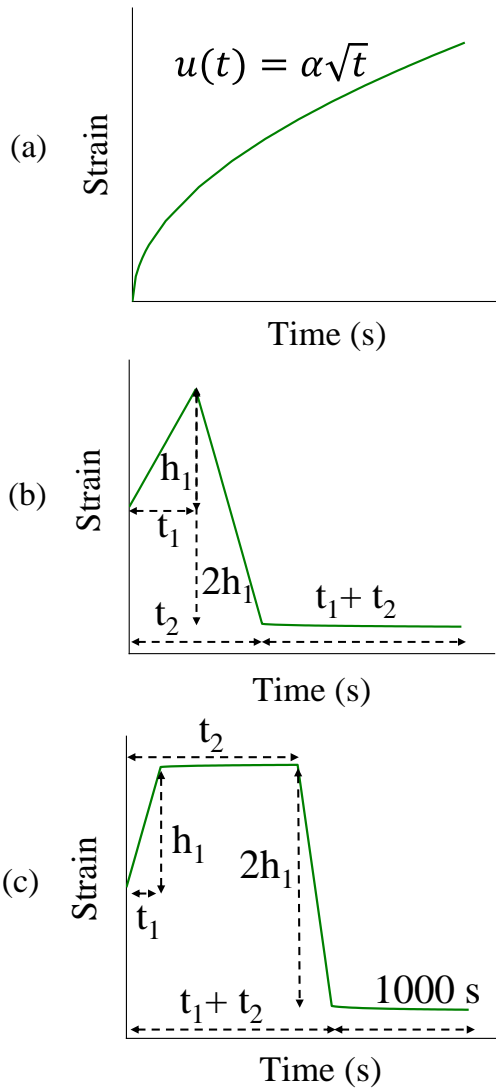
To assess the ability of the RNN model to extrapolate, several tests were conducted with strain sequences beyond the training range and different strain sequences, such as the nonlinear strain input with  $\epsilon_{11}(t) = \alpha \sqrt{t}$  strain reversal, and two-step stress relaxation sequences, as illustrated in Fig. 8.  $h_1$  indicates the normal strain in the X direction experienced by the two-dimensional model, while  $\alpha$  serves as a scaling factor to define the time-dependence of  $\epsilon_{11}(t)$ . The parameters  $t_1$  and  $t_2$  represent the time intervals used to adjust the displacement magnitude in the case of step strains. The extrapolation data were processed in a manner

**Table 4** Configuration of parameters for strain generation in extrapolation tests

Condition	Variable	Value
Square root strain	$\alpha$ [ $s^{-1/2}$ ]	$1.5 \times 10^{-4}$
Strain reversal	$h_1$	0.05
	$t_1$ [s]	200
Two-step stress relaxation	$t_2$ [s]	400
	$h_1$	0.05
	$t_1$ [s]	100
	$t_2$ [s]	500

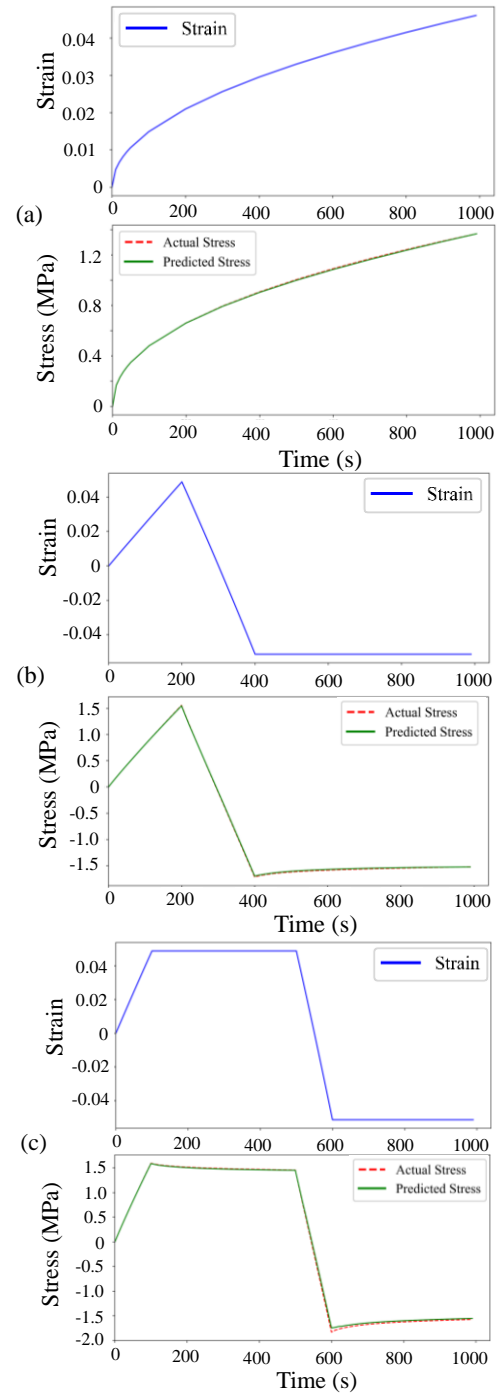
similar to the training data. The parameters for these tests are detailed in Table 4.

The RNN model developed from the initial training was



**Fig. 8 Strain sequences under loading conditions of single element for extrapolation tests: (a) square root strain; (b) strain reversal; (c) two-step stress relaxation**

later used to predict entirely new types of data sequences. Fig. 9 presents the strain and corresponding stress plots for the first test involving a limited range of extrapolation, with strain sequences shown in the upper subplots. The lower subplots compare the predicted stress responses with the actual values. Overall, all three cases showed strong performance in extrapolation. The RNN model accurately predicted the responses for previously unseen square root, strain reversal, and two-step stress relaxation sequences, demonstrating good generalization within a certain range.



**Fig. 9 Extrapolation test evaluating the constructed RNN-viscoelasticity model using three previously unseen strain sequences that are similar in type and beyond the range of those in the training dataset: (a) square root strain; (b) strain reversal; (c) two-step stress relaxation. For each scenario, the upper subfigure illustrates the strain sequence, and the lower subfigure displays the predicted and actual stress values**

## 5. Conclusion and future work

In this study, the feasibility of RNNs to describe viscoelastic behavior was investigated. This approach is driven by the intrinsic similarity between the algorithmic modeling of viscoelastic materials and RNNs. The performance of RNN models was assessed through a one-dimensional FEM dataset utilizing step strain conditions. The developed RNN models demonstrated good performance when tested on unseen data, despite the strain series and corresponding stresses exceeding the magnitude range present in the training data. Moreover, the RNN-viscoelasticity model exhibited a significant degree of generalization, performing well on square root, strain reversal, and two-step stress relaxation sequences, even though such patterns were absent in the training data.

This study has shown that RNNs can effectively learn the constitutive laws of viscoelasticity, suggesting that DL techniques hold great promise as a new paradigm in computational mechanics. For a simple one-dimensional problem and specified stress-strain data sequence generated from FEM, a single LSTM with up to 50 hidden neurons may suffice to achieve good extrapolation capabilities. For practical applications, a model with longer time sequences will be more suitable and flexible for predicting responses of viscoelastic materials in the real world. More complex problems require a laborious trial-and-error approach and extensive hyperparameter tuning to develop an effective model. Optimizing the number of sequence data points and understanding their underlying mechanisms for training effort is crucial to achieving good extrapolation ability in RNN models. Therefore, a method for data generation and an RNN architecture specifically designed to address viscoelastic behavior is necessary for wider application of deep learning in computational mechanics of viscoelastic materials.

## Acknowledgement

This work was supported by a 2-Year Research Grant of Pusan National University.

## REFERENCES

- [1] G. A. Holzapfel, 2002, *Nonlinear Solid Mechanics: A Continuum Approach for Engineering Science*, Wiley, New York
- [2] G. Rafael, A. Jorge, Timothy D. Hirzel, D. David, D. Maclaurin, B. Martin A., H.S. Chae, E. Markus, D.G. Ha, T. Wu, M. Georgios, S. Jeon, H. Kang, H. Miyazaki, M. Numata, S. Kim, W. Huang, S.I. Hong, M. Baldo, Ryan P. Adams, A. Alán, 2016, Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach, *Nature Materials*, Vol. 15, No. 10, pp. 1120-1127. <https://doi.org/10.1038/nmat4717>
- [3] G. Chen, Z. Shen, A. Lyer, U.F. Ghumman, S. Tang, J. Bi, W. Chen, Y. Li, 2020, Machine-Learning-Assisted De Novo Design of Organic Molecules and Polymers: Opportunities and Challenges, *Polymers*, Vol. 12, No. 1, p. 163. <https://doi.org/10.3390/polym12010163>
- [4] K. Guo, Z. Yang, C. Yu, M.J. Buehler, 2021, Artificial intelligence and machine learning in design of mechanical materials, *Materials Horizons*, Vol. 8, No. 4, pp. 1153-1172. <https://doi.org/10.1039/D0MH01451F>
- [5] G. Chen, Z. Shen, Y. Li, 2020, A machine-learning-assisted study of the permeability of small drug-like molecules across lipid membranes, *Physical Chemistry Chemical Physics*, Vol. 22, No. 35, pp. 19687-19696. <https://doi.org/10.1039/D0CP03243C>
- [6] D.S. Jo, P. Kahhal, J.H. Kim, 2023, Optimization of Friction Stir Spot Welding Process Using Bonding Criterion and Artificial Neural Network, *Materials*, Vol. 16, No. 10, p. 3757. <https://doi.org/10.3390/ma16103757>
- [7] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, 2014, Learning phrase representations using RNN encoder-decoder for statistical machine translation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, pp. 1724-1734. 10.3115/v1/D14-1179



- [8] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, M.A. Bessa, 2019, Deep learning predicts path-dependent plasticity, *Proceedings of the National Academy of Sciences (PNAS)*, Vol. 116, No. 52, pp. 26414-26420.  
<https://doi.org/10.1073/pnas.1911815116>
- [9] M.B. Gorji, M. Mozaffar, J.N. Heidenreich, J. Cao, D. Mohr, 2020, On the potential of recurrent neural networks for modeling path dependent plasticity, *Journal of the Mechanics and Physics of Solids*, Vol. 143, p. 103972.  
<https://doi.org/10.1016/j.jmps.2020.103972>
- [10] L. Wu, V.D. Nguyen, N.G. Kilingar, L. Noels, 2020, A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths, *Computer Methods in Applied Mechanics and Engineering*, Vol. 369, p. 113234.  
<https://doi.org/10.1016/j.cma.2020.113234>
- [11] F. Ghavamian, A. Simone, 2019, Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network, *Computer Methods in Applied Mechanics and Engineering*, Vol. 357, p. 112594.  
<https://doi.org/10.1016/j.cma.2019.112594>
- [12] G. Chen, 2021, Recurrent neural networks (RNNs) learn the constitutive law of viscoelasticity, *Computational Mechanics*, Vol. 67, No. 3, pp. 1009-1019. <https://doi.org/10.1007/s00466-021-01981-y>
- [13] S. Hochreiter, J. Schmidhuber, 1997, Long Short-Term Memory, *Neural Computation*, Vol. 9, No. 8, pp. 1735-1780.  
<https://doi.org/10.1162/neco.1997.9.8.1735>