Proc. of the 9th International Symposium on Geotechnical Safety and Risk (ISGSR)
Edited by Zhongqiang Liu, Jian Dai and Kate Robinson
Published by Research Publishing, Singapore. ©2025 The Author(s), CC-BY 4.0 licence.
doi: 10.3850/GRF-25280825_isgsr-095-P371-cd

AN IMPROVED RECURSIVE FEED FORWARD NEURAL NETWORK BASED SAND CONSTITUTIVE MODELLING

Toiba Noor

Department of Civil Engineering, Indian Institute of Technology Delhi, India. E-mail: cez217542@iitd.ac.in

G. V. Ramana

Department of Civil Engineering, Indian Institute of Technology Delhi, India. E-mail:ramana@iitd.ac.in

Rajdip Nayek

Department of Applied Mechanics, Indian Institute of Technology Delhi, India. E-mail: rajdipn@iitd.ac.in

Data-driven predictive modeling offers a powerful alternative to traditional physics-based models for understanding the complex constitutive behavior of sands. This work introduces a novel deep learning framework based on recursive feedforward neural networks (R-FFNNs) to capture the inherent temporal dependencies and nonlinear relationships within experimental and simulated data. The proposed R-FFNN model is rigorously evaluated against other state-of-the-art deep learning architectures, including feedforward neural networks (FFNNs), long short-term memory (LSTM) networks, bidirectional LSTMs (Bi-LSTMs), and gated recurrent units (GRUs). Through comprehensive training, validation, and testing, it is shown that the R-FFNN model demonstrates superior performance in predicting the evolution of stress and strain states in sand.

Keywords: Soil constitutive modeling, Recursive feedforward neural network, Deep learning, Granular soils.

1. Introduction

Modeling the constitutive behavior of sands remains a challenge in geotechnical engineering due to their complex response. Traditional physics-based models, from elasticity-based formulations to advanced elastoplastic frameworks, often struggle with the trade-off between accuracy and practicality—simpler models neglect key behaviors, while complex ones introduce numerous, sometimes non-physical, parameters that complicate calibration and interpretation. Deep learning (DL) offers an alternative by leveraging extensive experimental datasets to learn intricate strain-stress-void ratio relationships without predefined physical formulations.

Early applications using feedforward neural networks (FFNNs) for stress-strain modeling of sand [1] failed to account for data's sequential nature, leading to error accumulation. Recursive FFNNs (R-FFNNs) [2] addressed this by incorporating past predictions as inputs(Fig. 1b). However, traditional training methods for R-FFNNs can still lead to subpar predictions due to the compounding of errors over time.

(a) FFNN (b) Recursive-FFNN

Figure 1: Difference between working of FFNNs and Recursive-FFNNs.

This study introduces a novel training approach for R-FFNNs using a sliding window technique to enhance multi-step prediction accuracy by capturing short-term dependencies. The proposed model is rigorously compared against state-of-the-art deep learning architectures, including FFNN, LSTM, Bi-LSTM, and GRU [3,4]. Extensive training, validation, and testing onexperimental data demonstrate its superior performance, particularly in extrapolating to unseen conditions. Thestudygives a robust data-driven framework for predicting the constitutive behavior of sands, with applications in geotechnical engineering and computational mechanics.Its integration into FE analysis enables efficient numerical simulations of complex geotechnical problems, capturing soil behavior without relying on conventional constitutive theories [5].

2. Problem Setup

A DL-based constitutive model is developed using$M$strain-controlled consolidated drained (CD) triaxial tests, capturing the stress-strain-volume change response of sand under controlled loading. Each sample is consolidated under constant confining pressure σ_3. Quasi-static axial strain increments$d\epsilon$are applied and vertical stresses $\sigma_1$and volume change are recorded at each step. The mean effective stress $p$and deviatoric stress $q$are computed as

$p = (\sigma_1 + 2\sigma_3)/3$and $q = \sigma_1 - \sigma_3$. The void ratio $e$is calculated from the initial void ratio and volume change data. For test m, the soil state at time-step $t$is given as$\boldsymbol{s}_t^{(m)} = \left[p_t^{(m)}, q_t^{(m)}, e_t^{(m)}\right]^T$, governed by experimental constants$\boldsymbol{\theta}^{(m)} = \left[\sigma_3^{(m)}, e_0^{(m)}\right]^T$. Across all tests, the dataset is structured as follows:

$$\mathcal{D} = \left(\boldsymbol{\theta}^{(m)}, \left(p_0^{(m)}, q_0^{(m)}, e_0^{(m)}\right), \left(p_t^{(m)}, q_t^{(m)}, e_t^{(m)}, \epsilon_t^{(m)}, \mathrm{d}\epsilon_t^{(m)}\right)_{t=1}^{N}\right)_{m=1}^{M} \tag{1}$$

A deterministic map f predicts estimates of the next state from the previous state estimate and other inputs:

$$\boldsymbol{s}_t^{(m)} = f\left(\boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{u}_t^{(m)}, \boldsymbol{\theta}^{(m)}\right), \quad \boldsymbol{u}_t^{(m)} = [\epsilon_t, \mathrm{d}\epsilon_t]^T \tag{2}$$

The goal is to learn f effectively, enabling accurate prediction of stress-strain-volume change under unseen (denoted by *) conditions, including unseen initial states$\boldsymbol{s}_0^*$, exogenous inputs$\boldsymbol{u}_t^*$, and experimental constants $\boldsymbol{\theta}^*$.

3. Data

The proposed approach is validated using experimental data from 28 CD triaxial testson Baskarp sand [6]. The dataset includes samples with initial void ratios$e_0 = (0.85, 0.70, 0.61)$ and confining pressures$\sigma_3 = (5, 10, 20, 40, 80, 160, 320, 640, 800 \sim \mathrm{kPa})$. Training utilized samples with σ_3 from20 to 320 kPa, validation used10 and 640 kPa, while testing employed5 and 800 kPa to access extrapolation (50% below and 25% abovethe training range, respectively) were used for testing.Despite its small size, the dataset captures key stress-strain-volume change relationships in monotonic loading and has been used in a previous study[3].

4. Methodology

R-FFNNs employ recursive predictions over extended horizons during training to enhancetemporalmodelingand reduce error accumulation prevalent in FFNNs. However, it has been found that training R-FFNNs using recursive predictions with a single pass over the entire dataset does not lead to effective learning of the R-FFNN parameters. A more effective training strategy involves partitioning the dataset into multiple smaller windows of data. By training the R-FFNN recursively using these smaller windows, the model can learn more effectively.

Overlapping sub-sequences of length $H$are extracted from a time series of length N, forming $(N - H + 1)$windows: $\mathcal{S}_{H,k}^{(m)} = \left(\boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{u}_t^{(m)}, \boldsymbol{\theta}^{(m)}\right)_{\mathrm{t=k}}^{\mathrm{k+H-1}}, \quad \mathrm{k} = 1, \ldots, \mathrm{N} - \mathrm{H} + 1$, where $\boldsymbol{s}_{t-1}^{(m)}$is the state, $\boldsymbol{u}_t^{(m)}$the exogenous inputs, and $\boldsymbol{\theta}^{(m)}$ the experimental constants. The current state $\boldsymbol{s}_{k-1}^{(m)}$ within a window $\mathcal{S}_{H,k}^{(m)}$ predicts subsequent states recursively:

$$\hat{\boldsymbol{s}}_t^{(m)} = f\left(\boldsymbol{s}_{t-1}^{(m)}, \boldsymbol{u}_t^{(m)}, \boldsymbol{\theta}^{(m)}\right), \quad t = k, \ldots, k + H - 1 \tag{3}$$

The total loss for the full dataset is obtained by averaging the windowed-data loss over window $\mathcal{S}_{H,k}^{(m)}$-- denoted by$L_{H,k}^{(m)}$-- over all windows and all experiments:

$$L_{\mathrm{tot}} = \frac{1}{M(N - H + 1)} \sum_{m=1}^{M} \sum_{k=1}^{N-H+1} \underbrace{\left[\frac{1}{H} \sum_{t=k}^{k+H-1} l\left(\boldsymbol{s}_t^{(m)}, \hat{\boldsymbol{s}}_t^{(m)}\right)\right]}_{L_{H,k}^{(m)}} \tag{4}$$

where lis a user-defined loss function. While the root-mean-square (RMS) is the most commonly used loss function, here we use a more robust Huber loss function as the use of this loss function provided better learning of the parameters. The loss is backpropagated to minimize error across all windows, improving long-term predictions.

Further, to improve training, stress invariants $\mathrm{p}_\mathrm{t}^{(\mathrm{m})}$and $\mathrm{q}_\mathrm{t}^{(\mathrm{m})}$are normalized on a subset-by-subset basis. The total Mexperiments are divided into$J$subsets, where each subset$\mathrm{S_j}(\mathrm{j} = 1, \ldots, \mathrm{J})$contains experiments conducted under the same initial confining pressure σ_3. Within each subset, the RMS values for $p$and$q$are computed individually for each experiment and averaged to obtained subset-specific normalization factors. The normalization can be expressed as:

$$\bar{p}_t^{(m)} = \frac{p_t^{(m)}}{p_{\mathrm{rms}}^{S_j}}, \quad m \in S_j, \quad p_{\mathrm{rms}}^{S_j} = \sqrt{\frac{1}{|S_j|\, N} \sum_{s=1}^{S_j} \sum_{t=1}^{N} \left(p_t^{(s)}\right)^2} \tag{5a}$$

$$\bar{q}_t^{(m)} = \frac{q_t^{(m)}}{q_{\mathrm{rms}}^{S_j}}, \quad m \in S_j, \quad q_{\mathrm{rms}}^{S_j} = \sqrt{\frac{1}{|S_j|\, N} \sum_{s=1}^{S_j} \sum_{t=1}^{N} \left(q_t^{(s)}\right)^2} \tag{5b}$$

Here, N represents the number of time-steps in each experiment, and $|S_j|$ is the number of experiments in subset S_j. Void ratios, exogenous inputs, and experimental constants are normalized using min-max scaling. After scaling, the data is partitioned into training and testing subsets of user-defined length H. This hyperparameter H is critical, as it influences the parameter learning directly. Optimal H was determined by iteratively increasing H up to a length of 13 points in each window, with the final value of $H = 8$ selected based on validation loss.
The optimal R-FFNN architecture, evaluated for hidden layers $(1-3)$ and nodes $(10-200)$, used ReLU activation for efficiency and Huber loss for robustness against outliers. Huber gives better validation and test results in comparison to MSE. Adam optimizer was employed for efficient parameter updates, ensuring stable convergence. The optimal architecture, identified by the lowest validation loss, was then applied consistently across all H for a meaningful performance comparison, as depicted in Figure 2.

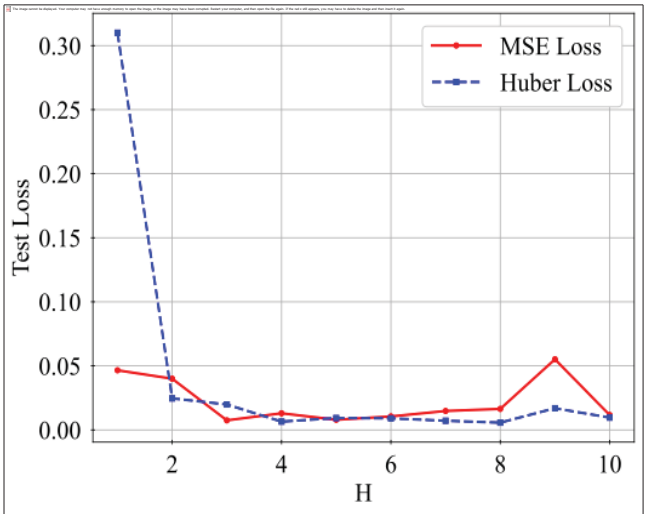

Figure 2: Comparison of test loss using different loss functions, specifically Huber

A default weight initialization of PyTorch was used for all models. The details of all models are summarized in Table 1.

Table 1: Configuration of different models trained on experimental data

ML	*Configuration of Models*						
	Hidden Layers	Nodes/Layer	Optimizer	Loss Function	Activation	Learning Rate	Epochs
LSTM	2	100	Adam	Huber	ReLU	0.001	4000
Bi-LSTM	2	100	Adam	Huber	ReLU	0.001	2100
GRU	2	100	Adam	Huber	ReLU	0.001	5000
FFNN	2	115	Adam	Huber	ReLU	0.0001	2000
R-FFNN(H=8)	2	115	Adam	Huber	ReLU	0.0001	9500

5. Results

The subsequent analysis pertains to the performance of R-FFNNs and RNNs when applied to laboratory data using metrics mean-absolute-error (MAE) and RMS error (RMSE).R-FFNNs exhibit good long-time testing performance on laboratory data at a much lower training cost in comparison to advanced sequence-based DL models.The performance evaluation, detailed in Table 2, highlights the superior performance of the R-FFNN model. Graphical representations further validate this, with Figure 3 showing accurate predictions of $p$at 5 kPa and 800 kPa, and similar alignment observed for q (Figure 4) and e (Figure 5). The R-FFNN framework effectively minimizes accumulative errors and captures long-term temporal dependencies, ensuring robust model performance.

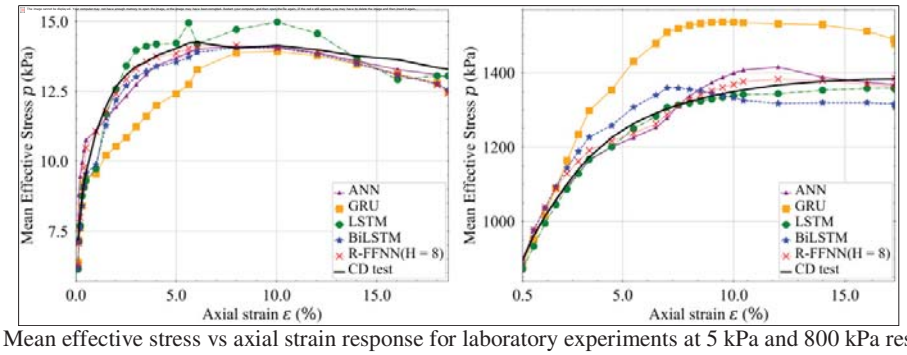

Figure 3: Mean effective stress vs axial strain response for laboratory experiments at 5 kPa and 800 kPa respectively.

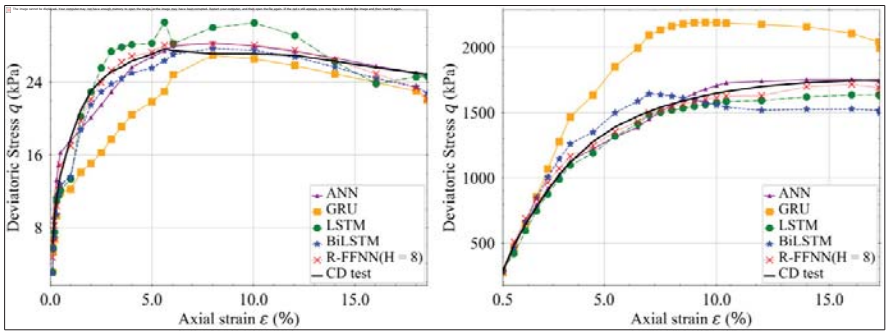

Figure 4:Deviatoric stress vs axial strain response for laboratory experiments at 5 kPa and 800 kPa respectively.

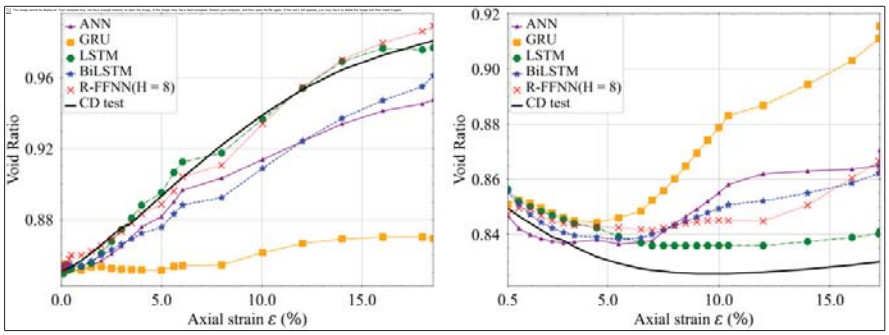

Figure 5: Void Ratio vs axial strain response for laboratory experiments at 5 kPa and 800 kPa respectively.

Table 2: Performance metrics for DL models on test set for experimental data.

ML	*Test set prediction errors*						*Training Time*
	p		*q*		*e*		
	MAE	RMSE	MAE	RMSE	MAE	RMSE	
LSTM	0.0516	0.0747	0.0946	0.1312	0.0309	0.0467	11m10.6sec
Bi-LSTM	0.0522	0.0701	0.1023	0.1334	0.0277	0.0428	12m45.0sec
GRU	0.0792	0.0991	0.1449	0.1801	0.0641	0.0987	11m26.8sec
FFNN	0.0678	0.0853	0.1064	0.1329	0.0367	0.0508	2.4sec
R-FFNN(H=8)	**0.0494**	**0.0671**	**0.0754**	**0.1069**	**0.0228**	**0.0336**	45.9sec

References

[1] G. W. Ellis, C. Yao, and R. Zhao, "Neural network modeling of the mechanicalbehavior of sand," in Engineering Mechanics. ASCE, 1992, pp. 421–424.
[2] D. Penumadu and R. Zhao, "Triaxial compression behavior of sand and gravelusing Artificial Neural Networks," Computers and Geotechnics, vol. 24, no. 3,pp. 207–230, 1999.
[3] P. Zhang, Z.-Y. Yin, and Y.-F. Jin, "State-of-the-art review of machine learningapplications in constitutive modeling of soils," Archives of ComputationalMethods in Engineering, pp. 1–26, 2021.
[4] N. Zhang, S.-L. Shen, A. Zhou, and Y.-F. Jin, "Application of LSTM approachfor modelling stress–strain behaviour of soil," Applied Soft Computing, vol. 100,p. 106959, 2021.
[5] Q. Guan, Z. Yang, N. Guo, and Z. Hu, "Finite element geotechnical analysisincorporating deep learning-based soil model," Computers and Geotechnics, vol.154, p. 105120, 2023.
[6] L. B. Ibsen and L. B. Bødker, "Baskarpsand no. 15: data report 9301," Aalborg:Geotechnical Engineering Group. Data Report, No. 9401, 1994.