Title
Modeling injection-induced fault slip using long short-term memory networks

Journal
Journal of Rock Mechanics and Geotechnical Engineering, 16(11)

ISSN
1674-7755

Authors
Mital, Utkarsh
Hu, Mengsu
Guglielmi, Yves
et al.

Publication Date
2024-11-01

DOI
10.1016/j.jrmge.2024.09.006

Journal of Rock Mechanics and Geotechnical Engineering 16 (2024) 4354–4368

Contents lists available at ScienceDirect

Journal of Rock Mechanics and Geotechnical Engineering

journal homepage: www.jrmge.cn

Full Length Article

Modeling injection-induced fault slip using long short-term memory networks

Utkarsh Mital [a, *], Mengsu Hu [a], Yves Guglielmi [a], James Brown [b], Jonny Rutqvist [a]

[a] Energy Geosciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA
[b] Environmental Genomics & Systems Biology, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA

ARTICLE INFO

Article history:
Received 25 October 2023
Received in revised form
5 August 2024
Accepted 1 September 2024
Available online 5 September 2024

Keywords:
Machine learning
Long short-term memory networks
Fault
Fluid injection

ABSTRACT

Stress changes due to changes in fluid pressure and temperature in a faulted formation may lead to the opening/shearing of the fault. This can be due to subsurface (geo)engineering activities such as fluid injections and geologic disposal of nuclear waste. Such activities are expected to rise in the future making it necessary to assess their short- and long-term safety. Here, a new machine learning (ML) approach to model pore pressure and fault displacements in response to high-pressure fluid injection cycles is developed. The focus is on fault behavior near the injection borehole. To capture the temporal dependencies in the data, long short-term memory (LSTM) networks are utilized. To prevent error accumulation within the forecast window, four critical measures to train a robust LSTM model for predicting fault response are highlighted: (i) setting an appropriate value of LSTM lag, (ii) calibrating the LSTM cell dimension, (iii) learning rate reduction during weight optimization, and (iv) not adopting an independent injection cycle as a validation set. Several numerical experiments were conducted, which demonstrated that the ML model can capture peaks in pressure and associated fault displacement that accompany an increase in fluid injection. The model also captured the decay in pressure and displacement during the injection shut-in period. Further, the ability of an ML model to highlight key changes in fault hydromechanical activation processes was investigated, which shows that ML can be used to monitor risk of fault activation and leakage during high pressure fluid injections.

1. Introduction

Stress changes caused by subsurface temperature and fluid pressure changes can lead to reactivation of pre-existing faults and fractures (Rutqvist et al., 2007). This can be due to anthropogenic fluid injection associated with wastewater disposal, carbon dioxide sequestration, enhanced geothermal exploration and hydrocarbon recovery. Fault slip during injection can induce earthquakes which in turn can cause leakage of contaminant fluids into groundwater (for a review, see Cheng et al., 2023). For example, industrial wastewater disposal has led to an enormous rise in seismic activity in the Central and Eastern United States (Keranen and Weingarten, 2018), some of which has culminated in major events such as the 2011 Prague earthquake (magnitude 5.7) and the 2016 Pawnee earthquake (magnitude 5.8). In 2018–2019, shale gas hydraulic fracturing induced two magnitude 5+ earthquakes in South Sichuan Basin in China (Lei et al., 2019). In 2017, injection related to enhanced geothermal systems led to a magnitude 5.5 earthquake in Pohang, South Korea (Lee et al., 2019).

Fault slip is also an issue of concern in nuclear waste disposal and needs to be assessed while evaluating the performance of any nuclear waste repository site. Model simulations have shown that the long-term heating of the host rock may cause stress changes due to thermal expansion, mineral dehydration and thermal pressurization, which could activate pre-existing fractures and faults thousands of years after closure of the repository (Rutqvist et al., 2020; Urpi et al., 2019). Thermal pressurization can be particularly strong in low permeability rocks such as Argillaceous clay stone or shale (Gens et al., 2007; Rutqvist et al., 2014). Fault slip can also be caused by fluid pressurization due to gas generation (Ortiz et al., 2002).

The above literature highlights the widespread potential of hazards associated with fluid injections and nuclear waste disposal. The need for alternative energy sources to reduce dependence on

* Corresponding author.
E-mail address: umital@lbl.gov (U. Mital).
Peer review under responsibility of Institute of Rock and Soil Mechanics, Chinese Academy of Sciences.

https://doi.org/10.1016/j.jrmge.2024.09.006

fossil fuels means that geological activity associated with enhanced geothermal systems, nuclear waste disposal, and CO_2 storage will continue to rise. In the United States alone, federal investments to the tune of $2.5 billion are expected to fund multiple large-scale CO_2 storage projects which will involve significant injection volumes resulting in large-scale sub-surface pressure changes (DOE News, 2022). As many as 24 countries are developing technology for disposal of spent nuclear fuel in geological formations (Faybishenko et al., 2017). As a result, there is an ever-increasing potential for undesirable and unexpected geomechanical effects such as reactivation of existing faults and development of new leakage pathways.

Clearly, there is a rising need to assess the short and long-term safety of applications related to fluid injections and geologic disposal of nuclear waste. Typically, this is achieved with the aid of physics-based numerical models that seek to predict the thermo-hydro-mechanical response of a fault that eventually leads to the nucleation of an earthquake (Cheng et al., 2023). However, such models require a robust and complex formulation to capture the rapid and strong coupling between temperature, fluid flow and deformation of porous media. It is also challenging to develop constitutive laws that can capture the nonlinear and discontinuous slip behavior of a fault. Additionally, the models need information about the complex three-dimensional (3D) fault geometry as well as material properties that vary with depth – such information is typically not easy to obtain. Finally, such models can be very time-consuming to run, implying that their ability to assess short-term safety of fluid injections and geological disposal of nuclear waste is limited.

Given the challenges associated with conventional modeling techniques, in situ field experiments are sometimes conducted to gather data that provide insight into how fluid injections can affect seismic activity. A prominent example includes in situ experiments of controlled fault activation at the Mont Terri Underground Rock Laboratory in Switzerland (Guglielmi et al., 2021). These experiments involve the injection of high-pressure synthetic pore water in a fault zone intersecting the Opalinus Clay argillite formation. Opalinus clay is considered to be a perfect analogue for caprocks overlaying underground reservoirs and experimenting on it provides vital information about the stability of faults during injection (Guglielmi et al., 2020). Different types of techniques (e.g. active seismic systems, step-rate injection sensors, distributed acoustic sensors) are used to monitor the changes in pore pressure and fault displacements. These experiments have shown that fault activation can be caused by a fluid pressure increase, due to a nucleation phase characterized by the development of fault slip and induced seismicity (De Barros et al., 2023). A range of complex physics-based numerical models have also been conducted to simulate the hydro-mechanical response of one of the Mont Terri fault activation experiments (Rutqvist et al., 2020). As discussed above, application of these models requires complex and detailed 3D mesh construction, constitutive fault model development and calibration of many input parameters to achieve an acceptable match to the field data.

Experiments such as those at Mont Terri provide a large quantity of diverse data which motivate us to explore the application of machine learning (ML) algorithms to model injection-induced fault slip. ML provides an alternate yet powerful modeling framework that can address some of the limitations of physics-based numerical modeling. For instance, ML models do not require prior information about physical constraints, material properties, or constitutive relationships, making it more straightforward to implement such approaches. By learning from sufficient data, ML can predict physical changes in time and space without explicitly solving coupled physical processes.

ML approaches are increasingly being used in a variety of applications of rock mechanics and geotechnical engineering. Lawal and Kwon (2021) provided an overview of ML applications in rock mechanics and concluded that they can perform better than traditional empirical, mathematical, or statistical methods. Recent applications include the work by Cevallos et al. (2023) who developed a convolutional network to process 3D tomography scans of soil samples and identify correctly segmented digital twins of individual soil grains. Mital and Andrade (2022) developed a convolutional framework that can obtain a micromechanical model of macroscopic soil properties using incomplete data, while Hu et al. (2021) used neural style transfer to generate and optimize meshes from rock fracture images. Mahmoodzadeh et al. (2022) used a suite of ML techniques to predict rock strength parameters from laboratory experiments of sandstone, and Banerjee and Chatterjee (2022) used probabilistic neural networks to map sub-surface pore pressure by integrating seismic and well log data.

This study uses an ML approach to model the hydro-mechanical fault response due to high-pressure fluid injection. Modeling the fault response involves making predictions based on sequences (or time-series) of data, which motivates the use of long short-term memory (LSTM) networks. LSTM networks possess the ability to extract temporal dependencies in data, making them an ideal candidate to analyze and predict the sequence of pore fluid injection, fault slip, fluid pressure dissipation, and the cycling of such a sequence. The memory aspect of LSTMs was recently utilized by Ma et al. (2022), who trained their network on massive discrete element simulations and modeled the mechanical responses of a wide range of granular materials subjected to complex loading paths. Since their invention, LSTMs have been used extensively in problems such as speech recognition, language modeling, language translation and image captioning. In earth sciences, LSTM networks have been used in applications such as rainfall-runoff modeling, earthquake detection, and landslide modeling (e.g. Kratzert et al., 2018; Yang et al., 2019; Wang et al., 2020). The proliferation of ML applications has been aided by the advent of open-source libraries such as TensorFlow and PyTorch, which make it straightforward to implement increasingly complex ML architectures.

The central objective of this study is to investigate the general ability of an LSTM-based ML model to predict fault displacements and pore pressure in response to high-pressure fluid injections. It is demonstrated that given sufficient data, ML models can learn fault behavior. A recursive strategy is considered, where the response predicted at one time step is used as part of the input for predicting the next time step. It is shown that the training of ML models for the purpose of recursive predictions is not trivial, and several steps are outlined for a modeler to ensure robust predictions. Using this training procedure, numerical experiments are conducted that test the general ability of an ML model to predict injection-induced fault response. The focus of the study is on fault behavior near the injection borehole. ML is observed to reproduce key hydro-mechanical aspects of injection-induced fault response – such as rapid peaks in pore pressure and displacements during fluid injection, and gradual decays in pore pressure and displacements following injection shut-in. The applicability of ML models for quantitative prediction is found to be constrained by the nature of physical processes prevalent in training data.

The rest of the paper is organized as follows. Section 2 starts by presenting a brief description of the fault slip data used in this work. This is followed by a brief overview of LSTM networks that are used to model the fault response (Section 2.2). Sections 2.3-2.5 describe the model setup, which clearly outlines the model inputs and outputs, as well as the steps needed to reliably capture fault response. The experimental design is summarized next (Section 2.6), which involves an outline of various numerical experiments to

highlight the strengths and limitations of the ML approach. Results start by highlighting the importance of carefully training a model (Section 3). This is followed by the outcomes of various numerical experiments, which are used to demonstrate how ML could be used to detect key changes in fault activation processes (Section 4). Thereafter, some salient aspects of results and some future directions are briefly discussed (Section 5), followed by a conclusion that summarizes the main results (Section 6).

As far as notations are concerned, boldface lower-case variables correspond to vectors (e.g. $\boldsymbol{b}$), and boldface upper-case variables correspond to matrices (e.g. $\boldsymbol{W}$).

2. Data and methods

2.1. Mont Terri fault slip FS-B experiment data

This work utilizes data from the Mont Terri fault slip FS-B experiment. The FS-B experiment was conducted at the Mont Terri Underground Rock Laboratory in Switzerland and was centered on the main fault zone. The main fault intersects all the laboratory's galleries and consists of 23 fully cored boreholes which are used to deploy instrumentation (Fig. 1a). The main fault zone also exhibits variations in strike, dip, and thickness (Fig. 1b). The FS-B experiment was designed to study the integrity of a faulted caprock (here, Opalinus clay) and involved seismic imaging of fluid flow and stress variations during six constant flowrate pore water injections directly in the fault zone. A detailed description of the fault geology and the FS-B experiment along with its instrumentation can be found in Guglielmi et al. (2022). This study only describes the data obtained via a SIMFIP probe (Step-Rate Injection Method for Fracture In situ Properties, Guglielmi et al., 2014) installed in the injection borehole (BFSB2; Fig. 1a). The SIMFIP probe monitors the relative displacement between the hanging wall and footwall of the main fault, in addition to pore pressure and injection flowrate.

The FS-B experiment took place on November 21, 2020. Six injections were conducted at flow rates of 2 L/min (injection 1; 1 L/min = 1.67×10^{-5} m^3/s), 6 L/min (injection 2), and ~10 L/min (injections 3, 4, 5, 6; Fig. 2). Injections 1–5 were 10 min long while Injection 6 was 20 min long. Instantaneous increase in pressure and displacements were observed with injection pressure increase (Fig. 2). During injection shut-in periods, which were about 45–90 min long, slow decays in pressure and displacements were

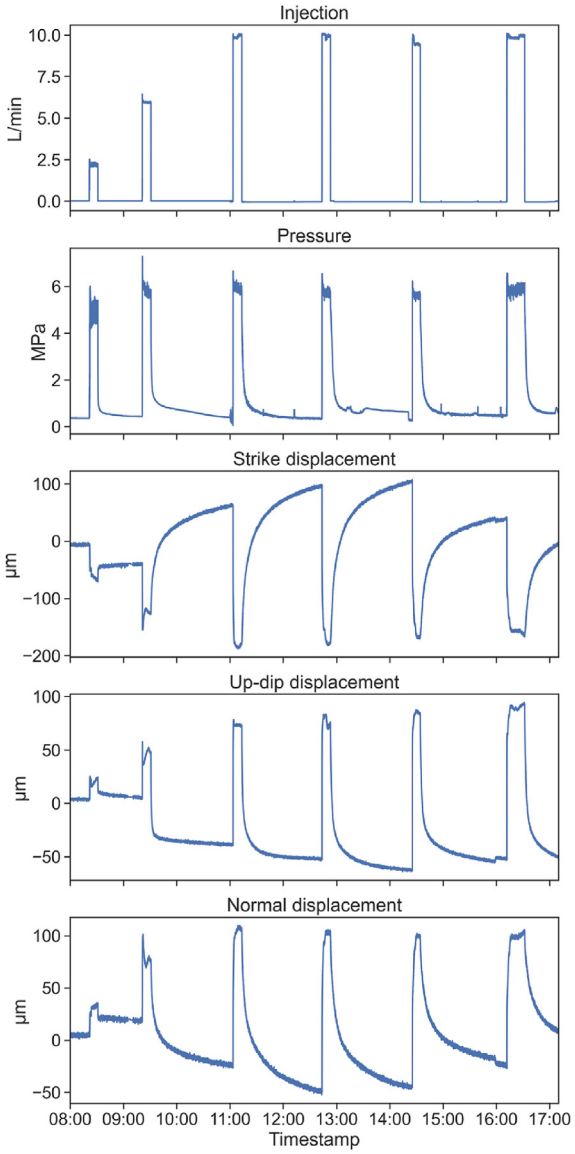

Fig. 2. Injection borehole data obtained via a SIMFIP probe (Guglielmi et al., 2022).

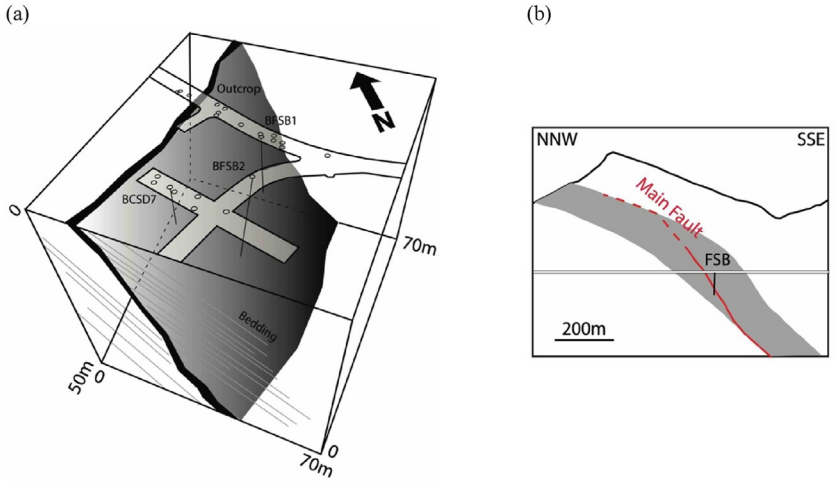

Fig. 1. Fault zone geology: (a) 3D view of the FS-B experiment setting showing various borehole locations (BFSB2, BFSB1, BCSD7), main fault (inclined surface) and galleries; and (b) Vertical cross-section of the fault zone showing the Opalinus clay caprock (gray) being intersected by the main fault (red).

observed following an initially rapid decline. Note that the sampling frequency of the data was 0.5–1 kHz. For this work, the data were downsampled to 2 Hz. The resultant time-series of injection and response was about 9 h (or 16,000 time steps) long.

We highlight some salient features of the fault response (Guglielmi et al., 2022). Cycle 1 is characterized by the onset of fault rupture or fracture opening. The flow rate was kept small to induce a gradual progression of fault slip. Cycle 2 involved an increase in flow rate which was accompanied with an increase in permeability and dilation of the fault. Shut-in periods after cycle 2 exhibit more shear and more closing of the fault compared to the injection periods. This is likely because the various displacement components are computed by projecting the SIMFIP probe displacements on an assumed single fault plane, but the actual displacements may be occurring across multiple planes within the thick fault zone. Cycle 3 involved the formation of a hydraulic connection (i.e. development of a new leakage pathway) between the injection borehole (BFSB2) and a neighboring borehole (BFSB1, which is located up-dip of BFSB2). For more details about the hydraulic connection, see Guglielmi et al. (2022). As fluid was now leaking in the fault zone, the injection protocol was modified, and flow rate was not increased for subsequent cycles. Cycle 4 induced a similar fault response to cycle 3 (with minor differences in up-dip displacement decay during shut-in), while cycles 5 and 6 were characterized by an increasing dilatant behavior of the fault, where irreversible displacements that become successively larger were observed. The formation of a new leakage pathway after cycle 3 ostensibly constrains the pore pressure response for subsequent cycles.

2.2. LSTM networks

LSTM networks (Hochreiter and Schmidhuber, 1997) belong to a family of neural networks, called recurrent neural networks (RNNs), that process sequential data. A neural network consists of a collection of nodes, or 'neurons', which are grouped into three types of layers: input layer, hidden layer(s), and output layer. The conceptual basis of an RNN is formed using a feedforward neural network, which is schematically represented in Fig. 3a. The nodes in the input layer consist of model input, while nodes in the output layer report the model output.

Each node in the hidden layer(s) and the output layer takes in all the nodes from the preceding layer as inputs and produces a single output by computing a weighted sum. For any layer after the input layer, given an n-dimensional input vector $\boldsymbol{x}$, the output is expressed as

$$\hat{\boldsymbol{y}} = F(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \tag{1}$$

where $\boldsymbol{W}$ is a weight matrix (unique to each layer), $\boldsymbol{b}$ is a bias vector, and F is an activation function (such as sigmoid or tanh) whose purpose is to introduce nonlinearity in the computation. $\hat{\boldsymbol{y}}$ approximates the true output $\boldsymbol{y}$. If the input consists of a sequence of data given by $(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\boldsymbol{x}^{(3)},\ldots,\boldsymbol{x}^{(k)})$, then the above formulation will result in a sequence of outputs $(\hat{\boldsymbol{y}}^{(1)},\hat{\boldsymbol{y}}^{(2)},\hat{\boldsymbol{y}}^{(3)},\ldots,\hat{\boldsymbol{y}}^{(k)})$, where $\hat{\boldsymbol{y}}^{(t)}$ is the output corresponding to input $\boldsymbol{x}^{(t)}$. Each input can be thought of an n-dimensional vector, with the superscript t corresponding to the position in the sequence.

If the input sequence comprises of a time series, then the output $\hat{\boldsymbol{y}}^{(t)}$ will depend on not just $\boldsymbol{x}^{(t)}$, but also on all the past input values $(\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\boldsymbol{x}^{(3)},\ldots,\boldsymbol{x}^{(t-1)})$. The formulation in Eq. (1) is not capable of accounting for this dependency since it can consider input only from the current time step t. This motivates the use of RNNs, schematically represented in Fig. 3b. Each RNN cell takes in two inputs: (i) $\boldsymbol{x}^{(t)}$, the input at time step t, and (ii) $\boldsymbol{h}^{(t-1)}$, the hidden state at the previous time step.

In an RNN cell, the output for each time step t is computed as follows:

$$\boldsymbol{h}^{(t)} = \tanh\left(\boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)}\right) \tag{2}$$

where $\boldsymbol{h}^{(t)}$ is the hidden state vector at time step t, and $\boldsymbol{U}$ is a weight matrix. The output of the RNN cell at each time step is then fed into a fully connected layer to produce the final output $\hat{\boldsymbol{y}}$ of the network:

$$\hat{\boldsymbol{y}}^{(t)} = F\left(\boldsymbol{d} + \boldsymbol{V}\boldsymbol{h}^{(t)}\right) \tag{3}$$

where $\boldsymbol{d}$ is a bias vector, and $\boldsymbol{V}$ is a weight matrix. For a regression problem, the activation function F is taken to be the identity function (i.e. $F(\boldsymbol{z}) = \boldsymbol{z}$).

The above RNN maps an input sequence to an output sequence of the same length. In this study, an RNN that reads an entire sequence and produces a single output corresponding to the final time step is considered. To capture dependencies in sequences of length up to 1000 time steps, it is common to consider gated variants of RNNs such as LSTM networks.

Fig. 4 shows the schematic diagram of an LSTM cell typically implemented in standard ML libraries (e.g. TensorFlow, PyTorch). At a given time step, an LSTM cell takes in three inputs: (i) $\boldsymbol{x}^{(t)}$, the input vector at time step t; (ii) $\boldsymbol{h}^{(t-1)}$, the hidden state vector at the previous time step; and (iii) $\boldsymbol{c}^{(t-1)}$, the cell state vector at the previous time step. $\boldsymbol{x}^{(t)}$ and $\boldsymbol{h}^{(t-1)}$ are used to compute the following four quantities:

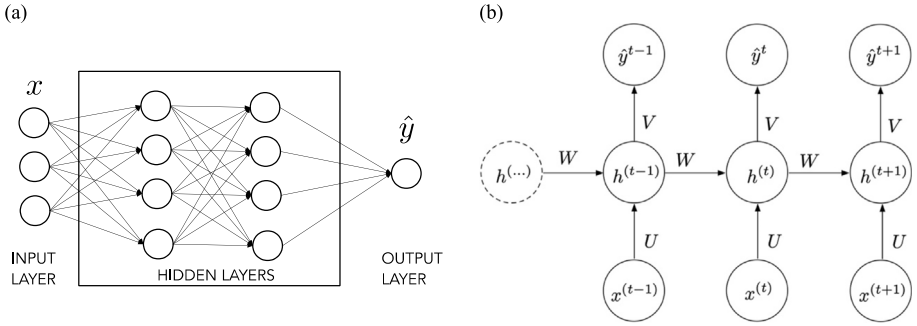

Fig. 3. (a) Schematic diagram of various layers in a feed-forward neural network, and (b) Schematic diagram of an RNN (after Goodfellow et al., 2016).

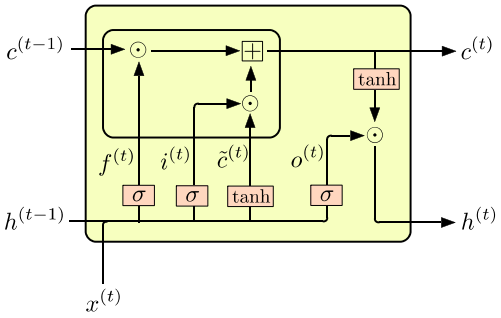

Fig. 4. Schematic diagram of an LSTM cell.

$$\boldsymbol{f}^{(t)} = \sigma\left(\boldsymbol{b}_{\mathrm{f}} + \boldsymbol{U}_{\mathrm{f}}\,\boldsymbol{x}^{(t)} + \boldsymbol{W}_{\mathrm{f}}\,\boldsymbol{h}^{(t-1)}\right) \tag{4}$$

$$\boldsymbol{i}^{(t)} = \sigma\left(\boldsymbol{b}_{\mathrm{i}} + \boldsymbol{U}_{\mathrm{i}}\,\boldsymbol{x}^{(t)} + \boldsymbol{W}_{\mathrm{i}}\,\boldsymbol{h}^{(t-1)}\right) \tag{5}$$

$$\boldsymbol{o}^{(t)} = \sigma\left(\boldsymbol{b}_{\mathrm{o}} + \boldsymbol{U}_{\mathrm{o}}\,\boldsymbol{x}^{(t)} + \boldsymbol{W}_{\mathrm{o}}\,\boldsymbol{h}^{(t-1)}\right) \tag{6}$$

$$\tilde{\boldsymbol{c}}^{(t)} = \tanh\left(\boldsymbol{b}_{\mathrm{c}} + \boldsymbol{U}_{\mathrm{c}}\,\boldsymbol{x}^{(t)} + \boldsymbol{W}_{\mathrm{c}}\,\boldsymbol{h}^{(t-1)}\right) \tag{7}$$

where $\boldsymbol{f}^{(t)}$ is the forget gate vector; $\boldsymbol{i}^{(t)}$ is the input gate vector; $\boldsymbol{o}^{(t)}$ is the output gate vector; $\tilde{\boldsymbol{c}}^{(t)}$ is an intermediate cell state; $\boldsymbol{b}_{\mathrm{f}}$, $\boldsymbol{b}_{\mathrm{i}}$, $\boldsymbol{b}_{\mathrm{o}}$ and $\boldsymbol{b}_{\mathrm{c}}$ are the bias vectors; $\boldsymbol{U}_{\mathrm{f}}$, $\boldsymbol{U}_{\mathrm{i}}$, $\boldsymbol{U}_{\mathrm{o}}$, $\boldsymbol{U}_{\mathrm{c}}$, $\boldsymbol{W}_{\mathrm{f}}$, $\boldsymbol{W}_{\mathrm{i}}$, $\boldsymbol{W}_{\mathrm{o}}$ and $\boldsymbol{W}_{\mathrm{c}}$ are the weight matrices; and σ refers to the sigmoid function. The cell state $\boldsymbol{c}^{(t)}$ is then computed with the help of the forget gate and the input gate vectors as follows:

$$\boldsymbol{c}^{(t)} = \boldsymbol{f}^{(t)} \odot \boldsymbol{c}^{(t-1)} + \boldsymbol{i}^{(t)} \odot \tilde{\boldsymbol{c}}^{(t)} \tag{8}$$

where $\odot$ refers to the Hadamard product (i.e. element-wise multiplication) of two vectors. Finally, the hidden state vector $\boldsymbol{h}^{(t)}$ is computed using the output gate vector and cell state vector:

$$\boldsymbol{h}^{(t)} = \boldsymbol{o}^{(t)} \odot \tanh \boldsymbol{c}^{(t)} \tag{9}$$

At time step t, each LSTM cell outputs the cell state vector $\boldsymbol{c}^{(t)}$ and the hidden state vector $\boldsymbol{h}^{(t)}$. The hidden state vector can then be used to compute the output $\hat{\boldsymbol{y}}^{(t)}$ as shown in Eq. (3). LSTMs have been shown to capture long-term dependencies more effectively when compared to basic RNNs because a higher-order temporal interpolation is obtained by combining Eqs. (4)–(9). The use of various gates along with the cell state vector allows for a dynamic time scale of integration that can change based on the input sequence (Goodfellow et al., 2016).

2.3. Model setup and training

Injecting high-pressure fluid can lead to fault slip by decreasing the shear strength of a fault due to reduction of effective stress. This study develops a model that outputs the hydro-mechanical behavior of a fault (as characterized by evolution of pore pressure and displacements near the injection borehole) in response to fluid injection. Specifically, to model pore pressure and fault displacements (normal, strike, and up-dip) at a given time step, injection at the current time step and the previous $n-1$ time steps is considered. In addition, pore pressure and displacements from the previous n time steps are considered. Incorporating a lag (of n time steps) enables the model to capture temporal dependencies in data. In functional terms, the recursive model can be expressed as

$$\left(P^{(t)}, \boldsymbol{\delta}^{(t)}\right) = g\left(I^{(t,t-1,\ldots,t-n+1)}, P^{(t-1,t-2\ldots,t-n)}, \boldsymbol{\delta}^{(t-1,t-2\ldots,t-n)}\right) \tag{10}$$

where I refers to the injection flow, P refers to the pore pressure, $\boldsymbol{\delta}$ refers to the 3D vector of displacements (normal, strike, and up-dip) and g is the function learned by the LSTM model. $I^{(t,t-1,\ldots,t-n+1)}$ is a shorthand for $(I^{(t)}, I^{(t-1)}, \ldots, I^{(t-n+1)})$, $P^{(t-1,t-2\ldots,t-n)}$ is a shorthand for $(P^{(t-1)}, P^{(t-2)}, \ldots, P^{(t-n)})$, and $\boldsymbol{\delta}^{(t-1,t-2\ldots,t-n)}$ is a shorthand for $(\boldsymbol{\delta}^{(t-1)}, \boldsymbol{\delta}^{(t-2)}, \ldots, \boldsymbol{\delta}^{(t-n)})$. The superscript (e.g. (t)) refers to the time step. Eq. (10) shows that for a given time step t, the model input consists of five sets of time series, while the model output consists of four scalar predictions. The length of each input time series, or the lag, is n (see below for details on n). Table 1 summarizes the model setup and architecture.

The function g (which is a collection of weights and biases) is learned by minimizing the mean-squared error between the modeled and true output. The gradients of the loss function (i.e. mean-squared error) were calculated using back-propagation (Rumelhart et al., 1986), and the weights (and biases) were updated using the Adam optimization algorithm (Kingma and Ba, 2017). Training was stopped once the loss function showed no further improvements. The models were implemented using the TensorFlow platform in Python (Abadi et al., 2015).

2.4. Model evaluation

The various models were evaluated using the root mean squared error (RMSE) and the coefficient of determination (R^2). RMSE is the square root of mean squared error and has the same units as the target variables, making it easy to interpret the models. R^2 is defined as

$$R^2 = 1 - \frac{MSE}{MST} \tag{11}$$

where MSE is the mean squared error of predicted values and MST is the sum of squares of observed values. R^2 normalizes the spread of prediction errors by the spread of observed values and is a highly informative metric for evaluating a regression model (Chicco et al., 2021; Mital et al., 2022). R^2 is unitless and varies from $-\infty$ to 1. Higher values are desirable with a value of 1 indicating a perfect match between predictions and observations.

2.5. Model testing via recursive generation

During testing, the model is evaluated by generating fault pore pressure and displacements in response to fluid injection. A recursive forecasting strategy was adopted (Ben Taieb et al., 2012) which has a significant departure from the training procedure. During testing, the model output is predicted for one time step which is then fed back into the model as part of the input for the next time step, unlike in training where the actual values of pore pressure and displacements are used. This can propagate prediction errors from early to later time steps. To enhance the likelihood that the training produces a robust model that is resistant to error propagation, four measures were adopted as described below.

2.5.1. Setting the LSTM lag n

When building an LSTM network, lag is an important hyperparameter to consider and refers to the number of previous time steps used to predict the output. Put differently, lag refers to the

Table 1
Model setup and LSTM network architecture.

Layer type	Input layer (lag n)	LSTM layer (dimension d)	Fully connected output layer
Output shape	$(n, 5)$*	$(d, 4)$	$(1, 4)$
Layer output description	Injection flow rate, pore pressure, normal displacement, strike displacement, up-dip displacement	Hidden features	Pore pressure, normal displacement, strike displacement, up-dip displacement

Note: *Each variable in the input layer is a sequence of length n. Injection flow rate corresponds to the value at the current time step and the previous $n-1$ time steps. All other inputs correspond to values at the previous n time steps.

length of the input sequence over which an LSTM captures dependencies. A value of lag that is too small may prohibit the network from capturing enough information about the past to make an accurate prediction, while a value of lag that is too large may lead to over-fitting and poor generalization. In the FS-B experiment, injections 1–5 were about 10 min long. The injections were followed by shut-in periods during which an initially rapid decline in pressure and displacement was followed by a slow decay. The period of rapid decline lasted less than 10 min. Therefore, a lag of 20 min was picked to capture most of the time-dependent behavior of the fault during and after an injection. Since the data was down sampled to a frequency of 2 Hz, a lag of 20 min corresponded to 600 time steps. The limitations of choosing a small value of lag are demonstrated in Section 3.

2.5.2. Calibrating the LSTM cell dimension d

Another important hyper-parameter is the LSTM cell dimension, also referred to in literature as "LSTM units". The LSTM cell dimension refers to the dimension of the hidden state vector $\boldsymbol{h}$ (as shown in Eqs. (3) and (9)), which is the same as the dimensions of other vectors associated with an LSTM cell (as shown in Eqs. (4)–(8)). The LSTM cell dimension influences the complexity of relationships that can be learned by the network. Generally, a higher cell dimension facilitates complexity but comes with added computational cost and a potential for overfitting. A lower cell dimension facilitates simplicity and lower computational cost but could lead to underfitting. For the FS-B experiment, a suite of models were trained whose cell dimension was varied manually. Cell dimensions between 25 and 50 were found to be the most optimal.

2.5.3. Weight optimization: learning rate reduction

The learning rate determines the rate at which the weights (and biases) of a neural network get updated during the training process. A high learning rate can lead to faster convergence. This is especially useful during the early stages of training as the neural network weights are far from their optimal values. However, as the training progresses and model weights start converging, a high learning rate could cause the optimizer to overshoot and oscillate near the global minimum. Therefore, the learning rate was reduced by a factor of 2 every time the improvement in loss function plateaued.

2.5.4. No validation cycle during training

Our objective is to train a model on the first few injection cycles and evaluate its ability to generate pore pressure and fault displacement response for subsequent cycles. In an ML framework, it is a common practice to set aside a subset of data (called validation set) which is used to estimate the model's ability to generalize to new unseen data. The validation set is typically also used to make decisions such as choosing hyperparameters, reducing the learning rate, and when to stop the training. This suggests setting aside an injection cycle for validation. However, the various cycles elicit distinct responses from the fault, as the fault evolves with each successive injection cycle (Section 2.1). This precludes the selection of a validation set that can be used to reliably assess model performance during training. However, a validation cycle is still recommended to evaluate a model after training to confirm the quality of model forecasts.

2.6. Experimental design

The model architecture consists of an input layer, an LSTM layer, and a fully-connected layer which produces the final output. The models were trained on Google colab GPU (free version). Each model took approximately 10–20 min to train and 2–3 min to generate response for each test cycle.

To highlight the need for various measures in the training process (as outlined in Sections 2.5.1-2.5.4), a model was trained on injection cycles 1–3, and response was generated for injection cycle 4. This choice of training and testing cycles was motivated by two reasons. First, cycles 1–3 exhibit variability in response (as discussed in Section 2.1). Second, the response of cycle 4 is similar to the response of cycle 3. Hence, the trained model should be able to predict most of the response of cycle 4, and any major discrepancy in predictions could be attributed to shortcomings in the training process. While evaluating the effect of different values of lag and LSTM dimensions on model error, a pseudo-logarithmic sampling strategy was considered which sought to consider a wide range of parameter selection (as shown in Table 2). After highlighting the necessary measures for training, additional numerical experiments were conducted which considered different combinations of injection cycles used for training and testing (Table 3). As discussed in Section 2.1, each injection cycle induces a distinct fault response, with some cycles being more similar to each other. These numerical experiments are used to investigate how the ability of an ML model to predict pressure/displacements is affected by the physical variability of fault response encapsulated in the training cycles, which could enable the ML model in detecting key changes in fault hydromechanical response.

3. Results: highlighting the training process (experiment 1)

3.1. Error analysis and optimal network architecture

Fig. 5 presents an error analysis showing the effect of lag and LSTM dimensions on forecasting cycle 4. Higher values of lag tend to lower the forecasting error and higher values of LSTM dimension tend to increase the forecasting error (necessitating the use of a logarithmic scale to visualize RMSE). For smaller values of LSTM dimension, lags of 600 and 900 yielded similar errors. This suggests

Table 2
Values of lag and LSTM dimension considered in the study.

Lag, n	LSTM dimension, d
{1, 100, 600, 900}	{20, 25, 30, 35, 40, 50, 100, 200}

Table 3
Overview of numerical experiments.

Experiment	Training cycles	Testing cycles	Training dataset length	Testing dataset length
1	1, 2, 3	4	7900	3000
2	1, 2, 3	4, 5, 6	7900	8000
3	4, 5	6	6200	1950
4	3, 4	5	6100	3200
5	1, 2	4	4800	3000

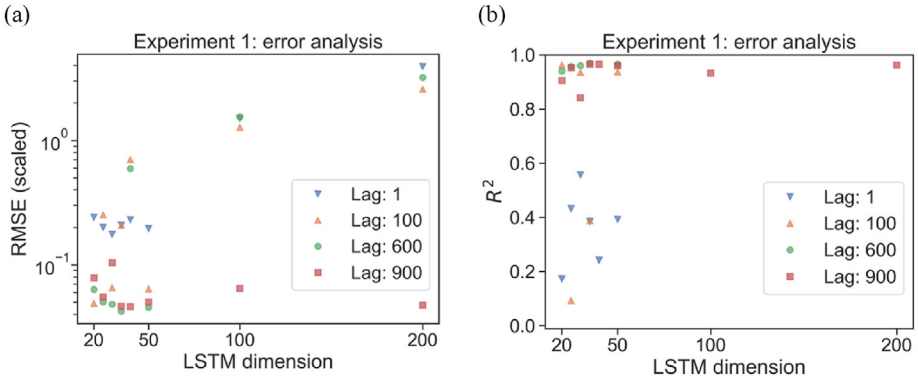

Fig. 5. Error analysis showing the effects of lag and LSTM dimensions on forecasting cycle 4 for experiment 1 in terms of (a) RMSE and (b) R^2. To enable a joint evaluation of pressure and displacements, RMSE is expressed as a scaled quantity which is dimensionless and is computed on pressure and displacement values scaled between 0 and 1. R^2 is computed separately for each variable and then averaged over all variables. The y-axis of R^2 values is truncated to 0, resulting in the omission of negative values from the plot.

an upper limit beyond which increasing the lag does not yield additional benefits in the current scenario. Given the physical rationale discussed in Section 2.5.1, a lag value of 600 (or 20 min) was picked for the remaining experiments. LSTM cell dimensions in the range of [25, 50] tended to be optimal. For experiment 1, a cell dimension of 35 yielded the lowest forecasting error. Table 4 summarizes the final model architecture used for experiment 1.

Sections 3.2-3.5 describe how straying away from even one of the steps described above (Section 2.5) can hamper the predictive ability of the model. Finally, results of the best performing model are shown which incorporated all the above steps.

3.2. A small value of lag hampers predictions during shut-in

Fig. 6 shows the model limitations when the value of lag is too small. Here, the model has a lag of 100, which corresponds to 200 s or ~3.33 min, and is otherwise identical to the model architecture in Table 4. At the onset of injection, the peak pressure is slightly over-predicted, but otherwise the response is well captured during the injection period. However, there are significant errors in the model response during the shut-in period. Interestingly, the predicted response diverges from the true values at approximately 13.33 min (shown by the green dotted line), which corresponds to 3.33 min (or the lag value) into the shut-in period. This is a clear indication that a memory of more than 3.33 min is needed to accurately model the decay. At about 20 min, which is about 10 min in the shut-in period, the predictions flatten out suggesting a complete memory loss for the model. Table 5 shows the variation in errors for pressure and displacements for different values of lag. The forecasting errors tend to be higher for smaller values of lag.

Table 4
Network architecture for the best performing model (experiment 1).

Layer type	Output shape
Input	(600, 5)
LSTM	35
Fully connected (output)	4

3.3. A high value of cell dimension may lead to overfitting

Fig. 7 shows the adverse impact of choosing a value of cell dimension that is too large. Here, the model has a cell dimension of 200 and is otherwise identical to the architecture in Table 4. Once again, the model is observed to accurately predict the response during the injection period, while the shut-in period is characterized by oscillations which is reminiscent of a model that may be overly complex. The predicted response diverges from the true values at approximately 30 min, which corresponds to 20 min (or the lag value) into the shut-in period. The oscillations suggest that the model may have overfit itself to the training data. Table 6 shows the variation in errors for pressure and displacements for different values of the LSTM cell dimension. Note that forecasting errors tend to initially improve with an increase in LSTM dimension and then subsequently tend to get worse, reaching a minimum at LSTM dimension of 35.

3.4. A constant learning rate limits learning

Fig. 8 considers the network architecture described in Table 4. However, the learning rate is not manually reduced during the training process and learning starts with a default rate of 0.001. Once again, accurately forecasting response during the shut-in period was found to be challenging. It is interesting to note that the Adam optimizer adapts the learning rate during the training process by considering the mean and variance of gradients during earlier time steps. This means that as the global minimum of the loss function is approached, the learning rate at which the neural network weights are updated automatically slows down. However, there are still limitations to how well the trained model can generate the shut-in response. This suggests that the built-in reduction in learning rate by the Adam optimizer may not be

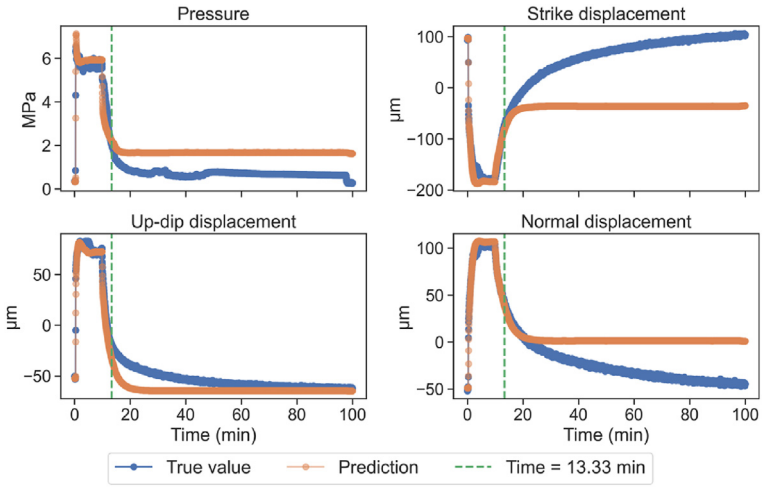

Fig. 6. Predictive capability of the LSTM model declines when lag is small, demonstrated using a value of 100 timesteps or 3.33 min. *X*-axis refers to time elapsed since the onset of injection for cycle 4.

Table 5
Forecasting errors in experiment 1 for different lag values (in timesteps). For all cases, LSTM dimension = 35.

Lag	Pressure		Strike displacement		Up-dip displacement		Normal displacement	
	RMSE (MPa)	R^2	RMSE (μm)	R^2	RMSE (μm)	R^2	RMSE (μm)	R^2
1	0.57	0.87	84.3	−0.06	26.65	0.56	38.28	0.17
100	0.93	0.65	100.54	−0.51	10.95	0.93	30.23	0.48
600	0.34	0.95	8.49	0.99	9.39	0.95	3.9	0.99
900	0.35	0.95	10.35	0.98	8.52	0.96	7.55	0.97

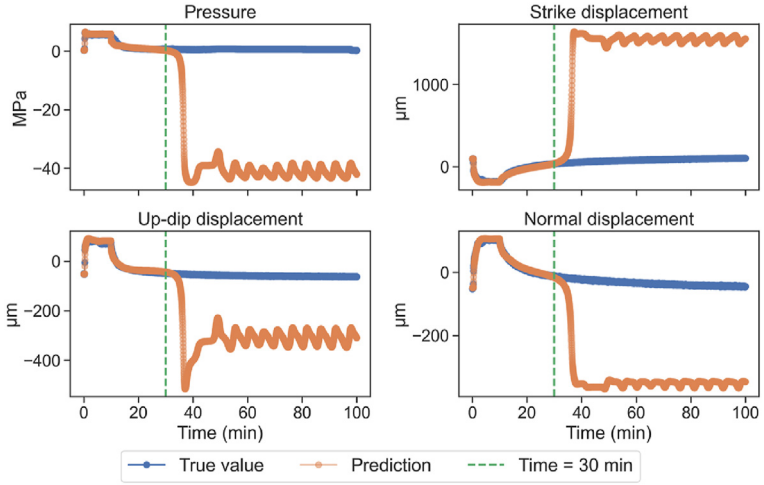

Fig. 7. Predictive capability of the LSTM model that is adversely impacted when the value of LSTM dimension is too large (200 in this case). *X*-axis is the same as Fig. 6.

Table 6
Forecasting errors in experiment 1 for different LSTM dimensions. For all cases, lag = 600 timesteps.

LSTM dimension	Pressure		Strike displacement		Up-dip displacement		Normal displacement	
	RMSE (MPa)	R^2	RMSE (μm)	R^2	RMSE (μm)	R^2	RMSE (μm)	R^2
20	0.31	0.96	23.34	0.92	11.54	0.92	8.43	0.96
25	0.33	0.95	11.66	0.98	12.2	0.91	3.6	0.99
30	0.33	0.95	11.35	0.98	9.83	0.94	6.91	0.97
35	0.34	0.95	8.49	0.99	9.39	0.95	3.9	0.99
40	7.63	−22.81	16.88	0.96	85.38	−3.48	10.26	0.94
50	0.35	0.95	8.87	0.99	10.41	0.93	4.47	0.99
100	19.96	−161.89	373.83	−19.92	16.08	0.84	99.33	−4.6
200	33.32	−452.86	1173.59	−205.13	210.94	−26.33	253.57	−35.51

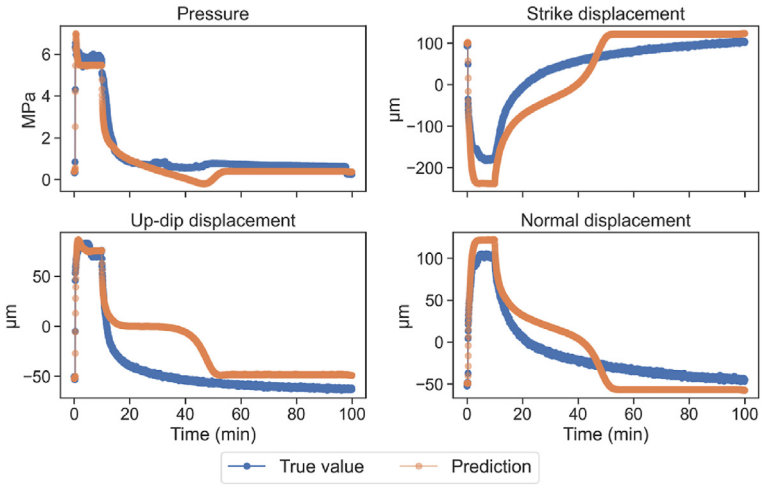

Fig. 8. Predictive capability of the LSTM model that is compromised when the learning rate is held constant. *X*-axis is the same as Fig. 6.

sufficient, suggesting a need to manually reduce the learning rate once the learning curve plateaus. During training, not reducing the learning rate caused the loss function to oscillate which indicates that the optimizer was hampered in its ability to reach the global minimum and may have led to underfitting (learning curve not shown for brevity). This primarily affects the modeling of the shut-in response (as opposed to the injection response) which is characterized by small magnitudes of pressure and displacements.

3.5. Using a separate injection cycle for validation limits learning

Fig. 9 shows how the predictive ability of a model gets compromised if an injection cycle is used as a validation set during the training process. The model setup was otherwise identical to Table 4, the only difference being that cycle 4 was used as a validation set to make decisions about when the learning rate should be reduced and when the training should be stopped. Once again, the main challenge involves predicting the shut-in response. A priori, it was expected that since the model was generating response for the same cycle that was used for validation (i.e. cycle 4), the predictions should be very close to the true values on account of data leakage (e.g. Kaufman et al., 2011). Even though cycle 4 exhibits a response similar to cycle 3 (Section 2.1), there are minor differences (as discussed in Section 3.6 below). This suggests that cycle 4 is not representative of data in the training set. Therefore, by assessing its performance against cycle 4 during training (which affects learning rate reduction and early stopping), the model effectively underfits and is not able to accurately learn all the characteristics of fault response encapsulated in the training cycles. This underfitting affects the overall model robustness while predicting the shut-in response.

3.6. Best performing model: incorporating all training measures

Fig. 10 shows the generated response using a model that incorporates all the measures described in Section 2.5. The model (whose architecture is summarized in Table 4) captures the fault response both qualitatively and quantitatively. At the onset of injection, the peak pressure is slightly over-predicted. However, the displacements are captured accurately throughout the injection period. Prediction of the response during the shut-in period, which proved to be the most challenging aspect for the modeling process, displayed significant improvements. The evolutions of pressure, strike and normal displacements very closely matched the true response. There is a small under-prediction in the decay of up-dip displacement, which suggests that the model is slightly over-

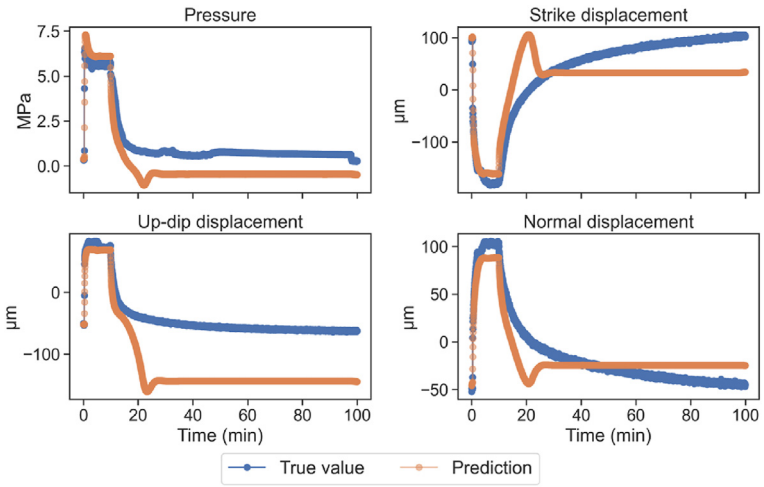

Fig. 9. Predictive capability of the LSTM model that is compromised when a validation set is used during the training process. *X*-axis is the same as Fig. 6.

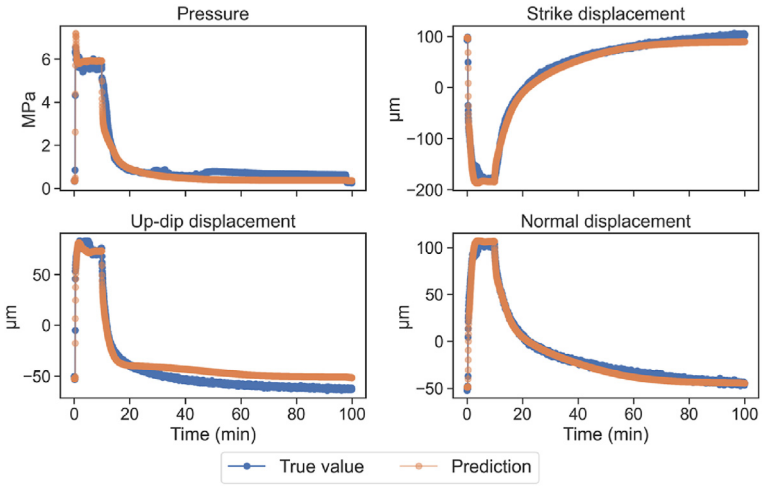

Fig. 10. LSTM model predictions for cycle 4, when trained on cycles 1–3. *X*-axis is the same as Fig. 6.

estimating the magnitude of irreversible shear. This can be attributed to the slight discrepancy in up-dip decay following cycles 3 and 4 due to the formation of a leakage pathway in the up-dip direction (Section 2.1, Fig. 2). Information about the leakage pathway is not provided to the ML model.

Table 7 compares the training and forecasting errors of the best performing model (model M1) with models that use (i) a constant learning rate during training (model M2) and (ii) a separate injection cycle for validation during training (model M3). Models M2 and M3 are characterized by higher values of training error, suggesting that their inability of models to accurately forecast the shut-in period of cycle 4 can be attributed to underfitting. R^2 values are not shown for the sake of brevity.

3.7. Training errors vs. forecasting errors

Table 7 shows that the forecasting errors are significantly higher than the training errors. Table 8 reports the corresponding error ratios (or "overfitting ratios"), which can be defined as the forecasting error normalized by the training error (Khan et al., 2016). A model that generalizes perfectly must yield an error ratio of 1. However, this assumes that the training and test datasets are derived from the same distribution. As pointed out in Section 2.1, cycle 4 (which was the forecast cycle for experiment 1) has minor differences compared to cycle 3. Therefore, some discrepancy is to be expected which precludes an exact match between model and observations. Furthermore, the experimental data are characterized by high-frequency noise which is challenging to replicate and contributes to higher values of error ratios. Therefore, in the present context, it is important to consider the absolute error values when evaluating forecasting skill.

Table 7
Training and forecasting errors in experiment 1 for different training strategies, quantified using the root mean-squared error. For all cases, lag = 600 timesteps and LSTM dimension = 35.

Case	Pressure (MPa)		Strike displacement (μm)		Up-dip displacement (μm)		Normal displacement (μm)	
	T	F	T	F	T	F	T	F
M1	0.08	0.34	1.7	8.49	1.04	9.39	1.1	3.9
M2	0.08	0.46	3.72	48.05	1.41	27.03	1.23	21.21
M3	0.08	1.12	3.66	51.9	1.03	78.97	1.94	18.1

Note: T: Training error; F: Forecasting error; M1: Best training strategy; M2: same as M1 but using constant learning rate; M3: same as M1 but using a separate injection cycle for validation.

4. Results: detecting key changes in fault hydromechanical activation processes (experiments 2–5)

The network architecture and training procedure for detecting changes in fault processes (experiments 2–5) were the same as that used for highlighting the training process (experiment 1). The only difference was in the choice of the LSTM cell dimension, which was found to vary between [25, 50] to yield the lowest values of forecasting error.

4.1. Detecting an emerging fault dilatant slip (experiments 2, 3 and 4)

Experiment 2 considered the same model that was trained in experiment 1 and used it to generate the response for cycles 4–6 (Fig. 11 and Table 9). Cycles 5–6 show a significantly higher dilatant behavior compared to earlier cycles, with larger magnitudes of irreversible shear displacements (strike and normal). Since the model is trained on cycles that exhibit lower magnitudes of irreversible displacements, the magnitude of reversible displacements is overestimated. More specifically, the model reasonably captures the decay in up-dip displacements, which suggests that the additional irreversible displacements are mostly occurring along the strike and in the direction normal to the fault plane. For instance, Table 9 shows that R^2 for strike displacement reduces from 0.99 to 0.31 as dilative behavior increases from cycle 4 to cycle 6. This highlights that a dilatant strike-slip movement of the activated fault is emerging while additional high-pressure water is injected. The model continues to accurately predict both the rise in pore pressure during injection, and the decay in pore pressure during shut-in. This shows that the fault pore pressure response is consistent across all six cycles although the fault is experiencing an irreversible dilatant strike-slip activation. One possible explanation could be that the mechanical dilation rate is directly coupled to the injection flow rate in the fault zone, independent of pore pressure evolution.

Experiment 2 highlighted the shortcomings of using a model to predict highly dilative fault slip response, when trained on early injection cycles that do not exhibit comparable dilatant slip activation. Experiment 3 investigated if predictions of dilative fault slip response would improve when trained on injection cycles later

Table 8
Error ratios in experiment 1 for different training strategies, defined as forecasting error normalized by training error. For all cases, lag = 600 timesteps and LSTM dimension = 35.

Case	Pressure	Strike displacement	Up-dip displacement	Normal displacement
M1	4.25	4.99	9.55	3.55
M2	5.75	12.92	19.17	17.24
M3	14	14.18	76.67	9.33

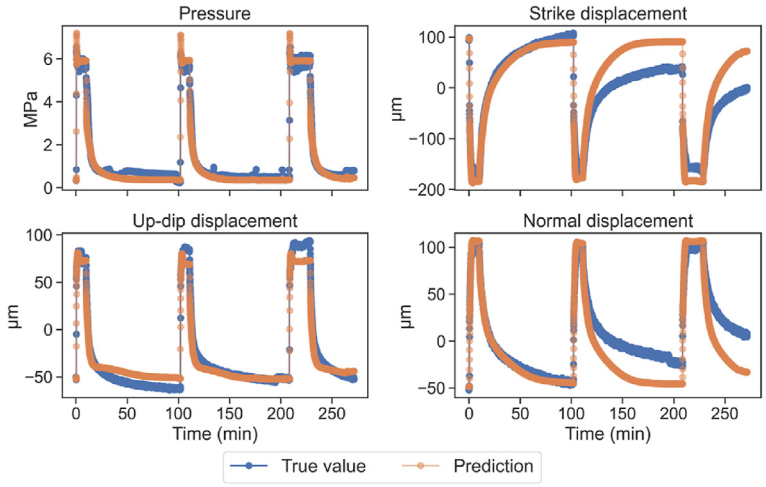

Fig. 11. LSTM model predictions for experiment 2 (cycles 4–6, when trained on cycles 1–3). *X*-axis is the same as Fig. 6.

Table 9
Forecasting errors for different injection cycles in experiment 2. The model corresponds to the best performing model determined in experiment 1.

Cycle	Pressure		Strike displacement		Up-dip displacement		Normal displacement	
	RMSE (MPa)	R^2	RMSE (μm)	R^2	RMSE (μm)	R^2	RMSE (μm)	R^2
4	0.34	0.95	8.49	0.99	9.39	0.95	3.9	0.99
5	0.21	0.98	51.01	0.25	6.33	0.97	26.1	0.36
6	0.24	0.99	52.9	0.31	11.27	0.96	29.94	0.38

than cycles 1 to 3. Therefore, a model was trained on cycles 4–5 (ignoring cycles 1–3 as they do not exhibit dilative behavior) and used to predict response for cycle 6. Fig. 12 shows that this model is indeed able to predict irreversible displacements more accurately, when compared with predictions for cycle 6 in experiment 2 (Tables 9 and 10). The model also accurately predicts variations in fault pore-pressure as earlier. Experiment 3 thus shows that an LSTM model can predict an increase in dilatant slip related to fault

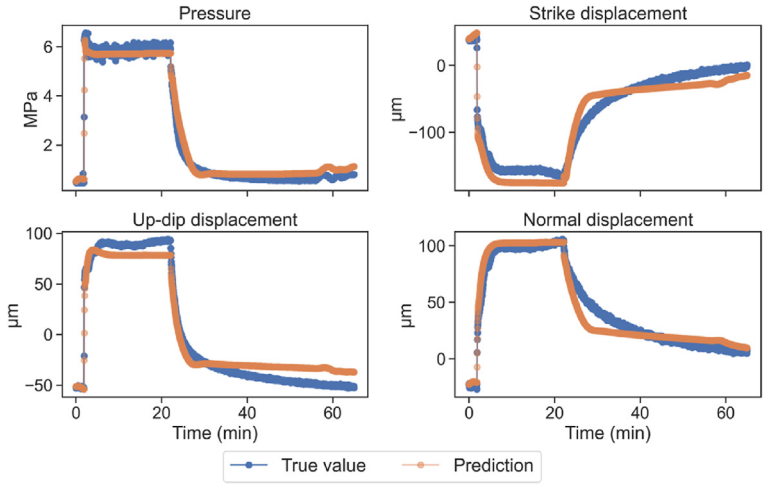

Fig. 12. LSTM model predictions for experiment 3 (cycle 6, when trained on cycles 4–5; cell dimension: 25). *X*-axis refers to time elapsed since the onset of injection for cycle 6.

Table 10
Forecasting errors for experiments 3–5. The injection cycle that is forecasted is reported in parenthesis to enable comparison with results on experiment 2 (Table 9).

Experiment	Pressure		Strike displacement		Up-dip displacement		Normal displacement	
	RMSE (MPa)	R^2	RMSE (μm)	R^2	RMSE (μm)	R^2	RMSE (μm)	R^2
3 (cycle 6)	0.26	0.99	16.34	0.93	11.36	0.96	8.55	0.95
4 (cycle 5)	0.11	0.99	32.3	0.7	6.69	0.97	16.44	0.75
5 (cycle 4)	0.69	0.8	20.64	0.94	20.07	0.75	15.31	0.87

activation.

Experiment 3 highlighted the model's ability to predict an emerging dilatant slip during cycle 6 by training on cycles 4–5. Subsequently, experiment 4 investigated if it is possible to predict the dilatant slip emerging in cycle 5 by training the model on cycles 3–4. Cycles 1–2 were excluded from training as their response is quite distinct from cycle 5. Fig. 13 shows the best predicted response for cycle 5. Although there are improvements in predictions of strike and normal displacements compared with experiment 2 (Tables 9 and 10), there is still a tendency to overestimate their magnitude during the shut-in period. This experiment suggests that there is a change in fault hydromechanical response during cycle 5 which triggers a dilatant strike-slip response, after a "nucleation" phase that occurred over cycles 1 to 4.

4.2. Detecting fault leakage along the up-dip direction (experiment 5)

The final numerical experiment sought to predict the response of injection cycle 4, but by using a model trained only on cycles 1–2. Cycle 3 (which induces a similar response compared to cycle 4) was excluded from training and the ability of the ML model to predict cycle 4 was evaluated (as shown in Fig. 14). The model was found to overestimate the peak values of pore pressure and strike and normal displacements, while making reasonable predictions for the shut-in period. As discussed in Section 2.1, pore pressure generation gets constrained after cycle 3 due to development of a leakage pathway along the up-dip direction. Since that is a new physical process unknown to ML model, the peak pore pressure gets overestimated, which in turn seems to affect the peak value predictions of strike and normal displacements (compared with the results of experiment 2 in Tables 9 and 10).

5. Discussion

5.1. Error propagation during shut-in period

The fault response was modeled using a recursive strategy that predicted one time-step at a time. Predictions for the first time-step were fed as input for the next time step. Although this is an intuitive modeling strategy, it is susceptible to error propagation, especially during the shut-in period when pressure and displacements gradually recover from high values obtained during the preceding injection. During the training process, the model optimizes its weights to reduce the prediction error which is disproportionately affected by higher magnitudes of pressure/displacements during injection. The shut-in response is characterized by declining magnitudes of pressure/displacements, and further optimizing the model weights to reduce error during shut-in requires careful training as outlined in Section 2.5.

5.2. Detecting changes in fault hydro-mechanical response

When trained properly, the ML model can generate robust predictions of response to injection cycles. However, as demonstrated in experiments 2–5 (Section 4), the accuracy of the predictions depends on the physical processes encapsulated during training. If the fault response in training data does not capture highly irreversible dilative fault slip response, then ML is unable to reproduce such behavior while generating fault response (experiment 2 (Fig. 11) and experiment 4 (Fig. 13)). Results also showed that if the fault response in training data did not contain any information about the development of a leakage pathway, the model over-estimated pore pressure (experiment 5 (Fig. 14)). This characteristic of an ML framework can be exploited to highlight when changes in fault activation occur during an injection process.

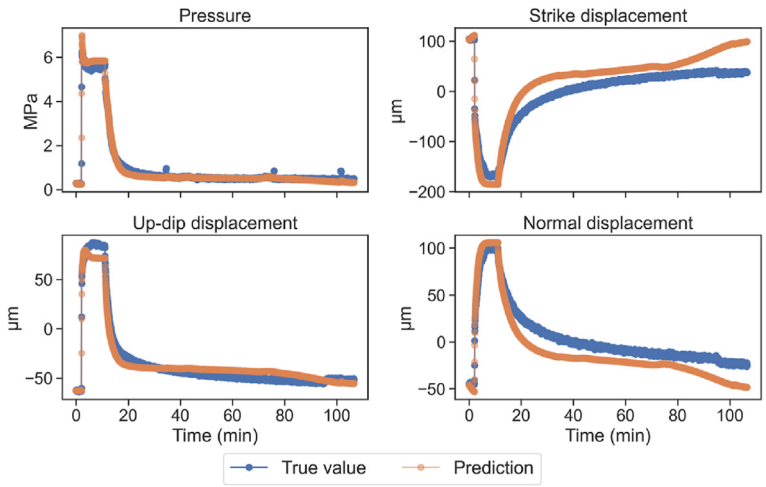

Fig. 13. LSTM model predictions for experiment 4 (cycle 5, when trained on cycles 3–4; cell dimension: 30). *X*-axis refers to time elapsed since the onset of injection for cycle 5.

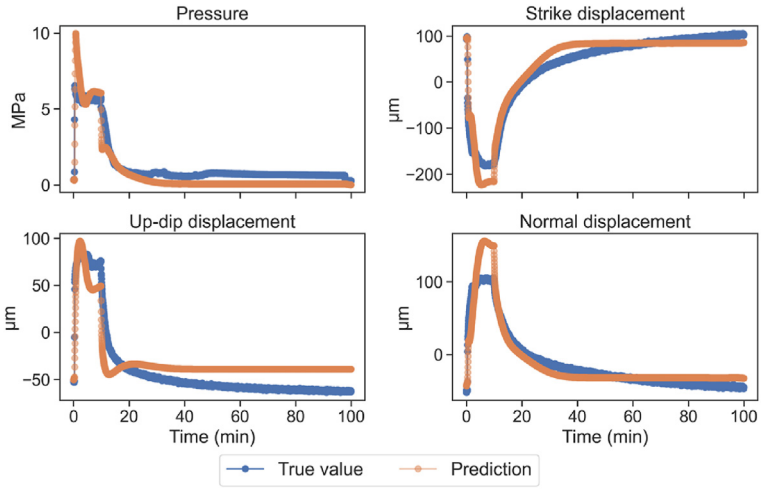

Fig. 14. LSTM model predictions for experiment 5 (cycle 4, when trained on cycles 1–2; cell dimension: 50). *X*-axis is the same as Fig. 6.

5.3. Focus of this work and extension to future research directions

It is important to note that the focus of the current study is on modeling fault pressure and displacements, and not on predicting induced seismicity, during fluid injections. In general, predicting seismicity is challenging as there is a lack of data on past seismic events prior to the onset of injection activities. This data gap can be addressed with the advent of distributed acoustic sensing fibers eventually cemented behind the well casing that may enable strain monitoring along the entire length of the injection borehole (e.g. Correa et al., 2021). Such systems provide access to very high-frequency strain and strain-rate datasets which could be coupled with data from geophones and seismometers to develop more sophisticated ML models that could detect changes in strain/stress rates associated with fault activation (e.g. Wozniakowska and Eaton, 2020; Ma and Chen, 2022; Anikiev et al., 2023). This could involve isolating reversible and irreversible unstable fault displacements, which are key for assessing and modeling induced seismicity.

Another focused goal of the current study is to fully explore the usage of pressure and displacement data from a single point near the injection site. The data came from controlled experiments where fluid was injected directly into a fault. Ideally, fluid injection would take place in a reservoir away from any major faults and direct injection inside a fault would be an unanticipated result. A more generalizable ML model will need to model pressure and displacements in the entire reservoir. In addition to requiring a rich set of temporal data of pressure and displacement from multiple boreholes, a reservoir scale ML model would need information about the spatial positions and depths of boreholes within the reservoir, reservoir geology, and orientations and focal mechanisms of known faults. Determination of subsurface geology and fault mechanisms may require constructing a 3D seismic wavefield with the aid of passive and active seismic data. Subsequently, ingesting such information would require specialized ML architectures such as convolutional or graph neural networks, which can consider spatial context and correlations in data. Such ML models could enable a deeper investigation into how subsurface geology and faulting regimes can impact fault pressure and displacements during subsurface fluid injections. Additionally, the scope of such models could be extended to model multiple reservoirs by considering data from a diverse set of field experiments such as the EGS Collab project, fault activation experiments in crystalline formations at Grimsel (Switzerland) and faulted carbonates in France (Amann et al., 2018; Fu et al., 2021; Guglielmi et al., 2021).

The current study focused on modeling fault multiphysics by using multiphysical data obtained from underground laboratory experiments without explicitly solving coupled equations of multiphysics. As demonstrated, the complex and evolving coupled processes are implicitly considered since the ML model can capture rapid and irreversible changes in pore pressure and displacements. To develop a more general reservoir scale ML model, it may become important to explicitly consider multiphysics so that a variety of physical processes and properties can be efficiently learned and accounted for. This could be realized by implementing physics-informed neural networks where the loss function is augmented using physical relationships (Raissi et al., 2019; Haghighat et al., 2021).

6. Conclusions

This study investigated the general ability of an LSTM-based ML model to predict fault displacement and pore pressure in response to high-pressure geological fluid injection. Data were obtained from an in situ experiment in Mont Terri from November 2020, which involved controlled fault activation using six injection cycles. The focus of this study was on fault response near the injection borehole. A recursive modeling strategy was used where the fault response was predicted for one time step and was then used as part of the input for predicting the next time step. To prevent error accumulation while modeling injection-induced fault pressure and displacements, four practices were outlined and demonstrated to train a robust model, namely (i) setting an appropriate value of lag, (ii) calibrating the LSTM dimension, (iii) reduction of learning rate when loss plateaus and (iv) not using a separate cycle for validation during model training. While predicting response of cycle 4 in experiment 1, these practices resulted in forecasts with R^2 values close to 1 for pore pressure and fault displacements.

Five numerical experiments were conducted, each comprising a different combination of injection cycles used for training and testing, to investigate the ability of an ML model to predict fault pressure and displacements. The ML model captured peaks in pressure and displacements that accompany an increase in fluid injection. The model could also capture the decay in pressure and displacements during the injection shut-in period. The predictive ability of the model was highly dependent on the physical processes of fault behavior observed during the training process, exemplified prominently by a reduction in R^2 for strike

displacement from 0.99 to 0.31 as dilative behavior increased from cycle 4 to cycle 6 in experiment 2. This motivates a strategy where ML could be used to identify key changes in fault activity such as the creation of a major leakage pathway and an emerging dilatant slip after several injection cycles.

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.

CRediT authorship contribution statement

Utkarsh Mital: Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Mengsu Hu:** Writing – review & editing, Methodology, Investigation, Funding acquisition, Conceptualization. **Yves Guglielmi:** Writing – review & editing, Methodology, Funding acquisition, Conceptualization. **James Brown:** Writing – review & editing, Methodology. **Jonny Rutqvist:** Writing – review & editing, Conceptualization.

Acknowledgments

This work was supported by the US Department of Energy (DOE), the Office of Nuclear Energy, Spent Fuel and Waste Science and Technology Campaign, under Contract Number DE-AC02-05CH11231 and the National Energy Technology Laboratory under the award number FP00013650 at Lawrence Berkeley National Laboratory. The authors would also like to thank Carlos F. Jové Colón from Sandia National Laboratory for providing feedback on the manuscript.

References

Abadi, M., Agarwal, A., Barham, P., et al., 2015. TensorFlow: large-scale machine learning on heterogeneous systems. https://doi.org/10.48550/arXiv.1603.04467.

Amann, F., Gischig, V., Evans, K., et al., 2018. The seismo-hydromechanical behavior during deep geothermal reservoir stimulations: open questions tackled in a decameter-scale in situ stimulation experiment. Solid Earth 9, 115–137.

Anikiev, D., Birnie, C., Waheed, U.B., 2023. Machine learning in microseismic monitoring. Earth Sci. Rev. 239, 104371.

Banerjee, A., Chatterjee, R., 2022. Pore pressure modeling and in situ stress determination in Raniganj basin, India. Bull. Eng. Geol. Environ. 81, 49.

Ben Taieb, S., Bontempi, G., Atiya, A.F., Sorjamaa, A., 2012. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. Expert Syst. Appl. 39, 7067–7083.

Cevallos, S.B., Jerves, A.X., Mital, U., et al., 2023. Towards a more accurate characterization of granular media 2.0: involving AI in the process. Comput. Geotech. 160, 105510.

Cheng, Y., Liu, W., Xu, T., et al., 2023. Seismicity induced by geological CO2 storage: a review. Earth Sci. Rev. 239, 104369.

Chicco, D., Warrens, M.J., Jurman, G., 2021. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ. Comput. Sci. 7, e623. https://doi.org/10.7717/peerj-cs.623.

Correa, J., Pevzner, R., Freifeld, B.M., et al., 2021. Continuous downhole seismic monitoring using surface orbital vibrators and distributed acoustic sensing at the CO2CRC otway project: field trial for optimum configuration. In: Li, Y., Karrenbach, M., Ajo-Franklin, J.B. (Eds.), Geophysical Monograph Series, first ed. Wiley, pp. 175–189.

De Barros, L., Guglielmi, Y., Cappa, F., Nussbaum, C., Birkholzer, J., 2023. Induced microseismicity and tremor signatures illuminate different slip behaviours in a natural shale fault reactivated by a fluid pressure stimulation (Mont Terri). Geophys. J. Int. 235, 531–541.

DOE News, 2022. Biden-harris administration announces $4. 9 Billion to Deploy Infrastructure Necessary to Manage and Store Carbon Pollution. https://www.energy.gov/articles/biden-harris-administration-announces-49-billion-deploy-infrastructure-necessary-manage. (Accessed 18 September 2023).

Faybishenko, B., Birkholzer, J., Sassani, D., Swift, P., 2017. International approaches for nuclear waste disposal in geological formations: geological challenges in radioactive waste isolation—fifth worldwide review. Technical report. https://doi.org/10.2172/1353043.

Fu, P., Schoenball, M., Ajo-Franklin, J.B., et al., 2021. Close observation of hydraulic fracturing at EGS Collab experiment 1: fracture trajectory, microseismic interpretations, and the role of natural fractures. J. Geophys. Res. Solid Earth 126, e2020JB020840. https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2020JB020840.

Gens, A., Vaunat, J., Garitte, B., Wileveau, Y., 2007. In situ behaviour of a stiff layered clay subject to thermal loading: observations and interpretation. Geotechnique 57 (2), 207–228.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Guglielmi, Y., Ajo-Franklin, J., Birkholzer, J., et al., 2022. Imaging leakage associated with caprock fault activation: results from the fault slip experiment in Mt. Terri opalinus clay analogue caprock. In: Gerdes, K.F. (Ed.), Carbon Dioxide Capture for Storage in Deep Geologic Formations – Results from the CO2 Capture Project. Volume 5: CCS Technology Development and Demonstration Results (2015-2022). CO_2 Capture Project, pp. 715–732. https://www.co2captureproject.org/xt457PuN/report/CCP4v5_full_version-v2.pdf.

Guglielmi, Y., Cappa, F., Lançon, H., et al., 2014. ISRM suggested method for step-rate injection method for fracture in-situ properties (SIMFIP): using a 3-components borehole deformation sensor. Rock Mech. Rock Eng. 47, 303–311.

Guglielmi, Y., Nussbaum, C., Cappa, F., De Barros, L., Rutqvist, J., Birkholzer, J., 2021. Field-scale fault reactivation experiments by fluid injection highlight aseismic leakage in caprock analogs: implications for CO2 sequestration. Int. J. Greenh. Gas Control 111, 103471.

Guglielmi, Y., Nussbaum, C., Jeanne, P., Rutqvist, J., Cappa, F., Birkholzer, J., 2020. Complexity of fault rupture and fluid leakage in shale: insights from a controlled fault activation experiment. J. Geophys. Res. Solid Earth 125, e2019JB017781. https://onlinelibrary.wiley.com/doi/abs/10.1029/2019JB017781.

Haghighat, E., Raissi, M., Moure, A., Gomez, H., Juanes, R., 2021. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. Comput. Methods Appl. Mech. Eng. 379, 113741.

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

Hu, M., Rutqvist, J., Steefel, C.I., 2021. Mesh generation and optimization from digital rock fractures based on neural style transfer. J. Rock Mech. Geotech. Eng. 13, 912–919.

Kaufman, S., Rosset, S., Perlich, C., 2011. Leakage in data mining: formulation, detection, and avoidance. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, San Diego, California, USA, pp. 556–563.

Keranen, K.M., Weingarten, M., 2018. Induced seismicity. Annu. Rev. Earth Planet Sci. 46, 149–174.

Khan, S.Z., Suman, S., Pavani, M., Das, S.K., 2016. Prediction of the residual strength of clay using functional networks. Geosci. Front. 7, 67–74.

Kingma, D.P., Ba, J., 2017. Adam: a method for stochastic optimization. arXiv: 14126980 [Cs]. https://doi.org/10.48550/arXiv.1412.6980.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M., 2018. Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks. Hydrol. Earth Syst. Sci. 22, 6005–6022.

Lawal, A.I., Kwon, S., 2021. Application of artificial intelligence to rock mechanics: an overview. J. Rock Mech. Geotech. Eng. 13, 248–266.

Lee, K.-K., Ellsworth, W.L., Giardini, D., et al., 2019. Managing injection-induced seismic risks. Science 364 (6442), 730–732.

Lei, X., Wang, Z., Su, J., 2019. The december 2018 ML 5.7 and january 2019 ML 5.3 earthquakes in South Sichuan basin induced by shale gas hydraulic fracturing. Seismol Res. Lett. 90 (3), 1099–1110.

Ma, G., Guan, S., Wang, Q., Feng, Y.T., Zhou, W., 2022. A predictive deep learning framework for path-dependent mechanical behavior of granular materials. Acta Geotech 17, 3463–3478.

Ma, X., Chen, T., 2022. Small seismic events in Oklahoma detected and located by machine learning–based models. Bull. Seismol. Soc. Am. 112 (6), 2859–2869.

Mahmoodzadeh, A., Mohammadi, M., Ghafoor Salim, S., et al., 2022. Machine learning techniques to predict rock strength parameters. Rock Mech. Rock Eng. 55, 1721–1741.

Mital, U., Andrade, J.E., 2022. Bridging length scales in granular materials using convolutional neural networks. Comp. Part. Mech. 9, 221–235.

Mital, U., Dwivedi, D., Özgen-Xian, I., Brown, J.B., Steefel, C.I., 2022. Modeling spatial distribution of snow water equivalent by combining meteorological and satellite data with lidar maps. Artif. Intell. Earth Syst. 1 (4), e220010. https://journals.ametsoc.org/view/journals/aies/1/4/AIES-D-22-0010.1.xml.

Ortiz, L., Volckaert, G., Mallants, D., 2002. Gas generation and migration in Boom Clay, a potential host rock formation for nuclear waste storage. Eng. Geol. 64, 287–296.

Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536.

Rutqvist, J., Birkholzer, J., Cappa, F., Tsang, C.-F., 2007. Estimating maximum sustainable injection pressure during geological sequestration of CO2 using

coupled fluid flow and geomechanical fault-slip analysis. Energy Convers. Manag. 48 (6), 1798–1807.
Rutqvist, J., Graupner, B., Guglielmi, Y., et al., 2020. An international model comparison study of controlled fault activation experiments in argillaceous claystone at the Mont Terri Laboratory. Int. J. Rock Mech. Min. Sci. 136, 104505.
Rutqvist, J., Zheng, L., Chen, F., Liu, H.-H., Birkholzer, J., 2014. Modeling of coupled thermo-hydro-mechanical processes with links to geochemistry associated with bentonite-backfilled repository tunnels in clay formations. Rock Mech. Rock Eng. 47, 167–186.
Urpi, L., Rinaldi, A.P., Rutqvist, J., Wiemer, S., 2019. Fault stability perturbation by thermal pressurization and stress transfer around a deep geological repository in a clay formation. J. Geophys. Res. Solid Earth 124, 8506–8518.
Wang, Q., Guo, Y., Yu, L., Li, P., 2020. Earthquake prediction based on spatio-temporal data mining: an LSTM network approach. IEEE Trans. Emerg. Topics Comput. 8, 148–158.
Wozniakowska, P., Eaton, D.W., 2020. Machine learning-based analysis of geological susceptibility to induced seismicity in the Montney formation, Canada. Geophys. Res. Lett. 47, e2020GL089651. https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2020GL089651.
Yang, B., Yin, K., Lacasse, S., Liu, Z., 2019. Time series analysis and long short-term memory neural network to predict landslide displacement. Landslides 16, 677–694.

Dr. Utkarsh Mital obtained his PhD in Applied Mechanics from California Institute of Technology, USA, and worked as a Project Scientist in Energy Geosciences at Lawrence Berkeley National Laboratory, USA. His research interests include physics-based and machine learning modeling of earth system processes spanning hydrology and geomechanics. Notable accomplishments include (i) modeling mechanical behavior of soils and faults using discrete-element, continuum, and machine learning methods, (ii) geostatistical modeling to quantify uncertainty in field measurements of shear wave velocity, (iii) spatial downscaling of current and future climate using machine learning methods, and (iv) modeling mountainous snowpack using machine learning methods. Dr. Mital is currently an Earth Scientist at Arva Intelligence.