Engineering with Computers (2023) 39:3353–3370
https://doi.org/10.1007/s00366-022-01756-w

ORIGINAL ARTICLE

An auto-tuned hybrid deep learning approach for predicting fracture evolution

Sheng Jiang[1] · Zifeng Cheng[1] · Lei Yang[1] · Luming Shen[1]

Received: 28 February 2022 / Accepted: 25 October 2022 / Published online: 8 November 2022

Abstract
In this study, a novel auto-tuned hybrid deep learning approach composed of three base deep learning models, namely, long short-term memory, gated recurrent unit, and support vector regression, is developed to predict the fracture evolution process. The novelty of this framework lies in the auto-determined hyperparameter configurations for each base model based on the Bayesian optimization technique, which guarantees the fast and easy implementation in various practical applications. Moreover, the ensemble modeling technique auto consolidates the predictive capability of each base model to generate the final optimized hybrid model, which offers a better prediction of the overall fracture pattern evolution, as demonstrated by a case study. The comparison of the different prediction strategies exhibits that the direct prediction is a better option than the recursive prediction, in particular for a longer prediction distance. The proposed approach may be applied in various sequential data predictions by adopting the adaptive prediction scheme.

Keywords Fracture prediction · Deep learning · Bayesian optimization · Hybrid modeling · Prediction strategy

1 Introduction

The prediction of fracture growth is pivotal and has extensive application prospect in many research fields, including geoscience, engineering, materials science and so forth. The prevailingly conventional approach is performing either on-site or laboratory experiments to gain a good physical understanding of the fracture mechanisms. Then physics-based constitutive modellings are performed to simulate the fracture pattern development [1, 2]. Although this classic approach has hitherto been appreciated, it has obvious drawbacks. The prediction accuracy is saliently dependent on the understanding of the underlying physics, which significantly impedes its wider and deeper practical application scenarios. Fracture propagation is regarded as a complex process interlinked with coupled mechanical, thermal, and chemical processes [3, 4]. With the increase in the material or structure complexity, the investigation of basic interactions and failure mechanisms faces severe challenges [5]. Consequently, existing non-linear constitutive modellings are not sufficiently mature to fully capture the spatial and temporal complexity inherent to fracturing behaviors [6]. And, for some certain multi-phase modeling, establishing and meshing the high-fidelity model are time-consuming [7]. This brings another unavoidable challenge, the high computational cost. Such expensive computations are not always available for quick practical applications.

In this context, the deep learning models are fast emerging [8]. The deep learning model specializes in studying the inherent law of the sample data and bypassing the understanding of the physical laws [9]. With the unprecedented growing data from both experiments and numerical simulations, the various data-driven deep learning models can provide efficient surrogate models to directly achieve the fracture dynamics forecasting. For instance, Nguyen-Le et al. [10] proposed a coupled technique of long short-term memory model and hidden Markov model for the prediction of single crack propagation. By comparing with the standalone long short-term memory model, the advantages of the proposed model lie in higher predicting accuracy and smaller required sequential dataset. Fernández-Godino et al. [11] leveraged the sequence-to-sequence encode–decode emulator to generate the crack length prediction by integrating two sequential data, namely, longest crack length and the maximum tensile stress. In addition, the random forest

✉ Luming Shen
luming.shen@sydney.edu.au

[1] School of Civil Engineering, The University of Sydney, Camperdown, NSW 2006, Australia

model or convolutional-based models are also applied to the fracture path prediction problems [12, 13].

In terms of the prediction of fracture propagation under different complex loading conditions or initial crack configurations, a large volume of well-labeled data is required for the model training purpose [14]. Otherwise, the trained model may not generalize very well for new data. Meanwhile, the noise in the sparse training data may sometimes render the model ineffectively because of the model overfitting. However, the sufficient well-labelled data are usually expensive in many scenarios, especially for the high-fidelity fracture predictions [15, 16]. Moreover, physics-informed neural networks (PINNs) have recently been implemented to overcome the low data availability [17]. PINNs assimilate physical governing equations into loss functions, which means PINNs are trained to satisfy the given training data as well as the imposed governing equations [18]. On the other hand, potentially, the transfer learning technique can also be leveraged to guide the training process with new training data that do not necessarily need to be large and complete. Liu et al. [19] applied transfer learning for data-driven knowledge extraction in fracture mechanics, allowing the efficient treatment of three-dimensional fracture problems based on two-dimensional solutions.

It is worth noting that different deep learning models have their unique hyperparameters that define the model architecture and govern the learning behaviors of the training algorithms, required to be determined with caution before training [20, 21]. Despite its importance, the hyperparameter selection process is usually omitted or not mentioned in most previous fracture prediction studies [10, 11]. As reported in [18], the weights of the loss-terms have to be modulated manually to strike a balance between the data-driven loss and the energy loss. The manually tuning process not only demands relevant knowledge and experience, but also takes more time. Moreover, the final tuned hyperparameters may not be suitable if the dataset is changed. To improve the practicability and reproducibility of the deep learning model, the automatic hyperparameter optimization, which is usually based on specific objectives, leading to the selection of the most appropriate hyperparameters and consequently the improved ultimate model performance [22], is introduced. Herein, the Bayesian optimization (BO) is applied [23]. BO is an effective method for solving functions that are computationally expensive to find the extrema, as well as solving a function which does not have a closed-form expression. Compared with other optimization methods, e.g., grid searching and random searching, BO uses less iteration to find the best optimal because the search space is probabilistically guided and reduced.

To predict the fracture evolution, we employ an ensemble technique to formulate a hybrid deep learning framework [24]. Three base models, namely, long short-term memory (LSTM), gated recurrent unit (GRU), and support vector regression (SVR), are chosen to form the final hybrid model. The novelty of this proposed framework lies in the use of Bayesian optimization to automatically complete hyperparameter tuning for each base model. In addition, instead of simply assessing and making comparisons between different base models [25, 26], this study utilizes the ensemble method to improve the performance and reliability of the final hybrid predictive model. Ensemble learning combines the forecasts from two or more base models, and has the mechanism to reduce the variance component of the prediction errors made by the each contributing models [27]. Empirically, ensembles can generate better predictive accuracy when there is a significant diversity among the individual base models [28].

The remaining sections of the paper are organized as follows. Section 2 provides the essential background knowledge of the base models. The Bayesian optimization is then outlined in Sect. 3, which is followed by the model framework elucidated in Sect. 4. A case study is adopted in Sect. 5 to demonstrate the performance of the proposed framework and discuss the predictive capacity difference by implementing different prediction strategies. Finally, concluding remarks are provided in Sect. 6.

2 Base models

In this study, the developed hybrid deep learning framework is composed by three base models, namely, LSTM, GRU, and SVR. The main selection criterion is their ability to process sequential data. The LSTM and GRU can capture the dynamics of sequential data and are able to retain the memory of previous patterns through a loop of feedback connections between network layers. Meanwhile, the SVR can also extract the inherent temporal characteristics of the given data and automatically learn the arbitrary mapping between inputs and outputs. The details of each base model will be elaborated in the following sub-sections.

2.1 Long short-term memory (LSTM)

IN principle, recurrent neuron network (RNN) is customized for capturing the relationships among sequential data via its unique recurrent hidden state [29]. However, such long-term dependency in sequence modeling is hard to be preserved by utilizing the simple RNNs due to the vanishing or exploding of stochastic gradients with long sequences [30]. Therefore, the LSTM architecture is proposed under the original framework of RNN to solve the problem by means of introducing both cell state and gating mechanisms [31].

The schematic drawing of a single LSTM unit at the time step t is presented in Fig. 1a. The cell state c_t remembers the values over arbitrary time intervals and the three gates (input gate i_t, output gate o_t, and forget gate f_t) work together to regulate the flow of temporal information into and out of the cell. Herein, we consider x_t as the input data and y_t as the output data. The relevant calculations within the LSTM unit are expressed by the following equations,

$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i + b_i\right) \tag{1}$$

$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f + b_f\right) \tag{2}$$

$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o + b_o\right) \tag{3}$$

$$\tilde{c}_t = tanh\left(x_t U^g + h_{t-1} W^g + b_c\right) \tag{4}$$

$$c_t = \sigma\left(f_t \odot c_{t-1} + i_t \odot \tilde{c}_t\right) \tag{5}$$

$$h_t = tanh\left(c_t\right) \odot o_t \tag{6}$$

In Eqs. (1, 2, and 3), the gating signals i_t, f_t, and o_t decide what information from the previous hidden state output h_{t-1} and the current new input x_t is discarded or is transmitted,

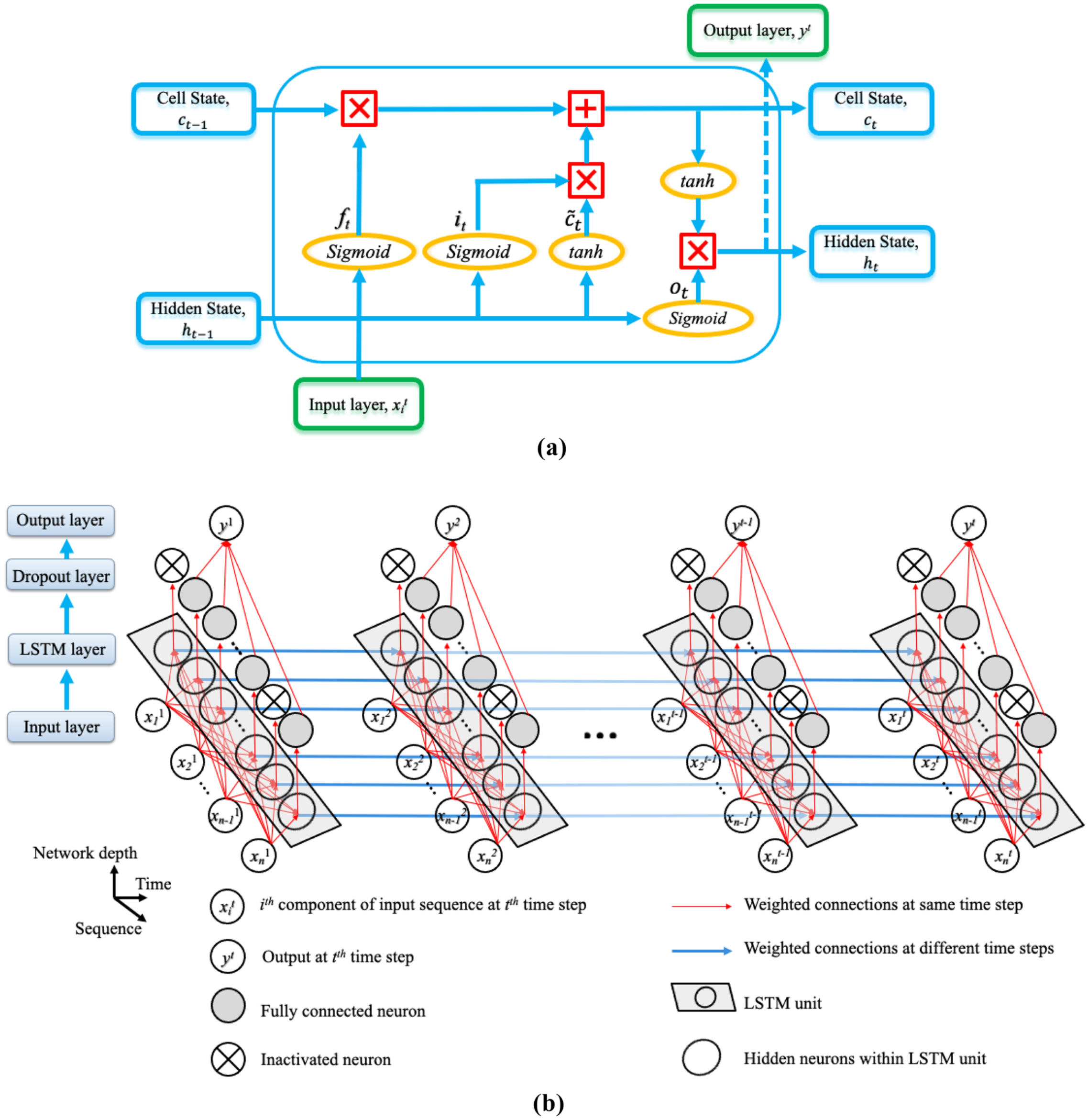


Fig. 1 **a** The internal architecture of a LSTM unit, **b** the three-dimensional data transformation within the LSTM deep learning model

and σ represents the sigmoid activation function to limit the gating signals within the range 0–1. Equation (4) is the replica of the gating signal equations to compute the internal memory cell state $\tilde{c}_t$ and replace the logistic function σ by the hyperbolic tangent activation function *tanh*. The weight matrices and bias vectors, denoted by $U^i, U^f, U^o, U^g; W^i, W^f, W^o, W^g; b_i, b_f, b_o$, and b_g, are all updated at each training step. The current cell state c_t and hidden layer state h_t at time t can be succinctly calculated using Eqs. (5 and 6), where $\odot$ is the Hadamard product.

A chain of the repeated LSTM units constitutes the typical structure of the LSTM layer along the time dimension. The three-dimensional schematic drawing of the input data x_t passing through a single LSTM layer is shown in Fig. 1b. To avoid overfitting and to increase the generality performance of the trained model, the dropout technique is applied [32]. The basic idea of dropout is to temporarily remove certain number of neurons from the network along with all its connections during the training process, as shown in Fig. 1b. Consequently, these remaining activated neurons become more robust to capture more general features instead of co-adaptation. Note that the thinner output can be further connected with one or several combined LSTM and dropout layers along the network depth dimension till the final output layer. The deeper network usually has more powerful capacity to process more complex sequential data.

2.2 Gated recurrent unit (GRU)

The GRU, as another particular variant of conventional RNN unit, plays the similar role as the LSTM [33]. The internal computing structure of the GRU is exampled in Fig. 2. The GRU simplifies gating signals by only using two gates, an update gate z_t and a reset gate r_t. Meanwhile, it mixes the cell state and hidden state. Due to the reduced gating system, the GRU has fewer tuning parameters during the training process, leading to a comparatively fast training speed than the LSTM. The application of the GRU is similar as the LSTM, by replacing all the LSTM units in the deep learning network. The relevant equations are listed below,

$$z_t = \sigma\left(x_t U^z + h_{t-1} W^z + b_z\right) \tag{7}$$

$$r_t = \sigma\left(x_t U^r + h_{t-1} W^r + b_r\right) \tag{8}$$

$$\tilde{h}_t = tanh\left(x_t U^h + (r_t \odot h_{t-1}) W^h + b_h\right) \tag{9}$$

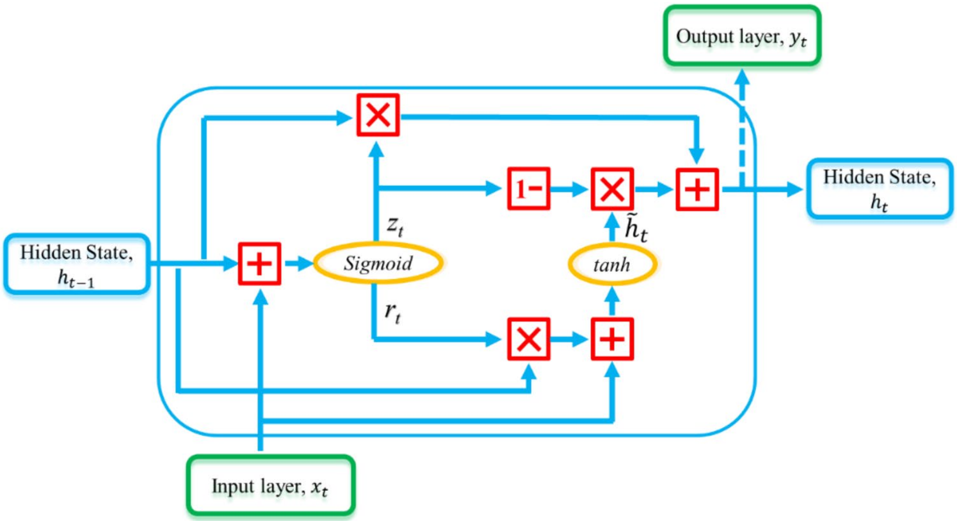
$$h_t = \left(1 - z_t\right) \odot \tilde{h}_t + z_t \odot h_{t-1} \tag{10}$$

where $U^z, U^r, U^h; W^z, W^r, W^h; b_z, b_r$, and b_h are the related weight matrices and bias vectors. Through the gate structure, the information embedded in the input layer x_t and previous hidden layer h_{t-1} is selected and screened, expressed as Eqs. (7 and 8). $\tilde{h}_t$ in Eq. (9) is called the candidate state computed by the input layer x_t, previous hidden state h_{t-1} and reset gate r_t. The new hidden state h_t, listed in Eq. (10), is updated by the linear interpolation between the previous state h_{t-1} and the current candidate state $\tilde{h}_t$ with the new temporal information of the input sequence.

2.3 Support vector regression (SVR)

SVR is capable of solving numerous regression problems [34]. The underlying concept for non-linear problems is mapping the original data into a feature space with higher dimensions and searching for the linear function that has the minimum reasonable intricacy to that specific feature space. Figure 3 states the illustration of the transformation of the SVR model. Given a set of the data $\left\{\left(x_1, y_1\right), \dots \left(x_i, y_i\right)\right\} \in R^n \times R$, where

Fig. 2 The schematic drawing of the GRU gate mechanisms



x_i is the input vector with n data patterns and y_i is the output value, the relation between x_i and y_i can be expressed as

$$y_i = f(x_i) = \omega\phi(x_i) + b \tag{11}$$

where $\phi(x_i)$ denotes the non-linear feature after mapping from the input vector x_i. ω and b are the SVR model parameters that are determined by minimizing the following optimization function,

$$\min_{\omega, b} \frac{\omega^2}{2} + C \times L_\varepsilon\left(f(x_i) - y_i\right), \tag{12}$$

where C is the regularization constant. L_ε is the ε-insensitive loss function as follows:

$$L_\varepsilon(z, \varepsilon) = \begin{cases} 0, if|z| \le \varepsilon; \\ |z| - \varepsilon, otherwise. \end{cases} \tag{13}$$

The loss equals zero if the forecasted value is within the ε-tube, as shown in Fig. 3c. The first term of the optimization function directly describes the flatness of Eq. (11), called as structural risk, while the second term named empirical risk indicates the difference between the predicted value and true value. Thereafter, the structural and empirical risks are adjusted by the regularization constant C. Both C and ε are user defined model hyperparameters.

Two positive slack variables ξ_i and $\widetilde{\xi}_i$ will be allocated to represent the distance from the actual values to the corresponding boundary values of the ε-tube. Then Eq. (12) can be rewritten as

$$\min_{\omega, b, \xi_i, \widetilde{\xi}_i} \frac{\omega^2}{2} + C\sum_{i=1}^{n}(\xi_i + \widetilde{\xi}_i), \tag{14}$$

with the constraints,

$$f(x_i) - y_i \le \varepsilon + \xi_i, \tag{15}$$

$$y_i - f(x_i) \le \varepsilon + \widetilde{\xi}_i, \tag{16}$$

$$\xi_i \ge 0, \widetilde{\xi}_i \ge 0, i = 1, 2, \ldots, n. \tag{17}$$

The above constrained convex optimization problem can be solved by the well-known Lagrangian multiplier strategy. Proper kernel function should also be imposed to overcome the contradiction between the high dimensional featured space and the computation complexity. The kernel function is considered as the model hyperparameter since different kernel functions can greatly affect the SVR model performance. For the sake of simplicity, only a brief introduction to SVR is given here and the detailed derivation process can be found in [35].

3 Bayesian optimization

Bayesian optimization method derived from Bayes' theorem is typically on the basis of Gaussian process (GP) [36]. A smooth GP prior distribution is first assumed over the unknown function and is combined with the sample information (observed evidence) to obtain the posterior of the function. The posterior information is used to find the next point to evaluate according to the acquisition function. The value of the acquisition function at a point characterizes the importance of evaluating that point to maximize the unknown function. The criterion is represented by an acquisition function. The next chosen point is determined to maximize the acquisition function.

The formalization of the above procedure is given as follows. We assume the unknown function $f(x)$ follows $GP(m(x), k(x, x'))$ for an arbitrary $x \in R$,

$$f(x) \sim GP\left(m(x), k\left(x, x'\right)\right), \tag{18}$$

where $m(x)$ is the Gaussian process mean function and $k(x, x')$ is the kernel function to quantify the similarity between points x and x'. The mean function $m(x)$ is set to be 0 for convenience. The kernel function is chosen as the squared exponential kernel,

$$k(x, x') = \exp\left[-\frac{1}{2}(x - x')^T (x - x')\right], \tag{19}$$

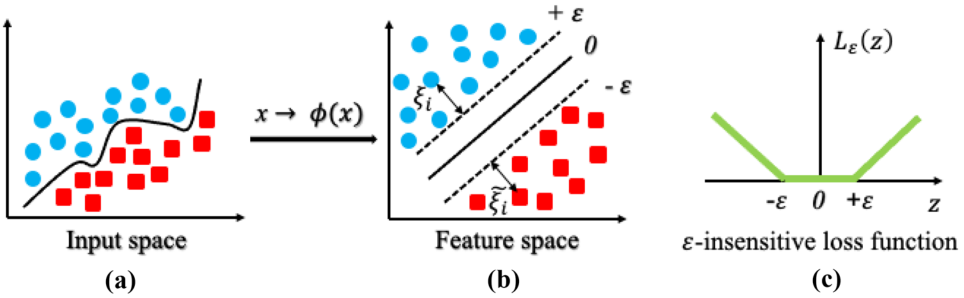


Fig. 3 Transformation process illustration of a SVR model

when x and x' get close to each other, and the value of $k\left(x,x'\right)$ approaches 1; otherwise, it approaches 0. It is noted that when two sampling points are close to each other, they have a strong correlation and a mutual influence; when they get further apart, the mutual influence is weak.

We denote the observation set $f_{1:t}=\{x_n,f_n\}_{n=1}^{t}, with f_n=f(x_n)$ after the t^{th} iteration. According to Eq. (19), the $t\times t$ kernel matrix can be represented as $\boldsymbol{K}$, where $\boldsymbol{K}_{i,j}=k\left(x_i,x_j\right)$, for all $i,j\in t$. Based on function $f(x)$, the posterior distribution at f_{t+1} at new point x_{t+1} can be given as $f_{t+1}\sim GP\left(\mu_{t+1},\sigma_{t+1}^2\right)$. By the properties of the joint Gaussian, the mean and standard deviation of the function f_{t+1} are given as,

$$\mu_{t+1}\left(x_{t+1}\right)=\mathbf{k}^{\mathbf{T}}\mathbf{K}^{-1}f_{1:t}, \tag{20}$$

$$\sigma_{(t+1)}^2(x_{(t+1)})=k^T K^{(-1)}k+k(x_{(t+1)},x_{(t+1)}), \tag{21}$$

where $\mathbf{k}=[k(x_{t+1},x_1),k(x_{t+1},x_2),\ldots,k(x_{t+1},x_t)]$.

As alluded to earlier, an acquisition function called expected improvement (EI), uses the above posterior distribution to select the next point. Herein, EI selects an x that maximizes $E[I(x)]$,

$$x=\ \text{argmax}\ E[I(x)]=\text{argmax}\ E\left[\max\left(0,f_{t+1}(x)-f(x^*)\right)\right], \tag{22}$$

where the degree of improvement $I(x)$ denotes the difference between the function at sample point value $f_{t+1}(x)$ and current optimum value $f(x^*)$. Note that $I(x)$ still satisfies the Gaussian distribution with the mean $\mu(x)-f(x^*)$ and the standard deviation $\sigma^2(x)$. The more detailed derivation processes can be found in [37].

4 Auto-tuned hybrid deep learning framework

As shown in Fig. 4a, we attempt to use previously observed fracture pattern evolution to train the proposed hybrid deep learning model and consequently achieve the goal of fracture dynamic prediction. The flow chart illustrated in Fig. 4b depicts the proposed hybrid deep learning framework, including four main stages—data preparation, Bayesian-based auto hyperparameter tuning, model training and optimization, and model performance evaluation. The implementation of the hybrid model is based on the concept of forecasting sequential data of distinctive

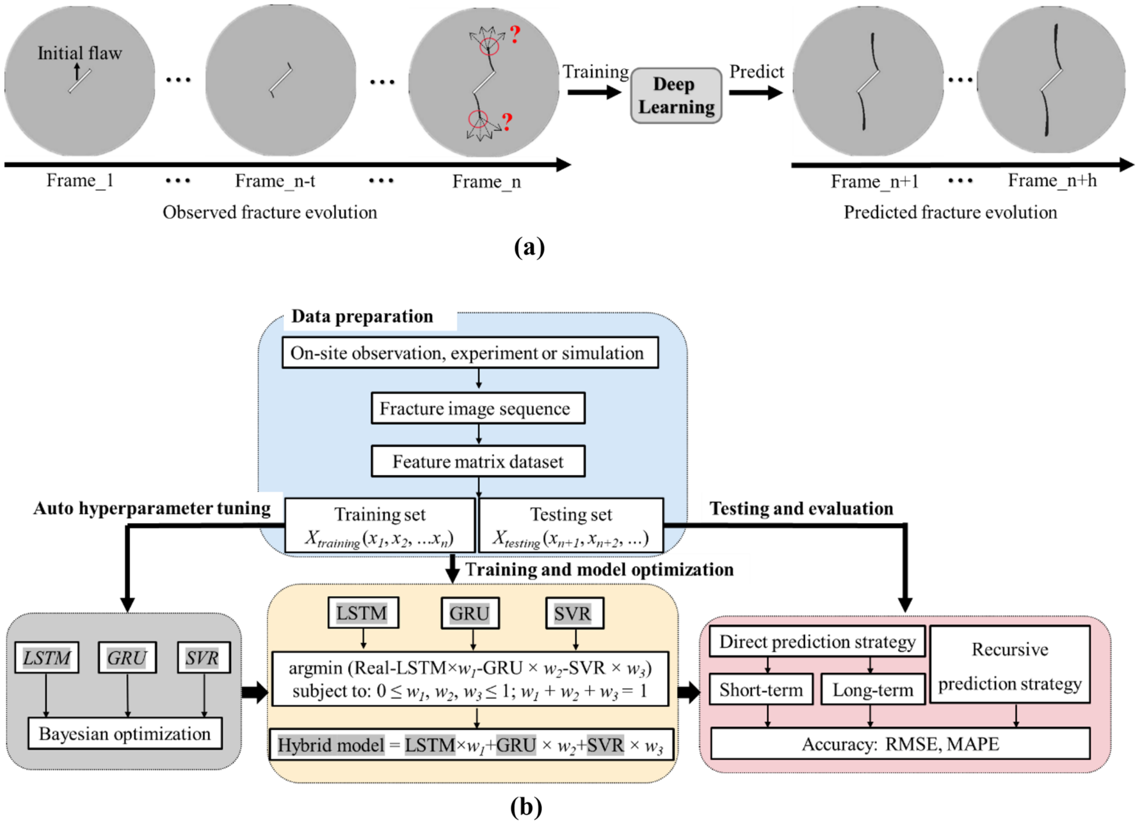

Initial flaw
Frame_1
Frame_n-t
Frame_n
Observed fracture evolution
Training
Deep Learning
Predict
Frame_n+1
Frame_n+h
Predicted fracture evolution
(a)
Data preparation
On-site observation, experiment or simulation
Fracture image sequence
Feature matrix dataset
Auto hyperparameter tuning
Training set
$X_{training}(x_1,x_2,\ldots x_n)$
Testing set
$X_{testing}(x_{n+1},x_{n+2},\ldots)$
Testing and evaluation
Training and model optimization
LSTM
GRU
SVR
Bayesian optimization
argmin (Real-LSTM×w_1-GRU × w_2-SVR × w_3)
subject to: $0\le w_1, w_2, w_3\le 1$; $w_1+w_2+w_3=1$
Hybrid model = LSTM×w_1+GRU × w_2+SVR × w_3
Direct prediction strategy
Short-term
Long-term
Recursive prediction strategy
Accuracy: RMSE, MAPE
(b)

Fig. 4 **a** Schematic of the fracture dynamic prediction problem by the deep learning method; **b** the flowchart of the proposed hybrid deep learning framework fracture features embedded in the consecutive fracture images.

At the first step, raw sequential image datasets of the fracture evolution are extracted from either the on-site observations, experiments or simulations. Then the selected fracture attributes are calculated by the image processing to form the feature matrix $F^{A \times M}$, where A denotes the number of the considered attributes and M represents the total number of the image frames. Note that feature attributes can have various physical quantity options, such as total damage, path length, fracture orientation, etc. [12]. The selection of the fracture attributes solely depends on the purpose of the prediction focus.

Then the raw feature matrix dataset is divided into training and testing data sets at a specific division ratio. Next, the considered hyperparameters are optimized based on the Bayesian optimization by implementing the training dataset for each base model. After the models are built, each base model is trained separately with the same training epoch. This is followed by the ensemble technique to further minimize the overall training error, which can be treated as the convex optimization shown below and solved by the Matlab built-in function [38].

$$\begin{aligned} &\text{argmin (Real} - \text{LSTM} \times w_1 - \text{GRU} \times w_2 - \text{SVR} \times w_3) \\ &\text{subject to}:\ 0 \leq w_1, w_2, w_3 \leq 1; \\ &w_1 + w_2 + w_3 = 1. \end{aligned} \tag{23}$$

where w_1, w_2, w_3 are the weight of each base model in the final generated hybrid model. The weight vector $w=(w_1, w_2, w_3)$ varies when the training dataset is changed.

In the last step, the hybrid model is used for testing using different prediction strategies, namely, direct prediction strategy (using short and long-term prediction models) and recursive prediction strategy. The definitions and main differences between different prediction strategies are elaborated in Sect. 5.1.2. The statistical measures, root mean square error (RMSE), and mean absolute percentage error (MAPE) are considered to quantitatively evaluate the model prediction accuracy. Their definitions are given as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\overline{x}_i - x_i\right)^2}, \tag{24}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\overline{x}_i - x_i}{x_i}\right| \times 100\%, \tag{25}$$

where N is the number of testing dataset; x_i and $\overline{x}_i$ are the i^{th} real and predicted values, respectively.

The developed hybrid deep learning framework is performed in Matlab R2020a platform using one CPU. The LSTM and GRU models both have a single LSTM layer or GRU layer connected with one dropout layer in this study.

5 Framework implementation and discussion

A complete case study using the numerical simulation results is carried out to evaluate the performance of the proposed hybrid deep learning framework. The details of each stage within the framework are provided in the following subsection. The comparisons between the hybrid deep learning model and standalone base model are made to demonstrate the efficiency and robustness of the proposed hybrid model. In addition, inspired by different practical application scenarios, different prediction strategies are discussed and compared.

5.1 Data preparation and prediction strategy

The following subsections provide the data acquisition process and introduce the essential background of different prediction strategies, as well as their corresponding real application scenarios.

5.1.1 Data acquisition

A sequence of the fracture evolution images is obtained from the phase-field simulation of crack propagation using the commercial Abaqus software. The phase-field model is based on the variational principle proposed by Francfort and Marigo [39]. Consider an arbitrary three-dimensional solid body $\Omega \subset \mathbb{R}^d$ $(d \in \{1,2,3\})$ with a crack surface Γ and external boundary $\partial\Omega$ which includes the Dirichlet boundary $\partial\Omega_u$ and the Neumann boundary $\partial\Omega_t$, the governing equation of the total energy is given by:

$$W(\Gamma, u) = \int_{\Omega} \psi(u, \varphi) d\Omega + \int_{\Gamma} g_c d\Gamma - \int_{\Omega} b \cdot u d\Omega - \int_{\partial\Omega_t} t \cdot u dS \tag{26}$$

Here $\boldsymbol{u}$, ψ, g_c, $\boldsymbol{b}$ and $\boldsymbol{t}$ are displacement, elastic energy density, energy release rate, body force and traction on the boundary, respectively. It is worth noting that $\int_{\Omega} \psi(\boldsymbol{u}, \varphi) d\Omega$ and $\int_{\Gamma} g_c d\Gamma$ are the strain energy and the surface energy, respectively, while the other two negative components represent the external work done by body force and traction on the boundary. The phase field, which represents damage variable, is introduced in the strain energy density function as follow:

$$\psi(\boldsymbol{u},\varphi) = (1-\varphi)^2 * \psi_0^+\left(\boldsymbol{\varepsilon}_+\right) + \psi_0^-\left(\boldsymbol{\varepsilon}_-\right), \tag{27}$$

where the elastic strain energy is decomposed to ψ_0^+ and ψ_0^-, which are defined as the tension part and the compression part, to avoid unrealistic crack by compression, defined as follow:

$$\psi_0^{\pm}(\boldsymbol{\varepsilon}(\boldsymbol{u})) = \frac{\lambda tr(\boldsymbol{\varepsilon})_{\pm}^2}{2} + \mu tr\left(\boldsymbol{\varepsilon}^2\right)_{\pm},$$

where λ and μ are elastic bulk modulus and shear modulus, respectively. Furthermore, a history field $\mathrm{H} = \max \psi_0^+\left(\boldsymbol{\varepsilon}_+\right)$ representing maximum positive reference energy is used for describing crack evolution.

The surface density function $\int_\Gamma d\Gamma$ is represented by introducing a fracture surface density $\gamma(\varphi, \nabla\varphi)$, of which further information about mathematical theory can be found in the work by Molnár and Gravouil [40]. Their definitions are given as follows:

$$\int_\Gamma d\Gamma = \int_\Omega \gamma(\varphi, \nabla\varphi) d\Omega, \tag{28}$$

$$\gamma(\varphi, \nabla\varphi) = \int_\Omega \left\{ \frac{1}{2l_c}\varphi^2 + \frac{l_c}{2}|\nabla\varphi|^2 \right\} d\Omega. \tag{29}$$

Consider the variation of energy and writing the equations in strong form, the governing equations of the phase-field method is formulated as

$$\begin{cases} \nabla \cdot \sigma + b = 0 in\Omega \\ \sigma \cdot n = t on\partial\Omega_t \\ 2(1-\varphi)\mathrm{H} - \frac{g_c}{l_c}\varphi + g_c l_c \Delta\varphi = 0 \\ u = \overline{\boldsymbol{u}} on\partial\Omega_u \\ \nabla\varphi \cdot n = 0 on\partial\Omega \end{cases} \tag{30}$$

The Brazilian test simulated by the phase-field approach is the compression of a single-flawed Brazilian disc made of three-dimensional-printed rock-like materials. A two-dimensional time-independent model with quasi-static loading condition is implemented for the simulation, of which the schematic drawing is given in Fig. 5a. The pre-existing crack has a length of 8 mm and a width of 0.6 mm. It is located at the disc center and orients at an angle of 45° along the loading direction. The material properties of this Brazilian disc include Young's modulus of 1.375 GPa and Poisson ratio of 0.285. In addition, the phase-field model is based on G-criterion and the critical energy release rate is chosen as 0.1 N/mm. After the convergence study, a mesh size of 0.1 mm is used in the model, and the characteristic length parameter is accordingly set as 0.2 mm. Detailed information of the foundation of the phase-field model and the numerical setup can be found in [41]. Figure 5b presents the final fracture patterns at the last time step. The crack paths from the simulation are in good agreement with our experimental results in Sharafisafa et al. [42]. Notably, the validated phase-field approach is more convenient to generate a comparatively large dataset for the further study, such as the prediction for cracking branching and kinking.

In this study, we aim to predict the entire fracture pattern evolution based on the previous observations. The evolving of the crack tip coordinates (x, y) are utilized to simply represent the spatiotemporal information with respect to the fracture pattern evolution. The inspiration

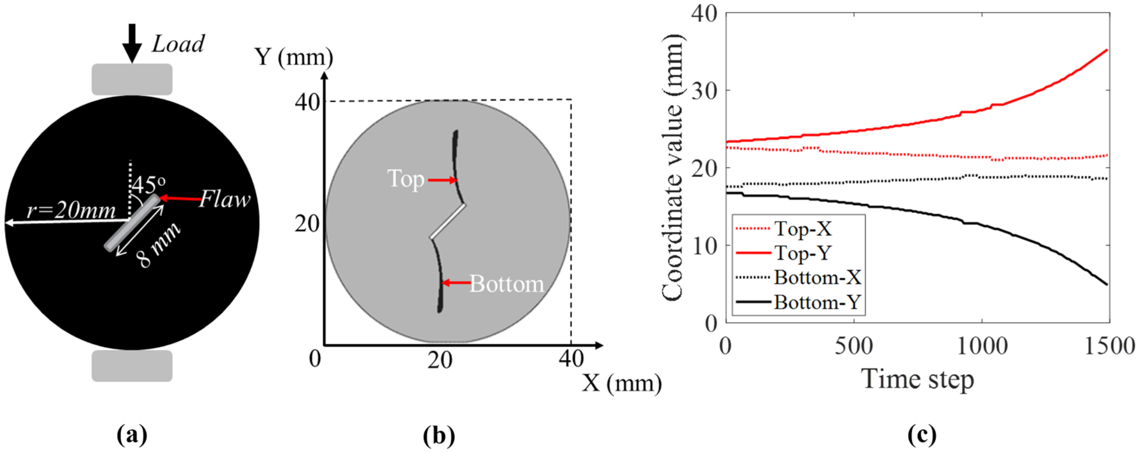


Fig. 5 **a** The schematic drawing of the simulated disc with a pre-existing flaw, **b** the final fracture patterns obtained from the phase field simulation, **c** the collected coordinate evolutions for the tips of the top and bottom cracks

of attribute selection in this paper comes directly from the feature attributes in [12], where mean fracture x and y locations are selected. The rule of thumb is that the selected attributes can completely represent the spatiotemporal information with respect to the fracture path development.

The crack tip coordinates (X_t, Y_t) at the t^{th} image frame are sequentially obtained by the image subtraction between the adjacent images. For the single fracture propagation, the coordinates x and y, are two independent values, suggesting that the x and y variables can be predicted individually. Four sequential datasets (Top-X, Top-Y, Bottom-X, Bottom-Y), for both the top and bottom cracks, are generated to form the feature matrix and denote the entire fracture pattern evolution of the pre-flawed Brazilian disc under compression. A total of 1,491 data points are collected, as shown in Fig. 5c. Although the overall trends of these four curves are smooth, the local region is quite tortuous, which will be elaborated in Sect. 5.3.1. It is worth noting that the image resolution may affect the collected data distribution and thus the prediction accuracy. An appropriate image resolution needs to be cautiously determined depending on the specific application scenarios.

5.1.2 Prediction strategy

Given the collected time series data of the crack tip coordinate, $x(x_1, x_2, _{...}x_t)$, the main goal is to predict the crack tip coordinates at next h steps $x(x_{t+1}, x_{t+2}, _{...}\ x_{t+h})$ based on the previous observations. Before constructing the deep learning method, the embedding dimension (or lagged value) needs to be determined for the time series data. Embeddings are the continuous vector representations of the discrete variables and an appropriate embedding dimension can reduce the computation time [43]. According to Cao's algorithm [44], the embedding dimension of the observed data series is calculated as 5 for all the four time series data. Hence, the correlation can be further formulated as $x(x_{t+1}, x_{t+2, \dots}\ x_{t+h}) = f(x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t)$ for the following discussion. In fact, as the time series data forecast, the longer the forecast, the more difficult it is to achieve an accurate prediction due to the increase in uncertainty and the accumulation of errors.

To address this issue, two diverse prediction strategies, namely, direct and recursive prediction strategies [25], are introduced and compared to demonstrate the capability of predictive models. When we customize the direct prediction strategy to forecast the crack tip location at the next h time steps, a set of h different predictive models need to be constructed and fed by the same input data, which are described as follows:

$$
\begin{aligned}
&\overline{x}_{t+1} = f_1\left(x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t\right)\\
&\overline{x}_{t+2} = f_2\left(x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t\right)\\
&\dots\\
&\overline{x}_{t+h} = f_h\left(x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t\right)
\end{aligned}
\tag{31}
$$

The representative formulation is expressed as $\overline{x}_{t+i} = f_i$ $(x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t)$, $i \in [1, h]$, to which $\overline{x}$ and x represent predicted and real values, respectively, and i is called prediction gap. Each individual prediction model f_i is trained to forecast the future single i^{th} time step, ultimately aggregating the predictions of the next h steps.

By contrast, the recursive prediction strategy usually only needs to train one single prediction model to achieve the predictions of the next h steps. Note that in most cases, this trained prediction model is referred to the f_1 prediction model with $i = 1$. The predictions are recursively calculated by constantly updating the input data based on the model predicted results. The form of the prediction procedure is given below:

$$
\begin{aligned}
&\overline{x}_{t+1} = f_1\left(x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t\right)\\
&\overline{x}_{t+2} = f_1(x_{t-3}, x_{t-2}, x_{t-1}, x_t, \overline{x}_{t+1})\\
&\dots\\
&\overline{x}_{t+h} = f_1\left(\overline{x}_{t+h-5}, \overline{x}_{t+h-4}, \overline{x}_{t+h-3}, \overline{x}_{t+h-2}, \overline{x}_{t+h-1}\right)
\end{aligned}
\tag{32}
$$

Based on the representative formulative expressions, Eqs (31) and (32), it is clear that the principal difference between these two strategies lies in whether the predicted results are used for the future prediction.

Moreover, for the direct prediction strategy, the number of the predictable time step h should be equal to the number of the trained prediction models because it is assumed by default that the fixed observation data are used to train the prediction models and make forecasting. In the previous studies, the h value has been mainly in the range 1–15, as reported in [25], which means that a maximum of 15 prediction models are trained. Considering the fact that the length of the testing dataset is usually greater than the number of trained prediction models in their studies, it remains quite unclear what is the input data during the model testing stage once the predicted time step exceeds the value of h.

To better address this issue and make the model testing stage clearer, we artificially divide the trained prediction models into three categories, namely, short-, medium-, and long-term prediction models depending on the i/n value, in which i is the model prediction gap and n represents the length of the training dataset. As illustrated in Fig. 6, if i/n is less than 1%, these trained prediction models are classified as short-term. By contrast, if i/n exceeds 10%, we call these kinds of prediction models as long-term. And the prediction model with the i/n value in between 1

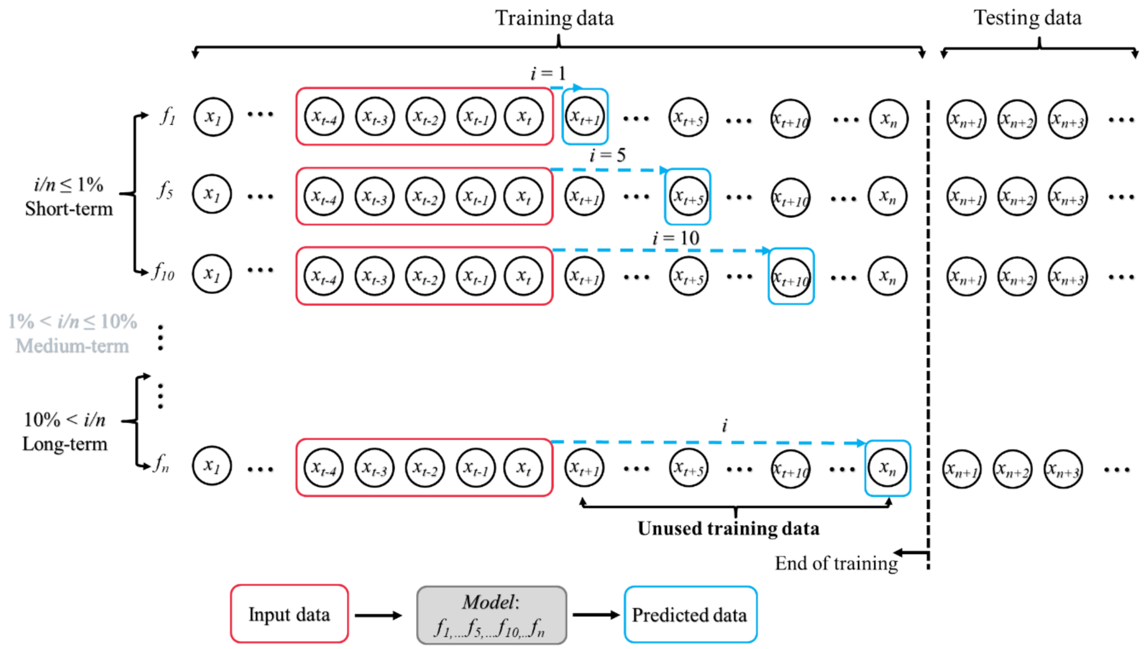


Fig. 6 The schematic drawings of different prediction models when the prediction gap varies

and 10% is denoted as medium-term, and is not included in this study.

For some slow propagating fractures, such as fatigue cracks and erosion cracks [45], we believe that the new observation data are concurrently accessible when the multi-step ahead predictions are made. Under this circumstance, the prediction can be made by recursively replacing the input data (x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t) in Eq. (31) with the newly observed data. The implementation of the trained prediction models is exemplified by f_1, where Eq. (31) is modified as

$$\begin{aligned}
&\overline{x}_{t+1} = f_1\left(x_{t-4}, x_{t-3}, x_{t-2}, x_{t-1}, x_t\right)\\
&\overline{x}_{t+2} = f_1\left(x_{t-3}, x_{t-2}, x_{t-1}, x_t, x_{t+1}\right)\\
&\ldots\\
&\overline{x}_{t+h} = f_1\left(\overline{x}_{t+h-5}, \overline{x}_{t+h-4}, \overline{x}_{t+h-3}, \overline{x}_{t+h-2}, \overline{x}_{t+h-1}\right)
\end{aligned} \tag{33}$$

And we believe that previous studies mostly fall into this category since only 1–15 prediction models are trained, as compared with the lengthy testing dataset. When the short-term prediction models are implemented to perform the long-run prediction, the new observation data are always required. Otherwise, this long-run prediction cannot be achieved.

However, with respect to the dynamic brittle fractures, the new observation data may not be available due to the fast crack growth rate. Hence, the prediction must be made purely based on the previous observation data in the training dataset without the involvement of the new observation data. Thus, the long-term prediction models may be more reasonable when applying the direct prediction strategy. As marked in Fig. 6, a portion of the training data is not used by the end of the training phase, and the dataset length is equal to the i value. Consequently, these irrelevant training data can be used as the input data when the prediction is made. And it should also be noted that the maximum predictable future time step is also equal to the i value.

In a nutshell, the direct prediction strategy usually needs one or more trained prediction models.

The number of the trained prediction models depends on how many multi-steps ahead predictions need to be made at the same time. However, for the purpose of a fair comparison with the recursive prediction strategy and investigating the performances among the prediction models with different prediction gaps, the following direct prediction strategy will only contain one single trained prediction model. It should also be admitted that the number of the trained prediction models is always limited. If the required predicted time step exceeds the model's prediction gap, three different solutions can be chosen. If the new observation data from the testing dataset is accessible, then the short-term prediction models can be applied in the context of the direct prediction strategy. And if the new observation data is not obtained, we can crudely increase the model prediction gap to generate the long-term prediction models and then implement the direct prediction strategy. The third option is to use the short-term prediction models to do the recursive prediction. All these conditions coincide with real application scenarios.

5.2 Auto hyperparameter optimization

Table 1 lists the selected tuned hyperparameters for LSTM, GRU, and SVR base models, respectively, along with the corresponding value sets or ranges. The number of the optimization trials is 30 for each model for a fair comparison. The mean squared error (MSE) of the training dataset is used as the objective function for the model selection, as shown below,

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(\overline{x}_i - x_i\right)^2, \tag{34}$$

where N is the number of testing dataset; x_i and $\overline{x}_i$ are the *ith* real and predicted values, respectively. At each optimization trial, the hyperparameter configuration with the lowest prediction error is chosen.

The optimized hyperparameters for the representative f_1 case are summarized in Table 2. Because of the uniqueness of the different datasets, even the same base model has totally different optimal hyperparameter configurations. The results are then used for the model construction and training in the next stage. The hyperparameters can continue varying once the prediction gap i is changed because the model output is changed. Hence, the auto hyperparameter tuning fits perfectly for the direct prediction strategy that requires a set of models due to the significant reduction of human effort.

5.3 Model training and performance evaluation

The initial 80% of the data series is selected to train the models, and the rest 20% is employed to evaluate the model performance. The raw training dataset is normalized in the range between 0 and 1 before training. Three base models are constructed and trained according to the optimal hyperparameters under different prediction gap conditions. The SVR model is performed to train the training set using the five-fold cross-validation approach. The number of the training epochs for the LSTM and GRU is set as 400 for training convergence.

The typical training processes of the LSTM and GRU for the Top-X case (when $i = 1$) are shown in Fig. 7a and b, respectively. It can be found that the RMSE converges around 0 during the model training process, which indicates that the model is well trained. Figure 7c presents the training performance comparisons for each base model in the Top-X case. Although the training results of the three models are comparatively good, the LSTM still has a slight advantage based on the error calculation, seeing Fig. 7d. According to Eq. (23), the final assigned weights for the LSTM, GRU, and SVR model are calculated as 0.9989, 0.0002, and 0.0009,

Table 1 Tuning hyperparameters of the different base models

Model	Hyperparameter	Value set or range	Type
LSTM/GRU	Number of neurons	[20 200]	Integer
	Dropout rate	[0 1]	Real
	Learning rate	[10^{-2} 1]	Real
	L2 Regularization	[10^{-10} 10^{-2}]	Real
SVR	Kernel function	(RBF, Poly, Sigmoid)	Categorical
	Regularization parameter (c)	[10^{-7} 10^{7}]	Real
	Kernel coefficient (gamma)	[10^{-7} 10^{7}]	Real

Table 2 The optimal hyperparameters for different base models when $i = 1$ using Bayesian optimization

Model	Hyperparameter	Datasets			
		Top-X	Top-Y	Bottom-X	Bottom-Y
f_{1_LSTM}	Number of neurons	120	114	60	125
	Dropout rate	0.17	0.08	0.09	0.47
	Learning rate	0.010	0.010	0.011	0.027
	L2 Regularization	2.7e-8	5.4e-5	1.3e-4	1.2e-6
f_{1_GRU}	Number of neurons	149	194	154	197
	Dropout rate	0.16	0.35	0.03	0.05
	Learning rate	0.012	0.095	0.011	0.27
	L2 Regularization	1.6e-6	2.6e-8	1.9e-10	1.3e-7
f_{1_SVR}	Kernel function	Poly	Sigmoid	Poly	Poly
	Regularization parameter (c)	74.92	0.10	14.52	0.0095
	Kernel coefficient (gamma)	0.087	15.95	0.35	88.92

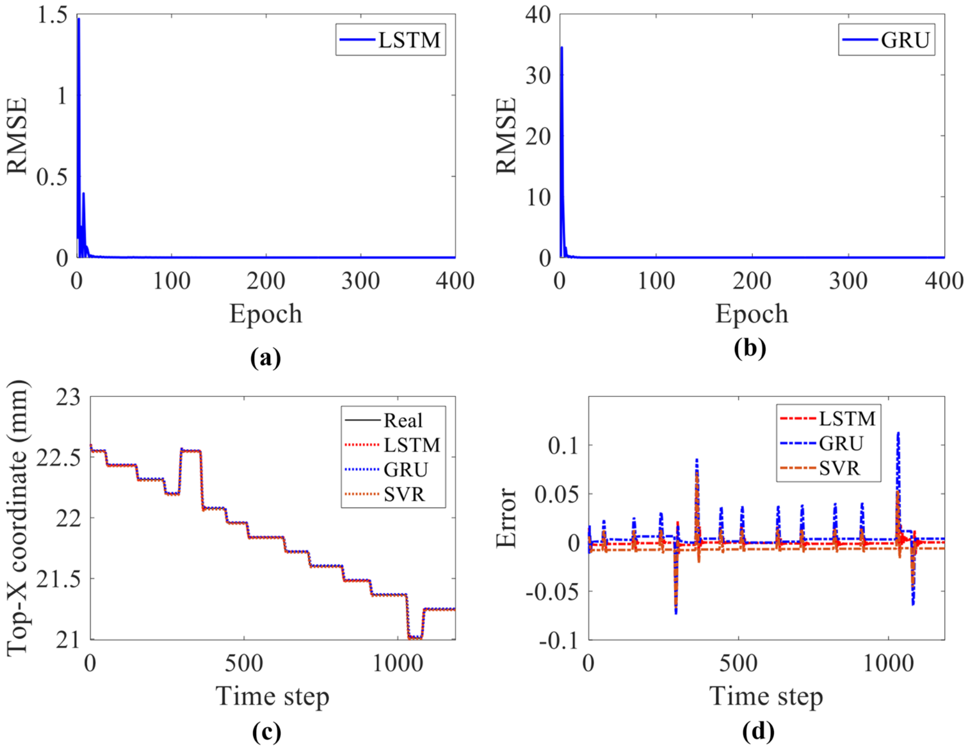


Fig. 7 Typical training process of the **a** LSTM and **b** GRU models for the Top-X case, **c** and (d) the corresponding training performance comparisons for each base model

respectively, to constitute the final hybrid model for this typical case. For other different cases, the weights of each respective base model can be automatically adjusted on account of their diverse training performances.

5.3.1 Short-term prediction models

In the following section, the short-term prediction models focus on the three cases, namely, one-step ($i=1$), five-step ($i=5$), and ten-step ($i=10$) ahead cases. In one-step ahead case, the proposed hybrid model is compared with each of the three base models to demonstrate the superiority. In addition, the influence of the model prediction gap on the prediction performance is discussed.

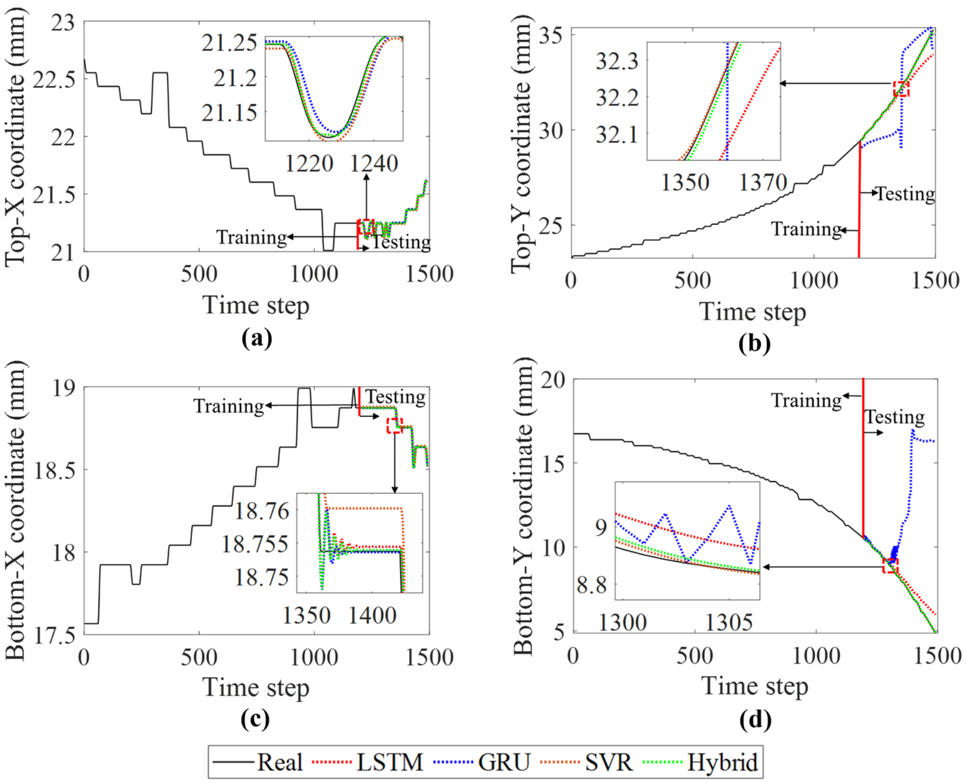
Figure 8 illustrates the prediction results for one-step ahead case. The individual base models have diverse prediction performance with respect to the different datasets. For the x coordinate predictions, despite the opposite trend appears in the testing data range, all the base models can achieve a satisfactory prediction accuracy. In contrast, the LSTM and GRU models present much worse predictions than the SVR model for the y coordinate predictions. It is appealing that the proposed hybrid model has relatively more stable prediction performances on the arbitrary datasets.

To further examine the reliability of our hybrid model, we pay close attention to the different cases by varying the prediction gaps, and the quantitative comparative results in terms of the RMSE and MAPE are depicted in Fig. 9. Although the hybrid model does not always have the best prediction accuracy for each single dataset at different prediction gap cases, it still offers competitive prediction results compared with the three base models. The detected subtle differences are mainly due to the fact that the final constructed hybrid model is based on the performance for the training dataset of each base model. For a certain base model and dataset, it is possible that it happens to have better prediction results on the testing dataset than the hybrid model although their training performance is unsatisfactory. Nevertheless, the hybrid model is able to achieve more outstanding holistic prediction performances and can forecast the entire fracture pattern dynamics, which is reflected by the overall values to the right of the dashed line in Fig. 9.

In general, the hybrid model can outperform the three individual base models on the prediction accuracy of the overall fracture pattern evolution. This brings one assumption that with more diverse datasets, the hybrid model may further prevail over any single base model to some extent. And in the following sections, only the hybrid model results are presented.

According to Fig. 9, the overall prediction accuracy decreases with the increment of the prediction gap i. To directly visualize the influence of the prediction gap on the prediction accuracy, the final predicted fracture patterns are given in Fig. 10a–c. As a whole, the predictions made by the short-term prediction models agree well with the real fracture pattern on a global scale as the zigzag fracture tip path is successfully predicted. The enlarged insets quantality indicate that prediction model with smaller

Fig. 8 The prediction results of the four datasets for the one-step ahead case



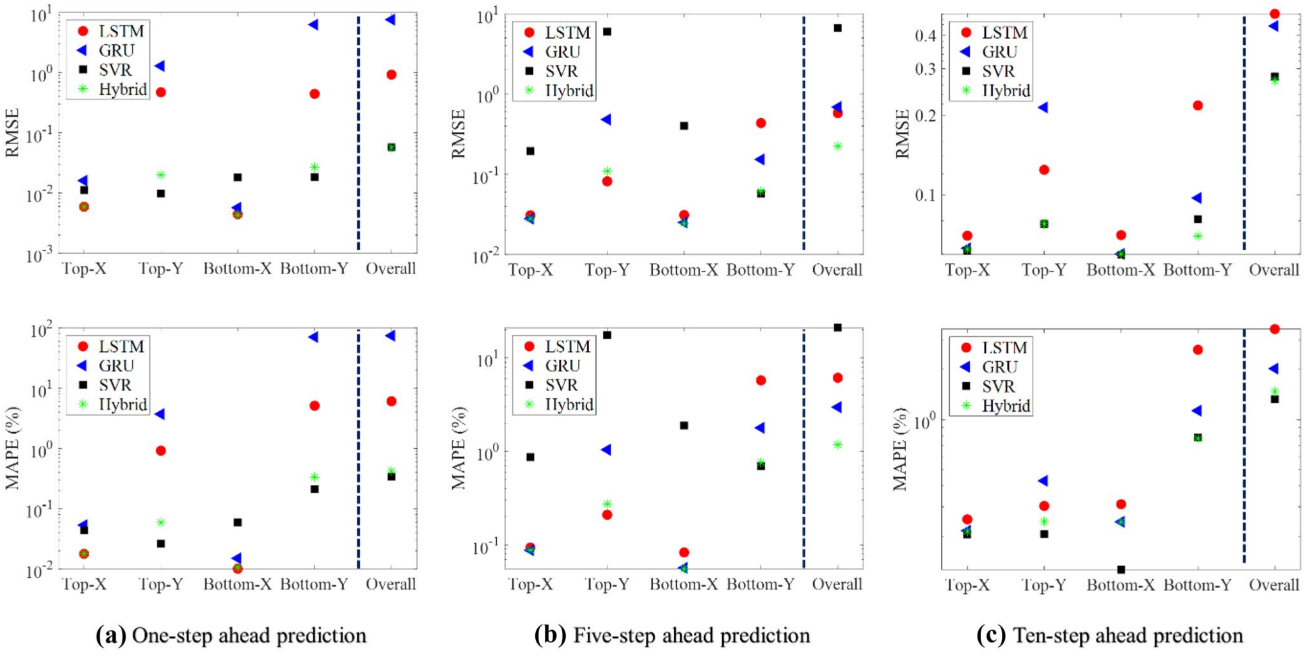


Fig. 9 The prediction results for the different prediction gap cases: **a** one-step ahead case, **b** five-step ahead case, **c** ten-step ahead case

prediction gap is more effective to forecast the detailed local fracture dynamics to a certain level.

The relevant quantitative analysis in Fig. 10d and e highlights that the predictive power of the short-term prediction models is affected by both the prediction gap and number

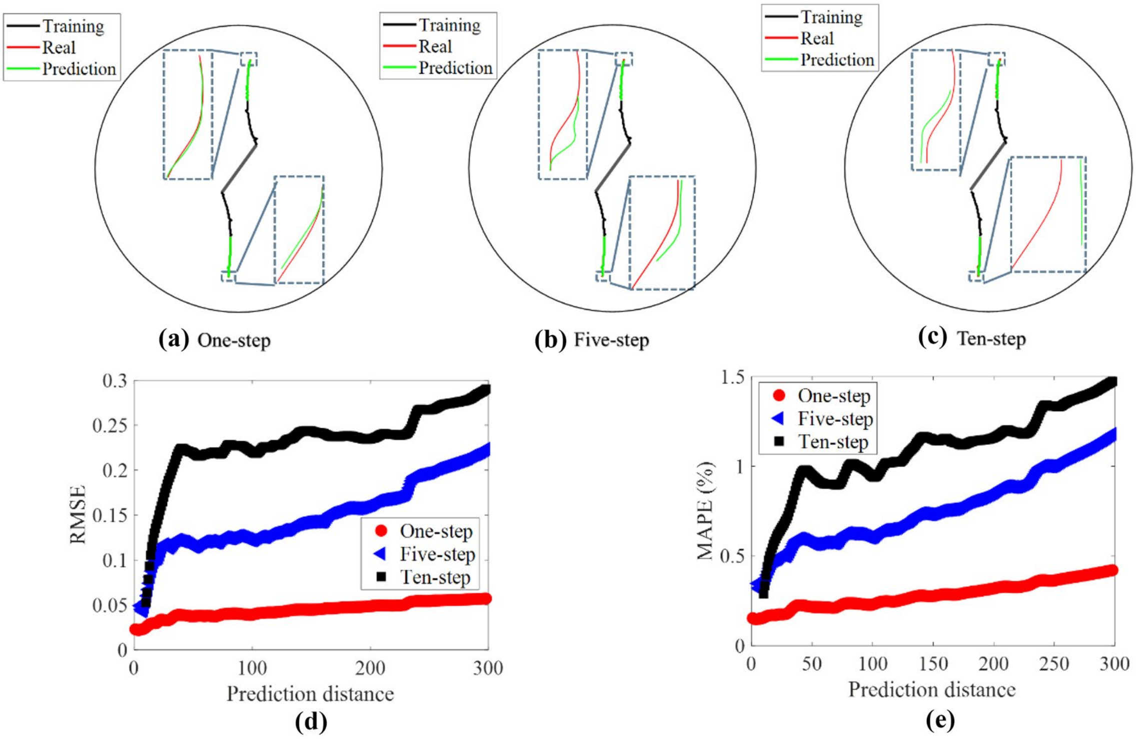


Fig. 10 Final fracture patterns predicted by the hybrid models for different prediction gap cases: **a** one-step ahead, **b** five-step ahead, **c** ten-step ahead; the performance evolutions of the hybrid model represented by **d** RMSE and **e** MAPE

of predicted time steps. Herein, the number of predicted time steps is called prediction distance for brevity. For all three cases here, increasing prediction distance can gradually attenuate the prediction accuracy to a certain degree. Considering the real applications, the new observation data may not be immediately available during the prediction process. Consequently, the prediction models with smaller prediction gaps cannot be used despite of their higher prediction accuracy. Instead, the prediction models with larger prediction gaps should be applied to guarantee the prediction efficiency. It is obvious that a trade-off should be made to achieve a balance between the prediction efficiency and accuracy. The adaptive prediction scheme may be a good solution for handling this situation because of the fluctuation state of the model performance (see Fig. 10d and e). Before the prediction starts, an error tolerance (represented by the statistical measures, like the RMSE and MAPE) should be predefined. During the prediction process, once the monitoring error exceeds the upper error tolerance, the prediction gap can automatically drop to a smaller value at a preset gap decrease rate. Then the hybrid model with the smaller predict gap is trained to predict the following steps. To enhance the efficiency, the prediction gap can also be increased according to a gap increase rate if the model with the smaller prediction gap can continuously have the predictive error below the lower error tolerance for several steps. The adaptive prediction scheme could be even more flexible when the error tolerance bound and gap rate are the function of the prediction distance.

5.3.2 Long-term prediction models

For the long-term prediction models, two cases are chosen for the intuitive comparison, $i/n = 10\%$ and $i/n = 20\%$, respectively. Since the long-term prediction models only adopt the unused training data as model input during the testing process, the maximum prediction distance is equal to the prediction gap i (Fig. 6). It is also worth noting that since the crack propagation speed increases with time, the maximum prediction distance of the 10% case is about one third of the total fracture length, while the 20% case corresponds to one half of the total fracture length. Meanwhile, to investigate the effect of the training data number on the prediction performance, the initial dataset with 1491 data points is amplified to 14,901 data points by uniform interpolation. Actually, the enlarged data points represent a higher data acquisition frequency during the fracture propagation. The embedding dimension of the 14,910 data points case is re-calculated and is found to remain the same value as that for the initial 1,491 data points case.

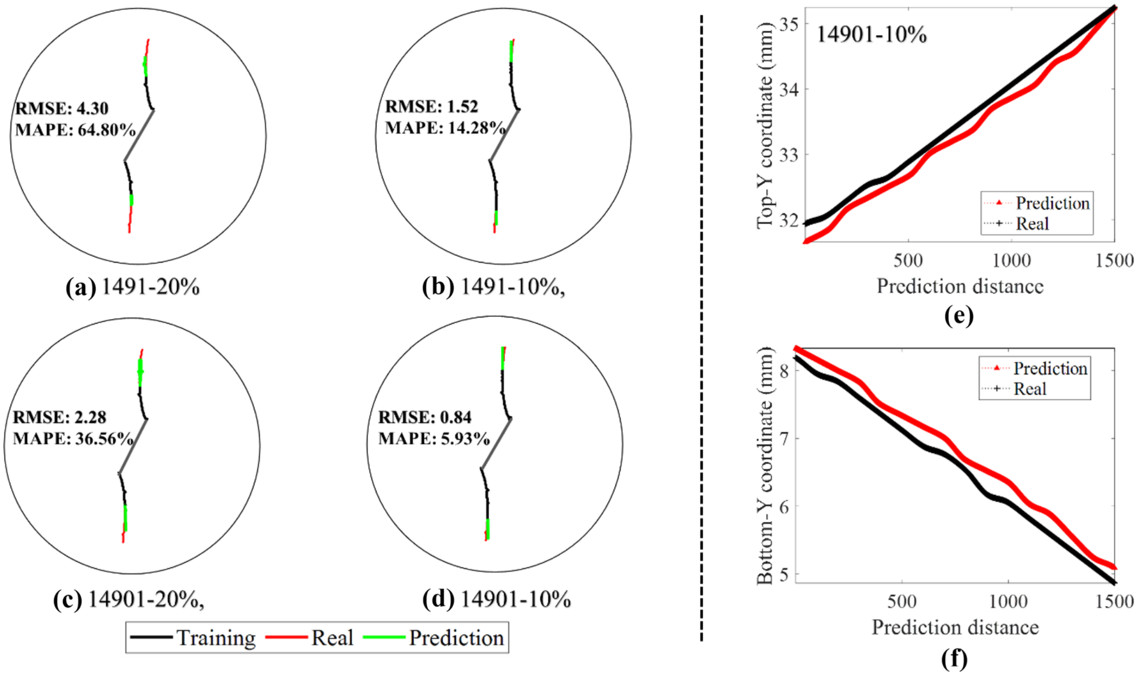


Fig. 11 The comparative results of different case for the long-term prediction scheme by implementing various datasets (**a**–**d**), the representative results of the y coordinate prediction results obtained from the 14,901-10% case (**e** and **f**)

Figure 11a–d demonstrate the final predicted fracture patterns for the different cases, along with the RMSE and MAPE values. The 14,901-10% case gives a more acceptable prediction than the others. It is highlighted that only a large amount of the training data can enable the predictive model to achieve the precise prediction of a considerably long crack propagation. Overall, the proposed hybrid model can capture the future fracture propagation behaviors, albeit the details of the twisted fracture tip are not completely matched. Figure 11e and f depicts the convergence prediction results of the y coordinates for the 14,910-10% case. The gradient of the evolving y value roughly reflects the fracture growth rate, considering that the x coordinate varies within a narrow range.

5.3.3 Prediction strategy comparison

To evaluate the performance of the different prediction strategies further visually, the 14,901-10% case is applied to illustrate the discrepancies, as shown in Fig. 12. Both the direct prediction strategy (short-term) and recursive prediction strategy utilize the one-step ahead hybrid prediction model to forecast the fracture propagation.

The prediction accuracy for the recursive prediction is much lower due to the fast error accumulation in the initial stage. The prediction errors of the initial steps can easily further accumulate in the following predictions, resulting in the decrease of accuracy in the long run. By contrast, according to the definition, the direct prediction can use different prediction model for different future steps. Therefore, the error accumulation can be avoided. We can infer that the recursive prediction is not the ideal solution for the prediction made for a long prediction distance. The similar statement is also reported in [25].

Besides, the direct prediction (short-term) possesses stronger quantitative predictive ability than the direct prediction (long-term). The result is consistent with the above discussion, where the increasing prediction gap has a negative influence on the prediction accuracy. If the new observation data are occasionally accessible during the long-distance prediction, the implementation of the short-term prediction (using the new observation data as input) can somehow amend the current long-term prediction. The coupled short- and long-term predictions can concurrently generate more detailed long-distance predictions.

5.4 Model discussion

The main advantages of the proposed hybrid model lie in its strong applicability and robustness. The hybrid model is automatically tuned and constructed to reduce the manual efforts. This is pivotal in practical applications because different fractures can create diverse datasets, which requires different hyperparameter configurations. In addition, the hybrid model is superior to the standalone base models in terms of the overall prediction accuracy. The parallel

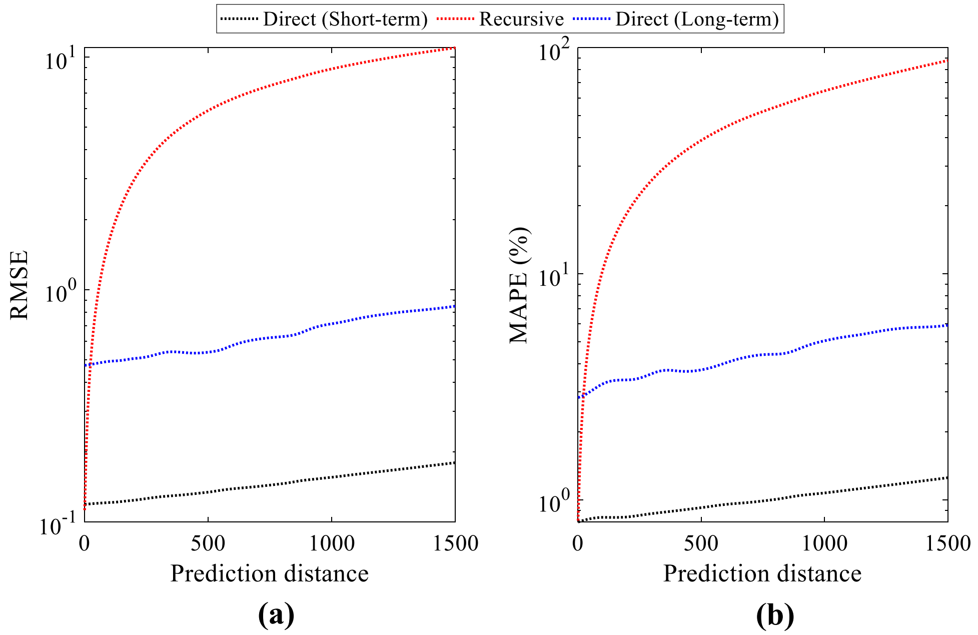


Fig. 12 The evolving prediction accuracy results of the different prediction strategies for the 14,901-10% case

architecture also highlights the efficiency and scalability characteristics of the developed framework, which facilitates the subsequent parallel computing when the dataset is greatly enlarged.

The comparison between the different prediction models shows that the short-term prediction models can offer more details, while the long-term prediction models can achieve the long-distance prediction but requires more training data. Due to required predictive precision differences for different tasks, the more flexible adaptive predictive scheme should be further explored because it has the potential to achieve a more efficient and accurate dynamic prediction.

6 Conclusion

This paper addresses the problem of the fracture evolution prediction using the deep learning method. We have proposed an automatically tuned hybrid deep learning approach to predict dynamic fracture propagation behaviors based on the previously observed fracture patterns. The key innovation is to implement ensemble technique with assistance of the Bayesian optimization to learn the complex temporal and spatial information imbedded in the image-based raw data structure. The case study demonstrates the advantages of this approach in terms of the applicability and predictive accuracy. For further prediction tasks, the adaptive prediction scheme by dynamically coupling the short- and long-term prediction models has shown its potential to generate a more efficient and accurate prediction. Since the developed auto-tuned hybrid deep learning model is designed for the time series data, besides the fracture prediction, it has wider application scenarios, including stock market forecasting, weather forecasting, sales forecasting, budgetary analysis, and so forth [35–37].

Despite the good performance of the proposed auto-tuned hybrid deep learning model, it is important to note that the current approach has certain limitations. First, it is still hard to directly predict the complex cracking branch phenomena by simply extending the model for multiple outputs. Multiple crack initiations and relative spatial locations need to be considered as new feature attributes included into the feature matrix. However, the appropriate representations for these feature attributes require further investigation. Inspired by the fashion of physics-informed neural networks, the addition of physical constrains might be a possible solution. Second, as a data-driven approach, the trained model is a 'black box'. It remains unclear what is learned during the training process. Thus, the generalization ability of the trained model cannot be well assessed. In other words, the trained model should be better explained and understood before its implementation in further studies.

Acknowledgements The work is supported in part by the Australian Research Council through Discovery Projects (DP190102954 and DP200101919). The authors acknowledge the use of the National Computational Infrastructure (NCI) which is supported by the Australian Government, and accessed through the Sydney Informatics Hub HPC Allocation Scheme, which is supported by the Deputy Vice-Chancellor (Research), University of Sydney. The authors also acknowledge the use of the University of Sydney HPC service at The University of Sydney.

Author contribution SJ: conceptualization, methodology, software, investigation, formal analysis, validation, writing—original draft preparation. ZC: resources, visualization, investigation. LY: resources, visualization, investigation. LS: conceptualization, investigation, writing—reviewing and editing, supervision, funding acquisition.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability The data that support the findings of this study are available from the authors upon reasonable request.

Declarations

Conflict of 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.

References

1. Abe A, Kim TW, Horne RN (2021) Laboratory hydraulic stimulation experiments to investigate the interaction between newly formed and preexisting fractures. Int J Rock Mech Min Sci 141:104665
2. Volkov G, Smirnov I (2022) A probabilistic approach to evaluate dynamic and static strength of quasi-brittle materials through high-rate testing. Int J Mech Sci 216:106960
3. Tomac I, Sauter M (2018) A review on challenges in the assessment of geomechanical rock performance for deep geothermal reservoir development. Renew Sustain Energy Rev 82:3972–3980
4. Li T, Rutqvist J, Hu M (2021) TOUGH-RFPA: coupled thermal-hydraulic-mechanical rock failure process analysis with application to deep geothermal wells. Int J Rock Mech Min Sci 142:104726
5. Bordeaux-Rego F et al (2021) Improvements on modelling wettability alteration by Engineered water injection: Surface complexation at the oil/brine/rock contact. Fuel 284:118991
6. Le LA et al (2018) Localised failure mechanism as the basis for constitutive modelling of geomaterials. Int J Eng Sci 133:284–310
7. Jin W, Arson C (2020) Fluid-driven transition from damage to fracture in anisotropic porous media: a multi-scale XFEM approach. Acta Geotech 15(1):113–144
8. Jalal M et al (2020) A new nonlinear formulation-based prediction approach using artificial neural network (ANN) model for rubberized cement composite. Eng Comput 38:238–300
9. Maleki E, Unal O (2021) Fatigue limit prediction and analysis of nano-structured AISI 304 steel by severe shot peening via ANN. Eng Comput 37(4):2663–2678
10. Nguyen-Le DH et al (2020) A data-driven approach based on long short-term memory and hidden Markov model for crack propagation prediction. Eng Fract Mech 235:107085
11. Fernández-Godino MG et al (2021) Accelerating high-strain continuum-scale brittle fracture simulations with machine learning. Comput Mater Sci 186:109959
12. Schwarzer M et al (2019) Learning to fail: Predicting fracture evolution in brittle material models using recurrent graph convolutional neural networks. Comput Mater Sci 162:322–332
13. Pierson K, Rahman A, Spear AD (2019) Predicting microstructure-sensitive fatigue-crack path in 3D using a machine learning framework. Jom 71(8):2680–2694
14. Hsu Y-C, Yu C-H, Buehler MJ (2020) Using deep learning to predict fracture patterns in crystalline solids. Matter 3(1):197–211
15. Moore BA et al (2018) Predictive modeling of dynamic fracture growth in brittle materials with machine learning. Comput Mater Sci 148:46–53
16. Wang Y et al (2021) StressNet-Deep learning to predict stress with fracture propagation in brittle materials. npj Mater Degradation 5(1):1–10
17. Zheng B et al (2022) Physics-informed machine learning model for computational fracture of quasi-brittle materials without labelled data. Int J Mech Sci 223:107282
18. Goswami S et al (2022) A physics-informed variational DeepONet for predicting crack path in quasi-brittle materials. Comput Methods Appl Mech Eng 391:114587
19. Liu X et al (2021) Knowledge extraction and transfer in data-driven fracture mechanics. Proc Natl Acad Sci 118(23):e2104765118
20. Shankar K et al (2020) Hyperparameter tuning deep learning for diabetic retinopathy fundus image classification. IEEE Access 8:118164–118173
21. Shukla V et al (2020) Artificial neural network based predictive negative hydrogen ion helicon plasma source for fusion grade large sized ion source. Eng Comput 38:347–364
22. Kotthoff L et al (2019) Auto-WEKA: Automatic model selection and hyperparameter optimization in WEKA. Automated Machine Learning. Springer, Cham, pp 81–95
23. Frazier PI (2018) A tutorial on Bayesian optimization. arXiv preprint arXiv:1807.02811
24. Wolpert DH (1992) Stacked generalization. Neural Netw 5(2):241–259
25. Nguyen H-P, Liu J, Zio E (2020) A long-term prediction approach based on long short-term memory neural networks with automatic parameter optimization by Tree-structured Parzen Estimator and applied to time-series data of NPP steam generators. Appl Soft Comput 89:106116
26. Wang J, Li Y (2018) Multi-step ahead wind speed prediction based on optimal feature extraction, long short term memory neural network and error correction strategy. Appl Energy 230:429–443
27. Polikar R (2012) Ensemble learning. Ensemble machine learning. Springer, pp 1–34
28. Nguyen HT, Le Nguyen M (2019) An ensemble method with sentiment features and clustering support. Neurocomputing 370:155–165
29. Medsker LR, Jain L (2001) Recurrent neural networks. Design Applications 5:64–67
30. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Networks 5(2):157–166
31. Breuel TM (2015) Benchmarking of LSTM networks. arXiv preprint arXiv:1508.02774
32. Nandini GS, Kumar AS, Chidananda K (2021) Dropout technique for image classification based on extreme learning machine. Global Transitions Proc 2(1):111–116

33. Dey R, Salem FM (2017) Gate-variants of gated recurrent unit (GRU) neural networks. in 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS). IEEE
34. Awad M, Khanna R (2015) Support vector regression. Efficient learning machines. Springer, pp 67–80
35. Ahmad MS et al (2020) A novel support vector regression (SVR) model for the prediction of splice strength of the unconfined beam specimens. Constr Build Mater 248:118475
36. Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. Adv Neural Info Process Syst 25:2960–2968
37. Kopsiaftis G et al (2019) Gaussian process regression tuned by bayesian optimization for seawater intrusion prediction. Comput Intell Neurosci. https://doi.org/10.1155/2019/2859429
38. Available from: https://www.mathworks.com/help/optim/ug/fmincon.html.
39. Francfort GA, Marigo J-J (1998) Revisiting brittle fracture as an energy minimization problem. J Mech Phys Solids 46(8):1319–1342
40. Molnár G, Gravouil A (2017) 2D and 3D Abaqus implementation of a robust staggered phase-field solution for modeling brittle fracture. Finite Elem Anal Des 130:27–38
41. Cheng Z (2020) Modelling brittle fractures with finite elements: a time-independent phase-field model. The University of Sydney, Master thesis. https://doi.org/10.25910/Z3NK-5D95
42. Sharafisafa M, Shen L, Xu Q (2018) Characterisation of mechanical behaviour of 3D printed rock-like material with digital image correlation. Int J Rock Mech Min Sci 112:122–138
43. Rüdisüli M et al (2013) Measurement, monitoring and control of fluidized bed combustion and gasification. Fluidized bed technologies for near-zero emission combustion and gasification. Elsevier, pp 813–864
44. Cao L (1997) Practical method for determining the minimum embedding dimension of a scalar time series. Physica D 110(1–2):43–50
45. Xu Y, Wan W, Dunne FP (2021) Microstructural fracture mechanics: stored energy density at fatigue cracks. J Mech Phys Solids 146:104209