A new natural fracture width prediction method based on fluid dynamics constrained neural network

Cite as: Phys. Fluids **36**, 086618 (2024); doi: 10.1063/5.0216197
Submitted: 28 April 2024 · Accepted: 8 August 2024 ·
Published Online: 23 August 2024

Junqiao Liang (梁君巧),[1] Hu Yang (杨虎),[1,a)] Shilong Xiong (熊仕龙),[1] Donghua Zhang (张栋桦),[1] Xinlong Xian (线新龙),[1] and Penggao Zhou (周鹏高)[2]

AFFILIATIONS

[1]The College of Petroleum, China University of Petroleum (Beijing) at Karamay, Karamay, Xinjiang 834000, China
[2]Karamay Vocational and Technical College, Karamay, Xinjiang 834000, China

[a)]Author to whom correspondence should be addressed: 411195188@qq.com

ABSTRACT

The width of natural fractures is an important parameter in the leak prevention and plugging operations for oil and gas drilling. To accurately predict the fracture width of the leaking formation when well leakage occurs during oilfield drilling, based on the mud logging data of adjacent oil wells during the loss process, the Spearman correlation analysis and data normalization methods were used to preprocess the data. A new method was then established to predict the natural fracture width based on FDCNN (Fluid Dynamics Constrained Neural Network), which is a neural network algorithm constrained by the prior knowledge of fluid dynamics. In this method, first, based on the fluid dynamics prior knowledge that there is a strict positive partial derivative relationship between the natural fracture width and the leakage volume and pressure difference in the fluid dynamics model, the constraint conditions of the neural network were optimized. Second, the augmented Lagrange multiplier method was used to establish the performance index of the neural network through a multiplier and a penalty factor. Finally, the model was trained using the backpropagation learning rule and gradient descent training methods. The results indicate that this method, utilizing FDCNN, can train the model with a small sample training set, demonstrating superior generalization ability and prediction accuracy compared to traditional fluid dynamics models and Data-Driven Neural Network (which refers to an algorithm training a neural network solely on data samples without using fluid dynamics prior knowledge). This approach can effectively reduce the prediction error of fracture width, providing valuable reference for the development of field plugging programs.

I. INTRODUCTION

In the field of oil and gas exploration, when drilling encounters fractured rock strata, it will cause a large amount of drilling mud to seep into the strata, leading to significant economic losses. Therefore, accurately measuring the width of natural fractures in fractured strata is crucial for selecting effective plugging materials and determining reasonable material gradation.[1–3] Although techniques such as imaging logging, acoustic logging, and nuclear magnetic resonance exist to identify fracture width, these methods are not only costly but may also be inapplicable in some cases due to complex downhole conditions.[4] Therefore, it is essential for mud loss prevention and plugging to utilize the drilling mud logging data during well leakage to establish an inversion model for fracture width and quickly identify connected fractures and their widths.

Currently, the main methods for inverting fracture width based on well leakage statistics include calculation models based on fluid dynamics models and inversion models based on artificial neural networks.[5–10] Van Golf-Racht *et al.*[6] derived a formula for calculating fracture width based on the plate model and Darcy's law, giving the relationship between fracture porosity, fracture permeability, and fracture width. Lietard *et al.*[11] studied the leakage problem of Bingham fluid in fractures, established a leakage pressure model for a single infinite fracture under plane laminar flow conditions, introduced dimensionless time and dimensionless radius, and successfully predicted the fracture width by plotting leakage characteristic curves with actual drilling data. Sanfillippo *et al.*[12] assumed that the drilling fluid is a Newtonian fluid and the wellbore is a constant pressure boundary. They combined Poiseuille's law with the diffusion equation to establish a relationship between fracture width, leakage volume, fracture porosity, leakage pressure difference, and wellbore radius. Verga *et al.*[13] established a fluid leakage model in fractures under a plate model based on the diffusion equation under steady-state conditions and derived the relationship between fracture width and mud loss volume, drilling mud viscosity, and wellbore radius. Based on the Lietard model, Griffiths *et al.*[14] conducted further equation derivation and arrived at a cubic equation that can directly solve for the width of natural fractures, effectively overcoming the cumbersome issue associated with the traditional chart method. Olson *et al.*[15] conducted experimental analyses based on field experience in shale gas and some tight gas sandstones to explore the significant interaction between hydraulic fractures and natural fractures, particularly those that are cemented (or healed), revealing complex interaction patterns between hydraulic fractures and natural fractures during their propagation processes. Li *et al.*[16] further validated the values of parameters in the Sanfillippo model and conducted drilling fluid loss experiments using a high-temperature and high-pressure drilling fluid loss dynamic evaluation instrument. Through fitting the experimental data, they obtained the key parameters required for model calculations. Zhao *et al.*[17] improved the Lietard–Griffiths model, studied the numerical solution of fracture width for the Lietard–Griffiths model based on the Newton iteration method, and validated the accuracy of the improved method using field logging data, overcoming the issue of low efficiency associated with the chart method. Chen[18] established a fracture width calculation model based on the rock mechanics fracture width calculation model, optimized the input feature values of the neural network, and established a BP (backpropagation) neural network-based formation fracture width calculation model with fracture width as the output target value. Xu *et al.*[19] studied the effects of differential pressure, consistency coefficient, flow index, and boundary distance of faults on the dynamic fracture width, and set up a model for predicting the dynamic fracture width, and applied the model to fracture width prediction of fault fractures, which achieved good results. Wang *et al.*[20] used genetic algorithm (GA) and Adaboost algorithm to optimize the BP neural network, overcoming the problem of easily falling into local minimum values, and established an Adaboost-GA-BP neural network prediction model to predict the width of reservoir fractures. The results show that the prediction accuracy of the model has been improved.

Although non-mechanistic models represented by neural networks have been widely used in the prediction of natural fracture width, such models only predict and model fracture width based on data features, lacking consideration of the physical laws and prior knowledge contained in natural fractures.[21–23] The methodological foundation of this study is Physics Informed Machine Learning, which is a method of integrating physical laws into machine learning models and has currently become a hot topic in machine learning research. This method can achieve good prediction results with a small number of training samples.[24–26] Common strategies include the following: (1) Initialization strategy guided by physical laws, which uses simulated data generated by mechanistic prediction models for pre-training of neural networks. This strategy can solve the problem of sparse observations. However, generating simulated data using mechanistic models requires empirical determination of model parameters; (2) Design of network loss functions guided by physical laws, which adds relevant penalty terms to the loss function of the neural network based on physical laws. This strategy is applicable to most neural networks due to its low coupling characteristics, and the form of the loss function is easy to understand and interpret.[27–32]

This paper will combine the prior knowledge of partial derivatives relationships in the classical fluid dynamics models commonly used in fracture width prediction to guide the design of the neural network's loss function, and establish a new method of fracture width prediction for fluid dynamics prior knowledge constrained neural network (the method is trained based on the FDCNN algorithm, FDCNN refers to a neural network algorithm constrained by the prior knowledge of fluid dynamics). To solve the problem of not being able to quickly obtain large amounts of training data on the oilfield site, FDCNN has solved the problem of improving the generalization ability and prediction accuracy of the neural network in the case of relatively small data sample size, so that the neural network can truly be applied to the prediction of fracture width in drilling leakage. The application results show that when the training sample is insufficient, the relative error of the fracture width predicted by the neural network trained by FDCNN is significantly smaller than that of the neural network trained by DDNN (DDNN refers to an algorithm of training a neural network using only data samples without using the fluid dynamics prior knowledge) and the classical fluid dynamics model.

The subsequent sections of this paper are organized as follows: Sec. II delves into the fluid dynamics model and elaborates on the method for solving fracture width based on this model. Section III discusses the sources and processing methods of data and conducts a thorough analysis of the main influencing factors of fracture width. Building on the analysis in Secs. II and III, Sec. IV constructs the training algorithm for FDCNN. Section V performs performance testing and application analysis on FDCNN. Finally, Sec. VI provides a comprehensive and concise summary of this paper.

II. FLUID DYNAMICS MODEL FOR PREDICTION OF FRACTURE WIDTH

At present, the commonly used fluid dynamics models for predicting natural fracture width are Sanfillippo model, Lietard–Griffiths model, and Verga model. The Sanfillippo model is limited to Newtonian fluid and does not apply to non-Newtonian ordinary drilling fluids. Therefore, this study primarily focuses on introducing the Lietard–Griffiths model and the Verga model, which can be utilized for comparative analysis with the FDCNN-based approach. The basic assumptions of these two models are as follows: (1) Assume that the rheological model of non-Newtonian fluid drilling fluid is Bingham fluid; (2) the fracture surface is smooth and the fracture width is certain; (3) every time lost circulation occurs, and it is only related to a single horizontal fracture; (4) the leakage of drilling fluid in the wellbore is along the radial laminar flow; and (5) ignoring the matrix porosity does not consider the fracture surface leakage.[11,13,14]

A. Lietard–Griffiths model

The model by Lietard *et al.*[11,12] builds upon Darcy's law, elucidating mud flow through a fracture by resolving the equation for local pressure loss incurred by laminar plastic fluid flow in a slot of width w. The internal and external radii of the mud-invasion fracture are denoted by r_w and $(r_s)_{\max}$, respectively, as shown in Fig. 1.

The local pressure drop is (Lietard *et al.* 1999, 2002)

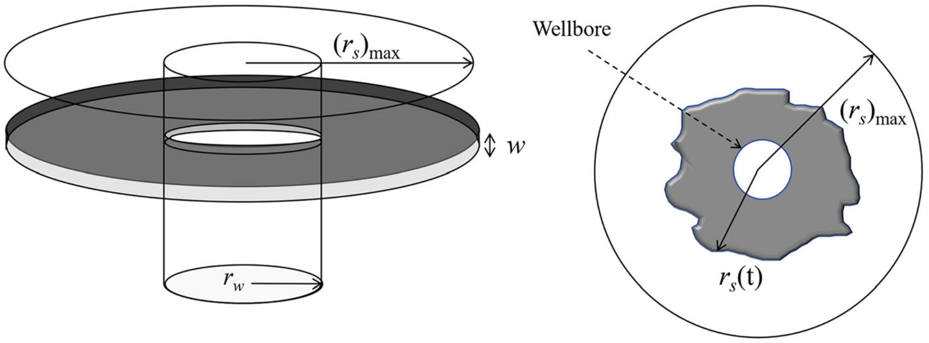

FIG. 1. Mud invasion into a slot of width *w*.

$$\frac{\mathrm{d}p}{\mathrm{d}r} = \frac{12\mu_p v_m}{w^2} + \frac{3\tau_y}{w}, \tag{1}$$

where μ_p is the plastic viscosity, τ_y is the drilling mud yield value, and v_m denotes the local velocity of the mud in the fractures under the radial-flow conditions around the well, given by

$$v_m(t) = \frac{q_m(t)}{2\pi r w}. \tag{2}$$

If V_m represents the cumulative volume of mud loss at a given time, then the volumetric rate of mud loss q_m is given by

$$q_m(t) = \frac{\mathrm{d}V_m(t)}{\mathrm{d}t}. \tag{3}$$

The cumulative volume of mud loss V_m is given by

$$V_m(t) = \pi w\left\{[r_s(t)]^2 - r_w^2\right\}, \tag{4}$$

where $r_s(t)$ is the invasion radius r_s at time t.

Thus, substituting Eqs. (2) and (3) into Eq. (1) results in

$$\frac{\mathrm{d}p(t)}{\mathrm{d}r} = \frac{6\mu_p}{\pi r w^3}\frac{\mathrm{d}V_m(t)}{\mathrm{d}t} + \frac{3\tau_y}{w}. \tag{5}$$

Integration of Eq. (5) over the mud-invasion region extending from the wellbore at r_w to the depth of mud-invasion radius $r_s(t)$ yields

$$\Delta p = \frac{6\mu_p}{\pi w^3}\frac{\mathrm{d}V_m(t)}{\mathrm{d}t}\ln\left[\frac{r_s(t)}{r_w}\right] + \frac{3\tau_y}{w}[r_s(t) - r_w], \tag{6}$$

where Δp is the difference between circulating pressure and the static reservoir pressure.

Substituting Eq. (4) into Eq. (6) yields an expression for the drilling overpressure Δp as

$$\Delta p = \frac{12\mu_p}{w^2} r_s(t)\ln\left[\frac{r_s(t)}{r_w}\right]\frac{\mathrm{d}r_s(t)}{\mathrm{d}t} + \frac{3\tau_y}{w}[r_s(t) - r_w]. \tag{7}$$

The dimensionless mud-invasion radius and time are defined as

$$r_D = \frac{r_s}{r_w}, \tag{8}$$

$$t_D = \frac{t}{t_c}, \tag{9}$$

respectively, in which the characteristic timescale is taken as

$$t_c = \frac{3\mu_p}{\Delta p (w/r_w)^2}. \tag{10}$$

In addition, a new parameter α_D, referred to as the dimensionless mud-invasion factor, is defined as

$$\alpha_D = \frac{3r_w\tau_y}{w\Delta p}. \tag{11}$$

Following substitution of Eqs. (8)–(11) into Eq. (7) yields a dimensionless ordinary-differential equation in terms of the mud-invasion radius r_D, as a function of dimensionless time[14] t_D:

$$\frac{\mathrm{d}r_D}{\mathrm{d}t_D} = \frac{1 - \alpha_D(r_D - 1)}{4r_D \ln r_D}. \tag{12}$$

Equation (12) illustrates $\frac{\mathrm{d}r_D}{\mathrm{d}t_D} \to 0$ when $r_D \to (1 + 1/\alpha_D)$. In other words, a Bingham mud, merely because of its positive yield value, has a finite invasion $(r_D)_{max}$ equal to

$$(r_D)_{max} = 1 + \frac{1}{\alpha_D}. \tag{13}$$

According to Eq. (4), it can be obtained that

$$(V_m)_{max} = \pi w\left[(r_s)_{max}^2 - r_w^2\right]. \tag{14}$$

Substituting Eq. (8) into Eq. (14) gives

$$(V_m)_{max} = \pi w r_w^2\left[(r_D)_{max}^2 - 1\right]. \tag{15}$$

On this basis, the joint Eqs. (11), (13), and (15) can be obtained[14]

$$\left(\frac{\Delta p}{\tau_y}\right)^2 w^3 + 6r_w\left(\frac{\Delta p}{\tau_y}\right)w^2 - \frac{9}{\pi}(V_\mathrm{m})_{max} = 0. \tag{16}$$

The derivative analysis of Eq. (16) is carried out

$$f'(w) = 3\left(\frac{\Delta p}{\tau_y}\right)^2 w^2 + 12r_w\left(\frac{\Delta p}{\tau_y}\right)w. \tag{17}$$

Notice $f(w=0) = -9V_m/\pi < 0$ that when $w > 0$, Eq. (17) is always positive. Therefore, the fracture width value of fractured strata can be obtained by solving the cubic equation in Eq. (16).

B. Verga model

Verga *et al.*[13] posited that the radial flow of mud into a fracture of width w can be described by the solution to the diffusivity equation under steady-state conditions, namely, the Muskat equation[12]

$$\frac{\partial^2 P}{\partial r^2} + \frac{1}{r}\frac{\partial P}{\partial r} = \frac{\mu \phi c_t}{k}\frac{\partial P}{\partial t}. \tag{18}$$

The bottom hole drilling overpressure, ΔP, can then be expressed as

$$\Delta P = \frac{6 q_m(t)\mu}{\pi w^3}\ln\frac{\left(\frac{V_m(t)}{\pi w} + r_w^2\right)^{\frac{1}{2}}}{r_w}, \tag{19}$$

where $q_m(t)$ represents the mud loss rate values recorded by the flowmeter during each time step, μ is the mud viscosity, r_w is the wellbore radius, and $V_m(t)$ is the cumulative mud volume lost in the fracture at time t.

It is assumed that the mud can be regarded as a Bingham fluid, and thus, the shear stress, τ, can be described by proper values for the yield point, τ_y, and plastic viscosity, μ_p:

$$\tau = \tau_y + \mu_p \gamma, \tag{20}$$

where γ is the shear rate in the fracture.

Considering that γ ranges between 10^4 and 10^7 s^{-1} for fracture width ranging between 100 and 1000 μm, the yield point value can be neglected, and the mud viscosity can be approximated to the plastic viscosity[13]

$$\mu = \frac{\tau}{r} = \mu_p. \tag{21}$$

The solution to the fracture width w in Eq. (19) is obtained numerically by application of the iterative method.

III DATA COLLECTION AND PROCESSING

A. Data collection

The modeling data used in this paper all come from the J block of a certain oil field in the western part of China. As an important energy development area, this block not only contains abundant oil and gas resources but also has complex and variable geological structures. The strata of the J block are composed of various types of rocks, including hard igneous rocks, fragile sedimentary rocks, and metamorphic rocks with significant anisotropy. The differences in the physical and mechanical properties between these rocks are large, making the rock strata prone to forming widely distributed natural fractures. These fractures provide a quick seepage channel for drilling fluid, leading to frequent drilling mud losses incidents during the drilling process. Faced with this serious challenge, the oil field development team has tried a variety of plugging techniques and methods, from traditional bridge plugging to advanced cement plugging, and then to composite plugging schemes that combine the advantages of multiple materials and technologies. Although to some extent, it has alleviated the drilling mud losses problem, due to the complexity of the strata fractures and the difficulty in predicting the fracture width, the success rate of single plugging is still relatively low.

The J-5 well is the focus of our study within the J block. Throughout the drilling process, there have been several severe instances of drilling mud loss, which have hindered the conduct of imaging logging and core sampling operations. Consequently, this has made it impossible to effectively calculate the underground fracture width. The logging data, however, are the only significant data source available at this juncture. There is an immediate necessity to develop a neural network prediction method for fracture width based on this logging data, in order to identify effective plugging solutions. We have systematically collected the logging data from 11 neighboring wells within the J block that exhibit similar geological characteristics to the J-5 well, and these data include crucial information about mud loss. After a thorough statistical organization, we have obtained a total of 85 comprehensive sets of logging data records. The specific distribution of sample sizes for each well is illustrated in Fig. 2. Following this, we plotted these data points according to the changes in drilling depth, creating a visual distribution map of the logging parameters (as shown in Fig. 3). It is evident from Fig. 3 that these logging data encompass nine key parameters during the drilling process: drilling mud gel strength, pump pressure, mud loss rate, drilling rate, drilling mud displacement, mud loss volume, plastic viscosity, well depth, and fracture width (measured by Formation Micro Scanner—FMI). Our analysis revealed that at the strata boundaries, the logging data exhibit significant characteristic changes: an increase in the frequency of losses, an increase in the volume of loss, and a noticeable widening of the fracture width. Additionally, during this period, there is a general acceleration in drilling rate, and the pump pressure fluctuates. Generally, these phenomena occur due to the significant differences in the mechanical properties of rocks at the strata boundaries, which makes it likely for fracture zones to form.

B. Data processing

Selecting appropriate stratigraphic fracture characteristic parameters is crucial for the accuracy of the prediction results. The width of stratigraphic fractures is closely related to many factors, and the influencing factors are quite complex. First of all, the mud loss rate and the mud loss volume have a direct relationship with the fracture width. The loss rate of drilling mud increases with the increase in fracture width, and the mud loss volume per unit time also increases accordingly. In addition, a high density of drilling mud, as well as high

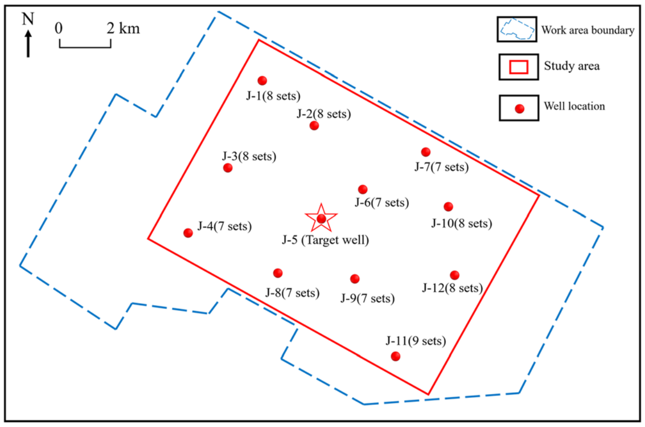

FIG. 2. Distribution of well positions in Block J.

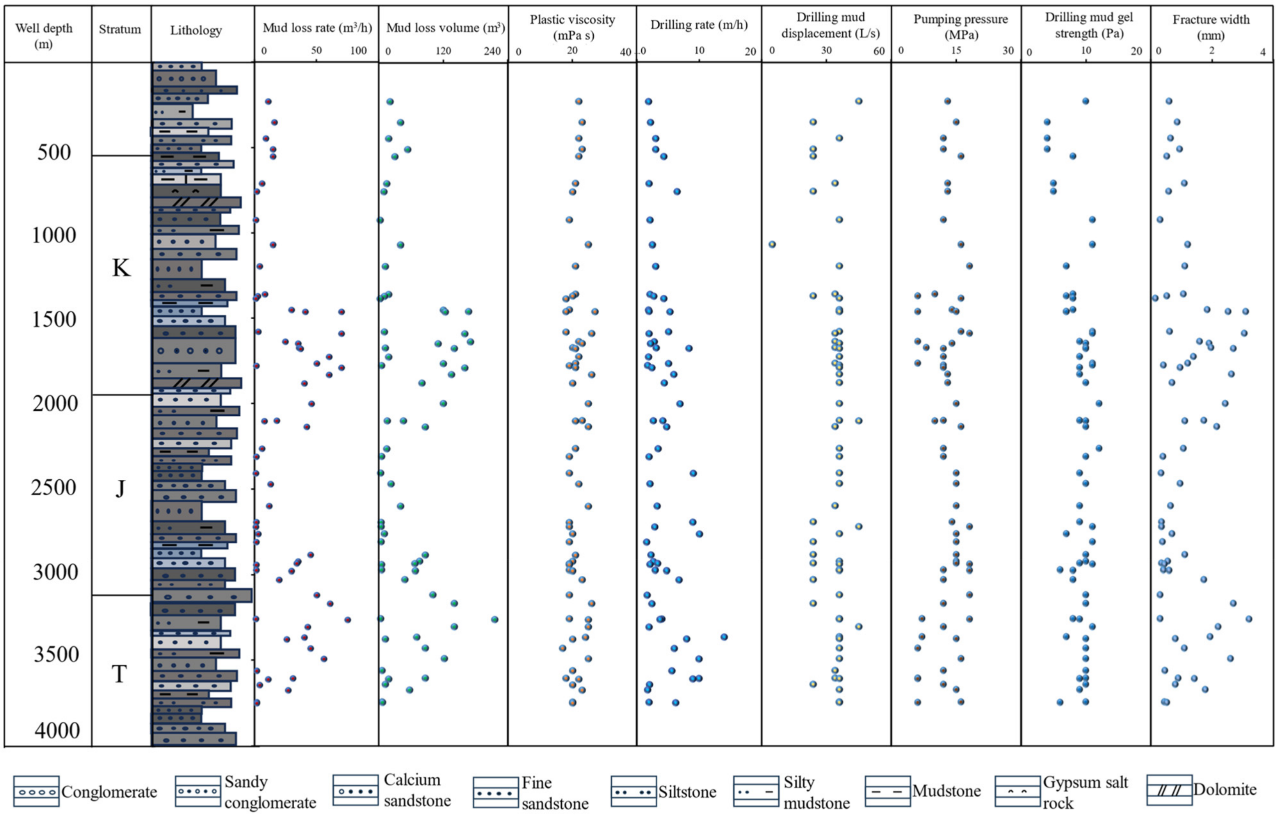

FIG. 3. Distribution of data points for logging parameters.

plastic viscosity, will damage the mesh structure in the drilling mud, resulting in a pressure difference between the effective liquid column pressure and the formation, which will cause changes in the fractures around the wellbore. The Lietard model established by Lietard *et al.* summarizes the relationship between plastic viscosity, drilling mud loss rate, and fracture width. Changes in pump pressure will also affect the bottom hole pressure, thereby causing changes in fracture width. As the well depth increases, the compaction effect of the strata increases, and the fracture width decreases. Excessive drilling speed can cause pressure fluctuations at the bottom of the well, leading to the expansion of fractures around the wellbore. In order to explore the extent of the impact of the above factors on fracture width and to identify the main influencing factors, a total of 85 sets of data were collected. Using the Spearman correlation analysis method, the correlation coefficients between eight parameters including mud loss volume, well depth, plastic viscosity, drilling mud gel strength, pump pressure, mud loss rate, drilling rate, and drilling mud displacement were calculated, and then, the correlation coefficients were subjected to significance testing. The results are shown in Fig. 4.

The Spearman correlation analysis indicates (as shown in Fig. 4) that the factors most closely associated with fracture width are the mud loss rate and the mud loss volume, with correlation coefficients of 0.93 and 0.92, respectively. In comparison, the association of other factors with fracture width is slightly weaker, ranked from strongest to weakest as follows: pump pressure, drilling mud displacement, drilling rate, well depth, plastic viscosity, and drilling mud gel strength. Furthermore, significance level testing shows that the correlation coefficients of pump pressure, drilling mud displacement, mud loss rate, and mud loss volume with fracture width are significant, while the correlation coefficients of drilling rate, well depth, plastic viscosity, and drilling mud gel strength with fracture width are not significant. Considering that although drilling speed, well depth, and plastic viscosity show no correlation with fracture width in terms of data statistics, they are physically related, so these three factors are still taken into account when building the model, which helps to improve the model's generalization ability. The drilling mud gel strength does not show a good correlation with fracture width in both physical and data statistical terms, so this factor is not considered, and instead, the other seven factors are chosen as input parameters. For specific data characteristics of these parameters, see Table I.

As can be seen from Table I, there is a large span in the range of each parameter. For example, the well depth ranges from 119 to 3619 m, covering different strata. Other parameters, such as drilling speed and loss volume, also have a wide range of spans. This reflects the diversity under different strata and operating conditions, and this diversity allows the trained model to adapt to a variety of different situations. In order to accurately predict the fracture width of the loss layer in the J-5 well, which suffers from severe mud loss on the site, 85 sets of data collected from neighboring wells in the same block as the J-5 well are divided into training data and test data. These are used to

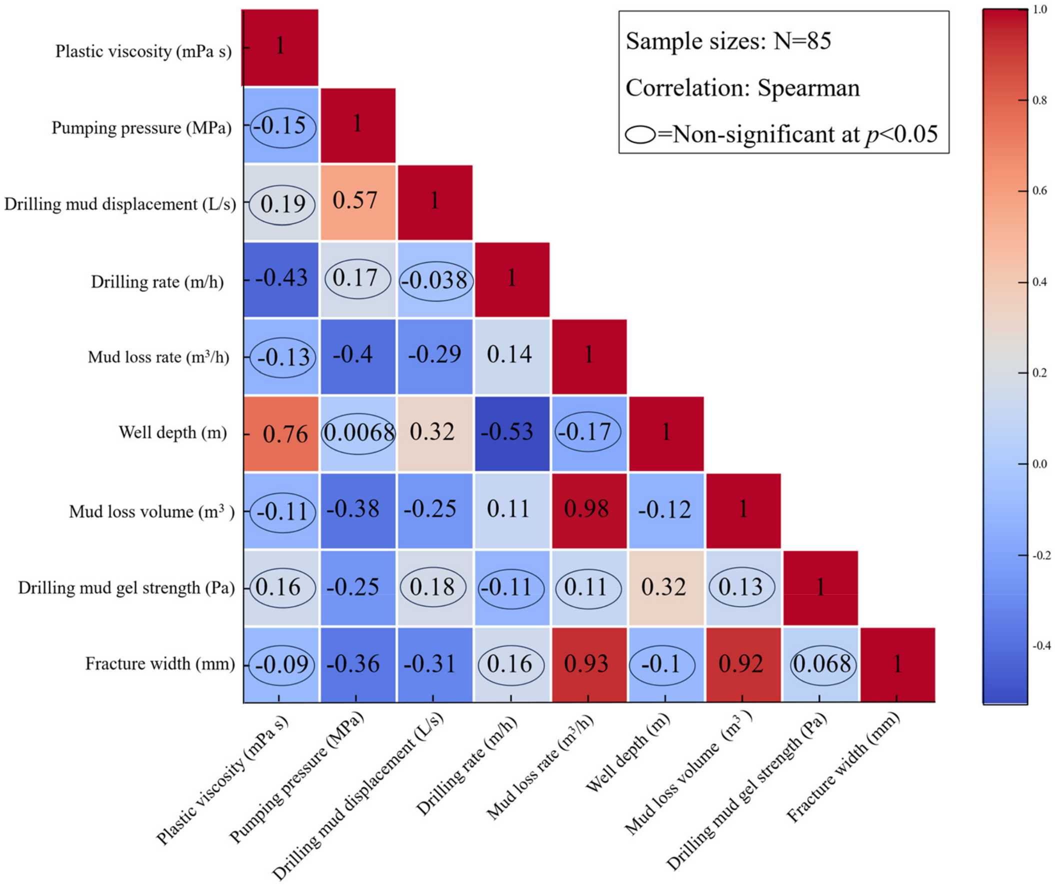

FIG. 4. Significant spearman correlation matrix in the analysis of fracture influencing factors.

train the model and test the optimal training parameters, with 70 sets of data serving as training data for model training and the remaining 15 sets of data acting as test data. The test data do not participate in the model establishment and training; they are only used to test and evaluate the established model and its optimal training parameters. Additionally, due to the different units of each parameter and their large difference in magnitude, such as well depth and plastic viscosity, this will affect the prediction effect of the prediction model. To address this, in order to eliminate the impact of the large difference in magnitude between different parameters, the data in this paper are normalized to the interval [0, 1]. The normalization equation is as follows:

$$X_i' = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}}. \tag{22}$$

In Eq. (22), X_i' is the normalized data; X_i is the input data; and $X_{\min}$ and $X_{\max}$ are the minimum and maximum values of the corresponding parameters, respectively.

TABLE I. Dataset characteristics of parameters.

Parameter name	Plastic viscosity (mPa s)	Pumping pressure (MPa)	Drilling mud displacement (L/s)	Drilling rate (m/h)	Mud loss rate (m^3/h)	Well depth (m)	Mud loss volume (m^3)	Fracture width (mm)
Maximum	29	19	45	11.76	77	3619	219	3.24
Minimum	10	5	5	1.35	0	119	0	0
Median	21	12	36	3	2	2100	9	0.57
Mean	20.54	13.09	32.87	3.89	11.92	2092.36	27.64	0.88
Standard deviation	3.57	3.91	6.69	2.38	18.82	931.55	44.38	0.89

Two indexes are selected to evaluate the prediction effect of the method. The calculation equations are the coefficient of determination R^2 and root mean square error E_{RMS}, as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (\hat{y}_i)^2}, \tag{23}$$

$$E_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \times 100\%. \tag{24}$$

In the equation, y is the observed values; $\hat{y}$ is the corresponding predicted value; and N is the number of samples.

IV. ESTABLIHMENT OF FDCNN

A. Fluid dynamics principles extraction and FDCNN establishment

The research object of this paper is the prediction of fracture width. According to the analysis of Sec. III, $\mathbf{X} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)^{\mathrm{T}}$ is selected as the input characteristic value of the neural network. $x_1 \sim x_7$ corresponds to seven logging parameters such as mud loss volume, mud loss rate, pump pressure, drilling mud displacement, drilling rate, well depth, and plastic viscosity in turn. At the same time, $\hat{y} = \hat{f}(\mathbf{X})$ is the output prediction value of fracture width. The input characteristic $\mathbf{X}$ and the output prediction value $\hat{f}(\mathbf{X})$ have a very complex nonlinear relationship. Using the powerful function of neural network to approximate nonlinear function, the BP neural network model of fracture width can be established. The neural network is a three-layer network with seven inputs and one output.

From the derivation of the Lietard–Griffiths model, we can extract two key fluid dynamics principles:

(1) The first fluid dynamics principle: From Eq. (2), it can be obtained that the fracture width w and the mud loss volume V_m have a monotonically increasing principle. That is, under the condition of keeping other input variables unchanged, the larger the leakage rate is, the larger the corresponding fracture width is. Because the fracture width is monotonically increasing with the leakage rate, the partial derivative of the predicted fracture width $\hat{f}(\mathbf{X})$ of the neural network to the leakage rate x_1 should be greater than zero, that is, $\partial\hat{f}(\mathbf{X})/\partial x_1 > 0$, so there is $h_1 = \partial\hat{f}(\mathbf{X})/\partial x_1 > 0$.

(2) The second fluid dynamics principle: From Eq. (14), it can be obtained that the fracture width w and the mud loss rate Q_{loss} have a monotonically increasing principle. That is, under the condition of keeping other input variables unchanged, the larger the leakage is, the larger the corresponding fracture width is. Because the fracture width is monotonically increasing with the leakage, the neural network prediction value of the fracture width $\hat{y} = \hat{f}(\mathbf{X})$ should be greater than zero for the partial derivative of the leakage x_2, that is, $\partial\hat{f}(\mathbf{X})/\partial x_2 > 0$, so there is $h_2 = \partial\hat{f}(\mathbf{X})/\partial x_2 > 0$.

These two fluid dynamics principles can also be extracted from the Verga model. It can be seen that this is not a special case of a model, but an objective mechanical principle. Training the neural network model under the guidance of this mechanical principle can improve the interpretability and generalization ability of the neural network model. In the case of a small amount of data, a model with better prediction stability can also be constructed. Based on the above fluid dynamics principles constraints, the established neural network model is shown in Fig. 5.

In Fig. 5, f_1 is the activation function in the hidden layer, which applies the sigmoid function; f_2 is the activation function in the output layer, which uses a linear function and the output function is $\hat{y} = \hat{f}(\mathbf{X})$.

$$\begin{aligned}\hat{y} &= \mathbf{v}^{\mathrm{T}}\mathbf{a}^{1} + b^{(2)} \\ &= \mathbf{v}^{\mathrm{T}}\frac{1}{1+\exp(-(\boldsymbol{\omega}\mathbf{X}+\mathbf{b}^{(1)}))} + b^{(2)} \\ &= \sum_{m=1}^{M}\left(\frac{v_m}{1+\exp-\left(\sum_{i=1}^{7} w_{mi}x_i + b_m^{(1)}\right)} + b^{(2)}\right).\end{aligned} \tag{25}$$

M is the number of hidden neurons, and it can be determined by experiments; $\boldsymbol{\omega} = (\omega_{1i}, \omega_{2i}, \ldots \omega_{mi})^{\mathrm{T}}$ is the matrix connecting the weights of the nodes in the input layer and the neurons in the hidden layer; $\mathbf{b}^{(1)} = (b_1{}^{(1)}, b_2{}^{(1)}, \ldots b_m{}^{(1)})^{\mathrm{T}}$ is the bias vector of neurons in the hidden layer; $\mathbf{v} = (v_1, v_2, \ldots v_m)^{\mathrm{T}}$ is the weight vector of the neurons connecting the hidden layer and the output layer; $b^{(2)}$ is the bias of neurons in the output layer; and a_m is the output of the mth neuron of the hidden layer, and it can be expressed as follows:

$$a_m^{(1)} = \frac{1}{1+\exp\left(-\left(\sum_{i=1}^{7}\omega_{mi}x_i + b_m^{(1)}\right)\right)}. \tag{26}$$

B. Training algorithm of FDCNN

In the fluid dynamics model, it is found that the natural fracture width observed values y has a monotonically increasing principle with

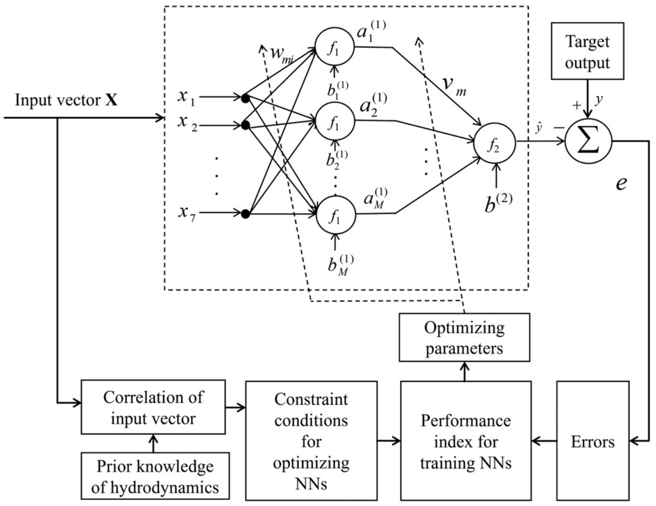

FIG. 5. FDCNN implementation flow.

the leakage rate x_1 and the leakage amount x_2, that is, the first-order partial derivatives of the output function $\hat{y}$ to the input of x_1 and x_2 are positive. We can use these fluid dynamics principles to construct the constraints of training neural network (NN). Assuming that the performance index of the training neural network is E, the training method of K training samples is adopted, and E is expressed as follows:

$$\begin{cases} \min E = \sum_{k=1}^{K} \frac{1}{2}\left[y_k - \hat{f}(\mathbf{X}_k)\right]^2 = \sum_{k=1}^{K} \frac{1}{2} e_k^2 \\ s.t. h_i(\boldsymbol{\omega}, \mathbf{v}) = \frac{\partial \hat{f}(\mathbf{X}_k)}{\partial x_i} > 0, \quad i = 1, 2, \ldots, I. \end{cases} \tag{27}$$

Among them, I is the number of input neurons with constraint effect in NN, and the maximum value in this model is 2; and $h_i(\boldsymbol{\omega}, \mathbf{v})$ can be expressed as follows:

$$h_i(\boldsymbol{\omega}, \mathbf{v}) = \frac{\partial \hat{f}(\mathbf{X}_k)}{\partial x_i} = \mathbf{v}^{\mathrm{T}}(1 - \mathbf{a}^{(1)})\mathbf{a}^{(1)}\boldsymbol{\omega} = \sum_{m=1}^{M} (1 - a_m^{(1)}) a_m^{(1)} v_m \omega_{mi}. \tag{28}$$

Because $0 < a_m^{(1)} = 1/(1 + \exp(-(\sum_{i=1}^{7} \omega_{mi} x_i + b_m^{(1)}))) < 1$, to ensure $h_i(\boldsymbol{\omega}, \mathbf{v}) > 0$, it is only necessary that $\sum_{m=1}^{M} v_m \omega_{mi} > 0$.

The augmented Lagrange multiplier method is used to solve the equation, and the augmented objective function $P(\boldsymbol{\omega}, \mathbf{v}, \mathbf{b}^{(1)}, b^{(2)}, \sigma, \lambda)$ is constructed as follows:[33,34]

$$\begin{aligned} &\min P(\boldsymbol{\omega}, \mathbf{b}^{(1)}, \mathbf{v}, b^{(2)}, \sigma, \lambda) \\ &\quad = \min \sum_{k=1}^{K} \left\{ \frac{1}{2} e_k^2 + \frac{1}{2\sigma} \sum_{i=1}^{2} \left\{ [\min(0, \sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i)]^2 - \lambda_i^2 \right\} \right\} \\ &\quad = \min \sum_{k=1}^{K} \{E_k + E_c\}. \end{aligned} \tag{29}$$

λ is the Lagrange multiplier, and σ is the penalty factor, $E_c = \sum_{i=1}^{2} \left\{ [\min(0, \sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i)]^2 - \lambda_i^2 \right\} / 2\sigma$, and $E_k = e_k^2/2$.

Using the backpropagation learning rule (BP) and gradient training method, and assuming that the learning rate is η, the iterative equation of the parameter $\boldsymbol{\omega}$, $\mathbf{b}^{(1)}$, $\mathbf{v}$, $b^{(2)}$ of the neural network is as follows:

$$\begin{cases} \omega_{mi}(n+1) = \omega_{mi}(n) - \eta \frac{1}{K} \frac{\partial \phi}{\partial \omega_{mi}} = \omega_{mi}(n) - \eta \frac{1}{K} \sum_{k=1}^{K} \left(\frac{\partial E_k}{\partial \omega_{mi}} + \frac{\partial E_c}{\partial \omega_{mi}} \right) \\ b_m^{(1)}(n+1) = b_{mi}(n) - \eta \frac{1}{K} \frac{\partial \phi}{\partial b_m^{(1)}} = b_m^{(1)}(n) - \eta \frac{1}{K} \sum_{k=1}^{K} \left(\frac{\partial E_k}{\partial b_m^{(1)}} + \frac{\partial E_c}{\partial b_m^{(1)}} \right) \\ v_m(n+1) = v_m(n) - \eta \frac{1}{K} \frac{\partial \phi}{\partial v_m} = v_m(n) - \eta \frac{1}{K} \sum_{k=1}^{K} \left(\frac{\partial E_k}{\partial v_m} + \frac{\partial E_c}{\partial v_m} \right) \\ b^{(2)}(n+1) = b^{(2)}(n) - \eta \frac{1}{K} \frac{\partial \phi}{\partial b^{(2)}} = b^{(2)}(n) - \eta \frac{1}{K} \sum_{k=1}^{K} \left(\frac{\partial E_k}{\partial b^{(2)}} + \frac{\partial E_c}{\partial b^{(2)}} \right), \end{cases} \tag{30}$$

where

$$\begin{cases} \frac{\partial E_k}{\partial \omega_{mi}} = -e_k v_m (1 - a_m) a_m x_i \\ \frac{\partial E_k}{\partial b_m^{(1)}} = -e_k v_m (1 - a_m) a_m \\ \frac{\partial E_k}{\partial v_m} = -e_k a_m \\ \frac{\partial E_k}{\partial b^{(2)}} = -e_k \\ \frac{\partial E_c}{\partial \omega_{mi}} = \theta_i(\sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i) \frac{\partial h_i(\boldsymbol{\omega}, \mathbf{v})}{\partial \omega_{mi}} \\ \quad = \theta_i(\sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i) \times (1 - a_m) \\ \quad\quad \times a_m v_m (\omega_{mi} x_i - 2 a_m \omega_{mi} x_i + 1) \\ \frac{\partial E_c}{\partial b_m^{(1)}} = \frac{1}{2\sigma} \sum_{i=1}^{2} 2\theta_i \sigma (\sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i) \frac{\partial h_i(\boldsymbol{\omega}, \mathbf{v})}{\partial b_m^{(1)}} \\ \quad = \frac{1}{2\sigma} \sum_{i=1}^{2} 2\theta_i \sigma (\sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i) \\ \quad\quad \times (1 - a_m)(1 - 2a_m) a_m v_m \omega_{mi} \\ \frac{\partial E_c}{\partial v_m} = \frac{1}{2\sigma} \sum_{i=1}^{2} 2\theta_i \sigma (\sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i) \frac{\partial h_i(\boldsymbol{\omega}, \mathbf{v})}{\partial v_m} \\ \quad = \frac{1}{2\sigma} \sum_{i=1}^{2} 2\theta_i \sigma (\sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i)(1 - a_m) a_m \omega_{mi} \\ \frac{\partial E_c}{\partial b^{(2)}} = \frac{1}{2\sigma} \sum_{i=1}^{2} 2\theta_i \sigma (\sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i) \frac{\partial h_i(\boldsymbol{\omega}, \mathbf{v})}{\partial b^{(2)}} \\ \quad = \frac{1}{2\sigma} \sum_{i=1}^{2} 2\theta_i \sigma (\sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i), \end{cases} \tag{31}$$

where

$$\theta_i = \begin{cases} 1 & if \;\; \sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i \le 0 \\ 0 & if \;\; \sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i > 0. \end{cases} \tag{32}$$

Considering the augmented Lagrange multiplier method, the iterative function of the multiplier λ_i in Eq. (31) is given by the following equation:

$$\lambda_i(j+1) = \begin{cases} 0 & \text{if } \sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i(j) > 0 \\ \lambda_i(j) - \sigma h_i(\boldsymbol{\omega}, \mathbf{v}) & \text{if } \sigma h_i(\boldsymbol{\omega}, \mathbf{v}) - \lambda_i(j) \le 0. \end{cases} \tag{33}$$

The main steps of the augmented Lagrange multiplier method for solving the constrained optimization neural network (FDCNN) are summarized as follows:

Step 1: Initialize the FDCNN parameters, including the neural network's starting weights and biases $(\boldsymbol{\omega}(0), \mathbf{b}^{(1)}(0), \mathbf{v}(0), b^{(2)}(0)$, the initial Lagrange multiplier $\lambda(\lambda \ge 0)$, the penalty factor $\sigma(\sigma(0) \ge 0)$, the initial termination criterion $\beta(0)$, the neural network's learning rate $\eta(0 < \eta \le 1)$, the error tolerance for the termination criterion $\varepsilon(0 \le \varepsilon \le 1)$, the count of internal iterations *Epoch_num*, the maximum allowed internal iterations *Epoch_Max*, the target mean square error $0 \le Err\_MSE \le 1$, the count of external iterations j=1, the maximum allowed external iterations *Iterate_Max*, the penalty factor update rate $\mu(\mu > 1)$, and the termination criterion update factor $\gamma(0 < \gamma \le 1)$.

Step 2: Utilize the parameters from the $(j-1)$th iteration $\boldsymbol{\omega}(j-1)$, $\mathbf{b}^{(1)}(j-1)$, $\mathbf{v}(j-1)$, $b^{(2)}(j-1)$ and the algorithm from Eqs. (30)–(33) to address the augmented objective function in Eq. (29). Terminate the neural network training if the mean square error of the augmented function falls below *Err_MSE* or if *Train_num* exceeds *Train_Max*. During this step, λ and σ remain unchanged.

Step 3: Evaluate the FDCNN's termination condition. If $\beta(j) \le \varepsilon$ or $j \ge Iterate\_Max$, halt the FDCNN, conclude the process, and present the optimal parameters $\boldsymbol{\omega}(k)$, $\mathbf{b}^{(1)}(k)$, $\mathbf{v}(k)$, $b^{(2)}(k)$. If not, proceed to the next step.

Step 4: Adjust the penalty factor σ. If $\beta(j) \ge \gamma\beta(j-1)$, then increase σ by a factor of μ for the next iteration; otherwise, retain the current value of σ, where $\beta(j)$ following equation:

$$\beta(j) = \sqrt{\left(\sum_{i=1}^{2}\left\{\min\left[h_i(\boldsymbol{\omega},\mathbf{v}), \frac{\lambda_i(j)}{\sigma}\right]\right\}^2\right)}. \tag{34}$$

Step 5: Refine the multiplier λ using Eq. (33).

Step 6: Increment the external iteration counter $(j = j+1)$ and repeat from Step 2.

V. EXPERIMENTAL STUDIES AND APPLICATION

This experiment aims to explore the training performance of the FDCNN under different conditions, while testing the DDNN as a control experiment (the DDNN is equivalent to the FDCNN when $\sigma(0) = 0$, which is the BPNN prediction model proposed by predecessors). First, a variable redundancy analysis test is conducted to determine the final model input feature vector. Second, by comparing the training effects under different sample sizes, the predictive ability of the FDCNN in small sample training is assessed. In addition, in the neural network objective function constructed by the augmented Lagrangian, the weight of the penalty factor determines the importance of the constraint conditions to the objective function. If the penalty factor is too small, it may not be sufficient to push the solution into the feasible region; if it is too large, it may lead to numerical problems, making the solution updates too drastic and difficult to stabilize. An appropriate penalty factor can help the algorithm converge to the optimal solution more quickly.[34] The relative size between the number of hidden neurons and the number of training samples can lead to overfitting or under-fitting of the neural network.[31] Therefore, it is necessary to adjust the penalty factor and the number of hidden neurons to test the performance of the FDCNN, thereby verifying the rationality of the hyperparameter values selected in the experiment. Finally, based on the experimental results, the optimal training parameters are selected to provide theoretical support for the practical application of the FDCNN.

Neural networks (NN) trained by FDCNN- and DDNN-based algorithms, respectively. The initial parameters of this algorithm are: $Err\_MSE = 0.00001$, $\eta = 0.8$, $M = 5$, $Epoch\_Max = 10000$, $Iterate\_Max = 100$, $\varepsilon = 0.0001$, $\gamma = 0.7$, $\mu = 5.0$, $\lambda(0) = 1.0$, $\sigma(0) = 5.0$ and the initial values of $\boldsymbol{\omega}(0)$, $\mathbf{b}^{(1)}(0)$, $\mathbf{v}(0)$, $b^{(2)}(0)$ are obtained using a random function. In addition, the reproducibility of the experiment is one of the key factors to ensure the credibility and reliability of the research results. To achieve this goal, the random seed is set in the PyTorch training environment using torch.manual_seed(), and the same seed (42 in this experiment) is set to ensure that the random processes (weight initialization, data division) will produce consistent results when the experiment is run multiple times, thus eliminating the influence of randomness on the experimental results. After the correlation analysis and normalization in Sec. III, 85 sets of data were obtained. The same random seed will be maintained for data division to ensure that the training set and the test set remain unchanged in the experiment for comparative analysis, with 70 sets in the training set and 15 sets in the test set.

A. Variable redundancy analysis test

Collinearity among features is a critical issue in machine learning, referring to the strong correlation existing between two or more features, which can potentially impact the performance of models. As revealed by the Spearman correlation analysis in Sec. III, the correlation coefficient between the mud loss volume and mud loss rate is extremely high, reaching 0.98. Furthermore, from the derivation process of the Lietard–Griffiths model outlined in Sec. II, it can be discerned that the mud loss volume is essentially the integral of the mud loss rate over time. Both the correlation analysis and the physical interpretation suggest the existence of collinearity between mud loss volume and mud loss rate. To ensure the robustness of our algorithm, we will conduct two sets of experiments with the initial parameters, aiming to test the impact of removing redundant features on the performance of DDNN and FDCNN and to determine the final model input feature vectors.

In Sec. IV, we specifically identified seven input features. First, we tested the performance of the DDNN with input features x_1 to x_7 and after removing the redundant variable (here, the removed variable is x_1, the mud loss volume), the input features are x_2 to x_7. The test results show (as shown in Fig. 6), after removing the redundant feature, the predictive accuracy of the DDNN has improved and the error fluctuation is smaller, indicating that redundant features can reduce

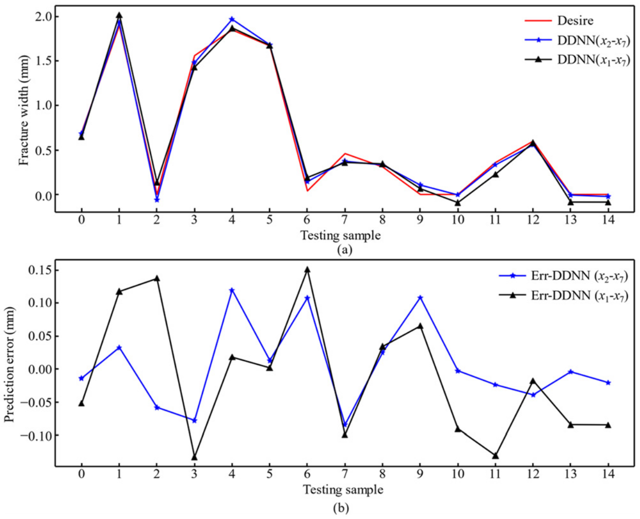

FIG. 6. Comparative testing results of DDNNs (the input features are $\mathbf{X}_1 = [x_1, x_2, \ldots x_7]$ and $\mathbf{X}_2 = [x_2, x_3, \ldots x_7]$, respectively) with 70 training samples when $\sigma(0) = 5$. (a) Comparisons of the desire outputs, DDNN (x_1–x_7) and DDNN(x_2–x_7); (b) errors of DDNN (x_1–x_7) and DDNN(x_2–x_7).

the stability of traditional unconstrained models and weaken the model's predictive power. Subsequently, keeping the same two sets of input feature values as DDNN, and applying prior knowledge constraints, when the input features are x_1 to x_7, the physical constraints are $\partial\hat{f}(\mathbf{X})/\partial x_1$ and $\partial\hat{f}(\mathbf{X})/\partial x_2$, and after removing the redundant variable (input features are x_2 to x_7), the physical constraint is $\partial\hat{f}(\mathbf{X})/\partial x_2$, which correspond to $I = 1$ and $I = 2$ in Eq. (27), that is, the number of input neurons with constraint effect in the NN. The test results show (as shown in Fig. 7), after removing the redundant features, the generalization error fluctuation of the FDCNN is very small, and the predictive results change little, indicating that the FDCNN, which uses prior knowledge to design the loss function, has the ability to consider the relationships between features, and reduces the impact of redundant features on the FDCNN by penalizing highly collinear feature combinations through penalty factors.

The experiment separately investigated the performance of the DDNN and FDCNN models after removing redundant input features. The results showed that redundant features lead to a decrease in the performance of the DDNN, while the impact on the performance of the FDCNN is relatively small. To form a more convincing control group experiment and more effectively test the performance of the FDCNN, the parameter selection was based on the optimal training performance of the DDNN as a fundamental principle. Therefore, the following experiments are all conducted with $I = 1$.

B. The performance testing of FDCNN

1. FDCNN training

To evaluate the generalization capability of the FDCNN, the training dataset was further divided into three distinct subsets for sequential training, each containing 30, 50, and 70 datasets, respectively. It is worth noting that the size of the test set remains consistent across the three sets of tests, all consisting of 15 datasets, and the order of the data remains consistent. The fracture width prediction results generated by FDCNN and DDNN are shown in the attached figures. In Figs. 8–10, part (a) displays the expected results (referred to as "desire," i.e., the fracture width values measured by FMI), the test results of FDCNN, and the test results of DDNN, while part (b) provides a detailed explanation of the corresponding errors.

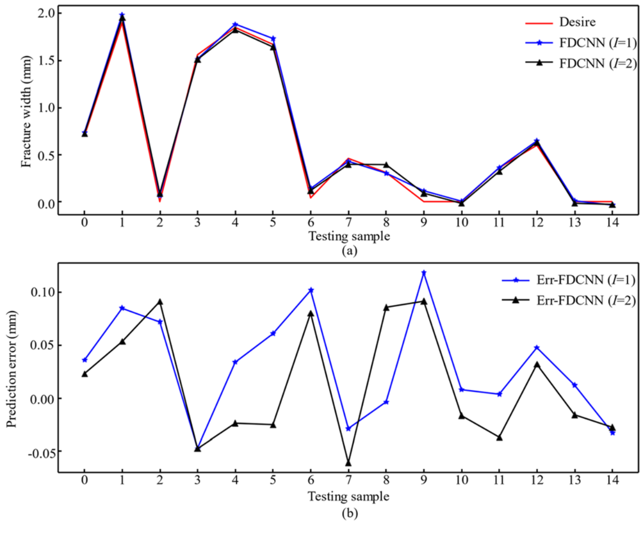

FIG. 7. Comparative testing results of FDCNNs ($I = 1$, 2, respectively) with 70 training samples when $\sigma(0) = 5$. (a) Comparisons of the desire outputs, FDCNN ($I = 1$) and FDCNN ($I = 2$); (b) errors of FDCNN ($I = 1$) and FDCNN ($I = 2$).

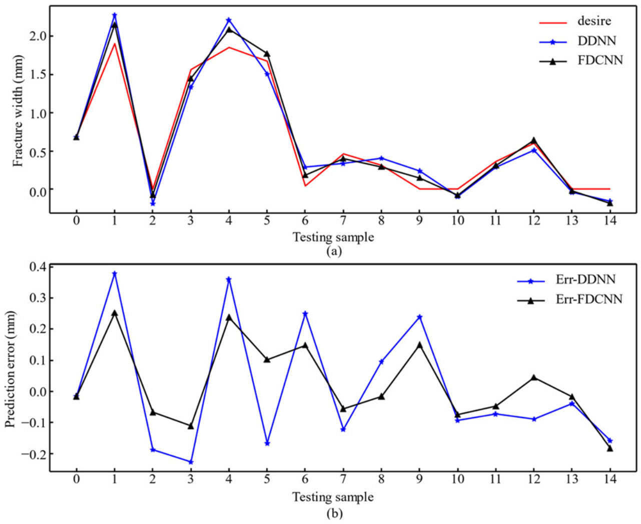

FIG. 8. Comparative testing results of FDCNN and DDNN with 30 training samples when $\sigma(0) = 5$. (a) Comparisons of the desire outputs, FDCNN and DDNN; (b) errors of FDCNN and DDNN.

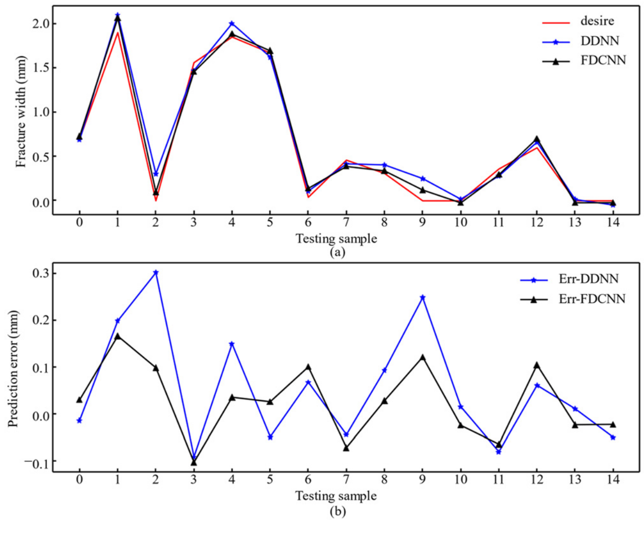

FIG. 9. Comparative testing results of FDCNN and DDNN with 50 training samples when $\sigma(0) = 5$. (a) Comparisons of the desire outputs, FDCNN and DDNN; (b) errors of FDCNN and DDNN.

The analysis of Figs. 8–10 and Table II reveals that the FDCNN exhibits a lower generalization error compared to the DDNN. This advantage is consistent across varying quantities of training data, indicating that the incorporation of fluid dynamics principles enhances the

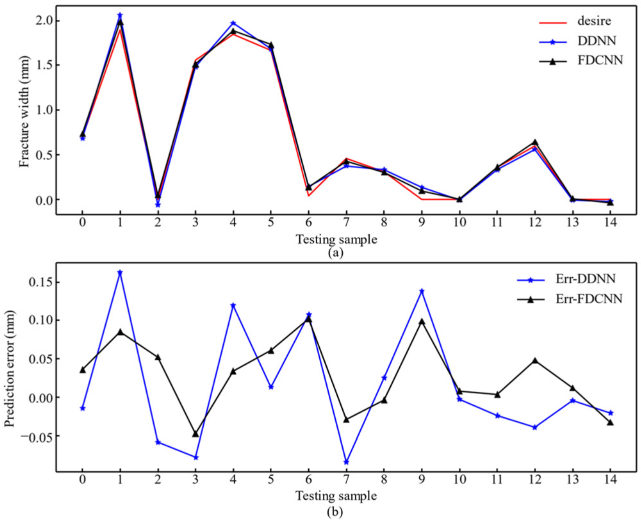

FIG. 10. Comparative testing results of FDCNN and DDNN with 70 training samples when $\sigma(0) = 5$. (a) Comparisons of the desire outputs, FDCNN and DDNN; (b) errors of FDCNN and DDNN.

neural network's ability to generalize. However, as the training dataset expands, such as an increase from 30 to 50 sets, and subsequently to 70 sets, the FDCNN's capacity to diminish the generalization error is observed to decrease progressively. In fact, when the number of training samples is sufficient, for example, the number of training set is 70 sets, which exceeds the number of adjustable neural network parameters[34] [i.e., $(K > (I+2) \times M + 1$, where I is the number of neural network input neurons (in this experiment, $I = 6$), and M is the number of hidden neurons (in this experiment, $M = 5$)], the generalization error of DDNN is greatly reduced. At this time, although the performance of FDCNN is still better than that of DDNN, the generalization error of FDCNN changes little. When the training samples are large enough, the performance of DDNN will approach the performance of FDCNN. The findings from the experiment indicate that FDCNN is suitable for training neural networks in the case of insufficient training samples.

2. Testing the performance of FDCNN with different penalty factors

The efficacy of the FDCNN is significantly influenced by the selection of the penalty factor. An inadequately low penalty factor may fail to impose necessary constraints on the performance index as outlined in Eq. (22), while an excessively high penalty factor can lead to numerical instability or divergence within the FDCNN. To explore this sensitivity, the initial penalty factor $\sigma(0)$ was varied across the values of 0.1, 1, 5, 10, and 30, with the comparative analysis of FDCNN and the DDNN performance under training set of 70 data sets presented in Fig. 11. This figure contrasts the outputs of FDCNN and DDNN with the target expected values in section (a), with the respective error analyses provided in section (b).

TABLE II. Prediction errors and correlation coefficients of different sample sizes.

	FDCNN			DDNN		
Evaluation index	30 samples	50 samples	70 samples	30 samples	50 samples	70 samples
E_{RMS} (%)	13.5	8.1	5.2	29	14	8
R^2	0.93	0.97	0.98	0.89	0.93	0.97

Upon examination of Fig. 11, it becomes evident that the penalty factor σ plays a crucial role in enhancing the neural network's generalization capabilities. When $\sigma(0)$ is set to a low value, such as 0.1, the error associated with FDCNN approximates that of DDNN. This observation can be attributed to the penalty factor's insufficient magnitude to enforce the constraints effectively during neural network optimization. Conversely, when $\sigma(0)$ is elevated to a high value, for instance, 10, the error for FDCNN escalates, with a marked increase observed when $\sigma(0)$ reaches 30. This sharp increase in error is indicative of the penalty factor's oversized impact, which can impede the convergence of FDCNN. However, when $\sigma(0)$ is chosen to be moderate, for example, $\sigma(0) = 5.0$, the FDCNN demonstrates a markedly lower error rate compared to DDNN. Under these conditions, the constraints are not only effective but also instrumental in achieving neural network optimization.

This experiment explores the importance of judiciously choosing penalty factors in the context of FDCNN training, highlighting the impact of prior knowledge on the ability of neural networks to generalize from training data. Suitable penalty factors are balanced to facilitate convergence of the network while maintaining its prediction accuracy, thus providing valuable guidance on parameter selection for the practical application and tuning of FDCNN algorithms.

3. Testing the performance of FDCNN with different numbers of hidden neurons M

To examine the potential for both over-fitting and under-fitting in the FDCNN, a series of experiments were executed with training set

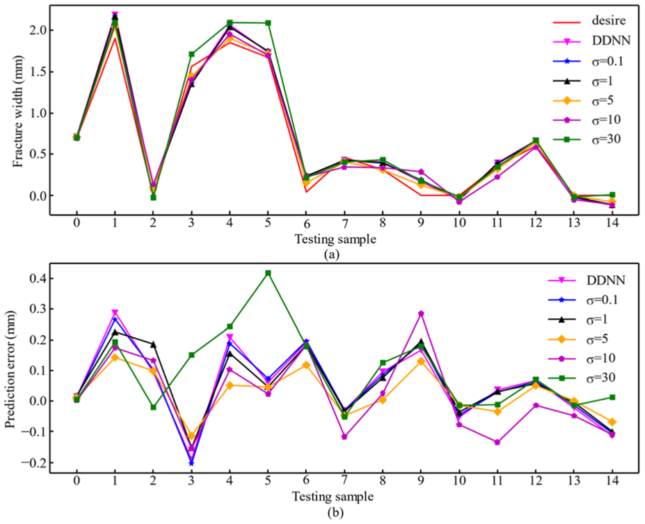

FIG. 11. Comparative testing results of FDCNN and DDNN with 70 training samples when $\sigma(0) = 0.1$, 1, 5, 10, and 30, respectively. (a) Comparisons of the desire outputs, FDCNN and DDNN; (b) errors of FDCNN and DDNN.

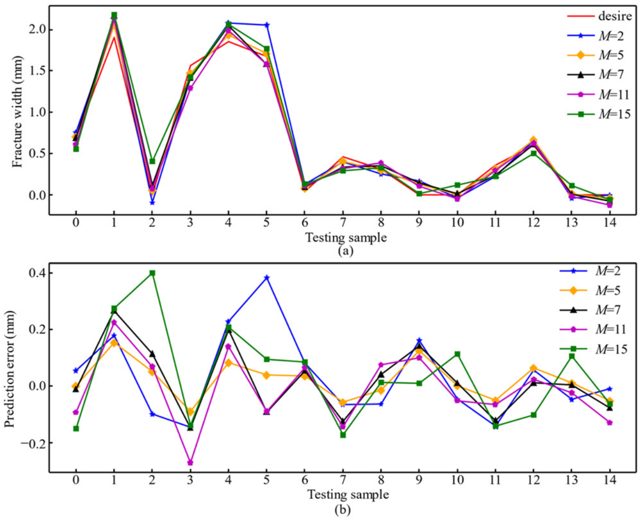

FIG. 12. Comparative testing results of FDCNN and DDNN with 70 training samples when $\sigma(0)=5$ and $M=2$, 5, 7, 11, and 15, respectively. (a) Comparisons of the desire outputs, FDCNN and DDNN; (b) errors of FDCNN and DDNN.

of 70 data sets and a fixed penalty factor $\sigma(0)$ set to 5. The pivotal parameter varied was the count of neurons in the hidden layer, denoted as M, which was systematically altered through the values of 2, 5, 7, 11, and 15. The neural network models were both trained and evaluated using the provided Eqs. (22)–(25), with the outcomes of these experiments detailed in Fig. 12. This figure delineates the predictive outputs of the FDCNN against the anticipated results in section (a), while the associated error metrics are delineated in section (b).

The analysis revealed that the FDCNN, despite its incorporation of fluid dynamics principles during the neural network's training regimen, encounters a pronounced increase in fitting error under conditions of an excessively sparse or abundant number of hidden neurons. Specifically, when M is particularly low ($M=2$), the model's architecture, although sufficient in number to meet the FDCNN's training criteria[34] [$K>(I+2)\times M+1$, where I represents the input layer's neuron count and K denotes the training instances], the structure is overly simplistic. This simplicity leads to under-fitting, where the model fails to adequately capture the data's nuances, thus amplifying the generalization error. Conversely, an overly large M, such as when $M=15$, results in a 7-15-1 layered network that succumbs to over-fitting. This over-fitting is attributed to the disproportionate ratio of model complexity to the quantity of training data, which consequently impairs the model's generalization capability.

The findings from the experiment indicate that an intermediate value of $M=5$ yields the most favorable results, with the FDCNN demonstrating a minimized generalization error under this configuration. This suggests an optimal balance between the model's complexity and the size of the training dataset, thereby enabling the FDCNN to achieve a higher degree of accuracy in its predictions.

C. The practical application of FDCNN

The objective well of this study is J-5 well. During the drilling process, J-5 well experienced three significant mud losses at different strata depths of 2309, 3169, and 3675 m. To better formulate a plugging scheme, it is often necessary to evaluate the fracture width at the well leakage location on-site. The optimal training parameters obtained from the test experiments [$\sigma(0)=5$, $M=5$, $I=1$, and the training set consists of 85 logging data collected from neighboring wells of J-5 well] are used to train the FDCNN. The well-trained FDCNN model is then saved. Three sets of engineering mud loss logging data for J-5 well were collected and organized on-site, and the corresponding fracture widths were measured using FMI imaging logs as the observed values. The mud loss rate, pump pressure, drilling mud displacement, drilling rate, well depth, and plastic viscosity are used as input parameters for the FDCNN model, with fracture width as the target output parameter. Then, the fracture width of the mud loss strata in J-5 well is predicted and compared with the neural network prediction method DDNN [trained using FDCNN at $\sigma(0)=0$, equivalent to the BP neural network] and the fluid dynamics models (Verga and Lietard–Griffiths, both of which use iterative methods for calculation). Specific logging data and final results are shown in Tables III and IV.

It can be seen from Table IV that Verga model and Lietard–Griffiths model have the worst prediction performance in the prediction of fracture width of three leakage points in J-5 well, and the relative error range of prediction is 18.57%–34.88%. The second is the DDNN model without fluid dynamics prior knowledge constraints, and the relative error range of prediction is 7.14%–19.76%. The error between the FDCNN model prediction and the FMI observed fracture width is the smallest. The error between the predicted value and the observed value of the three samples is not more than 0.1 mm, and the relative error range is only 2.38%–8.14%. The prediction results show that the FDCNN model has better generalization performance and prediction accuracy than the fluid dynamics model and DDNN under the training conditions of small samples, which confirms that the FDCNN model has the ability to predict the target well leakage fracture width. The field operators can adjust the construction parameters based on the prediction results and formulate the plugging scheme to improve the drilling construction efficiency of the oilfield.

TABLE III. Three sets of logging data for the mud loss stratigraphy of well J-5.

Well depth (m)	Plastic viscosity (mPa s)	Pumping pressure (MPa)	Drilling mud displacement (L/s)	Drilling rate (m/h)	Mud loss rate (m^3/h)	Mud loss volume (m^3)	Yield point (Pa)	Wellbore radius (m)	Mud viscosity (mPa s)	Drilling overpressure (MPa)
2309	21	12	36	3.38	4	12	9.1	0.108	43	3.54
3169	22	10	34	4.76	9	20	9.6	0.108	59	2.20
3675	25	8	24	7	30	86	9.8	0.108	87	2.17

TABLE IV. Prediction results and relative errors for several methods.

	Fracture width predicted values (mm)			Relative error (%)		
Methods of prediction	2309 m	3169 m	3675 m	2309 m	3169 m	3675 m
FMI (observed values)	0.86	1.4	2.1	...	...	...
Verga model	0.56	0.95	1.65	34.88	32.14	21.43
Lietard–Griffiths model	0.62	1.03	1.71	27.90	26.43	18.57
DDNN (BPNN)	0.69	1.5	1.94	19.76	7.14	7.62
FDCNN	0.79	1.46	2.05	8.14	4.28	2.38

VI. CONCLUSION

In this study, a new natural fracture width prediction method based on Fluid Dynamics Constrained Neural Network (FDCNN) is proposed, aiming to improve the accuracy of fracture width prediction during oil and gas drilling and to provide scientific basis for leakage prevention and plugging operations. By analyzing the drilling practice and logging data of J-5 well, we draw the following conclusions:

1. Effectiveness of the FDCNN model: Compared with the traditional hydrodynamic model and data-driven neural network (DDNN), the FDCNN model shows better generalization ability and prediction accuracy under the small sample training conditions. The FDCNN model utilizes the a priori knowledge of hydrodynamics and optimizes the performance index of the neural network through the augmented Lagrange multiplier method, which effectively reduces the error of fracture width prediction. The low prediction accuracy of the hydrodynamic model and the difficulty that the data-driven neural network relies on a large number of data samples for training while the field conditions are difficult to meet are overcome.
2. Importance of data preprocessing and feature optimization: Spearman correlation analysis and significance test for logging data, and then further by variable redundancy analysis test, provides a scientific basis for the selection of model input parameters, which ensures the effectiveness of model training.
3. Model training and optimization: The FDCNN model is trained using backpropagation (BP) and gradient descent training methods. The experimental results show that the appropriate penalty factor and the number of hidden layer neurons are crucial for improving the prediction performance of the model.
4. Practical application verification: In the practical application of J-5 well, the relative error between the fracture width predicted by the FDCNN model and the actual measured value of FMI imaging logging is relatively small, which verifies the applicability and validity of the model in practical engineering.
5. Practicality of the model: the FDCNN model provides a new technical means for fracture width prediction during drilling in oil and gas fields, which helps to improve the efficiency and success of drilling construction and reduce economic losses.
6. Future research directions: Although the FDCNN model achieved good prediction results in this study, there is still room for further improvement. Future research can explore more a priori knowledge of hydrodynamics, further optimize the network structure, and improve the prediction ability of the model under more complex geological conditions. In addition, applying the FDCNN model to other types of fracture prediction and geoengineering problems is also a direction worth exploring in depth.

The results of this study not only enrich the theory of fracture width prediction in the field of oil and gas drilling, but also provide strong technical support for engineering practice.

ACKNOWLEDGMENTS

This work was funded by the scientific research collaboration project on wellbore leakage mechanism research of “Research on Geological Engineering and Chemical Prevention and Plugging Technology in Junggar Basin” provided by the Western Drilling Engineering Co., Ltd., of CNPC, with the Project No. XBZTGC-2023-JS-370.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Junqiao Liang: Data curation (equal); Methodology (equal); Writing – original draft (equal); Writing – review & editing (equal). **Hu Yang:** Funding acquisition (equal); Resources (equal); Supervision (equal). **Shilong Xiong:** Data curation (equal); Investigation (equal); Software (equal). **Donghua Zhang:** Data curation (equal); Software (equal); Visualization (equal). **Xinlong Xian:** Conceptualization (equal); Investigation (equal); Validation (equal). **Penggao Zhou:** Formal analysis (equal); Software (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available on request from the corresponding author.

NOMENCLATURE

I	number of input neurons with constraint effect
M	number of hidden neurons
N	total number of samples collected
q_m	mud loss rate, m^3/s
R	dimensionless radius

$r_s(t)$	invasion radius r_s at time t
$(r_s)_{\max}$	maximum invasion radius, m
r_w	wellbore radius, m
V_m	mud loss volume, m^3
w	natural fracture width, m
Δp	drilling overpressure, Pa
μ	mud viscosity, Pa s
μ_p	plastic viscosity, Pa s
$\sigma(0)$	penalty factors
τ	the shear stress, Pa
τ_y	yield point, Pa

REFERENCES

[1] A. Lavrov, "Flow of truncated power-law fluid between parallel walls for hydraulic fracturing applications," J. Non-Newtonian Fluid Mech. **223**, 141–146 (2015).

[2] D. Ye *et al.*, "A novel interdisciplinary model for optimizing coalbed methane recovery and carbon dioxide sequestration: Fracture dynamics, gas mechanics, and its application," Phys. Fluids **36**(7), 076607 (2024).

[3] Z. Tao *et al.*, "Numerical simulation study on propagation mechanism of fractures in tight oil vertical wells with multi-stage temporary plugging at the fracture mouth," Phys. Fluids **36**(7), 076603 (2024).

[4] C. Zou *et al.*, "Enhancing fluid classification using meta-learning and transformer through small-sample drilling data to interpret well logging data," Phys. Fluids **36**(7), 076608 (2024).

[5] L.-H. Bai and H. Xu, "Accurate storm surge forecasting using the encoder–decoder long short term memory recurrent neural network," Phys. Fluids **34**(1), 016601 (2022).

[6] T. D. Van Golf-Racht, *Fundamentals of Fractured Reservoir Engineering* (Elsevier, 1982).

[7] C. G. Dyke, B. Wu, and D. Milton-Tayler, "Advances in characterizing natural-fracture permeability from mud-log data," SPE Form. Eval. **10**(3), 160–166 (1995).

[8] R. Majidi, S. Miska, and J. Zhang, "Fingerprint of mud losses into natural and induced fractures," in *SPE European Formation Damage Conference*, Noordwijk, The Netherlands, April 7–11 (Society of Petroleum Engineers, 2011), Paper No. SPE-143854-MS.

[9] K. Lyu *et al.*, "Deterioration of compressive properties of coal rocks under water and gas coupling," J. Cent. South Univ. **31**(2), 477–495 (2024).

[10] R. A. Nelson, "Fractured reservoirs: Turning knowledge into practice," J. Pet. Technol. **39**(4), 407–414 (1987).

[11] O. Lietard *et al.*, "Fracture width LWD and drilling mud/LCM selection guidelines in naturally fractured reservoirs," in *SPE Europec Featured at EAGE Conference and Exhibition* (SPE, 1996).

[12] F. Sanfillippo *et al.*, "Characterization of conductive fractures while drilling," in *SPE European Formation Damage Conference and Exhibition* (SPE, 1997).

[13] F. M. Verga *et al.*, "Detection and characterization of fractures in naturally fractured reservoirs," in *SPE Annual Technical Conference and Exhibition* (SPE, 2000).

[14] J. Huang, D. V. Griffiths, and S.-W. Wong, "Characterizing natural-fracture permeability from mud-loss data," SPE J. **16**(1), 111–114 (2011).

[15] J. E. Olson, B. Bahorich, and J. Holder, "Examining hydraulic fracture-natural fracture interaction in hydrostone block experiments," in *SPE Hydraulic Fracturing Technology Conference and Exhibition* (SPE, 2012).

[16] D. Li, "Numerical and experimental investigations of drilling fluid losses in fractured formations," Ph.D. thesis (Southwest Petroleum University, Chengdu, 2012), pp. 64–75.

[17] Y. Zhao, M. Y. Deng, W. Q. Zeng, and L. S. Xiao, "Study on model prediction width of Griffiths natural fracture," Drill. Prod. Technol. **40**, 102–105 (2017).

[18] Z. W. Chen, "Downhole fracture diagnosis and mud loss control technologies bases on neural network algorithm," Drill. Fluid Completion Fluid **36**, 20–24 (2019).

[19] C. Xu *et al.*, "Dynamic fracture width prediction for lost circulation control and formation damage prevention in ultra-deep fractured tight reservoir," Fuel **307**, 121770 (2022).

[20] J. Wang and J. F. Xu *et al.*, "Prediction of fracture width of drilling fluid leakage based on neural network," Coal Geol. Explor. **51**, 81–88 (2023).

[21] M. Reichstein *et al.*, "Deep learning and process understanding for data-driven Earth system science," Nature **566**(7743), 195–204 (2019).

[22] A. Karpatne *et al.*, "Theory-guided data science: A new paradigm for scientific discovery from data," IEEE Trans. Knowl. Data Eng. **29**(10), 2318–2331 (2017).

[23] J. M. Hunter *et al.*, "Framework for developing hybrid process-driven, artificial neural network and regression models for salinity prediction in river systems," Hydrol. Earth Syst. Sci. **22**(5), 2987–3006 (2018).

[24] Y. Xu *et al.*, "Physics-informed machine learning for reliability and systems safety applications: State of the art and challenges," Reliab. Eng. Syst. Saf. **230**, 108900 (2023).

[25] K. Kashinath *et al.*, "Physics-informed machine learning: Case studies for weather and climate modelling," Philos. Trans. R. Soc. A **379**(2194), 20200093 (2021).

[26] C. Meng *et al.*, "When physics meets machine learning: A survey of physics-informed machine learning," arXiv:2203.16797 (2022).

[27] M. Jeon *et al.*, "Application of artificial neural network model to a voltage–current system in the estimation of void-fraction in a gas–liquid flow," IEEE Sens. J. **19**(13), 5098–5109 (2019).

[28] S. Ferrari and M. Jensenius, "A constrained optimization approach to preserving prior knowledge during incremental training," IEEE Trans. Neural Networks **19**(6), 996–1009 (2008).

[29] D. Merhej *et al.*, "Embedding prior knowledge within compressed sensing by neural networks," IEEE Trans. Neural Networks **22**(10), 1638–1649 (2011).

[30] H. Tamura *et al.*, "Lagrangian object relaxation neural network for combinatorial optimization problems," Neurocomputing **68**, 297–305 (2005).

[31] Z. Zhang *et al.*, "Optimal system design for weigh-in-motion measurements using in-pavement strain sensors," IEEE Sens. J. **17**(23), 7677–7684 (2017).

[32] A. M. M. Almassri *et al.*, "Self-calibration algorithm for a pressure sensor with a real-time approach based on an artificial neural network," Sensors **18**(8), 2561 (2018).

[33] H. Lin *et al.*, "Nonlinear error compensation for load cells based on the optimal neural network with an augmented Lagrange multiplier," IEEE Trans. Instrum. Meas. **64**(11), 2850–2862 (2015).

[34] P. Venkataraman, *Applied Optimization with MATLAB Programming* (John Wiley & Sons, 2009).