https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Hydrology and Earth System Sciences
Discussions

Enhancing Inverse Modeling in Groundwater Systems through Machine Learning: A Comprehensive Comparative Study

Junjun Chen[1,2], Zhenxue Dai[2,3], Shangxian Yin[4], Mingkun Zhang[5], Mohamad Reza Soltanian[6]

[1] National and Local Joint Engineering Laboratory of Internet Application Technology on Mine, China University of Mining and Technology, Xuzhou, 221008, China
[2] College of Construction Engineering, Jilin University, Changchun, 130026, China
[3] School of Environmental and Municipal Engineering, Qingdao University of Technology, Qingdao, 273400, China
[4] College of Safety Engineering, North China Institute of Science and Technology, Langfang, 065201, China
[5] Shandong Rui Yi technology development Co., Ltd., Jinan, 250000, China
[6] Departments of Geosciences and Environmental Engineering, University of Cincinnati, OH, 45220, USA

Correspondence to: Zhenxue Dai (dzx@jlu.edu.cn), Shangxian Yin (yinshx03@126.com)

Abstract. Machine learning has significantly improved inverse modeling for groundwater systems. One promising development is the tandem neural network architecture (TNNA), which integrates surrogate modeling and reverse mapping for efficient forward simulations and data assimilation. Although TNNA has shown success in groundwater inverse modeling, its application scenarios remain limited, and its advantages over conventional methods have not been fully explored. This paper aims to address these gaps by comparing the TNNA method with four conventional metaheuristic algorithms: Particle Swarm Optimization, Genetic Algorithm, Simulated Annealing, and Differential Evolution. Two synthetic solute transport numerical cases are designed, with aquifer parameters characterized by low- and high-dimensional scenarios, respectively. The surrogate model is constructed using a deep residual convolutional neural network (ResNet), selected based on a comparative evaluation against three other popular machine learning models. Inversion performance is evaluated based on the accuracy of calibrated hydraulic heads, solute concentrations, and parameter estimation errors. The results demonstrate that the TNNA algorithm yields more reliable inversion results and significantly reduces computational burden across both low- and high-dimensional cases, effectively balancing exploration and exploitation in global optimization. This study highlights the significant advantages of machine learning in advancing groundwater system inversions.

1 Introduction

Numerical models are essential for quantifying flow and mass transport dynamics within aquifers, providing significant insights into hydrological and biogeochemical processes (Steefel et al., 2005; Sanchez-Vila et al., 2010; Sternagel et al., 2021; Xu et al., 2022). However, directly measuring aquifer parameters, such as permeability fields, remains challenging due to limitations in current hydrogeological exploration techniques and budgetary constraints (Dai and Samper, 2004; Castaings et al., 2009; J. Chen et al., 2021). Inverse modeling has become a key approach for estimating these uncertain model parameters, improving the accuracy of numerical simulations (Zhou et al., 2014; Bandai and Ghezzehei, 2022; Abbas et al., 2024).

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024


Hydrology and Earth System Sciences
Discussions

Inverse modeling within Bayesian theorem-based data assimilation frameworks has garnered significant attention from the hydrogeological community over the past few decades (Scharnagl et al., 2011; X. Chen et al., 2013; J. Zhang et al., 2018; Xia et al., 2021). Among available algorithms, deterministic inversion methods are a significant category, where model parameters are estimated through maximizing the posterior distribution probability using optimization techniques (Tsai et al., 2003; N. Sun, 2013; Vrugt, 2016). One type is local optimization algorithms, which update model parameters from initial guesses towards optimal solutions according to gradient directions, such as the Gaussian-Newton method (Dragonetti et al., 2018; Qin et al., 2022) and the Levenberg-Marquardt method (Schneider-Zapp et al., 2010; Nhu, 2022). These methods are highly efficient but may converge to local optima when dealing with nonconvex inversion problems. Another category is to achieve global optima solutions through metaheuristic searches, which typically incorporate processes of exploration (to search the entire parameter space for a diverse range of estimates) and exploitation (to leverage local information to refine estimates). Popular metaheuristic algorithms include the Genetic Algorithm (GA) (Ines and Droogers, 2002; Lindsay et al., 2016), Simulated Annealing (SA) (Kirkpatrick et al., 1983; Jaumann and Roth, 2018), Differential Evolution (DE) (E. Li, 2019; Yan et al., 2023), and Particle Swarm Optimization (PSO) (Rafiei et al., 2022; Travaš et al., 2023). Nevertheless, their computational efficiency may be reduced by extensive exploration and exploitation processes in achieving globally optimal inversion results. Accurate and efficient estimation of uncertain model parameters across various scenarios remains one of the most significant challenges for developing inversion frameworks.

In recent years, machine learning has experienced rapid developments and demonstrated significant performance in addressing complex problems characterized by high dimensionality and nonlinearity (Hinton and Salakhutdinov, 2006; LeCun et al., 2015; Bentivoglio et al., 2022; Shen et al., 2023). Integrating conventional inversion methods with cutting-edge machine learning techniques has become increasingly popular in addressing the challenges of inversion studies. One effective strategy is constructing surrogate models to accelerate forward simulations, ensuring that inversion algorithms perform comprehensive searches across the entire parameter space more efficiently (Razavi et al., 2012). For instance, Zhan et al. (2021) identified lithofacies structures by utilizing a deep octave convolution residual network to construct a surrogate model for predicting solute concentrations and hydraulic heads in heterogeneous aquifers. N. Wang et al. (2021) constructed a subsurface flow surrogate model under heterogeneous conditions through physically informed neural network methods, specifically for uncertainty quantification and parameter inversion. M. Liu et al. (2023) constructed a CNN surrogate model to combine with a hierarchical homogenization method to estimate effective permeability of digital rocks. More related studies can also be found in recent reviews (Yu and Ma, 2021; J. Luo et al., 2023b; Zhan et al., 2023).

In addition to surrogate models, parameter optimization through machine learning-based reverse mapping represents another significant advancement in inversion techniques. Previous studies have outlined at least two strategies to achieve reverse mapping models. The first strategy is the data-driven approach, where reverse regressions are trained using datasets that comprise pairs of model outputs and inputs. For example, A.Y. Sun (2018) developed a regression model from hydraulic heads to heterogeneous conductivity fields using a CNN-based generative adversarial network (GAN) approach. Kuang et al. (2021) succeeded in real-time identification of earthquake focal mechanisms by training a DNN regression on seismic

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024


Hydrology and Earth System Sciences
Discussions

waveform data. Yang et al. (2022) established the relationship between gravity data and CO_2 plumes to perform real-time inversion for geologic carbon sequestration. Another strategy is to train a reverse network within the tandem neural network architecture (TNNA) integrated with a pre-trained surrogate model (i.e., forward network). The TNNA method was introduced with the advent of deep learning and has been successfully applied in computed tomography reconstruction (Adler and Öktem, 2017), nanophotonic structure inverse design (D. Liu et al., 2018; Yeung et al., 2021), and photonic topological state inverse design (Long et al., 2019). Our previous research expanded the application of the TNNA algorithm within groundwater science, evaluating its performance in reactive transport inverse modeling and improving inversion results by incorporating an adaptive update strategy to reduce local predictive errors of surrogate models. The findings indicated that accurate surrogate model predictive results around the actual parameter values yield dependable TNNA inversion outcomes (J. Chen et al., 2021).

The advantage of the TNNA algorithm is that it requires only one forward simulation per parameter update, whereas conventional metaheuristic algorithms necessitate multiple forward simulations. Despite this approach is innovative, the application of TNNA is primarily limited to low-dimensional parameter settings, leaving its advantages over conventional optimization algorithms uncertain. This study aims to comprehensively compare the TNNA method with four conventional metaheuristic algorithms across both low-dimensional and high-dimensional parameter settings. Major contributions of this study include (1) systematically improving and applying the TNNA algorithm to high-dimensional heterogeneous aquifer parameter inversion, thereby filling a significant research gap in the field, and (2) quantitatively evaluating the advantages and limitations of the TNNA method in comparison to conventional deterministic inversion methods. The inversion accuracy of the TNNA algorithm depends on the predictive accuracy of the surrogate models. Based on a comparative analysis of four machine learning models, the most accurate DNN for forward simulation will be chosen to build the surrogate model. With advances in artificial intelligence, the intended research outcomes are anticipated to significantly enhance the development of appropriate frameworks for novel inversion algorithms, providing fresh insights for future inversion studies.

This paper is structured as follows: Section 2 introduces the fundamental principles of the methodology involved in this study. Section 3 provides detailed information on numerical models for low- and high-dimensional scenarios. Section 4 presents the results and discussions. Finally, Section 5 presents a summary and conclusions drawn from this research, along with recommendations for future investigations.

2 Methodology

The nonlinear inversion optimization model of this study is formulated as follows:

$$\begin{cases} \min \sum_{i=1}^{N_{\text{obs}}} \frac{1}{\sigma_i} \left[\mathbf{y}_{\text{obs}}[i] - \hat{\mathbf{y}}[i] \right]^2 \\ \hat{\mathbf{y}} = \boldsymbol{F}_{HF}(\boldsymbol{m}) \approx \boldsymbol{F}_{Forward}(\boldsymbol{m}, \boldsymbol{\theta}_{Forward}) \\ \boldsymbol{m}^L \le \boldsymbol{m} \le \boldsymbol{m}^U \end{cases} \quad (1)$$

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024


Hydrology and Earth System Sciences
Discussions

Where $\mathbf{y}_{\text{obs}} \in \mathbb{R}^{N_{\text{obs}} \times 1}$ and $\hat{\mathbf{y}} \in \mathbb{R}^{N_{\text{obs}} \times 1}$ represent the observed data vector and the corresponding model simulation output vector. $\mathbf{y}_{\text{obs}}[i]$ and $\hat{\mathbf{y}}[i]$ refer to the ith element of the observed and simulated vectors, respectively, and σ_i denotes the standard deviation of the ith observed data. $\boldsymbol{m}^L$ and $\boldsymbol{m}^U$ are the vectors representing the lower and upper limit values of the model parameters, respectively. $\boldsymbol{F}_{HF}(\cdot)$ and $\boldsymbol{F}_{Forward}(\cdot)$ represent the high-fidelity numerical model and the surrogate model, respectively. $\boldsymbol{\theta}_{Forward}$ represents the trainable parameters of the surrogate model (Lykkegaard et al., 2021; Jiannan Luo et al., 2023a).

The uncertain model parameters $\boldsymbol{m}$ are estimated through optimization algorithms, subject to the constraints defined in the nonlinear optimization model. Specifically, the high-fidelity forward model output $\boldsymbol{F}_{HF}(\boldsymbol{m})$ is approximated by the surrogate model $\boldsymbol{F}_{Forward}(\boldsymbol{m},\boldsymbol{\theta}_{Forward})$, ensuring computational efficiency. Detailed information about for surrogate modelling methods and optimization algorithms is provide in Sections 2.1 and 2.2, respectively.

2.1 Surrogate modeling methods

As shown in Figure 1, surrogate models are developed using a data-driven strategy. The process begins by sampling model parameters from prior distributions and calculating their responses using high-fidelity numerical models. A training dataset of paired model parameters and responses is then obtained, which is used to construct surrogate models via supervised machine learning. Specifically, four popular machine learning models are evaluated for surrogate modeling: multi-output support vector regression (MSVR), fully connected deep neural network (FC-DNN), convolutional neural network (LeNet), and deep residual convolutional neural network (ResNet). Each model represents a distinct period in the development of machine learning. Despite rapid advancements in artificial intelligence, these four methods remain broadly applicable for constructing surrogate models in most groundwater modeling scenarios.

The detailed principles of MSVR and the three deep learning-based methods are illustrated in the following two sub-sections. The surrogate model for inversion will be constructed using the most accurate among them. Before constructing surrogate models, the training datasets are normalized to ensure that the values for different simulation components fall within the range of [0,1].

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Hydrology and Earth System Sciences
Discussions

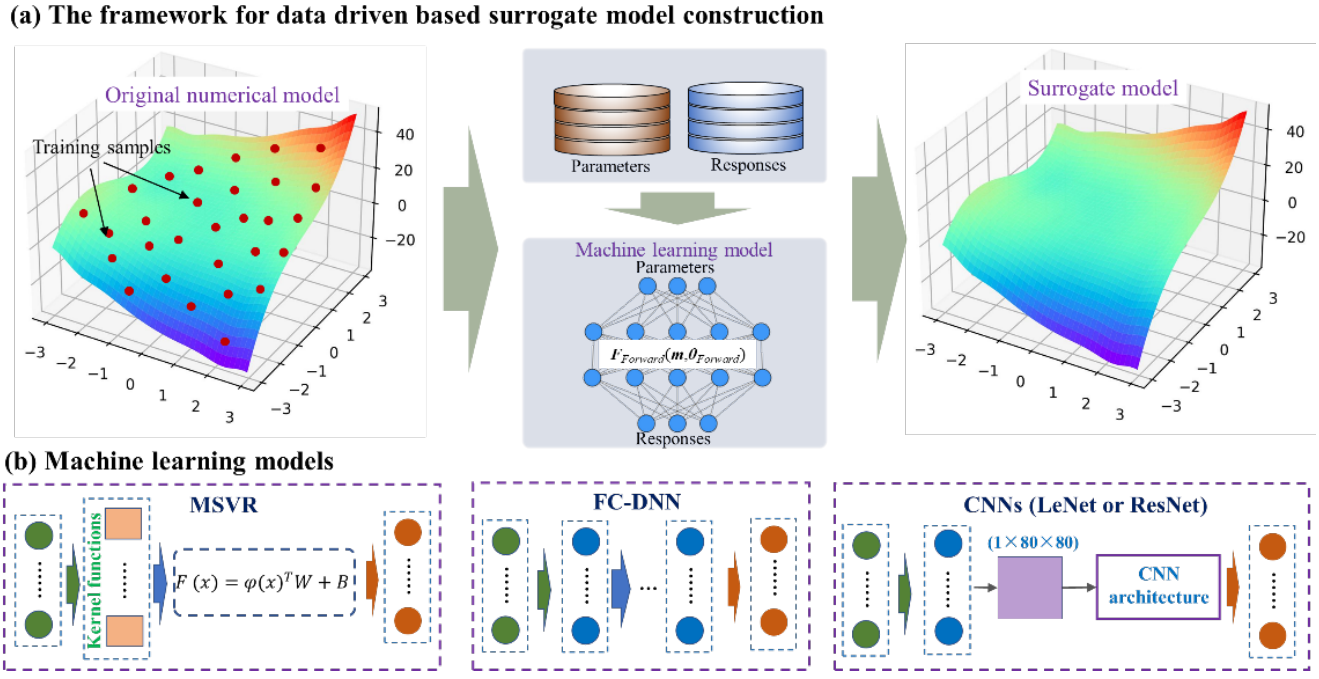


Figure 1. The framework for data-driven based surrogate model construction and the machine learning models employed.

2.1.1 MSVR

MSVR is developed from the original support vector machine (SVM) for realizing multivariate regression (Pérez-Cruz et al., 2002; Tuia et al., 2011). The mathematical expression is given as follows:

$$y = F(x)=\varphi(x)^T W+B \tag{2}$$

where $\varphi(x)$ is a nonlinear regression function. W and B are regression coefficients determined by minimizing the structural risk, as outlined in Eq.(3)~(6):

$$W,B = argminL(W,B) = \frac{1}{2}\sum_{j=1}^{N_{obs}} \left\|w^j\right\|^2 + C\sum_{j=1}^{N_{train}} L(u_i) \tag{3}$$

where N_{train} is the sample size of the training dataset; C is a penalty parameter; and L(u) is a quadratic ε-insensitive loss function, expressed as:

$$L(u) = \begin{cases} 0, & u < \varepsilon \\ u^2 - 2u\varepsilon + \varepsilon^2, & u \geq \varepsilon \end{cases} \tag{4}$$

where $u_i = \|e_i\| = \sqrt{e_i^T e_i}$; $e_i^T = y_i^T - \varphi^T(x_i)W - B^T$; ε in L(u) is the radius of the insensitive tube. For ε=0, this problem is equivalent to an independent regularized kernel least square regression for each component. For $\varepsilon \neq 0$, it becomes feasible to develop individual regression functions for each dimension based on the model outputs and to generate their corresponding support vectors. Solving the optimization problem directly is challenging, and the desired solutions for W and B are determined

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024


Hydrology and Earth System Sciences
Discussions

using an iterative reweighted least squares (IRWLS) procedure, employing the quasi-Newton approach. During the IRWLS process, the term $L(u)$ in Eq.(3) is first transformed into a discrete first-order Taylor expansion, and the corresponding quadratic programming approximation is constructed. Meanwhile, a linear expression is derived based on the principle that the first-order derivatives of the objective function with respect to W and B are zero. Finally, the optimal values of W and B are obtained through a line search. Further details on the IRWLS procedure can be found in (Sanchez-Fernandez et al., 2004).

The performance of the MSVR model is influenced by three hyperparameters: the penalty parameter C, the kernel function parameter σ and ε (Ma et al., 2022). This study optimizes these hyperparameters by minimizing the root mean square error (RMSE) using the four metaheuristic algorithms introduced in this study.

2.1.2 Deep learning based surrogate models

(1) DNN architectures

The three DNN models are all feedforward neural networks. In DNN model construction, various neural network layers can yield diverse DNN models, resulting in different predictive performances (LeCun et al., 2015). For the DNN models adopted in this study, the involved neural network types are the fully connected layer, the convolutional layer, and the residual block layer.

In fully connected layers, both input and output layers are in vector forms. Assume $\boldsymbol{X}_{\text{input}} \in \mathbb{R}^{n\times 1}$ is the input vector and $\boldsymbol{X}_{\text{output}} \in \mathbb{R}^{m\times 1}$ is the output vector. The transformation in a fully connected layer is expressed as:

$$X_{\text{output}}=\sigma\left(W\times X_{\text{input}}+B\right) \tag{5}$$

where $\sigma(\cdot)$ is a non-linear active function; $W \in \mathbb{R}^{m\times n}$ is the weight matrix; and $B \in \mathbb{R}^{m\times 1}$ is the bias vector.

In a convolutional layer, both the input and output are in matrix forms. A convolutional layer transfers information through sparse connections by several convolution kernels, essentially small matrices. The mathematical formula of a convolutional layer is as follows (Y. Wang et al., 2019; Jardani et al., 2022):

$$h_{u,v}^{q}(x_{u,v})=\sigma\left(\sum_{i=1}^{k_i^{'}}\sum_{j=1}^{k_j^{'}} w_{i,j}^{q} x_{u+i,v+j}+b\right) \tag{6}$$

where $x_{u,v}$ is the pixel value at position (u, v) of the input matrix; $h_{u,v}^{q}(x_{u,v})$ is the output feature $h_{u,v}^{q}(x_{u,v})$ calculated by employing the qth (q=1,…,N_{out}) convolutional kernel filter $\boldsymbol{w}^{q} \in \mathbb{R}^{k_i^{'}\times k_j^{'}}$. In a convolutional layer with N_{out} filters, the output matrix contains N_{out} feature layers. The output size (S_{out}) of each convolutional layer is determined by the input size (S_{in}) and the hyperparameters (i.e., zero padding p, kernel size k' and stride s). A pooling layer is often used after a convolutional layer to remove redundant information from the extracted features and improve the efficiency of model training (J. Chen et al., 2021).

The residual block is a fundamental component of residual networks (ResNets). It is designed to mitigate the vanishing and exploding gradients commonly encountered in the training of deep neural networks. In a residual block, an intermediate layer is designed to learn a residual mapping, $F(x)=H(x)-x$ (or $H(x)-G(x)$, where $G(x)$ represents another transformation of x).

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024


Hydrology and Earth System Sciences
Discussions

Here, x is the input to the block. The output of the block is then computed as $F(x)+x$, which is intended to approximate $H(x)$. This design ensures that the output of the module at least replicates the input, thus avoiding overcoming the challenges posed by vanishing gradients. The mathematical formula of a residual block is expressed as follows:

$$y_l = F(x_l, W_l) + x_l \tag{7}$$

$$x_{l+1} = f(y_l) \tag{8}$$

where x_l and W_l are the input data and the connection weight matrix for the l-th residual block, respectively. $F(\bullet)$ is the residual function. Within this framework, the function $f(\cdot)$ is configured as an identity map, such that $x_{l+1}=y_l$. Then, the relationship between the L-th residual block in a deeper layer and the l-th residual block is expresses as follows (He et al., 2016):

$$x_L = x_l + \sum_{i=1}^{L-1} F(x_i, W_i) \tag{9}$$

According to the chain rule in derivatives, the gradient of the loss function ε with respect to x_l can be expressed as:

$$\frac{\partial \varepsilon}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i, \omega_i)\right) \tag{10}$$

This formulation highlights two key properties of the residual network. First, the gradient does not vanish during network training processes because the term $\frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i, \omega_i)$ is never equal to -1. Second, the gradient of the deepest residual block $\frac{\partial \varepsilon}{\partial x_L}$ can directly affect all preceding layers, ensuring effectively transmission of gradients throughout the network (Chang et al., 2022).

The FC-DNN of this study is constructed using fully connected layers, and each hidden layer consists of 512 neurons. The activation function for the output layer is Sigmoid, and the other hidden layers use Swish. The number of hidden layers n is determined by comparing the model prediction accuracy with different configurations, where n varies from 1 to 7. For the LeNet and ResNet models, the initial processing maps the input vector to a fixed matrix shaped 1×80×80 using a combination of a fully connected layer and a reshaped layer, as shown in Figure 1(b). Specifically, LeNet consists of two convolutional blocks and two fully connected layers. Each convolutional block consists of a convolutional layer followed by a max-pooling layer. The fully connected layers have 1024 and 512 neurons, respectively. ResNet consists of four stages and two different Res blocks are adopted. The first stage includes two residual units without down-sampling, while the remaining three stages each have one residual unit with down-sampling and one residual unit without down-sampling. Activation functions in all layers are Rectified Linear Units (ReLUs), except for the output layer, where Sigmoid activation is used. Detailed architecture information for LeNet and ResNet is provided in Figure S1 and Figure S2, respectively.

(2) DNN model training

The purpose of a surrogate model is to minimize the difference between the predicted outputs $\hat{y}_i = f_{DNN}(m_i, \theta_{DNN})$ and the numerical modeling outputs y_i. Consequently, the loss function is formulated with L1 norm constraints:

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024


Hydrology and Earth System Sciences
Discussions

$$\theta_{DNN}=\text{argmin}\ \frac{1}{N}\sum_{i=1}^{N}|F_{DNN}(m_i,\theta_{DNN})\text{-}y_i|+\frac{w_d}{2}\theta_{DNN}^{T}\theta_{DNN} \tag{11}$$

where w_d is the weight decay to avoid overfitting, referred to as the regularization coefficient. This study implemented the DNN models using PyTorch (https://pytorch.org/), a widely used machine learning framework. The neural network weights were initialized using the default initialization method of PyTorch and optimized using the stochastic gradient descent method via the Adam algorithm.

2.2 Optimization algorithms

2.2.1 Metaheuristic algorithms

(1) Particle swarm optimization algorithm

Particle swarm optimization (PSO) is a population-based intelligent optimization algorithm inspired by the foraging behavior of birds (Eberhart and Kennedy, 1995). It is realized through the following steps:

Step 1: Initialize a population with n particles of a m-dimensional space $X=(X_1,X_2,\ldots,X_n)$. For an arbitrary particle (i), denote its position, velocity and best position at the kth iteration as $X_i^k=(x_{i1}^k,\ldots,x_{im}^k)$, $V_i^k=(v_{i1}^k,\ldots,v_{im}^k)$, and $P_i^k=(p_{i1}^k,\ldots,p_{im}^k)$, respectively.

Step 2: Calculate the best solution for each particle ($X_{pbest_i^k}$) according to Eq.(12):

$$X_{pbest_i^k}=\begin{cases} X_{pbest_i^{k-1}}, & f(X_i^k)\geq f(X_{pbest_i^{k-1}}) \\ X_i^k, & f(X_i^k)<f(X_{pbest_i^{k-1}}) \end{cases} \tag{12}$$

where $f(\cdot)$ is the objective function, also known as the fitness function.

Step 3: Calculate the best position of the population ($X_{gbest_i^k}$) according to Eq.(13).

$$X_{gbest_i^k}=\min\{f(X_1^k),\ldots,f(X_n^k)\} \tag{13}$$

Step 4: Updated the velocity and position for each particle (i) according to Eq.(14) and Eq.(15):

$$V_i^{k+1}=w_iV_i^k+r_1c_1(X_{pbest_i^k}\text{-}X_i^k)+r_2c_2(X_{gbest_i^k}\text{-}X_i^k) \tag{14}$$

$$X_i^{k+1}=X_i^k+V_i^{k+1} \tag{15}$$

where c_1 and c_2 are learning parameters, generally taken as two equal non-negative constants and are set to 0.5 and 0.1 here; r_1 and r_2 are two random values within the range of [0, 1]; w_i is the inertia weight and set to 0.8 for this study.

(2) Genetic algorithm

Genetic algorithm (GA) is initially introduced by Holland John (1975). It draws inspiration from natural evolution and genetics, where individuals within a population are selected or eliminated based on their adaptability to the environment. The GA is realized through the following steps:

Step 1: Generate an initial population $X=(X_1,X_2,\ldots,X_n)$ randomly.

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024


Hydrology and Earth System Sciences
Discussions

Step 2: Perform binary encoding on all individuals in the population X to obtain their respective binary symbol strings. These binary symbol strings are called chromosomes, and each value ("0" or "1") on a symbol string is called a gene.

Step 3: Crossover: Perform crossover operations on randomly paired combinations of individuals in X. The essence of crossover is to exchange some values in the symbol strings of a pair of individuals.

Step 4: Mutation: Perform mutation operations on some random individuals in X by changing some values of their symbol strings.

Step 5: Selection: Perform selection operations based on the fitness values of each individual (X_i) to generate the next generation population. This step is realized through the roulette wheel selection method, where individuals with higher fitness values are more likely to be selected.

Step 6: Determine whether the current results satisfy the iteration termination condition. If not, return to step (2); otherwise, output the optimal individual in the current population as the final result.

(3) Simulated Annealing

The SA method is a Monte Carlo-based stochastic optimization algorithm proposed by Metropolis et al. (1953) and initially applied to combinational optimization problems by Kirkpatrick et al. (1983). The realization steps for SA method are as follows:

Step 1: Set the starting temperature as T_0 and draw an initial optimal solution as X_i.

Step 2: Generate a new solution X_j from the neighborhood of the current solution X_i.

Step 3: Calculate the objective function values $f(X_i)$ and $f(X_j)$. If $f(X_i) \geq f(X_j)$, then X_j becomes the current solution X_i; otherwise, X_j becomes the current solution X_i with a probability calculated as:

$$P(X_i \rightarrow X_j) = \exp\left(\frac{f(X_i)-f(X_j)}{a^t T_0}\right) \tag{16}$$

where t is the current time and a is the temperature decay constant.

Step 4: Under the current temperature conditions, repeat steps (2) and (3) until reaching the predetermined number of internal iterations. Then, update the temperature and time as follows: set $t=t+1$ and $T_t=a_t T_0$, then proceed to the next step.

Step 5: Return to step (2) and continue the iteration according to the new temperature (T_t) and time (t) until the termination conditions are met. The iterations in this step can be considered outer iterations, distinguished from step (4).

(4) Differential evolution

DE is another evolutionary algorithm proposed by Storn and Price (1997). Similar to GA, DE also employs mutation, crossover and selection operators, but they update uncertain model parameters in different ways (Tran et al., 2022). The detailed steps for realizing DE are as follows:

Step 1: Generate the initial population $X=(X_1,X_2,\ldots,X_n)$ randomly.

Step 2: Perform encoding for each individual in X. The encoding method used in DE is floating-point real encoding, rather than binary encoding used in GA.

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024


Hydrology and Earth System Sciences
Discussions

Step 3: Mutation**:** After completing individual encoding, DE performs mutation operations to generate new individuals according to Eq. (17):

$$x_{perturbed}(g+1)=x_{rand_1}(g)+F_{DE}\times(x_{rand_2}(g)-x_{rand_3}(g)) \quad (17)$$

where x_{rand_1}, x_{rand_2} and x_{rand_3} are randomly selected individuals among the candidate solutions of the current population and must be different from each other. F_{DE} is a scaling parameter within the range of [0,1], controlling differential variations. g represents the sequence number of iterations.

Step 4: Crossover: Perform crossover operations to generate the trial vector by combining the mutant and target vectors. The formula for this step is as follows:

$$u_j(g+1)=\begin{cases} x^j_{perturbed}(g+1) & \text{if } P_j\leq CR \\ x_j(g) & \text{if } P_j>CR \end{cases} \quad (18)$$

where P_j is a random number in the range of [0,1], CR is the crossover rate. If some variables of the trial vector have the same values, keep one of them and reset the others with random integer numbers in the range [1, D].

Step 5: Selection: Perform selection operations to determine whether the new generated trial vector $u_j(g+1)$ can survive the next generation, $x_j(g+1)$. Therefore, a candidate solution replaces the parent only if it has better objective function value.

Step 6: Return to step 3 until the convergence criteria are met.

2.2.2 TNNA algorithm

The TNNA algorithm aims to obtain a reverse network that maps the observation vector to model parameters, as shown in Eq. (19).

$$m=F_{Reverse}(\tilde{y}_{\mathrm{obs}},\theta_{Reverse}) \quad (19)$$

where $\theta_{Reverse}$ are the trainable parameters of $\boldsymbol{F}_{\mathrm{Reverse}}$. The training of the reverse network is guided by the constraints of the nonlinear optimization model defined in Eq. (1). The loss function for training is expressed as follows:

$$\theta_{Reverse}=argmin\sum_{i=1}^{N_{obs}}\frac{1}{\sigma_i}\left[\hat{y}_{obs}[i]-F^i_{Forward}(F_{Reverse}(\hat{y}_{obs},\theta_{Reverse}),\theta_{Forward})\right]^2 \quad (20)$$

The $\boldsymbol{F}_{\mathrm{Reverse}}$ is also trained within the pytorch framework. The required training data here are the normalized observation data. Specifically, the reverse network for this study is designed using an FC-DNN with three hidden layers, each containing 512 neurons.

During reverse network training processes, each iteration of updating the trainable parameters $\theta_{Forward}$ involves two steps: First, the vector $\hat{y}_{obs}$ is input into the reverse network F_{Reverse} to obtain the parameter prediction $\tilde{m}$. This predicted parameter $\tilde{m}$ is then input into the forward network F_{Forward} to generate the corresponding forward prediction results. Subsequently, the trainable parameters $\theta_{Reverse}$ of the reverse network are updated based on the error feedback from the loss function in Eq. (20) through DNN model training. This process demonstrates that F_{Reverse} and F_{Forward} are connected in a TNNA, wherein the forward simulation realization is executed once during each epoch to update the trainable parameters of $\theta_{Reverse}$. This is a marked

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Hydrology and Earth System Sciences
Discussions

difference from the four selected metaheuristic algorithms, which require numerous forward simulations for each update of estimated model parameters. Upon completion of F_{Reverse} training, the final optimal parameters are predicted by inputting observation data into F_{Reverse}. Further details on TNNA can be found in (J. Chen et al., 2021).

3. Case Study

We introduce two numerical cases to compare the TNNA algorithm with conventional metaheuristic algorithms: one with low-dimensional parameters and the other with high-dimensional parameters. Initially, four surrogate models will be assessed using the low-dimensional parameter case, and the model with the highest accuracy will be integrated into the inversion framework. Based on hypothetical observation scenarios, we will compare the inversion performance of the TNNA method and the four metaheuristic algorithms in low- and high-dimensional cases.

3.1 Case 1: Low-dimensional zoned permeability field scenario

As shown in Figure 2, the numerical model for the low-dimensional scenario focuses on conservative solute transport within a zoned permeability field. The model domain is a two-dimensional rectangular area measuring 10m×20m. The left and right boundaries are Dirichlet boundary conditions, with a hydraulic head difference of 1 m. The heterogeneous permeability is divided into eight homogeneous permeability zones, denoted as k_1 to k_8. The prior range for these eight permeabilities is from 1×10^{-12} to 9.9×10^{-12} m^2. The contaminant source is located at the left boundary with a fixed release concentration ranging from 1×10^{-3} to 1 mol/L. The simulation area is uniformly discretized into 3,200 (40×80) meshes, and the simulation time is set to 20 days.

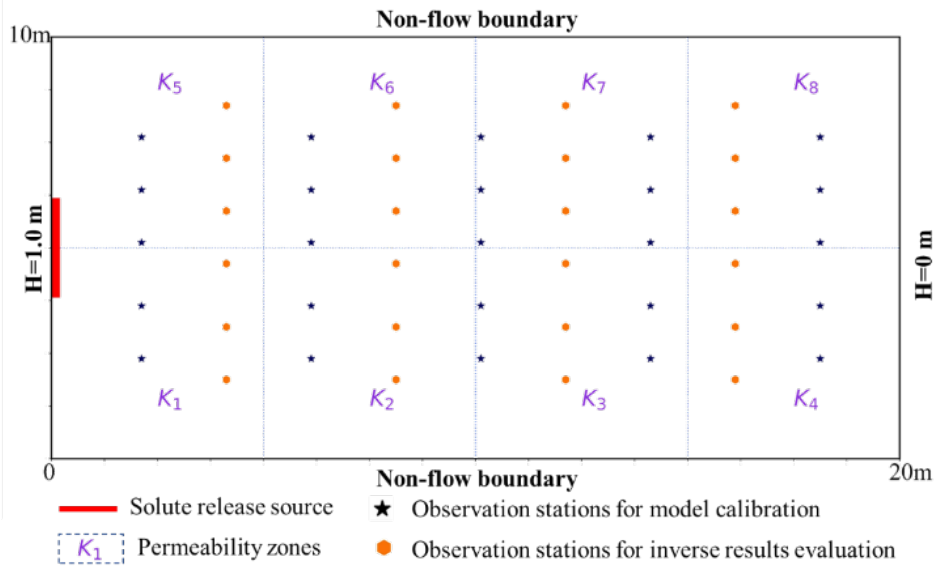


Figure 2. Flow domain of the solute transport model for the low-dimensional scenario.

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024


Hydrology and Earth System Sciences
Discussions

According to these model conditions, there are nine uncertain model parameters to be estimated: eight permeability parameters (k_1 to k_8) and the source release concentration. As shown in Figure 2, these parameters will be estimated using the observation data of hydraulic heads and solute concentrations collected from 25 locations, denoted by black pentagrams. Additionally, observation data from another 24 locations, denoted by orange hexagons, will be used to validate the prediction accuracy of the calibrated numerical model.

3.2 Case 2: High-dimensional gaussian random permeability field scenario

The numerical model for the high-dimensional scenario features a domain size of 10m×10m, with impervious upper and lower boundaries and constant head boundaries at the left (1m) and right (0m) sides. The domain is discretized into 4,096 (64 ×64) grids. The log-permeability field follows a Gaussian distribution, and the permeability value of the i-th mesh is defined as follows:

$$k_i=\alpha_i k_{ref} \tag{26}$$

where k_{ref} is the reference permeability, set to 2×10^{-13}m^2. The modifier α for the logarithmic Gaussian random field satisfies the following formula:

$$\alpha(\boldsymbol{s}) = \exp\big(G(\boldsymbol{s})\big), G(\cdot) \sim N\big(m, C(\cdot,\cdot)\big), \tag{27}$$

where $m = 0$ is the constant mean and L_2 exponentiated quadratic covariance function for two arbitrary spatial locations, $\boldsymbol{s} = (s_x, s_y)$ and $\boldsymbol{s}' = (s_x', s_x')$:

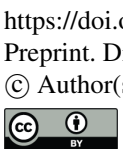
$$C\left(\boldsymbol{s}, \boldsymbol{s}'\right) = \sigma_G^2 \exp\left(-\sqrt{\left(\frac{s_x - s_x'}{\lambda_x}\right)^2 - \left(\frac{s_y - s_y'}{\lambda_y}\right)^2}\right), \tag{28}$$

where $\sigma_G^2 = 2$ is the variance and λ_x=λ_y= 2.5 m are the correlation lengths along the x and y directions, respectively.

The Karhunen-Loève expansion (KLE) is utilized to parameterize the permeability field (D.X. Zhang and Lu, 2004). In this case, 100 KLE terms are used to preserve more than 92.67% of the field variance. Consequently, estimating the permeability field is equivalent to identifying these 100 KLE terms. The observational data used for inverse modeling include hydraulic heads from a stable flow field and solute concentrations measured every two days over 40 days, starting from the 2nd day to the 40th day (day: t= 2i, i=1,…,20). To mitigate inversion errors arising from equifinality, actual permeability values at observed locations are included as regularization constraints. The standard deviation of Gaussian noise for the normalized observations is set to 0.01.

As the degrees of freedom significantly increase in high-dimensional models, the influence of observation data on inversion results becomes increasingly significant. Five scenarios with different monitoring networks are considered to comprehensively evaluate the performance of different inversion algorithms using various observations. Figure 3 displays the monitoring station locations for each scenario.

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024



Hydrology and Earth System Sciences
Discussions

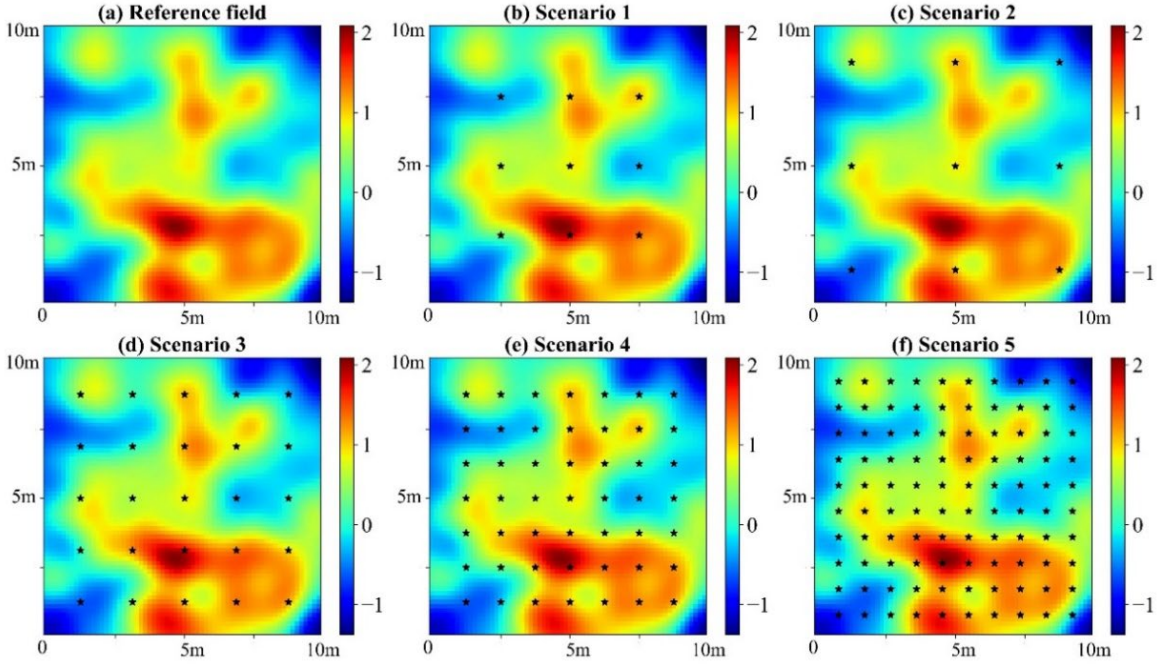


Figure 3. The reference log-permeability field and locations of observation stations for five scenarios. The observation stations are represented by black pentagrams.

4. Results and discussion

4.1 Surrogate model evaluations

Surrogate models were first compared using the low-dimensional parameter case. Four training datasets $\boldsymbol{D}_{train}=\{\boldsymbol{M}_{train},\boldsymbol{Y}_{train}\}$ with 200, 500, 1000, and 2000 samples (represented as $\boldsymbol{D}_{train\text{-}200}$, $\boldsymbol{D}_{train\text{-}500}$, $\boldsymbol{D}_{train\text{-}1000}$ and $\boldsymbol{D}_{train\text{-}2000}$, respectively) and a testing dataset $\boldsymbol{D}_{test}=\{\boldsymbol{M}_{test},\boldsymbol{Y}_{test}\}$ with 100 samples (represented as $\boldsymbol{D}_{test\text{-}100}$) are prepared. These datasets were generated using Latin hypercube sampling (LHS) and numerical simulations. The predictive accuracy of surrogate models was quantitatively evaluated using root mean square error (RMSE) and determination coefficient (R^2) metrics (J. Chen et al., 2022).

For solute transport inverse modeling problems, it is crucial to consider observations of both hydraulic heads and solute concentrations simultaneously. Therefore, the surrogate model within an inversion framework should have accurate predictive capabilities for hydraulic heads and solute concentrations. This study calculates RMSE and R^2 values separately for hydraulic heads, solute concentrations, and all model response data, resulting in the following evaluation criteria: $RMSE_{ALL}$ and R^2_{ALL} for overall data, $RMSE_H$ and R^2_H for hydraulic heads, and $RMSE_C$ and R^2_C for solute concentrations.

Figure 4 and Figure 5 display the RMSE and R^2 values of each surrogate model, and Figure S3~Figure S6 present the pairwise comparison results. The optimal values for C, σ, and ε in the MSVR method are provided in Table S1. Moreover, the

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Hydrology and Earth System Sciences
Discussions

optimal number of hidden layers in the FC-DNN for $\boldsymbol{D}_{train\text{-}200}$, $\boldsymbol{D}_{train\text{-}500}$, $\boldsymbol{D}_{train\text{-}1000}$ and $\boldsymbol{D}_{train\text{-}2000}$ are 2, 4, 3, and 3, respectively, as determined by the corresponding $RMSE_{\mathrm{All}}$ and R^2_{All} values in Table S2 and Table S3.

According to the performance criteria in Figure 4 and Figure 5, the prediction accuracy of each surrogate model significantly improves with an increasing number of training samples. Based on $RMSE_{\mathrm{All}}$ and R^2_{All} values, their performance ranks as follows: ResNet, LeNet, FC-DNN, and MSVR. The MSVR method accurately predicts hydraulic heads but performs the worst in predicting solute concentration. Training MSVR with the four prepared datasets, the $RMSE_H$ values are below 0.02, and R^2_H values are near 1. Notably, with a training sample size of 200, the prediction accuracy of MSVR for hydraulic heads is higher than that of FC-DNN and LeNet, as indicated by their $RMSE_H$ and R^2_H values, closely matching that of ResNet. However, when using 200 training samples, the $RMSE_C$ value for MSVR exceeds 0.08, and the R^2_C value falls below 0.85. Even with a dataset size of 2000, the enhancement in the MSVR-based surrogate model is limited, as the $RMSE_C$ value remains around 0.05, and the R^2_C value stays below 0.95. FC-DNN demonstrates a significant advantage over MSVR in predicting solute concentration, particularly with larger training sample sizes of 1000 or 2000. However, there are still some obvious biases between some surrogate modeling results and their numerical modeling results (see Figure S2(d)). When adopting CNN-based surrogate models (LeNet and ResNet), the prediction accuracy for solute concentrations significantly improves (see Figure 4(b) and Figure 5(b)). With training datasets of 2000 samples, LeNet and ResNet achieve RMSE values below 0.02 and R^2 values close to 1. It is worth noting that the ResNet performs well even with smaller sample sizes. For example, with 200 training samples, the $RMSE_C$ and R^2_C values for LeNet are around 0.06 and 0.9, respectively, while these criteria values for ResNet are around 0.04 and 0.95 (see Figure 4(b) and Figure 5(b)). As the number of training samples increases, the advantages of ResNet become more apparent. According to Figure S4(d), when the training sample size reaches 2000, the prediction results of ResNet are closely consistent with the numerical simulation results for both hydraulic heads and solute concentrations.

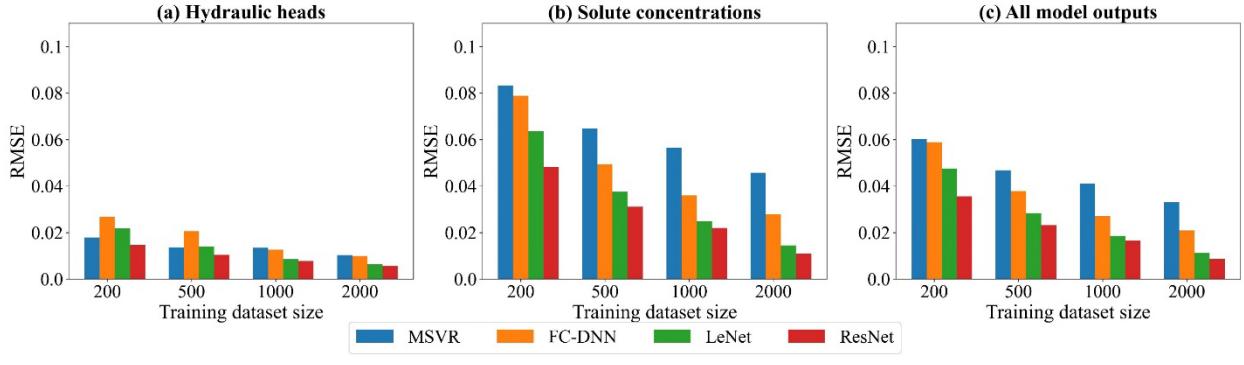


Figure 4. The RMSE results of surrogate model predictions. (a)~(c) are respectively the RMSE values of hydraulic heads, solute concentrations and all model outputs.

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Hydrology and Earth System Sciences
Discussions

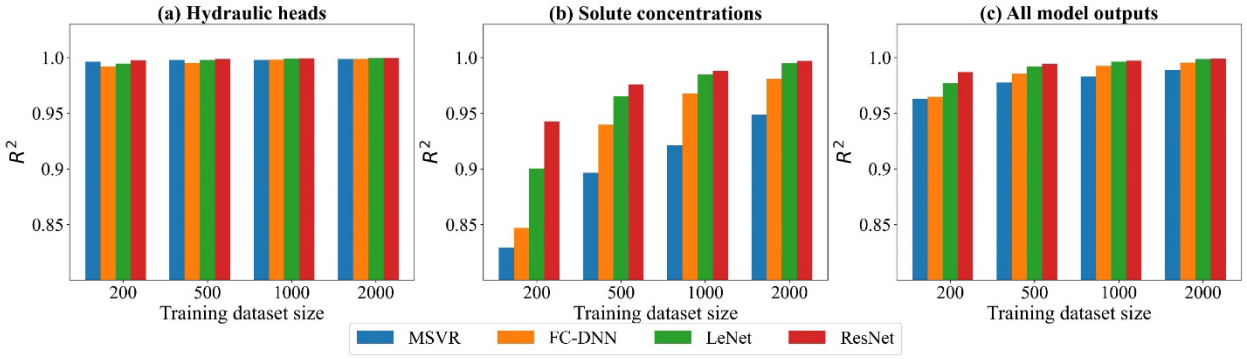


Figure 5. The R^2 results of surrogate model predictions. (a)~(c) are respectively the R^2 values of hydraulic heads, solute concentrations and all model outputs.

The comparison results of the surrogate models reflect a trend of enhanced robustness attributable to advancements in machine learning methodologies. Different machine learning approaches employ distinct strategies for achieving nonlinear mappings in developing surrogate models. Generally, deeper or larger models contain more trainable parameters, resulting in higher degrees of freedom to capture more robust nonlinear relationships. The essence of machine learning development lies in addressing the challenge of training these complex DNNs. Current state-of-the-art machine learning techniques have demonstrated proficiency in training each of the four selected surrogate modeling methods. With sufficient training samples, a surrogate model of greater complexity exhibits enhanced capability in representing higher levels of non-linearity (LeCun et al., 2015; He et al., 2016). This also explains why, despite having a sufficient number of training samples, the improvement in prediction accuracy of the MSVR for solute concentration is limited. In CNNs, sparse connections and weight-sharing in convolutional layers reduce redundant weight parameters in DNNs, enhancing the feature extraction of hidden layers. Consequently, LeNet demonstrates better performance than FC-DNN. The ResNet, which employs residual blocks in conjunction with convolutional layers, effectively addresses the issues of gradient vanishing and exploding, making the successful training of deeper CNNs possible.

According to J. Chen et al. (2021), a more globally accurate surrogate model can enhance the performance of TNNA inversion results. Thus, we selected the ResNet trained with 2000 samples for the subsequent inversion procedure. In the low-dimensional scenario, its RMSE values for hydraulic head and solute concentration data are less than 0.02, with R^2 values greater than 0.99. Subsequently, the surrogate models for the five high-dimensional scenarios designed in this study were all constructed using the ResNet-2000. An additional 500 samples were used to assess the accuracy of their predictions. The RMSE values for hydraulic head and solute concentration data are less than 0.02, and the R^2 values are greater than 0.99, as shown in Table 1. Hence, inversion simulations in the five high-dimensional scenarios of this study are also appropriate to the ResNet-2000 surrogate model.

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Hydrology and Earth System Sciences
Discussions

400 **Table 1.** The *RMSE* and R^2 values for surrogate model predictions in designed five high-dimensional scenarios.

	RMSE			R^2		
	$RMSE_H$	$RMSE_C$	$RMSE_{\mathrm{All}}$	R_H^2	R_C^2	R_{All}^2
Scenario 1	0.0108	0.0174	0.0172	0.9990	0.9980	0.9982
Scenario 2	0.0102	0.0138	0.0136	0.9995	0.9989	0.9990
Scenario 3	0.0120	0.0165	0.0163	0.9991	0.9981	0.9983
Scenario 4	0.0123	0.0161	0.0159	0.9990	0.9984	0.9985
Scenario 5	0.0137	0.0156	0.0155	0.9989	0.9985	0.9986

4.2 Parameter inversion method comparison results

4.2.1 Inversion results of the low-dimensional parameter scenario

For the low-dimensional parameter scenario, the performance of optimization algorithms is thoroughly evaluated across 100 parameter scenarios using the Monte Carlo strategy. The observation data for these scenarios are derived from the testing
405 dataset after adding Gaussian random noise $\varepsilon \sim N(0,0.01)$. The population sizes of GA, DE, and PSO, along with the chain length in SA, are set in four distinct scenarios: 20, 40, 60 and 80 (these population size or chain length values are represented as N_{PC} in subsequent discussions). These settings determine the number of forward modeling calls required for each iteration, significantly influencing the convergence rate and computational efficiency of optimization procedures. Maximum iterations for these four metaheuristic algorithms are set to 200. The learning rate, epoch number and weight decay for the TNNA
410 algorithm are set to 6×10^{-5},1000, and 1×10^{-6}, respectively.

The performance of the five optimization algorithms is evaluated according to three aspects: average convergence efficiency and accuracy in inversion procedures, predictive accuracy of calibration models for hydraulic heads and solute concentrations, and statistical analysis of the estimated errors for each model parameter. Figure 6 presents the logarithmic average convergence curves of four metaheuristic algorithms and the TNNA algorithm throughout 100 parameter scenarios.
415 Specifically, sub-figures (a)~(d) represent the N_{PC} values for metaheuristic algorithms set at 20, 40, 60, and 80, respectively. These figures clearly illustrate the average convergence speed and accuracy of five optimization algorithms. Figure 7 displays the comparison of calibration and validation between the simulation results and the observed values across all 100 parameter scenarios. Sub-figures (a) and (b) illustrate the comparative prediction fit at the 25 observation locations used for model calibration, whereas sub-figures (c) and (d) display the comparative prediction fit at the 24 observation locations. In this figure,
420 distinct symbols are used to represent the five optimization algorithms. It should be noted that the N_{PC} values for the four metaheuristic algorithms are uniformly set to 80 during this comparison. Figure 8 illustrates the probability density curves of the estimation errors for nine model parameters across 100 parameter scenarios, with different colours representing the five optimization algorithms.

The results in Figure 6 demonstrate that the TNNA algorithm achieves the best convergence accuracy, with its
425 convergence logarithmic objective function value (i.e., approximately -4.4) being smaller than those of the other four

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Hydrology and Earth System Sciences
Discussions

metaheuristic algorithms across these N_{PC} settings. The influence of N_{PC} on the convergence speeds of these four metaheuristic algorithms is not significant, exhibiting a distinct transition from rapid to slower convergence around the 75th iteration. As N_{PC} increased from 20 to 80, each metaheuristic algorithm showed distinct improvements in the accuracy of the final objective function. The DE algorithm showed the least improvement in final convergence accuracy as the N_{PC} value increased from 20 to 80, with the logarithmic value of its objective function dropping from just above -4.0 to slightly below -4.0. The SA algorithm also showed limited improvement, with its logarithmic average convergence value increasing from around -4.1 at N_{PC}=20 to slightly below -4.3 at N_{PC}=80, close to that of the TNNA algorithm. Among the four metaheuristic algorithms, SA exhibited the highest average convergence accuracy. Contrary to the SA and DE algorithms, the PSO and GA algorithms significantly enhanced average convergence accuracy as N_{PC} increased. Specifically, as N_{PC} increased from 20 to 80, the logarithmic convergence values of PSO and GA decreased by more than 0.5. While increasing N_{PC} values may help metaheuristic algorithms reduce the gap in average convergence accuracy compared to the TNNA algorithm, larger N_{PC} settings also require additional computational burdens. The above results indicate that the TNNA algorithm has a significant efficiency advantage over the four metaheuristic algorithms in parameter optimization. For instance, the DE algorithm requires 32,000 forward model realizations (80×2×200) when N_{PC} is set to 80, while the other three metaheuristic algorithms (PSO, GA, and SA) each require 16,000 realizations (80×200). In significant contrast, the TNNA algorithm requires only one forward model realization per iteration, resulting in 200 realizations. These comparisons illustrated that the TNNA method is more effective than the other four metaheuristic algorithms in achieving robust convergence results.

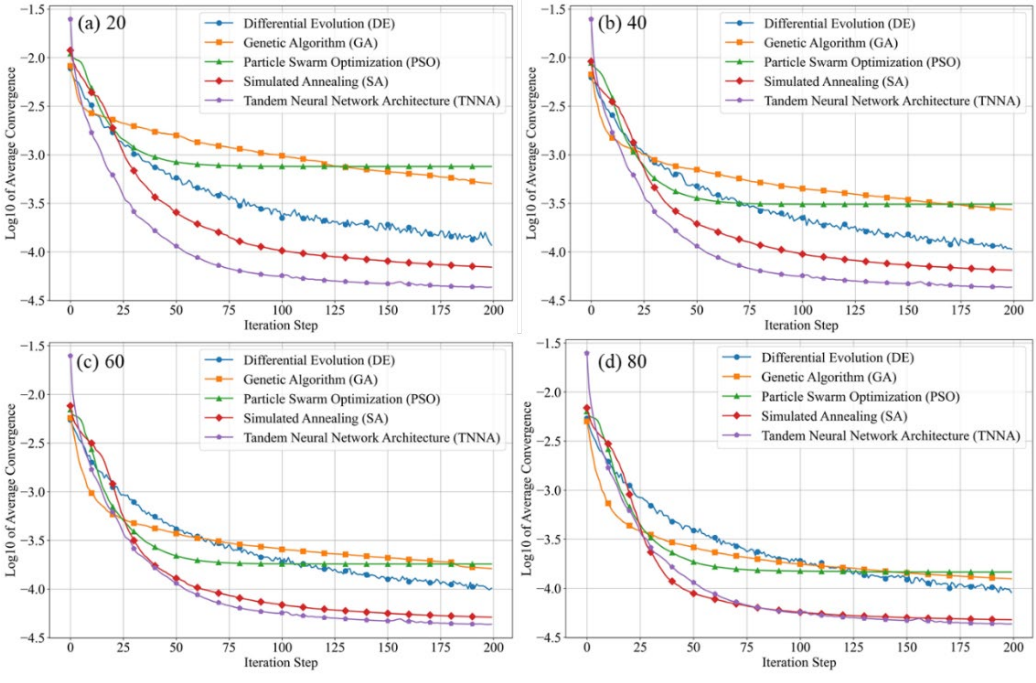


Figure 6. Comparative convergence trends of five optimization algorithms (Markers indicate convergence values at every 10 steps to indicate convergence values; for TNNA, only the first 200 out of 1000 iterations are presented).

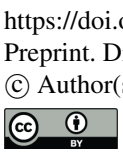
https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024


Hydrology and Earth System Sciences
Discussions

The results presented in Figure 7 indicate that, among the five optimization algorithms, the TNNA algorithm achieves the smallest RMSE values and R^2 values closest to 1.0 for both hydraulic heads and solute concentration during model calibration and validation. Furthermore, the distribution of comparison points demonstrates that the calibrated and validated modeling results of the TNNA algorithm are more accurately matched with their actual values than the other four metaheuristic algorithms, particularly for solute concentrations. Among the four metaheuristic algorithms, SA and DE outperform GA and PSO regarding RMSE and R^2 values. During model calibration and validation, PSO exhibits the worst predictive accuracy, recording the highest RMSE and R^2 values for both hydraulic heads and solute concentrations. It is noteworthy that the RMSE and R^2 values for SA during hydraulic head calibration are 0.0085 and 0.9992, respectively, while those for DE during solute concentration calibration are 0.0112 and 0.9969. These values are almost equal to those of the TNNA algorithm. The robustness of an inversion algorithm is determined by its accuracy in both calibration and validation for hydraulic heads and solute concentrations. However, DE and SA demonstrate appropriate calibration accuracy only for one of the two simulation components. Overall, the TNNA algorithm provides more robust model calibration and validation results than the other four metaheuristic algorithms.

Figure 8 indicates that the estimated error distributions for the nine model parameters derived from the TNNA algorithm are more concentrated than those obtained from the four metaheuristic algorithms. The mean estimated error values for the nine numerical model parameters using the TNNA algorithm are also the lowest. These results highlight the high accuracy and reliability of the TNNA inversion algorithm. Among the four metaheuristic algorithms, DE and SA outperform GA and PSO. This is because the probability density curves of estimation errors for the nine parameters using DE and SA are more concentrated around zero, with mean values lower than those of GA and PSO. The DE algorithm shows a more concentrated distribution around zero for the overall estimation errors of parameters K_1 to K_8. In contrast, the SA reveals reduced estimation errors for the C_0 parameter in most cases, ranking just behind the TNNA algorithm. GA outperforms PSO in estimation accuracy for seven of the nine model parameters, with PSO matching its probability density curves to that of GA only for parameters K_2 and K_4. As a whole, the statistical results of the estimated model parameter errors illustrate that the machine learning-based TNNA algorithm exhibits enhanced inversion performance compared to the four metaheuristic optimization algorithms. However, the findings also reveal that none of the five algorithms consistently offers completely reliable inversion solutions across all scenarios. For example, the TNNA algorithm, despite its generally better performance, demonstrates estimation errors as high as 0.4 for parameters K4 and K6 in some scenarios. Such results are likely because the provided observational data cannot ensure equifinality in some scenarios. In these cases, it is essential to introduce additional regularization constraints to attenuate the equifinality (G.S. Wang and Chen, 2013; Arsenault and Brissette, 2014). These findings emphasize the importance of employing the Monte Carlo method in comparative studies of inversion algorithms to ensure comprehensive evaluations and avoid misleading conclusions.

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024



Hydrology and Earth System Sciences
Discussions

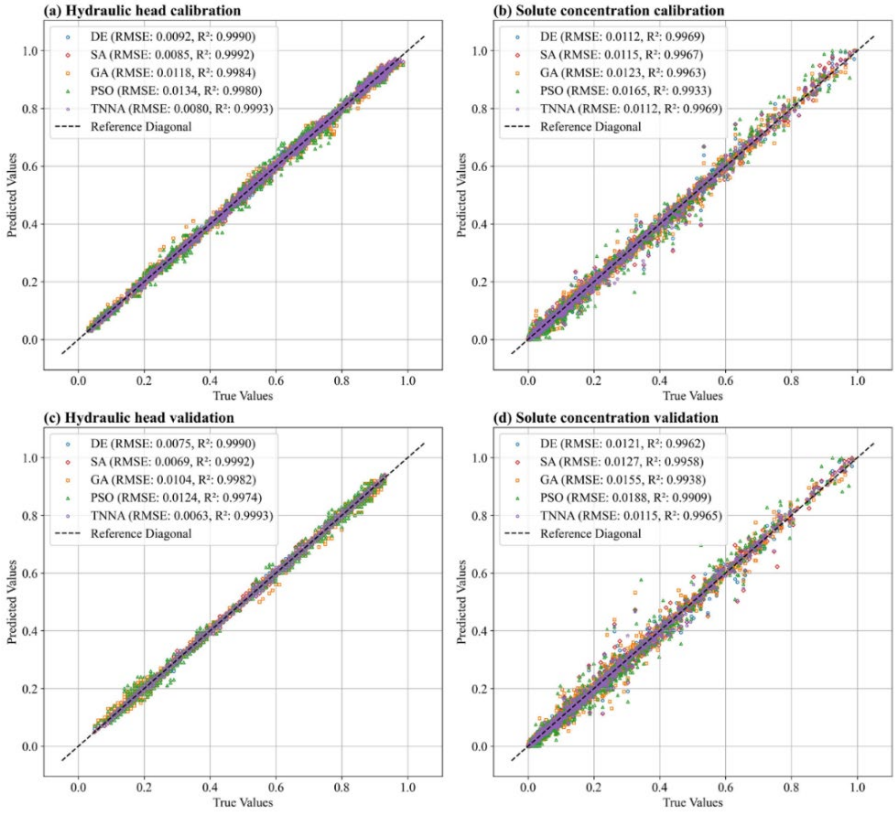


Figure 7. Comparison of calibrated and validated model predictive accuracy for hydraulic heads and solute concentrations by the four metaheuristic algorithms and the TNNA method.

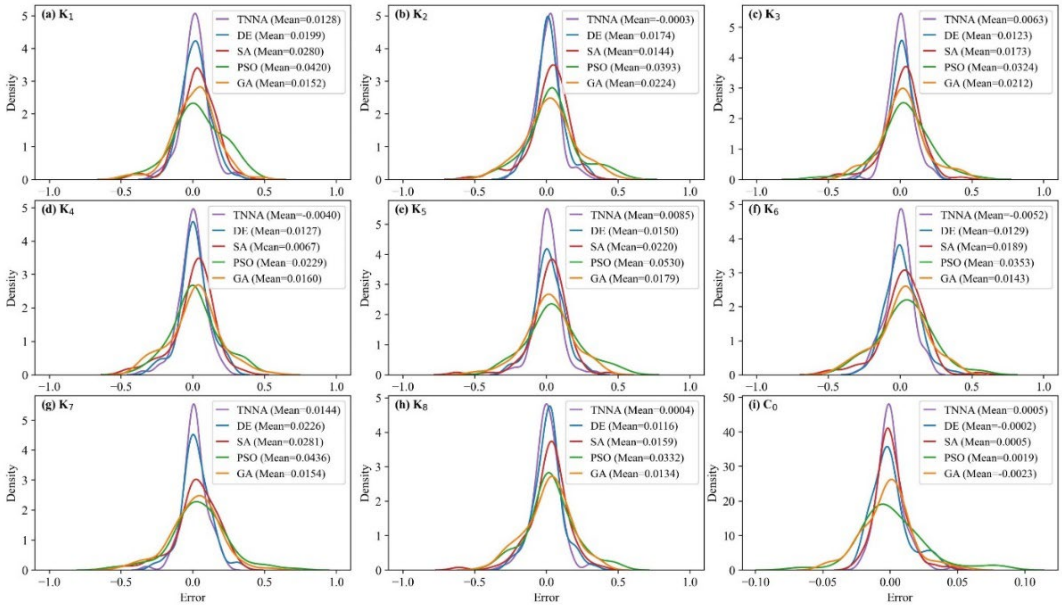


Figure 8. Probability density curves of estimation errors for nine model parameters using five optimization methods.

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024


Hydrology and Earth System Sciences
Discussions

The above comparison results indicated that the machine learning-based TNNA algorithm outperforms the other four metaheuristic algorithms in both inversion accuracy and computational efficiency. The primary advantage of the TNNA algorithm over the four metaheuristic algorithms is its highly deterministic updating direction of model parameters, guided by the loss function, which serves as the objective function for inverse modeling. Research on machine learning applications indicates that DNNs can approximate continuous functions by adjusting weights and biases (LeCun et al., 2015; Goodfellow et al., 2016). The TNNA algorithm leverages this capability by transforming the model parameter inversion issue into the training of a reverse network to achieve reverse mappings. By establishing a loss function based on inversion constraints from the Bayesian theorem, the TNNA algorithm ensures that training the reverse network brings each parameter update closer to the optimal solution during each epoch, thereby improving accuracy and convergence speed. In contrast, the four metaheuristic algorithms require numerous forward simulations for each parameter update. The optimization direction for model parameters is determined by evaluating the objective function. This process is governed by the exploration and exploitation strategies inherent in metaheuristic algorithms. However, these approaches introduce randomness in the direction of model parameter updates, making it challenging to ensure that updates move towards the direction of fastest convergence under specific hyperparameter settings. This also explains why the TNNA algorithm can update model parameters more efficiently and achieve higher convergence accuracy despite requiring only one forward realization in each training epoch.

4.2.2 Inversion results of the high-dimensional parameter scenario

For estimating the permeability field under five designed observational scenarios, the iteration number for the four metaheuristic algorithms was set at 200, with N_{PC} values of 100, 500, and 1000. The learning rate and weight decay for training reverse networks within the TNNA framework were set to 1×10^{-3} and 1×10^{-4}, respectively.

Figure 9 and Figure 10 illustrate the log-permeability field estimation results and error distributions for the four metaheuristic algorithms and the TNNA algorithm under the most densely observed scenario (i.e., Scenario 5). The corresponding results for Scenarios 1-4 are presented in Figure S7-S14. Figure 11 compares the RMSE values for the log-permeability fields estimated by the four metaheuristic algorithms and the TNNA algorithm across all five scenarios. These detailed RMSE values can be found in Table 2 (Scenario 5) and Table S4 (Scenarios 1-4). For Scenario 5, the accuracy of permeability estimations by each metaheuristic algorithm improves as the N_{PC} value increases (see Figure 9 and Table 2). Notably, the GA achieves the best results with an N_{PC} of 1000, recording an RMSE of 0.1057. The DE and SA algorithms yield their most accurate permeability estimations with RMSE values of 0.1597 (N_{PC}=100) and 0.1549 (N_{PC}=1000), respectively. The PSO method is the least effective, achieving an RMSE of 0.3334 at N_{PC} =1000. As shown in Figure 10 and Table 2, the TNNA algorithm provides inversion results with an RMSE of 0.1063 after training the reverse network for 200 epochs. This suggests that the TNNA algorithm can estimate high-dimensional permeability fields with accuracy comparable to that of the GA method (N_{PC}=1000) with significantly fewer forward model realizations (200 compared to 200,000), reducing the computational burden by 99.9% and improving inversion efficiency by a factor of 1000. Increasing the training epochs of the reverse network to 1000 further reduces the RMSE of the TNNA method to 0.0595, demonstrating its advantages over the

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Hydrology and Earth System Sciences
Discussions

four metaheuristic algorithms in this scenario. Across all scenarios, the accuracy of the estimated permeability fields correlates positively with the density of observation wells, and estimation errors are generally higher in areas not covered by monitoring wells (see Figure S7-S14). Figure 11 further demonstrates that the RMSE values for permeability estimation using the TNNA algorithm are consistently lower than those of the four metaheuristic algorithms across Scenarios 1-4, indicating that the TNNA algorithm exhibits greater robustness compared to the metaheuristic algorithms in all five scenarios.

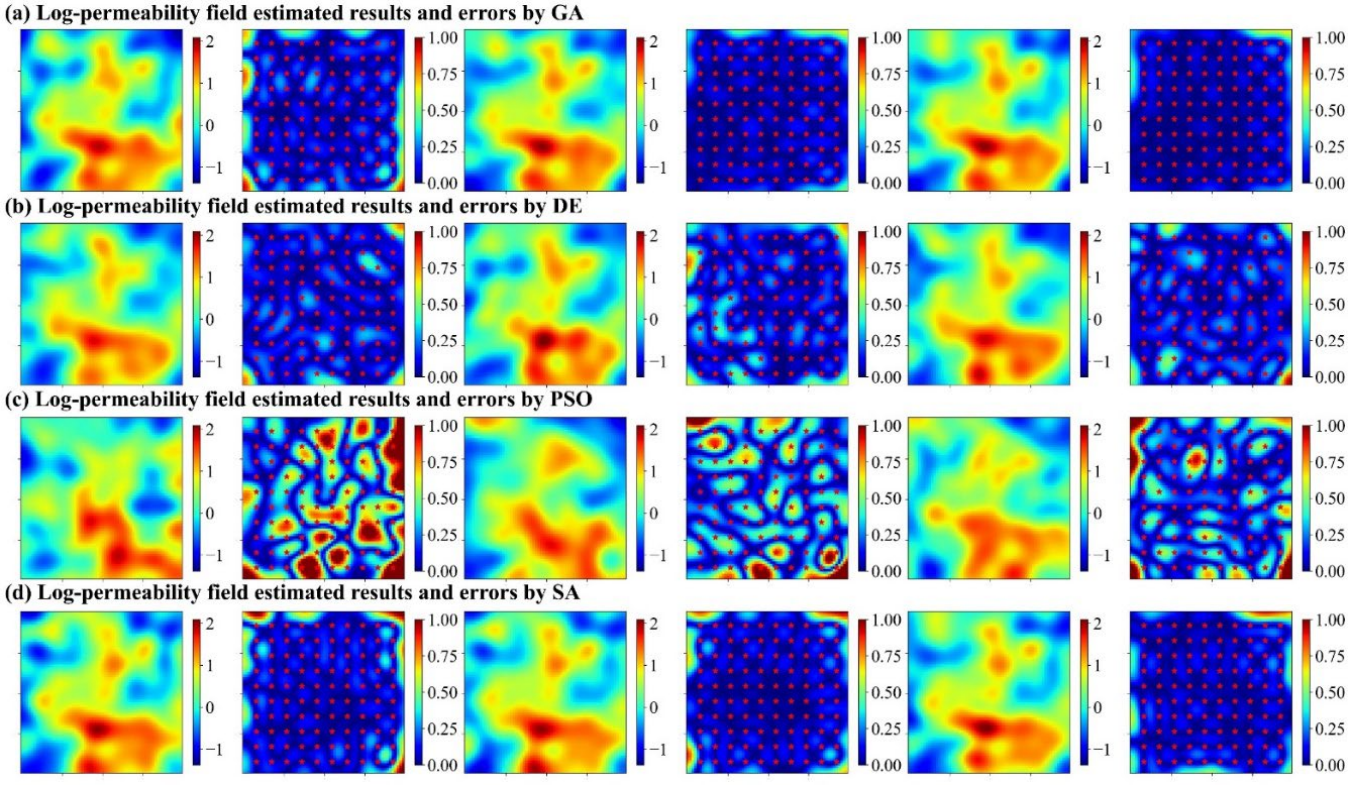


Figure 9. Spatial distributions of log-permeability field estimation results (row 1, 3, and 5 for N_{PC}=100, 500, and 1000, respectively) and absolute errors (row 2, 4, and 6 for N_{PC}=100, 500, and 1000, respectively) for Scenario 5, achieved by four metaheuristic algorithms.

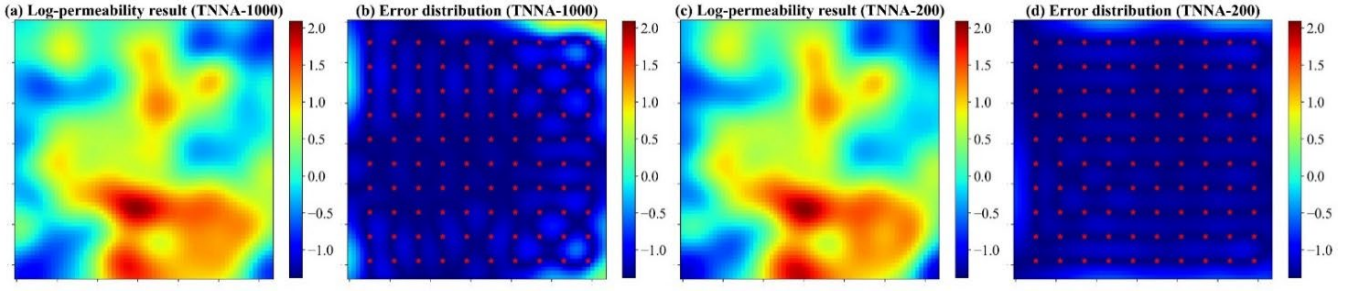


Figure 10. Spatial distributions log-permeability field estimation results and absolute errors for Scenario 5, achieved by the TNNA.

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Hydrology and Earth System Sciences
Discussions

Table 2. RMSE values of estimated log-permeability fields for the four metaheuristic algorithms and the TNNA algorithm under Scenario 5.

Metaheuristic algorithms					TNNA	
	GA	DE	PSO	SA		
N_{PC}=100	0.1940	0.1597	0.5399	0.2071	epoch=200	0.1063
N_{PC}=500	0.1142	0.1904	0.3810	0.1781	epoch=1000	0.0595
N_{PC}=1000	0.1057	0.1748	0.3334	0.1549		

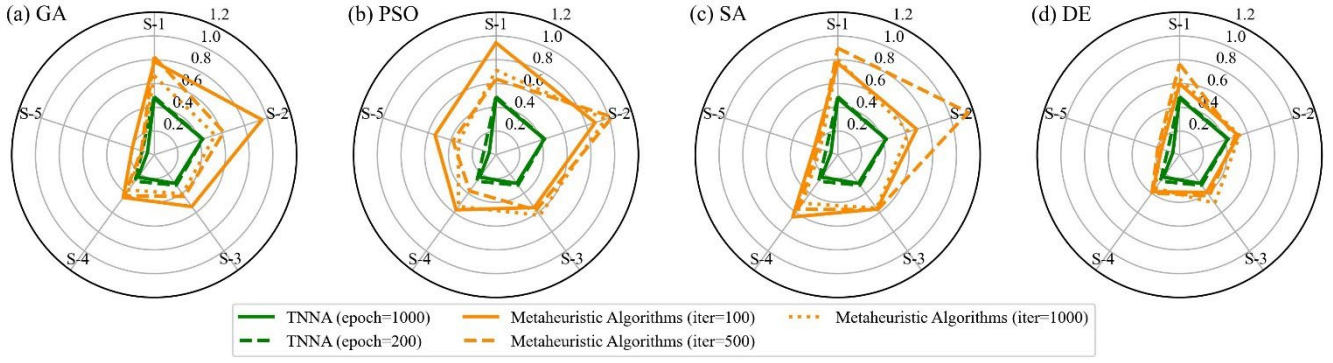


Figure 11. Comparison of RMSE in estimating log-permeability fields using four metaheuristic algorithms and the TNNA algorithm across five scenarios (S-1 to S-5).

To evaluate the predictive performance of the numerical model calibrated by various inversion methods, simulations of hydraulic heads and solute concentrations were conducted over 60 days, starting on the 2nd day with bi-daily recordings, using the permeability fields with the lowest RMSE values identified by each inversion method. Observation data from the 2nd day to the 40th day were used for model calibration, while additional data from the 42nd to the 60th day were employed to evaluate the future predictions of the calibrated numerical models. The RMSE values for the calibrated hydraulic heads and time series solute concentrations are presented in Table 3 and Figure 12. Figure 13 displays the spatial distribution of the calibrated numerical simulation results and errors for hydraulic heads and solute concentration simulation results at three specific times (t=4th, 20th, and 52nd days). Results for the entire 60-day period are presented in Figure S15-S44.

According to Figure 13(a), the calibrated simulation errors for hydraulic heads did not exceed 0.02 meters for the TNNA method and three of the four considered metaheuristic algorithms, except PSO method, which exhibited hydraulic head errors larger than 0.06 meters in certain areas. Among the four metaheuristic algorithms, the GA method achieved the lowest RMSE in hydraulic head simulations, with a value of 7.4837×10^{-4}. For solute concentrations, the GA algorithm consistently has the highest prediction accuracy among the metaheuristic algorithms, with RMSE values generally around 0.005 (Figure 12). The TNNA algorithm achieved a similar level of accuracy to GA in the calibrated numerical model predictions. Specifically, during the initial 10 days and from the 41st day to the 60th day, the TNNA algorithm showed slightly higher prediction accuracy than the GA-calibrated model. However, during the intermediate period from the 10th day to the 40th day, the GA-calibrated model had a slight advantage over the TNNA algorithm. The normalized absolute errors in the solute transport simulation results

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Hydrology and Earth System Sciences
Discussions

obtained using the TNNA algorithm remained consistently below 0.02 throughout the simulation period (Figure 13(b~c)). These results indicate that in high-dimensional settings, the TNNA algorithm provides inversion outcomes that enable the calibrated model to deliver simulation results comparable to those of the best-performing metaheuristic algorithm. Overall, the TNNA method also demonstrates advantages over the four metaheuristic optimization algorithms in the designed high-dimensional scenarios, excelling in both inversion efficiency and accuracy.

Table 3. RMSE values of calibrated hydraulic heads for the four metaheuristic algorithms and the TNNA algorithm.

	TNNA	**DE**	**GA**	**PSO**	**SA**
RMSE	6.8537×10^{-4}	1.2181×10^{-3}	7.4837×10^{-4}	2.1683×10^{-3}	1.0316×10^{-3}

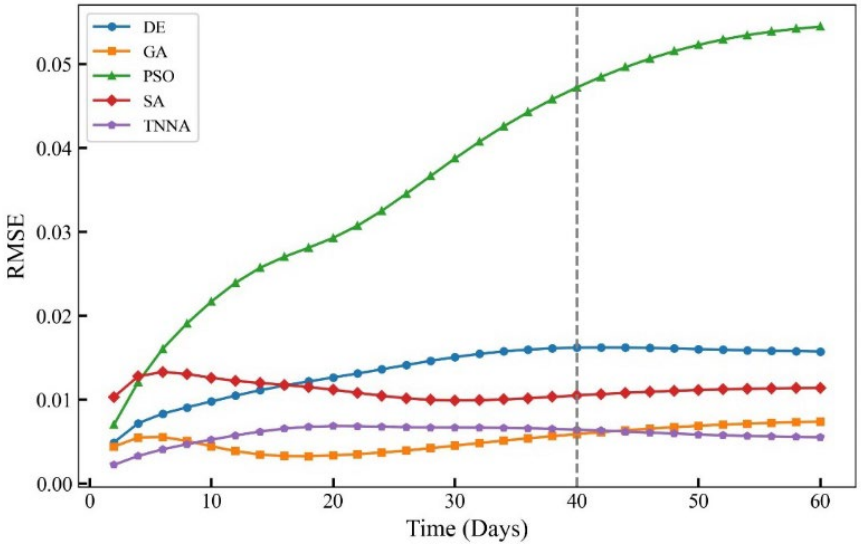


Figure 12. RMSE values of calibrated solute concentrations over 60 days for the four metaheuristic algorithms and the TNNA algorithm.

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

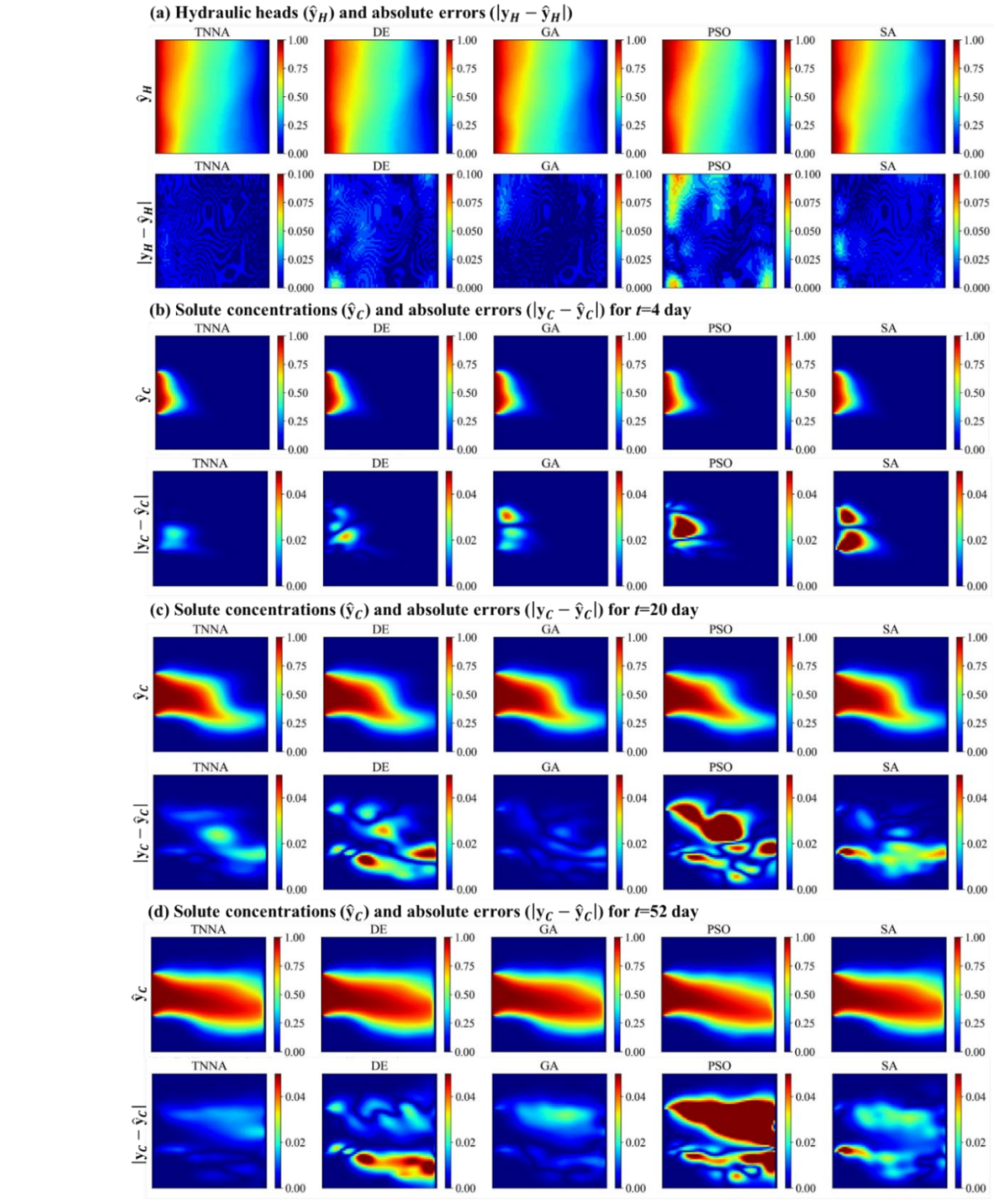


Figure 13. Spatial distributions of calibrated numerical simulation results and absolute errors for hydraulic heads and solute concentrations at three dynamic times (t=4, 20, and 50 day) using the TNNA algorithm and four metaheuristic algorithms.

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024


Hydrology and Earth System Sciences
Discussions

4.3 Insights from synthetic cases for practical research

The model conditions of the two synthetic cases in this study are primarily based on previous studies, as well as large-scale sandbox (Jose et al., 2004; J. Zhang et al., 2018; Mo et al., 2019). It is worth noting that the domain sizes for field-scale groundwater models are commonly on the kilometre scale, which is significantly larger than 10m or 20m sizes used in this study. Moreover, the hydraulic gradient of groundwater in natural alluvial aquifers is generally below 0.01, while hydraulic gradients of 0.1 and 0.05 may occur when the underlying aquitard of the aquifer has a natural slope (Chai et al., 2024). Additionally, the monitoring stations in the synthetic cases of this study are densely distributed, whereas in practical studies, the number and locations of monitoring stations are constrained by financial budgets. They are often designed based on multi-objective optimization criteria, such as maximizing information and minimizing redundancy (Keum et al., 2018; J. Chen et al., 2022; Cao et al., 2025). Therefore, a common concern that may arise is whether the results of this study can guide the selection of appropriate algorithms in practical research scenarios.

As summarized in the Introduction, the performance of an inversion algorithm depends on the degree of model nonlinearity and the complexity of parameter space. The mathematical model of this study consists of the groundwater flow continuity equation and the advection-dispersion equation. Within a fixed system mathematical model, variations in the average hydraulic gradient by an order of magnitude primarily affect flow velocity and solute transport rates, with little impact on the degree of nonlinearity in the system. Therefore, the inversion algorithms are expected to perform similarly in these real-world scenarios, provided that the system variables at field sites adhere to the same mathematical models and exhibit comparable parameter heterogeneity to that in this study. Additionally, monitoring strategies with densely distributed stations are commonly employed in the evaluation of inversion algorithms to ensure sufficient observational information, thereby reducing non-uniqueness in parameter inversion results (Bao et al., 2020; Mo et al., 2020; J. Zhang et al., 2024). Although such monitoring networks are unlikely to be used in field-scale groundwater studies, the effectiveness of the inversion results ultimately depends on the amount of effective information in the observational data. Furthermore, as optimized monitoring networks exclude redundant monitoring stations, dynamic responses from optimal monitoring stations typically exhibit higher sensitivity to model parameters, making it easier to achieve data assimilation goals for inversion algorithms. As a whole, the findings of this study are also applicable to research conducted at field scales with similar mathematical model conditions.

5. Summary and conclusions

Recent advancements in machine learning have significantly contributed to the development of inverse modeling. This study aims to compare the universality and advantages of the novel TNNA algorithm with four popular metaheuristic algorithms (GA, PSO, DE, and SA) across various parameter dimensions in solute transport models. Surrogate models for these inversion methods were constructed using ResNet, which achieved the highest predictive accuracy for hydraulic heads and solute concentrations among four evaluated surrogate modeling methods (MSVR, FC-DNN, LeNet, and ResNet).

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024


Hydrology and Earth System Sciences
Discussions

The inversion results indicate that the TNNA algorithm outperforms the four conventional metaheuristic algorithms in both high-dimensional and low-dimensional scenarios, providing more accurate results while significantly reduces the computational burden. Moreover, it has been verified that the TNNA algorithm consistently delivers reliable inversion results with just a single forward simulation per iteration in scenarios featuring various complex and uncertain model parameters. This characteristic offers a practical approach to balancing exploration and exploitation with a reduced computational burden, contrasting with conventional metaheuristic algorithms that require increasing forward simulations as the inversion problem grows more complex.

This study demonstrates that achievements in machine learning can significantly enhance inversion results compared to conventional methods. Given that nonlinearity and ill-posedness are two common challenges in inversion problems across various disciplines, establishing constraints on nonlinear relationships and applying appropriate machine learning techniques can be treated as vital approaches for future research. Meanwhile, the equifinality induced by ill-posedness can be attenuated through monitoring network optimizations. The model processes in this study are clearly defined, but in real-world scenarios, model response measurements often involve significant uncertainty, especially in groundwater systems with multi-component reactive transport. In such scenarios, it is crucial to consider the inversion of model parameters and integrate the identification of key processes within the groundwater system models. These optimization challenges may also benefit from the capabilities of machine learning, since they often involve uncertain quantifications. Furthermore, it is essential to continuously follow the latest developments of machine learning and consider integrating more advanced DNN models to address increasingly complex groundwater system inversion problems. For example, the emergence of large language models offers opportunities for complex system modeling and inversion studies across various scientific and engineering disciplines (Birhane et al., 2023; Buehler, 2023). The latent capacity of large language models has yet to be fully explored. Significant achievements have primarily focus on protein designs (Jumper et al., 2021; Ferruz et al., 2022; Lin et al., 2023), drug development (Peng et al., 2023; Duffy et al., 2024), and molecular discovery (Flam-Shepherd et al., 2022; J. Li et al., 2023). Developing groundwater system inversion frameworks based on large language models is of great significance for the advancement of hydrology and earth science (Deng et al., 2023; Foroumandi et al., 2023).

Competing interests

The contact author has declared that none of the authors has any competing interests.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities (XJ2023005201), the National Natural Science Foundation of China (NSFC: 42402241, U2267217, 42141011, and 42002254).

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Hydrology and Earth System Sciences
Open Access
Discussions

Data Availability Statement

The data and codes for four surrogate models and five optimization algorithms are available on: https://doi.org/10.5281/zenodo.10499582

References

Abbas, S. A., Bailey, R. T., White, J. T., Arnold, J. G., White, M. J., Čerkasova, N., and Gao, J. (2024), A framework for parameter estimation, sensitivity analysis, and uncertainty analysis for holistic hydrologic modeling using SWAT+, *Hydrology and Earth System Sciences*, *28*(1), 21-48. https://10.5194/hess-28-21-2024

Adler, J., and Öktem, O. (2017), Solving ill-posed inverse problems using iterative deep neural networks, *Inverse Problems*, *33*(12). https://10.1088/1361-6420/aa9581

Arsenault, R., and Brissette, F. P. (2014), Continuous streamflow prediction in ungauged basins: The effects of equifinality and parameter set selection on uncertainty in regionalization approaches, *Water Resources Research*, *50*(7), 6135-6153. https://10.1002/2013wr014898

Bandai, T., and Ghezzehei, T. A. (2022), Forward and inverse modeling of water flow in unsaturated soils with discontinuous hydraulic conductivities using physics-informed neural networks with domain decomposition, *Hydrology and Earth System Sciences*, *26*(16), 4469-4495. https://10.5194/hess-26-4469-2022

Bao, J., Li, L., and Redoloza, F. (2020), Coupling ensemble smoother and deep learning with generative adversarial networks to deal with non-Gaussianity in flow and transport data assimilation, *Journal of Hydrology*, *590*. https://10.1016/j.jhydrol.2020.125443

Bentivoglio, R., Isufi, E., Jonkman, S. N., and Taormina, R. (2022), Deep learning methods for flood mapping: a review of existing applications and future research directions, *Hydrol. Earth Syst. Sci.*, *26*(16), 4345-4378. https://10.5194/hess-26-4345-2022

Birhane, A., Kasirzadeh, A., Leslie, D., and Wachter, S. (2023), Science in the age of large language models, *Nature Reviews Physics*, *5*(5), 277-280. https://10.1038/s42254-023-00581-4

Buehler, M. J. (2023), MeLM, a generative pretrained language modeling framework that solves forward and inverse mechanics problems, *Journal of the Mechanics and Physics of Solids*, *181*. https://10.1016/j.jmps.2023.105454

Cao, M., Dai, Z., Chen, J., Yin, H., Zhang, X., Wu, J., Thanh, H. V., and Soltanian, M. R. (2025), An integrated framework of deep learning and entropy theory for enhanced high-dimensional permeability field identification in heterogeneous aquifers, *Water Research*, *268*, 122706. https://https://doi.org/10.1016/j.watres.2024.122706

Castaings, W., Dartus, D., Le Dimet, F. X., and Saulnier, G. M. (2009), Sensitivity analysis and parameter estimation for distributed hydrological modeling: potential of variational methods, *Hydrol. Earth Syst. Sci.*, *13*(4), 503-517. https://10.5194/hess-13-503-2009

Chai, J., Zhang, W., Zhao, K., Li, S., Baloch, M. Y. J., Wang, Z., Zhang, D., and Yang, Y. (2024), Multi-biological risk in groundwater-surface water system under landfill stress: Driven by bacterial size and biological toxicity, *Journal of Hydrology*, *636*. https://10.1016/j.jhydrol.2024.131282

Chang, Z., Lu, W., and Wang, Z. (2022), Study on source identification and source-sink relationship of LNAPLs pollution in groundwater by the adaptive cyclic improved iterative process and Monte Carlo stochastic simulation, *Journal of Hydrology*, *612*. https://10.1016/j.jhydrol.2022.128109

Chen, J., Dai, Z., Yang, Z., Pan, Y., Zhang, X., Wu, J., and Reza Soltanian, M. (2021), An improved tandem neural network architecture for inverse modeling of multicomponent reactive transport in porous media, *Water Resources Research*, *57*(12), 2021WR030595. https://10.1029/2021wr030595

Chen, J., Dai, Z., Dong, S., Zhang, X., Sun, G., Wu, J., Ershadnia, R., Yin, S., and Soltanian, M. R. (2022), Integration of deep learning and information theory for designing monitoring networks in heterogeneous aquifer systems, *Water Resources Research*, *58*(10), 2022WR032429. https://10.1029/2022wr032429

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Chen, X., Hammond, G. E., Murray, C. J., Rockhold, M. L., Vermeul, V. R., and Zachara, J. M. (2013), Application of ensemble-based data assimilation techniques for aquifer characterization using tracer data at Hanford 300 area, *Water Resources Research*, *49*(10), 7064-7076. https://10.1002/2012wr013285

Dai, Z., and Samper, J. (2004), Inverse problem of multicomponent reactive chemical transport in porous media: Formulation and applications, *Water Resources Research*, *40*(7), W07407. https://10.1029/2004wr003248

Deng, C., Zhang, T., He, Z., Xu, Y., Chen, Q., Shi, Y., Fu, L., Zhang, W., Wang, X., Zhou, C., et al. (2023), K2: A Foundation Language Model for Geoscience Knowledge Understanding and Utilization, *arXiv:2306.05064v2*.

Dragonetti, G., Comegna, A., Ajeel, A., Deidda, G. P., Lamaddalena, N., Rodriguez, G., Vignoli, G., and Coppola, A. (2018), Calibrating electromagnetic induction conductivities with time-domain reflectometry measurements, *Hydrol. Earth Syst. Sci.*, *22*(2), 1509-1523. https://10.5194/hess-22-1509-2018

Duffy, Á., Petrazzini, B. O., Stein, D., Park, J. K., Forrest, I. S., Gibson, K., Vy, H. M., Chen, R., Márquez-Luna, C., Mort, M., et al. (2024), Development of a human genetics-guided priority score for 19,365 genes and 399 drug indications, *Nature Genetics*. https://10.1038/s41588-023-01609-2

Eberhart, R., and Kennedy, J. (1995), Particle swarm optimization, paper presented at Proceedings of the IEEE international conference on neural networks, Citeseer.

Ferruz, N., Schmidt, S., and Hocker, B. (2022), ProtGPT2 is a deep unsupervised language model for protein design, *Nature Communications*, *13*(1), 4348. https://10.1038/s41467-022-32007-7

Flam-Shepherd, D., Zhu, K., and Aspuru-Guzik, A. (2022), Language models can learn complex molecular distributions, *Nature Communications*, *13*(1). https://10.1038/s41467-022-30839-x

Foroumandi, E., Moradkhani, H., Sanchez-Vila, X., Singha, K., Castelletti, A., and Destouni, G. (2023), ChatGPT in Hydrology and Earth Sciences: Opportunities, Prospects, and Concerns, *Water Resources Research*, *59*(10). https://10.1029/2023wr036288

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016), *Deep learning*, MIT press Cambridge.

He, K., Zhang, X., Ren, S., and Sun, J. (2016), Deep residual learning for image recognition, paper presented at Proceedings of the IEEE conference on computer vision and pattern recognition.

Hinton, G. E., and Salakhutdinov, R. R. (2006), Reducing the dimensionality of data with neural networks, *Science*, *313*(5786), 504-507.

Holland John, H. (1975), *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor.

Ines, A. V. M., and Droogers, P. (2002), Inverse modelling in estimating soil hydraulic functions: a Genetic Algorithm approach, *Hydrology and Earth System Sciences*, *6*(1), 49-66. https://10.5194/hess-6-49-2002

Jardani, A., Vu, T. M., and Fischer, P. (2022), Use of convolutional neural networks with encoder-decoder structure for predicting the inverse operator in hydraulic tomography, *Journal of Hydrology*, *604*. https://10.1016/j.jhydrol.2021.127233

Jaumann, S., and Roth, K. (2018), Soil hydraulic material properties and layered architecture from time-lapse GPR, *Hydrology and Earth System Sciences*, *22*(4), 2551-2573. https://10.5194/hess-22-2551-2018

Jose, S. C., Rahman, M. A., and Cirpka, O. A. (2004), Large-scale sandbox experiment on longitudinal effective dispersion in heterogeneous porous media, *Water Resources Research*, *40*(12). https://10.1029/2004wr003363

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021), Highly accurate protein structure prediction with AlphaFold, *Nature*, *596*(7873), 583-589. https://10.1038/s41586-021-03819-2

Keum, J., Coulibaly, P., Razavi, T., Tapsoba, D., Gobena, A., Weber, F., and Pietroniro, A. (2018), Application of SNODAS and hydrologic models to enhance entropy-based snow monitoring network design, *Journal of Hydrology*, *561*, 688-701. https://10.1016/j.jhydrol.2018.04.037

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983), Optimization by Simulated Annealing, *Science*, *220*(4598), 671-680. https://doi:10.1126/science.220.4598.671

Kuang, W., Yuan, C., and Zhang, J. (2021), Real-time determination of earthquake focal mechanism via deep learning, *Nature Communications*, *12*(1), 1432. https://10.1038/s41467-021-21670-x

LeCun, Y., Bengio, Y., and Hinton, G. (2015), Deep learning, *Nature*, *521*(7553), 436-444. https://10.1038/nature14539

Li, E. (2019), An adaptive surrogate assisted differential evolutionary algorithm for high dimensional constrained problems, *Applied Soft Computing*, *85*. https://10.1016/j.asoc.2019.105752

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Li, J., Liu, Y., Fan, W., Wei, X.-Y., Liu, H., Tang, J., and Li, Q. (2023), Empowering Molecule Discovery for Molecule-Caption Translation with Large Language Models: A ChatGPT Perspective, *arXiv preprint arXiv:.06615*. https://arxiv.org/abs/2306.05064

Lin, P., Tao, H., Li, H., and Huang, S.-Y. (2023), Protein–protein contact prediction by geometric triangle-aware protein language models, *Nature Machine Intelligence*, *5*(11), 1275-1284. https://10.1038/s42256-023-00741-2

Lindsay, A., McCloskey, J., and Bhloscaidh, M. N. (2016), Using a genetic algorithm to estimate the details of earthquake slip distributions from point surface displacements, *Journal of Geophysical Research-Solid Earth*, *121*(3), 1796-1820. https://10.1002/2015jb012181

Liu, D., Tan, Y., Khoram, E., and Yu, Z. (2018), Training deep neural networks for the inverse design of nanophotonic structures, *ACS Photonics*, *5*(4), 1365-1369. https://10.1021/acsphotonics.7b01377

Liu, M., Ahmad, R., Cai, W., and Mukerji, T. (2023), Hierarchical Homogenization With Deep-Learning-Based Surrogate Model for Rapid Estimation of Effective Permeability From Digital Rocks, *Journal of Geophysical Research: Solid Earth*, *128*(2). https://10.1029/2022jb025378

Long, Y., Ren, J., Li, Y., and Chen, H. (2019), Inverse design of photonic topological state via machine learning, *Applied Physics Letters*, *114*(18), 181105. https://10.1063/1.5094838

Luo, J., Li, X., Xiong, Y., and Liu, Y. (2023a), Groundwater pollution source identification using Metropolis-Hasting algorithm combined with Kalman filter algorithm, *Journal of Hydrology*, *626*. https://10.1016/j.jhydrol.2023.130258

Luo, J., Ma, X., Ji, Y., Li, X., Song, Z., and Lu, W. (2023b), Review of machine learning-based surrogate models of groundwater contaminant modeling, *Environmental Research*, *238*(Part 2), 117268. https://10.1016/j.envres.2023.117268

Lykkegaard, M. B., Dodwell, T. J., and Moxey, D. (2021), Accelerating uncertainty quantification of groundwater flow modelling using a deep neural network proxy, *Computer Methods in Applied Mechanics and Engineering*, *383*. https://10.1016/j.cma.2021.113895

Ma, J., Xia, D., Guo, H., Wang, Y., Niu, X., Liu, Z., and Jiang, S. (2022), Metaheuristic-based support vector regression for landslide displacement prediction: a comparative study, *Landslides*, *19*(10), 2489-2511. https://10.1007/s10346-022-01923-6

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), Equation of state calculations by fast computing machines, *The Journal of Chemical Physics*, *21*(6), 1087-1092.

Mo, S., Zabaras, N., Shi, X., and Wu, J. (2019), Deep autoregressive neural networks for high-dimensional inverse problems in groundwater contaminant source identification, *Water Resources Research*, *55*, 3856-3881. https://10.1029/2018wr024638

Mo, S., Zabaras, N., Shi, X., and Wu, J. (2020), Integration of adversarial autoencoders with residual dense convolutional networks for estimation of non-Gaussian hydraulic conductivities, *Water Resources Research*, *56*(2), 2019WR026082.

Nhu, V. H. (2022), Levenberg-Marquardt method for ill-posed inverse problems with possibly non-smooth forward mappings between Banach spaces, *Inverse Problems*, *38*(1). https://ARTN 015007

10.1088/1361-6420/ac38b7

Peng, C., Yang, X., Chen, A., Smith, K. E., PourNejatian, N., Costa, A. B., Martin, C., Flores, M. G., Zhang, Y., Magoc, T., et al. (2023), A study of generative large language model for medical research and healthcare, *npj Digital Medicine*, *6*(1), 210. https://10.1038/s41746-023-00958-w

Pérez-Cruz, F., Camps-Valls, G., Soria-Olivas, E., Pérez-Ruixo, J. J., Figueiras-Vidal, A. R., and Artés-Rodríguez, A. (2002), Multi-dimensional Function Approximation and Regression Estimation, paper presented at Artificial Neural Networks — ICANN 2002.

Qin, Y., Kavetski, D., Kuczera, G., McInerney, D., Yang, T., and Guo, Y. (2022), Can Gauss-Newton Algorithms Outperform Stochastic Optimization Algorithms When Calibrating a Highly Parameterized Hydrological Model? A Case Study Using SWAT, *Water Resources Research*, *58*(11). https://10.1029/2021wr031532

Rafiei, V., Nejadhashemi, A. P., Mushtaq, S., Bailey, R. T., and An-Vo, D.-A. (2022), An improved calibration technique to address high dimensionality and non-linearity in integrated groundwater and surface water models, *Environmental Modelling & Software*, *149*. https://10.1016/j.envsoft.2022.105312

https://doi.org/10.5194/hess-2024-315
Preprint. Discussion started: 6 December 2024

Razavi, S., Tolson, B. A., and Burn, D. H. (2012), Review of surrogate modeling in water resources, *Water Resources Research*, *48*(7). https://10.1029/2011wr011527

Sanchez-Fernandez, M., de-Prado-Cumplido, M., Arenas-Garcia, J., and Perez-Cruz, F. (2004), SVM multiregression for nonlinear channel estimation in multiple-input multiple-output systems, *IEEE Transactions on Signal Processing*, *52*(8), 2298-2307. https://10.1109/tsp.2004.831028

Sanchez-Vila, X., Donado, L. D., Guadagnini, A., and Carrera, J. (2010), A solution for multicomponent reactive transport under equilibrium and kinetic reactions, *Water Resources Research*, *46*(7). https://10.1029/2009wr008439

Scharnagl, B., Vrugt, J. A., Vereecken, H., and Herbst, M. (2011), Inverse modelling of in situ soil water dynamics: investigating the effect of different prior distributions of the soil hydraulic parameters, *Hydrology and Earth System Sciences*, *15*(10), 3043-3059. https://10.5194/hess-15-3043-2011

Schneider-Zapp, K., Ippisch, O., and Roth, K. (2010), Numerical study of the evaporation process and parameter estimation analysis of an evaporation experiment, *Hydrology and Earth System Sciences*, *14*(5), 765-781. https://10.5194/hess-14-765-2010

Shen, C., Appling, A. P., Gentine, P., Bandai, T., Gupta, H., Tartakovsky, A., Baity-Jesi, M., Fenicia, F., Kifer, D., Li, L., et al. (2023), Differentiable modelling to unify machine learning and physical models for geosciences, *Nature Reviews Earth & Environment*, *4*(8), 552-567. https://10.1038/s43017-023-00450-9

Steefel, C., Depaolo, D., and Lichtner, P. (2005), Reactive transport modeling: An essential tool and a new research approach for the earth sciences, *Earth and Planetary Science Letters*, *240*(3-4), 539-558. https://10.1016/j.epsl.2005.09.017

Sternagel, A., Loritz, R., Klaus, J., Berkowitz, B., and Zehe, E. (2021), Simulation of reactive solute transport in the critical zone: a Lagrangian model for transient flow and preferential transport, *Hydrology and Earth System Sciences*, *25*(3), 1483-1508. https://10.5194/hess-25-1483-2021

Storn, R., and Price, K. (1997), Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, *Journal of Global Optimization*, *11*(4), 341-359. https://10.1023/A:1008202821328

Sun, A. Y. (2018), Discovering state-parameter mappings in subsurface models using generative adversarial networks, *Geophysical Research Letters*, *45*(20), 11137-11146. https://10.1029/2018gl080404

Sun, N. (2013), *Inverse problems in groundwater modeling*, Springer Science & Business Media.

Tran, H.-N., Phan, G. T. T., Do, Q. B., and Tran, V.-P. (2022), Comparative evaluation of the performance of improved genetic algorithms and differential evolution for in-core fuel management of a research reactor, *Nuclear Engineering and Design*, *398*, 111953. https://https://doi.org/10.1016/j.nucengdes.2022.111953

Travaš, V., Zaharija, L., Stipanić, D., and Družeta, S. (2023), Estimation of hydraulic conductivity functions in karst regions by particle swarm optimization with application to Lake Vrana, Croatia, *Hydrology and Earth System Sciences*, *27*(6), 1343-1359. https://10.5194/hess-27-1343-2023

Tsai, F. T. C., Sun, N., and Yeh, W. W. G. (2003), Global-local optimization for parameter structure identification in three-dimensional groundwater modeling, *Water Resources Research*, *39*(2). https://10.1029/2001wr001135

Tuia, D., Verrelst, J., Alonso, L., Perez-Cruz, F., and Camps-Valls, G. (2011), Multioutput support vector regression for remote sensing biophysical parameter estimation, *IEEE Geoscience and Remote Sensing Letters*, *8*(4), 804-808. https://10.1109/lgrs.2011.2109934

Vrugt, J. A. (2016), Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB implementation, *Environmental Modelling & Software*, *75*, 273-316. https://10.1016/j.envsoft.2015.08.013

Wang, G. S., and Chen, S. L. (2013), Evaluation of a soil greenhouse gas emission model based on Bayesian inference and MCMC: Parameter identifiability and equifinality, *Ecological Modelling*, *253*, 107-116. https://10.1016/j.ecolmodel.2012.09.011

Wang, N., Chang, H., and Zhang, D. (2021), Deep-learning-based inverse modeling approaches: A subsurface flow example, *Journal of Geophysical Research: Solid Earth*, *126*(2), 2020JB020549. https://10.1029/2020jb020549

Wang, Y., Fang, Z., and Hong, H. (2019), Comparison of convolutional neural networks for landslide susceptibility mapping in Yanshan County, China, *Science of The Total Environment*, *666*, 975-993. https://10.1016/j.scitotenv.2019.02.263

Xia, C.-A., Luo, X., Hu, B. X., Riva, M., and Guadagnini, A. (2021), Data assimilation with multiple types of observation boreholes via the ensemble Kalman filter embedded within stochastic moment equations, *Hydrology and Earth System Sciences*, *25*(4), 1689-1709. https://10.5194/hess-25-1689-2021

Xu, Z. X., Serata, R., Wainwright, H., Denham, M., Molins, S., Gonzalez-Raymat, H., Lipnikov, K., Moulton, J. D., and Eddy-Dilek, C. (2022), Reactive transport modeling for supporting climate resilience at groundwater contamination sites, *Hydrology and Earth System Sciences*, *26*(3), 755-773. https://10.5194/hess-26-755-2022

Yan, Z., Ran, J., Xiao, Y., Xu, Z., Wu, H., Deng, X. L., Du, L., and Zhong, M. (2023), The Temporal Improvement of Earth's Mass Transport Estimated by Coupling GRACE-FO With a Chinese Polar Gravity Satellite Mission, *Journal of Geophysical Research: Solid Earth*, *128*(9). https://10.1029/2023jb027157

Yang, X., Chen, X., and Smith, M. M. (2022), Deep learning inversion of gravity data for detection of CO_2 plumes in overlying aquifers, *Journal of Applied Geophysics*, *196*, 104507. https://10.1016/j.jappgeo.2021.104507

Yeung, C., Tsai, J.-M., King, B., Pham, B., Ho, D., Liang, J., Knight, M. W., and Raman, A. P. (2021), Multiplexed supercell metasurface design and optimization with tandem residual networks, *Nanophotonics*, *10*(3), 1133-1143. https://10.1515/nanoph-2020-0549

Yu, S., and Ma, J. (2021), Deep Learning for Geophysics: Current and Future Trends, *Reviews of Geophysics*, *59*(3). https://10.1029/2021rg000742

Zhan, C., Dai, Z., Soltanian, M. R., and Zhang, X. (2021), Stage-wise stochastic deep learning inversion framework for subsurface sedimentary structure identification, *Geophysical Research Letters*, *49*(1), 2021GL095823. https://10.1029/2021gl095823

Zhan, C., Dai, Z., Yang, Z., Zhang, X., Ma, Z., Thanh, H. V., and Soltanian, M. R. (2023), Subsurface sedimentary structure identification using deep learning: A review, *Earth-Science Reviews*, *239*, 104370. https://10.1016/j.earscirev.2023.104370

Zhang, D. X., and Lu, Z. M. (2004), An efficient, high-order perturbation approach for flow in random porous media via Karhunen-Loeve and polynomial expansions, *Journal of Computational Physics*, *194*(2), 773-794. https://10.1016/j.jcp.2003.09.015

Zhang, J., Lin, G., Li, W., Wu, L., and Zeng, L. (2018), An iterative local updating ensemble smoother for estimation and uncertainty assessment of hydrologic model parameters with multimodal distributions, *Water Resources Research*, *54*(3), 1716-1733. https://10.1002/2017wr020906

Zhang, J., Cao, C., Nan, T., Ju, L., Zhou, H., and Zeng, L. (2024), A Novel Deep Learning Approach for Data Assimilation of Complex Hydrological Systems, *Water Resources Research*, *60*(2), e2023WR035389. https://https://doi.org/10.1029/2023WR035389

Zhou, H., Gómez-Hernández, J. J., and Li, L. (2014), Inverse methods in hydrogeology: Evolution and recent trends, *Advances in Water Resources*, *63*, 22-37. https://10.1016/j.advwatres.2013.10.014