© <2020>. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/
The definitive publisher version is available online at https://doi.org/ 10.1016/j.aei.2020.101097

Marked version Click here to view linked References

Marked Version

Reinforcement Learning based Optimizer for Improvement of Predicting Tunneling-induced ~~Settlement~~ Ground Responses~~Prediction Model~~

Pin Zhang[1], Heng Li[2], Q. P. Ha[3], Zhen-Yu Yin[1], Ren-Peng Chen[4, 5*]

1. Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China
2. Department of Building and Real Estate, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China
3. Faculty of Engineering and IT, University of Technology Sydney, NSW 2007, Australia
4. College of Civil Engineering, Hunan University, Changsha 410082, China
5. MOE Key Laboratory of Building Safety and Energy Efficiency, Changsha 410082, China

Corresponding author: Ren-Peng Chen; Email: chenrp@hnu.edu.cn

Abstract: Prediction of ground responses is important for improving performance of tunneling. This study proposes a novel reinforcement learning (RL) based optimizer with the integration of deep-Q network (DQN) and particle swarm optimization (PSO)~~ to~~. Such optimizer is used to improve the extreme learning machine (ELM) based tunneling-induced settlement prediction model. Herein, DQN-PSO optimizer is used to optimize the weights and biases of ELM. Based on the prescribed states, actions, rewards, rules and objective functions, DQN-PSO optimizer evaluates the rewards of actions at each step, thereby guides particles which action should be conducted and when should take this action. Such hybrid model is ~~adopted on~~applied in a practical tunnel project. Regarding the search of global best weights and biases of ELM, ~~The~~ the results indicate the DQN-PSO optimizer obviously outperforms conventional metaheuristic optimization algorithms with higher accuracy and lower computational cost~~ to identify the global best~~

~~weights and biases of ELM~~. Meanwhile, this model can identify relationships among influential factors and ground responses through self-practicing~~ and accurately predict tunneling-induced ground response in real time~~. The ultimate model can be expressed with an explicit formulation and used to predict tunneling-induced ground response in real time, facilitating its application in engineering practice.

Keywords: Tunnel; Ground response; Reinforcement learning; Extreme learning machine; Optimization

1. Introduction

Ground responses to shield machine tunneling is a sophisticated problem that is affected by tunnel geometry, shield machine operational parameters, geological conditions and anomalous conditions [1]. The development of a rigorous analytical solution for descripting tunneling-induced ground response is complicated, because tunneling process involves multi-disciplinary knowledge such as solid mechanics, fluid mechanics, thermodynamics and tribology. ~~Early~~The initial analytical models were developed upon the homogenous elastic half-space theory [2, 3], in which soils are treated as an isotropic elastic material with a single layer. To consider the tunneling-induced plastic deformation, classical plasticity solutions for soil stresses and displacements [4] were obtained by assuming a cavity contraction in a linearly-elastic, plastic material with Mohr-Coulomb yielding and nonassociative flow, but this method is merely appropriate in the case that plastic zone does not extend to the ground surface [5]. Few analytical models can consider the effect of fluid-solid coupling on the ground responses, although tunneling process can cause remarkable change in flow regime and cause large ground subsidence. Kinematical effects of ~~shield machine advance~~tunneling on the ground response have also be frequently reported [6, 7], but these phenomenological observations merely stated the effect of ground responses to these kinematical parameters~~ and~~. It means that an explicit model involving these parameters ~~have not been~~ cannot be developed.

Existing analytical solutions merely account for limited influential factors and simulate ground responses in a simplistic manner [8]. Engineers prefer to apply empirical formulations, which were derived from numerous in-situ observations, to predict ground response due to their simplicity [9].~~,~~ However,~~but~~ such phenomenological methods tend to be applicable to a specific engineering, because the influential factors such as soil types, construction methods and tunnel configuration are different for different tunneling projects. Numerical modelling methods such as finite element and discrete element have been extensively employed to investigate ground response to tunneling as the improvement in the software and hardware [10-12]~~, because~~. Such elaborate numerical models are able to simulate soil-shield machine interaction by considering numerous extrinsic and intrinsic factors such as the geological heterogeneity [12] and the shield machine operation [13]. Nevertheless, parameters of soil constitutive models need to be calibrated by numerous experimental tests and back analysis of parameters also requires considerable skills [14]. A problem that all engineers have to confront is how to timely predict ground responses to tunneling and mitigates potential risks. ~~Empirical, analytical and numerical methods obviously exist their own deficiency in capturing the ground responses to tunneling in real time, because~~ Considering the influential factors such as operational parameters and geological conditions vary frequently with the advance of shield machine, empirical, analytical and numerical methods obviously exist their own deficiency in capturing the ground responses to tunneling in real time.

To predict ground responses alone the whole tunnel alignment, machine learning (ML)-based surrogate models have recently been proposed to complement the deficiency of conventional methods. Such models have strong capability of identifying the nonlinear relationship between ground responses and various influential factors [15-17]. Prediction models are established offline by directly learning from the in-situ data and used to online prediction of ground response in real time with high accuracy. The current ML-based tunneling-induced ground response prediction models were developed upon quite limited datasets (within 1000 datasets), ~~consequently,~~thereby the model architecture is not sophisticated (within 20 input variables). Researchers thus utilized metaheuristic optimization algorithms such as particle swarm optimization (PSO) ~~for determining~~to search the hyper-parameters and general parameters of these ML-based models [18]. PSO has been successfully used in many domains [19], but the original PSO primarily exists two issues: premature convergence and high computational cost. The premature convergence means that PSO tends to be trapped in the local optima at the beginning of the search process. Meanwhile the computational cost can increase dramatically with the increasing population size, although the diversity of swarm is beneficial to obtain global optima. To mitigate these issues, numerous researchers have preoccupied with enhancing PSO algorithm, such as modified PSO with adaptive parameters [20, 21], hybrid PSO [22, 23]. Nevertheless, which action should be chosen for particles effectively moving towards the best position and when should take this action are still a key challenge.

In this study, a more general-purpose PSO optimizer enhanced by reinforcement learning (RL) deep Q-network (DQN) is proposed. In the past ~~three~~ several years, RL has driven impressive advances in artificial intelligence and rapidly extended their application scopes [24-27]. In particular, the models trained by DQN outperform human experts in Atari, Go and no-limit poker [28-30]. The most fundamental improvement is that deep RL algorithm does not rely on hand-crafted policy evaluation functions, compared with previous ML algorithms. The agent of deep RL interacts with environment and learn past experience like a human via self-playing, thereby continuously improve their performance. This success motivates us to propose a DQN-based PSO optimizer (DQN-PSO), in which agent guides particles to choose the optimum action at each generation and move towards the best position with the lowest computational cost. To the best knowledge of the authors, this is first work to combine RL algorithm DQN based optimizer to develop a global best ML based model for investigating ground responses to tunneling.

Hence, this study aims to develop an ELM-based ground response prediction model due to its fast calculation speed. The proposed DQN-PSO optimizer is used to optimize ELM for identifying the global best weights and biases~~ with higher accuracy and lower computational cost~~. A case study is implemented for validating the prediction performance of the proposed hybrid model. The framework of hybrid ELM and DQN-PSO optimizer proposed in this study can be replaced by various ML and metaheuristic algorithms to explore various issues.

2. Literature review and methodology

2.1 Literature review

Machine learning (ML) is a subsection of artificial intelligence that imparts the system to automatically learn from the data without being explicitly programmed. ML algorithms have made a significant breakthrough with appreciable performance in ~~a wide variety of~~many domains. They have been considered to be the best choice for discovering the intricate relationships ~~in~~ among high-dimensional data [31]. Ground responses to tunneling is complicated with the coupled effects of intrinsic and extrinsic factors such as geological, geotechnical, geometric, shield operational and anomalous parameters, which brings huge difficulties to accurately predict tunneling-induced settlement. Moreover, tunneling is a dynamic process and its influential factors always change with the advance of shield machine, thereby the real time prediction of settlement is vitally important in engineering practice. Conventional empirical, analytical, numerical and physical modelling methods have their limitations and cannot predict soil-shield machine interaction in real time. ML algorithms provide a novel method to overcome this issue.

Since the first application of ANN to predict tunneling-induced settlement conducted by Shi et al. [32], various ML algorithms have been extensively used to predict soil-shield machine interaction in the last two decades. The most widely used ML algorithm is the ANN with the error backpropagation [33-39]~~ and~~. Meanwhile its variants such as general regression neural network [40], wavelet neural network [14] and radial basis function neural network [40] have ~~also been employed~~gained popularity in predicting soil-shield machine interaction. In the last decade, the development of ML has experienced a course of blossom~~,~~. ~~consequently~~Consequently, researchers have implemented various ML algorithms to predict tunneling-induced ground settlement such as extreme learning machine [41], adaptive neuro fuzzy inference system [42, 43], relevance vector machine [44], least-squares support-vector machine [45], random forest [46-48] and genetic expression programming [42].

The key of developing a ML-based settlement prediction model is to determine the values of hyper-parameters. In addition, the weights and biases also need to be determined for ANN and its variants. The commonly used methods for determining hyper-parameters involve trial and error, grid search and meta-heuristic algorithms [34, 39, 40]~~, and the~~. The weights and biases of ANN-based models ~~can be~~are generally determined using ~~gradient descend~~deterministic and ~~meta-heuristic~~stochastic optimization algorithms [18, 35]. Trial and error and grid search methods can only search the parameters in a limited space. Deterministic optimization algorithm such as gradient descend may be trapped into local optima. Stochastic algorithms suffer from premature convergence and high computational cost. The global best~~ optimum ~~parameters are thus hard to be obtained by using such method~~, because ~~.~~trial and error and grid search methods can only search the parameters in a limited space, gradient descend may be trapped into local optima, and meta-heuristic algorithms suffer from premature convergence and high computational cost. Therefore~~ To this end, this study proposes a RL algorithm DQN based optimizer to search the global optimum parameters of ML algorithms with higher accuracy and lower computational cost.

2.2 Reinforcement learning

2.2.1 Framework of reinforcement learning

Reinforcement learning (RL) is originated from a discrete-time and finite *Markov decision process* (MDP). RL consists of a learning agent, an environment, states, actions, and rewards. The agent interacts with an environment at some discrete time scale, $t = 0, 1, \ldots$. On each time step t, the agent perceives or observes the state of the environment, S_t ($S_t \in S$), thereafter chooses a primitive action based on this perception or observation, A_t ($A_t \in A_{S_t}$). In response to each action, a, the environment thereafter produces a numerical reward, R_{t+1}, and changes to a next state, S_{t+1} ($S_{t+1} \in S$). The whole dynamic transition process can be mathematically expressed by:

$$\mathbb{P}_{ss'}^{a} = \Pr\left\{S_{t+1} = s' \middle| S_t = s, A_t = a\right\} \tag{1}$$

$$R_s^a = E\left\{R_{t+1} \middle| R_t = s, R_t = a\right\} \tag{2}$$

where $\mathbb{P}_{ss'}^{a}$ = state transition probability matrix; R_s^a = immediate reward.

Note that the action at a state is selected based on a policy, π.

$$\pi\left(a \middle| s\right) = P\left\{A_t = a \middle| S_t = s\right\} \tag{3}$$

Therefore, the objective of the learning agent is to learn a policy which maximizes the expected discounted future reward at each state after maps from states to probabilities of taking each available primitive action, as shown by:

$$\begin{aligned} V_\pi(s) &= E_\pi\left\{R_{t+1} + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots \middle| S_t = s\right\} \\ &= E_\pi\left\{R_{t+1} + \gamma V_\pi(S_{t+1}) \middle| S_t = s\right\} \\ &= \sum_{a\in A} \pi(s,a)\left[R_s^a + \gamma \sum_{s'} P_{ss'}^a V_\pi(s')\right] \end{aligned} \tag{4}$$

where $\gamma \in (0, 1)$ = a discount factor, it denotes the reward from next states gradually decreases; v_π = state-value function under policy, π; $v_\pi(s)$ = value of the state, s, under policy, π.

The state-value function is the expected return starting from state, s, and then following policy, π. There is another value function that is the expected return starting from state, s, taking action a, and then following policy, π, which is termed as action-value function, as shown following:

$$\begin{aligned} Q_\pi(s,a) &= E_\pi\left\{R_{t+1} + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots \middle| S_t = s, A_t = a\right\} \\ &= R_s^a + \gamma \sum_{s'\in S} P_{ss'}^a V_\pi(s') \\ &= R_s^a + \sum_{s'\in S} P_{ss'}^a \sum_{a'\in A} \pi(a'|s') Q_\pi(s',a') \end{aligned} \tag{5}$$

The objective of value-based RL algorithms such as Q-learning and Sarsa is to determine the optimum state-value V^*(s) or action-value functions Q^*(s, a), as shown in Eq. [6]–[7]. This study also utilizes value-based RL algorithm to establish model.

$$V^*(s) = \sum_{a\in A}\left[R_s^a + \gamma \sum_{s'} P_{ss'}^a V^*(s')\right] \tag{6}$$

$$Q^*(s,a) = R_s^a + \sum_{s'\in S} P_{ss'}^a \max_{a'\in A} Q^*(s',a') \tag{7}$$

2.2.2 Deep Q network

In this study, a deep ~~reinforcement learning~~RL ~~model is established by integrating a deep neural network (DNN) with a conventional RL to find out optimization strategy. This new~~ algorithm ~~model is~~ termed as

DQN proposed by Mnih et al. [29] is used. Conventional RL algorithms generally utilize a Q table to store states and actions. The values in the Q table update continuously complying with Eq. [8] during the learning process [49] (see Fig. 1(a)), thereby they have thus been limited to certain conditions with finite and discrete states and actions.

$$Q(s,a) = Q(s,a) + \alpha\left[R_s^a + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \tag{8}$$

where α = learning rate.

A DQN-based agent can interact with an environment with continuous states, because a DNN can parameterize an approximate action-value function $Q(s, a; \theta_i)$ (see Fig. 1(b)). Nevertheless, the sequence of observations lead to a strong correlation among these observations, thereby the neural network-based action-value function may be unstable and even diverge [50], thereby an experience replay method is proposed [29]. In this method, agent's experience $e_t = (S_t, A_t, R_t, S_{t+1})$ at the time- step t is stored in a replay memory pool D. DNN can be trained based on mini-batches $(s, a, r, s') \sim U(D)$ that are randomly drawn from the memory pool, which is beneficial to eliminate strong correlations among observations and ensures that the learning system is stable. The corresponding loss function of DNN is:

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim U(D)}\left[R_s^a + \gamma \max_{a'} Q(s',a';\theta_i^-) - Q(s,a;\theta_i)\right] \tag{9}$$

where θ_i = parameters of Q-DNN at the ith iteration; θ_i^- = parameters of target DNN at the ith iteration. Note that only the parameters of Q-DNN are updated in real time. Target DNN is a forward network and has a same architecture with Q-DNN. The update of parameters θ_i^- is achieved by directly extracting parameters from Q-DNN at a fixed interval. In this way, training can avoid falling into feedback loops and proceed in a more stable manner.

2.3 Particle swarm optimization

Particle swarm optimization (PSO) is a metaheuristic optimization algorithm [51], which is developed based on simulating search behavior and social interaction of animals such as fish school and bird flock. PSO consists of several populations of particles and each particle is represented by its position vector $\boldsymbol{X}_i^k$, velocity vector $\boldsymbol{V}_i^k$ and fitness value. The velocity and position of each particle are updated using the following equations:

$$\boldsymbol{V}_i^{k+1} = \omega * \boldsymbol{V}_i^k + c_1 * r_1 * \left(\boldsymbol{pBest}_i^k - \boldsymbol{X}_i^k\right) + c_2 * r_2 * \left(gBest^k - \boldsymbol{X}_i^k\right) \tag{10}$$

$$\boldsymbol{X}_i^{k+1} = \boldsymbol{X}_i^k + \boldsymbol{V}_i^{k+1} \tag{11}$$

where k = current generation, i = ith particle; ω = inertia weight; c_1, c_2= cognitive and social acceleration coefficients; r_1, r_2 = random numbers within the range [0, 1] complying with uniform distribution; $\boldsymbol{pBest}_{\mathrm{i}}$ = local best location of the ith particle; $gBest$ = global best location of all particles. The predominant objective of the PSO algorithm is to find the optimum fitness value and the corresponding location.

2.4 Extreme learning machine

Extreme learning machine (ELM) is a modification of the single-hidden layer feedforward neural network, that is, only one hidden layer in this algorithm. The weights of input layer and the biases of hidden layer are assigned randomly. The optimum ELM is obtained by calculating the weights of the hidden layer and the biases of the output layer, thereby the calculation speed is much faster. In general, ELM can be represented as:

$$\mathbf{H} = f\left(\mathbf{X}\mathbf{w} + \boldsymbol{b}\right) \tag{12}$$

$$\left\|\mathbf{H}\boldsymbol{\beta} - \boldsymbol{y}\right\| = 0 \tag{13}$$

where $\mathbf{w}$ = weight matrix of the input layer; $\boldsymbol{b}$ = bias vector of the hidden layer; $\mathbf{H}$ = hidden layer output matrix; $\boldsymbol{\beta}$ = weight vector connecting the hidden nodes and the output nodes; $\boldsymbol{y}$ = outputs; f = activation function. The optimum ELM algorithm can be achieved by minimizing the value of $\|\mathbf{H}\boldsymbol{\beta}-\boldsymbol{y}\|$. Detailed description of ELM algorithm can refer to Huang et al. [52].

3. Introduction of proposed model

3.1 Proposed ELM-based ground response prediction model

3.1.1 Feature selection

Recent work by Zhang et al. [48] demonstrated that the influential factors of tunneling-induced settlement can be mainly classified into four categories: tunnel geometry, geological condition, shield operational parameters and anomalous conditions. In detail, twelve parameters are vitally important to soil-tunnel interaction including one tunnel geometry factor (cover depth of tunnel *C*, it should be noted that cover depth of tunnel is the only geometric factor used in this study considering the tunnel specification along the whole section is consistent), five shield operational parameters (thrust *Th*, torque *To*, grout filling *Gf*, penetration rate *Pr*, chamber pressure *Cp*), five geological parameters (modified blow counts of standard penetration test of soil layers *MSPT*, modified blow counts of dynamic penetration test of soil layers *MDPT*, modified uniaxial compressive strength of weathered rocks *MUCS*, groundwater table *W* and soil type at the cutterhead face *St*) and one anomalous condition (shield stoppage *Sp*). This study still adopts these twelve parameters for developing ELM-based ground response prediction model. Herein, five shield operational parameters and *Sp* can be collected in real time during tunneling process~~, and~~. ~~the~~ The remaining geological and geometric parameters can be obtained during site investigation and route design process, which are conducted before the construction of tunnel. Therefore, tunnel-induced settlement can be predicted in real time.

3.1.2 Model architecture

The framework of the ELM-based ground response prediction model is presented in Fig. 2. Input layers have 12 neurons corresponding 12 input variables as mentioned above and ground maximum settlement *S* is the only output variable. ELM based model with different number of hidden neurons was pre-trained ~~to~~ for selecting an appropriate framework~~,~~. Considering the focus to this study is to highlight the superiority of proposed optimizer in the next section,~~but~~ the detailed processing for determine the optimum number of hidden neurons~~results~~ are not presented for brevity~~,~~. ~~because the focus to this study is to highlight the superiority of proposed optimizer in the next section.~~ The results indicate the performance of model is not sensitive to the hyper-parameters (the number of hidden neurons) when the number of hidden neurons exceed 15. Considering the computational cost and the model performance, 20 hidden neurons are ultimately adopted in this study. The training of ELM-based model can be obtained by:

$$\mathbf{H} = f_E\left(\mathbf{X}\mathbf{w} + \boldsymbol{b}\right); \quad \boldsymbol{\beta} = \mathbf{H}^{\dagger}\boldsymbol{y} \tag{14}$$

where **X** = input matrix (n×12, n is the number of datasets); **H** = output of hidden layer (n×20); $\boldsymbol{y}$ = output vector (n×1); **w** = weights matrix (12×20); $\boldsymbol{b}$ = bias vector (1×20); $\mathbf{H}^{\dagger}$ is obtained by *Moore–Penrose generalized inverse* of matrix H [53] (20×n), because **H** is a nonsquare matrix; $\boldsymbol{\beta}$ = ultimate training result (20×1); f_E is an activation function used in the hidden layer of ELM, and *sigmoid* activation function is adopted in this study, which can be expressed by:

$$f_E(x) = \frac{1}{1+e^{-x}} \tag{15}$$

3.2 Proposed deep reinforcement learning-based optimizer

3.2.1 States and actions

The novel optimizer is developed based on the integration of deep reinforcement learning algorithm DQN and meta-heuristic optimization algorithm PSO (DQN-PSO). The search space of population represents the environment of DQN, and positions of all particles represent the state of DQN. Three actions, i.e., exploration, exploitation and jump are considered in this study, as shown following:

(*i*) Exploration: in PSO, ω, c_1 and c_2 control the movement direction and scale of particles. At the early state of generation, particles tend to make a large movement to explore the search space and move far away from the current *gBest*. Therefore, ω and c_1 values are large, and c_2 value is small, as shown in Fig. 3(a). This operation is termed as exploration, and the update of each particle position and velocity complies with Eqs. [10]–[11].

(*ii*) Exploitation: at the later stage of generation, particles tend to make a small movement to slowly converge at *gBest* and avoid heavy vibration. Therefore, ω and c_1 values are small, and c_2 value is large, as shown in Fig. 3(b). This operation is termed as exploitation, and the update of each particle position and velocity complies with Eqs. [10]–[11].

(*iii*) Jump: the former two actions achieve the adaptive adjustment of parameters in PSO, but the algorithm is still likely to be trapped in the local optima and cannot jump out this status. Therefore, a jump action is assigned to the action space, which can be obtained by:

$$\boldsymbol{X}_i^{k+1} = \boldsymbol{pBest}_i^k + r * \left(X_{\max} - X_{\min} \right) \tag{16}$$

where r = random number within the range [–1, 1] complying with uniform distribution; $X_{\max}$ and $X_{\min}$ = upper and lower bound of particles location. This new location update method allows particles to jump out the local optima.

In this study, ε-greedy strategy is employed to select an action. The action that can generate maximum reward according to the results of Q-DNN is selected with probability(1–ε), but the action is randomly selected from all available actions for exploring unknown conditions with probability ε.

3.2.2 Boundary conditions

There are four boundary conditions in PSO, i.e., reflecting wall, damping wall, invisible wall, and absorbing wall. Absorbing wall is used to limit the position and velocity of particles in this study. As shown in Fig. 4, there are upper and lower bound for the position and velocity vectors. When they exceed this boundary condition, the position and velocity of each particle are reset as the values of upper or lower bounds, respectively (see Eqs. [17]–[18]). Otherwise, the update of position and velocity vector complies with Eqs. [10]–[11] and [16].

$$\boldsymbol{X}_i^{k+1} = \begin{cases} X_{\max}, & if \ \boldsymbol{X}_i^{k+1} > X_{\max} \\ X_{\min}, & if \ \boldsymbol{X}_i^{k+1} < X_{\max} \end{cases} \tag{17}$$

$$\boldsymbol{V}_i^{k+1} = \begin{cases} V_{\max}, & if \ \boldsymbol{V}_i^{k+1} > V_{\max} \\ V_{\min}, & if \ \boldsymbol{V}_i^{k+1} < V_{\max} \end{cases} \tag{18}$$

3.2.3 Basic framework

Fig. 5 presents the basic framework of the proposed optimizer DQN-PSO. This optimizer starts from creating several populations, which is also the current state of RL. The rewards of each action under this state will be estimated by the Q-DNN, ~~and~~ thereafter the action which can create maximum reward under this state will be selected. The velocity and position will be updated based on the selected action, thereby the new state will be generated. After the ***pBest*** and *gBest* are updated, the search process completes if the *gBest* ~~satisfy~~ satisfies the termination condition. Otherwise, the whole process will repeat.

3.3 Enhanced PSO optimizer

To validate the superiority of the proposed RL-based optimizer DQN-PSO, an enhanced optimizer is also developed for comparison. This enhanced PSO has two characteristics:

(*i*) Adaptive accelerator parameters: as mentioned above, particles start from exploring in the search space and thereafter transfer to exploitation operation with the increase of generations. Therefore, a search strategy that c_1 decreases linearly and c_2 increases linearly with the increase of generations has been developed [54]. This strategy can improve the global search capability of PSO ~~in~~ at the early stage and local optimization capability ~~in~~ at the later stage, as shown following:

$$c_1^k = c_{1\_initial} + \frac{k}{t}\left(c_{1\_final} - c_{1\_initial}\right) \tag{19}$$

$$c_2^k = c_{2\_initial} + \frac{k}{t}\left(c_{2\_final} - c_{2\_initial}\right) \tag{20}$$

where k = current generation; t = a total of generation; c_1^k, c_2^k = values of c_1 and c_2 at the kth generation, respectively; $c_{1\_initial}$, $c_{2\_initial}$ = initial values of c_1 and c_2, respectively; c_{1_final}, c_{2_final} = final values of c_1 and c_2, respectively. The update of each particle position and velocity complies with Eqs. [10]–[11].

(*ii*) Jump: unlike the DQN-PSO optimizer, enhanced PSO optimizer cannot intelligently select the action of particles based on the reward of each action. Therefore, when the number of generations exceeds a critical value (Eq. [21]) and ~~meanwhile~~ the difference of the objective function outputs generated at the adjacent time steps is less than a threshold value (Eq. [22]), a jump operation is activated. Thereafter the update of each particle position in the jump operation complies with Eq. [16].

$$k > N_J \tag{21}$$

$$f_{obj}^{k+1} - f_{obj}^{k} \le f_J \tag{22}$$

where k = current generation; f_{obj}^{k+1}, f_{obj}^{k} = output of objective function at the kth and (k+1)th generations, respectively; N_J, f_J = thresholds for the number of generations and the difference of adjacent outputs of the objective function, respectively.

3.4 Proposed hybrid deep RL model

3.4.1 Reward rule and actions selection

Note that the reward rule is not demonstrated in the former section, which needs to be determined by the hybrid model. As mentioned in the description of ELM, the training process of ELM is merely completed by computing the solution of a linear system. After the hyper-parameters (the number of hidden neurons) of ELM are determined, the model performance depends heavily on the weights and biases. To improve the performance of ELM-based ground response prediction model, the weights and biases of ELM are optimized by the DQN-based optimizer. In this hybrid algorithm, the state of the DQN-based optimizer represents the weights and biases of ELM, as shown in followings:

$$\mathbf{X} = \left[\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{i-1}, \boldsymbol{X}_i, \ldots, \boldsymbol{X}_n\right]_{n\times m} \tag{23}$$

$$\boldsymbol{X}_i = \left[x_1, x_2, \ldots, x_{i-1}, x_i, \ldots, x_m\right]_{1\times m} \tag{24}$$

where $\mathbf{X}$ = an aggregate of all populations; $\boldsymbol{X}_i$ = a single population. i. e, an aggregate of weights and biases of ELM; n = the size of population; m = the number of particles in each population, i.e. the number of weights and biases in ELM (20×12+20=260, as mentioned in section 3.1.2); x_i = a single particle of a population. Therefore, the number of particles in each state is 260n.

The objective function of the DQN-PSO optimizer is determined by the sum of squared errors (SSE), which is used to evaluate the reward value.

$$\mathrm{SSE} = \sum_{i=1}^{n}\left[\beta_i * f_E\left(\omega_i x_i + b_i\right) - y_i\right]^2 \tag{25}$$

where y_i = actual settlement; $\beta_i * f_E\left(\omega_i x_i + b_i\right)$ = predicted settlement using the ELM-based model, in which parameters β_i, ω_i and b_i derive from $\boldsymbol{\beta}$, $\mathbf{w}$ and $\boldsymbol{b}$ (see section 3.1.2), respectively; x_i = one set of input variables. The updates of ***pBest*** and *gBest* are related to the SSE value, in which all particles move towards the positions with low values of SSE. The reward rule is that the final reward r is 1 if the SSE yielded by *gBest* is less than the prescribed goal value, otherwise, the current model acquires reward ~~=~~ of 0. Note that the exact rewards are only known at the end of each episode.

3.4.2 Model framework

The pseudocode of the proposed hybrid ELM-based prediction model and DQN-PSO optimizer is presented in Algorithm 1. It can be observed that the hybrid algorithm involves prescribed number of episodes. In each episode, states are updated continuously until the SSE value yielded by the *gBest* can satisfies the termination condition.

Algorithm 1: Hybrid DQN and ELM algorithm

1. step = 0
2. **for** episode in range (number of prescribed episodes)
3. Randomly initialize state *s*
4. **while** True:
5. Estimate rewards of each action and choose an action *a*
7. Modify the current state *s* to *s_* by taking *a* and the corresponding *reward*
8. Store [*s*, *a*, *reward*, *s_*] in the replay memory pool
9. **if** step satisfies the update condition
10. Update target DNN

```
        if gBest generated by state s_ satisfies the termination conditions:
            Evaluate the reward of this episode
            break
        step = step + 1
Exit
```

4. Application of proposed model

4.1 Overview of case study

In order to investigate the feasibility and reliability of the proposed hybrid DQN-PSO optimizer and ELM-based tunneling-induced ground responses prediction model, an in-situ experiment conducted by Zhang et al. [48] on a practical tunneling project from Changsha city, China is used in this study. This experimental zone consisted of five tunnel sections with six metro stations. A total of 5.44 km was constructed using earth pressure balanced (EPB) shield machine (construction starting in 2016 and completing in 2019). The tunnel was primarily excavated in the weathered rocks, which means that the consolidation settlement completed rapidly after the tunnel was constructed. Therefore, this case study focused on the tunneling-induced ultimately steady ground settlement. Each monitoring cross-section of settlement was positioned at a fixed interval of around 10 m.

With regard to the collection of datasets, the geological conditions and geometric factor at each ring were obtained by the site investigation before tunneling process. The five operational parameters were recorded per minute by the shield machine data acquisition system, and the average operational parameters at each ring were preprocessed. The ground settlement monitoring points were installed at an interval of 10 m and was measured twice a day. The settlement of monitoring points and the 12 input variables at the corresponding positions were stored in the database for training ELM-based ground response prediction model, thereby synchronousness between the settlement data and the input variables can be guaranteed. The database used in this study can be downloaded in the Appendix section.

4.2 Results

Table 1 presents the values of parameters used in all algorithms in this study. The experimental results indicate the model performance is not particularly sensitive to the architecture of target DNN and Q-DNN. The number of hidden layers in the target DNN and Q-DNN is 1, and the corresponding number of neurons is 15. Q-DNN starts to training when the agent's experiences in the replay memory pool *D* reaches 200 and it is trained with an interval of 5 time steps. The size of mini-batch used for training Q-DNN is 32, and the parameters of the target DNN are updated with an interval of 300 time steps. A total of 100 game episodes are carried out by the intelligent agent. The computational results indicate the ELM-based prediction model showcases great performance with the *SSE* value of 5, thereafter the decrease in the goal value will lead to a dramatic increase in the computational cost and may fail to reach the goal value. Therefore, the goal value of SSE is ultimately defined as 5 in this study.

Fig. 6 presents the evolution of SSE value generated by the hybrid deep RL prediction model in a typical episode. It can be observed that this episode consumes 8802 steps to reach the goal SSE value. The whole evolution of SSE can be categorized into four phases according to the characteristics of SSE variation. At the phase I~~first phase~~ from *a* to *b*, SSE value experiences a remarkable decrease from 8.395 at the 1st step to 5.288 at the 1045th step. Thereafter, the change in the SSE value is not discernable, but three steady phases can be obviously observed. The first steady phase (phase II: *b* – *c*) continues a total of 3373 steps, followed by phase III (c – d) with 2919 steps and phase IV (d – e) with 665 steps.

The advantage of the RL algorithm DQN is that it can reveal the intelligent operation mechanism of the agent, while other ML-based models merely run as a black box. To investigate the operation mechanism of the DQN-PSO optimizer, the actions at four phases are presented, as shown in Fig. 7. At the phase I, it can be observed that the agent focuses on the exploration at the initial stage, implying this action can receive the largest reward based on the Q-DNN results. The performance of the hybrid deep RL prediction model at the initial stage is not steady, thereby the optimization of weighs and biases can easily improve the prediction accuracy, which complies with the obvious decrease in the SSE value (see Fig. 6). At the earth stage of phase II, the agent still starts from exploration, but this action cannot reduce the SSE value, thereby the agent transfers to conduct the exploitation action, and sometimes conduct the jump action for jumping out local optima. Consequently, the exploitation and jump actions alternately appear and dominate this phase. At the phase III, the agent focuses on the exploitation, because the action trials at the phase II cannot cause large decrease in the SSE value (see Fig. 6). It indicates the performance of the hybrid deep RL prediction model is roughly steady, and SSE will converge at a fixed value. Similar condition can also be observed at the phase IV, where exploitation still dominate this phase. SSE value varies within an acceptable range and end up with the prescribed goal value, thereby it is reasonable to deduce the optimum hybrid deep RL prediction model for predicting tunneling-induced ground response is obtained. The consistency of agent's action and the corresponding model performance at each phase ensures the reasonability of the hybrid deep RL prediction model. The agent like a human intelligently guides particles to choose the optimum action at each generation and move towards the best position.

To clearly reveal the evolution of the prediction performance of model, the predicted maximum settlement for the test set using the hybrid deep RL prediction model generated at three typical steps *a*, *b* and *e* are presented, as shown in Fig. 8. It can be seen that the predicted settlement using the model generated at the step *a* severely deviates from the measured settlement. It cannot accurately capture the evolution of tunneling-induced settlement and loses fidelity at some monitoring points, e.g. the largest settlement of 48 mm is not detected. The performance of model generated at the step *b* improves dramatically. The predicted evolution of settlement shows great agreement with the measured settlement ~~and the settlement values can also be accurately predicted~~. Meanwhile all of large settlement that exceeds 10 mm can be detected by this model, which is of great significance for avoiding risks in engineering practice. The performance of model generated at the step *e* is further refined with the lower SSE value, compared with the model generated at the step *b*. In detail, the difference of predicted and measured settlements at some monitoring points further reduces and shows better consistency with the measured evolution of ground maximum settlement.

5. Discussion

5.1 Compared with basic and enhanced PSO

To validate the superiority of the proposed RL-based optimizer DQN-PSO, a comparison among three optimizers, that is, basic PSO, enhanced PSO and DQN-PSO, is conducted. Fig. 9 presents the results of ELM-based ground responses prediction model optimized by three optimizers. The evolution of SSE value within 3000 generations is presented because three optimizers roughly converged at a fixed value. It can be observed that DQN-PSO obviously outperforms the basic and enhanced PSO with the lowest value of SSE and fastest convergence. The corresponding maximum generation of three types of optimizers when SSE values virtually converge at a constant value is~~are~~ presented in Table 2. In detail, the SSE value optimized by the DQN-PSO starts to be less than the basic and enhanced PSO when the number of generations exceeds 10, because the DQN-PSO optimizer always guides particles selecting a correct action. Meanwhile the whole optimization process virtually completes at around the 1000th generation with SSE value of 5.288, thereafter the objective of search operation is merely for achieving the prescribed goal value of SSE and the computational cost is expensive. It can be seen from Fig. 9 that the computational cost for decreasing SSE value from 5.288 to the prescribed goal value is appropriately seven times the figure for decreasing SSE value from the initial value to 5.288. It is noteworthy that the enhanced PSO also outperform the basic PSO with lower value of SSE from the 326th generation. Enhanced PSO indeed further optimizes the search trajectory of particles to a certain extent, but the key challenges including which action should be chose and when should take this action are still dodged. It means that the enhanced PSO cannot avoid being trapped in the local optima, thereby the decrease in the convergence SSE value is not discernable, compared with the basic PSO. The basic PSO conducts the exploration action throughout the whole optimization process, thereby it is easy to be trapped in the local optima. The premature convergence problem is obvious, because the SSE value roughly maintains constant when the number of generations reaches 500.

Fig. 10 presents the evolution of ground responses for the test set predicted by the ELM-based prediction model optimized by three optimizers as well as the MAE values computed using Eq. [~~24~~26]. It can be seen that the hybrid deep RL model outperforms ELM-based prediction models optimized by PSO and enhanced PSO. Enhanced PSO slightly refines the predicted settlement evolution with a slight decrease in the MAE value (from 2.64 to 2.51). The improvement in the prediction performance of the hybrid deep RL model is remarkable, in which the MAE value decreases to 1.97. The great agreement between the predicted and measured evolution of settlement and the improvement in recognizing maximum settlement is observed. Meanwhile all datasets are closer to the line with the slope of 1. Hence the tunneling-induced ground responses prediction model can be established using the hybrid deep RL algorithm.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|r_i - p_i\right| \tag{26}$$

where r = measured settlement; p = the predicted settlement; n = a total number of datasets.

5.2 Sensitivity analysis

To evaluate the performance of the proposed ELM based settlement prediction model optimized by DQN-PSO. Global sensitivity analysis (GSA) is conducted to reveal how model output uncertainty can be apportioned to the uncertainty in each input variable [55]. Variance-based GSA method has been extensively used in main domains [56-58], thereby it is used in this study. In this method, the total order index S_{Ti} in the variance-based GSA method measures the effect of an input parameter and its coupled effect with other input parameters on the model output. The calculation of S_{Ti} proposed by Jansen [59] is adopted in this study, and the detailed formulations are not presented for brevity, which can refer to Zhang [60]. The results of GSA are shown in Fig. 11, compared with the correlation coefficients which are calculated by absolute *Pearson* coefficients (see Eq. [27]). It can be observed that the parameters that have strong correlations with settlement (*Sp*, *St*, *C*) still have higher impact on the ELM based model. *Th* with the highest *Pearson* value among five operational parameters is also the most important operational parameter in the ELM based model. *Pr* with the lowest *Pearson* value is also the most insignificant parameter in the ELM based model. The rank of other parameters merely has a slight variation. Such factors indicate the ELM based model optimized by DQN-PSO obviously ~~learns~~ captures the potential correlations between the input and output parameters~~, and t~~. The generalization ability and the practicability of such model can thus be guaranteed.

$$R = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2}} \tag{27}$$

6. Conclusions

The contribution of this study is that a hybrid deep reinforcement learning (RL) model which integrates extreme learning machine (ELM) and deep RL algorithm deep-Q network (DQN) is proposed for predicting tunneling-induced ground responses in real time, in which the relationships among influential factors and ground response were explored through self-practicing~~, rather than developed based on a beforehand fixed formation~~. Another contribution is that the proposed optimizer DQN-PSO knows which action should be conducted and when should take this action, thereby ensures the global optima ~~can~~ to be obtained. Unlike previous metaheuristic optimization algorithms that guide the movement of particles in a rough manner, the reward rule of the DQN-based optimizer focuses on evaluating the reward of agent’s action, hence particles like an intelligent human always select the optimum action at each step. To authors’ best knowledge, this is the first work on using hybrid RL algorithm DQN and ML algorithm ELM to investigate tunneling-induced ground responses. The following conclusions can be drawn, based on the results of this work:

(1) Because DQN-PSO optimizer is able to guide particles to implement optimum action at each step, the global optima can be acquired when the value of objective function converges at a fixed value. In other words, the DQN-PSO optimizer can search the global best weights and biases of ELM with higher accuracy and lower computational cost, compared with basic or enhanced metaheuristic optimization

algorithms.

(2) The hybrid deep RL model with the integration of ELM and DQN-PSO optimizer can accurately predict tunneling-induced ground response in real time, overcoming the deficiency of empirical, analytical and numerical models established by domain experts. The ultimate ELM based model can be expressed with an explicit formulation, which is user-friendly in engineering practice. Meanwhile, the performance of prediction model can be improved with the increase in the datasets collected from the field construction.

(3) The hybrid deep RL model is genetic, which means that it can be used to various situations with different states, actions, rules, rewards and objective function defined by domain experts without any debugging. Meanwhile the basic meta-heuristic and machine learning algorithms used in the hybrid deep RL model can be randomly replaced based on different situations. Such model offers a pragmatic and reliable framework to develop a data-driven or physical model.

Appendix

The database used in this study can be download at following link:

https://www.researchgate.net/publication/336208927_Database_for_maximum_settlement_collected_from_Changsha_Metro_Line_4_Liugoulong_to_Fubuhe_station

Acknowledges

This study is sponsored by the National Natural Science Foundation of China (No. 51938005) and the program of High-level Talent of Innovative Research Team of Hunan Province 2019 (No. 2019RS1030).

The authors greatly appreciate these financial supports during this research.

References

[1] P. Zhang, H.-N. Wu, R.-P. Chen, T.H.T. Chan, Hybrid meta-heuristic and machine learning algorithms for tunneling-induced settlement prediction: A comparative study, Tunnell. Undergr. Space Technol., 99 (2020) 103383.
[2] C. Sagaseta, Analysis of undrained soil deformation due to ground loss, Géotechnique, 37 (1987) 301-320.
[3] A. Verruijt, J.R. Booker, Surface settlements due to deformation of a tunnel in an elastic half plane, Géotechnique, 48 (1996) 709-713.
[4] H.S. Yu, R.K. Rowe, Plasticity solutions for soil behaviour around contracting cavities and tunnels, Int. J. Numer. Anal. Met., 23 (1999) 1245–1279.
[5] F. Pinto, A.J. Whittle, Ground movements due to shallow tunnels in soft ground. I: analytical solutions, J. Geotech. Geoenviron., 140 (2014) 04013040.
[6] W. Broere, D. Festa, Correlation between the kinematics of a Tunnel Boring Machine and the observed soil displacements, Tunnell. Undergr. Space Technol., 70 (2017) 125-147.
[7] S. Suwansawat, H.H. Einstein, Describing settlement troughs over twin tunnels using a superposition technique, J. Geotech. Geoenviron. Eng., 133 (2007) 445-468.
[8] P. Zhang, Z.Y. Yin, R.P. Chen, Analytical and semi-analytical solutions for describing tunneling-induced transverse and longitudinal settlement troughs, Int. J. Geomech., (2020) in press.
[9] R.B. Peck, Deep excavations and tunneling in soft ground, Proceedings of 7th International Conference on Soil Mechanic and Foundation Engineering Mexico City, 1969, pp. 225–290.
[10] M. Jiang, Z.Y. Yin, Analysis of stress redistribution in soil and earth pressure on tunnel lining using the discrete element method, Tunnell. Undergr. Space Technol., 32 (2012) 251–259.
[11] P. Zhang, R.-P. Chen, H.-N. Wu, Y. Liu, Ground settlement induced by tunneling crossing interface of water-bearing mixed ground: A lesson from Changsha, China, Tunnell. Undergr. Space Technol., 96 (2020) 103224.
[12] X.-T. Lin, R.-P. Chen, H.-N. Wu, H.-Z. Cheng, Deformation behaviors of existing tunnels caused by shield tunneling undercrossing with oblique angle, Tunnell. Undergr. Space Technol., 89 (2019) 78-90.
[13] J. Ninić, S. Freitag, G. Meschke, A hybrid finite element and surrogate modelling approach for simulation and monitoring supported TBM steering, Tunnell. Undergr. Space Technol., 63 (2017) 12-28.
[14] A. Pourtaghi, M.A. Lotfollahi-Yaghin, Wavenet ability assessment in comparison to ANN for predicting the maximum surface settlement caused by tunneling, Tunnell. Undergr. Space Technol., 28 (2012) 257-271.
[15] P. Zhang, Z.-Y. Yin, Y.-F. Jin, T.H.T. Chan, A novel hybrid surrogate intelligent model for creep index prediction based on particle swarm optimization and random forest, Eng. Geol., 265 (2020) 105328.
[16] P. Zhang, Z.Y. Yin, Y.F. Jin, G.L. Ye, An AI-based model for describing cyclic characteristics of

granular materials, Int. J. Numer. Anal. Met., (2020) 1-21.

[17] P. Zhang, Z.Y. Yin, Y.F. Jin, T. Chan, Intelligent Modelling of Clay Compressibility using Hybrid Meta-Heuristic and Machine Learning Algorithms, Geoscience Frontiers, (2020) in press.

[18] D.J. Armaghani, E.T. Mohamad, M.S. Narayanasamy, N. Narita, S. Yagiz, Development of hybrid intelligent models for predicting TBM penetration rate in hard rock condition, Tunnell. Undergr. Space Technol., 63 (2017) 29-43.

[19] Z.Y. Yin, Y.F. Jin, S.J. S, P.Y. Hicher, Optimization techniques for identifying soil parameters in geotechnical engineering: comparative study and enhancement, Int. J. Numer. Anal. Met., 42 (2017) 1-25.

[20] K.E. Parsopoulos, Parallel cooperative micro-particle swarm optimization: A master–slave model, Appl. Soft Comput., 12 (2012) 3552-3579.

[21] W.H. Lim, N.A. Mat Isa, Two-layer particle swarm optimization with intelligent division of labor, Eng. Appl. Artif. Intel., 26 (2013) 2327-2348.

[22] A. Gálvez, A. Iglesias, A new iterative mutually coupled hybrid GA–PSO approach for curve fitting in manufacturing, Appl. Soft Comput., 13 (2013) 1491-1504.

[23] S.Z. Zhao, P.N. Suganthan, Q.-K. Pan, M. Fatih Tasgetiren, Dynamic multi-swarm particle swarm optimizer with harmony search, Expert Syst. Appl., 38 (2011) 3735-3742.

[24] M.A. Lopes Silva, S.R. de Souza, M.J. Freitas Souza, A.L.C. Bazzan, A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems, Expert Syst. Appl., 131 (2019) 148-171.

[25] K. Zhang, H. Zhang, Y. Mu, S. Sun, Tracking control optimization scheme for a class of partially unknown fuzzy systems by using integral reinforcement learning architecture, Appl. Math. Comput., 359 (2019) 344-356.

[26] Y. Ding, L. Ma, J. Ma, M. Suo, L. Tao, Y. Cheng, C. Lu, Intelligent fault diagnosis for rotating machinery using deep Q-network based health state classification: A deep reinforcement learning approach, Adv. Eng. Inform., 42 (2019).

[27] F. Hourfar, H.J. Bidgoly, B. Moshiri, K. Salahshoor, A. Elkamel, A reinforcement learning approach for waterflooding optimization in petroleum reservoirs, Eng. Appl. Artif. Intel., 77 (2019) 98-116.

[28] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of Go with deep neural networks and tree search, Nature, 529 (2016) 484-489.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature, 518 (2015) 529-533.

[30] Noam Brown, T. Sandholm, Superhuman AI for multiplayer poker, Science, (2019) 1-12.

[31] S. Dargan, M. Kumar, M.R. Ayyagari, G. Kumar, A survey of deep learning and its applications: a new paradigm to machine learning, Archives of Computational Methods in Engineering, (2019).

[32] J. Shi, J.A.R. Ortigao, J. Bai, Modular neural networks for predicting settlements during tunneling, J. Geotech. Geoenviron. Eng., 124 (1998) 389-395.

[33] C.Y. Kim, G.J. Bae, S.W. Hong, C.H. Park, Neural network based prediction of ground surface

settlements due to tunnelling, Comput. Geotech., 28 (2001) 517-547.

[34] S. Suwansawat, H.H. Einstein, Artificial neural networks for predicting the maximum surface settlement caused by EPB shield tunneling, Tunnell. Undergr. Space Technol., 21 (2006) 133-150.

[35] O.J. Santos, T.B. Celestino, Artificial neural networks analysis of São Paulo subway tunnel settlement data, Tunnell. Undergr. Space Technol., 23 (2008) 481-491.

[36] A. Marto, M. Hajihassani, R. Kalatehjari, E. Namazi, H. Sohaei, Simulation of longitudinal surface settlement due to tunnelling using artificial neural network, International Review on Modelling and Simulations, 5 (2012) 1024-1031.

[37] A. Darabi, K. Ahangari, A. Noorzad, A. Arab, Subsidence estimation utilizing various approaches – A case study: Tehran No. 3 subway line, Tunnell. Undergr. Space Technol., 31 (2012) 117-127.

[38] R. Boubou, F. Emeriault, R. Kastner, Artificial neural network application for the prediction of ground surface movements induced by shield tunnelling, Can. Geotech. J., 47 (2010) 1214-1233.

[39] M. Hasanipanah, M. Noorian-Bidgoli, D. Jahed Armaghani, H. Khamesi, Feasibility of PSO-ANN model for predicting surface settlement caused by tunneling, Eng. Comput-Germany, 32 (2016) 705-715.

[40] R.P. Chen, P. Zhang, X. Kang, Z.Q. Zhong, Y. Liu, H.N. Wu, Prediction of maximum surface settlement caused by EPB shield tunneling with ANN methods, Soils Found., 59 (2019) 284–295.

[41] R.P. Chen, P. Zhang, H.N. Wu, Z.T. Wang, Z.Q. Zhong, Prediction of shield tunneling-induced ground settlement using machine learning techniques, Front. Struct. Civ. Eng., 13 (2019) 1363–1378.

[42] K. Ahangari, S.R. Moeinossadat, D. Behnia, Estimation of tunnelling-induced settlement by modern intelligent methods, Soils Found., 55 (2015) 737-748.

[43] D. Bouayad, F. Emeriault, Modeling the relationship between ground surface settlements induced by shield tunneling and the operational and geological parameters based on the hybrid PCA/ANFIS method, Tunnell. Undergr. Space Technol., 68 (2017) 142-152.

[44] F. Wang, B. Gou, Y. Qin, Modeling tunneling-induced ground surface settlement development using a wavelet smooth relevance vector machine, Comput. Geotech., 54 (2013) 125-132.

[45] L.M. Zhang, X.G. Wu, W.Y. Ji, S.M. AbouRizk, Intelligent approach to estimation of tunnel-induced ground settlement using Wavelet Packet and Support Vector Machines, J. Comput. Civil. Eng., 31 (2017) 04016053.

[46] J. Zhou, X. Shi, K. Du, X.Y. Qiu, X.B. Li, H.S. Mitri, Feasibility of Random-Forest approach for prediction of ground settlements induced by the construction of a shield-driven tunnel, International Journal of Geomechanics, 17 (2016) 04016129.

[47] V.R. Kohestani, M.R. Bazargan-Lari, J. Asgari-marnani, Prediction of maximum surface settlement caused by earth pressure balance shield tunneling using random forest, Journal of AI and Data Mining, 5 (2017) 127-135.

[48] P. Zhang, R.P. Chen, H.N. Wu, Real-time analysis and regulation of EPB shield steering using Random Forest, Automat. Constr., 106 (2019) 102860.

[49] C.J.C.H. Watkins, P. Dayan, Q-Learning, Mach. Learn., 8 (1992) 279-292.

[50] Y. Yuan, Z.L. Yu, Z. Gu, Y. Yeboah, W. Wei, X. Deng, J. Li, Y. Li, A novel multi-step Q-learning method to improve data efficiency for deep reinforcement learning, Knowl-Based Syst., 175 (2019) 107-117.

[51] J. Kennedy, R. Eberhart, Particle swarm optimization, IEEE International Conference on Neural

NetworksPerth, Australia, 1995, pp. 1942-1948.
[52] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: Theory and applications, Neurocomputing, 70 (2006) 489-501.
[53] C.R. Rao, S.K. Mitra, Generalized inverse of matrices and its applications, Wiley1971.
[54] L. Yang, H. Su, Z. Wen, Improved PLS and PSO methods-based back analysis for elastic modulus of dam, Adv. Eng. Softw., 131 (2019) 205-216.
[55] A. Saltelli, I.M. Sobol, About the use of rank transformation in sensitivity analysis of model output, Reliab. Eng. Syst. Safe., 50 (1995) 225-239.
[56] C.Y. Zhao, A.A. Lavasan, R. Hölter, T. Schanz, Mechanized tunneling induced building settlements and design of optimal monitoring strategies based on sensitivity field, Comput. Geotech., 97 (2018) 246-260.
[57] L.M. Zhang, X.G. Wu, H.P. Zhu, S.M. AbouRizk, Performing global uncertainty and sensitivity analysis from given data in tunnel construction, J. Comput. Civil. Eng., 31 (2017) 04017065.
[58] K.M. Hamdia, H. Ghasemi, X.Y. Zhuang, N. Alajlan, T. Rabczuk, Sensitivity and uncertainty analysis for flexoelectric nanostructures, Comput. Method Appl. M., 337 (2018) 95-109.
[59] M.J.W. Jansen, Analysis of variance designs for model output, Comput. Phys. Commun., 117 (1999) 35-43.
[60] P. Zhang, A novel feature selection method based on global sensitivity analysis with application in machine learning-based prediction model, Appl. Soft Comput., 85 (2019) 105859.

Table

Table 1 Values of parameters in three algorithms

Algorithm	Parameter	Value
ELM	Number of hidden neurons	20
PSO	ω (exploration | exploitation)	0.9 | 0.4
	c_1 (exploration | exploitation)	2.5 | 0.4
	c_2 (exploration | exploitation)	0.4 | 2.5
	$[V_{min}, V_{max}]$	[-3, 3]
	$[X_{min}, X_{max}]$	[-10, 10]
	Population size	20
	Maximum generation	3000
Enhanced PSO	c_1 (initial | final)	2.5 | 0.5
	c_2 (initial | final)	0.5 | 2.5
	N_J	2000
	f_J	0.01
DQN	Number of hidden neurons	15
	RL learn (criteria| step)	200| 5
	Target DNN update interval	300
	Batch size	32
	Episode	100
	goal_*SSE*	5
	Reward decay coefficient γ	0.9
	ε-greedy	0.1
	Learning rate α	0.01

Table 2 Comparison among three optimizers

Optimizer	Generation	SSE
Basic PSO	1497	6.860
Enhanced PSO	1280	6.540
DQN-PSO	1045	5.288

Figure caption

Fig. 1 Schematic view of reinforcement learning: (a) Q-learning; (b) deep Q-network

Fig. 2 Architecture of ELM-based ground response prediction model

Fig. 3 Search methods of particles: (a) exploration; (b) exploitation

Fig. 4 Absorbing wall boundary condition

Fig. 5 Framework of proposed DQN-based PSO optimizer

Fig. 6 Evolution of SSE value in a typical episode

Fig.7 Actions at four phases

Fig. 8 Predicted settlement for the test set using the hybrid deep RL model generated at three steps

Fig. 9 Comparison of DQN-PSO optimizer with basic and enhanced PSO optimizers

Fig. 10 Predicted settlement for the test set using ELM-based prediction models optimized by three optimizers

Fig. 11 Comparison between sensitivity indices and correlation coefficients of input parameters

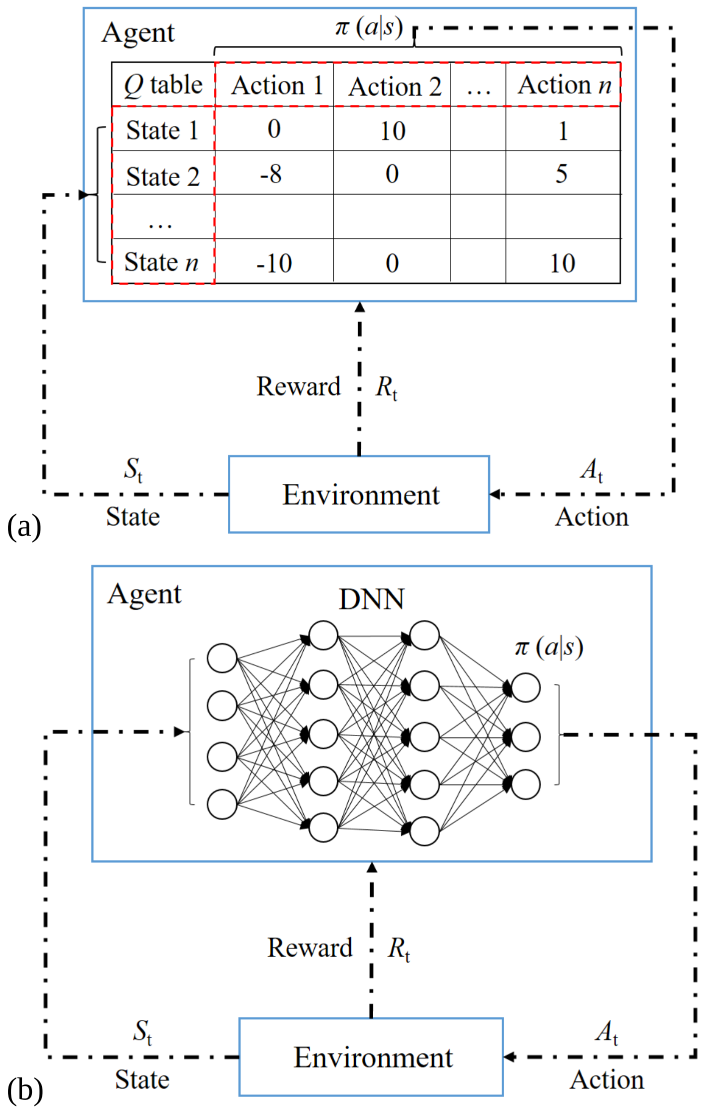


Fig. 1 Schematic view of reinforcement learning: (a) Q-learning; (b) deep Q-network

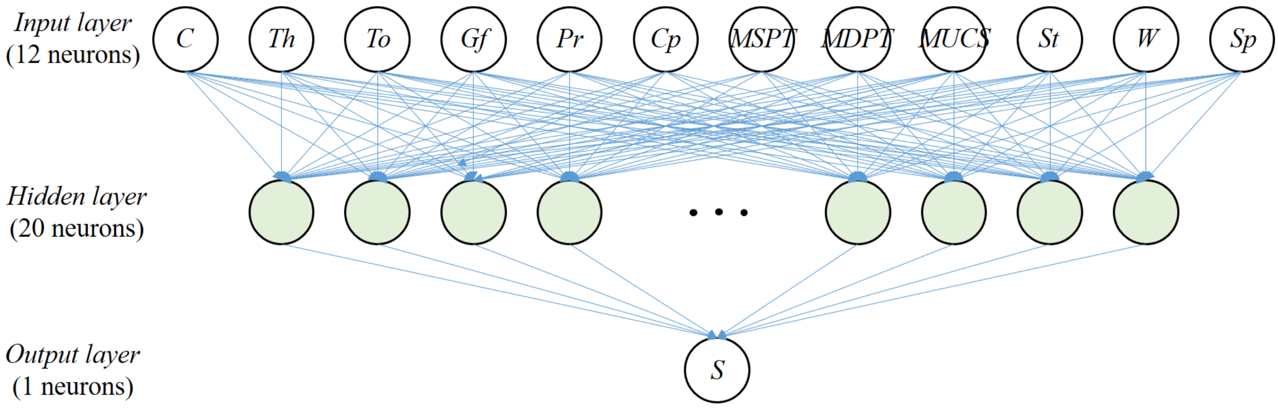


Fig. 2 Architecture of ELM-based ground response prediction model

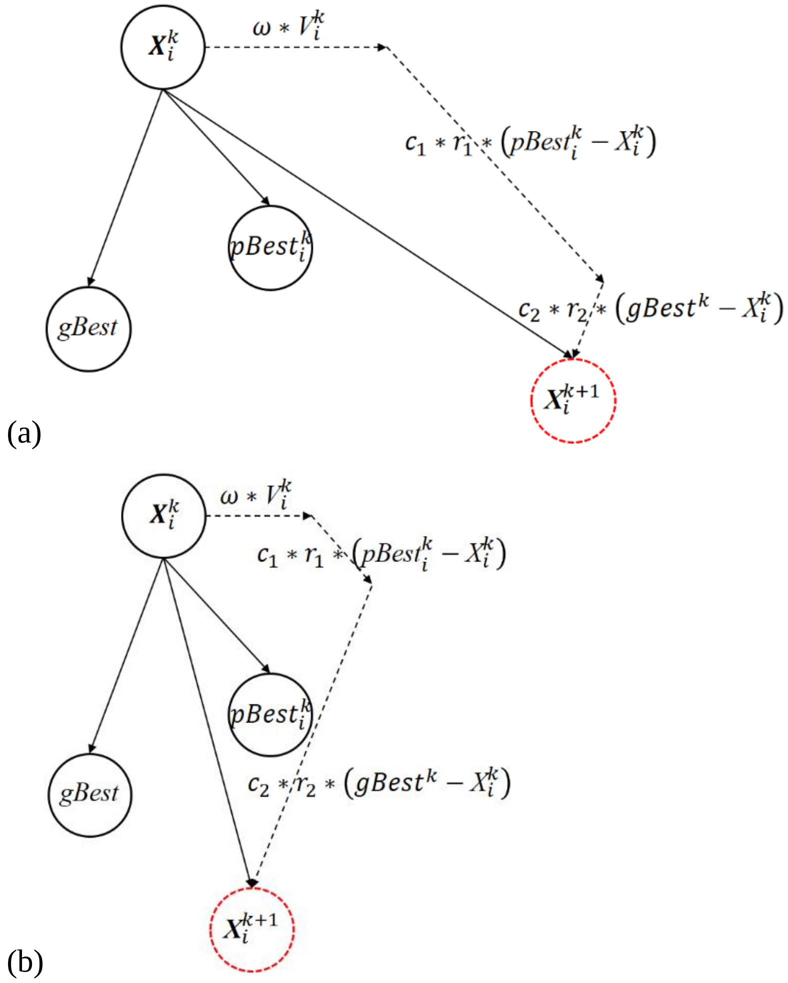


Fig. 3 Search methods of particles: (a) exploration; (b) exploitation

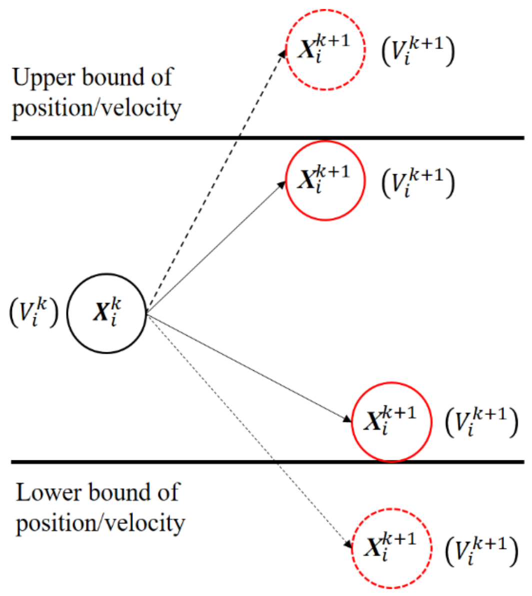


Fig. 4 Absorbing wall boundary condition

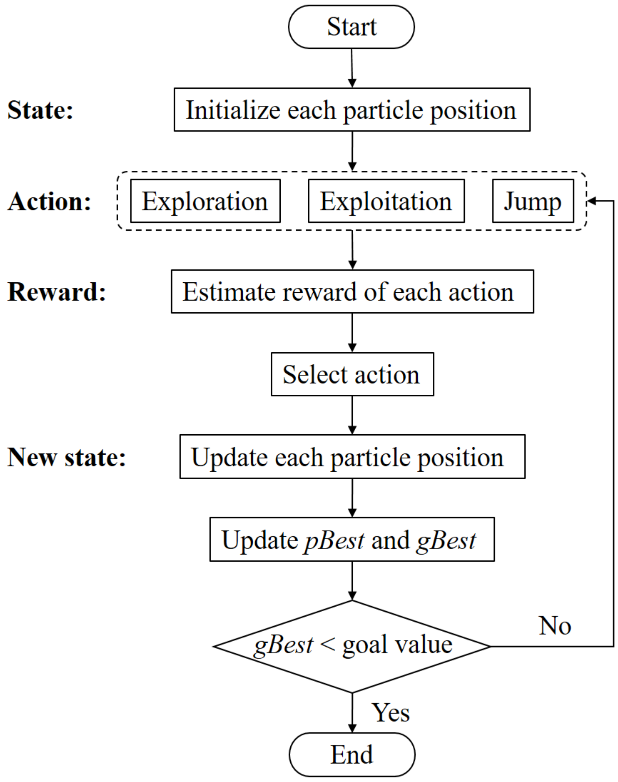


Fig. 5 Framework of proposed DQN-based PSO optimizer

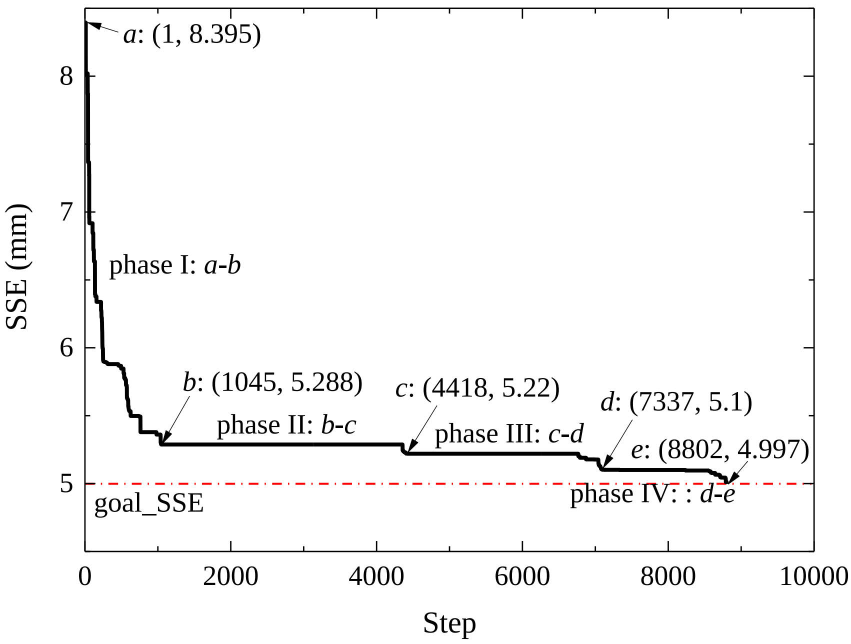


Fig. 6 Evolution of SSE value in a typical episode

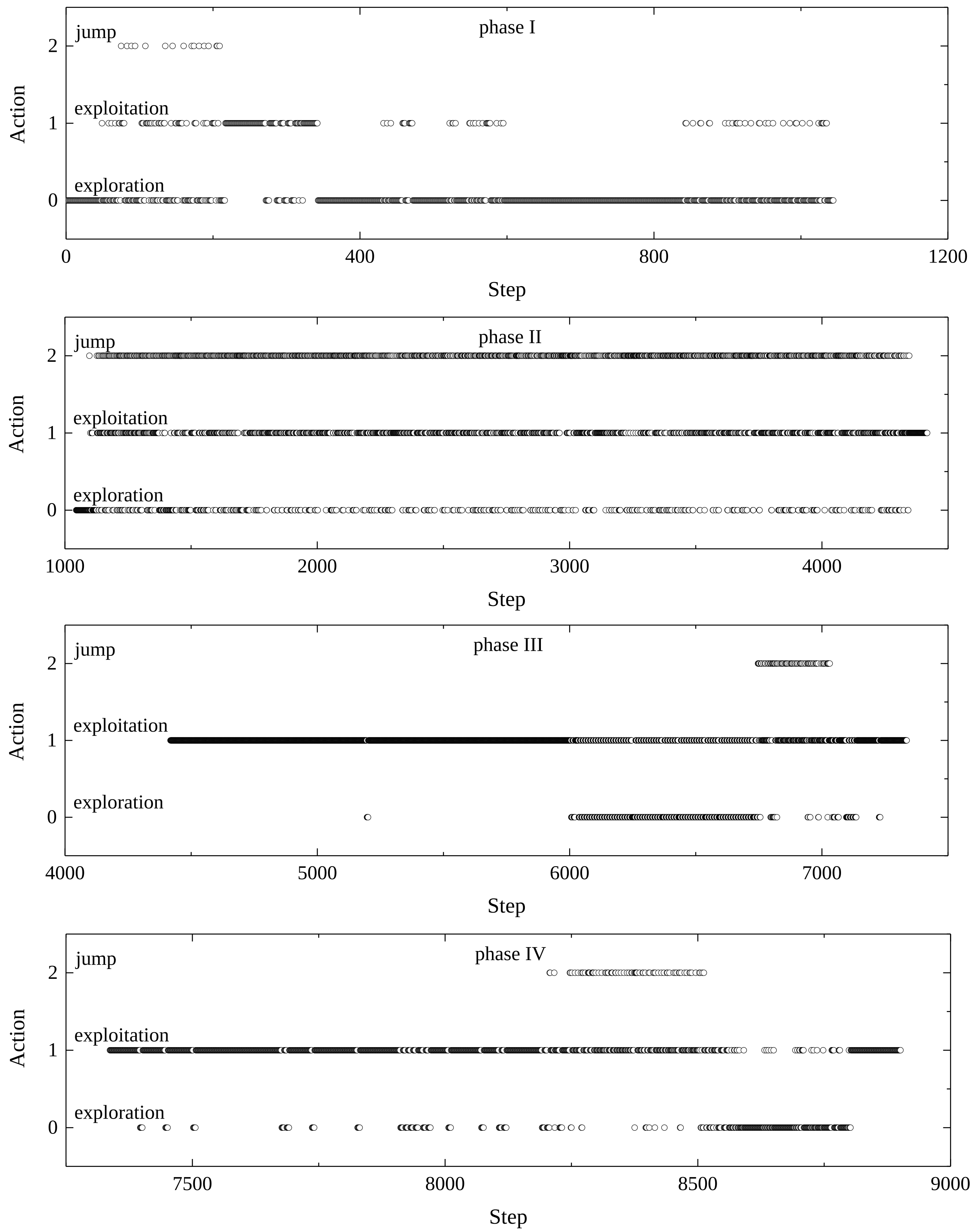


Fig.7 Actions at four phases

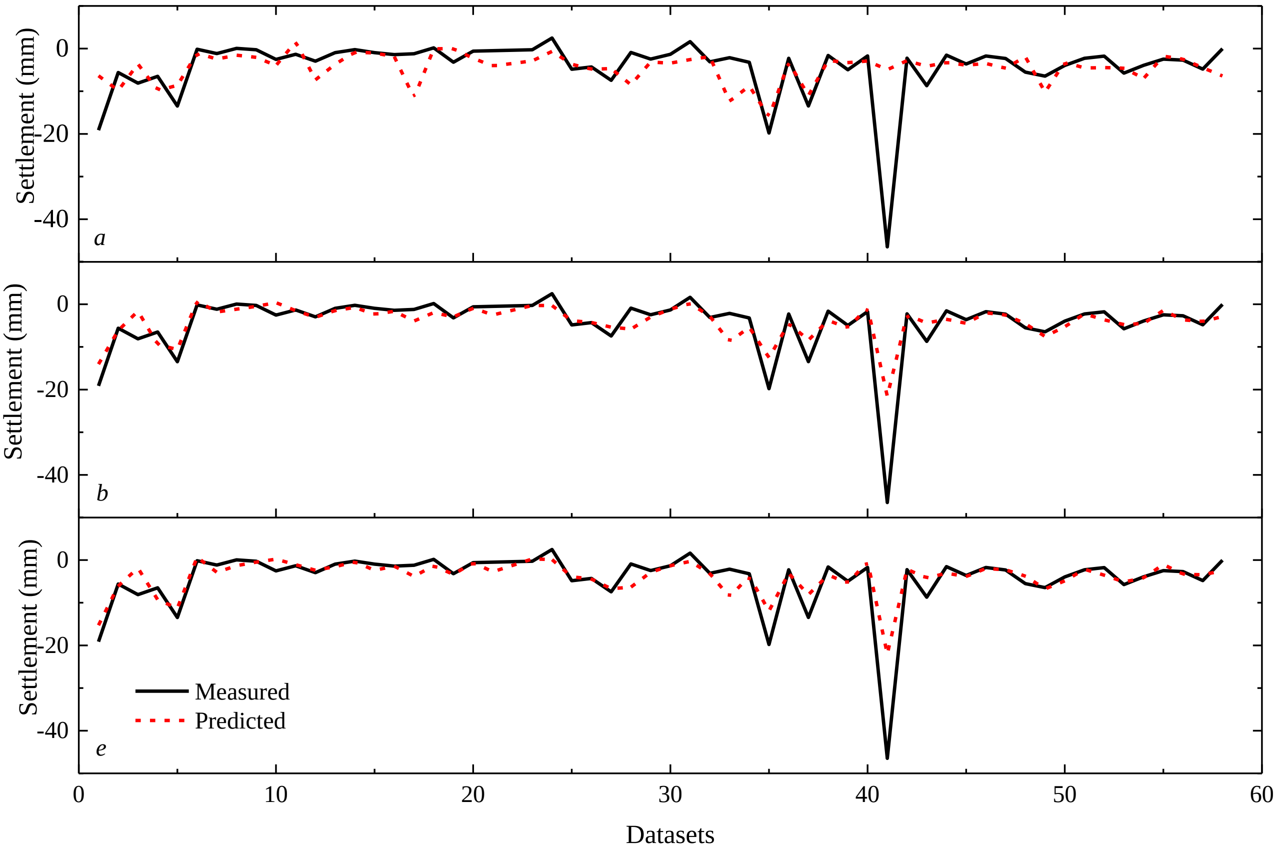


Fig. 8 Predicted settlement for the test set using the hybrid deep RL model generated at three steps

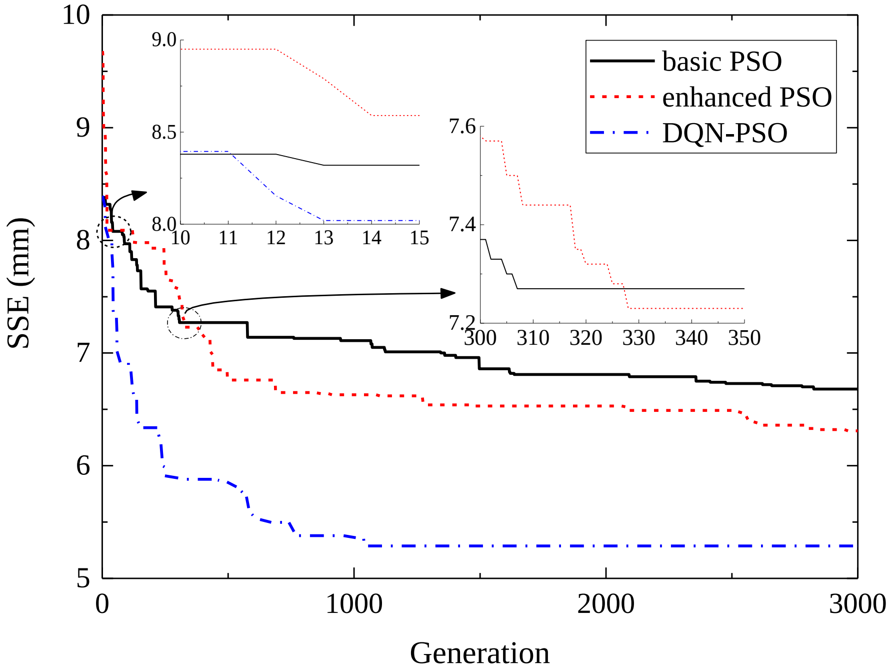


Fig. 9 Comparison of DQN-PSO optimizer with basic and enhanced PSO optimizers

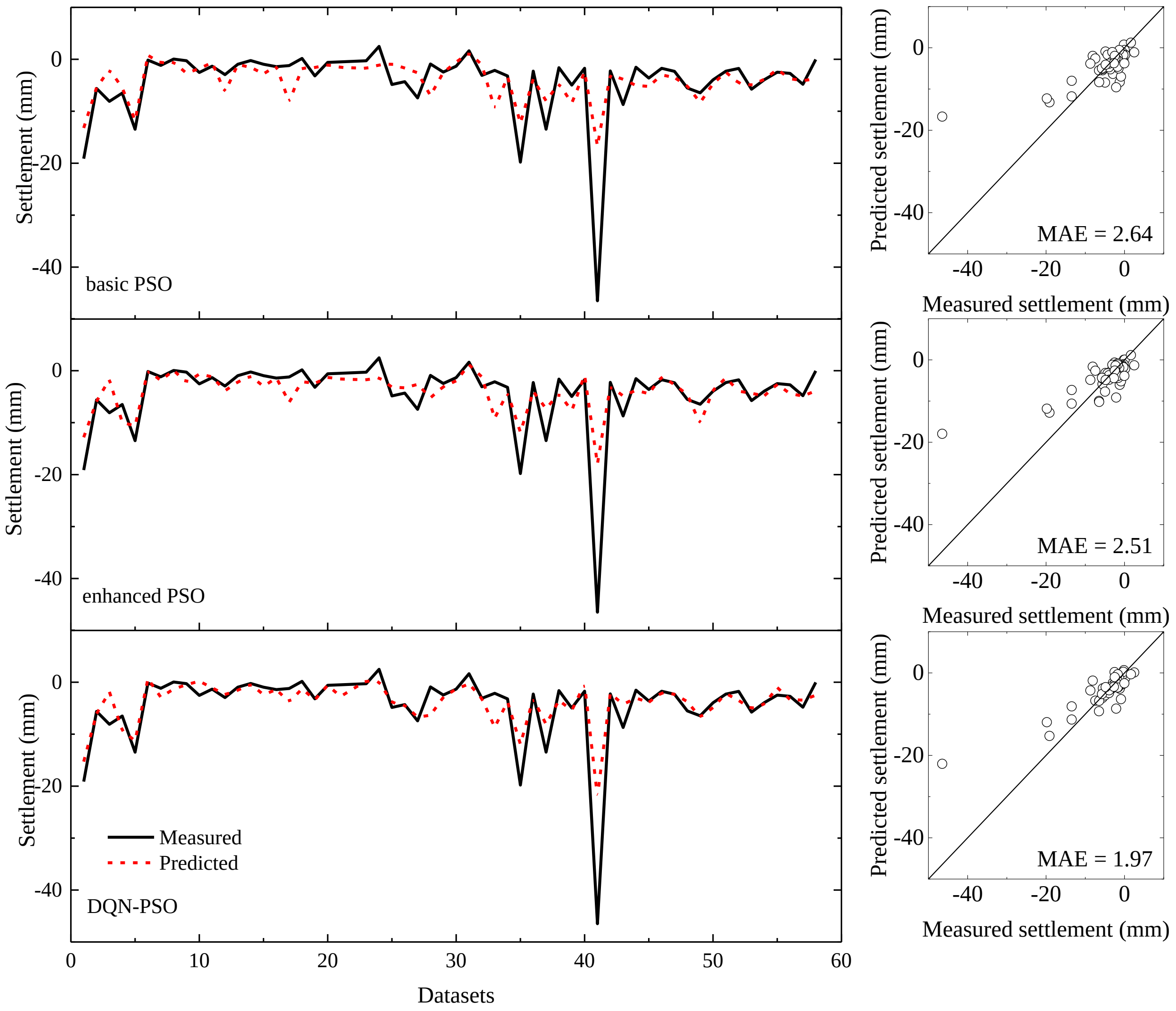


Fig. 10 Predicted settlement for the test set using ELM-based prediction models optimized by three optimizers

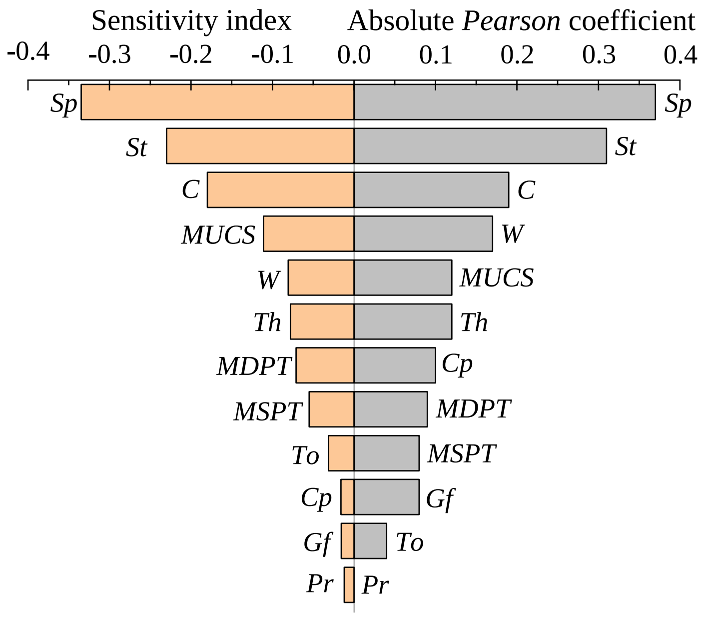


Fig. 11 Comparison between sensitivity indices and correlation coefficients of input parameters

Clean version Click here to view linked References

Clean Version

Reinforcement Learning based Optimizer for Improvement of Predicting Tunneling-induced Ground Responses

Pin Zhang[1], Heng Li[2], Q. P. Ha[3], Zhen-Yu Yin[1], Ren-Peng Chen[4, 5*]

1. Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China
2. Department of Building and Real Estate, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China
3. Faculty of Engineering and IT, University of Technology Sydney, NSW 2007, Australia
4. College of Civil Engineering, Hunan University, Changsha 410082, China
5. MOE Key Laboratory of Building Safety and Energy Efficiency, Changsha 410082, China

Corresponding author: Ren-Peng Chen; Email: chenrp@hnu.edu.cn

Abstract: Prediction of ground responses is important for improving performance of tunneling. This study proposes a novel reinforcement learning (RL) based optimizer with the integration of deep-Q network (DQN) and particle swarm optimization (PSO). Such optimizer is used to improve the extreme learning machine (ELM) based tunneling-induced settlement prediction model. Herein, DQN-PSO optimizer is used to optimize the weights and biases of ELM. Based on the prescribed states, actions, rewards, rules and objective functions, DQN-PSO optimizer evaluates the rewards of actions at each step, thereby guides particles which action should be conducted and when should take this action. Such hybrid model is applied in a practical tunnel project. Regarding the search of global best weights and biases of ELM, the results indicate the DQN-PSO optimizer obviously outperforms conventional metaheuristic optimization algorithms with higher accuracy and lower computational cost. Meanwhile, this model can identify relationships among influential factors and ground responses through self-practicing. The ultimate model

can be expressed with an explicit formulation and used to predict tunneling-induced ground response in real time, facilitating its application in engineering practice.

Keywords: Tunnel; Ground response; Reinforcement learning; Extreme learning machine; Optimization

1. Introduction

Ground responses to shield machine tunneling is a sophisticated problem that is affected by tunnel geometry, shield machine operational parameters, geological conditions and anomalous conditions [1]. The development of a rigorous analytical solution for descripting tunneling-induced ground response is complicated, because tunneling process involves multi-disciplinary knowledge such as solid mechanics, fluid mechanics, thermodynamics and tribology. The initial analytical models were developed upon the homogenous elastic half-space theory [2, 3], in which soils are treated as an isotropic elastic material with a single layer. To consider the tunneling-induced plastic deformation, classical plasticity solutions for soil stresses and displacements [4] were obtained by assuming a cavity contraction in a linearly-elastic, plastic material with Mohr-Coulomb yielding and nonassociative flow, but this method is merely appropriate in the case that plastic zone does not extend to the ground surface [5]. Few analytical models can consider the effect of fluid-solid coupling on the ground responses, although tunneling process can cause remarkable change in flow regime and cause large ground subsidence. Kinematical effects of tunneling on the ground response have also be frequently reported [6, 7], but these phenomenological observations merely stated the effect of ground responses to these kinematical parameters. It means that an explicit model involving these parameters cannot be developed.

Existing analytical solutions merely account for limited influential factors and simulate ground responses in a simplistic manner [8]. Engineers prefer to apply empirical formulations, which were derived from numerous in-situ observations, to predict ground response due to their simplicity [9]. However, such phenomenological methods tend to be applicable to a specific engineering, because the influential factors such as soil types, construction methods and tunnel configuration are different for different tunneling projects. Numerical modelling methods such as finite element and discrete element have been extensively employed to investigate ground response to tunneling as the improvement in the software and hardware [10-12]. Such elaborate numerical models are able to simulate soil-shield machine interaction by considering numerous extrinsic and intrinsic factors such as the geological heterogeneity [12] and the shield machine operation [13]. Nevertheless, parameters of soil constitutive models need to be calibrated by numerous experimental tests and back analysis of parameters also requires considerable skills [14]. A problem that all engineers have to confront is how to timely predict ground responses to tunneling and mitigates potential risks. Considering the influential factors such as operational parameters and geological conditions vary frequently with the advance of shield machine, empirical, analytical and numerical methods obviously exist their own deficiency in capturing the ground responses to tunneling in real time.

To predict ground responses alone the whole tunnel alignment, machine learning (ML)-based surrogate models have recently been proposed to complement the deficiency of conventional methods. Such models have strong capability of identifying the nonlinear relationship between ground responses and various influential factors [15-17]. Prediction models are established offline by directly learning from the in-situ data and used to online prediction of ground response in real time with high accuracy. The current ML-based tunneling-induced ground response prediction models were developed upon quite limited datasets (within 1000 datasets), thereby the model architecture is not sophisticated (within 20 input variables). Researchers thus utilized metaheuristic optimization algorithms such as particle swarm optimization (PSO) to search the hyper-parameters and general parameters of these ML-based models [18].

PSO has been successfully used in many domains [19], but the original PSO primarily exists two issues: premature convergence and high computational cost. The premature convergence means that PSO tends to be trapped in the local optima at the beginning of the search process. Meanwhile the computational cost can increase dramatically with the increasing population size, although the diversity of swarm is beneficial to obtain global optima. To mitigate these issues, numerous researchers have preoccupied with enhancing PSO algorithm, such as modified PSO with adaptive parameters [20, 21], hybrid PSO [22, 23]. Nevertheless, which action should be chosen for particles effectively moving towards the best position and when should take this action are still a key challenge.

In this study, a more general-purpose PSO optimizer enhanced by reinforcement learning (RL) deep Q-network (DQN) is proposed. In the past several years, RL has driven impressive advances in artificial intelligence and rapidly extended their application scopes [24-27]. In particular, the models trained by DQN outperform human experts in Atari, Go and no-limit poker [28-30]. The most fundamental improvement is that deep RL algorithm does not rely on hand-crafted policy evaluation functions, compared with previous ML algorithms. The agent of deep RL interacts with environment and learn past experience like a human via self-playing, thereby continuously improve their performance. This success motivates us to propose a DQN-based PSO optimizer (DQN-PSO), in which agent guides particles to choose the optimum action at each generation and move towards the best position with the lowest computational cost. To the best knowledge of the authors, this is first work to combine RL algorithm DQN based optimizer to develop a global best ML based model for investigating ground responses to tunneling.

Hence, this study aims to develop an ELM-based ground response prediction model due to its fast calculation speed. The proposed DQN-PSO optimizer is used to optimize ELM for identifying the global best weights and biases. A case study is implemented for validating the prediction performance of the proposed hybrid model. The framework of hybrid ELM and DQN-PSO optimizer proposed in this study can be replaced by various ML and metaheuristic algorithms to explore various issues.

2. Literature review and methodology

2.1 Literature review

Machine learning (ML) is a subsection of artificial intelligence that imparts the system to automatically learn from the data without being explicitly programmed. ML algorithms have made a significant breakthrough with appreciable performance in many domains. They have been considered to be the best choice for discovering the intricate relationships among high-dimensional data [31]. Ground responses to tunneling is complicated with the coupled effects of intrinsic and extrinsic factors such as geological, geotechnical, geometric, shield operational and anomalous parameters, which brings huge difficulties to accurately predict tunneling-induced settlement. Moreover, tunneling is a dynamic process and its influential factors always change with the advance of shield machine, thereby the real time prediction of settlement is vitally important in engineering practice. Conventional empirical, analytical, numerical and physical modelling methods have their limitations and cannot predict soil-shield machine interaction in real time. ML algorithms provide a novel method to overcome this issue.

Since the first application of ANN to predict tunneling-induced settlement conducted by Shi et al. [32], various ML algorithms have been extensively used to predict soil-shield machine interaction in the last two decades. The most widely used ML algorithm is the ANN with the error backpropagation [33-39]. Meanwhile its variants such as general regression neural network [40], wavelet neural network [14] and radial basis function neural network [40] have gained popularity in predicting soil-shield machine interaction. In the last decade, the development of ML has experienced a course of blossom. Consequently, researchers have implemented various ML algorithms to predict tunneling-induced ground settlement such as extreme learning machine [41], adaptive neuro fuzzy inference system [42, 43], relevance vector machine [44], least-squares support-vector machine [45], random forest [46-48] and genetic expression programming [42].

The key of developing a ML-based settlement prediction model is to determine the values of hyper-parameters. In addition, the weights and biases also need to be determined for ANN and its variants. The commonly used methods for determining hyper-parameters involve trial and error, grid search and meta-heuristic algorithms [34, 39, 40]. The weights and biases of ANN-based models are generally determined using deterministic and stochastic optimization algorithms [18, 35]. Trial and error and grid search methods can only search the parameters in a limited space. Deterministic optimization algorithm such as gradient descend may be trapped into local optima. Stochastic algorithms suffer from premature convergence and high computational cost. The global best parameters are thus hard to be obtained by using such method. To this end, this study proposes a RL algorithm DQN based optimizer to search the global optimum parameters of ML algorithms with higher accuracy and lower computational cost.

2.2 Reinforcement learning

2.2.1 Framework of reinforcement learning

Reinforcement learning (RL) is originated from a discrete-time and finite *Markov decision process* (MDP). RL consists of a learning agent, an environment, states, actions, and rewards. The agent interacts with an environment at some discrete time scale, $t = 0, 1, \ldots$. On each time step t, the agent perceives or observes the state of the environment, S_t ($S_t \in S$), thereafter chooses a primitive action based on this perception or observation, A_t ($A_t \in A_{S_t}$). In response to each action, a, the environment thereafter produces a numerical reward, R_{t+1}, and changes to a next state, S_{t+1} ($S_{t+1} \in S$). The whole dynamic transition process can be mathematically expressed by:

$$\mathbb{P}_{ss'}^{a} = \Pr\left\{S_{t+1} = s' \middle| S_t = s, A_t = a\right\} \tag{1}$$

$$R_s^a = E\left\{R_{t+1} \middle| R_t = s, R_t = a\right\} \tag{2}$$

where $\mathbb{P}_{ss'}^{a}$ = state transition probability matrix; R_s^a = immediate reward.

Note that the action at a state is selected based on a policy, π.

$$\pi\left(a \middle| s\right) = P\left\{A_t = a \middle| S_t = s\right\} \tag{3}$$

Therefore, the objective of the learning agent is to learn a policy which maximizes the expected discounted future reward at each state after maps from states to probabilities of taking each available primitive action, as shown by:

$$\begin{aligned} V_\pi\left(s\right) &= E_\pi\left\{R_{t+1} + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots \middle| S_t = s\right\} \\ &= E_\pi\left\{R_{t+1} + \gamma V_\pi\left(S_{t+1}\right) \middle| S_t = s\right\} \\ &= \sum_{a \in A} \pi\left(s, a\right)\left[R_s^a + \gamma \sum_{s'} P_{ss'}^a V_\pi\left(s'\right)\right] \end{aligned} \tag{4}$$

where $\gamma \in (0, 1)$ = a discount factor, it denotes the reward from next states gradually decreases; v_π = state-value function under policy, π; $v_\pi(s)$ = value of the state, s, under policy, π.

The state-value function is the expected return starting from state, s, and then following policy, π. There is another value function that is the expected return starting from state, s, taking action a, and then following policy, π, which is termed as action-value function, as shown following:

$$\begin{aligned} Q_\pi\left(s, a\right) &= E_\pi\left\{R_{t+1} + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots \middle| S_t = s, A_t = a\right\} \\ &= R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_\pi\left(s'\right) \\ &= R_s^a + \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi\left(a' \middle| s'\right) Q_\pi\left(s', a'\right) \end{aligned} \tag{5}$$

The objective of value-based RL algorithms such as Q-learning and Sarsa is to determine the optimum state-value V^*(s) or action-value functions Q^*(s, a), as shown in Eq. [6]–[7]. This study also utilizes value-based RL algorithm to establish model.

$$V^*(s)=\sum_{a\in A}\left[R_s^a+\gamma\sum_{s'}P_{ss'}^aV^*(s')\right] \tag{6}$$

$$Q^*(s,a)=R_s^a+\sum_{s'\in S}P_{ss'}^a\max_{a'\in A}Q^*(s',a') \tag{7}$$

2.2.2 Deep Q network

In this study, a deep RL algorithm termed as DQN proposed by Mnih et al. [29] is used. Conventional RL algorithms generally utilize a Q table to store states and actions. The values in the Q table update continuously complying with Eq. [8] during the learning process [49] (see Fig. 1(a)), thereby they have thus been limited to certain conditions with finite and discrete states and actions.

$$Q(s,a)=Q(s,a)+\alpha\left[R_s^a+\gamma\max_{a'}Q(s',a')-Q(s,a)\right] \tag{8}$$

where α = learning rate.

A DQN-based agent can interact with an environment with continuous states, because a DNN can parameterize an approximate action-value function $Q(s, a; \theta_i)$ (see Fig. 1(b)). Nevertheless, the sequence of observations lead to a strong correlation among these observations, thereby the neural network-based action-value function may be unstable and even diverge [50], thereby an experience replay method is proposed [29]. In this method, agent's experience $e_t = (S_t, A_t, R_t, S_{t+1})$ at the time step t is stored in a replay memory pool D. DNN can be trained based on mini-batches (s, a, r, s') ~ U(D) that are randomly drawn from the memory pool, which is beneficial to eliminate strong correlations among observations and ensures that the learning system is stable. The corresponding loss function of DNN is:

$$L_i(\theta_i)=\mathbb{E}_{(s,a,r,s')\sim U(D)}\left[R_s^a+\gamma\max_{a'}Q(s',a';\theta_i^-)-Q(s,a;\theta_i)\right] \tag{9}$$

where θ_i = parameters of Q-DNN at the ith iteration; θ_i^- = parameters of target DNN at the ith iteration. Note that only the parameters of Q-DNN are updated in real time. Target DNN is a forward network and has a same architecture with Q-DNN. The update of parameters θ_i^- is achieved by directly extracting parameters from Q-DNN at a fixed interval. In this way, training can avoid falling into feedback loops and proceed in a more stable manner.

2.3 Particle swarm optimization

Particle swarm optimization (PSO) is a metaheuristic optimization algorithm [51], which is developed based on simulating search behavior and social interaction of animals such as fish school and bird flock. PSO consists of several populations of particles and each particle is represented by its position vector $\boldsymbol{X}_i^k$, velocity vector $\boldsymbol{V}_i^k$ and fitness value. The velocity and position of each particle are updated using the following equations:

$$\boldsymbol{V}_i^{k+1} = \omega * \boldsymbol{V}_i^k + c_1 * r_1 * \left(\boldsymbol{pBest}_i^k - \boldsymbol{X}_i^k\right) + c_2 * r_2 * \left(gBest^k - \boldsymbol{X}_i^k\right) \tag{10}$$

$$\boldsymbol{X}_i^{k+1} = \boldsymbol{X}_i^k + \boldsymbol{V}_i^{k+1} \tag{11}$$

where k = current generation, i = ith particle; ω = inertia weight; c_1, c_2= cognitive and social acceleration coefficients; r_1, r_2 = random numbers within the range [0, 1] complying with uniform distribution; $\boldsymbol{pBest}_\mathrm{i}$ = local best location of the ith particle; $gBest$ = global best location of all particles. The predominant objective of the PSO algorithm is to find the optimum fitness value and the corresponding location.

2.4 Extreme learning machine

Extreme learning machine (ELM) is a modification of the single-hidden layer feedforward neural network, that is, only one hidden layer in this algorithm. The weights of input layer and the biases of hidden layer are assigned randomly. The optimum ELM is obtained by calculating the weights of the hidden layer and the biases of the output layer, thereby the calculation speed is much faster. In general, ELM can be represented as:

$$\mathbf{H} = f\left(\mathbf{X}\mathbf{w} + \boldsymbol{b}\right) \tag{12}$$

$$\left\|\mathbf{H}\boldsymbol{\beta} - \boldsymbol{y}\right\| = 0 \tag{13}$$

where $\mathbf{w}$ = weight matrix of the input layer; $\boldsymbol{b}$ = bias vector of the hidden layer; $\mathbf{H}$ = hidden layer output matrix; $\boldsymbol{\beta}$ = weight vector connecting the hidden nodes and the output nodes; $\boldsymbol{y}$ = outputs; f = activation function. The optimum ELM algorithm can be achieved by minimizing the value of $\left\|\mathbf{H}\boldsymbol{\beta} - \boldsymbol{y}\right\|$. Detailed description of ELM algorithm can refer to Huang et al. [52].

3. Introduction of proposed model

3.1 Proposed ELM-based ground response prediction model

3.1.1 Feature selection

Recent work by Zhang et al. [48] demonstrated that the influential factors of tunneling-induced settlement can be mainly classified into four categories: tunnel geometry, geological condition, shield operational parameters and anomalous conditions. In detail, twelve parameters are vitally important to soil-tunnel interaction including one tunnel geometry factor (cover depth of tunnel *C*, it should be noted that cover depth of tunnel is the only geometric factor used in this study considering the tunnel specification along the whole section is consistent), five shield operational parameters (thrust *Th*, torque *To*, grout filling *Gf*, penetration rate *Pr*, chamber pressure *Cp*), five geological parameters (modified blow counts of standard penetration test of soil layers *MSPT*, modified blow counts of dynamic penetration test of soil layers *MDPT*, modified uniaxial compressive strength of weathered rocks *MUCS*, groundwater table *W* and soil type at the cutterhead face *St*) and one anomalous condition (shield stoppage *Sp*). This study still adopts these twelve parameters for developing ELM-based ground response prediction model. Herein, five shield operational parameters and *Sp* can be collected in real time during tunneling process. The remaining geological and geometric parameters can be obtained during site investigation and route design process, which are conducted before the construction of tunnel. Therefore, tunnel-induced settlement can be predicted in real time.

3.1.2 Model architecture

The framework of the ELM-based ground response prediction model is presented in Fig. 2. Input layers have 12 neurons corresponding 12 input variables as mentioned above and ground maximum settlement S is the only output variable. ELM based model with different number of hidden neurons was pre-trained for selecting an appropriate framework. Considering the focus to this study is to highlight the superiority of proposed optimizer in the next section, the detailed processing for determine the optimum number of hidden neurons are not presented for brevity. The results indicate the performance of model is not sensitive to the hyper-parameters (the number of hidden neurons) when the number of hidden neurons exceed 15. Considering the computational cost and the model performance, 20 hidden neurons are ultimately adopted in this study. The training of ELM-based model can be obtained by:

$$\mathbf{H} = f_E\left(\mathbf{Xw} + \boldsymbol{b}\right); \quad \boldsymbol{\beta} = \mathbf{H}^{\dagger}\boldsymbol{y} \tag{14}$$

where $\mathbf{X}$ = input matrix (n×12, n is the number of datasets); $\mathbf{H}$ = output of hidden layer (n×20); $\boldsymbol{y}$ = output vector (n×1); $\mathbf{w}$ = weights matrix (12×20); $\boldsymbol{b}$ = bias vector (1×20); $\mathbf{H}^{\dagger}$ is obtained by *Moore–Penrose generalized inverse* of matrix H [53] (20×n), because $\mathbf{H}$ is a nonsquare matrix; $\boldsymbol{\beta}$ = ultimate training result (20×1); f_E is an activation function used in the hidden layer of ELM, and *sigmoid* activation function is adopted in this study, which can be expressed by:

$$f_E\left(x\right) = \frac{1}{1 + e^{-x}} \tag{15}$$

3.2 Proposed deep reinforcement learning-based optimizer

3.2.1 States and actions

The novel optimizer is developed based on the integration of deep reinforcement learning algorithm DQN and meta-heuristic optimization algorithm PSO (DQN-PSO). The search space of population represents the environment of DQN, and positions of all particles represent the state of DQN. Three actions, i.e., exploration, exploitation and jump are considered in this study, as shown following:

(*i*) Exploration: in PSO, ω, c_1 and c_2 control the movement direction and scale of particles. At the early state of generation, particles tend to make a large movement to explore the search space and move far away from the current *gBest*. Therefore, ω and c_1 values are large, and c_2 value is small, as shown in Fig. 3(a). This operation is termed as exploration, and the update of each particle position and velocity complies with Eqs. [10]–[11].

(*ii*) Exploitation: at the later stage of generation, particles tend to make a small movement to slowly converge at *gBest* and avoid heavy vibration. Therefore, ω and c_1 values are small, and c_2 value is large, as shown in Fig. 3(b). This operation is termed as exploitation, and the update of each particle position and velocity complies with Eqs. [10]–[11].

(*iii*) Jump: the former two actions achieve the adaptive adjustment of parameters in PSO, but the algorithm is still likely to be trapped in the local optima and cannot jump out this status. Therefore, a jump action is assigned to the action space, which can be obtained by:

$$\boldsymbol{X}_i^{k+1} = \boldsymbol{pBest}_i^k + r * \left(X_{\max} - X_{\min} \right) \tag{16}$$

where r = random number within the range [–1, 1] complying with uniform distribution; $X_{\max}$ and $X_{\min}$ = upper and lower bound of particles location. This new location update method allows particles to jump out the local optima.

In this study, ε-greedy strategy is employed to select an action. The action that can generate maximum reward according to the results of Q-DNN is selected with probability(1–ε), but the action is randomly selected from all available actions for exploring unknown conditions with probability ε.

3.2.2 Boundary conditions

There are four boundary conditions in PSO, i.e., reflecting wall, damping wall, invisible wall, and absorbing wall. Absorbing wall is used to limit the position and velocity of particles in this study. As shown in Fig. 4, there are upper and lower bound for the position and velocity vectors. When they exceed this boundary condition, the position and velocity of each particle are reset as the values of upper or lower bounds, respectively (see Eqs. [17]–[18]). Otherwise, the update of position and velocity vector complies with Eqs. [10]–[11] and [16].

$$\boldsymbol{X}_i^{k+1} = \begin{cases} X_{\max}, & if \ \boldsymbol{X}_i^{k+1} > X_{\max} \\ X_{\min}, & if \ \boldsymbol{X}_i^{k+1} < X_{\max} \end{cases} \tag{17}$$

$$\boldsymbol{V}_i^{k+1} = \begin{cases} V_{\max}, & if \ \boldsymbol{V}_i^{k+1} > V_{\max} \\ V_{\min}, & if \ \boldsymbol{V}_i^{k+1} < V_{\max} \end{cases} \tag{18}$$

3.2.3 Basic framework

Fig. 5 presents the basic framework of the proposed optimizer DQN-PSO. This optimizer starts from creating several populations, which is also the current state of RL. The rewards of each action under this state will be estimated by the Q-DNN, thereafter the action which can create maximum reward under this state will be selected. The velocity and position will be updated based on the selected action, thereby the new state will be generated. After the ***pBest*** and *gBest* are updated, the search process completes if the *gBest* satisfies the termination condition. Otherwise, the whole process will repeat.

3.3 Enhanced PSO optimizer

To validate the superiority of the proposed RL-based optimizer DQN-PSO, an enhanced optimizer is also developed for comparison. This enhanced PSO has two characteristics:

(*i*) Adaptive accelerator parameters: as mentioned above, particles start from exploring in the search space and thereafter transfer to exploitation operation with the increase of generations. Therefore, a search strategy that c_1 decreases linearly and c_2 increases linearly with the increase of generations has been developed [54]. This strategy can improve the global search capability of PSO at the early stage and local optimization capability at the later stage, as shown following:

$$c_1^k = c_{1\_initial} + \frac{k}{t}\left(c_{1\_final} - c_{1\_initial}\right) \tag{19}$$

$$c_2^k = c_{2\_initial} + \frac{k}{t}\left(c_{2\_final} - c_{2\_initial}\right) \tag{20}$$

where k = current generation; t = a total of generation; c_1^k, c_2^k = values of c_1 and c_2 at the kth generation, respectively; $c_{1\_initial}$, $c_{2\_initial}$ = initial values of c_1 and c_2, respectively; c_{1_final}, c_{2_final} = final values of c_1 and c_2, respectively. The update of each particle position and velocity complies with Eqs. [10]–[11].

(*ii*) Jump: unlike the DQN-PSO optimizer, enhanced PSO optimizer cannot intelligently select the action of particles based on the reward of each action. Therefore, when the number of generations exceeds a critical value (Eq. [21]) and the difference of the objective function outputs generated at the adjacent time steps is less than a threshold value (Eq. [22]), a jump operation is activated. Thereafter the update of each particle position in the jump operation complies with Eq. [16].

$$k > N_J \tag{21}$$

$$f_{obj}^{k+1} - f_{obj}^{k} \le f_J \tag{22}$$

where k = current generation; f_{obj}^{k+1}, f_{obj}^{k} = output of objective function at the kth and (k+1)th generations, respectively; N_J, f_J = thresholds for the number of generations and the difference of adjacent outputs of the objective function, respectively.

3.4 Proposed hybrid deep RL model

3.4.1 Reward rule and actions selection

Note that the reward rule is not demonstrated in the former section, which needs to be determined by the hybrid model. As mentioned in the description of ELM, the training process of ELM is merely completed by computing the solution of a linear system. After the hyper-parameters (the number of hidden neurons) of ELM are determined, the model performance depends heavily on the weights and biases. To improve the performance of ELM-based ground response prediction model, the weights and biases of ELM are optimized by the DQN-based optimizer. In this hybrid algorithm, the state of the DQN-based optimizer represents the weights and biases of ELM, as shown in followings:

$$\mathbf{X} = \left[\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{i-1}, \boldsymbol{X}_i, \ldots, \boldsymbol{X}_n\right]_{n\times m} \tag{23}$$

$$\boldsymbol{X}_i = \left[x_1, x_2, \ldots, x_{i-1}, x_i, \ldots, x_m\right]_{1\times m} \tag{24}$$

where $\mathbf{X}$ = an aggregate of all populations; $\boldsymbol{X}_i$ = a single population. i. e, an aggregate of weights and biases of ELM; n = the size of population; m = the number of particles in each population, i.e. the number of weights and biases in ELM (20×12+20=260, as mentioned in section 3.1.2); x_i = a single particle of a population. Therefore, the number of particles in each state is 260n.

The objective function of the DQN-PSO optimizer is determined by the sum of squared errors (SSE), which is used to evaluate the reward value.

$$\mathrm{SSE} = \sum_{i=1}^{n}\left[\beta_i * f_E\left(\omega_i x_i + b_i\right) - y_i\right]^2 \tag{25}$$

where y_i = actual settlement; $\beta_i * f_E\left(\omega_i x_i + b_i\right)$ = predicted settlement using the ELM-based model, in which parameters β_i, ω_i and b_i derive from $\boldsymbol{\beta}$, $\mathbf{w}$ and $\boldsymbol{b}$ (see section 3.1.2), respectively; x_i = one set of input variables. The updates of ***pBest*** and *gBest* are related to the SSE value, in which all particles move towards the positions with low values of SSE. The reward rule is that the final reward *r* is 1 if the SSE yielded by *gBest* is less than the prescribed goal value, otherwise, the current model acquires reward of 0. Note that the exact rewards are only known at the end of each episode.

3.4.2 Model framework

The pseudocode of the proposed hybrid ELM-based prediction model and DQN-PSO optimizer is presented in Algorithm 1. It can be observed that the hybrid algorithm involves prescribed number of episodes. In each episode, states are updated continuously until the SSE value yielded by the *gBest* can satisfies the termination condition.

Algorithm 1: Hybrid DQN and ELM algorithm

---

1. step = 0
2. **for** episode in range (number of prescribed episodes)
3. Randomly initialize state *s*
4. **while** True:
5. Estimate rewards of each action and choose an action *a*
7. Modify the current state *s* to *s_* by taking *a* and the corresponding *reward*
8. Store [*s*, *a*, *reward*, *s_*] in the replay memory pool
9. **if** step satisfies the update condition
10. Update target DNN
11. **if** *gBest* generated by state *s_* satisfies the termination conditions:
12. Evaluate the reward of this episode
13. **break**

14. step = step + 1

15. Exit

4. Application of proposed model

4.1 Overview of case study

In order to investigate the feasibility and reliability of the proposed hybrid DQN-PSO optimizer and ELM-based tunneling-induced ground responses prediction model, an in-situ experiment conducted by Zhang et al. [48] on a practical tunneling project from Changsha city, China is used in this study. This experimental zone consisted of five tunnel sections with six metro stations. A total of 5.44 km was constructed using earth pressure balanced (EPB) shield machine (construction starting in 2016 and completing in 2019). The tunnel was primarily excavated in the weathered rocks, which means that the consolidation settlement completed rapidly after the tunnel was constructed. Therefore, this case study focused on the tunneling-induced ultimately steady ground settlement. Each monitoring cross-section of settlement was positioned at a fixed interval of around 10 m.

With regard to the collection of datasets, the geological conditions and geometric factor at each ring were obtained by the site investigation before tunneling process. The five operational parameters were recorded per minute by the shield machine data acquisition system, and the average operational parameters at each ring were preprocessed. The ground settlement monitoring points were installed at an interval of 10 m and was measured twice a day. The settlement of monitoring points and the 12 input variables at the corresponding positions were stored in the database for training ELM-based ground response prediction model, thereby synchronousness between the settlement data and the input variables can be guaranteed. The database used in this study can be downloaded in the Appendix section.

4.2 Results

Table 1 presents the values of parameters used in all algorithms in this study. The experimental results indicate the model performance is not particularly sensitive to the architecture of target DNN and Q-DNN. The number of hidden layers in the target DNN and Q-DNN is 1, and the corresponding number of neurons is 15. Q-DNN starts to training when the agent's experiences in the replay memory pool D reaches 200 and it is trained with an interval of 5 time steps. The size of mini-batch used for training Q-DNN is 32, and the parameters of the target DNN are updated with an interval of 300 time steps. A total of 100 game episodes are carried out by the intelligent agent. The computational results indicate the ELM-based prediction model showcases great performance with the *SSE* value of 5, thereafter the decrease in the goal value will lead to a dramatic increase in the computational cost and may fail to reach the goal value. Therefore, the goal value of SSE is ultimately defined as 5 in this study.

Fig. 6 presents the evolution of SSE value generated by the hybrid deep RL prediction model in a typical episode. It can be observed that this episode consumes 8802 steps to reach the goal SSE value. The whole evolution of SSE can be categorized into four phases according to the characteristics of SSE variation. At the phase I from a to b, SSE value experiences a remarkable decrease from 8.395 at the 1st step to 5.288 at the 1045th step. Thereafter, the change in the SSE value is not discernable, but three steady phases can be obviously observed. The first steady phase (phase II: $b – c$) continues a total of 3373 steps, followed by phase III ($c – d$) with 2919 steps and phase IV ($d – e$) with 665 steps.

The advantage of the RL algorithm DQN is that it can reveal the intelligent operation mechanism of the agent, while other ML-based models merely run as a black box. To investigate the operation mechanism of the DQN-PSO optimizer, the actions at four phases are presented, as shown in Fig. 7. At the phase I, it can be observed that the agent focuses on the exploration at the initial stage, implying this action can receive the largest reward based on the Q-DNN results. The performance of the hybrid deep RL prediction model at the initial stage is not steady, thereby the optimization of weighs and biases can easily improve the prediction accuracy, which complies with the obvious decrease in the SSE value (see Fig. 6). At the earth stage of phase II, the agent still starts from exploration, but this action cannot reduce the SSE value, thereby the agent transfers to conduct the exploitation action, and sometimes conduct the jump action for jumping out local optima. Consequently, the exploitation and jump actions alternately appear and dominate this phase. At the phase III, the agent focuses on the exploitation, because the action trials at the phase II cannot cause large decrease in the SSE value (see Fig. 6). It indicates the performance of the hybrid deep RL prediction model is roughly steady, and SSE will converge at a fixed value. Similar condition can also be observed at the phase IV, where exploitation still dominate this phase. SSE value varies within an acceptable range and end up with the prescribed goal value, thereby it is reasonable to deduce the optimum hybrid deep RL prediction model for predicting tunneling-induced ground response is obtained. The consistency of agent's action and the corresponding model performance at each phase ensures the reasonability of the hybrid deep RL prediction model. The agent like a human intelligently guides particles to choose the optimum action at each generation and move towards the best position.

To clearly reveal the evolution of the prediction performance of model, the predicted maximum settlement for the test set using the hybrid deep RL prediction model generated at three typical steps *a*, *b* and *e* are presented, as shown in Fig. 8. It can be seen that the predicted settlement using the model generated at the step *a* severely deviates from the measured settlement. It cannot accurately capture the evolution of tunneling-induced settlement and loses fidelity at some monitoring points, e.g. the largest settlement of 48 mm is not detected. The performance of model generated at the step *b* improves dramatically. The predicted evolution of settlement shows great agreement with the measured settlement.

Meanwhile all of large settlement that exceeds 10 mm can be detected by this model, which is of great significance for avoiding risks in engineering practice. The performance of model generated at the step *e* is further refined with the lower SSE value, compared with the model generated at the step *b*. In detail, the difference of predicted and measured settlements at some monitoring points further reduces and shows better consistency with the measured evolution of ground maximum settlement.

5. Discussion

5.1 Compared with basic and enhanced PSO

To validate the superiority of the proposed RL-based optimizer DQN-PSO, a comparison among three optimizers, that is, basic PSO, enhanced PSO and DQN-PSO, is conducted. Fig. 9 presents the results of ELM-based ground responses prediction model optimized by three optimizers. The evolution of SSE value within 3000 generations is presented because three optimizers roughly converged at a fixed value. It can be observed that DQN-PSO obviously outperforms the basic and enhanced PSO with the lowest value of SSE and fastest convergence. The corresponding maximum generation of three types of optimizers when SSE values virtually converge at a constant value is presented in Table 2. In detail, the SSE value optimized by the DQN-PSO starts to be less than the basic and enhanced PSO when the number of generations exceeds 10, because the DQN-PSO optimizer always guides particles selecting a correct action. Meanwhile the whole optimization process virtually completes at around the 1000th generation with SSE value of 5.288, thereafter the objective of search operation is merely for achieving the prescribed goal value of SSE and the computational cost is expensive. It can be seen from Fig. 9 that the computational cost for decreasing SSE value from 5.288 to the prescribed goal value is appropriately seven times the figure for decreasing SSE value from the initial value to 5.288. It is noteworthy that the enhanced PSO also outperform the basic

PSO with lower value of SSE from the 326th generation. Enhanced PSO indeed further optimizes the search trajectory of particles to a certain extent, but the key challenges including which action should be chose and when should take this action are still dodged. It means that the enhanced PSO cannot avoid being trapped in the local optima, thereby the decrease in the convergence SSE value is not discernable, compared with the basic PSO. The basic PSO conducts the exploration action throughout the whole optimization process, thereby it is easy to be trapped in the local optima. The premature convergence problem is obvious, because the SSE value roughly maintains constant when the number of generations reaches 500.

Fig. 10 presents the evolution of ground responses for the test set predicted by the ELM-based prediction model optimized by three optimizers as well as the MAE values computed using Eq. [26]. It can be seen that the hybrid deep RL model outperforms ELM-based prediction models optimized by PSO and enhanced PSO. Enhanced PSO slightly refines the predicted settlement evolution with a slight decrease in the MAE value (from 2.64 to 2.51). The improvement in the prediction performance of the hybrid deep RL model is remarkable, in which the MAE value decreases to 1.97. The great agreement between the predicted and measured evolution of settlement and the improvement in recognizing maximum settlement is observed. Meanwhile all datasets are closer to the line with the slope of 1. Hence the tunneling-induced ground responses prediction model can be established using the hybrid deep RL algorithm.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|r_i - p_i\right| \tag{26}$$

where r = measured settlement; p = the predicted settlement; n = a total number of datasets.

5.2 Sensitivity analysis

To evaluate the performance of the proposed ELM based settlement prediction model optimized by DQN-PSO. Global sensitivity analysis (GSA) is conducted to reveal how model output uncertainty can be apportioned to the uncertainty in each input variable [55]. Variance-based GSA method has been extensively used in main domains [56-58], thereby it is used in this study. In this method, the total order index S_{Ti} in the variance-based GSA method measures the effect of an input parameter and its coupled effect with other input parameters on the model output. The calculation of S_{Ti} proposed by Jansen [59] is adopted in this study, and the detailed formulations are not presented for brevity, which can refer to Zhang [60]. The results of GSA are shown in Fig. 11, compared with the correlation coefficients which are calculated by absolute *Pearson* coefficients (see Eq. [27]). It can be observed that the parameters that have strong correlations with settlement (*Sp*, *St*, *C*) still have higher impact on the ELM based model. *Th* with the highest *Pearson* value among five operational parameters is also the most important operational parameter in the ELM based model. *Pr* with the lowest *Pearson* value is also the most insignificant parameter in the ELM based model. The rank of other parameters merely has a slight variation. Such factors indicate the ELM based model optimized by DQN-PSO obviously captures the potential correlations between the input and output parameters. The generalization ability and the practicability of such model can thus be guaranteed.

$$R = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2}} \tag{27}$$

6. Conclusions

The contribution of this study is that a hybrid deep reinforcement learning (RL) model which integrates extreme learning machine (ELM) and deep RL algorithm deep-Q network (DQN) is proposed for predicting tunneling-induced ground responses in real time, in which the relationships among influential factors and ground response were explored through self-practicing. Another contribution is that the proposed optimizer DQN-PSO knows which action should be conducted and when should take this action, thereby ensures the global optima to be obtained. Unlike previous metaheuristic optimization algorithms that guide the movement of particles in a rough manner, the reward rule of the DQN-based optimizer focuses on evaluating the reward of agent's action, hence particles like an intelligent human always select the optimum action at each step. To authors' best knowledge, this is the first work on using hybrid RL algorithm DQN and ML algorithm ELM to investigate tunneling-induced ground responses. The following conclusions can be drawn, based on the results of this work:

(1) Because DQN-PSO optimizer is able to guide particles to implement optimum action at each step, the global optima can be acquired when the value of objective function converges at a fixed value. In other words, the DQN-PSO optimizer can search the global best weights and biases of ELM with higher accuracy and lower computational cost, compared with basic or enhanced metaheuristic optimization algorithms.

(2) The hybrid deep RL model with the integration of ELM and DQN-PSO optimizer can accurately predict tunneling-induced ground response in real time, overcoming the deficiency of empirical, analytical and numerical models established by domain experts. The ultimate ELM based model can be expressed with an explicit formulation, which is user-friendly in engineering practice. Meanwhile, the performance of prediction model can be improved with the increase in the datasets collected from the field construction.

(3) The hybrid deep RL model is genetic, which means that it can be used to various situations with different states, actions, rules, rewards and objective function defined by domain experts without any debugging. Meanwhile the basic meta-heuristic and machine learning algorithms used in the hybrid deep RL model

can be randomly replaced based on different situations. Such model offers a pragmatic and reliable framework to develop a data-driven or physical model.

Appendix

The database used in this study can be download at following link:

https://www.researchgate.net/publication/336208927_Database_for_maximum_settlement_collected_from_Changsha_Metro_Line_4_Liugoulong_to_Fubuhe_station

Acknowledges

This study is sponsored by the National Natural Science Foundation of China (No. 51938005) and the program of High-level Talent of Innovative Research Team of Hunan Province 2019 (No. 2019RS1030). The authors greatly appreciate these financial supports during this research.

References

[1] P. Zhang, H.-N. Wu, R.-P. Chen, T.H.T. Chan, Hybrid meta-heuristic and machine learning algorithms for tunneling-induced settlement prediction: A comparative study, Tunnell. Undergr. Space Technol., 99 (2020) 103383.

[2] C. Sagaseta, Analysis of undrained soil deformation due to ground loss, Géotechnique, 37 (1987) 301-320.

[3] A. Verruijt, J.R. Booker, Surface settlements due to deformation of a tunnel in an elastic half plane, Géotechnique, 48 (1996) 709-713.

[4] H.S. Yu, R.K. Rowe, Plasticity solutions for soil behaviour around contracting cavities and tunnels, Int. J. Numer. Anal. Met., 23 (1999) 1245–1279.

[5] F. Pinto, A.J. Whittle, Ground movements due to shallow tunnels in soft ground. I: analytical solutions, J. Geotech. Geoenviron., 140 (2014) 04013040.

[6] W. Broere, D. Festa, Correlation between the kinematics of a Tunnel Boring Machine and the observed soil displacements, Tunnell. Undergr. Space Technol., 70 (2017) 125-147.

[7] S. Suwansawat, H.H. Einstein, Describing settlement troughs over twin tunnels using a superposition technique, J. Geotech. Geoenviron. Eng., 133 (2007) 445-468.

[8] P. Zhang, Z.Y. Yin, R.P. Chen, Analytical and semi-analytical solutions for describing tunneling-induced transverse and longitudinal settlement troughs, Int. J. Geomech., (2020) in press.

[9] R.B. Peck, Deep excavations and tunneling in soft ground, Proceedings of 7th International Conference on Soil Mechanic and Foundation Engineering Mexico City, 1969, pp. 225–290.
[10] M. Jiang, Z.Y. Yin, Analysis of stress redistribution in soil and earth pressure on tunnel lining using the discrete element method, Tunnell. Undergr. Space Technol., 32 (2012) 251–259.
[11] P. Zhang, R.-P. Chen, H.-N. Wu, Y. Liu, Ground settlement induced by tunneling crossing interface of water-bearing mixed ground: A lesson from Changsha, China, Tunnell. Undergr. Space Technol., 96 (2020) 103224.
[12] X.-T. Lin, R.-P. Chen, H.-N. Wu, H.-Z. Cheng, Deformation behaviors of existing tunnels caused by shield tunneling undercrossing with oblique angle, Tunnell. Undergr. Space Technol., 89 (2019) 78-90.
[13] J. Ninić, S. Freitag, G. Meschke, A hybrid finite element and surrogate modelling approach for simulation and monitoring supported TBM steering, Tunnell. Undergr. Space Technol., 63 (2017) 12-28.
[14] A. Pourtaghi, M.A. Lotfollahi-Yaghin, Wavenet ability assessment in comparison to ANN for predicting the maximum surface settlement caused by tunneling, Tunnell. Undergr. Space Technol., 28 (2012) 257-271.
[15] P. Zhang, Z.-Y. Yin, Y.-F. Jin, T.H.T. Chan, A novel hybrid surrogate intelligent model for creep index prediction based on particle swarm optimization and random forest, Eng. Geol., 265 (2020) 105328.
[16] P. Zhang, Z.Y. Yin, Y.F. Jin, G.L. Ye, An AI-based model for describing cyclic characteristics of granular materials, Int. J. Numer. Anal. Met., (2020) 1-21.
[17] P. Zhang, Z.Y. Yin, Y.F. Jin, T. Chan, Intelligent Modelling of Clay Compressibility using Hybrid Meta-Heuristic and Machine Learning Algorithms, Geoscience Frontiers, (2020) in press.
[18] D.J. Armaghani, E.T. Mohamad, M.S. Narayanasamy, N. Narita, S. Yagiz, Development of hybrid intelligent models for predicting TBM penetration rate in hard rock condition, Tunnell. Undergr. Space Technol., 63 (2017) 29-43.
[19] Z.Y. Yin, Y.F. Jin, S.J. S, P.Y. Hicher, Optimization techniques for identifying soil parameters in geotechnical engineering: comparative study and enhancement, Int. J. Numer. Anal. Met., 42 (2017) 1-25.
[20] K.E. Parsopoulos, Parallel cooperative micro-particle swarm optimization: A master–slave model, Appl. Soft Comput., 12 (2012) 3552-3579.
[21] W.H. Lim, N.A. Mat Isa, Two-layer particle swarm optimization with intelligent division of labor, Eng. Appl. Artif. Intel., 26 (2013) 2327-2348.
[22] A. Gálvez, A. Iglesias, A new iterative mutually coupled hybrid GA–PSO approach for curve fitting in manufacturing, Appl. Soft Comput., 13 (2013) 1491-1504.
[23] S.Z. Zhao, P.N. Suganthan, Q.-K. Pan, M. Fatih Tasgetiren, Dynamic multi-swarm particle swarm optimizer with harmony search, Expert Syst. Appl., 38 (2011) 3735-3742.
[24] M.A. Lopes Silva, S.R. de Souza, M.J. Freitas Souza, A.L.C. Bazzan, A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems, Expert Syst. Appl., 131 (2019) 148-171.
[25] K. Zhang, H. Zhang, Y. Mu, S. Sun, Tracking control optimization scheme for a class of partially unknown fuzzy systems by using integral reinforcement learning architecture, Appl. Math. Comput., 359 (2019) 344-356.
[26] Y. Ding, L. Ma, J. Ma, M. Suo, L. Tao, Y. Cheng, C. Lu, Intelligent fault diagnosis for rotating

machinery using deep Q-network based health state classification: A deep reinforcement learning approach, Adv. Eng. Inform., 42 (2019).
[27] F. Hourfar, H.J. Bidgoly, B. Moshiri, K. Salahshoor, A. Elkamel, A reinforcement learning approach for waterflooding optimization in petroleum reservoirs, Eng. Appl. Artif. Intel., 77 (2019) 98-116.
[28] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of Go with deep neural networks and tree search, Nature, 529 (2016) 484-489.
[29] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature, 518 (2015) 529-533.
[30] Noam Brown, T. Sandholm, Superhuman AI for multiplayer poker, Science, (2019) 1-12.
[31] S. Dargan, M. Kumar, M.R. Ayyagari, G. Kumar, A survey of deep learning and its applications: a new paradigm to machine learning, Archives of Computational Methods in Engineering, (2019).
[32] J. Shi, J.A.R. Ortigao, J. Bai, Modular neural networks for predicting settlements during tunneling, J. Geotech. Geoenviron. Eng., 124 (1998) 389-395.
[33] C.Y. Kim, G.J. Bae, S.W. Hong, C.H. Park, Neural network based prediction of ground surface settlements due to tunnelling, Comput. Geotech., 28 (2001) 517-547.
[34] S. Suwansawat, H.H. Einstein, Artificial neural networks for predicting the maximum surface settlement caused by EPB shield tunneling, Tunnell. Undergr. Space Technol., 21 (2006) 133-150.
[35] O.J. Santos, T.B. Celestino, Artificial neural networks analysis of São Paulo subway tunnel settlement data, Tunnell. Undergr. Space Technol., 23 (2008) 481-491.
[36] A. Marto, M. Hajihassani, R. Kalatehjari, E. Namazi, H. Sohaei, Simulation of longitudinal surface settlement due to tunnelling using artificial neural network, International Review on Modelling and Simulations, 5 (2012) 1024-1031.
[37] A. Darabi, K. Ahangari, A. Noorzad, A. Arab, Subsidence estimation utilizing various approaches – A case study: Tehran No. 3 subway line, Tunnell. Undergr. Space Technol., 31 (2012) 117-127.
[38] R. Boubou, F. Emeriault, R. Kastner, Artificial neural network application for the prediction of ground surface movements induced by shield tunnelling, Can. Geotech. J., 47 (2010) 1214-1233.
[39] M. Hasanipanah, M. Noorian-Bidgoli, D. Jahed Armaghani, H. Khamesi, Feasibility of PSO-ANN model for predicting surface settlement caused by tunneling, Eng. Comput-Germany, 32 (2016) 705-715.
[40] R.P. Chen, P. Zhang, X. Kang, Z.Q. Zhong, Y. Liu, H.N. Wu, Prediction of maximum surface settlement caused by EPB shield tunneling with ANN methods, Soils Found., 59 (2019) 284–295.
[41] R.P. Chen, P. Zhang, H.N. Wu, Z.T. Wang, Z.Q. Zhong, Prediction of shield tunneling-induced ground settlement using machine learning techniques, Front. Struct. Civ. Eng., 13 (2019) 1363–1378.
[42] K. Ahangari, S.R. Moeinossadat, D. Behnia, Estimation of tunnelling-induced settlement by modern intelligent methods, Soils Found., 55 (2015) 737-748.
[43] D. Bouayad, F. Emeriault, Modeling the relationship between ground surface settlements induced by shield tunneling and the operational and geological parameters based on the hybrid PCA/ANFIS method, Tunnell. Undergr. Space Technol., 68 (2017) 142-152.
[44] F. Wang, B. Gou, Y. Qin, Modeling tunneling-induced ground surface settlement development using a

wavelet smooth relevance vector machine, Comput. Geotech., 54 (2013) 125-132.
[45] L.M. Zhang, X.G. Wu, W.Y. Ji, S.M. AbouRizk, Intelligent approach to estimation of tunnel-induced ground settlement using Wavelet Packet and Support Vector Machines, J. Comput. Civil. Eng., 31 (2017) 04016053.
[46] J. Zhou, X. Shi, K. Du, X.Y. Qiu, X.B. Li, H.S. Mitri, Feasibility of Random-Forest approach for prediction of ground settlements induced by the construction of a shield-driven tunnel, International Journal of Geomechanics, 17 (2016) 04016129.
[47] V.R. Kohestani, M.R. Bazargan-Lari, J. Asgari-marnani, Prediction of maximum surface settlement caused by earth pressure balance shield tunneling using random forest, Journal of AI and Data Mining, 5 (2017) 127-135.
[48] P. Zhang, R.P. Chen, H.N. Wu, Real-time analysis and regulation of EPB shield steering using Random Forest, Automat. Constr., 106 (2019) 102860.
[49] C.J.C.H. Watkins, P. Dayan, Q-Learning, Mach. Learn., 8 (1992) 279-292.
[50] Y. Yuan, Z.L. Yu, Z. Gu, Y. Yeboah, W. Wei, X. Deng, J. Li, Y. Li, A novel multi-step Q-learning method to improve data efficiency for deep reinforcement learning, Knowl-Based Syst., 175 (2019) 107-117.
[51] J. Kennedy, R. Eberhart, Particle swarm optimization, IEEE International Conference on Neural NetworksPerth, Australia, 1995, pp. 1942-1948.
[52] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: Theory and applications, Neurocomputing, 70 (2006) 489-501.
[53] C.R. Rao, S.K. Mitra, Generalized inverse of matrices and its applications, Wiley1971.
[54] L. Yang, H. Su, Z. Wen, Improved PLS and PSO methods-based back analysis for elastic modulus of dam, Adv. Eng. Softw., 131 (2019) 205-216.
[55] A. Saltelli, I.M. Sobol, About the use of rank transformation in sensitivity analysis of model output, Reliab. Eng. Syst. Safe., 50 (1995) 225-239.
[56] C.Y. Zhao, A.A. Lavasan, R. Hölter, T. Schanz, Mechanized tunneling induced building settlements and design of optimal monitoring strategies based on sensitivity field, Comput. Geotech., 97 (2018) 246-260.
[57] L.M. Zhang, X.G. Wu, H.P. Zhu, S.M. AbouRizk, Performing global uncertainty and sensitivity analysis from given data in tunnel construction, J. Comput. Civil. Eng., 31 (2017) 04017065.
[58] K.M. Hamdia, H. Ghasemi, X.Y. Zhuang, N. Alajlan, T. Rabczuk, Sensitivity and uncertainty analysis for flexoelectric nanostructures, Comput. Method Appl. M., 337 (2018) 95-109.
[59] M.J.W. Jansen, Analysis of variance designs for model output, Comput. Phys. Commun., 117 (1999) 35-43.
[60] P. Zhang, A novel feature selection method based on global sensitivity analysis with application in machine learning-based prediction model, Appl. Soft Comput., 85 (2019) 105859.

Table

Table 1 Values of parameters in three algorithms

Algorithm	Parameter	Value
ELM	Number of hidden neurons	20
PSO	ω (exploration | exploitation)	0.9 | 0.4
	c_1 (exploration | exploitation)	2.5 | 0.4
	c_2 (exploration | exploitation)	0.4 | 2.5
	$[V_{min}, V_{max}]$	[-3, 3]
	$[X_{min}, X_{max}]$	[-10, 10]
	Population size	20
	Maximum generation	3000
Enhanced PSO	c_1 (initial | final)	2.5 | 0.5
	c_2 (initial | final)	0.5 | 2.5
	N_J	2000
	f_J	0.01
DQN	Number of hidden neurons	15
	RL learn (criteria| step)	200| 5
	Target DNN update interval	300
	Batch size	32
	Episode	100
	goal_*SSE*	5
	Reward decay coefficient γ	0.9
	ε-greedy	0.1
	Learning rate α	0.01

Table 2 Comparison among three optimizers

Optimizer	Generation	SSE
Basic PSO	1497	6.860
Enhanced PSO	1280	6.540
DQN-PSO	1045	5.288

Figure caption

Fig. 1 Schematic view of reinforcement learning: (a) Q-learning; (b) deep Q-network

Fig. 2 Architecture of ELM-based ground response prediction model

Fig. 3 Search methods of particles: (a) exploration; (b) exploitation

Fig. 4 Absorbing wall boundary condition

Fig. 5 Framework of proposed DQN-based PSO optimizer

Fig. 6 Evolution of SSE value in a typical episode

Fig.7 Actions at four phases

Fig. 8 Predicted settlement for the test set using the hybrid deep RL model generated at three steps

Fig. 9 Comparison of DQN-PSO optimizer with basic and enhanced PSO optimizers

Fig. 10 Predicted settlement for the test set using ELM-based prediction models optimized by three optimizers

Fig. 11 Comparison between sensitivity indices and correlation coefficients of input parameters

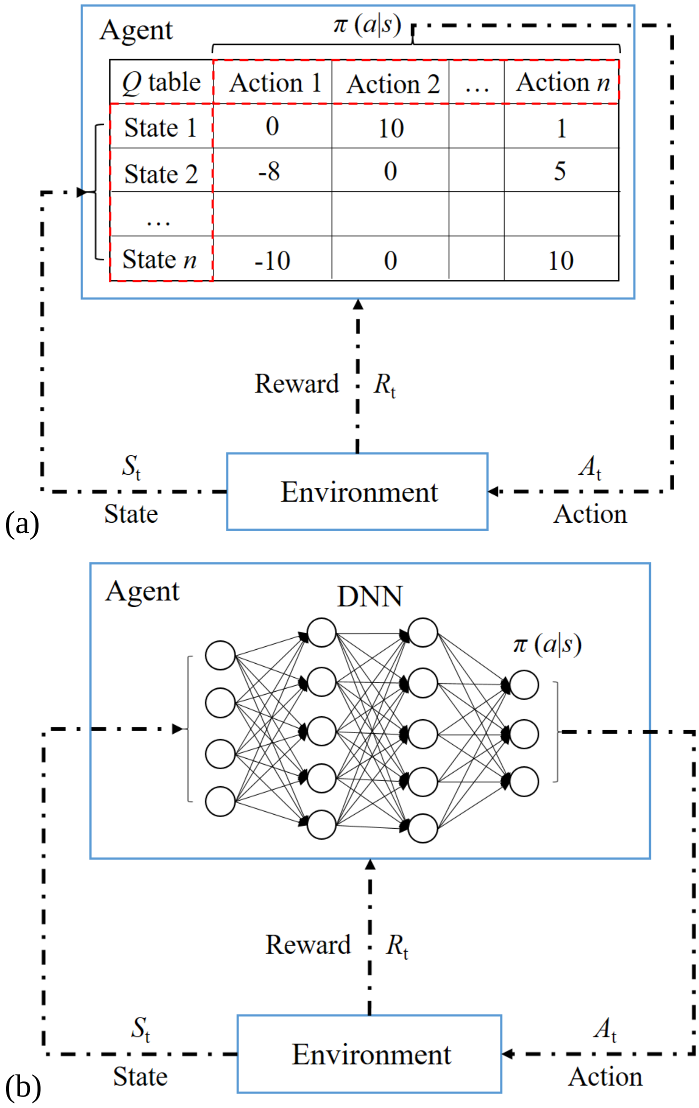


Fig. 1 Schematic view of reinforcement learning: (a) Q-learning; (b) deep Q-network

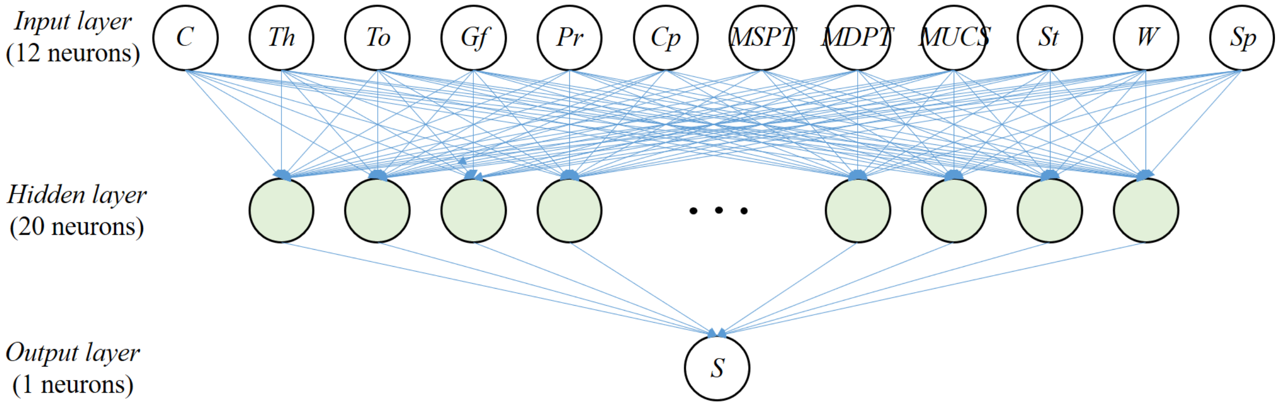
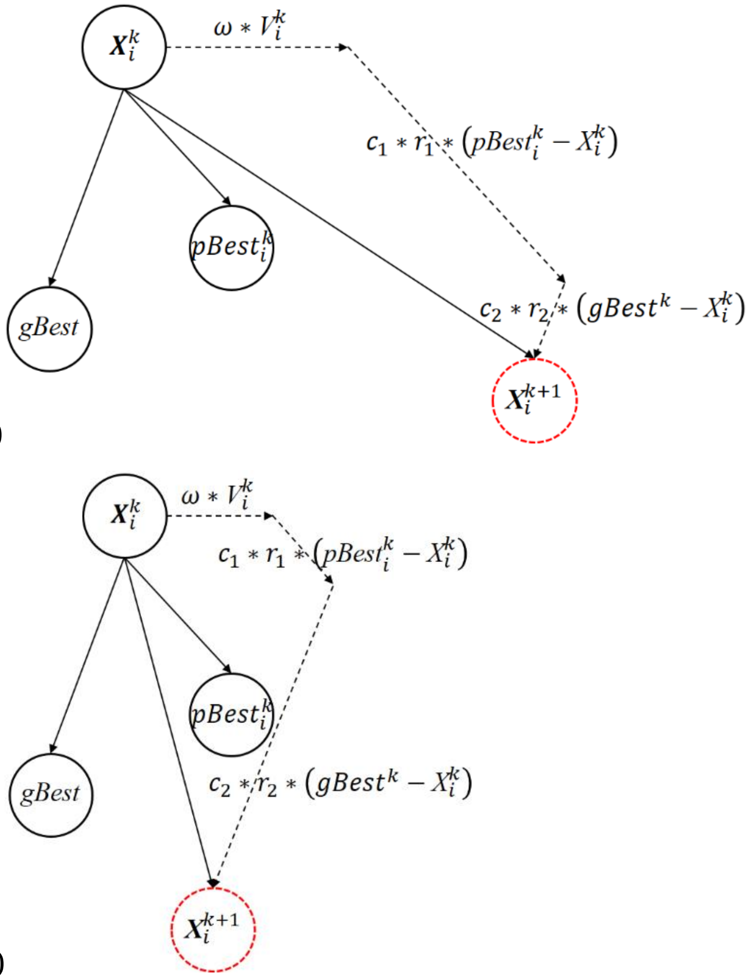


Fig. 2 Architecture of ELM-based ground response prediction model



(a)

(b)

Fig. 3 Search methods of particles: (a) exploration; (b) exploitation

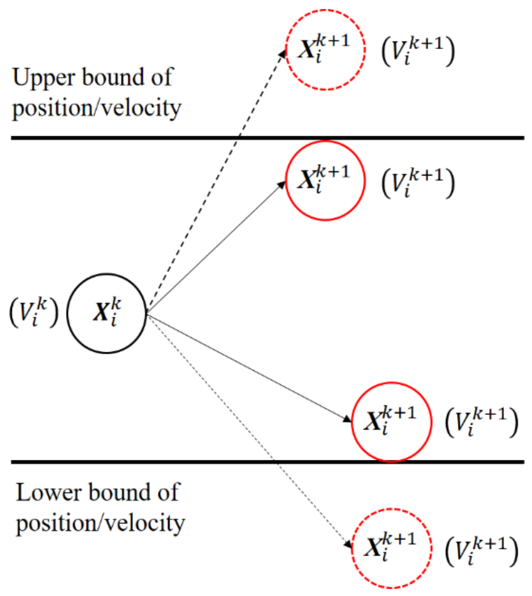


Fig. 4 Absorbing wall boundary condition

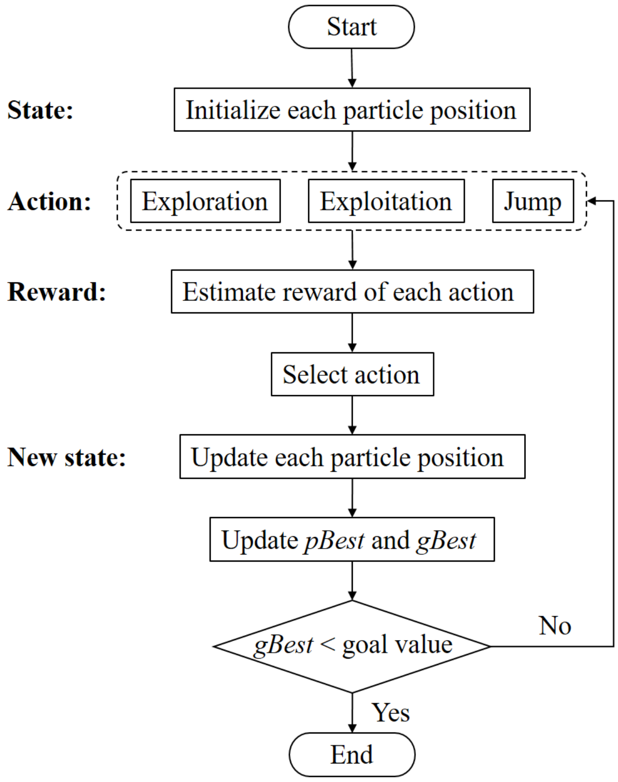


Fig. 5 Framework of proposed DQN-based PSO optimizer

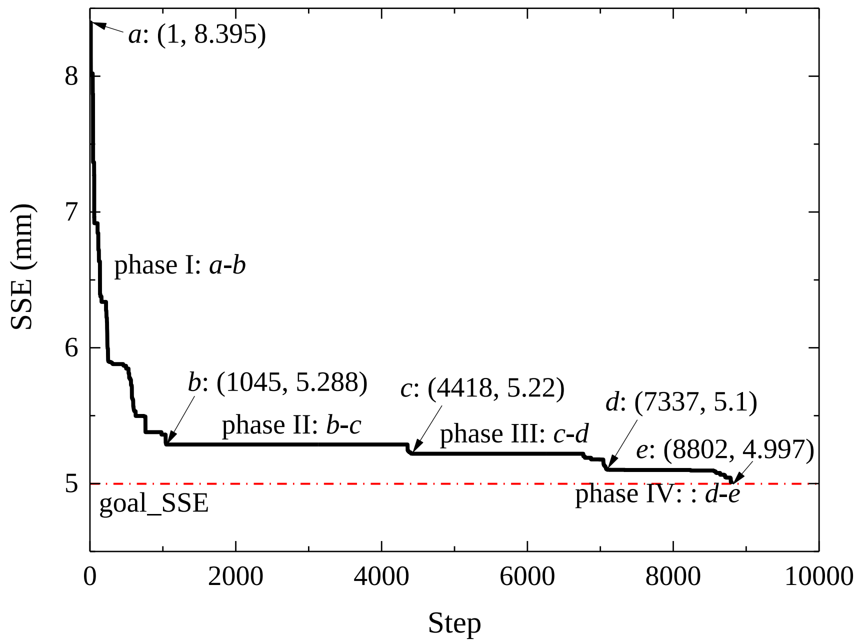


Fig. 6 Evolution of SSE value in a typical episode

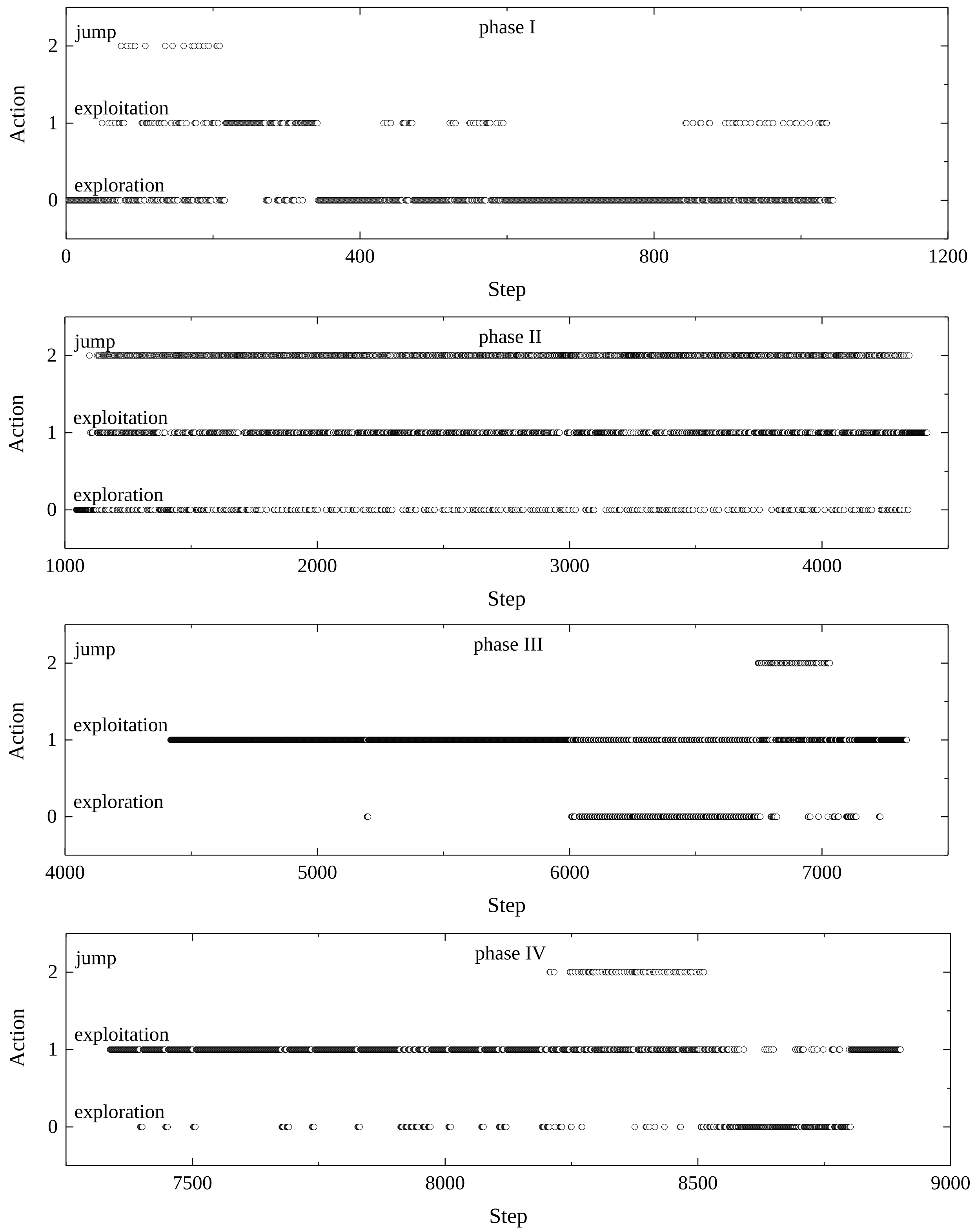


Fig.7 Actions at four phases

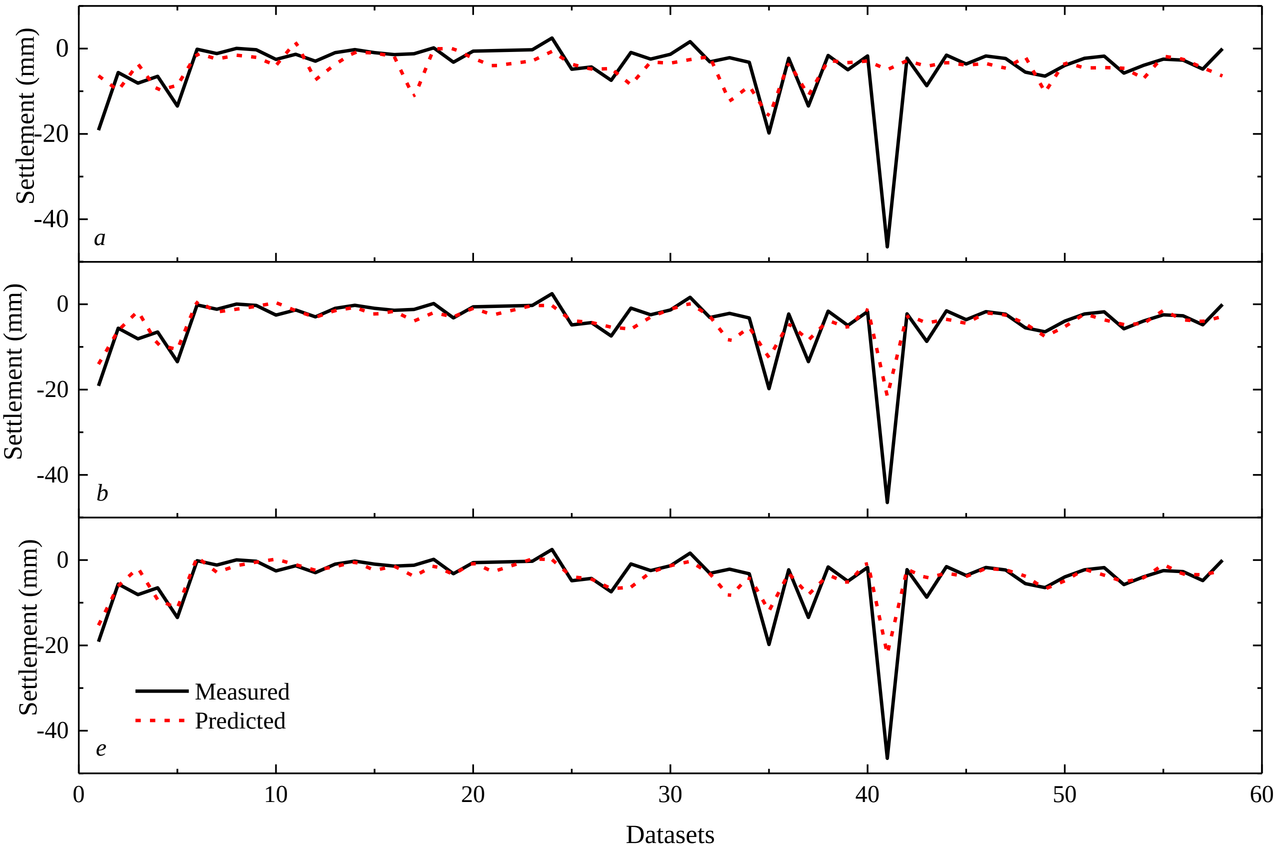


Fig. 8 Predicted settlement for the test set using the hybrid deep RL model generated at three steps

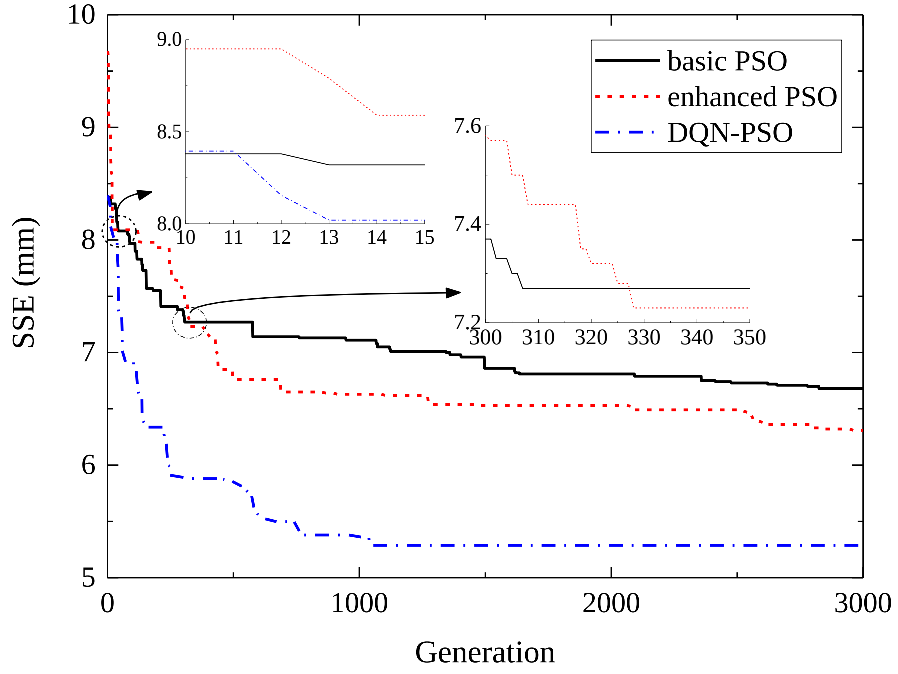


Fig. 9 Comparison of DQN-PSO optimizer with basic and enhanced PSO optimizers

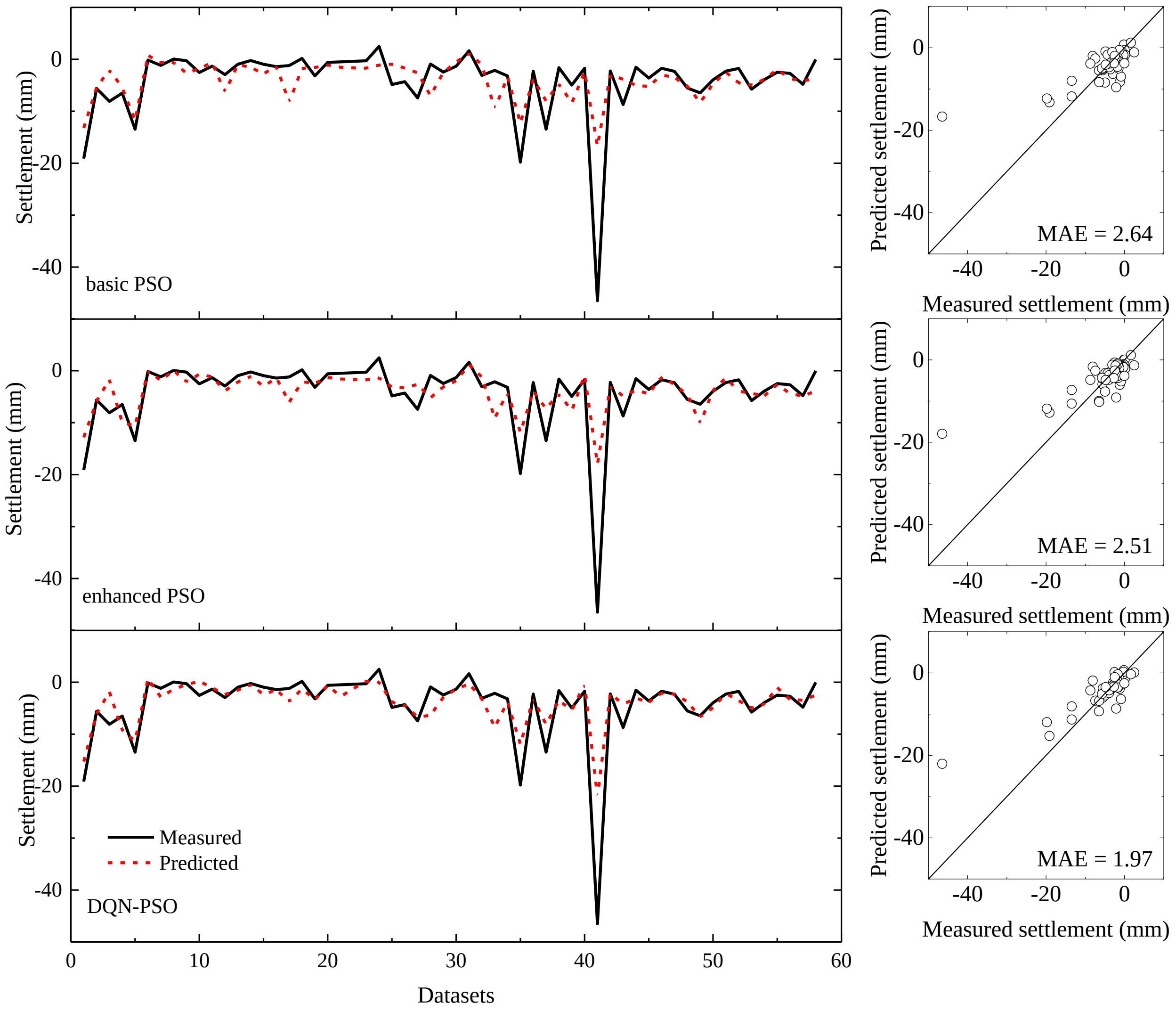


Fig. 10 Predicted settlement for the test set using ELM-based prediction models optimized by three optimizers

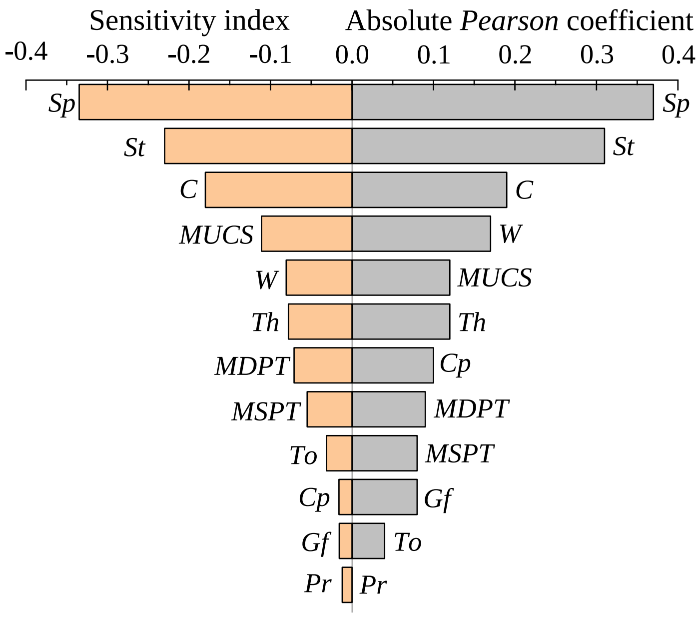


Fig. 11 Comparison between sensitivity indices and correlation coefficients of input parameters

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: