



Research article

Developing effective optimized machine learning approaches for settlement prediction of shallow foundation

Mohammad Khajehzadeh^{a,b}, Suraparb Keawsawasvong^{a,*}, Viroon Kamchoom^{c,**}, Chao Shi^d, Alimorad Khajehzadeh^e

^a Research Unit in Sciences and Innovative Technologies for Civil Engineering Infrastructures, Department of Civil Engineering, Faculty of Engineering, Thammasat School of Engineering, Thammasat University, Pathumthani, 12120, Thailand

^b Department of Civil Engineering, Anar Branch, Islamic Azad University, Anar, Iran

^c Excellent Centre for Green and Sustainable Infrastructure, Department of Civil Engineering, School of Engineering, King Mongkut's Institute of Technology Ladkrabang (KMUTL), Bangkok, 10520, Thailand

^d School of Civil and Environmental Engineering, Nanyang Technological University, 639798, Singapore

^e Department of Electrical Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran

ARTICLE INFO

Keywords:

Hybrid metaheuristic
Long short-term memory
Foundation settlement
Parameter optimization

ABSTRACT

The precise assessment of shallow foundation settlement on cohesionless soils is a challenging geotechnical issue, primarily due to the significant uncertainties related to the factors influencing the settlement. This study aims to create an advanced hybrid machine learning methodology for accurately estimating shallow foundations' settlement (S_m). The initial contribution of the current research is developing and validating a robust hybrid optimization methodology based on an artificial electric field and single candidate optimizer (AEFSCO). This approach is thoroughly tested using various benchmark functions. AEFSCO will also be used to optimize three useful machine learning methods: long short-term memory (LSTM), support vector regression (SVR), and multilayer perceptron neural network (MLPNN) by adjusting their hyperparameters for predicting the settlement of shallow foundations. A database consisting of 189 individual case histories, conducted through various investigations, was used for training and testing the models. The database includes five input parameters and one output. These factors encompassed both the geometric characteristics of the foundation and the properties of the sandy soil. The results demonstrate that employing effective optimization strategies to adjust the ML models' hyperparameters can significantly improve the accuracy of predicted results. The AEFSCO has increased the coefficient of determination (R^2) value of the MLPNN model by 9.3 %, the SVR model by 8 %, and the LSTM model by 22 %. Also, the LSTM-AEFSCO model is more accurate than the SVR-AEFSCO and MLPNN-AEFSCO models. This is shown by the fact that R^2 went from 0.9494 to 0.9290 to 0.9903, which is an increase of 4.5 % and 6 %.

1. Introduction

Shallow foundations are commonly used in construction to transfer the loads from a building or structure to the underlying soil or

* Corresponding author.

** Corresponding author.

E-mail addresses: ksurapar@engr.tu.ac.th (S. Keawsawasvong), viroon.ka@kmitl.ac.th (V. Kamchoom).

rock. This load causes the soil to compress, leading to settlement. Various factors contribute to the settlement, including the dimensions and depth of the foundation, soil characteristics, the weight of the structure, and groundwater conditions [1]. Settlement can be divided into two types: immediate settlement, also known as elastic settlement, and long-term settlement, often referred to as consolidation settlement. The elastic settlement occurs right after the foundation is constructed and is primarily caused by the elastic deformation of the soil. The load applied to the foundation compresses the soil, causing the foundation to settle to some extent. Long-term settlement happens gradually over an extended period after the construction is finished. It is mainly caused by soil consolidation, with the water being squeezed from the soil under load. This process can continue for years after construction and lead to further settlement [1].

Due to the high permeability of coarse-grained soils (sand and gravel), pore pressure quickly dissipates from these materials, and any elastic settlement brought on by a change in loading occurs immediately. Estimating settlement for shallow footings in granular soils is challenging due to the varying soil characteristics, including heterogeneity, density, grain distribution, etc. [2]. Due to this intricacy, numerous academics have made an effort to model the settlement utilizing various empirical and numerical techniques. It is possible to forecast settlement using methods like the plate load test (PLT), standard penetration test (SPT), and cone penetration test (CPT) [1]. The aforesaid reviews between predicted and measured settlement of shallow foundations in sandy soil [2–4] indicated that no particular method was persistently superior to the others in all cases and the calculated results of settlement were inconsistent. In addition, conventional analytical solutions are often developed under ideal/simple or homogeneous assumptions without considering interrelationships between different parameters and potential subsurface heterogeneities (e.g., density, SPT). Consequently, there remains a need for a more efficient method that can provide settlement prediction results with higher accuracy.

Recently, there has been a significant increase in the use of powerful machine learning (ML) and artificial intelligence (AI) methods for tackling complex non-linear and multivariate problems. These strategies possess outstanding characteristics and learning abilities for extracting features. Due to the evident advantages of these approaches, researchers conducted extensive research to implement ML techniques for solving various engineering issues. Table 1 provides a selection of the most recent publications that successfully integrate AI and ML techniques such as Gaussian processes (GP), deep neural networks (DNNs), decision tree algorithms (DT), and support vector machines (SVM) into engineering applications.

According to the mentioned merits of AI-based techniques, there has been an increasing trend in the adoption of these methods for precisely estimating the elastic settlement and bearing capacity of foundations. The majority of prediction models in this domain have been constructed using real-world data, with the aim of achieving satisfactory outcomes.

Shahin et al. [2] utilized ANNs to enhance the precision of settlement prediction for shallow foundations on cohesionless soils. The ANN model is developed and verified using a comprehensive database of real measured settlements. Shahnazari et al. [25] employed three novel evolutionary-based methodologies, including gene expression programming (GEP), classical genetic programming (GP), and evolutionary polynomial regression (EPR), to derive enhanced predictive settlement models. The models were created with an extensive data of case histories based on the SPT. The findings demonstrate that the GP-based models have the capability to accurately forecast the settlement of foundations on cohesionless soils. Rezaian and Javadi [26] presented the application of genetic programming (GP) to predict the settlement of footings. The predictions were based on a comprehensive database of SPT results from various case histories concerning quantified settlements of shallow foundations. Samui [27] employed the support vector machine (SVM) algorithm to forecast the settlement of foundations [27]. Jibanchand et al. [28] assess the viability of four ensemble learning models including Bagging, Adaptive Boosting, Random Forest, and Extreme Gradient Boosting in forecasting the settlement of spread foundations. The

Table 1
Engineering application of machine learning algorithms.

ML model	Application	References
convolutional neural networks	classification of engineering (CAD) models	[5]
combining an attention mechanism with a convolutional neural network	microseismic event recognition in rock engineering	[6]
deep neural network (DNN)	forecasting of bus traffic	[7]
convolutional neural network	model of feature learning for condition surveillance	[8]
deep neural network (DNN)	reduction of the compressive strength of foamed concrete	[9]
quasi-recurrent neural networks	remaining useful life (RUL) prediction of the engineering systems.	[10]
deep-gated recurrent neural network	prediction of petroleum production	[11]
recurrent neural network	forecasting wind speed	[12]
recurrent neural network	slope system dynamic response prediction	[13]
LSTM network	prediction of significant wave height	[14]
bidirectional LSTM network	aeroengine prediction interval estimation of remaining service life	[15]
support vector regression method	rock fracture toughness estimation	[16]
support vector machines	prediction of the impact of long-term temperature on concrete dam structural health monitoring	[17]
support vector regression method	forecasting the building energy consumption	[18]
gaussian process regression	building electricity usage	[19]
modified gaussian process regression	prediction of lithium-ion batteries' cycle capacity	[20]
Hybrid artificial neural networks and sine cosine algorithm	slope safety prediction	[21]
decision tree algorithm	supply fraud forecasting	[22]
gaussian process regression (GPR)	forecasting time and cost of tunnels construction	[23]
random forest	Risk estimation of deep foundation pits	[24]

analytical findings demonstrate that the Bagging and XGBoost models exhibit exceptional performance, outperforming that of other models [28]. Kalinli et al. [29] developed novel methodologies for ascertaining the ultimate bearing capacity of shallow foundations, utilizing ant colony optimization and artificial neural networks. The effectiveness of Gaussian process regression (GPR) as a soft computing technique to forecast the bearing capacity of soils under shallow foundations was investigated by Ahmad et al. [30]. To accurately determine the bearing capacity of ring foundations on rock masses, Kumar et al. [31] developed hybrid models that combine extreme learning machine (ELM) and least-square support vector machine (LSSVM) using two optimization approaches, namely particle swarm optimization (PSO) and Harris hawks optimization (HHO) and.

The ultimate outcome of the ML models is significantly influenced by the adjustment of their hyperparameters. Gradient descent optimization approaches are frequently employed for the purpose of optimizing the parameters of the existing ML models. However, a common issue with all of the aforementioned optimizers is the possibility of the iterative process converging to a local optimum rather than the global optimum. This might have a detrimental impact on the performance of the ML models [32].

The artificial electric field algorithm (AEFA) as a recently presented technique, simulates the dynamics of charged particles in the presence of the Coulomb force within an electrostatic field [33]. The AEFA derives advantages from its uncomplicated framework, limited adjustable parameters, and comprehensible algorithmic foundation. This technique has the potential to outperform other state-of-the-art approaches [33]. Nevertheless, similar to other methodologies, the AEFA approach is not exempt from some limitations, including the occurrence of local minima and the limited capabilities for exploitation [34]. In the present research, to mitigate the aforementioned limitations and enhance the precision of outcomes, a novel hybrid metaheuristic approach is introduced, integrating the artificial electric field algorithm (GBAEF) with the single candidate optimizer (AEFSCO). The hybrid AEFSCO combines the comprehensive global search capability of the AEFA with the localized search capability of the SCO. The AEFSCO technique is utilized for the purpose of fine-tuning the main parameters of the LSTM, SVR, and MLPNN models. Next, we proceed to test the precision of the recommended frameworks in estimating the settlement of shallow footings. The accuracy of the algorithms' predictions was assessed using a number of error metrics, most notably the MAE (mean absolute error), RMSE (root mean squared error), and R^2 (coefficient of determination).

Based on the explanations provided above, the study's primary contributions are as follows:

- 1 A hybrid artificial electric field with a single candidate optimizer (AEFSCO) is introduced.
- 2 Employing multiple benchmark functions, a comparative investigation is executed to verify the efficacy of the AEFSCO.
- 3 The suggested AEFSCO is employed to adjust the LSTM, SVR, and MLPNN hyperparameters.
- 4 The efficacy of the optimized ML models for forecasting the settlement of shallow foundations is examined, and the most powerful model is recommended.

2. Research methodology

2.1. Overview of the employed ML models

2.1.1. Long short-term memory network (LSTM)

The LSTM is a recurrent neural network (RNN) architecture that distinguishes itself from traditional RNNs by its ability to effectively capture long-term relationships while maintaining short-term memory through the utilization of specialized memory cells [14]. This approach overcomes the limitations of traditional recurrent neural networks by using memory cells and gate mechanisms. By employing long-term memory, the LSTM network increases learning capacity and is extremely effective at making more accurate predictions with multivariate data. The core idea of LSTMs is their ability to selectively retain and forget information over time. They achieve this through three main components, as demonstrated in Fig. 1. There are three gates: an input gate (it), an output gate (ot), and a forget gate (ft) that make up the LSTM, and they control the inside-out flow of information in the cell state (see Fig. 1). In this figure, C_{t-1} and h_{t-1} are the memory and output from prior block, C_t is the memory from current block, and x_t and h_t are the input and output of the network.

The input gate manages the information flow into the memory cell by deciding which parts of the incoming data should be stored. It utilizes a sigmoid activation function (σ) on inputs from the current and previous time steps, producing a value between 0 and 1. This

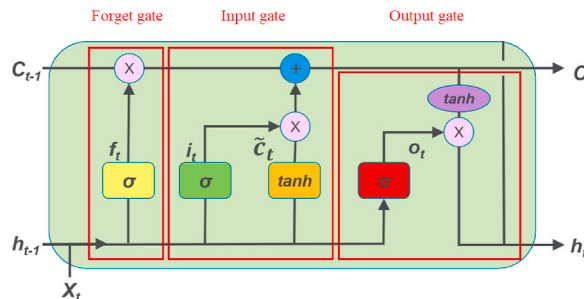


Fig. 1. The architecture of the LSTM Network.

value indicates the significance of the input for storage in the memory cell.

The forget gate allows the selective erasure of information from the memory cell. By applying a sigmoid activation function to inputs from the current and previous time steps, it generates a value between 0 and 1. This value signifies the extent to which information should be forgotten in the memory cell. A value close to 0 implies forgetting, while a value near 1 indicates retention.

The output gate controls the portions of the memory cell's content to be used as the LSTM's output at the present step. It takes input from the current and previous time steps and applies a sigmoid activation function. Additionally, a tanh activation function is applied to the current input, resulting in a candidate value vector. The output gate combines the output of the sigmoid activation function with the candidate values, yielding the final output of the LSTM.

The purpose of these gates is to selectively discard irrelevant information, as determined by Eq. (1), while retaining and transferring relevant information from the earlier to the next loop, as described by Eq. (2). Additionally, these gates generate an output, as specified by Eq. (3).

$$f_t = \sigma(W_{f,x} * X_t + W_{f,h} * h_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma(W_{i,x} * X_t + W_{i,h} * h_{t-1} + b_i) \quad (2)$$

$$o_t = \sigma(W_{o,x} * X_t + W_{o,h} * h_{t-1} + b_o) \quad (3)$$

By utilizing the present input (X_t) and the cell output derived from the prior hidden state, (h_{t-1}), the input modulate gate (\tilde{C}_t) is calculated based on Eq. (4).

$$\tilde{C}_t = \tanh(W_{c,x} * X_t + W_{c,h} * h_{t-1} + b_c) \quad (4)$$

In addition, using Eqs. (5) and (6), the present state cell (C_t) and hidden state (h_t) are evaluated, respectively.

$$C_t = C_{t-1} * f_t + i_t * \tilde{C}_t \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

The LSTM formulation involves the following steps:

Given a sequence of input data points, represented as $X = [x_1, x_2, \dots, x_n]$, where x_i is the input at time step i , the LSTM network aims to learn a function that maps this input sequence to an output sequence $Y = [y_1, y_2, \dots, y_n]$, where y_i denotes the result at a specific time step i .

1. Initialize the memory cell state, C_0 , and the hidden state, h_0 , with zeros or small random values.
2. For each time step i from 1 to n , perform the following computations:
 - Compute the input gate, i_i by Eq. (1)
 - Compute the forget gate, f_i using Eq. (2)
 - Compute the output gate, o_i based on Eq. (3)
 - Compute the candidate memory cell state, \tilde{C}_i using Eq. (4)
 - Update the current memory cell state, C_i by Eq. (5)
 - Update the hidden state, h_i based on Eq. (6)
3. After processing all time steps, the final hidden states, h_1, h_2, \dots, h_n , can be used for various tasks such as sequence classification, sequence generation, or further processing.
4. Train the LSTM network by adjusting the hyperparameters using a suitable optimization algorithm and a suitable loss function.

The forecasting precision of LSTM networks is affected by various parameters, including the number of training epochs (N_e), learning rate (r_l), batch size (bs), dropout rate (r_d), and hidden layer nodes (N_h). These parameters play a crucial role in determining the structure of the LSTM. Hence, an effective optimization technique is required for the automated optimization of the parameters associated with the LSTM model.

2.1.2. Support vector regression (SVR)

The SVR method is a regression methodology that is based on support vector machines (SVM). It was originally presented by Vapnik, who incorporated an insensitive loss function into SVM in order to effectively handle regression fitting [35]. One of the key benefits of SVR is that its processing expense is unaffected by the dimensions of the input space. It exhibits reduced computational efficiency in comparison to alternative regression methodologies. Moreover, this model exhibits exceptional generalization capacity, achieves high accuracy in prediction, and demonstrates resilience in the presence of outliers. Unlike traditional regression algorithms that aim to minimize the overall error, SVR focuses on controlling the error within a certain range (defined by epsilon) [35]. In the training phase of SVR, a specific subset of data points referred to as support vectors are employed to establish the hyperplane. The remaining data points are considered as irrelevant or outside the boundary. The support vectors play a crucial role in determining both the position and orientation of the hyperplane. The primary objective is to identify the hyperplane that maximizes the margin while ensuring that the errors (deviation from the margin) remain within the predefined epsilon range.

In SVR, the input of x by m number of various feature is nonlinear and then a linear model is generated utilizing these features. The linear model is as below:

$$f(x, w) = \sum_{i=1}^m w_i g(x_i) + b \quad (7)$$

where $g_i(x)$ points to a transition function, w represents the weight vector and b is the bias term.

This function should not be sensitive to small changes in the model and develop the regression model well. The desired function is known as loss ε -insensitive function. Therefore, the solution of the Eq. (7) can be expressed as a quadratic minimization issue presented below [36]:

$$f = \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i - \xi_i^*) \quad (8)$$

$$s.t. \begin{cases} y_i - f(x_i, w) \leq \varepsilon + \xi_i^* \\ f(x_i, w) - y_i \leq \varepsilon + \xi_i \\ \xi_i, \xi_i^* \geq 0, i = 1, 2, \dots, n \end{cases}$$

where, C and ξ denote the error regularization coefficient and slack variable. This optimum problem would be changed to a dual problem and the selection of carrier vectors and kernel function. The regression function's good efficiency relies on the good selection of C , ε and kernel function parameters.

After resolving the constrained quadratic optimization issue of Eq. (8), the parameter weight vector was solved as shown in Eq. (9):

$$w = \sum_{i=1}^n (\alpha_i^* - \alpha_i) g(x_i) \quad (9)$$

where, α_i^* and α_i are Lagrangian multipliers, which present the amounts gained with solving a quadratic program.

Finally, in dual area, the support vector regression function could be attained by Eq. (10) [36]:

$$f(x) = \sum_{i=1}^n (\alpha_i^* - \alpha_i) K(x_i, x) + b \quad (10)$$

In eq. (10), $K(x_i, x_j) = \exp(-0.5 \|x_i - x_j\|^2 / \sigma^2)$ represents the radial basis kernel function.

In general, the appropriate choice of the parameter set containing C , ε , and σ in SVR has a better influence on the predicting accuracy. Recently, some studies have been performed for utilizing optimization methods for optimizing the SVR model. However, finding the right SVR variables takes time, and the majority of existing optimization techniques prematurely converge to a local optimum.

2.1.3. Multilayer perceptron neural network (MLPNN)

The MLPNN generally refers to the combination of a multilayer perceptron (MLP) architecture with the backpropagation (BP) learning algorithm to train an ANN. The MLP architecture involves multiple layers of solid nodes, where information flows in a feedforward manner between the input and output layers [37]. The process of BP involves modifying the weights within a network by utilizing the computed error, aiming to reduce the disparity between the anticipated output and the intended output. By combining the MLP architecture and the BP algorithm, MLPBPANN enables the training of artificial neural networks to learn and make predictions or classifications based on input data [37]. The fundamental role of an ANN is to propagate input data across the network, with each neuron applying an activation function to the weighted sum of its inputs. This process generates output values that are used as inputs for the next layer until the final result is obtained [37]. During the training phase, an ANN adjusts its weights and biases by iteratively processing training examples and comparing the forecasted results with the real results. This process is typically done using optimization algorithms.

Additionally, some structural characteristics are required for proper determination, including the learning rate, the hidden layer's neuronal count, and the learning momentum [38]. The learning rate describes how the weights vary between neurons at each training stage, while the learning momentum ensures that the direction of the weight value change is constant. The number of neurons in the hidden layer also significantly influences how well MLP makes predictions. As a result, it's important to choose the right number of neurons for the hidden layer [38].

2.2. Proposed hybrid AEF and SCO (AEFSCO)

2.2.1. Artificial electric field algorithm

Coulomb's law indicates that the electrostatic interactions between two charged particles are directly related to the product of their electrical charges and inversely related to the square of the distance between them [33]. The concept forms the foundation of AEFA [33]. In this scenario, the charged particles are denoted as agents, and their charges are utilized to assess the potential of these agents. Charged particles can be subject to an electrostatic force that can either repel or attract them due to the motion of items in the search

space. Electrostatic forces are employed by the particles to establish direct communication, and their placements offer the optimal solution. Consequently, costs are designated based on the population's fitness and the prospective solution. The principle of electrostatic attraction dictates that charged particles with lower charge are attracted to charged particles with higher charge. Furthermore, the solution with a higher charge is considered to be the most optimal. The AEFA can be represented by a mathematical model as follows [33]:

Step 1. The AEFA begins by generating n particles (solutions) at random based on Eq. (11):

$$x_i = (x_{i \max} - x_{i \min}) \times \text{rand}(0, 1) + x_{i \min}, i = 1, 2, \dots, n \quad (11)$$

In Eq. (11), $x_{i \max}$ and $x_{i \min}$ denote the i th parameter's upper and lower boundaries, respectively. n represents an overall count of candidates who have been charged within the population.

Step 2. In this step, each charged particle's fitness value is evaluated. A fitness function takes a probable solution as input and produces an output that quantifies how well this solution aligns with the problem being addressed. In addition, Eqs. (12) and (13) are utilized to assess the best and worst fitness.

$$\text{best}(t) = \min_{i \in \{1, \dots, n\}} \text{fitness}_i(t) \quad (12)$$

$$\text{worst}(t) = \max_{i \in \{1, \dots, n\}} \text{fitness}_i(t) \quad (13)$$

The terms "best (t)" and "worst (t)" refer to the highest and lowest fitness values, respectively, among all the candidates at iteration t .

Step 3. In this stage, the Coulomb's constant, $C(t)$, is determined by Eq. (14):

$$C(t) = C_0 \times \exp\left(-30 \times \frac{t}{\text{Iter}_{\max}}\right) \quad (14)$$

where, C_0 represents the Coulomb's constant's initial value, which is established as 100 [33]. The terms Iter_{\max} reflect the maximum iterations count.

Step 4. The charge of the i th charged particle at time t is denoted as $F_i(t)$. The computation is derived from the present population's fitness based on Eq. (15):

$$F_i(t) = \frac{f_i(t)}{\sum_{i=1}^n f_i(t)} \quad (15)$$

In equation (15), the variable f_i represents a charge function that is selected to produce the highest normalized value for the optimal particle. This value is computed according to Eq. (16).

$$f_i(t) = \exp\left[\frac{\text{fitness}_{xi}(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)}\right] \quad (16)$$

Where, fitness_{xi} denotes the fitness of the i th solution.

Step 5. According to Equation (17), the total electric force operating on particle i at time t in dimension D is

$$EF_i^D(t) = EF_{ij}^D(t) \times \sum_{j=1, j \neq i} \text{rand}() \quad (17)$$

In which, $EF_{ij}^D(t)$ denotes the Coulomb force exerted between particles i and j during the t -th iteration and can be described through Eq. (18).

$$EF_{ij}^D(t) = C(t) \times \frac{(F_i(t) \times F_j(t)) \times (P_j^D(t) - X_j^D(t))}{R_{ij}(t) + \epsilon} \quad (18)$$

The variable ϵ indicates a tiny positive number. $R_{ij}(t)$ is the notation used to represent the distance between solutions i and j . $P_j^D(t)$ and $X_j^D(t)$ represent the most favorable and present location of the charged solution at time t .

Step 6. The acceleration of particle i at time t is represented by $a_i^D(t)$ and can be computed by Eq. (19) in accordance with Newton's second law of motion.

$$a_i^D(t) = \frac{E_i^D(t) \times F_i(t)}{M_i^D(t)}, \quad (19)$$

The unit mass of the i th charged candidate is represented by M_i^D . E_i^D denotes the electric field intensity based on Eq. (20).

$$E_i^D(t) = \frac{EF_i^D(t)}{F_i(t)} \quad (20)$$

Step 7. Finally, Eq. (21) is employed to update the velocity of the particle, whereas equation (22) is utilized to update its position.

$$vel_i^D(t+1) = vel_i^D(t) \times Rand_i + a_i^D(t) \quad (21)$$

$$x_i^D(t+1) = vel_i^D(t) + x_i^D(t) \quad (22)$$

In which, x_i^D denotes the i th charged particle's position, vel_i^D is the velocity of the i th solution. $Rand$ is a random value between 0 and 1.

Fig. 2 illustrates the sequential steps involved in the standard AEFA procedure.

2.2.2. Single candidate optimizer (SCO)

Unlike a wide range of currently employed searching techniques, which depend upon a group of elements for the entire time of the optimization procedure, the SCO only examines one solution in its quest for superior options [39]. The SCO method integrates the two-phase approach and single-candidate methodology to produce a single, robust approach. The essential feature of the proposed technique is that it only employs a single formula based on the present location of the potential answer to modify its location. The algorithm, particularly, employs a unique set of formulas to modify the candidate's location only on the foundation of its knowledge, i. e., its present position.

SCO generates a single potential solution x randomly and then updates it frequently in search of a superior one. Throughout the first phase of SCO, the potential candidate updates its location according to the Eq. (23):

$$x_j = \begin{cases} gbest_j + (w|gbest_j|) & \text{if } rand_1 < 0.5 \\ gbest_j - (w|gbest_j|) & \text{else} \end{cases} \quad (23)$$

where $gbest$ is the global best solution that has reached the best fitness so far and $rand_1$ is a number drawn at random from the interval

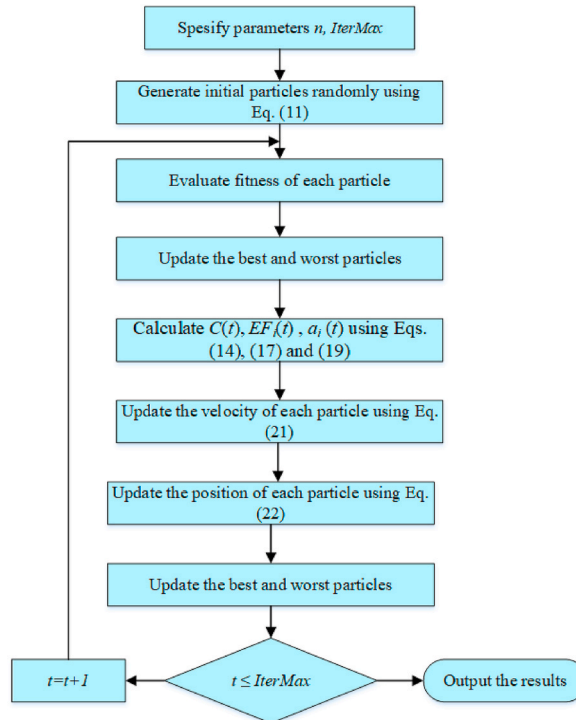


Fig. 2. AEFA flowchart.

[0, 1]. The following formula is a mathematical definition of w :

$$w(t) = \exp\left(-\left(\frac{bt}{T_{max}}\right)^b\right) \quad (24)$$

where t represents the present functional assessment, b represents a constant value, and the maximum iteration count is denoted by T_{max} .

The first phase of SCO ends when T_1 function assessments are finished, and T_2 function assessments begin in the second phase, where $T_1 + T_2 = T_{max}$. The extensive search component of the second phase of the SCO procedure starts with a thorough examination of the zone nearby the best position found in stage one. The final elements of the second phase aid in focusing on potential areas and narrowing the searching zone.

While the following phase is running, the potential solution modifies the position. It is provided in Eq. (25).

$$x_j = \begin{cases} gbest_j + rand_2(ub_j - lb_j) \times w & \text{if } rand_2 < 0.5 \\ gbest_j - rand_2(ub_j - lb_j) \times w, & \text{else} \end{cases} \quad (25)$$

where $rand_2$ is a separate random value having a range of [0, 1], ub_j and lb_j are the higher and lesser boundaries of the borders. w is the parameter that holds the most importance in SCO, as it is responsible for achieving a balance among both exploration and extraction. According to Eq. (24), the total amount of function estimations leads w to reduce exponentially. This characteristic is critical since a high initial value of w greatly aids in effectively exploring the search area at the start of the procedure; instead, a small value of w at the final stage of the optimization approach increases exploiting capability. One of the significant drawbacks of meta-heuristic approaches is being caught in local optimal conditions, particularly throughout the later phases of the search procedure. If there is no enhancement in fitness in the following m function evaluations, the SCO handles this issue in the second stage by changing the position of the potential solution in an alternative approach. A counter c is employed for recording the number of function assessments m that cannot be enhanced consecutively. The applicant's capacity to enhance fitness can be quantified utilizing the integer parameter p , where p equal to 1 indicates an excellent fitness modification and p equals zero denotes an unsuccessful fitness enhancement. In the following stage of

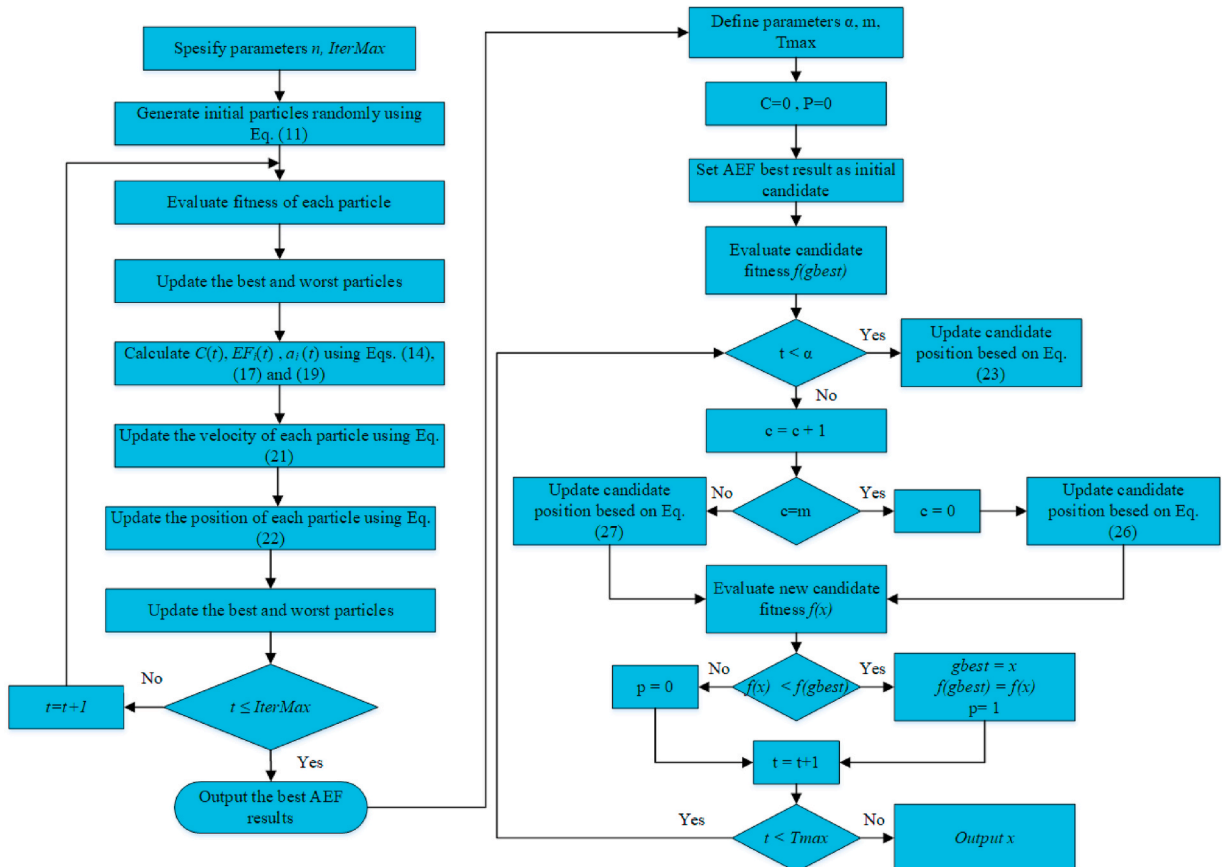


Fig. 3. Hybrid AEFSCO algorithm.

SCO, a potential solution adjusts its location in accordance with Eq. (25), but if doing m consecutive function assessments fails to modify the fitness value, the potential solution enhances its location in the following manner:

$$x_j = \begin{cases} gbest_j + rand_3(ub_j - lb_j) & \text{if } rand_3 < 0.5 \\ gbest_j - rand_3(ub_j - lb_j), & \text{else} \end{cases} \quad (26)$$

In Eq. (26), the potential solution may shift from exploitation to exploration, allowing it to get away from the local minimum. If a variable exceeds its higher or lower boundaries, the updated values are changed accordingly to avoid it going beyond those boundaries using Eq. (27).

$$x_j = \begin{cases} gbest_j & \text{if } x_j > ub_j \\ gbest_j & \text{if } x_j < lb_j \end{cases} \quad (27)$$

2.2.3. Hybrid AEFSCO

The ultimate aim of hybridization is to merge the positive features associated with each approach for improving the output precision [40]. A hybrid strategy may resolve a specific problem more accurately through the integration of two or more methodologies. The AEFSCO methodology, which combines the AEF and SCO methodologies, has been developed during this research. The artificial electric field, a global optimization approach, extensively searches the entire solution space and is probable to provide the best or nearly the best answer. As a consequence, it deserves to be employed in combination with local optimization procedures like SCO.

The global behavior of the artificial electric field is exceptional and easily prevents local optimum. By expanding the quantity of cycles, the precision of the outcomes can be improved by the AEF. Once the quantity of iterations is adequate, AEF can't manage to enhance the precision of the results. As a consequence, AEF's neighborhood search capability remains inadequate. The SCO is useful for exploring a limited area and it is a local optimization method. However, the initial location has a significant impact on the outcome of the SCO algorithm. without any knowledge, the SCO starts the search based on a random solution. If the randomly chosen answer is significantly far away from the optimal solution, the procedure fails to identify the global optimum and the method's stability decreases. If an excellent starting solution is chosen, the SCO will be a quick and efficient technique. It is also important to note that selecting the correct starting point will not only improve the precision of the outcome but additionally make the procedure more reliable. In the present study, we successfully integrate the beneficial features of SCO as a local optimizer and AEF as a global optimizer to determine the final best solution. Since the SCO is dependent on the starting answer, the recommended hybrid AEFSCO strategy begins by the AEF. The AEF is implemented for continuing the search procedure for a predetermined number of repeats. After then, the AEF's best findings is the starting point for the SCO's local exploration. Fig. 3 depicts the procedure for the hybrid approach.

2.3. Hybridization of ML models and AEFSCO for hyperparameters adjustment

This study utilized two approaches to fine-tune the hyperparameters of the machine learning models under consideration. In the initial method, the network's parameters were modified by a conventional trial-and-error technique. This approach is laborious and does not ensure that the optimal parameters for the model have been determined. To manually determine the values of the hyperparameters, it is apparent that one must construct an infinite number of models until an optimal model is obtained. Like most research articles, we only manually develop a restricted set of models by varying the quantity of hyper-parameters because doing so takes a lot of effort and is practically impossible.

In the second approach, the proposed AEFSCO optimization technique was utilized to optimize the key parameters of the LSTM, SVR, and MLPNN models employing the RMSE metric as a fitness function. By applying this methodology, it is feasible to obtain the most optimal model within a restricted time span. Integrating this hybridization technique could enhance both the precision and accuracy of the network's forecasts, as well as effectively address complex settlement prediction challenges. The methodology of optimizing the parameters for the LSTM, SVR, and MLPNN models using AEFSCO is nearly identical. Possible solutions for AEFSCO have been assumed to be the hyperparameter values that need to be optimized. A solution can be denoted as $X = [x_1, x_2, \dots, x_s]$, where s represents the number of model parameters.

In order to evaluate the performance sensitivity of the ML models, this study implements a repeated k-fold cross-validation (CV) scheme. Cross-validation is a widely used method for assessing the accuracy of the generalization of a model [41,42]. Cross-validation is a process that randomly splits the given data into K separate subsets of roughly similar size. It then uses $K-1$ of these subsets for training and the remaining subset for validation. The process continues for K times by modifying the remaining subset. The prediction error can be defined as the mean of K individual errors. In AEFSCO, the objective function amount for solution X is determined by calculating the average of the root mean squared error (RMSE) of cross-validation, as stated below [42]:

$$f(X) = RMSE_{CV} = \frac{1}{K} \sum_{i=1}^K RMSE_i(X) \quad (28)$$

where K represents the number of cross-validation subgroups, $RMSE_i(X)$ denote the RMSE value of the i th subset for cross-validation when the model's hyperparameters are set to X . The value of K in this study is set to 5, indicating that a 5-fold cross-validation approach is used for training.

In the following, the parameters' tuning procedure is presented as a series of basic steps.

1. AEFSCO initialization

The AEFSCO initial parameters, i.e., maximum iterations ($Iter_{max}$) and population size (n), are determined.

2. Generate the initial population

- Initialize a population of candidate solutions with random initial values. Each solution represents potential ML model parameters. It is worth mentioning that there are five parameters should be optimized for the LSTM as mentioned in section 2.1.1 (i.e., N_e , r_b , bs , r_d , N_h), three parameters should be optimized for the SVR as mentioned in section 2.1.2 (C , ε , and σ) and three parameters should be optimized for the MLP as mentioned in section 2.1.3.

3. Data preparation

- Scale and split the data into training and testing sets.

4. Cross-validation

- Apply 5-fold cross-validation approach for training.

5. Fitness assessment

- Evaluate the initial solution fitnesses based on Eq. (28).

6. Update the population

- Update the population and consider the solution with the lowest value of the objective function as the global best.

7. Finding the best parameters

- The procedure of optimization is iterated for a specified number of predetermined iterations ($Iter_{max}$), and the optimal parameters of the ML model are obtained as the output of the optimization procedure.

8. Prediction

- Once the considered ML model is trained and optimized, use it to test the model based on the testing data.

This process is depicted graphically in Fig. 4.

The parameters obtained by each approach for the considered networks have been represented in Table 2.

2.4. Data collection, preprocessing and evaluation indices

In order to create the models for the current research, the database was derived from prior investigations. The employed data set

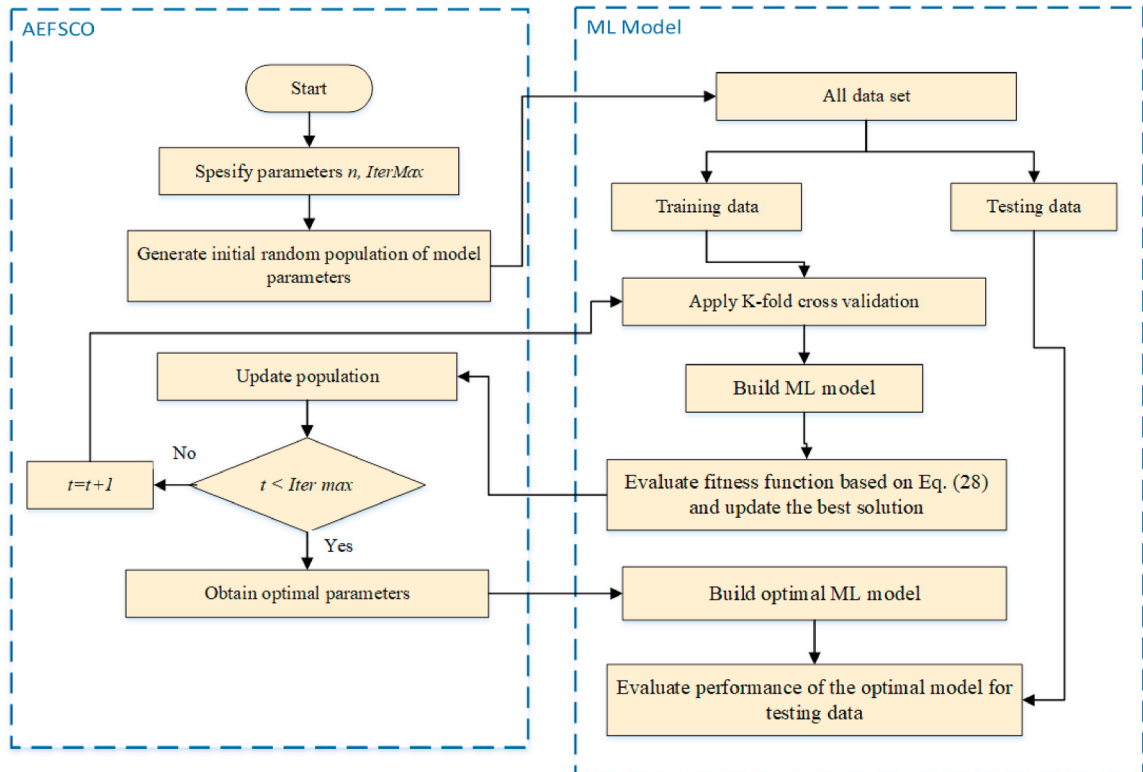


Fig. 4. The parameters' optimization of ML model using AEFSCO.

Table 2

The optimal parameters settings for each model using AEFSCO.

Machine learning model	Parameters	Manually adjusted value	Optimum value obtained by AEFSCO
MLPNN	hidden-layer neuron count.	17	29
	learning rate	0.0311	0.0554
	learning momentum	0.286	0.728
SVR	C	10	10
	ε	0.5	0.01
	γ	1	0.1
LSTM	training epochs number, N_e	10	30
	learning rate, r_l	0.1	0.01
	hidden layer nodes, N_h	64	128
	batch size, b_s	32	16
	dropout rate, r_d	0.2	0.2

includes 189 individual cases of shallow foundations based on experimental results reported in different publications. The authors who reported these cases are as follows: Bazaraa [43] (5 cases), Burbidge [44] (22 cases), Briaud and Gibbens [45] (4 cases), Burland and Burbidge [46] (125 cases), Picornell and Del Monte [47] (one case), Maugeri et al. [48] (2 cases), and Wahls [3] (30 cases). The database covers a wide range of foundation characteristics and soil properties.

The identification of relevant factors in the displacement of a shallow footing on cohesionless soil has an obvious effect on the accuracy of a forecasting model and substantially enhances the model's technical acceptability. In the most important traditional methods, the primary factors influencing foundation settlement are the applied pressure (q), the foundation's width (B), the foundation geometry (L/B), the foundation embedment ratio (D_f/B), and the soil's compressibility evaluated by the standard penetration test (N) [2]. In the testing and training phases, the input and output parameters' statistical characteristics are shown in Table 3.

Prior to using the provided data for machine learning purposes, it is imperative to preprocess the data into an appropriate form. In order to ensure equitable treatment of all variables during the training process, it is advisable to do data scaling prior to training. Standard scaling is a widely used method for scaling input values. The process entails the subtraction of the mean value from each input value, followed by the division of the resulting value by the standard deviation of the distribution. This process effectively adjusts the output values to have a mean of 0 and a standard deviation of 1.

$$x' = \frac{x - \bar{x}}{\sigma} \quad (29)$$

In Eq. (29), the transformed variable, denoted as x' , is associated with the input variable x . \bar{x} represents the mean of the distribution. Additionally, σ represents the standard deviation of the aforementioned distribution.

The performance of the ML models is evaluated using four statistical assessment indices, such as MAE, MSE, RMSE, and R^2 . The equations (Eqs. 30–33) given below can be used to compute these indices.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (30)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (31)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (32)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n \left(y_i - \frac{1}{n} \sum_{i=1}^n y_i \right)^2} \quad (33)$$

Table 3

Parameters' statistical characteristics.

Parameter	Symbol	Unit	Minimum	Maximum	Average	Standard deviation
foundation's width	B	m	0.8	60	8.13	9.72
foundation geometry	L/B	–	1.0	10.6	2.22	1.81
foundation embedment ratio	D_f/B	–	0.0	3.44	0.53	0.58
average SPT blow count	N	–	4.0	60	25.15	13.3
applied pressure	q	kN/m ²	18.32	697	187.31	123.72
settlement	S	mm	0.6	121	20.20	26.10

where n is the number of samples. The variables y and \hat{y} represent the expected and real values of the model, respectively.

3. Results and discussion

In the following sub-sections, the efficacy of the new AEFSCO is examined by means of a collection of thirteen benchmark functions. Subsequently, the accuracy of the generated machine learning models for forecasting the settlement of shallow foundations is investigated. Finally, the findings of the study are compared with those reported in prior research.

3.1. Validation of the AEFSCO approach

In this part, to validate the AEFSCO algorithm's performance as an effective optimization approach, seven uni-modal and six multi-modal benchmark functions are considered. Their outcomes are in comparison with some approaches like the standard AEFA, sine-cosine algorithm (SCA) [49], equilibrium optimizer (EO) [50], and grey wolf optimizer (GWO) [51]. The simulation was carried out in the Matlab version of 2016b. Test functions are presented in Table 4, which contains the optimal amount and formulation of these functions.

In the simulation process, the whole algorithms have been conducted by regarding similar amounts of the highest iteration and population equal to 30 and 1000, orderly. Furthermore, to provide reliable validation, whole algorithms have been run twenty times independently. The comparative analysis of the hybrid artificial electric field single candidate optimization algorithm (AEFSCO) and the other presented algorithms are demonstrated in Table 5. The analysis is according to 3 indexes containing, Min amount (Best), Mean amount, and Standard Deviation amount (Std).

According to this table, the best outcomes of minimum (Best) and average (Mean) are attained by the AEFSCO. This demonstrates that the suggested algorithm could deliver better precision for the analyzed functions. In addition, according to the standard deviation (Std) amount outcomes, the AEFSCO delivers the minimum Std that demonstrates its high reliability in twenty times independent runs. As previously stated, the suggested AEFSCO is employed for calculating the MLPNN, SVR, and LSTM networks parameters for reducing the model's training error.

3.2. Application of the developed models for settlement prediction

The shallow foundation settlement in cohesionless soil has been predicted in the present research employing six ML models. In line with Shahin et al. [2] and Shahnazari et al. [25], the ML models are fed with the following input parameters of the foundation: width (B), standard penetration value (N), applied pressure (q), embedment ratio (Df/B), and length-to-width ratio (L/B). The settlement (S_m) is the sole output variable.

In the conducted experiment, a stochastic selection procedure was employed to allocate 80 percent of the available data for training the ML models. Meanwhile, a portion of 20 % of the data is allocated for the purpose of testing. The created models were validated using the k-fold validation procedure, as recommended by prior publications [42].

In the following, all the findings produced by the developed ML models for settlement prediction are analyzed. In order to assess the impact of the AEFSCO algorithm on the performance of the considered models, a comparative analysis is conducted. The outcomes of the models with manually adjusted parameters (LSTM, SVR, and MLPNN) are systematically compared with those of the models with automatically adjusted parameters using AEFSCO (LSTM-AEFSCO, SVR-AEFSCO, and MLPNN-AEFSCO).

Figs. 5–10 depicts the level of accuracy in the forecasted outcomes for specific data points and the degree of alignment with the corresponding real values for the purpose of training and testing dataset estimated by each model.

The results presented in these figures indicate that, despite the predicted performance of the manually adjusted models (MLPNN, SVR, and LSTM) is reasonable, additional improvement of predictions is necessary for establishing the suitability of these models as reliable tools for forecasting the settlement based on novel data. To achieve this, it is essential to modify the hyperparameters of the ML models by utilizing effective AEFSCO optimization approaches. As shown in these figures, the optimized models could increase the accuracy of the results for both training and testing dataset.

Figs. 11–16 display the scatter plots for each model, illustrating the comparison of the coefficient of determination between real and estimated settlement data for both the train and test datasets.

Figs. 11 and 12 represent that the AEFSCO could increase the accuracy of the results. The MLP model has R^2 values of 0.8668 and 0.8608 for training and testing data, while the R^2 values for the optimized MLP with AEFSCO are 0.9494 and 0.9436, respectively.

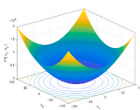
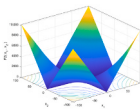
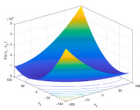
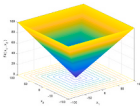
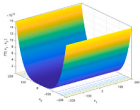
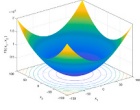
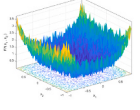
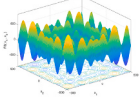
Upon examining the outcomes presented in Figs. 13 and 14, it is evident that the optimized SVR-AEFSCO model exhibits R^2 values of 0.9290 for the training portion and 0.9406 for the testing portion. These values exceed SVR by 0.8855 and 0.8752, respectively.

As depicted in Figs. 15 and 16, the R^2 values of the LSTM model are 0.9236 and 0.8075 for the training and testing data, while these values are 0.9903 and 0.9830 for the optimized LSTM-AEFSCO, which are the highest values among all developed models, indicating that the constructed LSTM-AEFSCO model outperforms the other models in terms of generalization ability and prediction effect. The results indicate the combination of the LSTM and AEFSCO algorithms has the potential to get superior scores compared to the other considered models, and it might be suggested as an appropriate way of determining the settlement of shallow foundations.

Table 6 displays the effectiveness of the developed machine learning models as measured by the statistical indices. These indices are employed to quantify the accuracy of the predictions made by each model.

The presented results in Table 6 indicate that the proposed AEFSCO could increase the performance of all considered models.

Table 4
Test functions.

Function	Name	f_{min}	n (Dim)	3D view
$F_1(X) = \sum_{i=1}^n x_i^2$	Sphere	0	30	
$F_2(X) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	Schwefel 2.22	0	30	
$F_3(X) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	Schwefel 1.2	0	30	
$F_4(X) = \max_i \{ x_i , 1 \leq i \leq n \}$	Schwefel 2.21	0	30	
$F_5(X) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	Generalized Rosenbrock	0	30	
$F_6(X) = \sum_{i=1}^n ((x_i + 0.5))^2$	step	0	30	
$F_7(X) = \sum_{i=1}^n ix_i^4 + random[0,1)$	Quartic	0	30	
$F_8(X) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	Generalized Schwefel 2.26	$428.9829 \times n$	30	

(continued on next page)

Table 4 (continued)

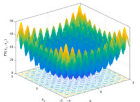
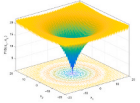
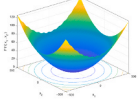
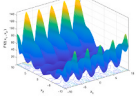
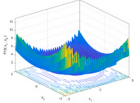
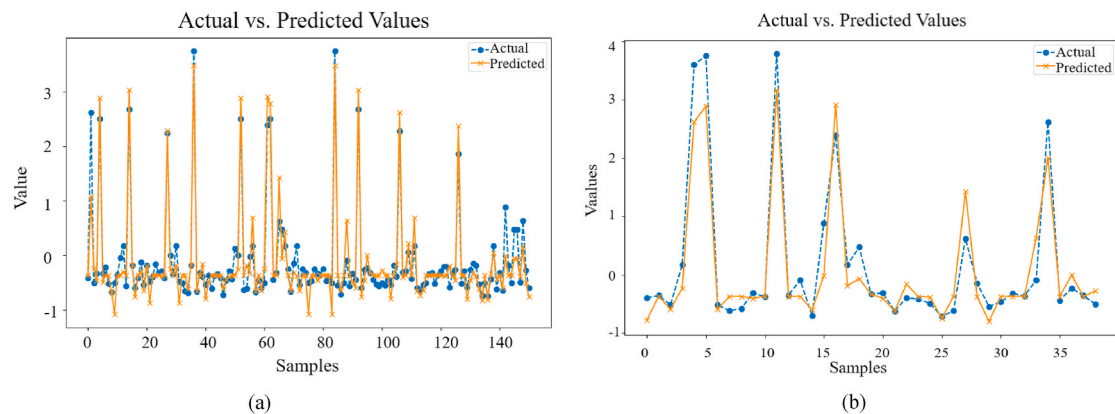
Function	Name	f_{min}	n (Dim)	3D view
$F_9(X) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	Generalized Rastrigin	0	30	
$F_{10}(X) = 20 - 20 \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + e$	Ackley	0	30	
$F_{11}(X) = \frac{1}{4000}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	Griewank	0	30	
$F_{12}(X) = \frac{\pi}{n}\left\{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$ $y_i = 1 + \frac{x_{i+4}}{4}$	Generalized Penalized 1	0	30	
$F_{13}(X) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]\right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	Generalized Penalized 2	0	30	

Table 5

Analyses of the proposed AEFSCO and other comparable techniques.

Function	Statistics	AEFSCO	AEF	EO	SCA	GWO
F_1	Best	0.0	2.25e-27	2.23e-40	1.45e-07	2.55e-61
	Mean	0.0	3.74e-27	3.25e-40	2.42e-04	4.87e-59
	Std.	0.0	6.24e-26	4.76e-40	7.85e-04	1.12e-58
F_2	Best	2.12e-118	1.56e-12	2.52e-23	1.2610	9.8174e-16
	Mean	1.88e-115	1.26e-11	7.05e-23	9.3080	1.8641e-14
	Std.	5.34e-111	1.65e-10	6.64e-23	8.0720	4.5127e-14
F_3	Best	0.0	6.78e-7	6.14e-10	69.81	1.125e-19
	Mean	0.0	1.23e-8	8.19e-9	1.895e+3	1.492e-14
	Std.	0.0	8.65e-8	1.87e-8	462.2	6.211e-14
F_4	Best	1.509e-239	6.86e-13	8.46e-11	1.181	9.716e-16
	Mean	7.785e-131	9.76e-13	5.27e-10	9.132	1.864e-14
	Std.	3.482e-130	6.81e-12	1.41e-9	7.956	4.354e-14
F_5	Best	2.973e-4	26.51	25.86	27.232	25.34
	Mean	1.9319	28.81	25.23	29.89	27.04
	Std.	2.5059	0.862	0.186	1.02	0.792
F_6	Best	0.0	0.0	4.29e-6	3.234	0.251
	Mean	0.0	0.0	8.14e-6	4.132	0.642
	Std.	0.0	0.0	7.49e-6	0.341	0.342
F_7	Best	1.43e-4	0.0121	8.16e-4	1.461e-3	1.462e-4
	Mean	3.52e-4	0.0364	1.17e-3	1.252e-2	7.543e-4
	Std.	4.28e-4	0.0201	8.32e-4	1.021e-2	4.518e-4
F_8	Best	-1.258e+04	-3.003e3	-9.788e+03	-5.321e+03	-8.742e+03
	Mean	-1.741e+04	-2.782e3	-9.166e+03	-4.152e+03	-6.314e+03
	Std.	289.9842	365.4671	543.25	329.869	849.474
F_9	Best	0.0	21.4451	0.0	1.0560e-06	0.00
	Mean	0.0	5.9694	0.0	6.0174	0.8853
	Std.	0.0	12.2476	0.0	11.856	2.4438
F_{10}	Best	4.44e-16	3.3313e-12	3.64e-14	1.497e-05	1.218e-14
	Mean	4.44e-16	2.3935e-12	8.25e-14	13.822	1.624e-14
	Std.	0.0	4.0025e-13	2.45e-14	9.0178	2.614e-15
F_{11}	Best	0.0	0.0	0.0	4.764e-7	0.0
	Mean	0.0	0.0	0.0	0.142	0.0
	Std.	0.0	0.0	0.0	0.218	0.0
F_{12}	Best	5.632e-7	2.685e-4	6.89e-6	0.2710	1.34e-2
	Mean	5.387e-7	0.259	7.75e-6	0.942	3.16e-2
	Std.	9.514e-8	0.175	7.24e-6	1.018	2.05e-2
F_{13}	Best	1.056e-4	1.049	0.0187	1.814	0.103
	Mean	0.019	0.197	0.0286	3.365	0.517
	Std.	0.024	0.256	0.0318	3.846	0.219

**Fig. 5.** Actual vs. estimated settlement comparison using MLPNN: (a) training data, (b) testing data.

Considering the testing data analysis, the AEFSCO has increased the coefficient of determination (R^2) value of the MLPNN model by 9.3 %, the SVR model by 8 %, and the LSTM model by 22 %. In addition, the findings from Table 6 demonstrate that the LSTM model, optimized with AEFSCO, performs exceptionally well, with a maximum R^2 of 0.9903 and 0.983 for the training and testing datasets, showing a strong relationship among measured and estimated settlement. The LSTM-AEFSCO model is more appropriate than the SVR-AEFSCO and MLPNN-AEFSCO by raising the value of R^2 from 0.9494 to 0.9290 to 0.9903, with a rise of about 4.5 % and 6 %. It is

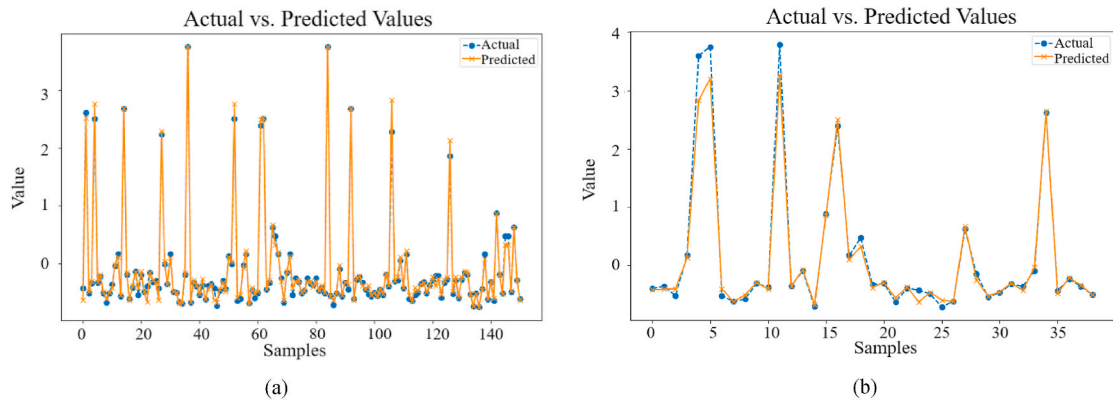


Fig. 6. Actual vs. estimated settlement comparison using MLPNN-AEFSCO: (a) training data, (b) testing data.

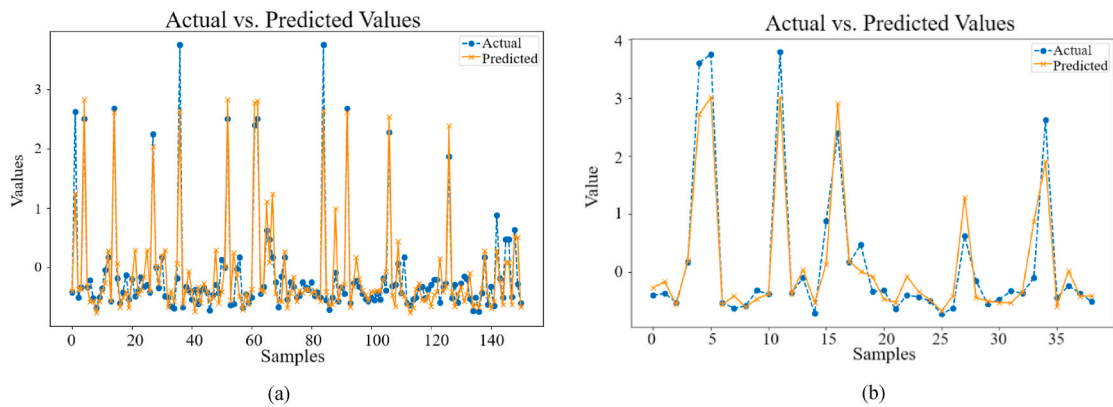


Fig. 7. Actual vs. estimated settlement comparison using SVR: (a) training data, (b) testing data.

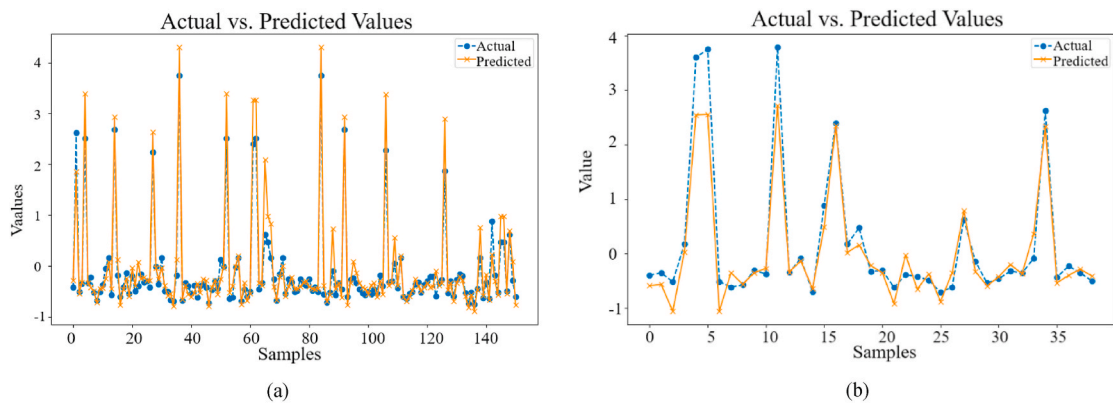


Fig. 8. Actual vs. estimated settlement comparison using SVR-AEFSCO: (a) training data, (b) testing data.

conceivable to claim that the LSTM optimized with AEFSCO performs better than the other models by taking other error indices like MEA and RMSE into account (the lower value, the higher accuracy). For the RMSE index, as an instance, LSTM-AEFSCO could gain 0.1486 for testing data, which is properly smaller than the 0.2696 and 0.2368 evaluated by the SVR-AEFSCO and MLPNN-AEFSCO models, respectively. According to the outcomes, it becomes evident that the LSTM-AEFSCO system exhibits the highest level of performance.

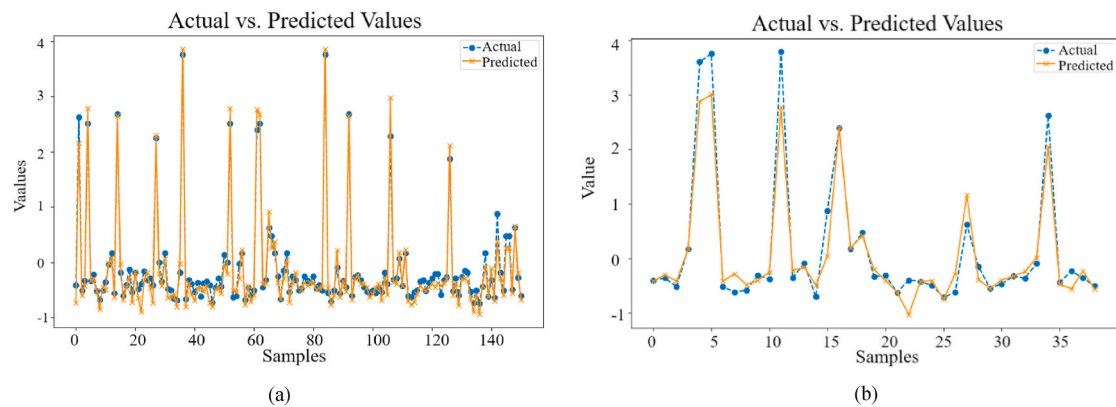


Fig. 9. Actual vs. estimated settlement comparison using LSTM: (a) training data, (b) testing data.

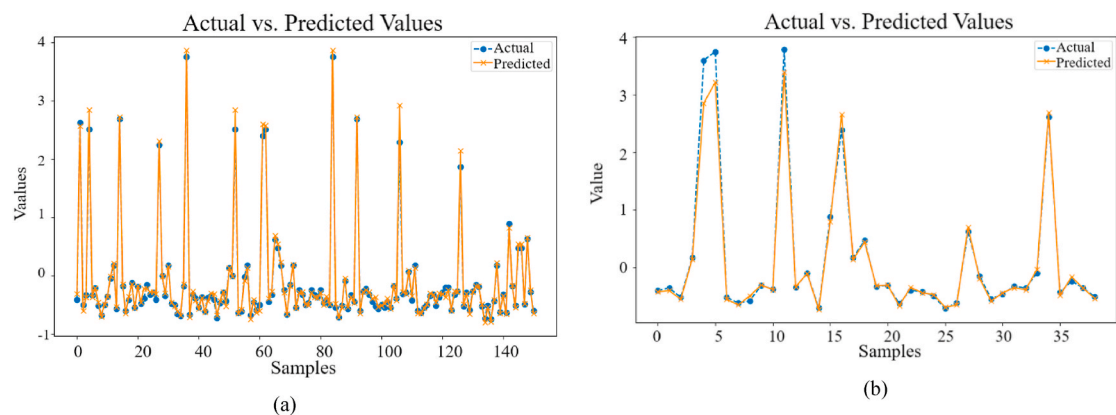


Fig. 10. Actual vs. estimated settlement comparison using LSTM-AEFSO: (a) training data, (b) testing data.

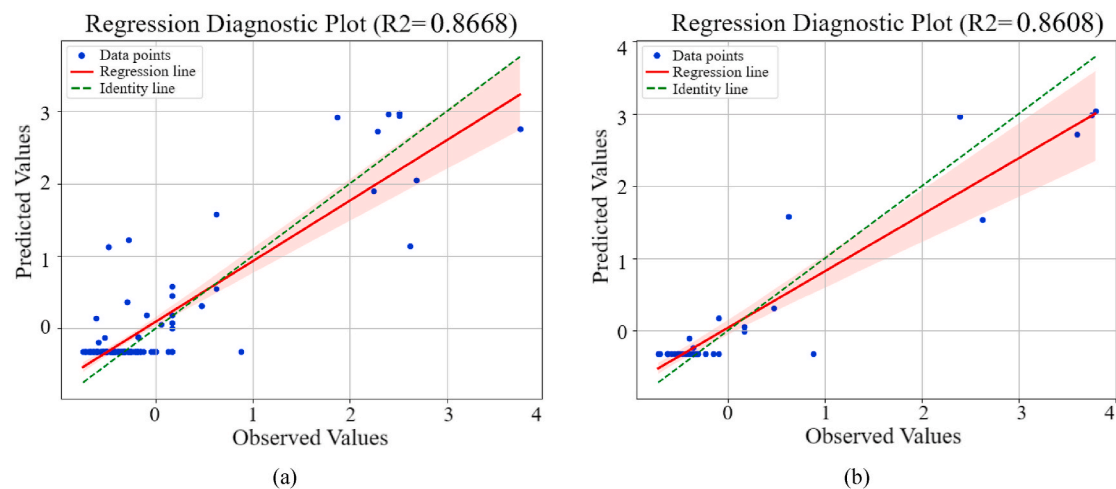


Fig. 11. Scatter plots for observed and the estimated settlement using MLPNN: (a) training data (b) testing data.

3.3. Comparative analysis

In order to further verify the developed models, the outcomes of the present study are compared with the findings presented in the literature. Shahin et al. [2] utilized ANNs, and Shahnazari et al. [25] employed gene expression programming (GEP), classical genetic

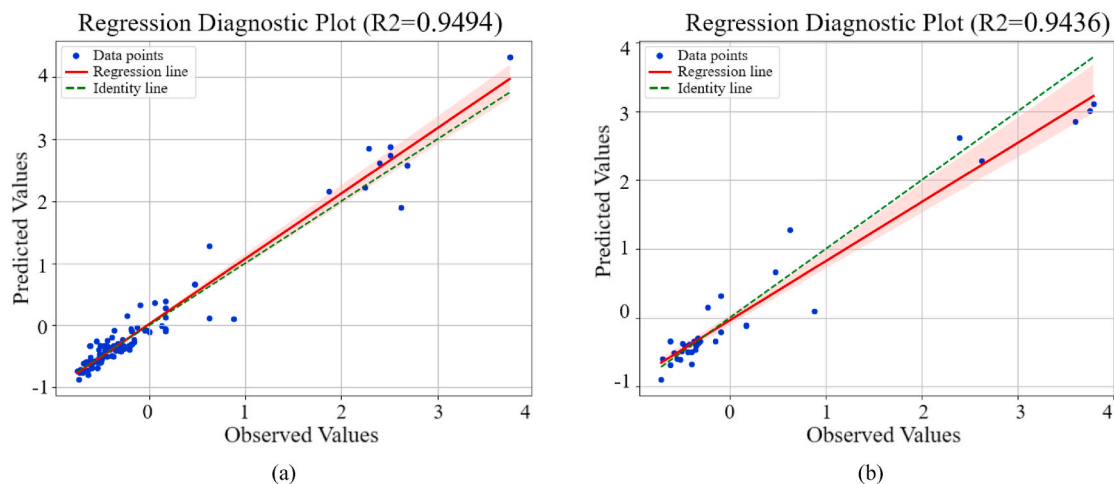


Fig. 12. Scatter plots for observed and the estimated settlement using MLPNN- AEFSCO: (a) training data (b) testing data.

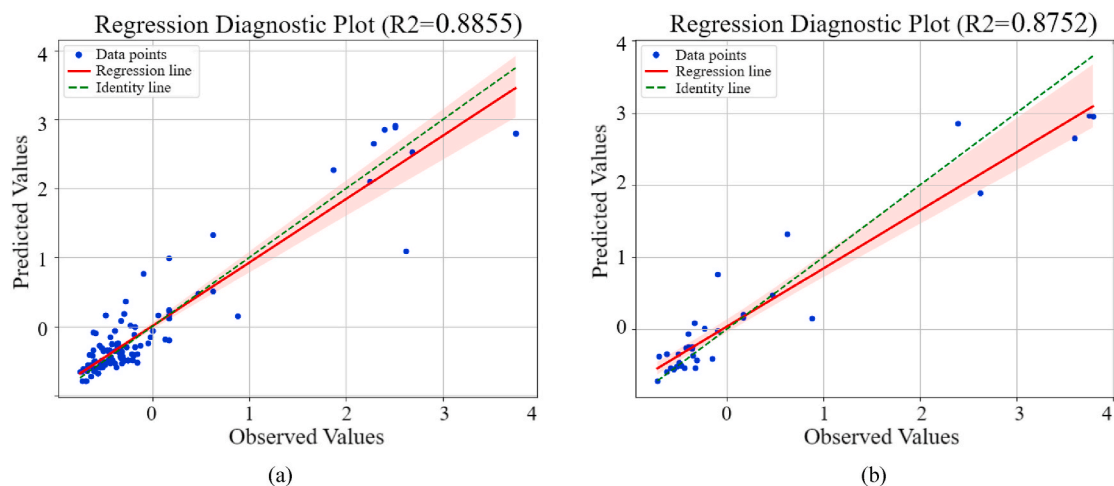


Fig. 13. Scatter plots for observed and the estimated settlement using SVR: (a) training data (b) testing data.

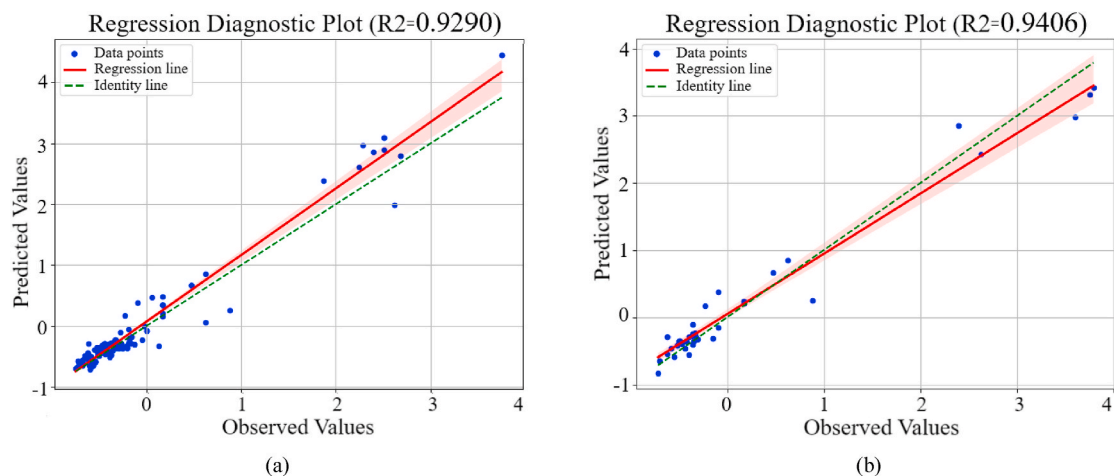


Fig. 14. Scatter plots for observed and the estimated settlement using SVR- AEFSCO: (a) training data (b) testing data.

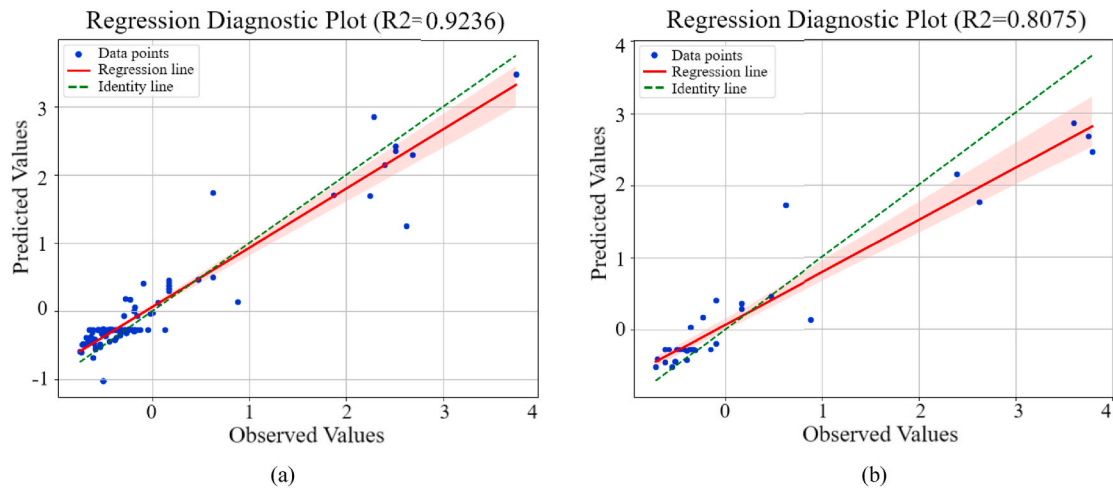


Fig. 15. Scatter plots for observed and the estimated settlement using LSTM: (a) training data (b) testing data.

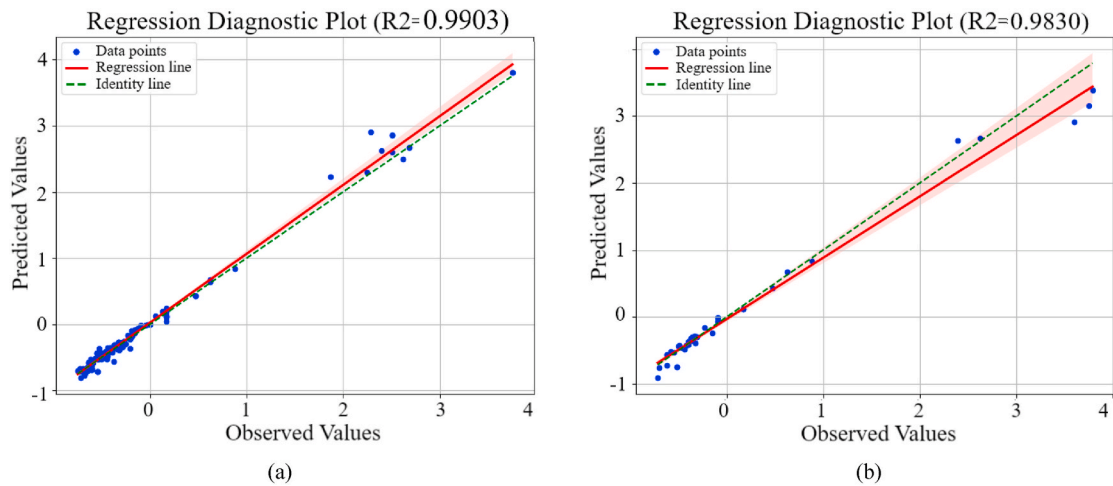


Fig. 16. Scatter plots for observed and the estimated settlement using LSTM-AEFSCO: (a) training data (b) testing data.

Table 6

Statistical assessment results for considered machine learning models.

ML Models	Data samples	R^2	MSE	MAE	RMSE
MLPNN	Train	0.8668	0.1342	0.2432	0.3664
	Test	0.8608	0.1396	0.2477	0.3736
MLPNN-AEFSCO	Train	0.9494	0.0508	0.1284	0.2254
	Test	0.9436	0.0561	0.1341	0.2368
SVR	Train	0.8855	0.1128	0.2162	0.3358
	Test	0.8752	0.1289	0.2384	0.3590
SVR-AEFSCO	Train	0.9290	0.0726	0.1586	0.2696
	Test	0.9406	0.0613	0.1386	0.2475
LSTM	Train	0.9236	0.0772	0.164	0.2778
	Test	0.8075	0.2153	0.2876	0.4639
LSTM-AEFSCO	Train	0.9903	0.0223	0.0775	0.1494
	Test	0.9830	0.0221	0.0768	0.1486

programming (GP), and evolutionary polynomial regression (EPR). In addition, Jibanchand et al. [28] assess the viability of Bagging, Random Forest, Adaptive Boosting, and Extreme Gradient Boosting methods to forecast the settlement of shallow foundations. Table 7 presents a comparison between the best model developed in this study (LSTM-AEFSCO) and other models found in the literature.

As presented in Table 7, the best value (highest value) of the coefficient of determination (R^2) of the previous study belongs to

Table 7

Results omparison with other models from previous researches.

Reference	ML Model	Data samples	R ²
Present Study (Best model)	LSTM-AEFSCO	Training	0.9903
		Testing	0.9830
Shahin et al. [2]	Artificial Neural Network	Training	0.865
		Testing	0.863
Shahnazari et al. [25]	GEP	Training	0.801
		Testing	0.791
Shahnazari et al. [25]	GP	Training	0.880
		Testing	0.868
Shahnazari et al. [25]	ERP	Training	0.874
		Testing	0.856
Jibanchand et al. [28]	Bagging	Training	0.926
		Testing	0.901
Jibanchand et al. [28]	Random Forest	Training	0.943
		Testing	0.886
Jibanchand et al. [28]	Adaptive Boosting	Training	0.876
		Testing	0.887
Jibanchand et al. [28]	Extreme Gradient Boosting	Training	0.995
		Testing	0.915

Extreme Gradient Boosting [28], which is equal to 0.995 and 0.915 for the training and testing datasets, respectively. The obtained outcomes of the LSTM-AEFSCO model reveal that the values of R² for the training and testing data are 0.9903 and 0.983, respectively. These values are in line with Extreme Gradient Boosting [28]. However, the LSTM-AEFSCO model has better accuracy for the testing data. Comparing the findings of this research with the developed techniques from previous research demonstrates that the LSTM-AEFSCO has excellent performance and carries a high success rate of prediction.

4. Conclusions

The present work aimed to assess and compare the predictive capabilities of three powerful machine learning models: MLPNN, SVR, and LSTM, optimized by an efficient optimization approach for estimation of shallow foundation settlement situated on cohesionless soil. To achieve this aim, a novel hybrid artificial electric field and single candidate algorithm (AEFSCO) is introduced. In the proposed method, the global best solution obtained by AEFA is fine-tuned by a single candidate optimizer. To evaluate the efficiency of the suggested technique, it is applied to several widely recognized benchmark problems. The results are then compared with those obtained from existing methods. In comparison to the AEFA and other approaches, the AEFSCO demonstrated superior statistical outcomes for the aforementioned test functions. In the following step, the AEFSCO is employed to optimize the hyperparameters of the ML models for estimating the settlement of footings. The models had been trained and validated utilizing a comprehensive database consisting of 189 practical cases was compiled from the existing literature. Based on the results obtained, it is possible to fine-tune the parameters of the ML models through the utilization of a suggested optimization technique. This approach has demonstrated the potential to greatly enhance the precision of the models' predictions. The study compared the outcomes of the hybrid models with the manually optimized models to investigate the impact of optimization on the predictive capabilities of the ML models. The AEFSCO has enhanced the coefficient of determination (R²) value of the MLPNN model by 9.3 %, increasing it from 0.8668 to 0.9494. Similarly, the SVR model's R² value has been improved by 8 %, rising from 0.8855 to 0.9290. Furthermore, the LSTM model's R² value has experienced a significant enhancement of 22 %, increasing from 0.9236 to 0.9903. The results also showed that the LSTM-AEFSCO model demonstrated higher accuracy and robustness in forecasting foundation settlement compared to other models. It achieved R² values of 0.9903 in training and 0.9830 in testing, outperforming the outcomes of the other models.

Limitations and recommendations for future studies

The outcomes of this study rely on a particular dataset, and the sample size might not adequately encompass the variety of soil properties and situations. Hence, the extent to which the results can be applied to all conditions may be limited.

In addition, the study primarily examined the suitability of the AEFSCO optimization technique for tuning hyperparameters. Alternative machine learning models and optimization strategies were not investigated in this study. However, future research can explore other optimization algorithms to investigate their ability to enhance the predictive accuracy of machine learning models. Furthermore, the suitability of the developed optimization technique and optimized machine learning models for addressing various engineering issues can be considered in future research.

Fundings

This work (Grant No. RGNS 65-112) was supported by Office of the Permanent Secretary, Ministry of Higher Education, Science, Research and Innovation (OPS MHESI), Thailand Science Research and Innovation (TSRI) and Thammasat University.

Data availability

The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Statements and declarations

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

CRedit authorship contribution statement

Mohammad Khajehzadeh: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Suraparb Keawsawasvong:** Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Viroon Kamchoom:** Writing – review & editing, Visualization, Validation, Supervision, Software, Methodology, Funding acquisition, Conceptualization. **Chao Shi:** Writing – review & editing, Visualization, Validation, Supervision, Resources, Methodology, Funding acquisition, Conceptualization. **Alimorad Khajehzadeh:** Writing – review & editing, Visualization, Validation, Software, Methodology, Investigation, Conceptualization.

Declaration of competing interest

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

Acknowledgements

The third author would like to acknowledge the grant (RE-KRIS/FF67/007) from King Mongkut's Institute of Technology Ladkrabang (KMUTL) and National Science, Research and Innovation Fund (NSRF).

References

- [1] B.M. Das, *Principles of Foundation Engineering*, Cengage learning, 2015.
- [2] M.A. Shahin, H.R. Maier, M.B. Jaksa, Predicting settlement of shallow foundations using neural networks, *J. Geotech. Geoenviron. Eng.* 128 (2002) 785–793.
- [3] H. Wahls, Settlement analysis for shallow foundations on sand, in: *Proceedings of the 3rd International Geotechnical Engineering Conference*, 1997, pp. 7–28.
- [4] C. Tan, J. Duncan, Settlement of footings on sands—accuracy and reliability, in: *Geotechnical Engineering Congress(ASCE)*, 1991, pp. 446–455.
- [5] B. Manda, P. Bhaskare, R. Muthuganapathy, A convolutional neural network approach to the classification of engineering models, *IEEE Access* 9 (2021) 22711–22723.
- [6] S. Tang, J. Wang, C. Tang, Identification of microseismic events in rock engineering by a convolutional neural network combined with an attention mechanism, *Rock Mech. Rock Eng.* 54 (2021) 47–69.
- [7] Y. Liu, C. Lyu, X. Liu, Z. Liu, Automatic feature engineering for bus passenger flow prediction based on modular convolutional neural network, *IEEE Trans. Intell. Transport. Syst.* 22 (2020) 2349–2358.
- [8] O. Janssens, et al., Convolutional neural network based fault detection for rotating machinery, *J. Sound Vib.* 377 (2016) 331–345.
- [9] T. Nguyen, A. Kashani, T. Ngo, S. Bordas, Deep neural network with high-order neuron for the prediction of foamed concrete strength, *Comput. Aided Civ. Infrastruct. Eng.* 34 (2019) 316–332.
- [10] Y. Cheng, K. Hu, J. Wu, H. Zhu, X. Shao, Autoencoder quasi-recurrent neural networks for remaining useful life prediction of engineering systems, *IEEE ASME Trans. Mechatron.* 27 (2021) 1081–1092.
- [11] R. Al-Shabandar, A. Jaddoa, P. Liatsis, A.J. Hussain, A deep gated recurrent neural network for petroleum production forecasting, *Machine Learning with Applications* 3 (2021) 100013.
- [12] Q. Cao, B.T. Ewing, M.A. Thompson, Forecasting wind speed with recurrent neural networks, *Eur. J. Oper. Res.* 221 (2012) 148–154.
- [13] Y. Huang, X. Han, L. Zhao, Recurrent neural networks for complicated seismic dynamic response prediction of a slope system, *Eng. Geol.* 289 (2021) 106198.
- [14] S. Fan, N. Xiao, S. Dong, A novel model to predict significant wave height based on long short-term memory network, *Ocean Eng.* 205 (2020) 107298.
- [15] C. Chen, N. Lu, B. Jiang, Y. Xing, Z.H. Zhu, Prediction interval estimation of aeroengine remaining useful life based on bidirectional long short-term memory network, *IEEE Trans. Instrum. Meas.* 70 (2021) 1–13.
- [16] A. Mahmoodzadeh, et al., Prediction of Mode-I rock fracture toughness using support vector regression with metaheuristic optimization algorithms, *Eng. Fract. Mech.* 264 (2022) 108334.
- [17] F. Kang, J. Li, J. Dai, Prediction of long-term temperature effect in structural health monitoring of concrete dams using support vector machines with Jaya optimizer and salp swarm algorithms, *Adv. Eng. Software* 131 (2019) 60–76.
- [18] H. Zhong, J. Wang, H. Jia, Y. Mu, S. Lv, Vector field-based support vector regression for building energy consumption prediction, *Appl. Energy* 242 (2019) 403–414.
- [19] A. Zeng, H. Ho, Y. Yu, Prediction of building electricity usage using Gaussian Process Regression, *J. Build. Eng.* 28 (2020) 101054.
- [20] K. Liu, X. Hu, Z. Wei, Y. Li, Y. Jiang, Modified Gaussian process regression models for cyclic capacity prediction of lithium-ion batteries, *IEEE Transactions on Transportation Electrification* 5 (2019) 1225–1236.
- [21] M. Khajehzadeh, M.R. Taha, S. Keawsawasvong, H. Mirzaei, M. Jebeli, An effective artificial intelligence approach for slope stability evaluation, *IEEE Access* 10 (2022) 5660–5671.
- [22] H. Wang, F. Yang, S. Shen, Supply fraud forecasting using decision tree algorithm, in: *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 2021, pp. 344–347.
- [23] A. Mahmoodzadeh, et al., Developing six hybrid machine learning models based on Gaussian process regression and meta-heuristic optimization algorithms for prediction of duration and cost of road tunnels construction, *Tunn. Undergr. Space Technol.* 130 (2022) 104759.
- [24] Y. Zhou, S. Li, C. Zhou, H. Luo, Intelligent approach based on random forest for safety risk prediction of deep foundation pit in subway stations, *J. Comput. Civ. Eng.* 33 (2019) 05018004.

- [25] H. Shahnazari, M.A. Shahin, M. Tutunchian, Evolutionary-based approaches for settlement prediction of shallow foundations on cohesionless soils, *Int. J. Civ. Eng.* 12 (2014) 55–64.
- [26] M. Rezaia, A.A. Javadi, A new genetic programming model for predicting settlement of shallow foundations, *Can. Geotech. J.* 44 (2007) 1462–1473.
- [27] P. Samui, Support vector machine applied to settlement of shallow foundations on cohesionless soils, *Comput. Geotech.* 35 (2008) 419–427.
- [28] N. Jibanchand, K.R. Devi, Application of ensemble learning in predicting shallow foundation settlement in cohesionless soil, *Int. J. Geotech. Eng.* (2023) 1–12.
- [29] A. Kalinli, M.C. Acar, Z. Gündüz, New approaches to determine the ultimate bearing capacity of shallow foundations based on artificial neural networks and ant colony optimization, *Eng. Geol.* 117 (2011) 29–38.
- [30] M. Ahmad, et al., Prediction of ultimate bearing capacity of shallow foundations on cohesionless soils: a Gaussian process regression approach, *Appl. Sci.* 11 (2021) 10317.
- [31] D.R. Kumar, et al., Machine learning approaches for prediction of the bearing capacity of ring foundations on rock masses, *Earth Science Informatics* 1–16 (2023).
- [32] B. Haznedar, H.C. Kilinc, F. Ozkan, A. Yurtsever, Streamflow forecasting using a hybrid LSTM-PSO approach: the case of Seyhan Basin, *Nat. Hazards* 117 (2023) 681–701.
- [33] A. Yadav, AEFA: artificial electric field algorithm for global optimization, *Swarm Evol. Comput.* 48 (2019) 93–108.
- [34] A. Demirören, S. Ekinci, B. Hekimoğlu, D. Izci, Opposition-based artificial electric field algorithm and its application to FOPID controller design for unstable magnetic ball suspension system, *Engineering Science and Technology, an International Journal* 24 (2021) 469–479.
- [35] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer science & business media, 1999.
- [36] R. Maity, P.P. Bhagwat, A. Bhatnagar, Potential of support vector regression for prediction of monthly streamflow using endogenous property, *Hydrol. Process.: Int. J.* 24 (2010) 917–923.
- [37] M. Hasanipناه, et al., Intelligent prediction of rock mass deformation modulus through three optimized cascaded forward neural network models, *Earth Science Informatics* 15 (2022) 1659–1669.
- [38] N. Babanouri, S.K. Nasab, S. Sarafrazi, A hybrid particle swarm optimization and multi-layer perceptron algorithm for bivariate fractal analysis of rock fractures roughness, *Int. J. Rock Mech. Min. Sci.* 60 (2013) 66–74.
- [39] T.M. Shami, D. Grace, A. Burr, P.D. Mitchell, Single candidate optimizer: a novel optimization algorithm, *Evolutionary Intelligence* 1–25 (2022).
- [40] I. Cherki, A. Chaker, Z. Djidar, N. Khalfallah, F. Benzerghua, A sequential hybridization of genetic algorithm and particle swarm optimization for the optimal reactive power flow, *Sustainability* 11 (2019) 3862.
- [41] T. Tvedskov, T. Meretoja, M. Jensen, M. Leidenius, N. Kroman, Cross-validation of three predictive tools for non-sentinel node metastases in breast cancer patients with micrometastases or isolated tumor cells in the sentinel node, *Eur. J. Surg. Oncol.* 40 (2014) 435–441.
- [42] P. Jiang, J. Chen, Displacement prediction of landslide based on generalized regression neural networks with K-fold cross-validation, *Neurocomputing* 198 (2016) 40–47.
- [43] A.R.S.S. Bazaraa, *Use Of the Standard Penetration Test for Estimating Settlements of Shallow Foundations on Sand* Ph. D. Thesis Thesis, University of Illinois at Urbana-Champaign, 1967.
- [44] M. Burbidge, *A Case Study Review of Settlements on Granular Soil* MSc. Thesis Thesis, University of London, 1982.
- [45] J.-L. Briaud, R. Gibbens, Behavior of five large spread footings in sand, *J. Geotech. Geoenviron. Eng.* 125 (1999) 787–796.
- [46] J. Burland, M. Burbidge, E. Wilson, Terzaghi. Settlement of Foundations on Sand and Gravel, vol. 78, *Proceedings of the institution of Civil Engineers*, 1985, pp. 1325–1381.
- [47] M. Picornell, E. Del Monte, Prediction of settlements of cohesive granular soils, in: *Measured Performance of Shallow Foundations*, 1988, pp. 55–72.
- [48] M. Maugeri, F. Castelli, M.R. Massimino, G. Verona, Observed and computed settlements of two shallow foundations on sand, *J. Geotech. Geoenviron. Eng.* 124 (1998) 595–605.
- [49] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, *Knowl. Base Syst.* 96 (2016) 120–133.
- [50] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: a novel optimization algorithm, *Knowl. Base Syst.* 191 (2020) 105190.
- [51] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software* 69 (2014) 46–61.