Received: 8 May 2024 | Revised: 6 July 2024 | Accepted: 14 July 2024

DOI: 10.1002/msd2.12124

RESEARCH ARTICLE

IJMSD International Journal of Mechanical System Dynamics Open Access WILEY

Machine learning and numerical investigation on drag coefficient of arbitrary polygonal particles

Haonan Xiang[1] | Cheng Cheng[1] | Pei Zhang[2] | Genghui Jiang[1]

[1]School of Energy and Power Engineering, Nanjing University of Science and Technology, Nanjing, China

[2]School of Engineering, Westlake University, Hangzhou, China

Correspondence

Prof. Cheng Cheng, School of Energy and Power Engineering, Nanjing University of Science and Technology, Nanjing 210094, China.

Email: chengcheng@njust.edu.cn

Funding information

National Natural Science Foundation of China, Grant/Award Number: 11972194

Abstract

The drag coefficient, as the most important parameter that characterizes particle dynamics in flows, has been the focus of a large number of investigations. Although good predictability is achieved for simple shapes, it is still challenging to accurately predict drag coefficient of complex-shaped particles even under moderate Reynolds number (*Re*). The problem is that the small-scale shape details of particles can still have considerable impact on the drag coefficient, but these geometrical details cannot be described by single shape factor. To address this challenge, we leverage modern deep-learning method's ability for pattern recognition, take multiple shape factors as input to better characterize particle-shape details, and use the drag coefficient as output. To obtain a high-precision data set, the discrete element method coupled with an improved velocity interpolation scheme of the lattice Boltzmann method is used to simulate and analyze the sedimentation dynamics of polygonal particles. Four different machine-learning models for predicting the drag coefficient are developed and compared. The results show that our model can well predict the drag coefficient with an average error of less than 5% for particles. These findings suggest that data-driven models can be an attractive option for the drag-coefficient prediction for particles with complex shapes.

KEYWORDS

drag coefficient, arbitrary polygonal particle, shape factor, DEM–LBM, machine Learning

1 | INTRODUCTION

Particle-fluid systems are widely applied in various fields, including chemical engineering (fluidization), civil engineering (sedimentation), and propulsion technology.[1] The system consists of particles with diverse polygonal shapes or geometries that are even more complex. The properties and behaviors of these particles are pivotal to the entire system. When studying the mechanisms of particle-fluid systems, the drag coefficient (C_d) is a fundamental parameter used to characterize the interactions between particles and fluids.

For the drag coefficient of spherical particles, Stokes proposed the first theoretical solution: for $Re \ll 1$:$C_d = 24/Re$. In the range of $2000 < Re < 30000$, the drag coefficient of the sphere is approximately a constant value of 0.45. Notably, Clift et al.,[2] Khan and Richardson,[3] and other researchers, have conducted extensive investigations into the sedimentation behavior of spherical particles influenced by gravity through both experimental studies[4–6] and numerical simulations.[7,8]

Previous research has shown that shape has a significant impact on the drag coefficient of particles,[4,9,10] which is difficult to

determine. There is a need for comprehensive tools to model the particle geometry and characterize particle dynamics in fluids. In recent years, the discrete element method (DEM)–lattice Boltzmann method (LBM) has been widely used and has shown accurate results of sedimentation and flow for complex particles.[11–13] The DEM employs the Lagrangian approach to track particle motion in space and time, whereas LBM acts as an efficient solver for Navier–Stokes equations owing to its inherently kinetic nature, particularly beneficial when dealing with intricate fluid–solid boundary conditions.[14] There are two categories of approaches to coupling DEM and LBM: diffusion-interface approach and sharp interface approach.[15] Although the diffusion-interface method can achieve a smooth transition between solid and fluid nodes, reducing hydrodynamic fluctuations, its nonphysical diffusion interface representation limits its accuracy. Previous research shows that the sharp interface approach is second-order accurate for the velocity, hydrodynamic force/torque, and stress, whereas the diffusion-interface approach only holds first-order accuracy when simulating laminar flows. The most simple interface approach is the interpolated bounce-back scheme (IBB).[16] However, the accuracy of this scheme decreases in some situations because it approximates interfaces as step-like boundaries.

In addition to precise modeling techniques, building the correlation between particle shape and the drag coefficient is also a challenging task. In the past, researchers attempted to investigate the influence of shape on the drag coefficient using conventional approaches and obtain fitting equations based on experimental data.[17,18] As shown in Table 1, shape factors were added to the drag coefficient equation to establish the relationship between shape and drag coefficient. Haider and Levenspiel[4] proposed a general drag coefficient equation applicable to both spherical and nonspherical particles. Dioguardi et al.[21] studied the drag coefficient of nonspherical particles by utilizing the ratio of the equivalent diameter of a sphere to the characteristic length of the particle. Yow et al.[20] established an explicit equation describing the influence of sphericity and *Re* on the drag coefficient of particles based on experimental data from particles with shapes such as spheres, cubes, and octahedra.

However, these studies are limited to specific or a few types of particles. Therefore, there is a need for developing a method that can broaden the range and diversity of particles studied to ensure a higher level of accuracy. In the past few decades, machine learning has emerged as a robust data processing framework and was applied in various fields such as unmanned vehicles,[22] image detection,[23] and natural language processing.[24] Notably, machine learning, particularly neural networks, has been successfully employed in fluid dynamics and flow modeling in recent years.[25,26] As shown in Table 2, several researchers have demonstrated that the process of particle sedimentation can be predicted by machine learning. Yan et al.[28] employed artificial neural network (ANN) and radial basis function neural network (RBFNN) to predict drag coefficients. The results demonstrated that ANN can be used to establish the relationship between a single shape factor (sphericity) and the drag coefficient.

The main goal of this article is to develop a generalized drag prediction framework for arbitrary polygonal particles. As shown in Figure 1, the framework can effectively take into account the flow effect and particle shape. To accurately model the particles and precisely simulate the sedimentation dynamics, the data set used for training is generated by the improved DEM–LBM. The remaining sections of the article are organized as follows. In Section 2, a brief introduction is provided for DEM–LBM, methods for constructing arbitrarily shaped particles, and machine learning. In Section 3, the settling behavior of particles is analyzed in a two-dimensional system. Machine learning models based on the generated data set are developed and compared. The conclusions drawn from the analysis of this study are presented in Section 4.

2 | METHODOLOGY

2.1 | DEM–LBM

In this section, the DEM–LBM is introduced for simulating the sedimentation of arbitrary polygonal particles. The fluid flow is simulated using the lattice Boltzmann equation (LBE), which is a discretized representation of the Boltzmann equation.[31–33] The D2Q9 lattice model, as shown in Figure 2, is used and the discrete velocities can be written as

$$\boldsymbol{e}_{\alpha} = \begin{cases} C\left[\cos(\alpha-1)\frac{\pi}{4}, \sin(\alpha-1)\frac{\pi}{4}\right], & \alpha = 1,3,5,7, \\ \sqrt{2}C\left[\sin(\alpha-1)\frac{\pi}{4}, \cos(\alpha-1)\frac{\pi}{4}\right], & \alpha = 2,4,6,8, \\ (0,0), & \alpha = 0, \end{cases} \tag{1}$$

TABLE 1 Empirical equations of nonspherical particles.

Author	Equations
Haider et al.[4]	$C_d = \frac{24}{Re}[1 + [8.1716\exp(-4.0655\phi)] \times Re^{0.0964+0.5565\phi}] + \frac{73.69Re\exp(-5.0748\phi)}{Re + 5.378\exp(6.2122\phi)}$
Tran-Cong et al.[19]	$C_d = \frac{24}{\left(\frac{d_A}{d_n}Re\right)}\left[1 + \frac{0.15}{\sqrt{c}}\left(\frac{d_A}{d_n}Re\right)^{0.687}\right] + \frac{0.42\left(\frac{d_A}{d_n}\right)^2}{\sqrt{c}\left[1 + 4.25 \times 10^4\left(\frac{d_A}{d_n}\right)^{-1.16}\right]}$
Yow et al.[20]	$C_d = \frac{15.21 + \frac{10.82}{\phi} - \frac{0.14}{\phi^2}}{Re} + \frac{13.41 - \frac{10.64}{\phi} - \frac{0.06}{\phi^2}}{Re} - 8.82 + \frac{5.70}{\phi} + \frac{0.23}{\phi^2}$

TABLE 2 Previous research of particles based on machine learning.

Author	Year	Main contribution
Yoon et al.[27]	2013	Predict the time-dependent sediment suspension by artificial neural network (ANN).
Yan et al.[28]	2019	Use ANN to build the correlation of sphericity and drag coefficient.
Rushd et al.[29]	2021	Toward optimal machine learning model for terminal settling velocity.
Let et al.[30]	2023	Predict the minimum elutriation of the velocity of the binary solid mixture by GA-ANN.

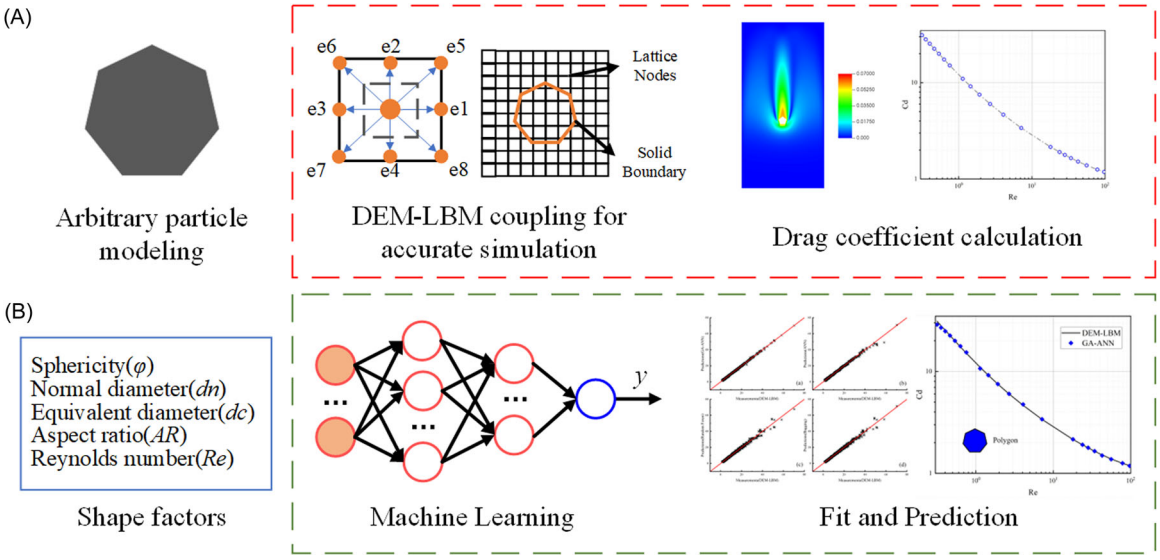

FIGURE 1 Flowchart of this work: (A) discrete element method (DEM)–lattice Boltzmann method (LBM) and (B) machine learning.

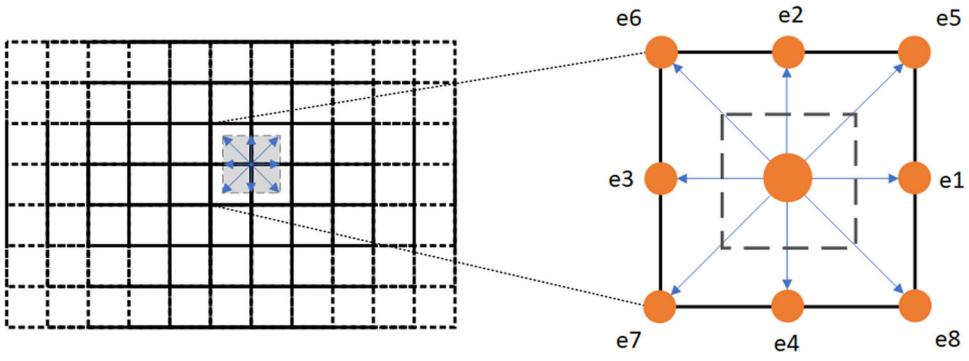

FIGURE 2 Discrete velocity vectors for D2Q9.

where C is the lattice speed parameter to be selected, defined as $\Delta x/\Delta t$ (Δx is the lattice size and Δt is the time step). An evolution rule[34] for each distribution function which is updated as

$$f_i(\boldsymbol{x} + \boldsymbol{e}_i\Delta t, t + \Delta t) = f_i(\boldsymbol{x}, t) + \Omega_{\text{col}}, \tag{2}$$

where f_i is the probability distribution function, $\boldsymbol{x}$ is the position of the local lattice, and Ω_{col} is the collision operator, here a single relaxation time (SRT) collision model is used and is given as

$$\Omega_{\text{col}} = \frac{\Delta t}{\tau}\left(f_i^{\text{eq}} - f_i\right), \tag{3}$$

where τ is the relaxation time and f_i^{eq} is the equilibrium distribution given by

$$f_i^{eq} = \omega\rho\left(1 + \frac{\boldsymbol{e}_i \cdot \boldsymbol{u}}{C^2} + \frac{9(\boldsymbol{e}_i \cdot \boldsymbol{u})^2}{2C^4} - \frac{3\boldsymbol{u}^2}{2C^2}\right), \tag{4}$$

where $\omega_0 = 4/9$, $\omega_{1-4} = 1/9$, and $\omega_{5-8} = 1/36$. Based on the Chapman–Enskog expansion, the LBE can be recovered to the Navier–Stokes equation. More details can be found in Ref. 34. The macroscopic variables such as density and velocity vector can be calculated, respectively, using the following equations:

$$\rho(\boldsymbol{x}) = \sum_{i=0}^{8} f_i(\boldsymbol{x}), \tag{5}$$

$$\boldsymbol{u}(\boldsymbol{x}) = \frac{1}{\rho(\boldsymbol{x})}\sum_{i=0}^{8} f_i(\boldsymbol{x})\boldsymbol{e}_i, \tag{6}$$

DEM is a method proposed by Cundall and Strack[35] that directly solves the particles' motion. Newton's equations and angular momentum conservation equations, respectively, govern the translation and rotation of particles. These equations are as follows:

$$\begin{cases} m_i\boldsymbol{a}_i = m_i\boldsymbol{g} + \sum_{j=0}^{N-1}\boldsymbol{F}_{ij}^{c} + \boldsymbol{F}_i^{h}, \\ \frac{d}{dt}(\boldsymbol{I}_i\boldsymbol{\omega}_i) = \sum_{j=0}^{N-1}\boldsymbol{T}_{ij}^{c} + \boldsymbol{T}_i^{h}, \end{cases} \tag{7}$$

where N represents the number of particles, m_i and $\boldsymbol{a}_i$ denote, respectively, the mass and acceleration of particle i. The forces acting include the gravitational force $m_i\boldsymbol{g}$, the hydrodynamic force $\boldsymbol{F}_i^h$, and the contact force $\boldsymbol{F}_{ij}^c$. $\boldsymbol{I}_i$ represents the inertia tensor, and $\boldsymbol{\omega}_i$ is the angular velocity vector. $\boldsymbol{T}_{ij}^c$ and $\boldsymbol{T}_i^h$ represent torques due to the contact and hydrodynamic forces, respectively.

To simulate fluid–particle interactions, no-penetration and nonslip boundary conditions are enforced at the fluid–solid interface, along with a calculation of hydrodynamic forces acting on particles. The commonly utilized approach for this purpose is the IBB scheme. The scheme divides the nodes involved as shown in Figure 3. The fundamental concept involves interpolating the absent distribution functions based on the available ones. The interpolation weights depend on the distance q defined as

$$q = \frac{\|\boldsymbol{x}_f - \boldsymbol{x}_w\|}{\|\boldsymbol{x}_f - \boldsymbol{x}_s\|}, \tag{8}$$

Here, $\boldsymbol{x}_w$ represents the point of intersection between the solid surface and discrete velocity, $\boldsymbol{x}_f$ denotes the fluid node, and $\boldsymbol{x}_s$ is the neighbor solid node. However, the conventional IBB schemes do not always ensure nonslip conditions at solid surfaces. This will cause a slipping error of settling velocity. To reduce the slipping error, an improved velocity interpolation-based bounce-back scheme (VIBB) introduced by Zhang et al.[36] is used in this research. The missing distribution function $f_i(\boldsymbol{x}_f, t + \Delta t_{LBM})$ is given as

$$f_i(\boldsymbol{x}_f, t + \Delta t_{LBM}) = f_{i'}^{+}(\boldsymbol{x}_d, t) + 6\omega_{i'}\rho_f\frac{\boldsymbol{e}_i \cdot \boldsymbol{u}_w}{C^2}, \tag{9}$$

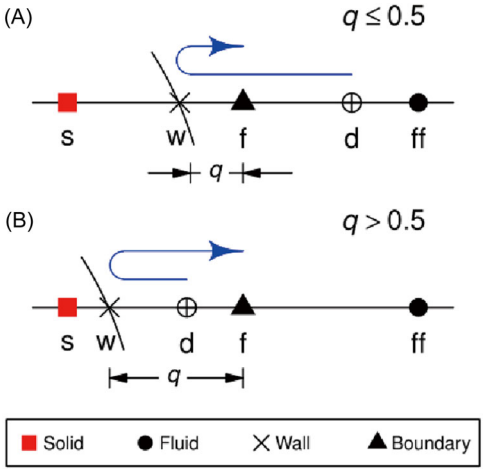

FIGURE 3 Discrete velocity vectors for D2Q9.[36] (A) $q \le 0.5$ and (B) $q > 0.5$.

where $f_{i'}^{+}$ is given by

$$f_{i'}^{+}(\boldsymbol{x}_d, t) = f_{i'}^{eq}(\rho_d, \boldsymbol{u}_d) + f_{i'}^{neq}(\boldsymbol{x}_d, t), \tag{10}$$

f_i^{neq} and ρ are interpolated with second-order accuracy as follows:

$$\begin{aligned} f_{i'}^{neq}(\boldsymbol{x}_d, t) = {} & 2q\left(f_{i'}^{+}(\boldsymbol{x}_f, t) - f_{i'}^{eq}(\boldsymbol{x}_f, t)\right) \\ & + (1 - 2q)\left(f_{i'}^{+}(\boldsymbol{x}_{ff}, t) - f_{i'}^{eq}(\boldsymbol{x}_{ff}, t)\right), \end{aligned} \tag{11}$$

The density at $\boldsymbol{x}_d$ can be evaluated as

$$\rho_d = \begin{cases} 2q\rho_f + (1 - 2q)\rho_{ff}, & q \leqslant 0.5, \\ \rho_f, & q > 0.5. \end{cases} \tag{12}$$

$\boldsymbol{u}_d$ plays a crucial role in determining unknown distributions, and it can be evaluated using the following linear interpolation:

$$\boldsymbol{u}_d^{*} = \begin{cases} 2q\boldsymbol{u}_f + (1 - 2q)\boldsymbol{u}_{ff}, & q \leqslant 0.5 \\ \frac{1 - q}{q}\boldsymbol{u}_f + \frac{2q - 1}{q}\boldsymbol{u}_w, & q > 0.5, \end{cases} \tag{13}$$

$$\boldsymbol{u}_d^{**} = \frac{1 - q}{1 + q}\boldsymbol{u}_{ff} + \frac{2q}{1 + q}\boldsymbol{u}_w, \tag{14}$$

$\boldsymbol{u}_d$ is determined by weighted averaging $\boldsymbol{u}_d^{*}$ and $\boldsymbol{u}_d^{**}$.

The hydrodynamic force and torque that act on the jth particle are given, respectively, as

$$\boldsymbol{F}_j^h = \sum_{i\in\Gamma_j}\left[(\boldsymbol{e}_i - \boldsymbol{u}_w)f_i(\boldsymbol{x}_f, t) - \left(\boldsymbol{e}_i' - \boldsymbol{u}_w\right)f_{i'}(\boldsymbol{x}_f, t)\right], \tag{15}$$

$$\boldsymbol{T}_j^h = \sum_{i\in\Gamma_j}(\boldsymbol{x}_w - \boldsymbol{x}_{pj})[(\boldsymbol{e}_i - \boldsymbol{u}_w)f_i(\boldsymbol{x}_f, t) - (\boldsymbol{e}_{i'} \cdot \boldsymbol{u}_w)f_{i'}(\boldsymbol{x}_f, t)], \tag{16}$$

where Γ_j represents the set of all the discrete velocities that intersect with the jth particle. The VIBB scheme, which combines the ideas of

IBB schemes and nonequilibrium extrapolation scheme, has been proved to capture complex particle shapes with good accuracy.[36,37] It can maintain a sharp interface and re-evaluate the hydrodynamic force from solid nodes to fit the LBM kinetic nature.

2.2 | Shape factors

For particles settling by gravity g in a stationary fluid, the drag coefficient is usually calculated by the following equation:

$$C_d = \frac{4d_n g(\rho_s - \rho_f)}{3\rho_f u_p^2}, \tag{17}$$

where ρ_s and ρ_f denote, respectively, the density of the particle and the fluid. Previous research[4] suggests that a composite consideration of shape factors is essential. However, integrating these shape factors into the equations is not simple. The characteristics of general particle shapes can be considered and described by many shape factors. As in Wachs's work,[38] some shape factors formulations can be simplified. The sphericity ϕ is defined by the following equation:

$$\phi = \frac{C_{circle}}{C}, \tag{18}$$

where C_{circle} and C denote the perimeters of the surface-equivalent circle and the polygon. Sphericity is the most common shape factor used to describe the irregularity of particles. However, from a practical point of view, many particles have the same sphericity and such a parameter does not fully characterize the particle, and it is usually not easy to measure in three dimensions.

The surface-equivalent-sphere diameter can be transformed to

$$d_c = \sqrt{\frac{4l_p}{\pi}}, \tag{19}$$

where l_p is the projected length of the particle.

Ojha et al.[39] introduced the Corey shape factor (CSF), which is directly related to the dimensions of the particle. In the two-dimensional case, this factor simplifies to the aspect ratio (AR), which is the ratio between the shortest and longest axes lengths and is defined as

$$AR = \frac{d_s}{d_l}. \tag{20}$$

The shape factors chosen in this study are relatively empirical, and they are recognized as being well characterized for the drag coefficient of particles. For illustrative purposes, 10 particles with different shapes are shown in Figure 4.

2.3 | Machine learning

2.3.1 | Neural networks

The structure of a neural network can be designed based on specific tasks and requirements. Typically, it comprises three layers: the input layer, the hidden layer(s), and the output layer. Each layer consists of multiple interconnected nodes, also known as neurons. The number of neurons plays a crucial role in improving the performance of ANN. If the number of neurons is too small, the nonlinear predictive performance may be reduced. Conversely, an excessive number of neurons can lead to a decrease in convergence speed, resulting in over-fitting. Each node has a specific output function called an activation function. The connections between the nodes represent the weighted values, called weights, assigned to the signals transmitted through these connections. The output of the network depends on the structural parameters, connection patterns, weights, and activation functions. In general, neural networks are intended to approximate some algorithm or function in nature or to express a logical strategy. ANN and traditional machine learning models focus on training to perform a target task by learning from a large amount of data, which can effectively mine data in complex systems. However, during the modeling process, the precision of ANN algorithms depends greatly on the input layer, hidden layers, initial weights, and thresholds. To compare the effect of training, four machine learning

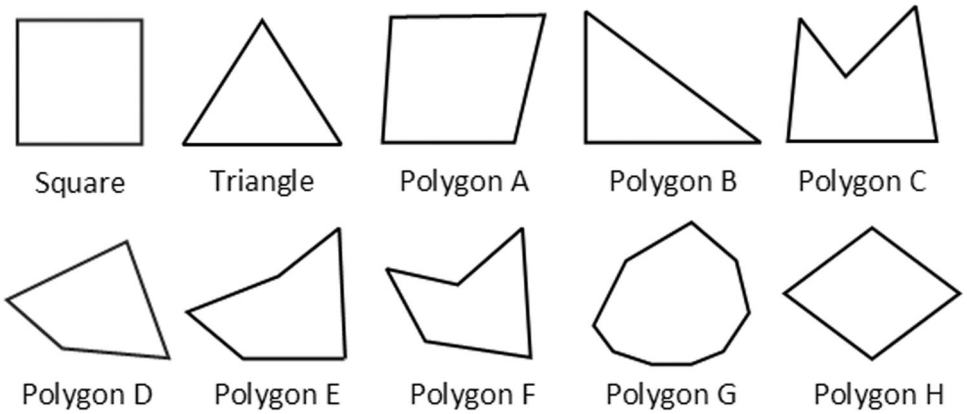

FIGURE 4 Sketch of particles.

models are built (genetic algorithms [GAs]-ANN, ANN, random forest,[40] and bagging[41]).

2.3.2 | Parameter optimization

In the field of machine learning, the selection of appropriate hyperparameters is crucial for enhancing the accuracy of models. Hyperparameters are manually set parameters before training the model, which cannot be learned directly from the data. They define the architecture of the model and control the learning process. Although hyperparameters can be adjusted manually through trial and error, this approach is only suitable for models with a small number of unrelated hyperparameters.

To reduce the task of parameter tuning, automatic optimization methods should be utilized. Decision-theoretic methods such as Grid search represent a straightforward method for hyperparameter optimization. They operate by defining a search space for hyperparameters, then identifying combinations of hyperparameters within this space, and finally selecting the combination that delivers the best performance. However, meta-heuristic algorithms, which are adept at tackling complex, large-search-space, and nonconvex optimization problems, have garnered significant interest from researchers. Among these algorithms, GAs[42] are considered a good choice for hyperparameter optimization. In our research, GAs are employed to optimize the ANN weights and topology.[43] The process of GA is outlined as follows:

(1) Initialize the population. A real-valued coding model is used to randomly generate a set of initial weight vectors.
(2) Construct the fitness function. The inverse of the network squared error is used as the objective function for optimizing the weights.

$$f = \frac{1}{\sum_{i=1}^{n}(\hat{y}_i - y_i)}, \tag{21}$$

(3) Select individuals for crossover and mutation. The probability of selecting an individual is

$$F(x_i) = \frac{f(x_i)}{\sum_{i=1}^{N} f(x_i)}, \tag{22}$$

where N denotes the number of population and n denotes the random number between [0,1].
(4) Select the jth gene m_{ij} of individual i and mutate it with the following equation:

$$m_{ij} = \begin{cases} m_{ij} + (m_{ij} - m_{\max}) \times f(e), & r > 0.5, \\ m_{ij} + (m_{\min} - m_{ij}) \times f(e), & r \le 0.5, \end{cases} \tag{23}$$

$$f(e) = r_2\left(1 - \frac{e}{G_{\max}}\right)^2, \tag{24}$$

$m_{\max}$ and $m_{\min}$ denote the upper and lower bound of m_{ij}, respectively, e is the number of iterations, $G_{\max}$ is its maximum, r indicates the proportion of gene m_{ij} in individual i, and r_2 is a random number.

The individual with the best fitness value is output until the function meets a set value or reaches the maximum number of iterations, thus improving the weight threshold and topology of the neural network. To maximize the accuracy of models other than GA-ANN, grid search was employed to fine-tune them.

2.3.3 | Model evaluation

To statistically assess the accuracy of the prediction model, four different evaluation metrics (mean average error [MAE], mean squared error [MSE], root mean squared error [RMSE], and R^2) are used in this study; they are defined as follows:

$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m}|(y_i - \hat{y}_i)|, \tag{25}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{26}$$

$$\mathrm{MSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}, \tag{27}$$

$$R^2 = 1 - \frac{\sum_i(\hat{y}_i - y_i)^2}{\sum_i(\bar{y}_i - y_i)^2}. \tag{28}$$

where, $\hat{y}_i$ is the predicted value, y_i is the measured value, and $\bar{y}_i$ is the average of the measured values. MAE is the average absolute difference between the actual value and the predicted value, which quantifies the average degree of deviation of the predicted value from the true value. MSE is the average of the sum of squares of the difference between the true value and the predicted value. It is often used as a loss function for linear regression because it is easy to differentiate. RMSE is the square root of the result of the MSE. R^2 is a statistical measure of how accurately a given model predicts a measured outcome; R^2 close to 1 indicates higher agreement between predicted and measured quantities.

3 | RESULTS

3.1 | Validation

A benchmark case about a single circle particle settling in the Newtonian fluid is conducted to validate the DEM–LBM coupling method. As in previous studies,[44] the fluid is assumed to have a viscosity of 0.1 g/(cm · s) and density of 1.0 g/cm^3. The particle has a radius of 0.125 cm and a density of 1.25 g/(cm · s). A uniform mesh of 201 × 601 is used in the current simulation. Considering the effect of gravity, the particle moves downward. As shown in Figure 5, the variation of Re over time during the particle settling process was compared with the results from Wu and Shu.[45] The results are consistent and show good agreement. The maximum Re of the particle in the present computation is 17.12, which is close to 17.15 provided by Wan and Turek.[46]

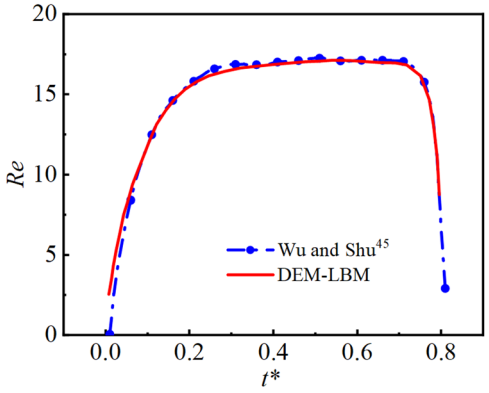

FIGURE 5 Validation of a single circular particle settling. DEM, discrete element method; LBM, lattice Boltzmann method.

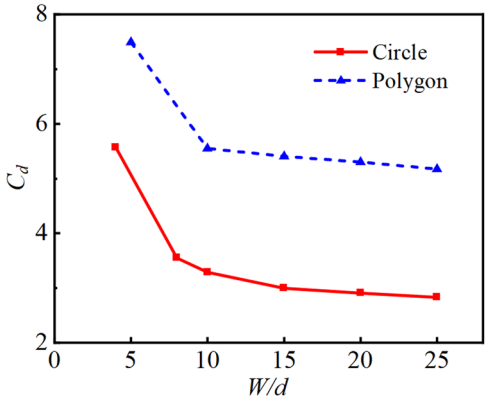

FIGURE 6 Drag coefficient of particles at different channel widths.

Given our objective of approximating the settling drag coefficient of polygonal particles, it is crucial to consider the wall effect. Wicke and Hedden[47] pointed out that the wall effect is negligible for fluidization if the wall width to particle diameter ratio is more than 10. However, for irregular shapes, this ratio may need to be higher. In fact, if W/D (the ratio of the computational domain W to the particle diameter D) is too small, then there are other factors that affect the settling of particles. One is that the particles are likely to hit the wall at higher Re due to lateral forces, which can lead to a reduction in the settling velocity, thus preventing proper observation of the correct settling behavior of the particles. Second, the location of the initial release of the particles is particularly important when the W/D ratio is too small, which may lead to different results. To control the effects of wall width on the drag coefficient, simulations were conducted based on the same conditions as described in Feng et al.[48] Specifically, the channel widths are varied and the drag coefficients are calculated for different values of W/D, including 4, 8, 10, 15, 20, and 25.

The results are presented in Figure 6, which clearly shows that increasing the wall width leads to a decrease in the drag coefficient of the particles. This phenomenon can be attributed to a reduction in the wall effect, resulting in an increased terminal settling velocity. Finally,

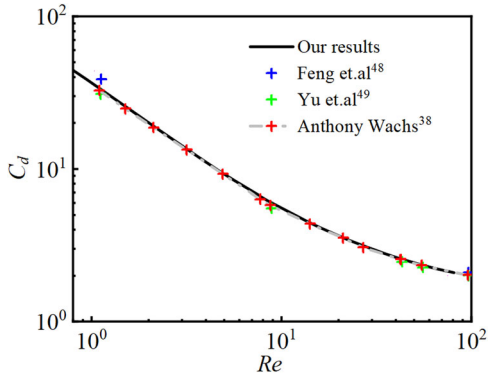

FIGURE 7 Comparison of drag coefficient for a single circle settling: $W/D = 4$.

as shown in Figure 7, the accuracy of our drag coefficient calculations was validated. The results align with those reported by previous researchers,[49] further supporting the accuracy of our simulations.

Moreover, for particles with complex shapes, the time required to reach steady-state sedimentation is typically greater than that of spherical particles, implying a need for a longer length of channel.

3.2 | Particle settling dynamics

Considering the computational efficiency, the case in which W/D is 20 and L/D is 60 is chosen. The radius of polygonal particles is around 0.0075 m. In settling problems, the particle's sedimentation motion is driven by the density difference with the fluid. The particle density ρ was set to 1.12 g/cm^3; The fluid density, dynamic viscosity, and other parameters are set according to Ten Cate et al.[50] and Zhang et al.[17] Wall boundary conditions were applied at the boundaries. To ensure acceptable accuracy, the diameter of the particles should be larger than nine LBM cells. Note that in this work, the gravity given by $(1 - \rho_f/\rho_s)g$ is used, as suggested by Feng and Michaelides.[51] Our study focuses on the stable settling regime (Re < 100). This decision serves a dual purpose: to maintain the stability of LBM numerical simulations and to enhance computational efficiency through the reduction of the channel length.

Figure 8 provides a visualization of the flow field around the particles, with the color map representing the magnitude of fluid velocity. When a particle undergoes rotation about the horizontal axis, it experiences significant influences from horizontal forces and exhibits a spiral trajectory, particularly at higher Re. This spiral motion continues until the particle's maximum cross-section becomes perpendicular to the settling direction.

In the case of low viscosity, the terminal velocity of particles may exhibit oscillations and display unstable behaviors. This phenomenon arises from the pressure gradient at the front end of the particle generating torque, which ultimately leads to an increase in the final settling velocity.

It should be noted that the settling velocity of particles, as shown in Figure 9, can manifest in three different scenarios. First, the

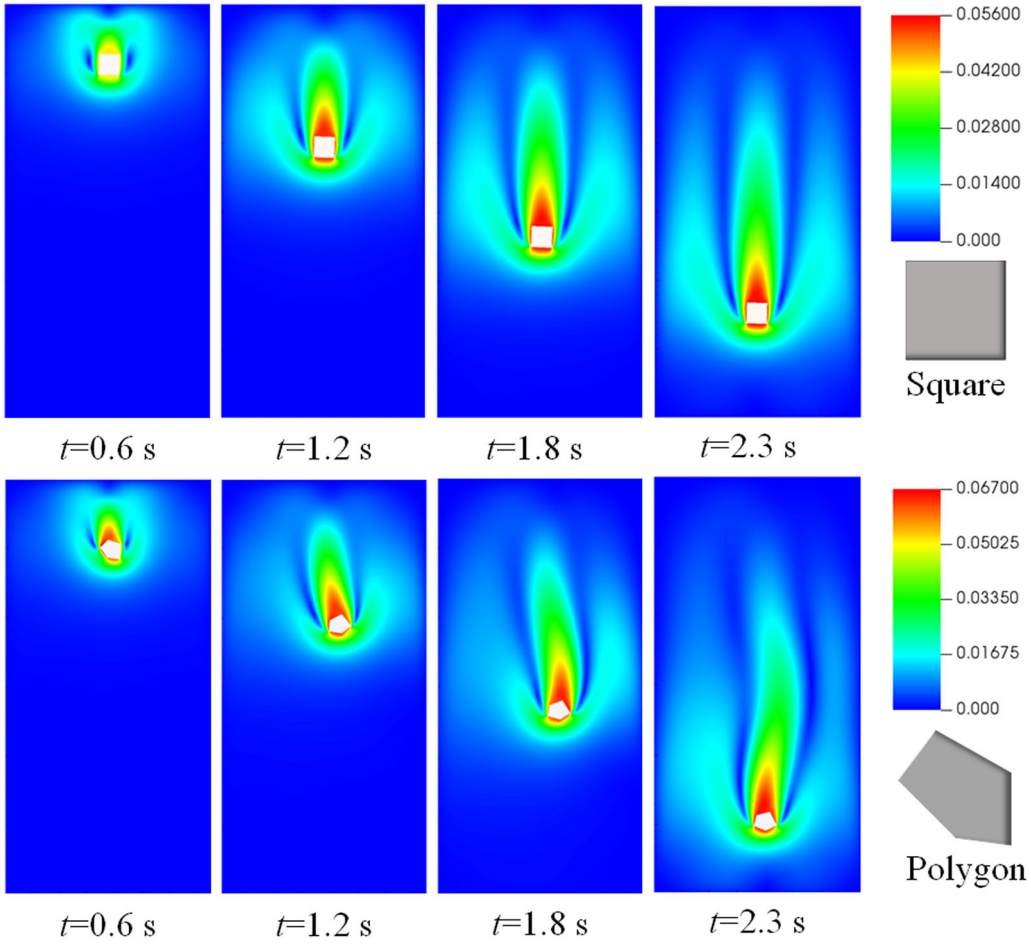

FIGURE 8 Contour plots of two settling particles: fluid velocity field.

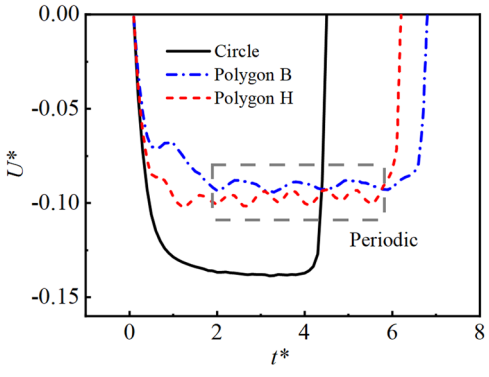

FIGURE 9 Three different types of terminal velocity.

velocity gradually increases and stabilizes, typically occurring at lower *Re*. Second, the settling velocity briefly exceeds the final settling velocity before fluctuating around it. Third, the settling velocity fluctuates around the final settling velocity and fluctuates around the final settling velocity. The second and third scenarios often arise when the particle configuration is unstable or when operating at higher *Re*. To obtain the drag coefficient, the average velocity over this period was artificially considered as the representative final settling velocity, which may introduce certain errors.

Figure 10 shows the variation of settling velocities for the square particle released at different angles at *Re* = 10. Under these conditions, the square particle exhibits the second type of settling, where the settling velocity changes due to the change in orientation during the sedimentation process.

Figure 11 shows the trajectories of some particles during their settling motion at *Re* = 10. It can be seen that particles with symmetrical shapes have smoother trajectory motion. Even at the same *Re*, irregular particles exhibit unstable settling and tend to stick to the wall. The drag coefficient curves for certain noncircular particles were computed in a two-dimensional scenario. The results clearly indicate that, regardless of whether it is a two- or three-dimensional case, the irregularity of particles leads to an increase in their drag coefficients within the fluid. In the Stokes regime, where viscous forces dominate, there is a simple inverse relationship between the drag coefficient and *Re*. The results of the drag coefficients for particles with different shapes in the Stokes regime are shown in Figure 12. It can be observed that the particle configurations are stable, and the particles maintain a stable settling posture. However,

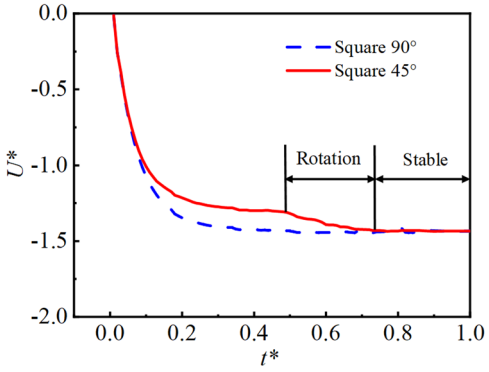

FIGURE 10 Time series of the square particle released at different angles.

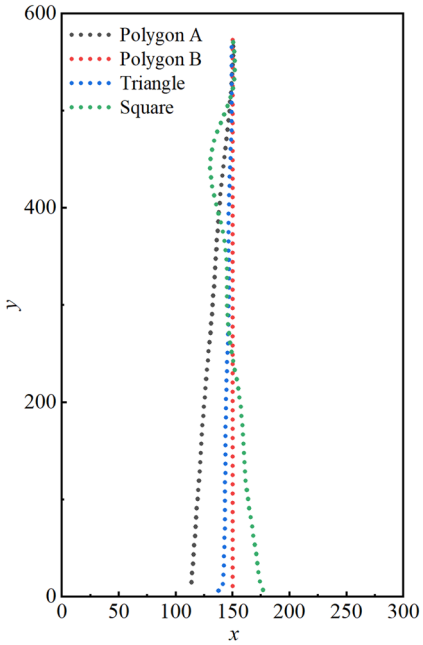

FIGURE 11 Trajectory of particles.

as shown in Figure 13, when $Re > 1$, in the intermediate Re range where both viscous and inertial forces are significant, the relationship between the drag coefficient and Re cannot be described by a simple linear proportion. Furthermore, for some particles, their configurations may change (leading certain deviations) at higher Re, which also contributes to the variation in the drag coefficients.

3.3 | Training process

The drag coefficient curves for over 100 particles were computed as the data set for our ML model.

The mesh information and shape factors of particles built can be found in Table 3. The sphericity distribution of the particles established in this study is shown in Figure 14. To control the variables, all particles had the same projected length on the x-axis.

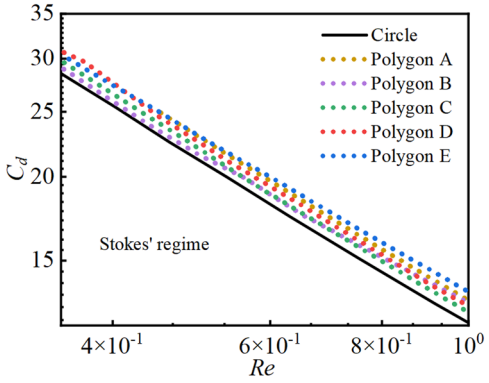

FIGURE 12 Drag coefficient for different shapes: Stokes regime.

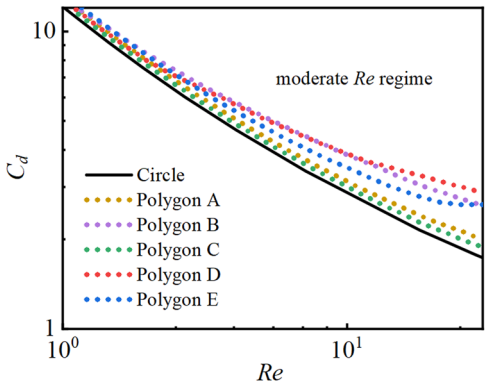

FIGURE 13 Drag coefficient for different shapes: moderate Re regime.

During the data preparation stage, normalizing the data obtained from DEM–LBM simulations is essential. Data normalization addresses issues related to scale units, range, and the inherent weight effects of input variables. By normalizing the data, incompatibility issues caused by these factors can be eliminated and ensure a more accurate and efficient training process for the machine learning model.

To achieve good training and generalization performance of the model, it is necessary to split the data set into training and testing sets. For model training, 70% of the data is utilized, while the remaining 30% is allocated for testing purposes. This allocation allowed for a sufficient amount of data to train the model effectively while also reserving a reasonable amount of data to test its performance. The chosen ratio strikes a balance between ensuring enough data for training and validating the model's ability to generalize to unseen data. For the GA-ANN model, the hyperparameters are determined by the GA: The population size is configured to be 40, the arithmetic crossover probability is set at 0.8, and the nonuniform mutation probability is defined as 0.01. Additionally, the number of generations is specified as 100. If the number of generations is too small, convergence may be difficult to achieve. Conversely, if the number of generations exceeds a

TABLE 3 Particle information and shape factors.

Type	Parameters	Min	Max
	Edges (n)	3	15
Input	Sphericity (ϕ)	0.593	1
	Aspect ratio (AR)	0.400	2.02128
	Diameter ratio (d_n/d_c)	1.858	2.545
	Reynolds number (Re)	0.115	124.18
Output	Drag coefficient (C_d)	1.127	86.44102

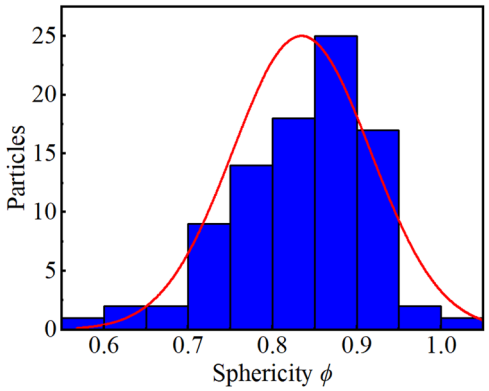

FIGURE 14 Sphericity distribution.

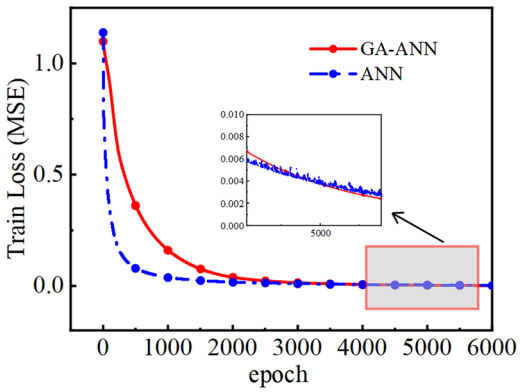

FIGURE 15 Loss curve of ANN and GA-ANN. ANN, artificial neural network; GA, genetic algorithm.

TABLE 4 Performances of different machine learning models on data set.

Model	MAE	MSE	RMSE	R^2
GA-ANN	0.0164	0.0016	0.0397	0.9983
ANN	0.0374	0.0050	0.0710	0.9948
Random forest	0.0545	0.0098	0.0991	0.9899
Bagging	0.0468	0.0090	0.0951	0.9907

Abbreviations: ANN, artificial neural network; MAE, mean average error; MSE, mean squared error; RMSE, root mean squared error.

certain threshold, the algorithm may have already converged. Continuing with additional evolution would be pointless and it only increases time and computation resources.

The hyperparameters of the other models are determined by grid search. The ANN model undergoes training with the Adam optimizer, configured with a batch size of 10 and a network architecture that includes two hidden layers, each with 15 neurons. The learning rate is set to 0.001, and the decay rate is 0.7. Using the parameter α for regularization (L2 regularization) helps to prevent over-fitting by penalizing large weights. Figure 15 illustrates the MSE loss curves of GA-ANN and ANN, demonstrating that both models have achieved convergence. Generally, a standard ANN model tends to converge faster than the GA-ANN. In the case of ANN, the weights and biases of the neural network are updated using a backpropagation algorithm or other optimization methods to minimize the loss function. This approach offers higher computational efficiency and convergence speed, enabling the loss curve of ANN to reach lower loss values more rapidly. On the other hand, GA-ANN combines GAs with ANN. In each genetic generation, GA-ANN optimizes the weights and biases of the neural network using GAs. As GAs involve operations such as selection, crossover, and mutation, and may require multiple iterations of genetic generations for search, their optimization process is relatively slower. Consequently, the loss curve of GA-ANN typically begins to exhibit a significant decline after a greater number of epochs. Despite ANN achieving faster convergence compared to GA-ANN, GA-ANN attains a smaller value of MSE.

3.4 | Prediction results

Four different evaluation metrics (MSE, RMSE, MAE, and R^2) were employed to quantify the performance of the models, and the specific performance indicators for each model are shown in Table 4. The results indicate that, compared with those of traditional models, all four evaluation metrics of the neural network were improved. Figure 16 displays the comparison between the predicted values and the actual values of the test set. When data points fall on the reference line, it indicates a better fit, whereas deviations on either side of the reference line indicate biases. In the drag coefficient fitting, traditional machine learning models exhibit an MAE of approximately 5%, which is larger than that of neural network models.

In the Stokes region, all prediction models except GA-ANN exhibit some degree of deviation, potentially due to data set biases. Variations in the settling behavior of different particles cause significant differences in the drag coefficients at lower Re. For models with poor generalization ability, this can result in obvious prediction errors or over-fitting. By combining the GA and ANN to optimize the initial weights and thresholds and automatically select

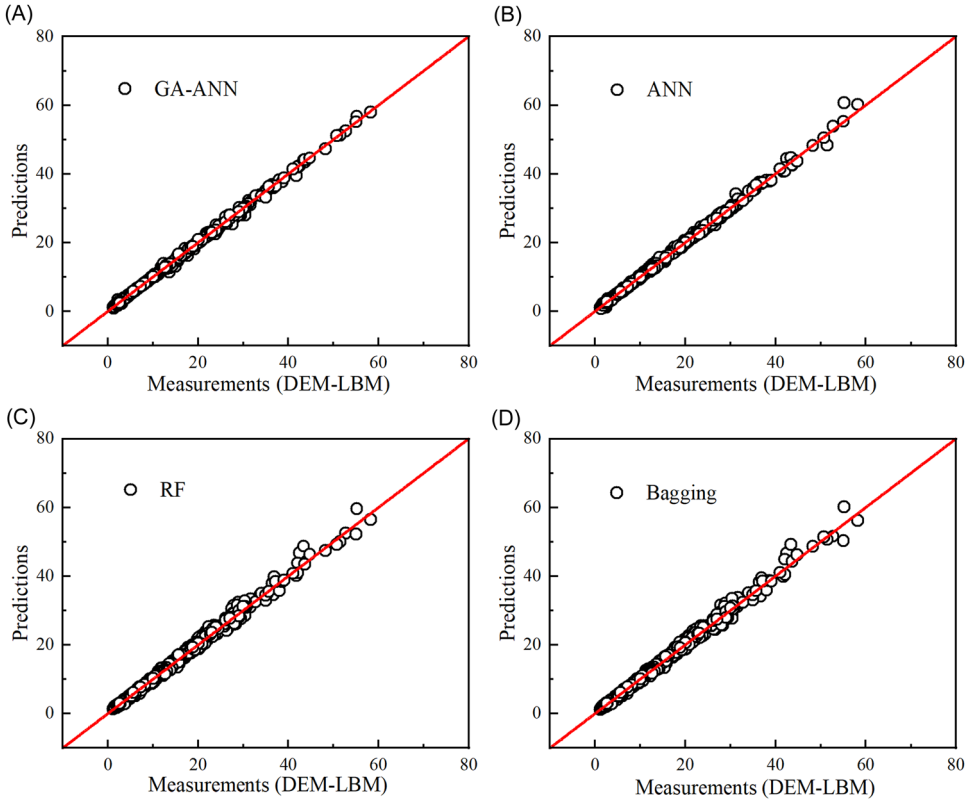

FIGURE 16 Comparison between DEM–LBM results and predictions from machine learning: (A) GA-ANN; (B) ANN; (C) random forest regression; and (D) bagging regression. ANN, artificial neural network; DEM, discrete element method; GA, genetic algorithm; LBM, lattice Boltzmann method.

the number of neurons in the hidden layer, the accuracy can be significantly improved. Compared with unoptimized models, the optimized models show a significant reduction in the error between the predicted and actual values. Significant over-fitting is eliminated, and the optimized model's generalization ability is greatly enhanced, indicating more accurate predictions. On the other hand, the traditional machine learning models (random forest and bagging) have larger training errors and lower accuracy compared with the neural network model.

In summary, GA-ANN can predict the drag coefficient of particles with different shapes. Figure 17 shows the results of simulation and prediction results of GA-ANN. The prediction error of results is within acceptable limits. For the sake of readability, not all curves are displayed. The predictions from GA-ANN for shapes with existing research were compared with numerical calculations from previous studies,[52–57] and these comparisons demonstrated good agreement.

4 | CONCLUSIONS

Obtaining the drag coefficient accurately is crucial for particle fluid dynamics. However, the drag coefficient for particles with complex shapes is influenced by small-scale shape details and cannot be described by a single shape factor. To address this issue, this article developed a generalized framework based on machine learning for arbitrary polygonal drag coefficient prediction. The framework considers key sedimentation process parameters, including multiple shape factors and *Re*. The main conclusions are as follows.

An improved coupling velocity interpolation scheme was applied to construct the DEM–LBM code for simulating the sedimentation dynamics of polygonal particles. This scheme reduced the slipping error and improved the accuracy compared with the simple IBB scheme. The results demonstrate good accuracy and stability, aligning well with the existing research.

The prediction results based on the generated data set indicate that the framework effectively captures the effects of shape and flow, establishing a relationship between the shape factors, *Re*, and drag coefficient, with an error less than 5%.

We further compared four different machine learning prediction models and optimized their hyperparameters using GA and decision-theoretic methods. The findings indicate that the GA-ANN model presents a much better performance (MAE = 0.0164, MSE = 0.0016, RMSE = 0.0397, and R^2 = 0.9983). This confirms the effectiveness of GA-ANN in developing a predictive model for the drag coefficient of particles.

However, the current study also has some limitations, such as the selection of the shape factors being empirical. Future studies will

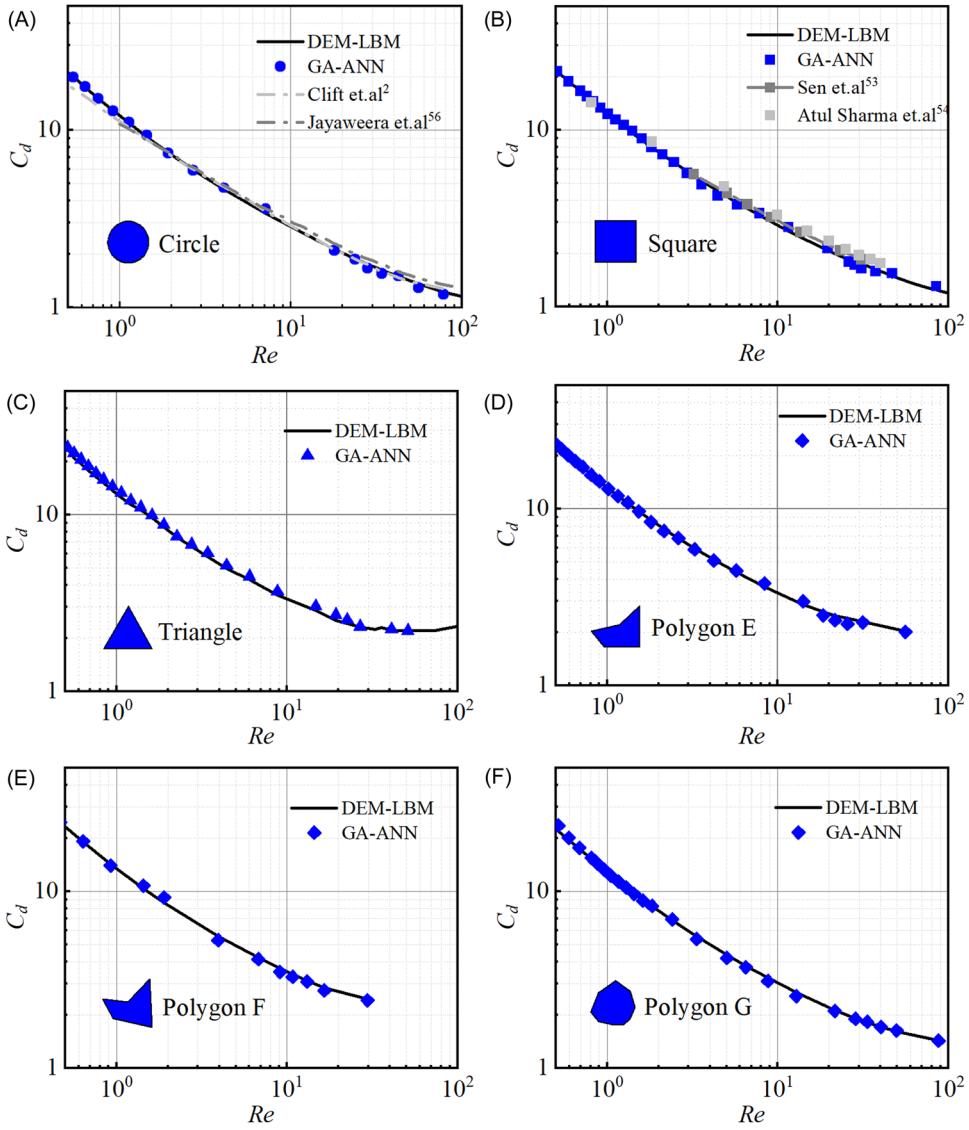

FIGURE 17 Drag coefficient prediction by GA-ANN: (A) circle; (B) square; (C) triangle; (D) polygon E; (E) polygon F; and (F) polygon G. ANN, artificial neural network; DEM, discrete element method; GA, genetic algorithm; LBM, lattice Boltzmann method.

focus on three-dimensional analysis and will take into account more characteristic parameters.

ACKNOWLEDGMENTS

The software used for all the simulations presented in this work is based on the source library ComFluSoM (URL: https//github.com/peizhang-cn/ComFluSoM) developed by the third author. This study was funded by the National Natural Science Foundation of China, Grant/Award Number: 11972194.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

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

ORCID

Cheng Cheng http://orcid.org/0000-0003-2286-3977

REFERENCES

1. Wang G, Tao L, Rong B, Rui X. Review of dynamics simulation methods for multi-field coupling systems. *Adv Mech*. 2023;53(2):468-495.
2. Clift R, Grace J, Weber M. Bubbles, drops, and particles. *Dry Technol*. 1978;11:263-264.
3. Khan AR, Richardson JF. The resistance to motion of a solid sphere in a fluid. *Chem Eng Commun*. 1987;62(1-6):135-150.
4. Haider A, Levenspiel O. Drag coefficient and terminal velocity of spherical and nonspherical particles. *Powder Technol*. 1989;58(1):63-70.
5. Willmarth WW, Hawk NE, Harvey RL. Steady and unsteady motions and wakes of freely falling disks. *Phys Fluids*. 1964;7(2):197-208.
6. Leith D. Drag on nonspherical objects. *Aerosol Sci Technol*. 1987;6(2):153-161.

7. Hölzer A, Sommerfeld M. Lattice Boltzmann simulations to determine drag, lift and torque acting on non-spherical particles. *Comput Fluids*. 2009;38(3):572-589.
8. Koochi H, Mac Intyre J, Korhonen M, Puisto A, Maleki-Jirsaraei N, Alava MJ. Interaction between a free-falling sphere and structure dynamics in a heterogeneous thixotropic fluid. *Phys Fluids*. 2023;35(8):083108.
9. Wang Z, Xiao Y, Liu J, et al. Investigation on settling behavior of single cuboid-like particle in a quiescent fluid. *Powder Technol*. 2024;439:119713.
10. Qiu Z, Xiao Q, Yuan H, Han X, Li C. Particle shape and clogging in fluid-driven flow: a coupled CFD-DEM study. *Powder Technol*. 2024;437:119566.
11. Zhang H, Yu A, Zhong W, Tan Y. A combined TLBM–IBM–DEM scheme for simulating isothermal particulate flow in fluid. *Int J Heat Mass Transfer*. 2015;91:178-189.
12. Jiao K, Han D, Li J, Yu B. Numerical simulations of polygonal particles settling within non-Newtonian fluids. *Phys Fluids*. 2022;34(7):073315.
13. Wang M, Feng YT, Qu TM, Zhao TT. A coupled polygonal DEM-LBM technique based on an immersed boundary method and energy-conserving contact algorithm. *Powder Technol*. 2021;381:101-109.
14. Cheng C, Galindo-Torres S, Zhang X, Zhang P, Scheuermann A, Li L. An improved immersed moving boundary for the coupled discrete element lattice Boltzmann method. *Comput Fluids*. 2018;177:12-19.
15. Wang M, Feng YT, Owen DRJ, Qu TM. A novel algorithm of immersed moving boundary scheme for fluid–particle interactions in DEM–LBM. *Comput Methods Appl Mech Eng*. 2019;346:109-125.
16. Li X, Ma H, Mei Z, Zhou D, Zhao Y. An approach to determine the solid volume fraction near solid boundaries for partially saturated method within the lattice Boltzmann method. *Comput Fluids*. 2024;271:106166.
17. Zhang P, Galindo-Torres SA, Tang H, Jin G, Scheuermann A, Li L. Lattice Boltzmann simulations of settling behaviors of irregularly shaped particles. *Phys Rev E*. 2016;93(6):062612.
18. Zhou C, Su J, Chen H, Shi Z. Terminal velocity and drag coefficient models for disc-shaped particles based on the imaging experiment. *Powder Technol*. 2022;398:117062.
19. Tran-Cong S, Gay M, Michaelides EE. Drag coefficients of irregularly shaped particles. *Powder Technol*. 2004;139(1):21-32.
20. Yow HN, Pitt MJ, Salman AD. Drag correlations for particles of regular shape. *Adv Powder Technol*. 2005;16(4):363-372.
21. Dioguardi F, Mele D, Dellino P. A new one-equation model of fluid drag for irregularly shaped particles valid over a wide range of Reynolds number. *J Geophys Res Solid Earth*. 2018;123(1):144-156.
22. Luo W, Ebel H, Eberhard P. An LSTM-based approach to precise landing of a UAV on a moving platform. *Int J Mech Syst Dynam*. 2022;2(1):99-107.
23. Jing J, Liu S, Wang G, Zhang W, Sun C. Recent advances on image edge detection: a comprehensive review. *Neurocomputing*. 2022;503:259-271.
24. Dogra V, Verma S, Kavita I, et al. A complete process of text classification system using state-of-the-art NLP models. *Comput Intell Neurosci*. 2022;2022:1-26.
25. Brunton SL, Noack BR, Koumoutsakos P. Machine learning for fluid mechanics. *Annu Rev Fluid Mech*. 2020;52(1):477-508.
26. Huang X, Cheng C, Zhang X. Machine learning and numerical investigation on drag reduction of underwater serial multi-projectiles. *Defence Technol*. 2022;18(2):229-237.
27. Yoon HD, Cox DT, Kim M. Prediction of time-dependent sediment suspension in the surf zone using artificial neural network. *Coastal Eng*. 2013;71:78-86.
28. Yan S, He Y, Tang T, Wang T. Drag coefficient prediction for non-spherical particles in dense gas–solid two-phase flow using artificial neural network. *Powder Technol*. 2019;354:115-124.
29. Rushd S, Parvez MT, Al-Faiad MA, Islam MM. Towards optimal machine learning model for terminal settling velocity. *Powder Technol*. 2021;387:95-107.
30. Let S, Maiti SB, Bar N, Basu RK, Das SK. Elutriation characteristics of binary particle mixture in pseudoplastic liquid: empirical correlation and GA-ANN modelling. *Powder Technol*. 2023;428:118796.
31. Galindo-Torres SA, Scheuermann A, Li L. Numerical study on the permeability in a tensorial form for laminar flow in anisotropic porous media. *Phys Rev E*. 2012;86(4):046306.
32. Galindo-Torres SA, Scheuermann A, Li L, Pedroso DM, Williams DJ. A lattice Boltzmann model for studying transient effects during imbibition–drainage cycles in unsaturated soils. *Comput Phys Commun*. 2013;184(4):1086-1093.
33. Galindo-Torres SA. A coupled discrete element lattice Boltzmann method for the simulation of fluid–solid interaction with particles of general shapes. *Comput Methods Appl Mech Eng*. 2013;265:107-119.
34. Mohamad A. *Lattice Boltzmann method*. Vol 70. Springer; 2011.
35. Cundall PA, Strack ODL. A discrete numerical model for granular assemblies. *Géotechnique*. 1979;29(1):47-65.
36. Zhang P, Qiu L, Chen Y, et al. Coupled metaball discrete element lattice Boltzmann method for fluid-particle systems with non-spherical particle shapes: a sharp interface coupling scheme. *J Comput Phys*. 2023;479:112005.
37. Zhao Y, Zhang P, Lei L, Kong L, Galindo-Torres SA, Li SZ. Metaball-Imaging discrete element lattice Boltzmann method for fluid–particle system of complex morpholo-gies with case studies. *Phys Fluids*. 2023;35(2):023308.
38. Wachs A. A DEM-DLM/FD method for direct numerical simulation of particulate flows: sedimentation of polygonal isometric particles in a Newtonian fluid with collisions. *Comput Fluids*. 2009;38(8):1608-1628.
39. Swamee PK, Ojha CSP. Drag coefficient and fall velocity of non-spherical particles. *J Hydraul Eng*. 1991;117(5):660-667.
40. Breiman L. Random forests. *Mach Learn*. 2001;45:5-32.
41. Breiman L. Bagging predictors. *Mach Learn*. 1996;24:123-140.
42. Forrest S. Genetic algorithms. *ACM Comput Surv*. 1996;28(1):77-80.
43. Zhang Y, Pan G, Chen B, Han J, Zhao Y, Zhang C. Short-term wind speed prediction model based on GA-ANN improved by VMD. *Renew Energy*. 2020;156:1373-1388.
44. Zhang H, Tan Y, Shu S, et al. Numerical investigation on the role of discrete element method in combined LBM–IBM–DEM modeling. *Comput Fluids*. 2014;94:37-48.
45. Wu J, Shu C. Particulate flow simulation via a boundary condition-enforced immersed boundary-lattice Boltzmann scheme. *Commun Comput Phys*. 2010;7(4):793-812.
46. Wan D, Turek S. Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method. *Int J Numer Methods Fluids*. 2006;51(5):531-566.
47. Wicke E, Hedden SK. Strömungsformen und Wärmeübertragung in von Luft aufgewirbelten Schüttgutschichten. *Chem Ing Tech*. 1952;24(2):82-91.
48. Feng J, Hu HH, Joseph DD. Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part 2. Couette and Poiseuille flows. *J Fluid Mech*. 1994;277:271-301.
49. Yu Z, Phan-Thien N, Fan Y, Tanner RI. Viscoelastic mobility problem of a system of particles. *J Non-Newtonian Fluid Mech*. 2002;104(2-3):87-124.
50. Ten Cate A, Nieuwstad CH, Derksen JJ, Van den Akker HEA. Particle imaging velocimetry experiments and lattice-Boltzmann simulations on a single sphere settling under gravity. *Phys Fluids*. 2002;14(11):4012-4025.

51. Feng ZG, Michaelides EE. The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems. *J Comput Phys*. 2004;195(2):602-628.
52. Saha AK, Biswas G, Muralidhar K. Three-dimensional study of flow past a square cylinder at low Reynolds numbers. *Int J Heat Fluid Flow*. 2003;24(1):54-66.
53. Sen S, Mittal S, Biswas G. Flow past a square cylinder at low Reynolds numbers. *Int J Numer Methods Fluids*. 2011;67(9):1160-1174.
54. Sharma A, Eswaran V. Heat and fluid flow across a square cylinder in the two-dimensional laminar flow regime. *Num Heat Transfer A: Appl*. 2004;45(3):247-269.
55. Pruppacher HR, Clair BPL, Hamielec AE. Some relations between drag and flow pattern of viscous flow past a sphere and a cylinder at low and intermediate Reynolds numbers. *J Fluid Mech*. 1970;44(4):781-790.
56. Schetz JA, Fuhs AE. *Fundamentals of Fluid Mechanics*. John Wiley & Sons; 1999.
57. Jayaweera KOLF, Mason BJ. The behaviour of freely falling cylinders and cones in a viscous fluid. *J Fluid Mech*. 1965;22(4):709-720.

AUTHOR BIOGRAPHIES

Haonan Xiang is a master's candidate at the School of Energy and Power Engineering, Nanjing University of Science and Technology, China. His research interests are fluid dynamics and multiphysics.

Cheng Cheng is a professor at the School of Energy and Power Engineering, Nanjing University of Science and Technology, China. His research interests cover computational fluid dynamics, gas–solid multiphase flow, fluid–structure interaction dynamics. He has published over 30 articles.

Pei Zhang is an associate professor at the School of Energy of Westlake University, China. He is mainly engaged in the development and application of fluid–structure interaction simulation algorithms, with research interests including lattice Boltzmann method and discrete element method. He published eight papers in computational fluid dynamics journals such as *JCP*, *CMAME*, and *Physics of Fluids*.

Genghui Jiang is a lecturer at the School of Energy and Power Engineering, Nanjing University of Science and Technology, China. During his graduate education, he was awarded an MS in Aeronautical and Astronautical Science and Technology (Dalian University of Technology), and a PhD in Power Engineering and Engineering Thermophysics (Shanghai Jiao Tong University). His research interests cover computational mechanics, model reduction method and inverse problems, and so forth. He has published over 10 articles in international peer-reviewed journals within these research areas.

How to cite this article: Xiang H, Cheng C, Zhang P, Jiang G. Machine learning and numerical investigation on drag coefficient of arbitrary polygonal particles. *Int J Mech Syst Dyn*. 2024;4:317-330. doi:10.1002/msd2.12124