Experimentally-trained hybrid machine learning algorithm for predicting turbulent particle-laden flows in pipes

Yang, Zhuangjian; Li, Kun; Barigou, Mostafa

Document Version
Publisher's PDF, also known as Version of record

[Link to publication on Research at Birmingham portal](#)

Experimentally trained hybrid machine learning algorithm for predicting turbulent particle-laden flows in pipes

ZhuangJian Yang (杨庄健) ; Kun Li (李坤) ; Mostafa Barigou ✉

Check for updates

CrossMark

View Online

Export Citation

Experimentally trained hybrid machine learning algorithm for predicting turbulent particle-laden flows in pipes

View Online   Export Citation   CrossMark

ZhuangJian Yang (杨庄健), ⬤ Kun Li (李坤), ⬤ and Mostafa Barigou[a) ⬤

AFFILIATIONS

School of Chemical Engineering, University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom

[a)]Author to whom correspondence should be addressed: m.barigou@bham.ac.uk

ABSTRACT

A hybrid learning algorithm consisting of a preprocessor, a k-nearest neighbors regressor, a noise generator, and a particle–wall collision model is introduced for predicting features of turbulent single-phase and particle–liquid flows in a pipe. The hybrid learning algorithm has the ability to learn and predict the behavior of such complex fluid dynamic systems using experimental dynamic databases. Given a small amount of typical training data, the algorithm is able to reliably predict the local liquid and particle velocities as well as the spatial distribution of particle concentration within and without the limits of the range of training data. The algorithm requires an order of magnitude less training data than a typical full set of experimental measurements to give predictions on the same level of accuracy (typically, 20 cf. 100 trajectories for phase velocity distribution and 40 cf. 500 trajectories for phase concentration distribution), thus leading to huge reductions in experimentation and simulation. A feature importance analysis revealed the effects of the different experimental variables on the particle velocity field in a two-phase particulate flow, with particle–liquid density ratio and particle vertical radial position being the most influential and particle concentration the least. The algorithm is amenable to extension by using more complex databanks to address a much more comprehensive range of flow situations.

I. INTRODUCTION

Machine learning (ML) is an artificial intelligence technique that allows complex systems with vast data to be studied, analyzed, and predicted.[1] Theoretically, it could endow the system with the ability to study and enhance experience automatically without extra human labor. Thus, in the last decade, the world has been witnessing a surge in a wide range of ML applications dealing with a variety of extensive data such as the multimedia industry,[2] image classification,[3] computer vision,[4] social network analysis,[5] text mining,[6] and energy system optimization.[7,8] The basic logic of these applications is to extract insights from the accessible data and, thus, establish intelligent models based on the available information.

Generally, ML algorithms can be classified into four main types: supervised, unsupervised, semi-supervised, and reinforcement learning methods.[9] By providing labeled training data to a supervised learning algorithm, it could predict specific targets from the input.[10] For example, to efficiently classify or regress targets that match the requirements and hobbies of customers, various automated recommender systems (e.g., goods, news, songs, videos, and movies) associated with different trained ML algorithms are established.[11] In healthcare, it has been applied to diagnose diseases with promising accuracy.[12] For example, an artificial neural network algorithm trained by a given database (myocardial perfusion scintigraphy data) could classify myocardial infarction with an accuracy of 97%.[13] In the field of chemical engineering, efficiently evaluating the reactivity (e.g., activity and stability) of catalyst surfaces has been a significant challenge in designing effective catalysts for industrial catalytic reactions. Recently, Liu et al.[14] tested several supervised learning algorithms to discover effective catalysts. They found that the so-called random forest models gave the best performance with reasonable accuracy, faster design speed, and lower cost.

The unsupervised learning method extracts hidden characteristics of the data without labeled data input[15] and is more likely to be used to analyze a database with presumed trends. One of its typical applications is clustering analysis, for example, discovering clusters of arbitrary shapes in point databanks, which is the cornerstone of artificial

TABLE I. Experimental conditions of particle–liquid flow.

Phase	Predicting parameter	ρ_L (kgm^{-3})	ρ_s (kgm^{-3})	d_p (mm)	$C_{s,m}$ (vol. %)
Single-phase	$\cdots$	1143	$\cdots$	$\cdots$	$\cdots$
Two-phase	$C_{s,m}$	1143	1165	2, 6, and 10	6–31
	d_p	1143	1165	2–10	12 and 31
	ρ_s	1145	1248	4	24

intelligence image processing and vision.[16] The semi-supervised learning approach is a hybrid of the supervised and unsupervised learning methods and can deal with both labeled and unlabeled data. It is an advantageous method since it can improve the efficiency of the learning system, e.g., requiring less labeled data and improving accuracy compared with using either of the two learning algorithms alone.[9] The reinforcement learning method is widely used to train machines or software agents under particular circumstances to improve their performance by estimating errors as rewards or penalties. It is applied to complex systems such as robotics[17] and autonomous driving tasks,[18] but is unsuitable for solving straightforward problems as it needs to interact with the environment.

In the field of fluid dynamics, different ML techniques have been exploited. For example, Li et al.[19] used the k-nearest neighbors (KNN) algorithm to learn the features of particle–liquid flows in a mixing tank from experimental Lagrangian trajectories. They achieved promising results, which demonstrated the ability of the KNN algorithm to deal with such complex dynamic systems. Bukka et al.[20] used the proper orthogonal decomposition—recurrent neural networks (POD-RNN) and convolution recurrent autoencoder network (CRAN) models to learn and predict the behavior of a fluid flowing past one cylinder and two side-by-side cylinders from Eulerian CFD simulation results. They extracted the dynamic features of the full-dimensional (2D) CFD data by projecting them onto a low-dimensional (1D) manifold via POD and convolutional neural networks models (part of CRAN), which reduced the calculation time and computation power. Furthermore, with an appropriately trained generative adversarial networks (GAN)

generator, Drygala et al.[21] managed to simulate the turbulent flow around a turbine stator. A small amount of data [e.g., a random noise vector from large eddy simulations (LES)] could generate results comparable to those of a full LES simulation.

Hashemizadeh et al.[22] applied multiple ML algorithms to predict the density of drilling mud by training the models with a data bank (817 datasets from wells in the South Pars gas field) to optimize the operating conditions of the drilling of the oil and gas wells. Ten parameters were analyzed by these ML algorithms, which indicated that the plastic viscosity and true vertical depth highly influenced the drilling mud density. As ML algorithms are very powerful in data and image classification tasks, Liu et al.[23] trained the k-nearest neighbors model with experimental void fraction data and managed to recognize and predict the flow pattern of gas–liquid flows in inclined tubes. Their results were comparable to previous experimental results reported by Dennis et al.[24] and Zhu et al.[25] Another common application of ML is to be coupled with numerical methods (e.g., CFD) to improve modeling performance. In a previous report,[26] a deep neural network was used to solve complex fluid dynamics equations, i.e., Navier–Stokes equations, of multiple bubble flow in a microchannel. Ouyang et al.[27] used an artificial neural network algorithm to predict mesoscale drag and solids stress in a gas-particle fluidized bed reactor. They concluded that their hybrid model was more powerful to handle flows with high superficial gas velocities compared with previous widespread models.

For dynamical systems such as turbulent particle-laden pipe flows, the complexity of the detailed internal information is the primary barrier to investigating them. For example, the local liquid velocity field varies as a result of the generation and dissipation of turbulent eddies, and particle velocities are also randomly affected by inter-particle and particle–wall collisions.[28] Moreover, the strong mutual influence between the solid and liquid phases (highly affected by particle volume concentration and particle-to-wall distance) results in turbulence modulation, which further complicates the dynamics of the flow.[28] Such phenomena are critical for understanding such complex systems and are captured in the Lagrangian flow trajectories of both the continuous and dispersed phases. Thus, various approaches have been reported to analyze such Lagrangian trajectories and extract useful flow information.[29–31] Additionally, to consider the effects of both
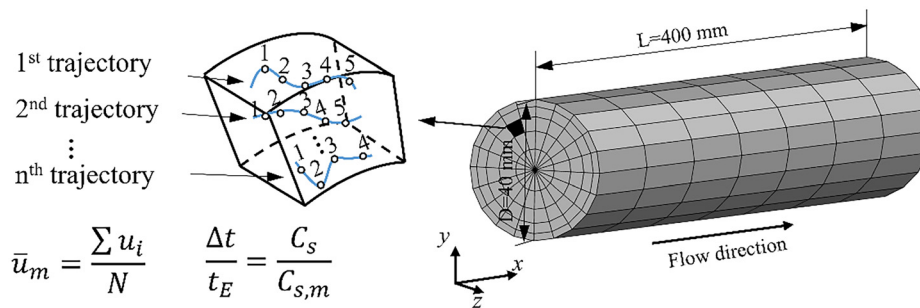
FIG. 1. Experimental data processing: estimation of local phase velocity and concentration.

$\bar{u}_m$ is the mean velocity of the mth cell; u_i is the ith local instantaneous velocity in this cell; N is the number of data points in this cell; Δt is the time that a particle spends in one cell; t_E is the ergodic time; C_s is the local particle concentration; $C_{s,m}$ is the mean particle concentration.
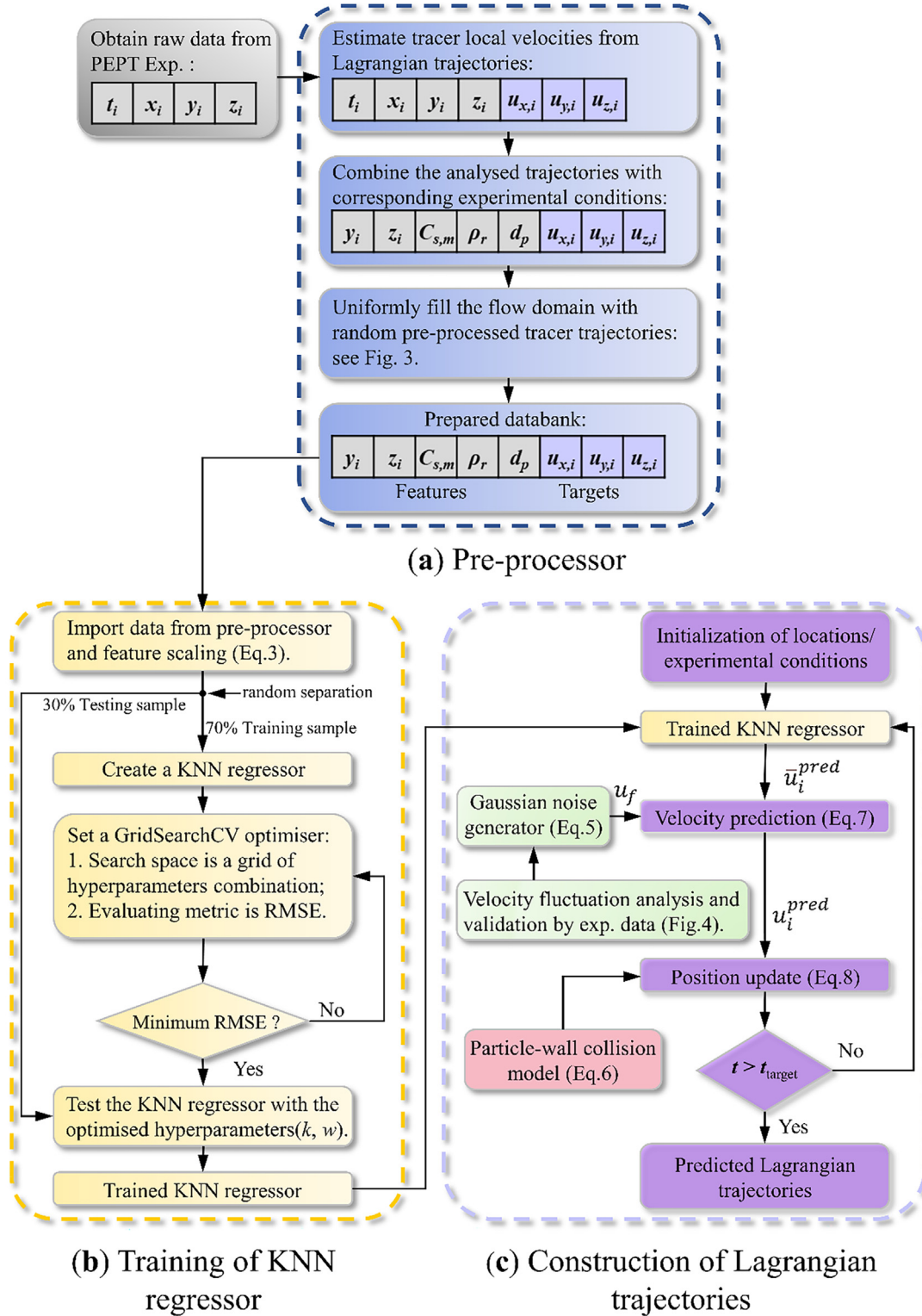
FIG. 2. Flowchart of hybrid machine learning framework for predicting Lagrangian flow trajectories in particle–liquid flow: (a) preprocessor; (b) training of KNN regressor; and (c) construction of Lagrangian trajectories.

particle and liquid velocity fluctuations while simplifying the problem, the combination of a dominant mean velocity with a stochastic fluctuation has been widely used to approximate the dynamics of particle–liquid flow systems.[32]

In this work, the hybrid ML algorithm approach is used to analyze turbulent particle transport in a horizontal pipe, due to its simplicity, straightforward implementation, and robustness for classification as well as regression.[23,33,34] High-quality training data are obtained from an experimental Lagrangian technique of 3D positron imaging particle tracking (PEPT), and the minimum amount required to accurately predict phase velocity and concentration distributions in the pipe by the hybrid ML algorithm is determined. In addition, by combining datasets corresponding to different flow conditions of particle concentration, size, and density, the hybrid ML model is trained to predict the multiphase flow behavior under new conditions within and without the range of experimental data available. Moreover, the importance of various flow parameters on the local velocity field of the particulate phase is also evaluated using a feature importance analysis technique to further understand the dynamics of the multiphase flow system.

II. EXPERIMENTAL

A. Experimental data bank acquisition

The experimental training data were obtained in a horizontal pipe conveying a turbulent particle–liquid suspension, as illustrated in Fig. S1 in the supplementary material. The particle–liquid mixture was continuously well mixed in a conical tank and pumped into a 4 m Perspex pipe with an internal diameter (D) of 40 mm by a vortex pump. Visualization of the two-phase flow by PEPT was conducted over a 400 mm long pipe section, 3 m downstream of the entrance bend, where flow was fully developed. This work focuses on studying the influence of particulate phase properties on the behavior of the two-phase flow. Thus, the liquid phase properties [e.g., liquid viscosity (μ_L), density (ρ_L), and temperature] and mixture flow rate (u_m) were kept unchanged, and the liquid Reynolds number [$Re_L = \frac{\rho_L u_m D}{\mu_L}$] was ~8000. The dynamic data bank consisted of particle mean concentration ($C_{s,m}$) ranging from 6 to 31 vol. %, particle size (d_p) ranging from 2 to 10 mm, and particle-to-liquid density ratio (ρ_r) ranging from 1.02 to 1.09. The particles used were nearly neutrally buoyant calcium alginate beads ($\rho_r = 1.02$). Particle density was increased by dosing silica powder into the alginate solution used to make the beads. Full experimental details and protocols can be found in our previous papers.[35–38] The experimental particle–liquid flow conditions under which the data bank utilized for training the ML algorithm as well as assessing its performance was generated are listed in Table I.

B. PEPT flow imaging and data processing

PEPT is a non-intrusive measurement technique that uses positron-emitting particle tracers to accurately track the motion of the fluid phase or the particle phase and record its 3D locations as a function of time. Thus, long-term Lagrangian trajectories of a complex dynamic system are obtained. The technique has the unique advantage of visualizing opaque flows (e.g., turbid fluids or particle–liquid flows with high solid loadings) while giving comparable accuracy with leading optical techniques such as particle image velocimetry (PIV)[39] and laser Doppler velocimetry (LDV).[40] Thus, it has been widely used to

study various particle–liquid flow systems, and details can be found in previous papers.[41,42]

To obtain sufficient dynamic information on the particle–liquid flow system while reducing experimental time, several radioactive tracers were used simultaneously, and at least 500 trajectories were determined in each experiment. To gain high-quality dynamic information on solid and liquid phases, tracers used in experiments were matched with their corresponding phases as closely as possible. Small neutrally buoyant resin particle tracers (400 μm) were used to track the liquid phase. Representative particles taken from the population of particles used in the experiments were radiolabelled and used as tracers to track the solid phase. Using several tracers at a time to track a particular phase produces proportionally more data, thus reducing experimental time considerably.

The tracer instantaneous velocity is first estimated by a regression analysis using a number of locations and time obtained from the raw PEPT trajectory data.[35] As shown in Fig. 1, the mean velocity inside a given cell ($\bar{u}_m$) is calculated by averaging all instantaneous velocities (u_i) in the cell. Given the previously proven correlation between tracer cell occupancy and local phase concentration, the local volume fraction of particles (C_s) in a cell is estimated from the particle Lagrangian trajectories, using the total time that a particle spends in a cell (Δt) and the ergodic time of this cell (t_E); thus,

$$\frac{\Delta t}{t_E} = \frac{C_s}{C_{s,m}}. \tag{1}$$

The theory of this occupancy method is detailed in our previous work.[43,44]

III. MACHINE LEARNING MODEL THEORY

The flow chart depicted in Fig. 2 outlines the hybrid ML framework used in this work, which consists of a preprocessor (blue part), a general KNN regression learning model (yellow part), a validated Gaussian noise model (green part), and a particle–wall collision model
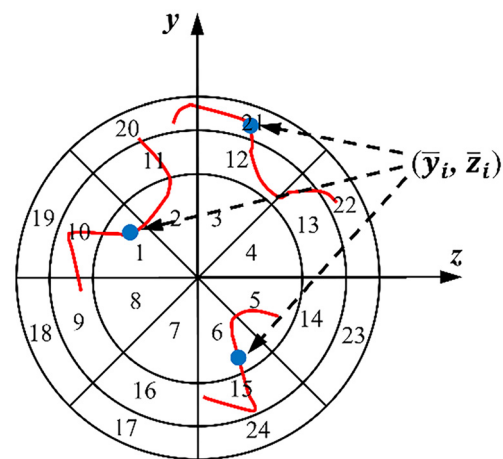
$\bar{y}_i$ and $\bar{z}_i$ are respectively the mean vertical and horizontal position of the ith trajectory

FIG. 3. Preprocessor grid for random selection of PEPT trajectories allocated to data bank.

(pink part). The theory underpinning these models is described in Secs. III A–III E.

A. Preprocessor

Since the performance of the hybrid ML framework relies highly on the quality of the input data, a preprocessor [Fig. 2(a)] was designed to obtain representative high-quality data of the dynamic two-phase flow system studied. First, accurate Lagrangian phase trajectories were obtained from the raw experimental PEPT data (t, x, y, z) using a regression analysis.[35] Combining these Lagrangian PEPT trajectory data (x, y, z, u_x, u_y, u_z) with the associated experimental conditions ($C_{s,m}$, ρ_r, d_p), a new data array was generated. Since the flow in the pipe section visualized by PEPT was steady and fully developed, the

axial x position was irrelevant. Thus, the preprocessor provided the database with y, z, $C_{s,m}$, ρ_r, and d_p as input features and u_x, u_y, and u_z as predicted targets. As discussed further below, one of the aims was to determine the minimum amount of experimental data needed in the preprocessor data bank to efficiently train the ML algorithm. Thus, different size datasets were tested, and the algorithm predictions evaluated against the full set of experimental data. To construct a data bank consisting of input data, which are uniformly distributed and could represent the full flow domain, the cross section of the pipe was divided into 24 cells of equal volume (Fig. 3). Starting from the first central cell and working radially outward toward the wall, the process consists of identifying the first random trajectory whose mean position ($\bar{y}_i$, $\bar{z}_i$) is situated within the cell. Such a trajectory is then selected and put into the data bank, before moving to the next cell. When all 24 cells
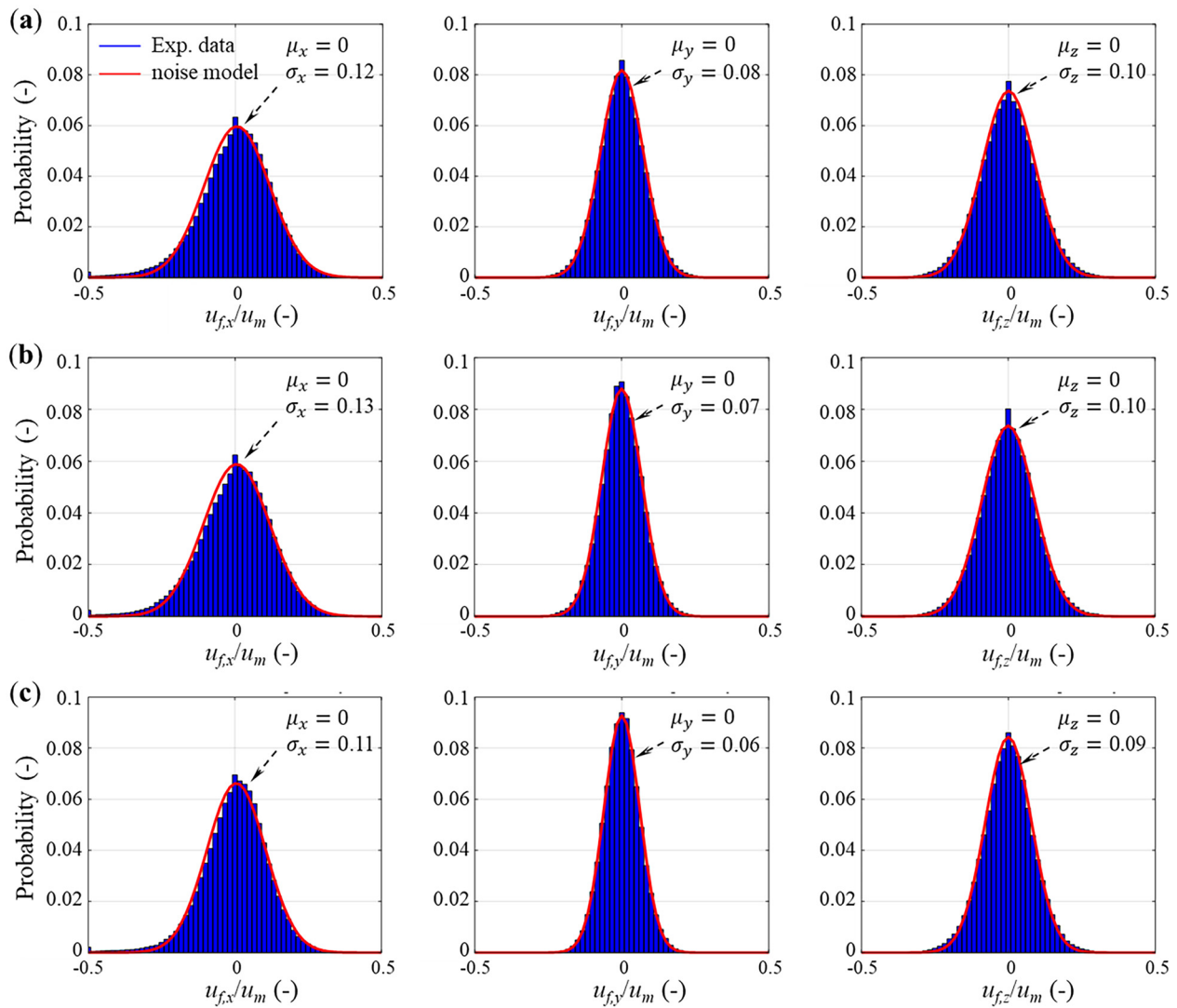
FIG. 4. Validation of Gaussian noise model against PEPT measurements in x, y, and z directions: (a) single-phase flow ($Re_L \approx 8000$); (b) liquid phase; and (c) particulate phase of particle–liquid flow ($Re_L \approx 8000$, $C_{s,m} = 12$ vol. %, $d_p = 2$ mm, and $\rho_r = 1.02$).
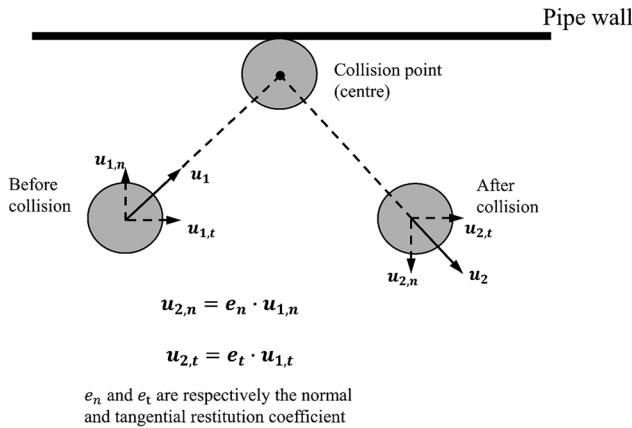
FIG. 5. Schematic illustration of tracer-wall collision model.

have been allocated their first trajectory, the preprocessor goes back to the first cell, and so on. The random selection stops when the target number of selected trajectories across the pipe is reached.

B. k-nearest neighbors model

The k-nearest neighbors (KNN) model has a faster modeling speed, and the prediction results are robust. KNN tends to have a better root mean square error (RMSE) compared to support vector regression (SVR), linear regression, and artificial neural network (ANN).[45] The basic mechanism of the KNN model is straightforward. Instead of establishing an internal model from the input training data, it stores the training data in n-dimensional space and predicts an unknown data point by evaluating its similarity to other nearest data points (k) in the training data using a weight function (w).[46] As shown in Eq. (2), the predicted mean velocity ($\bar{u}_i^{\text{pred}}$) is computed by the weighted average of the local instantaneous velocities at k nearest neighboring training data points,

$$\bar{u}_i^{\text{pred}} = \sum_{j=1}^{k} w_{ij} \times u_j, \qquad (2)$$

where u_j is the jth instantaneous velocity of the training data, which in this case comes from raw PEPT data; k is the number of the nearest neighbors used to predict $\bar{u}_i^{\text{pred}}$; and w_{ij} is the weight function, such as the distance weight function, which evaluates the importance of a training data point (u_j) by the inverse of the distance to the predicted data point ($\bar{u}_i^{\text{pred}}$).

Due to the existence of dimension and unit differences, all variables in the feature space are normalized to avoid the multiple features spanning effect, as follows:

$$X^* = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \qquad (3)$$

where X^*, X, $X_{\min}$, and $X_{\max}$ are the normalized, original, minimum, and maximum values of the feature considered. Thus, all the variables are scaled between 0 and 1. The distance function used to evaluate the similarity of the data in this work is the Euclidean distance given by

$$L_2\left(X_i^*, X_j^*\right) = \left(\sum_{l=1}^{n} \left|X_i^{*(l)} - X_j^{*(l)}\right|^2\right)^{\frac{1}{2}}, \qquad (4)$$

where l is the number of feature dimensions, which is equal to 5 (y, z, $C_{s,m}$, ρ_r, d_p). Due to the simplicity of the KNN model, k and w are the only critical parameters that need to be optimized. The optimization of these parameters and the validation of the model are discussed further below. As shown in Fig. 2(b), 70% of data randomly selected from the preprocessor data bank were used for training the KNN regressor and the remaining data (30%) were used for testing it.

C. Experimental validation of Gaussian noise model

To predict instantaneous velocities by the ML algorithm, the Gaussian noise model was used to predict the local fluctuation velocity (u_f); thus,
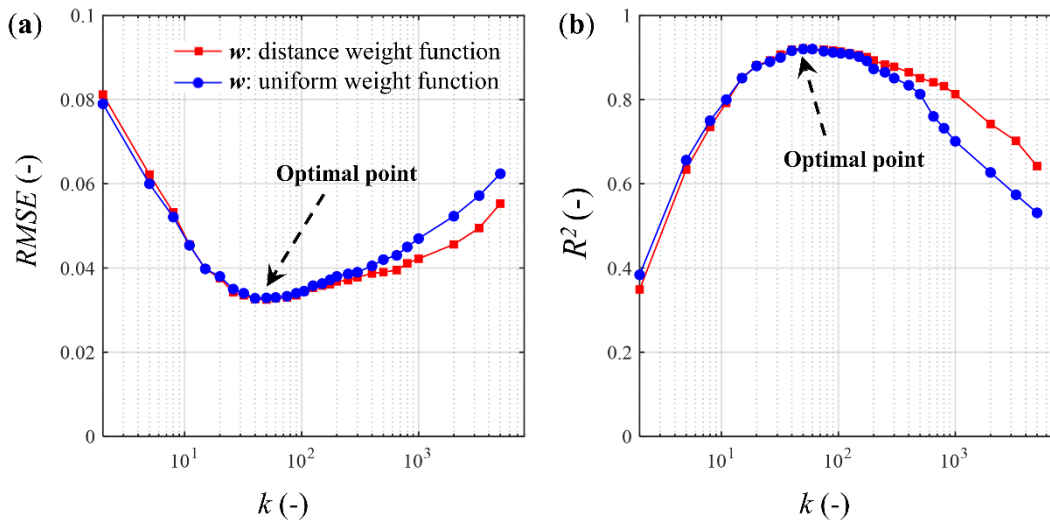

FIG. 6. Optimization of KNN algorithm parameters (k and w) based on (a) RMSE and (b) R^2.

$$u_f \sim \mathcal{N}(\mu_i, \sigma_i^2), \quad (5)$$

where μ_i and σ_i are the mean and standard deviation of the local fluctuation velocity. The model was successfully validated in 3D by comparing with real experimental PEPT fluctuation velocities, as shown in Fig. 4, for single-phase flow and particle–liquid flow. Thus, the Gaussian noise model was used to generate velocity fluctuations for the ML algorithm.

D. Particle-wall collision model

A particle-wall collision model, illustrated in Fig. 5, was added to the hybrid learning system to avoid physically unrealistic predictions of the particle tracer leaving the pipe flow domain. The model uses a restitution coefficient to readjust the velocity magnitude and direction of the tracer once it crosses the pipe wall; thus,

$$\begin{aligned} u_{2,n} &= e_n \times u_{1,n}, \\ u_{2,t} &= e_t \times u_{1,t}, \end{aligned} \quad (6)$$

where $u_{2,n}$ and $u_{1,n}$ are the tracer velocities in the normal direction before and after collision, while $u_{2,t}$ and $u_{1,t}$ are the tracer velocities in the tangential direction before and after collision. A sensitivity analysis showed that the ML predictions were not sensitive to the value of the restitution coefficient within the range [0, 1], so a value of 1 (i.e., perfectly elastic collisions) was adopted; details of the analysis can be found in our previous work.[19,47]

E. Generation of long-term Lagrangian trajectories

To obtain the Lagrangian trajectory of a given phase (liquid or particle) over a specific time length, the first step is to generate the instantaneous velocity of the initial zero position. By combining Eqs. (2) and (5), the instantaneous velocity is expressed as follows:

$$u_i^{\text{pred}} = \bar{u}_i^{\text{pred}} + u_f = \sum_{j=1}^{k} w_{ij} \times u_j + \mathcal{N}(\mu_i, \sigma_i^2). \quad (7)$$

Once the current value in the computation process of the 3D instantaneous velocity (u_i^{pred}) is predicted from Eq. (7), the next location of the tracer is estimated by calculating its displacement during a given time step (dt); thus,

$$P_i^{t+dt} = P_i^t + u_i^{t,\text{pred}} \times dt, \quad (8)$$

where P_i^t, $u_i^{t,\text{pred}}$, and P_i^{t+dt} are, respectively, the current location, current instantaneous velocity, and the next tracer location. Thus, a full Lagrangian trajectory is generated, as depicted by the flow chart in Fig. 2(c).

IV. RESULTS AND DISCUSSION

The results of this study are divided into four parts: (i) the optimization of the k and w values used in the KNN learning model; (ii) using the optimized KNN regressor, the tests conducted to determine the minimum size of training dataset of Lagrangian phase trajectories needed to reliably predict the full flow field for single-phase flow and for particle–liquid flow including spatial phase distribution, are presented; (iii) the ability of this hybrid learning algorithm to predict flow characteristics under new conditions within and without the range of experiments pertaining to the training PEPT data bank; and (iv) a feature importance analysis of the particle velocity based on the entire PEPT data bank is discussed.

A. Parameter tuning and model validation

To make the KNN learning model achieve the best performance in predicting the flows considered, the number of nearest datapoints (k) and weighting functions (w) were optimized by evaluating the root mean square error (RMSE) and coefficient of determination (R^2) of the prediction results, which are defined as follows:
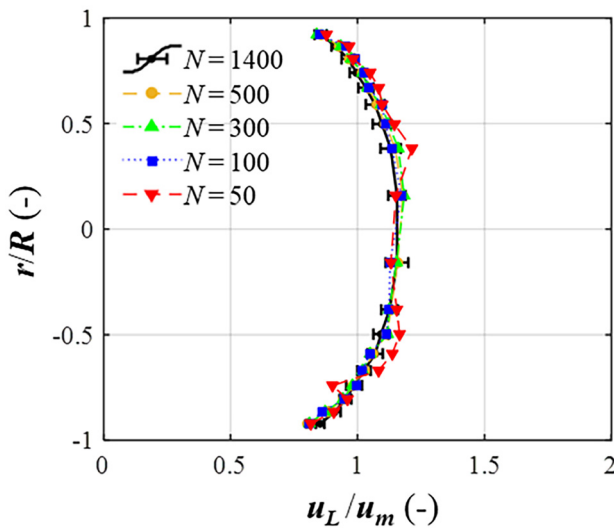
FIG. 7. Effects of the number of liquid PEPT-determined trajectories N used to obtain the experimentally measured radial liquid velocity distribution in single-phase flow ($Re_L \approx 8000$).
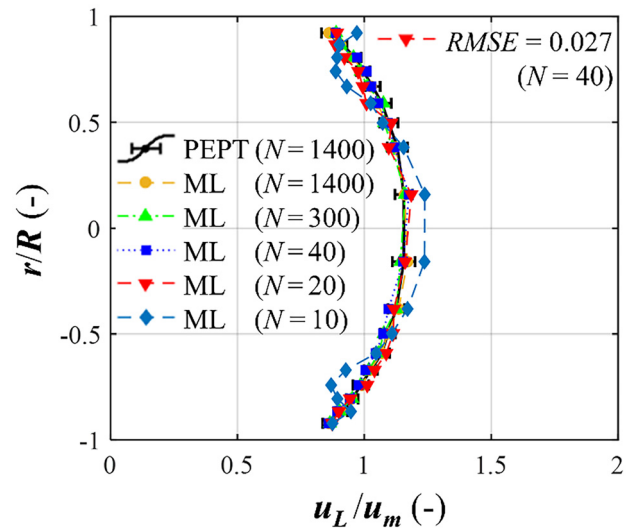
FIG. 8. Effects of the number of liquid PEPT-determined trajectories N used to train the ML algorithm on the predicted radial liquid velocity distribution in single-phase flow ($Re_L \approx 8000$).

$$RMSE = \sqrt{\frac{\sum_{n_p=1}^{N_p}\left|u_{n_p} - u_{n_p}^{pred}\right|^2}{N_p}},$$

$$R^2 = 1 - \frac{\sum_{n_p=1}^{N_p}\left(u_{n_p} - u_{n_p}^{pred}\right)^2}{\sum_{n_p=1}^{N}\left(u_{n_p} - \bar{u}_{n_p}\right)^2}, \tag{9}$$

where N_p is the number of training data samples. u_n, u_n^{pred}, and $\bar{u}_n$ are, respectively, the real experimentally measured local velocity of the n_pth sample in the database, the corresponding predicted velocity given by the KNN regressor and the statistical mean velocity at the location considered (i.e., the particular grid cell, Fig. 1). Ideally, the RMSE value of the ML predictions should tend to zero, while the R^2 value should tend to one. Different sizes of a dynamic dataset require different k values; thus, the optimal hyperparameters need to be determined for each database input.[19,46] The variations of RMSE and R^2 are plotted in

Fig. 6 as a function of the number of neighbors k and the type of weighting function w, namely, distance and uniform weight functions. The two weight functions perform equally well at lower k values, but the distance weight function outperforms the uniform weight function at higher k values. Thus, the distance weight function was adopted. In this case, the optimal value for k extracted from Fig. 6 was ~50.

B. Minimum training data requirement

For complex dynamic systems like turbulent fluid systems, a frequently asked question concerns the minimum amount of training data required to faithfully represent the majority characteristics of such systems. Specifically, for PEPT experiments, if the minimum number of tracer passes in the pipe capable of embodying the studied dynamic system can be determined, the experimental time can be optimized.

1. Single-phase flow

The radial distribution of liquid velocity in single-phase turbulent pipe flow is plotted in Fig. 7 as a function of the number N
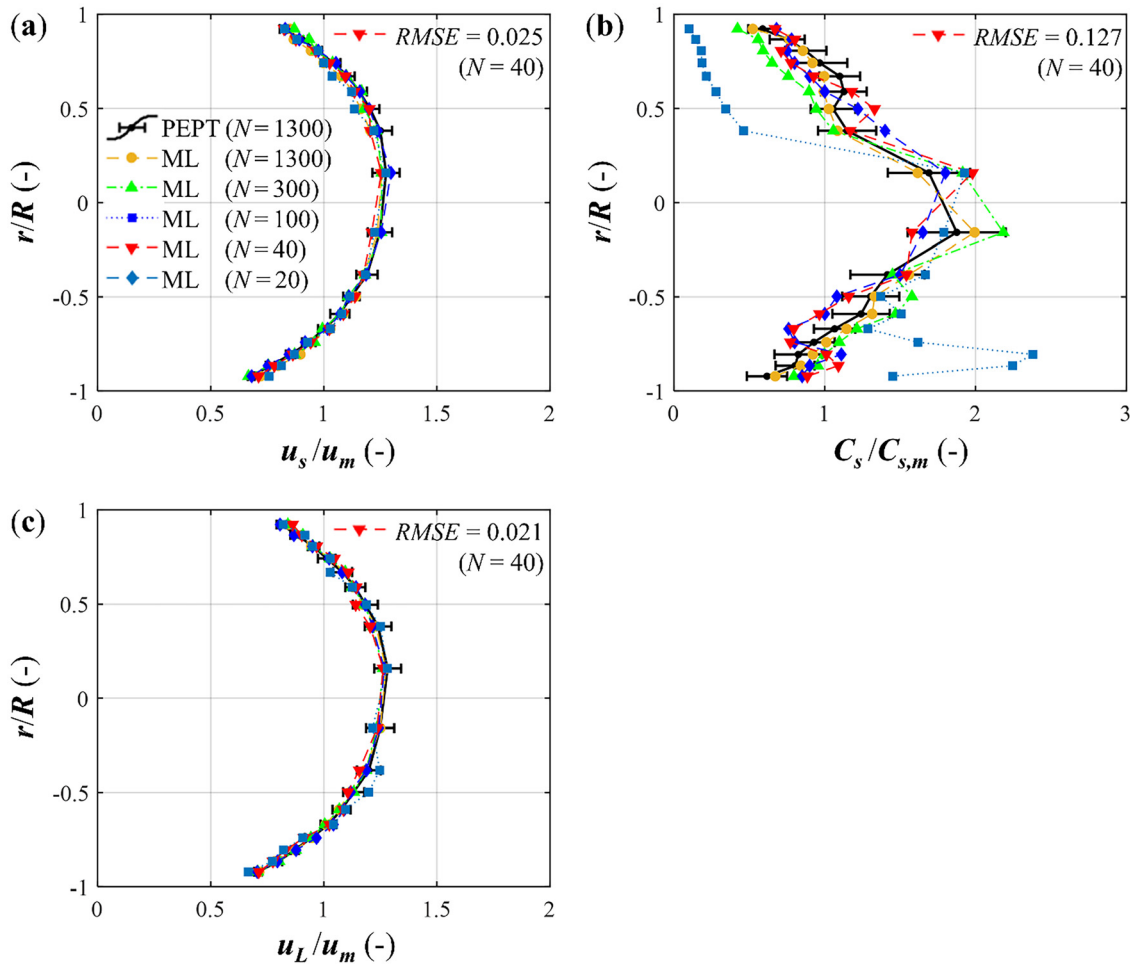
FIG. 9. Effects of number of PEPT-determined trajectories N used to train the ML algorithm on the predicted radial distribution of (a) particle velocity; (b) particle concentration; and (c) liquid velocity; $d_p = 2$ mm; $\rho_r = 1.02$; and $C_{s,m} = 21$ vol. %.

08 November 2023 14:09:21
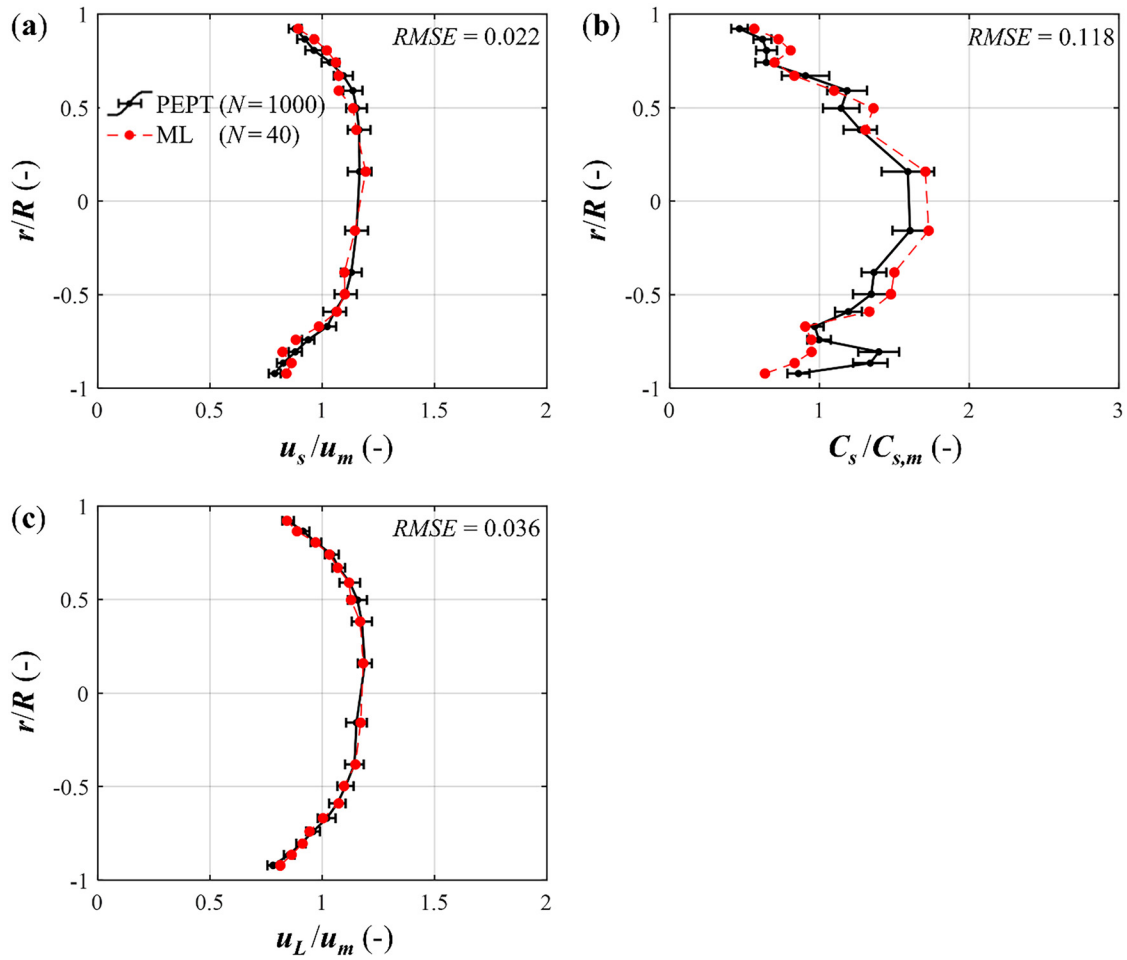
FIG. 10. ML algorithm predictions under conditions of larger particle size and concentration: (a) particle velocity distribution; (b) particle concentration distribution; and (c) liquid velocity distribution; $d_p = 6$ mm; $\rho_r = 1.02$; and $C_{s,m} = 31$ vol. %.
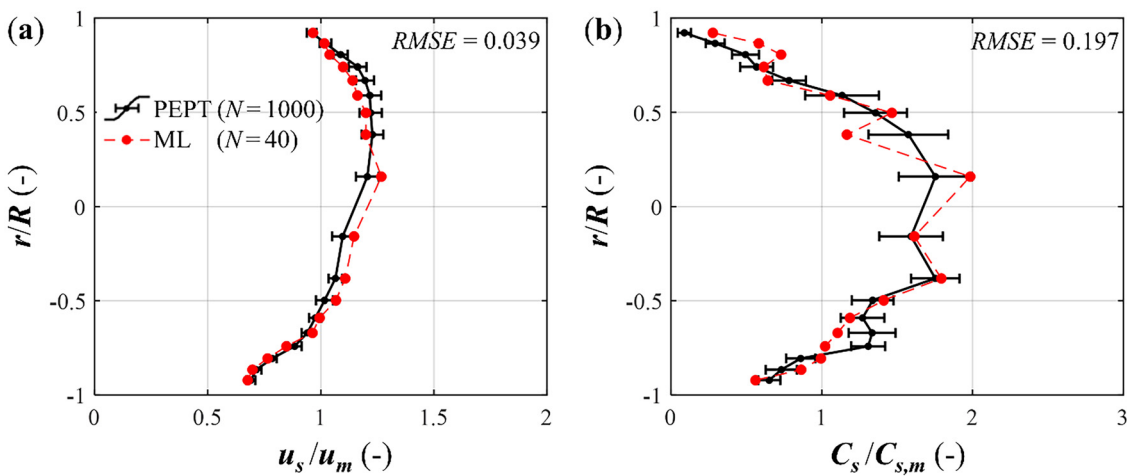
FIG. 11. ML algorithm predictions under conditions of denser particles: (a) particle velocity distribution; (b) particle concentration distribution; $d_p = 4$ mm; $\rho_r = 1.09$; and $C_{s,m} = 24$ vol. %.

08 November 2023 14:09:21

of experimentally measured liquid PEPT trajectories used. The results show that at least 100 experimentally measured trajectories are required to accurately represent the radial velocity profile. Using these data and by considering 40 annular rings, the local mass flow rate was calculated in each ring and summed up across the pipe section to check for mass continuity at eight axial positions along the pipe. The total mass flow rate, thus, obtained was within 3% of the value obtained by measuring the flow rate delivered at the end of the pipe using a stop watch and bucket method, which confirms the accuracy of the PEPT measurements.

In Fig. 8, a similar test is depicted where different numbers of experimental trajectories are used as data bank to train the KNN regressor and generate new Lagrangian trajectories from which the radial velocity profile is derived. The results show that feeding the ML hybrid algorithm with 20 experimental trajectories is sufficient to predict an accurate radial velocity profile. Larger input datasets do not significantly improve predictions, whereas less data lead to a significant deterioration in results.

2. Particle–liquid flow

A similar investigation was conducted to establish the minimum number of PEPT trajectories needed for the solid and liquid phases to give accurate radial distributions of particle/liquid velocity and particle concentration. The results plotted in Fig. S2 in the supplementary material show that at least 100 experimental trajectories are needed for accurate determination of the particle/liquid velocity profiles (identical to liquid in single-phase flow, as discussed above), but 500 experimental trajectories are needed for the concentration profile. Thus, to obtain a reliable representation of particle concentration distribution, a much greater number of experimental PEPT trajectories are required than for velocity. It should be noted that all these velocity profiles and particle concentration profiles verified the mass continuity in the pipe to within 3%.

To investigate the prediction capability of the hybrid learning algorithm, the effects of using different sample sizes of PEPT trajectories as training databanks to predict the particle and liquid velocity

FIG. 12. ML algorithm predictions under conditions within the range of experimental measurements: (a) particle velocity distribution; (b) particle concentration distribution; and (c) liquid velocity distribution; $d_p = 6$ mm; $\rho_r = 1.02$; and $C_{s,m} = 21$ vol. %.
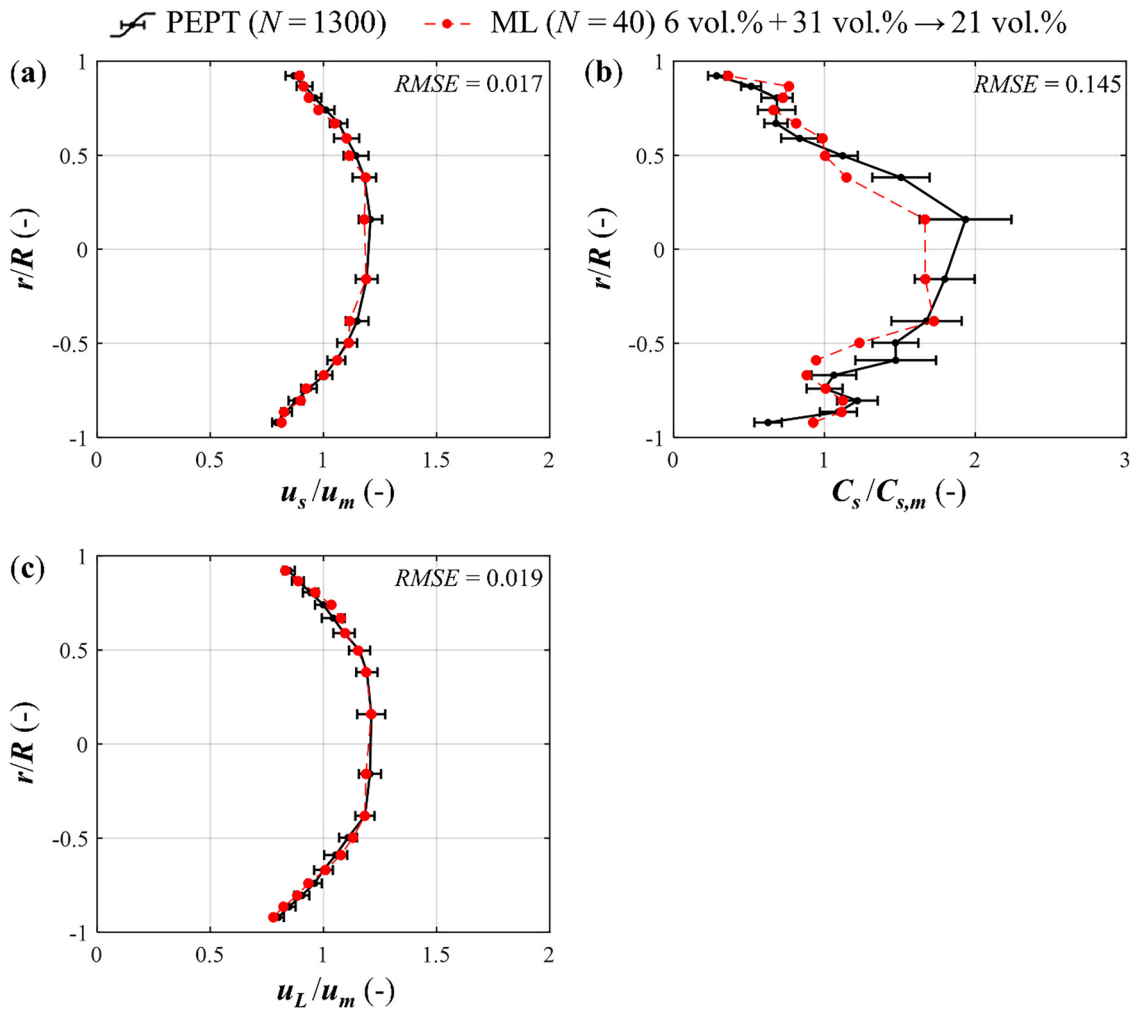
profiles as well as the particle volume concentration distribution are depicted in Fig. 9. In this case, the particle–liquid flow consists of 36 wt. % sugar solution conveying 2 mm alginate beads ($\rho_r = 1.02$, $C_{s,m} = 21$ vol. %). With 20 trajectories or more, the particle velocity profile is well predicted [Fig. 9(a)]. The predicted particle concentration distribution also agrees reasonably well with the full set of experimental PEPT data with only slight deviations near the center and bottom region of the pipe, when 40 trajectories or more are used [Fig. 9(b)].

As depicted in Fig. 9(c), the minimum number of trajectories required to generate an accurate liquid-phase velocity field is 20, similar to the solid-phase velocity field [Fig. 9(a)] and to the velocity field in single-phase flow (Fig. 8). Therefore, for single-phase or particle–liquid dynamics systems, the minimum number of phase trajectories for the hybrid learning model to learn and accurately predict a given phase velocity field is the same. However, a somewhat larger training dataset is required to obtain a reliable particle concentration distribution. This may be attributed to the complex behavior of particles, including

settling, margination, particle–particle, and particle–wall interactions, which affect their spatial distribution. Nevertheless, the results demonstrate the predictive power of the ML algorithm, which only requires a relatively small input dataset compared to what is experimentally required to predict either the phase velocity distribution (20 cf. 100) or phase concentration distribution (40 cf. 500 trajectories). Nevertheless, in the rest of the study, we opted to use 40 trajectories to predict local particle and liquid velocities as well as local particle concentration.

To further investigate the prediction ability of the hybrid learning algorithm to handle more challenging flow conditions, the algorithm was trained using dynamic data produced by experiments with higher solid loadings and larger particle sizes. The ML predictions for a two-phase flow with $d_p = 6$ mm, $C_{s,m} = 31$ vol. %, and $\rho_r = 1.02$ are presented in Fig. 10, based on a training data bank of 40 trajectories, as pointed out above. The agreement with the full set of PEPT data is excellent for both particle and liquid velocity distributions. The trend of particle concentration profile prediction also matches the

FIG. 13. ML algorithm predictions under conditions outside the range of experimental measurements: (a) particle velocity distribution; (b) particle concentration distribution; and (c) liquid velocity distribution; $d_p = 6$ mm; $\rho_r = 1.02$; and $C_{s,m} = 6$ vol. %.

experimental results but with some slight discrepancies near the central and bottom regions of the pipe. Another case with an even larger particle size (10 mm) is presented in the supplementary material (Fig. S3).

Another case using denser particles ($\rho_r = 1.09$, $C_{s,m} = 24$ vol. %, and $d_p = 4$ mm) is depicted in Fig. 11, showing very good ML predictions of particle velocity and concentration fields. Thus, trained by a small amount of experimental data, the hybrid learning algorithm can learn and generate numerical dynamic data (Lagrangian trajectories) for single-phase or two-phase particle–liquid flows of various particle sizes, concentrations, and densities, which are close to detailed experimental measurements.

C. Model investigation of predicting new particle–liquid flows

Since the ML technique seems to be able to extract inherent hidden information of a complex dynamic system, in the Secs. IV C 1

and IV C 2, the ability of the predictive hybrid algorithm will be tested for flow conditions, which are not covered by the information contained in the training data bank, i.e., new conditions not specifically measured but which are within the limits of the range of experimental measurements, or conditions which are outside the range of experiments, thus extending its application range to interpolate or extrapolate flow conditions.

1. Predicting flows with different particle concentration

As shown in Fig. 12, given a training dataset corresponding to nearly neutrally buoyant particles with $d_p = 6$ mm and $C_{s,m} = 6$ and 31 vol. %, the two-phase velocity field and the solid distribution corresponding to $d_p = 6$ mm and $C_{s,m} = 21$ vol. % are well predicted by the hybrid learning algorithm. There are some discrepancies in the particle concentration profile, but the predictions follow the trend of the experimental profile closely with most of the predictions being within the
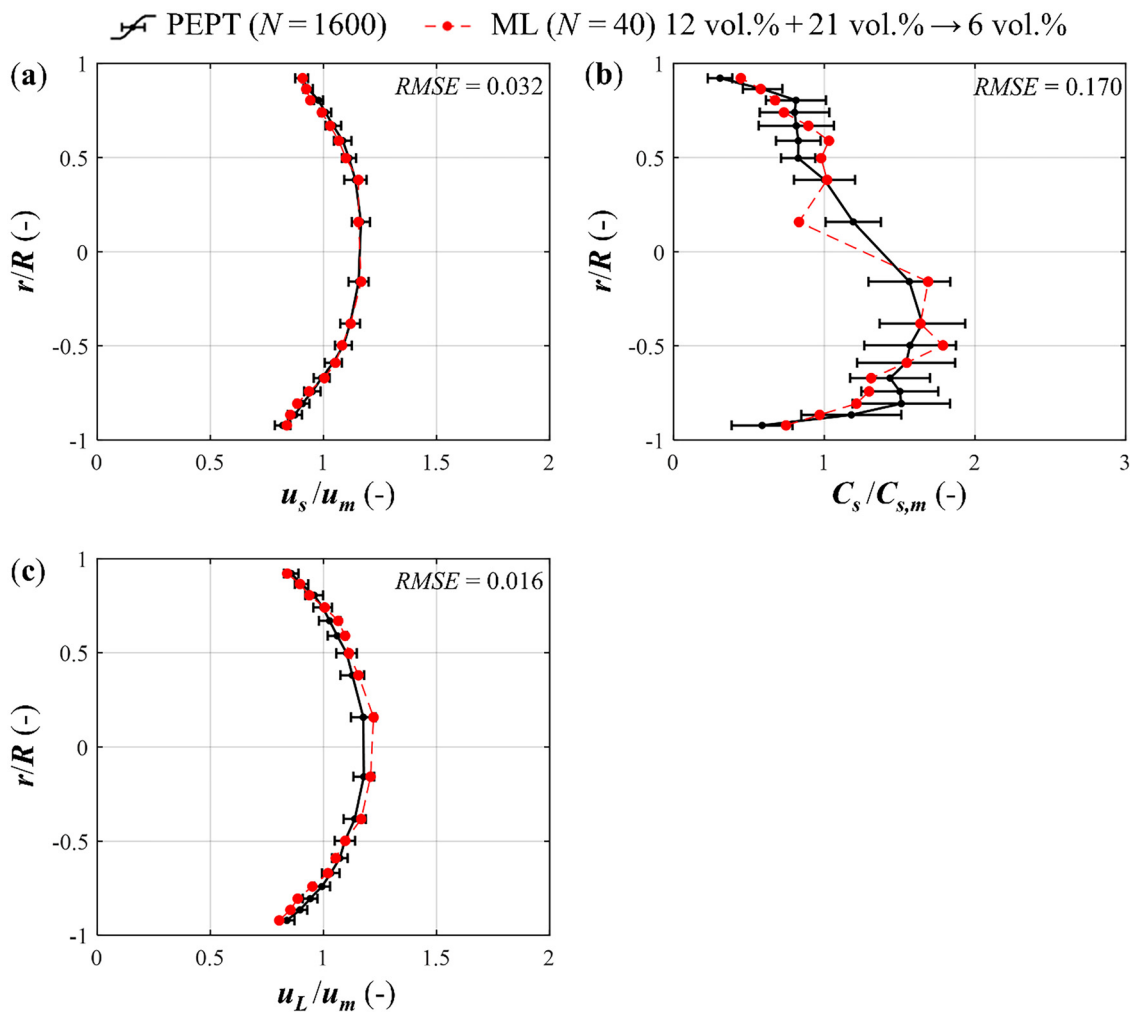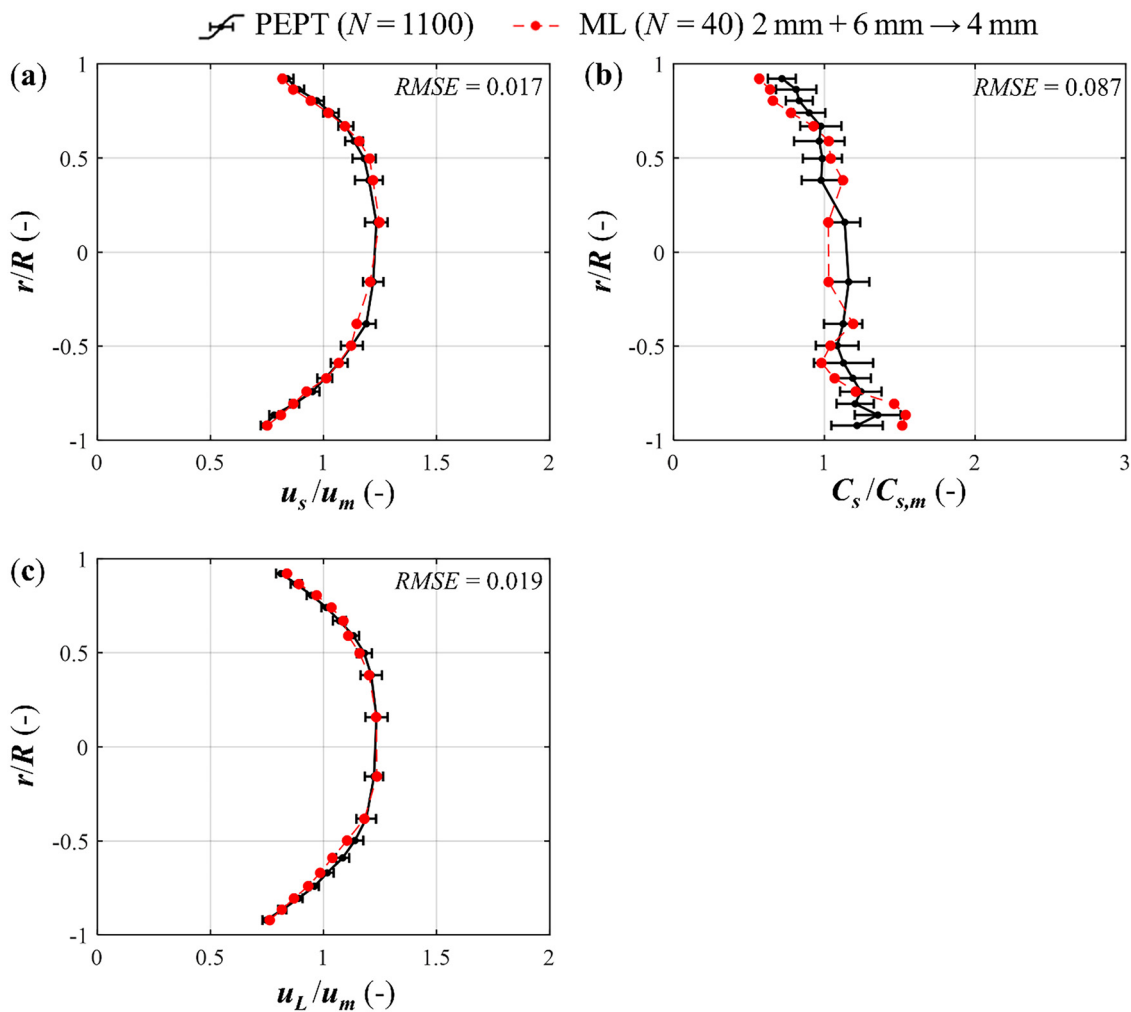

FIG. 14. ML algorithm predictions under conditions within the range of experimental measurements: (a) particle velocity distribution; (b) particle concentration distribution; and (c) liquid velocity distribution; $d_p = 4$ mm; $\rho_r = 1.02$; and $C_{s,m} = 31$ vol. %.
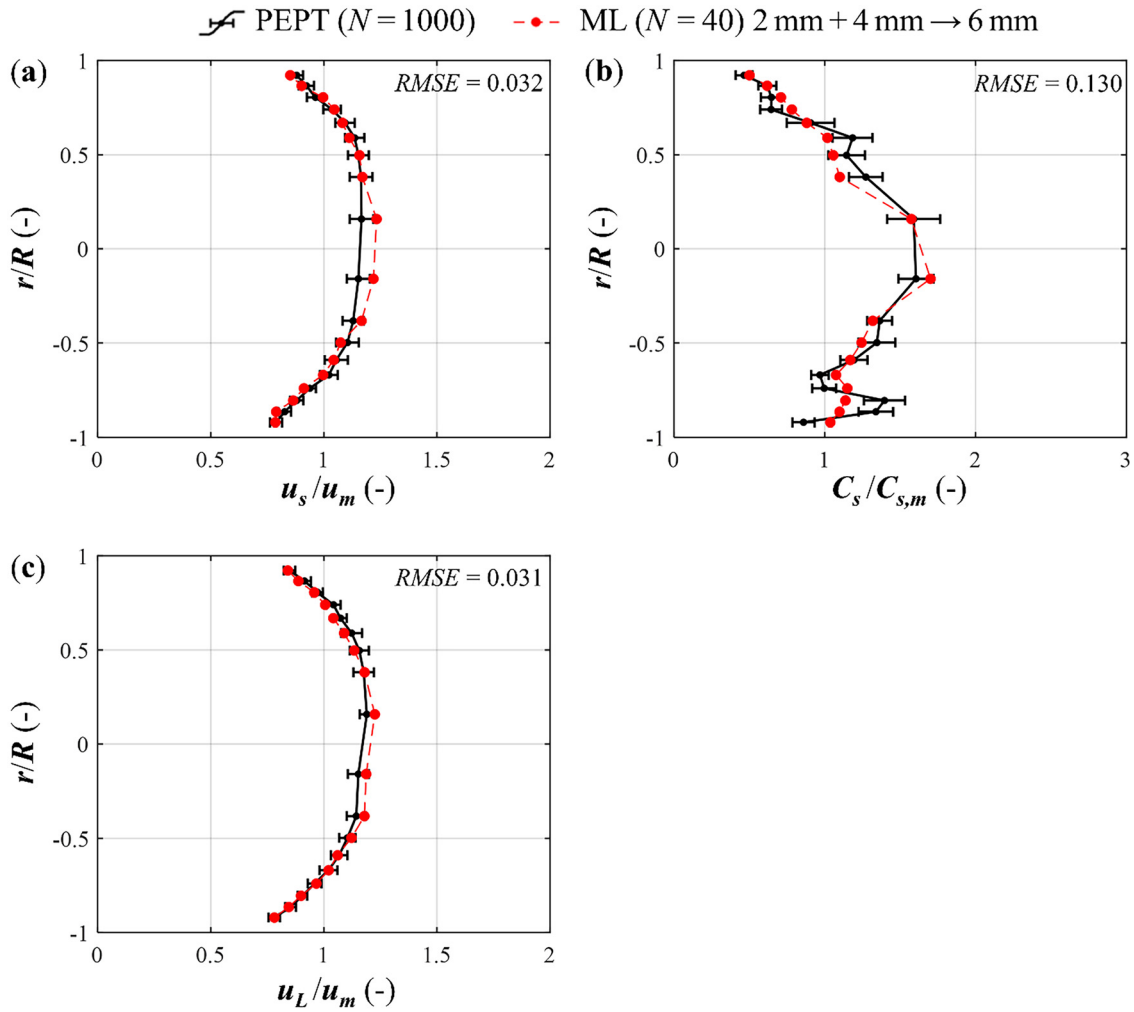
FIG. 15. ML algorithm predictions under conditions outside the range of experimental measurements: (a) particle velocity distribution; (b) particle concentration distribution; and (c) liquid velocity distribution; $d_p = 6$ mm; $\rho_r = 1.02$; and $C_{s,m} = 31$ vol. %.

error bars. Another example of a successfully predicted flow case with a different particle size of $d_p = 2$ mm and $C_{s,m} = 21$ vol. %, based on input data from the cases corresponding to $d_p = 2$ mm, $C_{s,m} = 6$, and 31 vol. %, is presented in the supplementary material (Fig. S4).

The case presented in Fig. 13 attempts to predict flow conditions of particle concentration, which are outside the range of experimental measurements. In other words, training data corresponding to $d_p = 6$ mm and $C_{s,m} = 12$ and 21 vol. % were fed to the ML algorithm to predict the behavior of a flow with $d_p = 6$ mm and $C_{s,m} = 6$ vol. %. The two-phase velocity profiles are very well predicted. The concentration distribution is also well predicted with only small discrepancies. A different case corresponding to $d_p = 2$ mm and $C_{s,m} = 6$ vol. %, based on input data from the cases corresponding to $d_p = 2$ mm and $C_{s,m} = 12$ and 21 vol. %, is included in the supplementary material (Fig. S5), showing very good predictions by the ML algorithm.
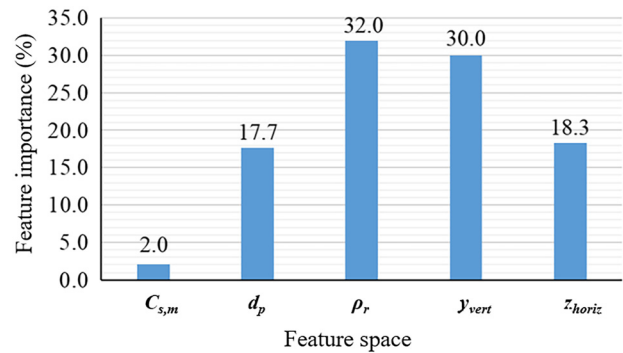

FIG. 16. Feature importance analysis of particle velocity using experimental PEPT data.

2. Predicting flows with different particle size

As shown in Fig. 14, providing the hybrid learning algorithm with data corresponding to flow with $C_{s,m} = 31$ vol. % and $d_p = 2$ and 6 mm, it manages to properly predict both liquid and particle velocity profiles as well as the particle concentration distribution for flow with $C_{s,m} = 31$ vol. % and $d_p = 4$ mm. Predictions for a different flow case with $C_{s,m} = 12$ vol. % and $d_p = 6$ mm is presented in the supplementary material (Fig. S6) based on training data pertaining to flows with $C_{s,m} = 12$ vol. % and $d_p = 2$ and 10 mm, which is further evidence of the predictive power of the ML algorithm.

The case presented in Fig. 15 shows predictions for flow conditions of particle size, which are outside the range of experimental measurements. In this case, training data corresponding to flows with $C_{s,m} = 31$ vol. % and $d_p = 2$ and 4 mm are fed to the ML algorithm to predict results for flow conditions of $C_{s,m} = 31$ vol. % and $d_p = 6$ mm. Another case using training data corresponding to $C_{s,m} = 12$ vol. % and $d_p = 6$ and 8 mm, to predict results for flow conditions of $C_{s,m} = 12$ vol. % and $d_p = 4$ mm, is reported in the supplementary material (Fig. S7). In both cases, the ML performance is very good.

Thus, the hybrid learning algorithm is able to reliably predict phase velocity fields and spatial phase distribution within and without the range of experimental data used for training the algorithm. Such a capability has important implications for facilitating and improving the design and operation of processes involving particulate flows.

D. Feature importance analysis

Permutation feature importance is a technique to estimate how useful the features are at predicting targets or the degree of dependence of the dataset and model on each input feature.[48] The feature importance score is defined as the drop in model performance when a single feature is completely shuffled, because reducing the input of a single feature can weaken the model relationship between feature space and prediction. This technique is model independent and can be calculated multiple times using different feature permutations.

In this study, five features are considered in the training database, which are particle loading ($C_{s,m}$), particle size (d_p), particle-to-liquid density ratio (ρ_r), particle vertical position (y), and horizontal position (z) in the pipe (Fig. 1). The permutation feature importance analysis
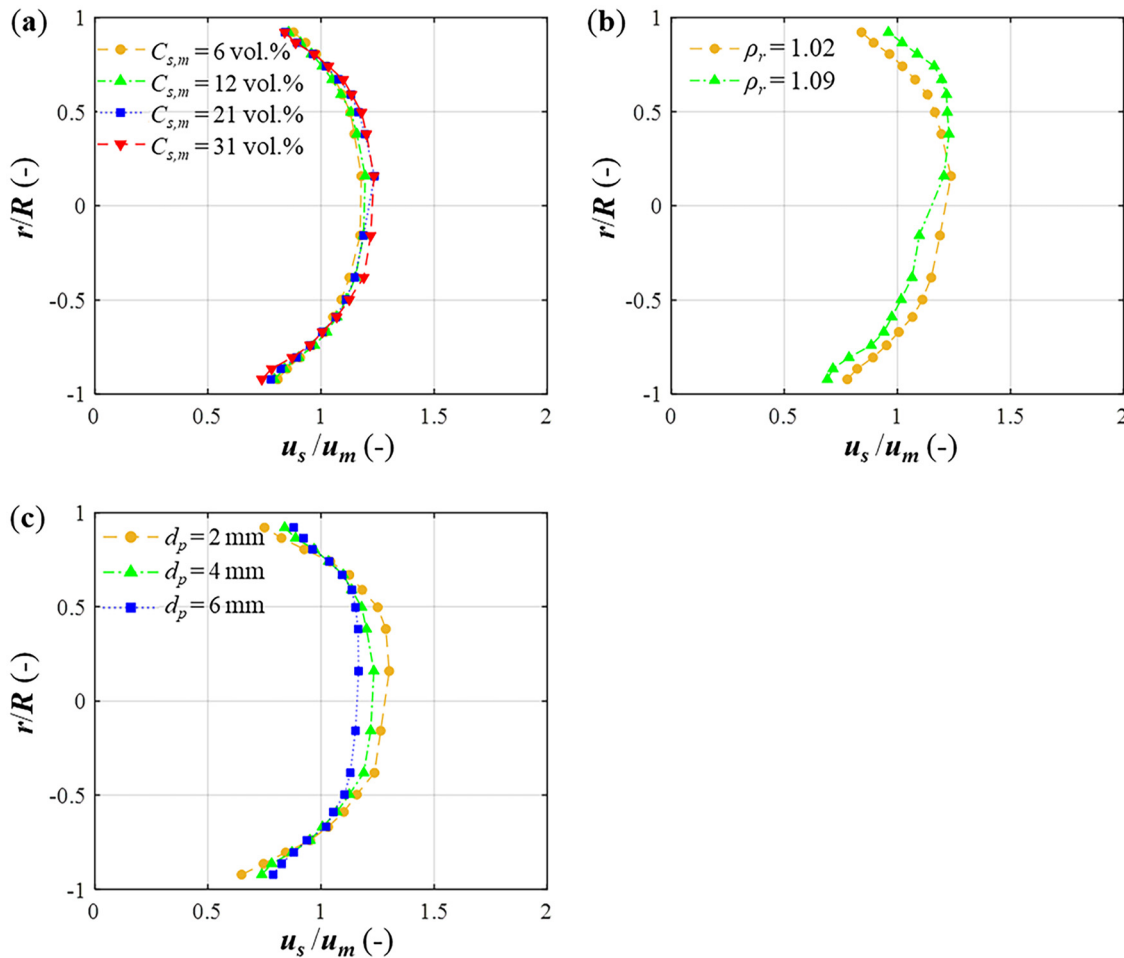

FIG. 17. Effects of different features on particle velocity distribution: (a) $C_{s,m}$ ($d_p = 4$ mm, $\rho_r = 1.02$); (b) ρ_r ($C_{s,m} = 21$ vol. %, $d_p = 4$ mm); and (c) d_p ($C_{s,m} = 31$ vol. %, $\rho_r = 1.02$).

was conducted five times to assess the feature influence on determining the solid phase velocity, as shown in Fig. 16. To make it easier to understand, Fig. 17 is plotted to show the influence of different features on the solid phase velocity distribution. The feature analysis in Fig. 16 shows that $C_{s,m}$ has the minimum influence on the particle velocity distribution and Fig. 17(a) shows that changing particle concentration from 6 to 31 vol. % has a minimal effect on the normalized particle velocity profile. Moreover, ρ_r has the most substantial influence on the particle velocity (Fig. 16), which is corroborated by Fig. 17(b), showing that the particle velocity distribution changes significantly by varying ρ_r from 1.02 to 1.09. The influence of particle size (d_p) is moderate, as indicated by both Figs. 16 and 17(c). For a solid phase, which is denser than the liquid phase, the vertical position (y) of a tracer particle in the pipe will always play a more important role than its horizontal position (z) (Fig. 1). Thus, considering the flow conditions considered in this study (Table I), ρ_r and y are the most influential parameters on the spatial particle velocity distribution, while $C_{s,m}$ has the smallest effect.

V. CONCLUSION

A machine learning hybrid algorithm for predicting turbulent single-phase and two-phase particle–liquid flows has been trained and validated using 3D Lagrangian phase flow trajectories determined by PEPT. Large experimental datasets consisting of hundreds of phase trajectories are required to accurately determine the phase velocity fields in single-phase flow and two-phase particle–liquid flow, as well as the spatial phase distribution. The algorithm, however, requires relatively small datasets (an order of magnitude less) for training to give predictions on the same level of accuracy for the velocity fields and spatial distributions of the phases. Furthermore, the hybrid algorithm is capable of extracting hidden inherent information to reliably predict such characteristics for flows under different conditions within and outside the limits of the range of the original experimental data bank, which has potential for facilitating and improving process design and operation.

The feature importance analysis showed that the particle velocity field is mostly influenced by particle-to-liquid density ratio and particle vertical radial position. Particle horizontal radial position and particle size have moderate effects, while particle concentration has the smallest effect. Thus, the hybrid learning algorithm presented in this work is powerful and capable of dealing with complex dynamic flow systems. The algorithm could potentially be enhanced further by using more complex training databanks to tackle a much more comprehensive range of flow situations, leading to huge reductions in the cost of experimentation and numerical simulation.

SUPPLEMENTARY MATERIAL

See the supplementary material for illustration of the experimental flow loop and additional validation results of the presented hybrid ML algorithm.

ACKNOWLEDGMENTS

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

ZhuangJian Yang: Conceptualization (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (lead). **Kun Li:** Conceptualization (equal); Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – review & editing (supporting). **Mostafa Barigou:** Conceptualization (lead); Funding acquisition (lead); Methodology (lead); Project administration (lead); Resources (lead); Supervision (lead); Validation (equal); Writing – review & editing (lead).

DATA AVAILABILITY

The data that support the findings of this study are available within the article and its supplementary material.

NOMENCLATURE

Symbols

C_s	Local particle volume concentration (–)
$C_{s,m}$	Mean particle volume concentration (–)
d_p	Particle diameter (mm)
D	Pipe diameter (m)
k	Number of nearest neighbors used in KNN regressor (–)
N	Number of PEPT-determined trajectories
r	Radial position (m)
R	Pipe radius (m)
Re_L	Liquid Reynolds number (–)
u_f	Local fluctuation velocity predicted by the hybrid model (ms^{-1})
u_i^{pred}	Instantaneous velocity predicted by the hybrid model (ms^{-1})
$\bar{u}_i^{\mathrm{pred}}$	Mean velocity predicted by the hybrid model (ms^{-1})
u_L	Liquid phase velocity (ms^{-1})
u_m	Mean mixture velocity (ms^{-1})
u_s	Particle velocity (ms^{-1})
w	Weight function of the KNN regressor (–)
x, y, z	Cartesian coordinates corresponding to the pipe flow direction, gravity direction, and horizontal direction, respectively (m)

Greek symbols

μ_L	Liquid phase dynamic viscosity (Pa s)
ρ_L	Liquid phase density (kgm^{-3})
ρ_r	Particle-to-liquid density ratio (–)
ρ_s	Particle density (kgm^{-3})

Abbreviations

CFD	Computational fluid dynamics
KNN	k-nearest neighbors

08 November 2023 14:09:21

ML   Machine learning
PEPT Positron emission particle tracking
RMSE Root mean square error

REFERENCES

[1] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," ACM Comput. Surv. **51**, 1 (2019).

[2] S. Raza and C. Ding, "News recommender system: A review of recent progress, challenges, and opportunities," Artif. Intell. Rev. **55**, 749 (2022).

[3] P. Wang, E. Fan, and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," Pattern Recognit. Lett. **141**, 61 (2021).

[4] M. Q. Huang, J. Ninić, and Q. B. Zhang, "BIM, machine learning and computer vision techniques in underground construction: Current status and future perspectives," Tunnelling Underground Space Technol. **108**, 103677 (2021).

[5] T. K. Balaji, C. S. R. Annavarapu, and A. Bablani, "Machine learning algorithms for social media analysis: A survey," Comput. Sci. Rev. **40**, 100395 (2021).

[6] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, "Text preprocessing for text mining in organizational research: Review and recommendations," Organ. Res. Methods **25**, 114 (2022).

[7] H. Ma, L. Xu, Z. Javaheri, N. Moghadamnejad, and M. Abedi, "Reducing the consumption of household systems using hybrid deep learning techniques," Sustainable Comput. **38**, 100874 (2023).

[8] B. Wang, X. Wang, N. Wang, Z. Javaheri, N. Moghadamnejad, and M. Abedi, "Machine learning optimization model for reducing the electricity loads in residential energy forecasting," Sustainable Comput. **38**, 100876 (2023).

[9] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," SN Comput. Sci. **2**, 160 (2021).

[10] I. H. Sarker, A. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: An overview from machine learning perspective," J. Big Data **7**(1), 41 (2020).

[11] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," ACM Comput. Surv. **52**(1), 1 (2020).

[12] S. Safdar, S. Zafar, N. Zafar, and N. F. Khan, "Machine learning based decision support systems (DSS) for heart disease diagnosis: A review," Artif. Intell. Rev. **50**, 597 (2018).

[13] J. Ellenius and T. Groth, "Dynamic decision support graph—visualization of ANN-generated diagnostic indications of pathological conditions developing over time," Artif. Intell. Med. **42**, 189 (2008).

[14] X. Liu, C. Cai, W. Zhao, H.-J. Peng, and T. Wang, "Machine learning-assisted screening of stepped alloy surfaces for C1 catalysis," ACS Catal. **12**, 4252 (2022).

[15] S. M. Mousavi, W. Zhu, W. Ellsworth, and G. Beroza, "Unsupervised clustering of seismic signals using deep convolutional autoencoders," IEEE Geosci. Remote Sens. Lett. **16**, 1693 (2019).

[16] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD-96 Proceedings* (AAAI, 1996).

[17] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," Annu. Rev. Control Rob. Auton. Syst. **5**, 411 (2022).

[18] H. Fujiyoshi, T. Hirakawa, and T. Yamashita, "Deep learning-based image recognition for autonomous driving," IATSS Res. **43**, 244 (2019).

[19] K. Li, C. Savari, H. A. Sheikh, and M. Barigou, "A data-driven machine learning framework for modelling of turbulent mixing flows," Phys. Fluids **35**, 015150 (2023).

[20] S. R. Bukka, R. Gupta, A. R. Magee, and R. K. Jaiman, "Assessment of unsteady flow predictions using hybrid deep learning based reduced-order models," Phys. Fluids **33**, 013601 (2021).

[21] C. Drygala, B. Winhart, F. di Mare, and H. Gottschalk, "Generative modeling of turbulence," Phys. Fluids **34**, 035114 (2022).

[22] A. Hashemizadeh, A. Maaref, M. Shateri, A. Larestani, and A. Hemmati-Sarapardeh, "Experimental measurement and modeling of water-based drilling mud density using adaptive boosting decision tree, support vector machine, and K-nearest neighbors: A case study from the South Pars gas field," J. Pet. Sci. Eng. **207**, 109132 (2021).

[23] L. Liu, B. Hu, S. Liu, K. Wang, and H. Gu, "Recognition of gas-liquid flow regimes in helically coiled tube using wire-mesh sensor and KNN algorithm," Int. J. Multiphase Flow **154**, 104144 (2022).

[24] S. Dennis, S. Singh, and D. Ingham, "The steady flow due to a rotating sphere at low and moderate Reynolds numbers," J. Fluid Mech. **101**, 257 (1980).

[25] H. Zhu, Z. Li, X. Yang, G. Zhu, J. Tu, and S. Jiang, "Flow regime identification for upward two-phase flow in helically coiled tubes," Chem. Eng. J. **308**, 606 (2017).

[26] H. Zhai, Q. Zhou, and G. J. Hu, "BubbleNet: Inferring micro-bubble dynamics with semi-physics-informed deep learning," arXiv:2105.07179 (2021).

[27] B. Ouyang, L.-T. Zhu, Y.-H. Su, and Z.-H. Luo, "A hybrid mesoscale closure combining CFD and deep learning for coarse-grid prediction of gas-particle flow dynamics," Chem. Eng. Sci. **248**, 117268 (2022).

[28] R. Zisselmar and O. Molerus, "Investigation of solid-liquid pipe flow with regard to turbulence modification," Chem. Eng. J. **18**, 233 (1979).

[29] C. Savari and M. Barigou, "Lagrangian wavelet analysis of turbulence modulation in particle–liquid mixing flows," Phys. Fluids **34**, 115121 (2022).

[30] H. A. Sheikh, C. Savari, and M. Barigou, "Lagrangian stochastic modelling of liquid flow in a mechanically agitated vessel," Chem. Eng. Sci. **249**, 117318 (2022).

[31] K. Li, C. Savari, and M. Barigou, "Computation of Lagrangian coherent structures from experimental fluid trajectory measurements in a mechanically agitated vessel," Chem. Eng. Sci. **254**, 117598 (2022).

[32] S. Heinz, R. Mokhtarpoor, and M. Stoellinger, "Theory-based Reynolds-averaged Navier–Stokes equations with large eddy simulation capability for separated turbulent flow simulations," Phys. Fluids **32**, 065102 (2020).

[33] Z. Wei, J. Zhang, R. Jia, and J. Gao, "An improved method for coherent structure identification based on mutual K-nearest neighbors," J. Turbul. **23**, 655 (2022).

[34] A.-J. Gallego, J. Calvo-Zaragoza, J. J. Valero-Mas, and J. R. Rico-Juan, "Clustering-based k-nearest neighbor classification for large-scale data with neural codes representation," Pattern Recognit. **74**, 531 (2018).

[35] Z. Yang, C. Savari, and M. Barigou, "Numerical and experimental investigations of horizontal turbulent particle–liquid pipe flow," Ind. Eng. Chem. Res. **61**, 12040 (2022).

[36] M. Eesa and M. Barigou, "Horizontal laminar flow of coarse nearly-neutrally buoyant particles in non-Newtonian conveying fluids: CFD and PEPT experiments compared," Int. J. Multiphase Flow **34**, 997 (2008).

[37] P. G. Fairhurst, M. Barigou, P. J. Fryer, J. P. Pain, and D. J. Parker, "Using positron emission particle tracking (PEPT) to study nearly neutrally buoyant particles in high solid fraction pipe flow," Int. J. Multiphase Flow **27**, 1881 (2001).

[38] C. Savari, K. Li, and M. Barigou, "Multiscale wavelet analysis of 3D Lagrangian trajectories in a mechanically agitated vessel," Chem. Eng. Sci. **260**, 117844 (2022).

[39] P. Pianko-Oprych, A. Nienow, and M. Barigou, "Positron emission particle tracking (PEPT) compared to particle image velocimetry (PIV) for studying the flow generated by a pitched-blade turbine in single phase and multi-phase systems," Chem. Eng. Sci. **64**, 4955 (2009).

[40] M. Barigou, "Particle tracking in opaque mixing systems: An overview of the capabilities of PET and PEPT," Chem. Eng. Res. Des. **82**, 1258 (2004).

[41] L. Liu and M. Barigou, "Experimentally validated computational fluid dynamics simulations of multicomponent hydrodynamics and phase distribution in agitated high solid fraction binary suspensions," Ind. Eng. Chem. Res. **53**, 895 (2014).

[42] A. Guida, A. W. Nienow, and M. Barigou, "PEPT measurements of solid–liquid flow field and spatial phase distribution in concentrated monodisperse stirred suspensions," Chem. Eng. Sci. **65**, 1905 (2010).

[43] A. Guida, A. W. Nienow, and M. Barigou, "Mixing of dense binary suspensions: Multi-component hydrodynamics and spatial phase distribution by PEPT," AIChE J. **57**, 2302 (2011).

[44]X. Lian, C. Savari, K. Li, and M. J. P. O. F. Barigou, "Coupled smoothed particle hydrodynamics and discrete element method for simulating coarse food particles in a non-Newtonian conveying fluid," Phys. Fluids **35**, 043325 (2023).

[45]E. Ulker and M. Sorgun, "Comparison of computational intelligence models for cuttings transport in horizontal and deviated wells," J. Pet. Sci. Eng. **146**, 832 (2016).

[46]X. Zhang, H. Xiao, R. Gao, H. Zhang, and Y. Wang, "K-nearest neighbors rule combining prototype selection and local feature weighting for classification," Knowl.-Based Syst. **243**, 108451 (2022).

[47]K. Li, C. Savari, and M. Barigou, "Predicting complex multicomponent particle–liquid flow in a mechanically agitated vessel via machine learning," Phys. Fluids **35**, 053301 (2023).

[48]L. Breiman, "Random forests," Mach. Learn. **45**, 5 (2001).

08 November 2023 14:09:21