



Deposited via The University of Leeds.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/221356/>

Version: Accepted Version

Proceedings Paper:

Mortimer, L.F. and Fairweather, M. (2024) Development and Demonstration of an Improved Hybrid Machine Learning Algorithm for Predicting Lagrangian Trajectories in Turbulent Wall-Bounded Flows. In: American Institute of Physics Conference Proceedings. International Conference on Applied Physics, Simulation and Computing (APSAC 2024), 20-22 Jun 2024, Rome, Italy. American Institute of Physics. ISSN: 0094-243X. EISSN: 1551-7616.

This is an author produced version of a conference paper accepted for publication in Powder Technology, made available under the terms of the Creative Commons Attribution License (CC-BY), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: <https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Development and Demonstration of an Improved Hybrid Machine Learning Algorithm for Predicting Lagrangian Trajectories in Turbulent Wall-bounded Flows

Lee Mortimer^{1,a)} and Michael Fairweather¹

Author Affiliations

¹*School of Chemical and Process Engineering
University of Leeds
Woodhouse Lane, Leeds LS2 9JT
UNITED KINGDOM*

Author Emails

^{a)} *Corresponding author: l.f.mortimer@leeds.ac.uk*

Abstract. This work focuses on the development and demonstration of a predictive model for simulating particle-laden, wall-bounded turbulent flows using a hybrid machine learning (ML) algorithm. By informing the model with training data from direct numerical simulation (DNS) and Lagrangian particle tracking (LPT) in turbulent channel flows, the model aims to reduce the required DNS-LPT particle trajectory data needed to obtain statistically smooth dynamic property profiles whilst maintaining high accuracy. The research validates the efficacy of the hybrid ML algorithm, a k -nearest neighbours model, in predicting particle trajectories and their resulting dynamic statistics. A Gaussian noise model is first validated and demonstrated to predict the correct distributions of velocity fluctuations throughout the channel flow domain. Hyperparameter optimization of the hybrid ML algorithm indicates that $k = 10$ and $N_p = 1000$ maximizes R^2 and minimizes the root mean squared error. The model is used to predict particle trajectories in a $Re_\tau = 180$ turbulent channel flow. Comparisons with DNS-LPT results show that the hybrid ML model accurately predicts mean streamwise velocities and provides a good approximation of root-mean-square velocity fluctuations, especially near the channel walls. While some minor discrepancies in the spanwise and wall-normal velocity fluctuations were noted in the core region of the channel, the overall agreement between the DNS-LPT results and the ML predictions serves to demonstrate the potential of the hybrid ML algorithm for efficiently simulating particle advection and wall-interaction in turbulent flows. The model also demonstrates a significant reduction in computational time, requiring only 34.16 seconds for an equivalent simulation that traditionally takes over 24 hours with DNS-LPT methods on high performance computing facilities. The study highlights further potential improvements to the algorithm, such as by adjusting the fluctuation distribution based on wall-normal position in order to enhance the accuracy obtained in the core region. Despite this, the hybrid ML approach offers substantial benefits in terms of reduced computational costs and increased efficiency, making it a promising tool for real-time applications in digital twin technologies and other fields.

Key-Words: *Machine learning, direct numerical simulation, Lagrangian particle tracking, channel flows, wall-bounded flows, k -nearest neighbours.*

INTRODUCTION

Across the past decade, machine learning (ML) has been utilized in many fields, acting as a fundamental computational tool. It has been used to augment analysis, study, prediction and decision-making across a broad range of complex systems. For instance, ML can provide such systems with the capacity to learn and improve from experience without the need for additional human intervention, providing sufficient training data is available or can be generated. There has been a significant increase in ML applications across various fields in recent years. In the healthcare industry, ML aids in the diagnosis and prediction of diseases by analyzing medical data and imaging [1]. In finance, ML algorithms are used for fraud detection, calculations of credit score, and algorithmic financial trading [2]. Furthermore, ML plays a crucial role in the development of self-driving vehicles, enabling self-driving cars to navigate, make split-second decisions and react in real time using the available journey and local data based on their surroundings [3]. The basis of these applications is to derive insights from available data in order to build intelligent

models based on this information, which can subsequently be used to either make logical, optimized decisions or generate knowledge which also aids in working towards a fundamental understanding of the system.

Motivated by access to large quantities of generated or measured data in both simulations and experiments, ML has recently received significant attention in the fluid dynamics community for addressing challenges such as turbulence closure modelling, reduced-order modelling, and flow optimization and control. These advancements leverage both high-fidelity data generated by first-principles numerical simulations, as well as by advanced experimental techniques [4]. For instance, [5] used a random forest regression-based ML algorithm to predict fluid flow in curved pipes, with training data generated from simulations. The model slightly mispredicted the positions of additional vortices, but it accurately mapped the important characteristics of the turbulent fluid flow. [6] employed a k-nearest neighbours (KNN) algorithm, aiming to learn the key features of particle-liquid flows in a mixing tank using Lagrangian trajectories obtained from experiments. Their results demonstrated the KNN algorithm's capability to learn from and further predict more complex dynamic systems. This technique was also employed by [7], who developed a hybrid ML algorithm which included a preprocessor, k-nearest neighbours regressor, noise generator, and particle-wall collision model to predict features of turbulent single-phase and particle-fluid flows in pipes. The authors demonstrated that the algorithm could accurately learn and predict local liquid and particle velocities, as well as the radial distribution of particle concentration, using dynamic Lagrangian trajectory databases generated from experiments. [8] used deep learning to predict the statistical properties of particle-laden vertical pipe flows, showing that deep neural network models accurately capture the system's dynamics and are particularly effective for particle statistics. Their findings demonstrated that these types of models are capable of significantly speeding up the model development and design process in industrial applications such as powder-based laser metal deposition. Further neural-network implementations have also been performed in homogeneous isotropic turbulence, improving the accuracy of recovering mechanisms such as preferential concentration [9].

For non-trivial multiphase systems such as particle-laden wall-bounded turbulent flows, the complexity of the underlying fundamental interactions between the two phases is the primary obstacle to investigation. In such systems, the local fluid velocity field fluctuates due to the generation and dissipation of turbulent eddies, and fluid and particle velocities are also randomly influenced by the local fluid properties, as well as particle-wall collisions. Additionally, the strong mutual influence between the solid and fluid phases, heavily impacted by particle concentration, particle-fluid density ratio and wall proximity distance, can result in turbulence modulation, further complicating the flow dynamics. Understanding these phenomena is essential for comprehending and optimising designs of analogous industrial-scale systems and clues to unravelling their nature are captured in the Lagrangian flow trajectories of both the continuous and dispersed phases.

The present work aims to develop an improved predictive model capable of estimating key fluid and particle behaviours using a hybrid ML algorithm. By leveraging dynamic databases derived from direct numerical simulation (DNS) and Lagrangian particle tracking (LPT) in turbulent channel flows, this model aims to reduce the required trajectory data needed to resolve dynamic statistical profiles significantly while still maintaining similar levels of accuracy. By training on simulation-generated databases of particle trajectories and local fluid dynamic properties, the model also aims to forecast further (and generate new) particle trajectories, recovering statistical profiles, within and beyond the bounds of the training data.

Improving on previous models, the current training algorithm considers the particle's previous velocity, which imparts the notion of particle momentum to the technique and avoids unrealistic motion. A primary application of such a capability would be in advancing digital twin technology, wherein the push towards real-time simulations is greatly advanced with the associated reduced computational cost from pre-trained ML algorithms.

PROBLEM FORMULATION

Direct numerical simulation

To achieve highly precise predictions of the flow field, encompassing all relevant turbulence length and timescales, we employ the DNS code, Nek5000 [10]. This software is utilized to simulate turbulent channel flows at a shear Reynolds number $Re_\tau = u_\tau \delta / \nu_F = 180$, where u_τ represents the shear velocity, δ represents the channel half-height, and ν_F denotes the fluid kinematic viscosity. The Eulerian solver within Nek5000 employs a high-order ($N=7$) spectral element method (SEM) to model the temporal and spatial evolution of the fluid phase. The selection of Nek5000 was based on its efficient parallelization capabilities, ease of use when incorporating the LPT, and extensive history of validation. The governing equations for the continuous phase dynamics, expressed in dimensionless, incompressible form, comprise the continuity and Navier-Stokes equations, as follows:

$$\nabla \cdot \mathbf{u}_F^* = 0, \quad (1)$$

$$\frac{D\mathbf{u}_F^*}{Dt^*} = -\nabla p^* + \frac{1}{Re_B} \nabla^2 \mathbf{u}_F^* + \mathbf{f}_{PG}^*. \quad (2)$$

Here, $\mathbf{u}_F^*(\mathbf{x}^*, t^*)$ is the fluid velocity vector at position $\mathbf{x}^*$ and time t^* , $p^*(\mathbf{x}, t)$ is the fluid pressure, Re_B is the bulk Reynolds number defined as $Re_B = U_B \delta / \nu_F$ and $\mathbf{f}_{PG}^*$ the constant pressure gradient forcing term. The equations presented above are nondimensionalized utilizing the channel half-height, δ , the bulk velocity, U_B , and the density of the fluid phase, ρ_F . The conservation equations above are solved numerically on a discretized, structured Cartesian grid. This grid comprises $27 \times 18 \times 23$ 7th-order spectral elements, totaling approximately 3.9 million equivalent Gauss-Lobatto-Legendre (GLL) nodes. In the wall-normal direction of the channel, the grid is scaled to accommodate more densely distributed elements closer to the wall. Conversely, the distribution of elements is uniform in the streamwise and spanwise directions. The computational domain (x, y, z) spans dimensions of $12\delta \times 2\delta \times 6\delta$, representing a channel, with boundary lengths chosen to ensure the capture of all the largest vortical structures. This domain is depicted in Fig. 1.

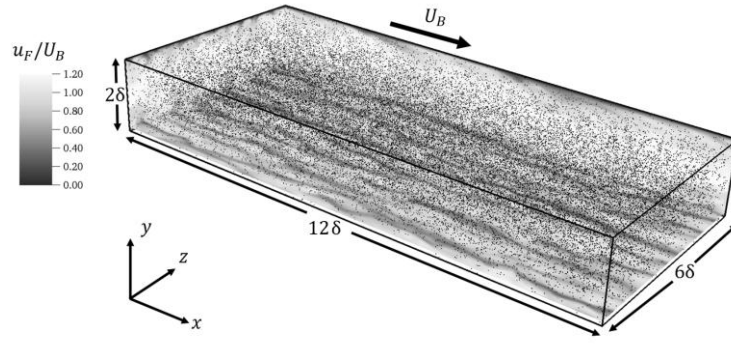


FIGURE 1. Schematic of the particle-laden turbulent channel flow at $Re_\tau = 180$ used in the DNS-LPT simulations.

The flow is maintained by a constant pressure gradient imposed in the streamwise (x) direction. The prescribed magnitude of the pressure gradient is as follows:

$$\frac{\partial p^*}{\partial x^*} = \left(\frac{Re_\tau}{Re_B} \right)^2. \quad (3)$$

Lagrangian particle tracking

To trace the trajectories of solid particles within the flow, a Lagrangian particle tracking routine was developed and implemented, which operates concurrently with Nek5000. Each particle in the ensemble is represented as a point-like, impenetrable, and undeformable computational sphere. Following a time step for the continuous phase, the LPT routine solves the non-dimensional equations of motion for each particle, considering the force balance between the particle's inertia and the fluid, as described by [11] and [12]. A primary aim of the development of the hybrid ML algorithm is to generalize the model across the Stokes number range, and consequently forces such as lift, virtual mass, and pressure gradient, in addition to drag, are likely to be relevant under certain conditions, as observed in previous studies [13], and as such are considered in the calculations. The Basset history force is however omitted as in previous studies due to its lengthy computation times and previous findings indicating minimal impact on particle motion [14]. The equations of motion solved for each particle are detailed as follows:

$$\frac{\partial \mathbf{x}_p^*}{\partial t^*} = \mathbf{u}_p^*, \quad (4)$$

$$M_{VM} \frac{\partial \mathbf{u}_p^*}{\partial t^*} = \underbrace{\frac{3C_D |\mathbf{u}_s^*|}{4d_p^* \rho_p^*} \mathbf{u}_s^*}_{\text{Drag}} + \underbrace{\frac{3}{4} \frac{C_L}{\rho_p^*} (\mathbf{u}_s^* \times \boldsymbol{\omega}_F^*)}_{\text{Lift}} + \underbrace{\frac{1}{2\rho_p^*} \frac{D\mathbf{u}_F^*}{Dt^*}}_{\text{Virtual Mass}} + \underbrace{\frac{1}{\rho_p^*} \frac{D\mathbf{u}_F^*}{Dt^*}}_{\text{Pressure Gradient}}. \quad (5)$$

In Eqs. (4) and (5), $\mathbf{x}_p^*$ is the particle position vector, $\mathbf{u}_p^*$ the particle velocity vector, $\mathbf{u}_F^*$ the fluid velocity vector spectrally interpolated at the position of the particle, $\mathbf{u}_s^* = \mathbf{u}_F^* - \mathbf{u}_p^*$ the slip velocity between the fluid and the particle, d_p^* the particle diameter non-dimensionalized by the channel half-height, ρ_p^* the density ratio between the fluid and the particle and $\boldsymbol{\omega}_F^*$ the vorticity of the fluid interpolated spectrally at the particle position, given by $\boldsymbol{\omega}_F^* = \nabla \times \mathbf{u}_F^*$.

M_{VM} is the virtual mass modification term given by $M_{VM} = \left(1 + \frac{1}{2\rho_p^*}\right)$. The drag coefficient, C_D , is calculated using the correlations of [15], where $C_D = 24f_D/Re_p$, with $f_D = (1 + 0.15Re_p^{0.687})$ when $Re_p > 0.5$ and $f_D = 24/Re_p$ otherwise (in the Stokes regime). Here, Re_p is the particle Reynolds number, given by $Re_p = Re_B d_p^* |\mathbf{u}_s^*|$. Further details on the calculation and origins of these terms are available in [11]. Particle motion is computed following the completion of a fluid time step. Initially, spectral interpolation is utilized to acquire the fluid velocity and spatial velocity derivative fields. Subsequently, Eqs. (4) and (5) are integrated employing a fourth-order accurate Runge-Kutta scheme, with a time step (Δt^*) matching that of the continuous phase solver. Interactions between particles and the channel walls are identified and resolved through elastic collisions, where the particle's wall-normal velocity is reversed upon collision with the channel wall. In the periodic directions (streamwise and spanwise), particles departing from the boundary are reintroduced at the corresponding location on the opposite side of the computational domain, preserving the periodic behaviour of the channel flow. Fluid and particle properties used to demonstrate the algorithm are presented in Table 1.

TABLE 1. Simulation properties used in demonstration of hybrid ML algorithm

Parameter	Value
Re_τ	180
ρ_p^*	2.5
d_p^*	0.0025
St^+	0.028
$N_{p,SIM}$	10,000

Hybrid ML algorithm

Lagrangian trajectories generated from the simulation methods presented in Sections 2.1 and 2.2 are first pre-processed into a new data array consisting of their vertical positions within the channel and their velocity components: $(y^*, u_{x,t}^*, u_{y,t}^*, u_{z,t}^*)$. These are combined with their simulation conditions, as well as their velocities at the previous timestep to create the full input and predictor feature array, which additionally consists of $(Re_\tau, \rho_p^*, d_p^*, u_{x,t-1}^*, u_{y,t-1}^*, u_{z,t-1}^*)$.

A KNN algorithm, which was chosen due to its efficiency and modelling speed as well as its robust predictions, is trained using the preprocessed data. In previous similar studies, this technique was shown to demonstrate faster modelling speeds and yield more reliable predictions than alternative approaches [16]. In terms of root-mean-square error (RMSE), KNN was shown to outperform support vector regression, linear regression, and artificial neural network models. The technique stores the training data in an n -dimensional space and predicts an unknown data point by assessing its similarity to the k nearest data points in the training set using a weight function, which are the two hyperparameters associated with the technique to be optimized. Before training, the input feature variables are normalized between 0 and 1 based on the minimum and maximum value of the data associated with that variable. The training procedure reads data from all $N_{p,SIM} = 10,000$ particles across 1000 instantaneous time states at intervals of $t^* = 0.05$. The effect of the number of trajectories used to train the algorithm on its performance is assessed in the analysis.

Once suitably trained, to construct fictional particle trajectories, the algorithm first initializes the position of each particle within the boundaries of the original channel flow. They are then assigned equivalent velocities based on the non-dimensional bulk velocity $U_B^* = 1.0$. The ML-informed simulation is then performed, which predicts the subsequent particle velocity based on the input feature array. Additionally, a Gaussian noise model is included to predict the local velocity fluctuations, $u'_F \sim \mathcal{N}(\mu, \sigma^2)$, where the values of the mean, μ , and standard deviation, σ , are validated against measurements of the fluid flow field from DNS predictions. Using this, the predicted velocity is calculated using:

$$u_p = \bar{u}_p + u'_F, \quad (6)$$

with $\bar{u}_p$ generated through the trained hybrid KNN ML algorithm based on the particle properties, their previous velocities, and their wall-normal position within the channel domain. Particle positions are then updated using a standard Euler time-stepping algorithm. A particle-wall collision detection and handling procedure was also used, identical to that present in the LPT, wherein for the vertical direction, the direction of the velocity is reversed, and in the streamwise and spanwise direction the particle is reintroduced into the corresponding location in the opposite end of the channel.

PROBLEM SOLUTION

Initially, a simulation of unladen turbulent flow at SEM order $N = 7$ was conducted in the channel, using a pressure gradient driving force to establish a statistically stationary turbulence field for $Re_\tau = 180$. The simulation began with an initial condition featuring a mean velocity profile with minor perturbations to promote the transition to turbulence. The simulation ran for $T_S^* = 100$ non-dimensional time units, with statistics collected during the final $50 \leq t^* \leq 100$. Additionally, statistics were measured at $t^* = 10$ intervals to ensure no temporal variations in the results. Figure 2 displays the mean streamwise velocity of the continuous phase, the root-mean-square (RMS) of velocity fluctuations, and the shear stress, compared with the DNS results from [17] for the $Re_\tau = 180$ simulation. The agreement is found to be very good, with only slight deviations in the shear stress profile. This may be attributed to an increased number of equivalent Gauss-Lobatto-Legendre nodes (3.9M) in the present study due to increased element density throughout the domain, compared to the computational mesh used in the validation reference (2.1M). Overall, the results generate confidence in the continuous phase predictions, and acts as a suitable foundation within which particle trajectories may be further obtained using Lagrangian particle tracking.

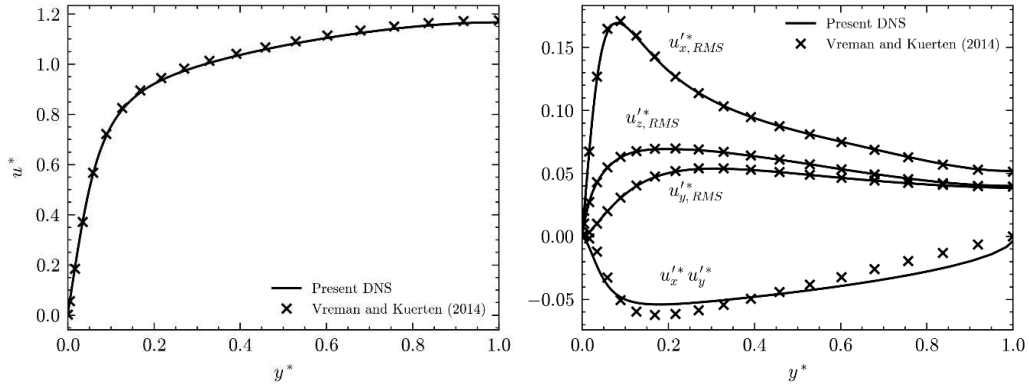


FIGURE 2. Validation of continuous phase first- and second-order fluid velocity statistics for $Re_\tau = 180$. Results compared against [17]. Left: mean streamwise velocity and right: root-mean-square of velocity fluctuations and shear stress.

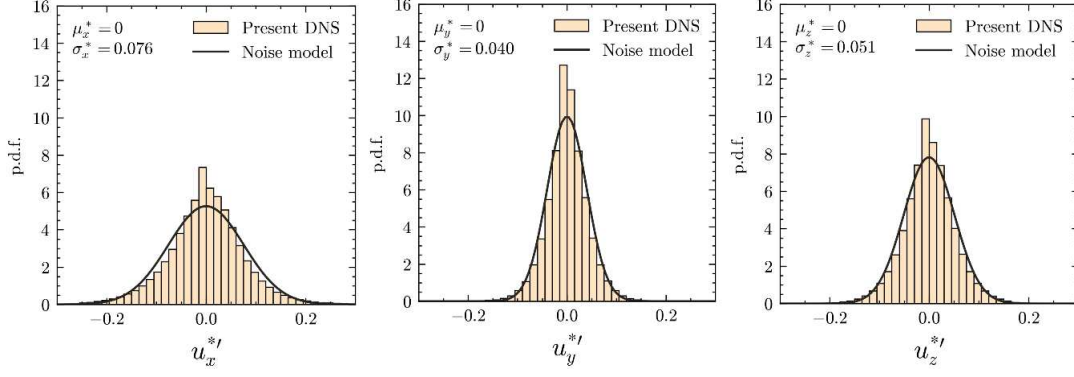


FIGURE 3. Gaussian noise model validation of velocity fluctuation components for single-phase DNS at $Re_\tau = 180$. Left: streamwise; middle: wall-normal; right: spanwise.

From the validated channel flow, statistics were gathered surrounding the velocity fluctuations throughout the domain across a new simulation of temporal duration, $t^* = 100$. Figure 3 illustrates the probability distribution functions (p.d.f.) for all three components of the velocity fluctuations, sampled from uniformly distributed locations throughout the entire channel domain and across the full simulation. The Gaussian noise model, used to predict the local fluctuations on the particles, is validated by comparing the sampled fluctuation velocities. In all three cases, the noise model provides a strong representation of the range and distribution of fluctuating fluid velocities. It is noted here that the characteristics of these profiles may differ depending on wall-proximity, however previous studies [9] demonstrated that the correct mean flow statistics are captured by sampling over the whole domain. That said, it is expected that a distance-based profile may improve the findings further, at the cost of increased complexity in the hybrid algorithm used to generate new particle trajectories.

To optimize the hybrid ML algorithm, the effect of various quantities on the root-mean-square error and the R^2 coefficient of determination was studied. These included the k parameter (or number of nearest neighbours considered in the training) and the number of sample trajectories used, N_p . The ML algorithm was trained on 70% of the trajectories generated from DNS, with the remaining 30% used as testing data in order to assess the performance via two error metrics, the coefficient of determination (R^2) and the root-mean-square error (RMSE). Figure 4 justifies the hyperparameter choices during the optimization of k and N_p . As the number of particle trajectories trained on increases, the k value at which the performance of the predictive algorithm is optimal also increases. For the comparison simulations using the hybrid ML model, we chose to use $k = 10$, $N_p = 1000$ since this provides an R^2 value close to unity whilst also an RMSE of around 0.25%. Beyond this point, the RMSE begins to increase, with the KNN algorithm taking into account too many nearby datapoints, likely contaminating the predictions with parts of trajectories which are irrelevant to the current trajectory. This is somewhat alleviated by an increase in the number of trajectories trained on because in this case, the number of local (in feature space) trajectories is then likely to be higher, so the algorithm can consider more similar datapoints. Note that the weight function metric was also studied, with the ‘distance’ metric performing slightly better than alternatives.

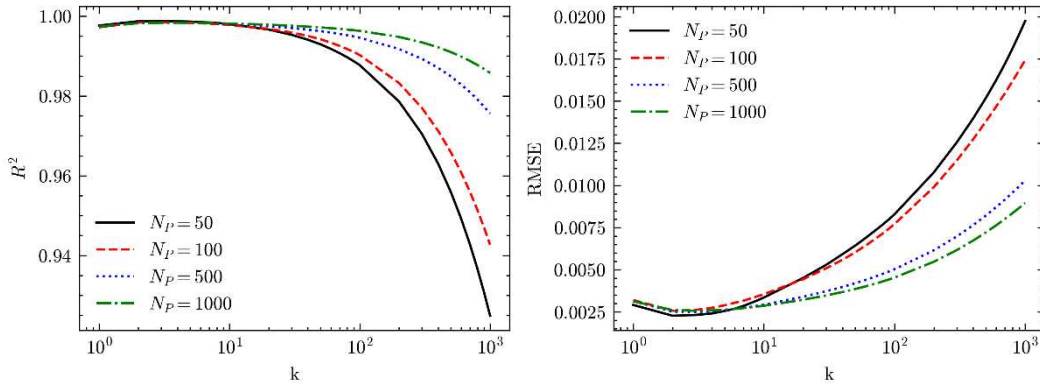


FIGURE 4. Effect of the number of particle trajectories used to train the hybrid ML algorithm on the R^2 value (left) and the RMSE (right) as a function of the hyperparameter, k .

Once trained, the model was used to perform simulations of particle-laden flows to compare against the DNS-LPT results of equivalent simulations. Figure 5 compares the trajectories of particles between the real and hybrid ML simulations. Good agreement is obtained visually, though it is clear that the wall-normal or lateral motion is slightly more pronounced in the DNS as compared to the ML simulation, particularly in the core of the channel flow. Due to the low Stokes number in the present particle species concerned here, $St^+ = 0.028$, more complex conveying and transport mechanisms such as turbophoresis or preferential concentration are very low in impact, and as such the cross-stream motion is low overall. This leads to trajectories sampling similar wall-distance regions throughout most of its trajectory, with very low wall migration experienced. The hybrid ML simulation captures this well, with trajectories mostly streamwise oriented, and further captures the more chaotic random motion close to the wall. Furthermore, the velocity magnitudes exhibit very similar wall-distance dependance. This is further confirmed in Fig. 6 (left) which compares the predicted mean streamwise velocities for the hybrid ML particles against the DNS-LPT values. Throughout the wall-normal direction of the channel, excellent agreement is obtained. Equivalent comparisons for the RMS velocity fluctuations are presented in Fig. 6 (right), which demonstrate strong agreement close to the wall, but discrepancies are present in the core of the channel. These are more pronounced for the wall-normal and spanwise components, as was observed in Fig. 5. It is clear that the hybrid ML algorithm overpredicts the velocity fluctuations, likely due to the way in which the fluctuations are factored into the simulation. As mentioned earlier, the fluctuations are currently directly imparted onto the fictional trajectories, with no wall-normal dependance. In a real flow, the local turbulence fluctuations addition to the mean conveying velocity that a particle experiences varies with wall-distance, as demonstrated in Fig 2. Improvements could hence be made by using a wall-normal position-based fluctuation distribution, which would lower the induced fluctuations in the core region of the channel. That said, trained on only 1000 trajectories, this technique shows strong promise for efficiently simulating the motion of particles in wall-bounded flows in a timely manner.

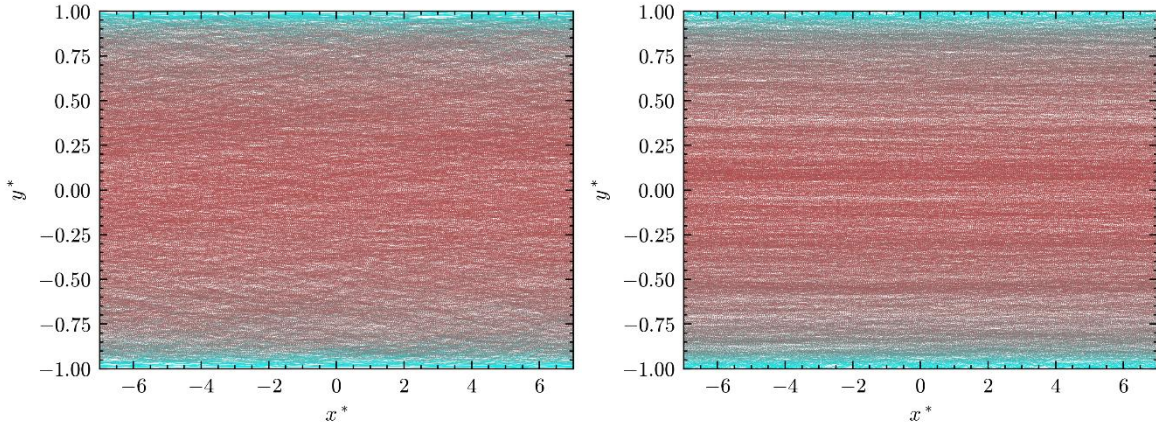


FIGURE 5. 1000 particle trajectories (left: present DNS; right: hybrid ML-predicted trajectories) in the (y^*, x^*) plane using 300 particles as training data (tested on 300) to reproduce 1000 trajectories. Colour represents particle streamwise velocity with decrease going from red to cyan. ML hyperparameters: $k = 10$, $w = \text{'distance'}$.

The hybrid ML algorithm took 34.16 seconds to simulate an equivalent amount of non-dimensional time, and when compared with the DNS-LPT simulation which took over 24 hours, the benefit is clear. Furthermore, the DNS-LPT technique ran on 36 computational cores, whereas the present ML technique, once trained, was performed on a standard desktop specification PC running on a single core.

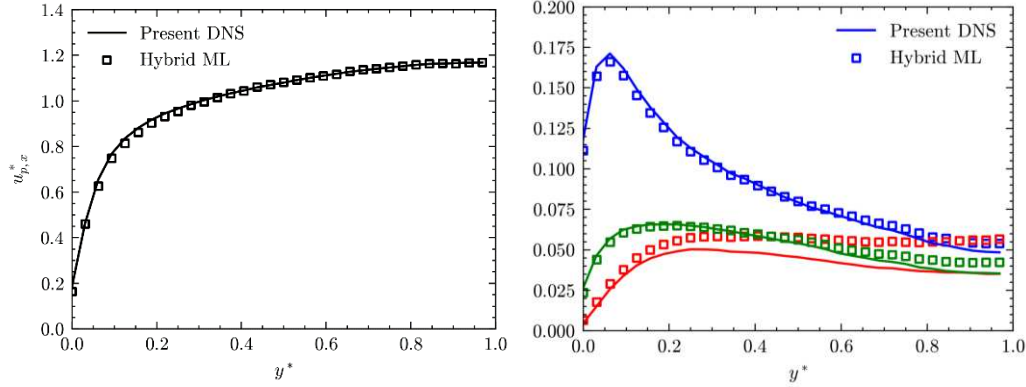


FIGURE 6. Comparison of mean particle streamwise velocity (left) and root-mean-square velocity fluctuations (right) predictions between the present DNS results and the hybrid ML-predicted trajectories. ML hyperparameters: $k = 10$, $w =$ ‘distance’. Blue: streamwise; red: wall-normal, green: spanwise.

CONCLUSIONS

This study aimed to develop and demonstrate an improved predictive model for simulating particle dispersion, advection and wall-interaction in turbulent channel flows using a hybrid machine learning algorithm. By leveraging dynamic databases obtained from DNS and LPT, we also aimed to significantly reduce the required simulation data needed to obtain statistically stationary and accurate profiles while maintaining prediction accuracy.

The study successfully demonstrates the efficacy of the hybrid ML algorithm, incorporating a KNN model, in predicting particle trajectories and fluid dynamic properties. The model not only significantly reduces the number of required DNS-LPT trajectories necessary to train the model and generate additional ML-informed trajectories, it also leads to a substantial decrease in computational time and resources in order to recover similar dynamics and statistical data. For example, simulating an identical system for an equivalent amount of non-dimensional time took only 34.16 seconds using the hybrid ML approach, compared to over 24 hours with traditional DNS-LPT methods on high performance computing equipment.

The accuracy of the predictions was validated through comparisons with DNS-LPT results, showing that the hybrid ML model accurately predicts mean streamwise velocities and provides a good approximation of RMS velocity fluctuations, especially near the channel walls. Though minor discrepancies were observed in the core region of the channel, particularly in the wall-normal and spanwise components, the overall agreement between the ML predictions and DNS-LPT results was strong. These findings serve to demonstrate the potential of the hybrid ML algorithm to efficiently simulate the motion of particles in wall-bounded flows, making it a promising tool for real-time applications. The study also identifies areas for further improvements, such as adjusting the fluctuation distribution based on the wall-normal position to improve accuracy in the core region of the channel. Despite these areas for improvement, the reduced computational cost and increased efficiency make this hybrid ML approach highly suitable for digital twin technology, where real-time simulation and analysis are critical.

Possible limitations of the present model include that mechanisms such as turbophoresis or preferential concentration may not be fully captured, since training data demonstrating that behaviour is not currently used to train the model. Future work should focus on firstly further optimising the model. The potential to use artificial neural networks, or convolution neural networks on particle histories has shown potential in other similar applications [8] [9]. Improvements to the model should be tested on a broader range of flow conditions, continuous phase geometries and particle species in order to verify its generalisability. Finally, the model should be used in to augment data from experimental particle image velocimetry trajectories, demonstrating its utility in both academic research and practical applications.

ACKNOWLEDGMENTS

The authors are grateful to the UK Engineering and Physical Sciences Research Council for funding through the TRANSCEND (Transformative Science and Engineering for Nuclear Decommissioning) project (EP/S01019X/1), and Sellafield Ltd. for funding from the University of Leeds-Sellafield Ltd, Centre of Expertise for Sludge (Particulates & Fluids)

REFERENCES

1. A. Barragán-Montero, U. Javaid, G. Valdés, D. Nguyen, P. Desbordes, B. Macq, S. Willems, L. Vandewinckele, M. Holmström, F. Löfman, and S. Michiels, Artificial intelligence and machine learning for medical imaging: A technology review, *Phys. Med.* **83**, 242–256 (2021). doi: 10.1016/j.ejmp.2021.04.016
2. A. Ali, S. Abd Razak, S. H. Othman, T. A. E. Eisa, A. Al-Dhaqm, M. Nasser, T. Elhassan, H. Elshafie, and A. Saif, Financial fraud detection based on machine learning: A systematic literature review. *Appl. Sci.* **12**(19), 9637 (2022). doi: 10.3390/app12199637
3. A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues, *Array* **10**, 100057 (2021). doi: 10.1016/j.array.2021.100057
4. S. L. Brunton, B. R. Noack, and P. Koumoutsakos, Machine learning for fluid mechanics, *Annu. Rev. Fluid Mech.* **52**, 477–508 (2020). doi: 10.1146/annurev-fluid-010719-060214
5. P. Jain, A. Choudhury, P. Dutta, K. Kalita, and P. Barsocchi, Random forest regression-based machine learning model for accurate estimation of fluid flow in curved pipes, *Processes* **9**(11), 2095 (2021). doi: 10.3390/pr9112095
6. K. Li, C. Savari, H. A. Sheikh, and M. Barigou, A data-driven machine learning framework for modeling of turbulent mixing flows, *Phys. Fluids* **35**, 015150 (2023). doi: 10.1063/5.0136830
7. Z. Yang, K. Li, and M. Barigou, Experimentally trained hybrid machine learning algorithm for predicting turbulent particle-laden flows in pipes, *Phys. Fluids* **35**, 113309 (2023). doi: 10.1063/5.0172609
8. A. Haghshenas, S. Hedayatpour, and R. Groll, Prediction of particle-laden pipe flows using deep neural network models, *Phys. Fluids* **35**, 083320 (2023). doi: 10.1063/5.0160128
9. J. Hu, Z. Lu, and Y. Yang, Improving prediction of preferential concentration in particle-laden turbulence using the neural-network interpolation, *Phys. Rev. Fluids* **9**(3), 034606 (2024). doi: 10.1103/PhysRevFluids.9.034606
10. P. F. Fischer, J. W. Lottes, and S. G. Kerkemeier, Nek5000, available at: <http://nek5000.mcs.anl.gov/>; accessed September 1, 2008.
11. M. R. Maxey, The gravitational settling of aerosol particles in homogeneous turbulence and random flow fields, *J. Fluid Mech.* **174**(1), 441–465 (1987). doi: 10.1017/S0022112087000193
12. J. J. Riley, and J. R. G. Patterson, Diffusion experiments with numerically integrated isotropic turbulence, *Phys. Fluids* **17**, 292–297 (1974). doi: 10.1063/1.1694714
13. L. F. Mortimer, D. O. Njobuenwu, and M. Fairweather, Near-wall dynamics of inertial particles in dilute turbulent channel flows, *Phys. Fluids* **31**, 063302 (2019). doi: 10.1063/1.5093391
14. M. Fairweather, and J. P. Hurn, Validation of an anisotropic model of turbulent flows containing dispersed solid particles applied to gas-solid jets, *Comput. Chem. Eng.* **32**(3), 590–599 (2008). doi: 10.1016/j.compchemeng.2007.04.006
15. L. Schiller, and A. Naumann, Fundamental calculations in gravitational processing, *Z. Ver. Dtsch. Ing.* **77**, 318–320 (1933).
16. E. Ulker, and M. Sorgun, Comparison of computational intelligence models for cuttings transport in horizontal and deviated wells, *J. Pet. Sci. Eng.* **146**, 832–837 (2016). doi: 10.1016/j.petrol.2016.07.022
17. A. W. Vreman, and J. G. M. Kuerten, Comparison of direct numerical simulation databases of turbulent channel flow at $Re_\tau = 180$, *Phys. Fluids* **26**, 015102 (2014). doi: 10.1063/1.4861064