Sensitivity and uncertainty analyses in deep-learning-augmented unsteady Reynolds-averaged Navier–Stokes turbulence modelling for particle-laden jet flows

X Zhang[1]*, Z Zhang[2], Z Sun[1], J Q Shi[2], G J Nathan[1] and R C Chin[1]

[1] Centre for Energy Technology, School of Electrical and Mechanical Engineering, The University of Adelaide, Adelaide, SA 5005, Australia

[2] Australian Institute for Machine Learning, The University of Adelaide, Adelaide, SA 5005, Australia

Keywords: Particle-laden flows; turbulence modelling; URANS; deep learning; prediction uncertainty.

Abstract. Building upon our previously developed deep learning (DL)-augmented framework for turbulence modelling in unsteady Reynolds-averaged Navier–Stokes (URANS) simulations of particle-laden flows, this study presents a comprehensive evaluation to enhance model efficiency and interpretability. The original framework integrated a deep neural network (DNN) into the Euler–Lagrangian gas–solid flow system to predict turbulent eddy viscosity fields based on instantaneous flow and particle features. During training, the DNN was trained using an existing high-fidelity direct numerical simulation database of four sets of mono-disperse particle-laden jet flows. In prediction, the trained DNN predicted and updated the closure term (eddy viscosity) every time step to solve the URANS equations and particle motion equations. This framework demonstrated significant improvements in predicting mean particle velocity and concentration distributions in jet flows compared to baseline URANS simulations using the standard realisable k–ϵ turbulence model. This study focuses on two aspects to refine the framework further: 1) conducting a sensitivity analysis of input features to identify their influence on the trained model, enabling potential model simplifications while retaining accuracy; and 2) evaluating prediction uncertainties resulting from variability in input features. These analyses examine how changes in critical input features propagate through the model and impact prediction fidelity, providing practical insights into the application of the framework to complex multiphase flow simulations.

Introduction

Machine learning (ML)-augmented computational fluid dynamics (CFD) models have gained significant attention in recent years due to their potential to enhance predictive accuracy while maintaining computational efficiency. Traditional Reynolds-averaged Navier–Stokes (RANS) simulations, although widely used in industry, are inherently limited by turbulence closure assumptions. High-fidelity simulations such as direct numerical simulations (DNS) and large-eddy simulations (LES) provide more accurate predictions but remain computationally expensive. Consequently, ML has been explored as a tool to enhance RANS predictions in a data-driven manner by leveraging high-fidelity data to train turbulence modelling [1-3]. However, despite these advancements, challenges persist regarding the robustness, interpretability, and reliability of ML-based CFD models.

Existing ML–CFD models suffer from generalisation issues as they are trained on specific flow conditions and exhibit low generality when applied to other cases. This is particularly evident in data-driven turbulence modelling, where models trained on high-fidelity DNS or LES data may exhibit poor performance when applied to industrial-scale simulations with different Reynolds numbers or geometric complexities [4]. The lack of interpretability further exacerbates this issue, as most ML models function as black-box approximators, making it difficult to assess the physical relevance of learned representations.

Another critical limitation is uncertainty propagation within ML–CFD models. While traditional CFD solvers incorporate well-defined numerical schemes to manage error sources, the ML–CFD frameworks introduce additional uncertainties stemming from the choice of training data, model architecture, and feature selection [5]. In particular, ML-based turbulence models may amplify small errors in the input variables, leading to cascading inaccuracies in downstream predictions [6]. Therefore, despite growing interest in ML-augmented CFD frameworks, systematic quantification and mitigation of uncertainties inherent to data-driven ML–CFD models remain underdeveloped. Existing studies have primarily focused on improving prediction accuracy, while the propagation of uncertainties across the ML–CFD workflow (from training data deficiencies to model generalisation) has received limited attention [7].

Furthermore, the sensitivity of ML models to input features remains an open research area. Many ML–CFD frameworks employ high-dimensional input spaces, including flow statistics [3], turbulence metrics [8], and geometric features [9]. However, not all features contribute equally to prediction accuracy, and redundant inputs may introduce noise that degrades model performance. Identifying the most influential features through sensitivity analysis is essential to improve model robustness, optimise computational efficiency and reduce reliance on unnecessary data sources.

In light of these research gaps, the present study is designed to evaluate our previously developed ML–CFD framework [3] through the assessment of input feature sensitivity and uncertainty propagation. An $a\ priori$ assessment was performed to quantify the impact of various combinations of key input features on training performance. Subsequently, $a\ posterior$ uncertainty analysis was carried out on the simulation predictions generated by the ML–CFD model. By systematically examining how variations in critical input features affect the simulation predictions, a deeper understanding of feature selection is gained of how the ML–CFD model responds to varying flow conditions, thereby enhancing its robustness for practical engineering applications.

Methodology

In this study, as in our previous work [3], a three-dimensional, round, turbulent co-flow jet was selected as the representative flow scenario. The numerical configurations, methods, and algorithms for the unsteady Reynolds-averaged Navier–Stokes (URANS) simulations, as well as the deep learning (DL)–URANS framework, were maintained as detailed by Zhang et al. [3]. The DL–URANS framework was designed to train a deep neural network (DNN) model using high-fidelity DNS data from Zhang et al. [10] and integrate it into the URANS simulation to predict turbulent eddy viscosity dynamically, thereby improving simulation accuracy. Building upon this established setup, the current study performed two complementary analyses: a sensitivity evaluation to the selection of input features and an uncertainty propagation assessment of the simulation predictions.

For the sensitivity study, the same training dataset as Zhang et al. [3] was employed. This dataset comprised 320 sets of instantaneous DNS data fields, encompassing both non-particle-laden and a series of mono-disperse particle-laden flows (with Stokes numbers characterising the particle-to-fluid response time ratio of $Sk = 0.3$, 1.4, and 11.2). Note that the particle-to-fluid density ratio and mass flux (i.e., mass loading) were fixed at 1000 and 0.4, respectively, for the particle-laden flows. Within this dataset, different combinations were created by systematically omitting one of the key input features during training. The $a\ priori$ training performance was quantified—using changes in the mean square error—as compared to the performance achieved when all key input features, as used by Zhang et al. [3], were included. The specific training models are summarised in Table 1. This process was intended to identify the most influential input features of the DL–URANS model and to refine feature selection.

For the uncertainty propagation assessment, the flow case examined was one of the particle-laden jets from Zhang et al. [3] with a Stokes number of $Sk = 0.3$. The $a\ posteriori$ evaluation followed the methodology described by Zhang et al. [3] by comparing the DL–URANS simulation predictions with high-fidelity DNS results reported by Zhang et al. [10]. Statistical metrics—including error variances and confidence intervals—were computed to quantify the uncertainty in the predicted turbulent eddy viscosity fields and other related flow statistics. This comparison was designed to assess how variability in the input features was reflected in the simulation outputs and to provide insights into the model's performance under varying flow conditions.

Table 1. List of cases and corresponding trained model expressions employed in the sensitivity analysis of input features. Here, the local-averaged values in each computational cell of the fluid velocity gradient $\nabla\tilde{\mathbf{u}}_f$, cell size properties Δ, fluid velocity $\tilde{\mathbf{u}}_f$, particle source term $\tilde{\mathbf{S}}_p$, fluid vorticity $\nabla\times\tilde{\mathbf{u}}_f$, pressure gradient $\nabla\tilde{p}$, pressure $\tilde{p}$, particle volume fraction $\tilde{\alpha}_p$, particle number density $\tilde{N}_p$, Kolmogorov Stokes number Sk_η and particle mass loading $\tilde{\Phi}_p$ are training input features, the local-averaged turbulent eddy viscosity $\tilde{\nu}_t$ is the training target.

Case	Trained model expression	Note
Case–0	$f(\nabla\tilde{\mathbf{u}}_f, \Delta, \tilde{\mathbf{u}}_f, \mathbf{S}_p, \nabla\times\tilde{\mathbf{u}}_f, \nabla\tilde{p}, \tilde{p}, \tilde{\alpha}_p, \tilde{N}_p, \tilde{Sk}_\eta, \tilde{\Phi}_p) \rightarrow \tilde{\nu}_t$	employed by Zhang et al. [3]
Case–$\tilde{N}_p$	$f(\nabla\tilde{\mathbf{u}}_f, \Delta, \tilde{\mathbf{u}}_f, \mathbf{S}_p, \nabla\times\tilde{\mathbf{u}}_f, \nabla\tilde{p}, \tilde{p}, \tilde{\alpha}_p, \tilde{Sk}_\eta, \tilde{\Phi}_p) \rightarrow \tilde{\nu}_t$	omitted $\tilde{N}_p$ (weight of 1.6% [3])
Case–$\nabla\times\tilde{\mathbf{u}}_f$	$f(\nabla\tilde{\mathbf{u}}_f, \Delta, \tilde{\mathbf{u}}_f, \mathbf{S}_p, \nabla\tilde{p}, \tilde{p}, \tilde{\alpha}_p, \tilde{N}_p, \tilde{Sk}_\eta, \tilde{\Phi}_p) \rightarrow \tilde{\nu}_t$	omitted $\nabla\times\tilde{\mathbf{u}}_f$ (weight of 7.1% [3])
Case–S_p	$f(\nabla\tilde{\mathbf{u}}_f, \Delta, \tilde{\mathbf{u}}_f, \nabla\times\tilde{\mathbf{u}}_f, \nabla\tilde{p}, \tilde{p}, \tilde{\alpha}_p, \tilde{N}_p, \tilde{Sk}_\eta, \tilde{\Phi}_p) \rightarrow \tilde{\nu}_t$	omitted S_p (weight of 8.4% [3])
Case–$\tilde{\mathbf{u}}_f$	$f(\nabla\tilde{\mathbf{u}}_f, \Delta, \mathbf{S}_p, \nabla\times\tilde{\mathbf{u}}_f, \nabla\tilde{p}, \tilde{p}, \tilde{\alpha}_p, \tilde{N}_p, \tilde{Sk}_\eta, \tilde{\Phi}_p) \rightarrow \tilde{\nu}_t$	omitted $\tilde{\mathbf{u}}_f$ (weight of 9.9% [3])
Case–$\nabla\tilde{\mathbf{u}}_f$	$f(\Delta, \tilde{\mathbf{u}}_f, \mathbf{S}_p, \nabla\times\tilde{\mathbf{u}}_f, \nabla\tilde{p}, \tilde{p}, \tilde{\alpha}_p, \tilde{N}_p, \tilde{Sk}_\eta, \tilde{\Phi}_p) \rightarrow \tilde{\nu}_t$	omitted $\nabla\tilde{\mathbf{u}}_f$ (weight of 32.9% [3])

Results

Figure 1 presents the evolution of the mean-square-error (MSE) to represent the model training performance during the training process of the DNN model under different cases in Table 1. It can be seen that Case–$\tilde{N}_p$, which omitted the input feature of particle number density in each computational cell $\tilde{N}_p$ from the training data set, exhibits a very close MSE profile to the reference Case–0 (the training case employed by Zhang et al. [3]). This can be explained by the low weight of input feature $\tilde{N}_p$, which accounts for 1.6% during training. Such a low weight is associated with an insufficient representation of the particle phase in the training dataset. As noted above, although particle-laden flows with various particle Stokes numbers were used, the density ratio and mass loading remained constant. Moreover, the training target $\tilde{\nu}_t$ is not directly related to $\tilde{N}_p$; they are only connected through the particle source term S_p in the Navier-Stokes equation (see Equation 2 in the paper by Zhang et al. [3]). As the weight of the omitted input feature increases sequentially (see Table 1), the MSE profiles deviate progressively further

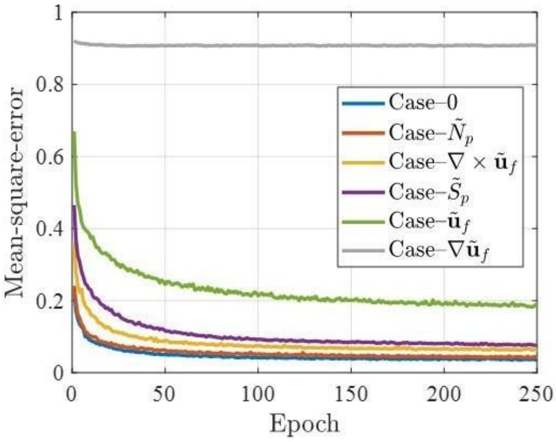

Figure 1. Sensitivity study of the mean square error (MSE) evolution in the training process for cases omitting various input features (see Table 1).

from that of Case–0. A notable transition is observed in Case–$\tilde{\mathbf{u}}_f$, where the fluid velocity was omitted, its MSE value remains relatively high (approximately 0.2) after 250 epochs and continues to decrease, suggesting that the training process may have been over-fitting. Although the omitted fluid velocity $\tilde{\mathbf{u}}_f$ represents 9.9% of the total weight, which is slightly higher than the 8.4% weight of the omitted particle source term S_p in Case–S_p, its impact on training performance increases non-linearly. Furthermore, in Case–$\nabla\mathbf{u}_f$, which omitted the fluid velocity gradient with a weight of 32.9%, the MSE value remained around 0.9 without any reduction, indicating that the training failed.

Figure 2 presents the instantaneous predictions of the turbulent eddy viscosity, fluid, and particle velocity fields for different cases as an *a posterior* evaluation of the DL–URANS model. Qualitatively, it is clear that the discrepancy between the predictions of the DL–URANS case from left to right and the DNS results reported in Zhang et al. [10] becomes increasingly apparent. In particular, the predicted turbulent eddy viscosity values tend to increase, while the decay and spread of both fluid and particle velocities become progressively less pronounced.

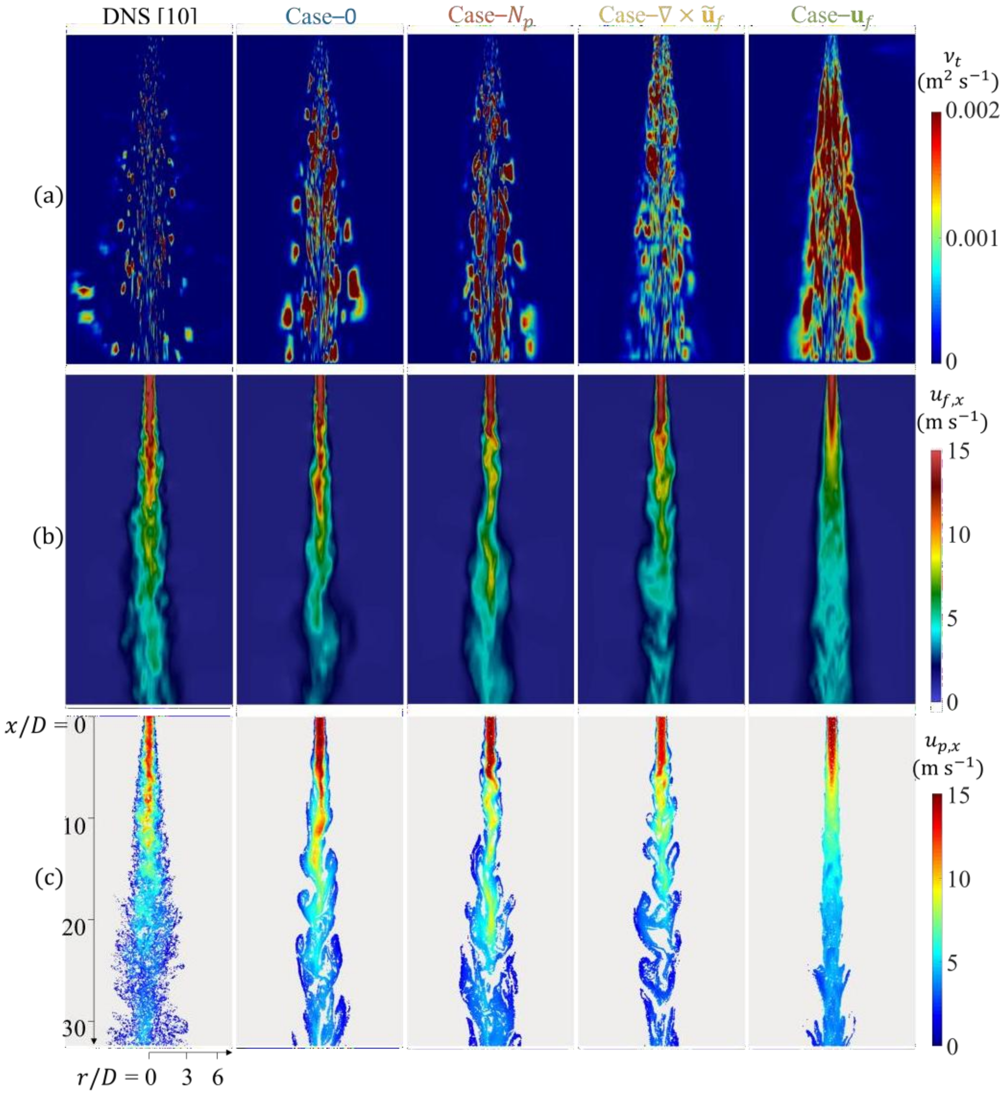

Figure 2. Representative instantaneous images of (a) the turbulent eddy viscosity field, (b) the fluid velocity field, and (c) the particle velocity field in the downstream jet domain for different DL–URANS cases (see Table 1) and the DNS [10].

Table 2 summarises the mean relative errors of the quantitative statistics compared to the DNS results [10], which include the mean centerline evolution of the axial velocity and the half-width of the mean axial velocity for both the fluid and particle phases. The error E was computed following the methodology outlined in Zhang et al. [3]. Combining the information from this table and Figure 1, it can be found that the error E increases with the rise in the MSE during training. In other words, worse performance in the $a\ priori$ training process is monotonically associated with more significant discrepancies in the $a\ posterior$ simulation predictions for both fluid and particle phases.

Table 2. Averaged relative error E of the DL–URANS predictions for the fluid or particle statistics from different cases (see Table 1) with respect to the previous DNS data [10].

Fluid or particle statistic	Case–0	Case–$\tilde{N}_p$	Case–$\nabla\times\tilde{\mathbf{u}}_f$	Case–$\tilde{\mathbf{u}}_f$
Centreline evolution of mean axial fluid velocity $U_{f,x-c-e}$	2.1%	3.5%	17.3%	43.3%
Half-width of mean axial fluid velocity $R_{0.5Uf,x-c}$	8.9%	8.1%	12.2%	24.8%
Centreline evolution of mean axial particle velocity $U_{p,x-c-e}$	2.9%	4.5%	14.2%	67.1%
Half-width of mean axial particle velocity $R_{0.5Up,x-c}$	4.8%	5.2%	16.6%	55.4%

Figure 3 compares the radial distribution of turbulent eddy viscosity ν_t in the jet downstream location at $x/D = 20$ between the DL–URANS cases and the DNS training dataset [10]. Figure 3(a) shows that the mean ν_t profile predicted by Case–0 is comparable to the DNS data, peaking at $r/D \approx 0.4$ before gradually decreasing. However, a secondary peak at $r/D \approx 2.5$ is observed in Case–0 but not in the DNS, which is also evident in the joint probability distribution function (PDF) of ν_t values and radial positions in Figure 3(c). This discrepancy may result from large data variability at the jet boundary, particularly in regions with steep velocity gradients, which the training model did not fully capture. Further investigation is needed to confirm this hypothesis. Apart from this, the PDF prediction of ν_t shows the same trend with the DNS, with values concentrated at $\nu_t \leq 0.001$ and smaller r/D, explaining the overall agreement in the mean ν_t profile. In contrast, Case–$\tilde{\mathbf{u}}_f$ exhibits significant deviations from the DNS, directly corresponding to its poor training performance (see Figure 1, where its MSE remains at 0.2).

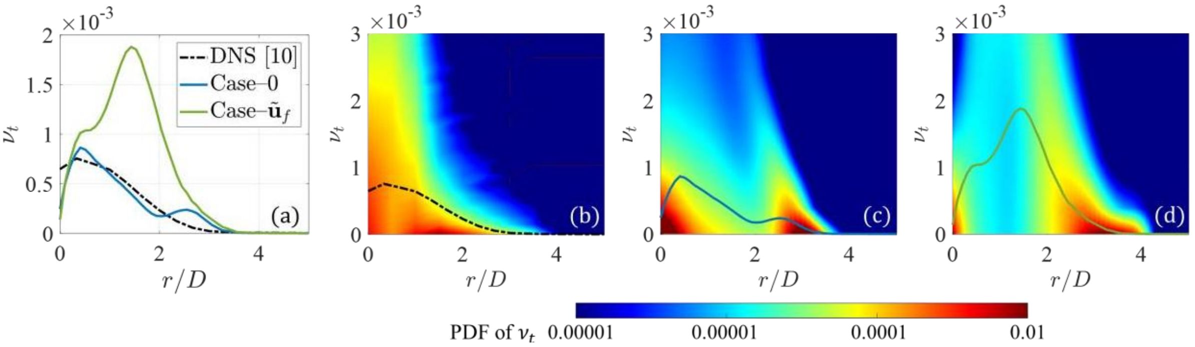

Figure 3. Radial distributions of the turbulent eddy viscosity ν_t in the jet downstream domain at $x/D = 20$, comparing (a) mean ν_t profiles for different DL–URANS cases and the DNS training dataset, as well as the joint probability distribution function (PDF) of ν_t values and their radial positions in (b) the previous DNS [10], (c) Case–0, and (d) Case–$\tilde{\mathbf{u}}_f$.

Conclusion

This study provides insight into the nonlinear influence of input feature weighting on training performance. While it is confirmed that lower-weighted particle-related features (e.g., particle number density and source term) have a weaker impact compared to higher-weighted fluid-related features (e.g., fluid velocity gradients), our findings reveal an intricate, nonlinear relationship that requires further quantitative investigation.

The consistently low weighting of particle-related features suggests an insufficient representation of the particle phase in the training dataset. Enhancing model performance may require incorporating a broader range of particle-laden flows into the training dataset (e.g., with varied mass loadings and density ratios) and/or treating particle-related parameters (e.g., particle source term) as an additional training target.

Furthermore, it is worth noting that the $a\ posterior$ evaluation shows that turbulent eddy viscosity is less accurately predicted near the jet boundary with high velocity gradients. This prediction bias is not evident from the $a\ priori$ training performance, despite the common assumption that a reduction in $a\ priori$ training performance directly translates into increased $a\ posteriori$ simulation errors. This highlights the need for further refinement to enhance model generalisation in high-shear regions, possibly by incorporating localised flow characteristics into the training process or employing more adaptive learning strategies.

Acknowledgements

Financial support for the project has been provided by the University of Adelaide Grant (DIGI+FAME). The authors acknowledge the Phoenix High-Performance Computing Centre at the University of Adelaide for resource and technical support. Xinchen Zhang would also like to acknowledge the Contribution to the Australian Scholarship and Science (CASS) Foundation for the travel award that supported conference attendance.

References

[1] K. Duraisamy, G. Iaccarino, and H. Xiao, "Turbulence modeling in the age of data," Annu. Rev. Fluid Mech. **51**, 357–377, 2019.

[2] B. Wang, and J. Wang, "Application of artificial intelligence in computational fluid dynamics," Ind. Eng. Chem. Res. **60**(7), 2772–2790, 2021.

[3] X. Zhang, Z. Zhang, A. Chinnici, Z. Sun, J.Q. Shi, G.J. Nathan, and R.C. Chin, "Physics-informed data-driven unsteady reynolds-averaged navier–stokes turbulence modeling for particle-laden jet flows," Phys. Fluids **36**(5), 2024.

[4] S.L. Brunton, B.R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," Annu. Rev. Fluid Mech. **52**(1), 477–508,2020.

[5] Y. Bae, K. Nam, and S. Kang, "CFD-ML: Stream-based active learning of computational fluid dynamics simulations for efficient product design," Comput. Ind. **161**, 104122, 2024.

[6] Y. Zhang, D. Zhang, and H. Jiang, "Review of challenges and opportunities in turbulence modeling: A comparative analysis of data-driven machine learning approaches," J. Mar. Sci. Eng. **11**(7), 1440, 2023.

[7] S. Cheng, C. Quilodrán-Casas, S. Ouala, A. Farchi, C. Liu, P. Tandeo, R. Fablet, D. Lucor, B. Iooss, J. Brajard, and others, "Machine learning with data assimilation and uncertainty quantification for dynamical systems: A review," IEEE/CAA J. Autom. Sin. **10**(6), 1361–1387, 2023.

[8] J. Wu, J. Wang, H. Xiao, and J. Ling, "A priori assessment of prediction confidence for data-driven turbulence modeling," Flow, Flow Turbul. Combust. **99**, 25–46, 2017.

[9] A. Jafarizadeh, M. Ahmadzadeh, S. Mahmoudzadeh, and M. Panjepour, "A new approach for predicting the pressure drop in various types of metal foams using a combination of CFD and machine learning regression models," Transp. Porous Media. **147**(1), 59–91, 2023.

[10] X. Zhang, Z.F. Tian, A. Chinnici, H. Zhou, G.J. Nathan, and R.C. Chin, "Particle dispersion model for RANS simulations of particle-laden jet flows, incorporating Stokes number effects," Adv. Powder Technol. **35**(3), 104345, 2024.