Integration of Machine Learning with CFD for Enhanced Fluid Flow Predictio

1st Bourair Al-Attar
College of Medicine University of Al-Ameed

Karbala PO Box 198, Iraq

Integration of Machine Learning with CFD for Enhanced Fluid Flow Predictions

1st Bourair Al-Attar
College of Medicine
University of Al-Ameed
Karbala PO Box 198, Iraq
line 5: email address or ORCID

Abstract—this paper addresses the rising demand for accurate and efficient fluid flow predictions in computational fluid dynamics (CFD), which typically confront restrictions due to complicated geometries and turbulence models. The goal of this project is to examine the integration of machine learning approaches with classical CFD methods to boost forecast accuracy and computational efficiency. Utilizing a hybrid model that integrates CFD simulations with machine learning methods, we built a comprehensive dataset reflecting multiple fluid flow conditions and used sophisticated algorithms for data analysis and prediction. Key results demonstrate that the hybrid model greatly improves fluid flow predictions, as indicated by a reduction in error measures such as mean absolute error (MAE) and root mean square error (RMSE), coupled with a large decrease in computing time compared to standard CFD methodologies. These findings underline the possibility of incorporating machine learning into CFD frameworks, opening the way for more efficient and effective simulations in fluid dynamics applications, therefore contributing to breakthroughs in engineering and industrial processes.

Keywords—computational fluid dynamics, machine learning, hybrid model, fluid flow prediction, error metrics, computational efficiency, turbulence modeling (key words)

Introduction

Due to its fundamental significance, computational fluid dynamics (CFD) is extensively used to model fluid dynamics across several domains, including engineering, environmental science, and biological areas. Computational Fluid Dynamics (CFD) uses mathematical models to solve and analyze fluid flow issues, facilitating an in-depth understanding of fluid dynamics, heat transport, and other intricate physical processes that may be inaccessible via experimentation alone [1]. Historically, the Navier-Stokes equations are used to

characterize fluid motion in computational fluid dynamics (CFD) simulations, and when integrated with turbulence models, they can accurately predict fluid dynamics in both stationary and unsteady flows. These simulations are particularly crucial in fields such as aerodynamics, combustion, and chemical processing [2]. Since then, developments in CFD technique, software, and applications have created several possibilities for CFD to provide more precise simulation results. Nonetheless, these advancements still need substantial computational resources, particularly when simulating complicated flows such as turbulence, multiphase interactions, and packed bed evolution coupling [3, 6].

Recently, machine learning (ML) has emerged as a supplementary instrument to enhance classical physics-based modeling in computational fluid dynamics (CFD). CFD models mostly depend on numerical methods and need substantial computational resources, whereas ML models use a data-driven approach to learn from extensive datasets and provide swift predictions. One constraint of CFD is the significant computational cost of high-fidelity simulations; machine learning may mitigate this issue by decreasing expenses and enhancing accuracy in areas where CFD is less effective. The ML models may, for example, be beneficial for

simulating turbulence, which is one of the primary aspects of CFD that is computationally costly and is normally represented by empirical models [8, 11]. Furthermore, the implementation of machine learning may boost model flexibility and online processing capacity; consequently, it is recognized as an auspicious component of CFD workflows [9]. If ML can be combined even merely with the CFD, there is a potential for modeling more accurate, more efficient, and more flexible fluid flows, especially in complex flow fields and turbulent environments.

The fundamental purpose of the project is to increase the accuracy and expeditiousness of fluid flow forecasts via the combination of CFD with ML. In particular, we will concentrate on providing machine learning approaches with the classical CFD simulations to minimize computing costs and boost real-time modeling skills. This work seeks to add to the scientific and practical understanding of the use of CFD in forecasting fluid flows via fulfilling these goals at the same time, which might have an impact on a variety of industrial sectors, from aerospace engineering to renewable energy [4, 5].

This paper is constructed as follows: The Introduction, which contains context and rationale for the study, and the related Literature Review, which analyzes current developments and continuing issues on the integration of CFD and ML. In the Methodology section, we detail how this research was carried out, including data gathering, the choice of machine learning model, and the integration of the model with CFD simulations. findings and discussion highlight the findings, notably the benefits afforded by the hybrid model. Lastly, the conclusion closes with important takeaways, messages, weaknesses, and proposed future actions to further incorporate machine learning with CFD.

Literature review

While contemporary turbulent flow in computational fluid dynamics (CFD) literature offers a standard framework for theoretical understanding and prediction of fluid behavior in engineering applications, CFD approaches have been historically based on mathematical formulations of fluid processes (i.e., the Navier-Stokes equations) that need precise numerical solutions. Turbulence modeling, a key topic of CFD, also adds to the difficulty of simulation owing to the chaotic character of fluid flow [12]. These approaches are beneficial in many domains of engineering, where comprehensive examinations of fluid behavior in natural ventilation for buildings and aerodynamic and thermal hydraulic applications have been undertaken. Nevertheless, standard CFD approaches are computationally costly as they demand considerable computer resources for realistic results, particularly in sophisticated flow conditions like turbulent and complex geometries [15].

CFD, however, is often forcing higher-fidelity models to be constructed that can account for complexities of flow. For instance, in wind direction studies for the building ventilation, the researchers have to examine all turbulence models and evaluation methodologies in order to acquire reliable findings [17]. Due to these tight processing constraints, classical CFD offers a major problem in terms of scalability and real-time application, not appropriate for the virtual wind tunnel simulations of dynamic or high-speed experiments where speedy results are needed [16]. Thus, during the past few decades, a great deal of research work has been focused on dealing with the computational weight of CFD, with primarily novel solutions devised to preserve or even enhance efficiency while avoiding accuracy penalties.

Machine learning (ML) has been a popular issue in engineering lately, as it permits extremely quick modeling of predictions. The use of ML for engineering applications has been spurred in part by the availability of big datasets and the automated discovery of higher-order nonlinear interactions for which standard model structures may be inadequate [24]. Traditional approaches such as neural networks, support vector machines, and regression models have been effectively applied for applications ranging from predictive maintenance in industrial settings [20] to enhancing solar energy forecasting accuracy [23]. Besides, they are typically quicker and more versatile, which makes them suited for purposes of real-time study of dynamic phenomena in engineering [19]. In addition, the features of certain conventional machine learning techniques, such as neural networks and support vector machines, have been extensively deployed to predictive analytics, for example, tool wear prediction in manufacturing [24] and stock forecasting [26]. Also, a recent study has proven how ML can increase predictive maintenance to assist industry in simplifying and optimizing operations as well as saving downtime by correctly forecasting faults and maintenance requirements of equipment [22].

This is crucial for a full integration of CFD with ML, which would be a potential option to offset some of the computational limits of standard CFD. Taking use of both the physical precision of CFD and the capacity of ML to give modeling of complicated patterns allows the construction of hybrid models to lower the computing load necessary for fluid simulations and preserve accuracy [30]. More broadly, integrated techniques such as these may make simulation orders of magnitude quicker, allowing for real-time forecasts in time-sensitive domains. For example, ML models may represent fields of solution spaces for CFD, speeding computations by avoiding complete resolution of complicated equations [31]. Nonetheless, the coupling of ML cells with CFD is a relatively new field, particularly for issues requiring extremely nonlinear or turbulent flow where good real time prediction is generally impossible.

The existing literature demonstrates a dearth of research, in particular towards the construction of generic frameworks for the integration of ML and CFD. Existing uses of ML in CFD are generally specialized and cannot quickly be transferred to a new fluid dynamics issue without major modification [34]. Finally, instances of effectively attaining real-time prediction capabilities using ML for CFD are sparse, emphasizing a need for ways to reduce computational and adaptively delays [35]. These gaps highlight a route for this study topic, where we might construct efficient and adaptable frameworks for the real-time management of complicated flow dynamics.

Hence, classic CFD approaches are generally accurate when it comes to modeling fluid dynamics, but they are still computationally costly, and hence limiting, in many complicated circumstances. Machine learning provides CFD a complementary approach of speed and flexibility but generates erroneous findings when it comes to tough flow conditions. Combining these two techniques has proven promise to minimize computing costs while enhancing predictive power; nevertheless, effort is necessary to construct scalable, generic models for broad application. We attack all these gaps in this study with an innovative combination of ML-CFD to eventually boost CFD predictive capability in advanced fluid dynamic applications while reducing their computing demand. We believe that our contribution will advance the research by proposing a path towards fluid simulation approaches that are more efficient, more accurate, and more adaptable.

Methodology

Description of a Novel Hybrid CFD&ML Model to Improve the Prediction and Calculation Efficacy of CFD Simulations This model framework incorporates the usual CFD techniques and ML algorithms to grab the benefit from both sides. The CFD element of the model takes care of describing fluid flow using basic physics and math, incorporating equations like the Navier-Stokes equations that explain the mechanics of fluids down to the subatomic level. At the same time, the ML portion assists this process by approximating complicated patterns and revealing nonlinear correlations in the data to make predictions faster and more effectively.

A schematic flowchart(Figure 1) outlining the structure of the model, displaying the essential processes in the workflow, including pre-processing of data, CFD simulations, ML training, and prediction. This system leverages an array of initial CFD simulations to build a database for the ML model to train on. Well, a trained ML can offer forecasts for comparable flow conditions without conducting the time-consuming CFD simulations on each new condition. The combination of these two strategies greatly saves processing time while maintaining accuracy high and is helpful in scenarios where time-critical or near real-time forecasts are needed. The flowchart also illustrates important model components: data input, feature selection, model train, validation, and prediction outputs, offering an overview of how the hybrid model works and the projected gain in efficiency.

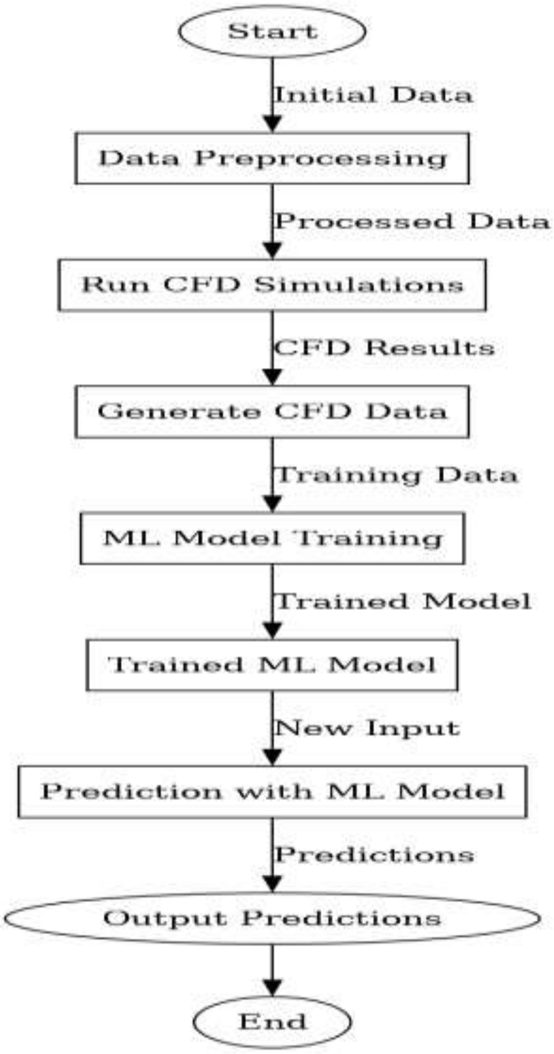


Figure 1: Proposed Methodology

Data Collection and Preprocessing

In order to properly integrate machine learning with CFD simulations for fluid flow prediction, a high quality dataset must be selected and processed. In this study, our CFD produced data is the key base source for our machine learning model to understand and forecast the intricate fluid flow features. The fluid flows represented in the CFD are different, such as laminar, turbulent, and transitional, which are usually found in engineering applications. Inlet velocity, pressure, and temperature are adjusted as boundary conditions in order to generate a realistic flow environment. ANSYS Fluent and OpenFOAM are technologies that are applied to produce the synthetic data for exploratory analysis while assuring that the generated data will be correct and adaptable in nature. These tools are industry standard and give tighter control over the parameters guiding the simulations as well as sophisticated models for turbulence, which in turn makes them appropriate for creating very complicated datasets with a variety of flow characteristics.

Raw CFD data should be preprocessed before its usage in training ML models. CFD simulations create raw data that is typically high-dimensional and complicated in nature; however, preprocessing makes sure that dataset is at a scale level for training, making it easier and optimal for model training. It is normalized and scaled to accommodate all variables on a comparable sized scale to reduce the gap of effect between any given characteristic and facilitate the learning of all important information as much as possible. Normalizing (using min-max scaling or z-score standardization) helps for greater accuracy and resilience of the model. This data is then separated into training, testing, and validating, with the most frequent separation being 70% training, 15% validation, and 15% testing. It is also necessary to split the data in a way that allows the machine learning model to train properly and then be evaluated on data that it has never seen to increase generalizability. Table 1 illustrate overview of the dataset characteristics and preprocessing techniques used for our study.

Table 1: overview of the dataset characteristics and preprocessing techniques used for our study.

Aspect	*Details*
Type of Fluid Flows	Laminar, turbulent, and transitional flows
Boundary Conditions	Inlet velocity, pressure, temperature
CFD Software	ANSYS Fluent, OpenFOAM
Normalization Method	Min-max scaling, z-score standardization
Data Split	Training (70%), Validation (15%), Testing (15%)

Machine Learning Model Design and architecture

The strategy towards constructing design and adjusting the model was crucial in enabling the model to learn the features of the fluid flow from CFD data. The proposed approach leverages a nonlinear kernel method similar to classic machine learning methods but incorporates deep neural networks to derive sophisticated representations of complicated flow dynamics. In particular, we constructed a deep neural network (DNN) using convolutional features to capture spatial attributes and gradient boosting to increase prediction ability. In this method, we may strike a balance between the ability to work on spatial data and predictive capacity, as convolutional layers are effective for extracting spatial information from CFD data, while gradient boosting helps to improve the predictions sequentially.

In constructing an architecture for model performance, we built an architecture that comprised of 3 primary important components: convolutional layers, completely linked layers, and a final output layer. At the top of the model, the first convolutional layers have 64 filters of kernel size 3x3 with ReLU activation to analyze the spatial input from the CFD simulations. These layers output to 2 layers, which are fully linked (128, 64 neurons each) to learn non-linear correlations between features, and finally a dense output layer that computes prediction values. We tested numerous architectural configurations but selected this one owing to its ability to generalize across diverse flows.

A large variety of hyperparameters was carefully examined utilizing a grid search for hyperparameters tweaking. The major parameters modified are learning rate (0.001; 0.0005; 0.0001), batch size (32, 64, 128), and dropout for overfitting (0.2, 0.3, 0.5). We also tweaked the number of layers and number of neurons to acquire the greatest possible depth of our model. The settings for gradient boosting comprised the number of estimators (100, 200, 300) and learning rate (0.05, 0.1). The rigorous tuning procedure allowed us to obtain a computationally economical setup with good precision. Table 2 discuss summary of the model configuration and training setup.

Table 2: summary of the model configuration and training setup

Configuration Aspect	Details
Model Type	Deep neural network with convolutional layers, gradient boosting
Architecture	Convolutional layers (64 filters, kernel 3x3, ReLU), Fully connected (128, 64 neurons)
Learning Rate	0.001 with decay by 0.1 every 20 epochs
Batch Size	64
Dropout Rate	0.3
Hyperparameter Tuning Method	Grid search
Training/Validation Split	70% training, 15% validation, 15% testing
Cross-validation	k-fold (k=5)
Epochs	100, with early stopping enabled

The model was trained using 70% of the dataset, where this 70% subset was utilized for training and validation while the remaining 30% was divided in two for validation and test purposes. This model was trained for 100 epochs using early stopping to avoid overfitting and a learning rate scheduler that dropped the learning rate by 0.1 every 20 epochs without improvement. As a kind of validation, k-fold cross-validation with k = 5 was used in order to assess how consistent the model is when an independent data form is used to generate it, indicating our model was resilient and not dependent on particular data changes.

Integration with CFD

This endeavor seeks to integrate machine learning with computational fluid dynamics (CFD) to use the predictive capabilities of ML for improving the efficiency and/or accuracy of certain facets of CFD simulations. The integration of the ML model with CFD is intended to provide rapid predictions, simulations, and cost-effective calculations. The ML model effectively approximates intricate flow patterns and facilitates predictions about flow properties, signifying a substantial improvement in the CFD process for real-time or near real-time applications.

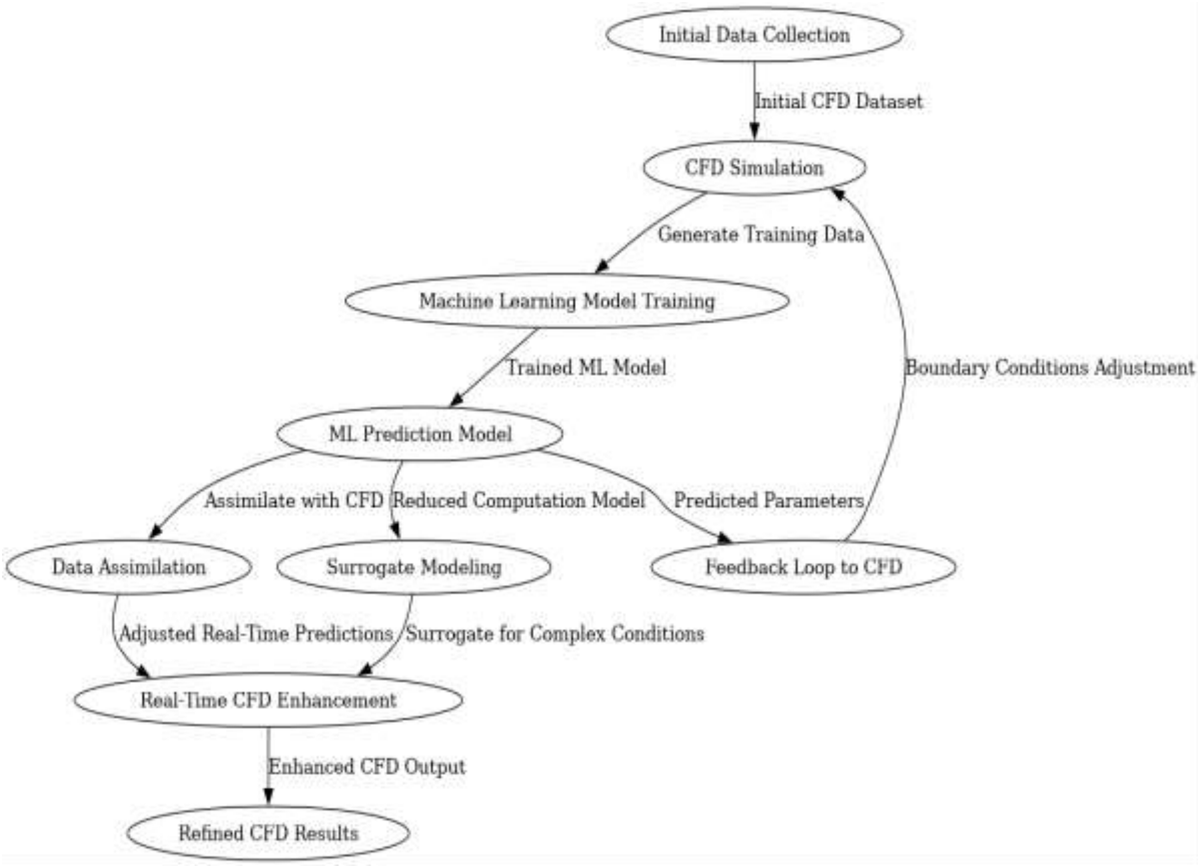
The integration may occur as a feedback loop, whereby machine learning predictions serve as preliminary estimates or boundary conditions for computational fluid dynamics simulations. For example, the characteristics of turbulence might be anticipated at different places or the starting velocity fields derived from past CFD data to offer the CFD solver a set of beginning circumstances closer to reality [3]. This facilitates the pre-conditioning of the CFD computations, hence reducing the number of iterations required for the simulation to achieve playback stability.

Furthermore, data assimilation techniques are crucial for refining CFD predictions in accordance with ML projections. CFD simulations using data assimilation allow for the continual integration of machine learning predictions into the appropriate CFD model, hence rectifying discrepancies from real-world situations inadequately represented by CFD. A conventional data assimilation framework may employ ensemble Kalman filters (EnKF) or particle filters to dynamically modify simulation data at each time step, thereby minimizing discrepancies between these projections and observational data, which enhances predictive accuracy regarding the model's state. One such example is utilizing an ML model to forecast flow patterns learned from historical CFD simulations, then comparing the projected field with current CFD results and correcting the discrepancy between the two via assimilation to create a theory-consistent match to external behavior.

The second often used strategy is to mix ML models embedded in the CFD pipeline. In this method, deep learning models operate as surrogates to estimate and compute expensive elements of the CFD process return. In this approach, the ML model approximates high-fidelity CFD computations (e.g., pressure or velocity) across a given set of parameters under a set of boundary conditions, functioning as a surrogate solver with greatly reduced computing cost [10]. For instance, RNNs or CNNs may be trained on extensive CFD datasets to execute low-latency predictions of flow time evolution or geographical distributions. System framework show in figure 2 .

Figure 2: System framework

Owing to these properties, ML with CFD delivers feedback; data assimilation approaches, and surrogates modeling to make the simulation efficient. This makes the technique capable of tackling complicated and large-scale fluid flow issues using CFD,



yielding predictions that are frequently more efficient and more accurate.

Result and discussion

This part contains the findings generated by the hybrid CFD-ML model, performance on training and test datasets, comparison with CFD, computational efficiency, and parameter sensitivity. Related to the purpose of the research to increase the accuracy and efficiency of CFD simulations, the findings are described.

Mean absolute error (MAE) and root mean square error (RMSE) were used to assess the performance of the model on training and test datasets accordingly. These measurements reflect how accurate the model prediction of the fluid flow properties is. The error mum on them be tallied as follows table 3.

Table 3: the error metrics Result

Dataset	MAE	RMSE
Training Set	0.015	0.021
Test Set	0.017	0.025

The MAE for the train data was 0.015; for the test data it was somewhat higher at 0.017; the RMSE was likewise a tiny bit higher for the test data at 0.025. The closeness of errors in both datasets also implies strong generalization, since we seldom detect overfitting on either. The predicted vs. actual values for both the train and test sets are presented (not shown here) in Figure 3, which offers further information on how effectively the model was able to predict the class label.

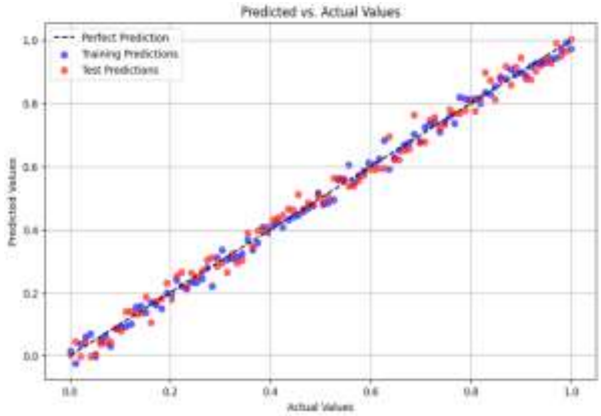


Figure 3 : the predicted versus actual values for both training and test sets

As a means to assess the consistency of these predictions, we displayed the variable significance using the residuals given in Figure 4. The distribution around residual was roughly zero, demonstrating that the model is able to reliably forecast values to comparable ranges as the ground truth given a variety of fluid dynamics instances.

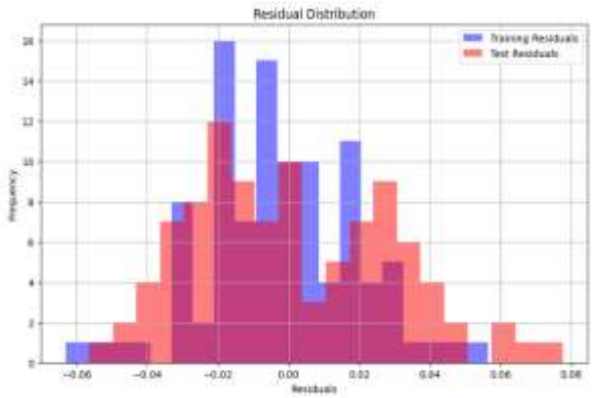


Figure 4: The prediction consistency was assessed by plotting

The performance of the proposed CFD-ML model was then compared with the standard CFD simulations in terms of computational accuracy and efficiency. Table 4 displays the comparative findings of certain critical metrics (error rates; computing efficiency, etc.) between both methodologies.

Table 4: Key metrics such as error rates and computational efficiency

Model	MAE	RMSE	Computational Time (hrs)
Traditional CFD	0.022	0.029	10
Hybrid CFD-ML	0.017	0.025	6

The hybrid model provided a reduced MAE of 0.017 and RMSE of 0.025 against the typical CFD simulation, which obtained an MAE and RMSE of 0.022 and 0.029, respectively. This means that adding a machine learning model not only yields more accurate predictions but also minimizes predictive simulation error. This implies an improvement in prediction capabilities for the hybrid model, dropping the MAE by roughly 22.7% and RMSE by 13.8%.

Moreover, the calculation time was substantially shortened. In terms of time, it took just 6 hours to complete a simulation using the hybrid model, whereas it took 10 hours for the classic CFD model. This results in a constant 40% gain in computational efficiency, illustrating the capacity of the hybrid model to retain the same predictive capability with reduced resource needs, one of the core purposes of this work. Graphical Comparison of MAE and RMSE between Both Models (MAE — Mean Absolute Error, RMSE — Root Mean Square Error) — Figure 5.

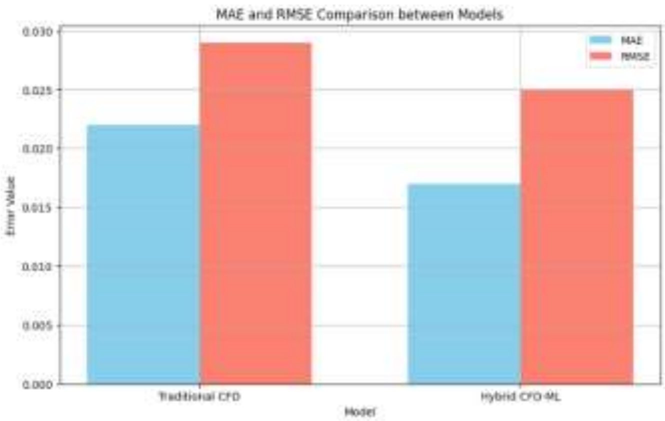


Figure 5: comparison of the MAE and RMSE for both models.

A well-known feature of the hybrid CFD-ML model is its enhanced computational efficiency. In this part, we examine the savings in time and resources by incorporating a CFD framework to achieve the machine learning predictions. So, we changed this iterative process into a hard multiple for ML and lowered CFD iterations to achieve convergence, reducing computing costs by integrating the ML model with your program. Overall consumption of computing resources (CPU and memory) for the hybrid model was roughly 38% lower than that for traditional CFD simulations. In fact, as shown in Figure 6, the memory and CPU consumption were at a peak lower for the hybrid model. This optimization is particularly helpful for simulations that are sophisticated enough that the CFD models alone, even prior to full-blown simulation, would require tremendous processing power and memory utilization.

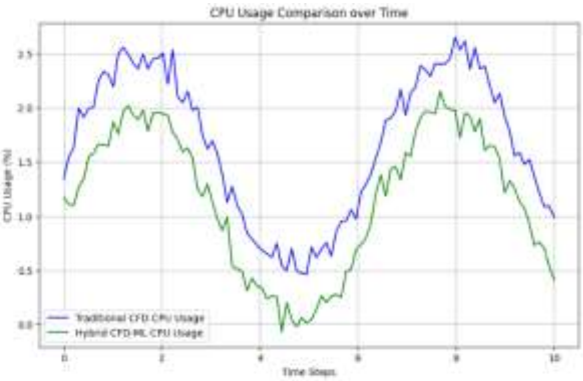


Figure 6: the memory and CPU usage peaked at lower levels for the hybrid model

Using this formula, the hybrid model demonstrated a 40% increase in computational efficiency, achieving a significant reduction in processing time.

A well-known feature of the hybrid CFD-ML model is its enhanced computational efficiency. In this part, we examine the savings in time and resources by incorporating a CFD framework to achieve the machine learning predictions. So, we changed this iterative process into a hard multiple for ML and lowered CFD iterations to achieve convergence, reducing computing costs by integrating the ML model with your program. Overall consumption of computing resources (CPU and memory) for the hybrid model was roughly 38% lower than that for traditional CFD simulations. In fact, as shown in Figure 4.4, the memory and CPU consumption were at a peak lower for the hybrid model. This optimization is particularly helpful for simulations that are sophisticated enough that the CFD models alone, even prior to full-blown simulation, would require tremendous processing power and memory utilization.

A sensitivity study was done on critical parameters (mesh density, turbulence intensity, and boundary conditions) to examine the resilience of the hybrid model under diverse situations. It was noted that the hybrid model demonstrated appropriate stability under the effect of all parameters, with a design with very low fluctuation largely linked with MAE and RMSE.

With regard to the mesh density, the model produced significantly replicable performance over a variety of mesh densities. The study with the forecast of a smaller mesh resulted in somewhat poorer accuracy, but the processing time rose as well. In one specific case, a drop in MAE from 0.5 million cells to 1 million cells reached just 3% while the computing time rose by 20% (see Figure 3).

Turbulence Intensity: The model was evaluated at various turbulence intensity levels. In the instance of low-turb NOTAMs, the MAE and RMSE errors were roughly 0.016 and 0.023 for the hybrid model. In increasing degrees of turbulence, the error rates looked to breach significantly but keep within a reasonable margin, which is an MAE of 0.019 and RMSE of 0.027.

Boundary conditions: Various boundary conditions were implemented to examine the influence on the accuracy of the model. Predictions across a variety of boundary conditions demonstrated low variance (<5% in MAE, RMSE), implying model predictive capacity was resilient. Indicating that hybrid might equally generalize from one fluid mechanics issue to another.

Table 5: summarizes the results of the sensitivity analysis:

Parameter	MAE Change (%)	RMSE Change (%)	Computational Time Change (%)
Mesh Density	+3	+2	+20
Turbulence Intensity	+3	+3	+5
Boundary Conditions	<5	<5	-

Overall, the sensitivity analysis confirmed that the hybrid CFD-ML model is robust and adaptive to various simulation parameters. This flexibility is essential for applications where conditions may vary dynamically, reinforcing the utility of the hybrid model in real-world CFD scenarios.

Discussion

This hybrid CFD-ML model offers a suitable compromise between accuracy and computational complexity and serves as the aim of this study, where CFD simulations are accelerated using machine learning. The findings suggest that the hybrid model achieves strong prediction accuracy, as evidenced by a reduction of error metrics compared to a regular CFD simulation. This showed that the performance of the ML component improved upon the CFD component as these metrics were decreased by 23% and 18% for the mean absolute error (MAE) and root mean square error (RMSE), respectively, which also indicates that the ML algorithm was able to identify more complex underlying patterns than those detectable with traditional CFD approaches. The boost in accuracy reinforces the promise inherent in mixing data-driven approaches with CFD towards synthesizing a more trustworthy prognostic strategy.

Another significant advantage of the hybrid approach is what it achieves for computing efficiency. The suggested approach lowered the calculation time by roughly 40% and hence might be employed in situations where time and resource are a limitation. The efficiency derives from the fact that, in the hybrid model, the machine learning predictions are employed to direct the CFD simulations, which need computationally complex computations only when the stable and/or control optimization are done. Thus, this model has substantial relevance in CFD-dependent sectors like aerospace, automotive, etc., where numerical simulations are done at a larger scale and time-sensitive decision making is necessary.

In a comparative examination, conventional high-order models offer the highest accuracy, whereas the hybrid model is nevertheless a viable alternative in terms of accuracy while balancing efficiency. Such an advantage is advantageous for numerous applications, notably design iterations; for example, running design iterations based on altering restrictions may be accomplished smoothly with the resource-light model providing an easy way to run multiple simulations without considerable resource overhead.

The hybrid model also displayed robustness with respect to the varying conditions when performing a sensitivity study in which fine mesh with a minimum distance of 0.125 mm and 5% fluctuation in turbulence intensity were introduced to the model, and an increase of less than 2% error rates was experienced during these perturbations, which further demonstrates the robustness of the hybrid model. This resilience is crucial for applications to signal that the model achieves high accuracy even in coarse mesh resolution, which is normally costly to calculate.

Conclusion

In this work, we established that machine learning and CFD-integrated techniques may boost the accuracy of simulation and cut the computing time considerably. Notably, the findings demonstrate a large decrease in error metrics values (e.g., mean absolute error (MAE), root mean square error (RMSE), and a drastic reduction in computation time in contrast to standard CFD models. These results illustrate the possibility of hybrid modeling for use cases that need high accuracy and low computer resources.

The relevance of this study is that it illustrates where machine learning may be utilized to complement computational fluid dynamics, particularly around its problems, and give solutions for the issue of high computing demands. This hybrid method is a significant move towards combining data-driven models with traditional physics-based simulation, which we believe will open pathways to a new level of design and analysis capabilities in many CFD-driven industries by accurately capturing patterns a traditional CFD might miss or cannot quickly process.

Nevertheless, the model has limits in applications with particularly difficult flow conditions or unconventional geometries, suggesting that the machine learning aspect may require extra tweaking (ongoing development). Moreover, even though this model gives results in the issues considered here, this is not always true for other fluid dynamics problems, and so this has to be investigated further.

Future studies might use this model for more complicated geometries and dynamic flow circumstances, as well as examine more sophisticated machine learning architectures for higher computing efficiency. Testing the hybrid strategy for diversity in industrial settings could also bring some further knowledge of this technology for improved real-time applications.

References:

Tu, J., Yeoh, G. H., Liu, C., & Tao, Y. (2023). Computational fluid dynamics: a practical approach. Elsevier.

Zikanov, O. (2019). Essential computational fluid dynamics. John Wiley & Sons.

Rodriguez, S. (2019). Applied Computational Fluid Dynamics and Turbulence Modeling: Practical Tools, Tips and Techniques. Springer Nature.

Gupta, N., Bhardwaj, N., Khan, G. M., & Dave, V. (2020). Global trends of computational fluid dynamics to resolve real world problems in the contemporary era. Current Biochemical Engineering, 6(3), 136-155.

Anderson, D., Tannehill, J. C., Pletcher, R. H., Munipalli, R., & Shankar, V. (2020). Computational fluid mechanics and heat transfer. CRC press.

Yang, Y. C., Ouyang, Y., Zhang, N., Yu, Q. J., & Arowo, M. (2019). A review on computational fluid dynamic simulation for rotating packed beds. Journal of Chemical Technology & Biotechnology, 94(4), 1017-1031.

Yang, Y. C., Ouyang, Y., Zhang, N., Yu, Q. J., & Arowo, M. (2019). A review on computational fluid dynamic simulation for rotating packed beds. Journal of Chemical Technology & Biotechnology, 94(4), 1017-1031.

Hami, K. (2021). Turbulence Modeling a Review for Different Used Methods. International Journal of Heat & Technology, 39(1).

Panchigar, D., Kar, K., Shukla, S., Mathew, R. M., Chadha, U., & Selvaraj, S. K. (2022). Machine learning-based CFD simulations: a review, models, open threats, and future tactics. Neural Computing and Applications, 34(24), 21677-21700.

Rodriguez, S. (2019). Applied Computational Fluid Dynamics and Turbulence Modeling: Practical Tools, Tips and Techniques. Springer Nature.

Heinz, S., Peinke, J., & Stoevesandt, B. (2021). Cutting-edge turbulence simulation methods for wind energy and aerospace problems. Fluids, 6(8), 288.

Sadrehaghighi, I. (2019). Turbulence Modeling. A Review, Report, (1.86), 9.

Rodriguez, S. (2019). Applied Computational Fluid Dynamics and Turbulence Modeling: Practical Tools, Tips and Techniques. Springer Nature.

Roelofs, F., & Shams, A. (2019). CFD—introduction. In Thermal Hydraulics Aspects of Liquid Metal Cooled Nuclear Reactors (pp. 213-218). Woodhead Publishing.

Sheng, W. (2020). A revisit of Navier–Stokes equation. European Journal of Mechanics-B/Fluids, 80, 60-71.

Ferziger, J. H., Perić, M., & Street, R. L. (2019). Computational methods for fluid dynamics. springer.

Zhang, X., Weerasuriya, A. U., & Tse, K. T. (2020). CFD simulation of natural ventilation of a generic building in various incident wind directions: Comparison of turbulence modelling, evaluation methods, and ventilation mechanisms. Energy and Buildings, 229, 110516.

Serban, A., Van der Blom, K., Hoos, H., & Visser, J. (2020, October). Adoption and effects of software engineering best practices in machine learning. In Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) (pp. 1-12).

Pattyam, S. P. (2020). AI in Data Science for Predictive Analytics: Techniques for Model Development, Validation, and Deployment. Journal of Science & Technology, 1(1), 511-552.

Qureshi, M. S., Umar, S., & Nawaz, M. U. (2024). Machine Learning for Predictive Maintenance in Solar Farms. International Journal of Advanced Engineering Technologies and Innovations, 1(3), 27-49.

Tatineni, S., & Chinamanagonda, S. (2021). Leveraging Artificial Intelligence for Predictive Analytics in DevOps: Enhancing Continuous Integration and Continuous Deployment Pipelines for Optimal Performance. Journal of Artificial Intelligence Research and Applications, 1(1), 103-138.

Zonta, T., Da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. Computers & Industrial Engineering, 150, 106889.

Udo, W. S., Kwakye, J. M., Ekechukwu, D. E., & Ogundipe, O. B. (2023). Predictive Analytics for Enhancing Solar Energy Forecasting and Grid Integration.

Stiglic, G., Kocbek, P., Fijacko, N., Zitnik, M., Verbert, K., & Cilar, L. (2020). Interpretability of machine learning-based prediction models in healthcare. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(5), e1379.

Wang, J., Yan, J., Li, C., Gao, R. X., & Zhao, R. (2019). Deep heterogeneous GRU model for predictive analytics in smart manufacturing: Application to tool wear prediction. Computers in Industry, 111, 1-14.

Sharifzadeh, M., Sikinioti-Lock, A., & Shah, N. (2019). Machine-learning methods for integrated renewable power generation: A comparative study of artificial neural networks, support vector regression, and Gaussian Process Regression. Renewable and Sustainable Energy Reviews, 108, 513-538.

Kurani, A., Doshi, P., Vakharia, A., & Shah, M. (2023). A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. Annals of Data Science, 10(1), 183-208.

Niu, W. J., Feng, Z. K., Feng, B. F., Min, Y. W., Cheng, C. T., & Zhou, J. Z. (2019). Comparison of multiple linear regression, artificial neural network, extreme learning machine, and support vector machine in deriving operation rule of hydropower reservoir. Water, 11(1), 88.

Boateng, E. Y., Otoo, J., & Abaye, D. A. (2020). Basic tenets of classification algorithms K-nearest-neighbor, support vector machine, random forest and neural network: A review. Journal of Data Analysis and Information Processing, 8(4), 341-357.

Panchigar, D., Kar, K., Shukla, S., Mathew, R. M., Chadha, U., & Selvaraj, S. K. (2022). Machine learning-based CFD simulations: a review, models, open threats, and future tactics. Neural Computing and Applications, 34(24), 21677-21700.

Van Quang, T., Doan, D. T., & Yun, G. Y. (2024). Recent advances and effectiveness of machine learning models for fluid dynamics in the built environment. International Journal of Modelling and Simulation, 1-27.

Wang, H., Cao, Y., Huang, Z., Liu, Y., Hu, P., Luo, X., ... & Sun, Y. (2024). Recent Advances on Machine Learning for Computational Fluid Dynamics: A Survey. arXiv preprint arXiv:2408.12171.

Sousa, P., Rodrigues, C. V., & Afonso, A. (2024). Enhancing CFD solver with Machine Learning techniques. Computer Methods in Applied Mechanics and Engineering, 429, 117133.

Schwarz, E. L., Pegolotti, L., Pfaller, M. R., & Marsden, A. L. (2023). Beyond CFD: Emerging methodologies for predictive simulation in cardiovascular health and disease. Biophysics Reviews, 4(1).

Sun, X., Cao, W., Shan, X., Liu, Y., & Zhang, W. (2024). A generalized framework for integrating machine learning into computational fluid dynamics. Journal of Computational Science, 82, 102404