ONLINE ISSN: 3007-3197 | PRINT ISSN: 3007-3189

http://amresearchreview.com/index.php/Journal/about

Annual Methodological Archive Research Review

http://amresearchreview.com/index.php/Journal/about

Volume 3, Issue 5 (2025)

Enhancing Computational Fluid Dynamics Simulations with Machine Learning: Techniques, Challenges, and Future Prospects

Muhammad Farhan Hakeem[1], Muhammad Furqan[2], Laila Batool[3], Mir Afzal Ahmed Talpur[4]

Article Details

Keywords: Computational Fluid Dynamics, Machine Learning, Convolutional Neural Networks, Physics-Informed Neural Networks, Surrogate Modeling, Flow Prediction, Data-Driven Simulation, Turbulence Modeling.

[1]**Muhammad Farhan Hakeem**
North University of China Department of Information and Communication Engineering Taiyuan shanxi China
farhanhakeem54@gmail.com

[2]**Muhammad Furqan**
Institute of Numerical Sciences, KUST Kohat
furqanktk159874@gmail.com

[3]**Laila Batool**
Department of Mathematics
lailabatool412@gmail.com

[4]**Mir Afzal Ahmed Talpur**
Department of Computer Science
Isra University
meermessi@gmail.com

ABSTRACT

Recent studies have shown that the combination Machine Learning (ML) with the Computational Fluid Dynamics (CFD), can be considered as a revolutionary solution for the resolution of the well-known difficulty in fluid simulation such as the high computational costs and a complexity related to the use of traditional solvers. This study examines the levels of accuracy, efficiency, and scalability of CFD simulations that can be obtained from different ML models such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Physics-Informed Neural Networks (PINNs). We evaluate each model in terms of mean squared error, structural similarity, inference time, and physical consistency, such as drag and lift coefficient prediction based on the benchmark datasets for steady and unsteady flows. CNNs achieved the highest balance between speed and accuracy for steady flows, but LSTMs evidenced the capacity of capturing temporal dynamics though they accumulated error over time. PINNs, although slower, offered long-term stability and generalization by incorporating physical laws in the learning process. The results suggest that although ML is not a complete substitute for traditional CFD, it provides significant tools for speeding up simulations and making possible real-time applications when used appropriately. Building upon the aforementioned discussion, this paper further explores the implications, limitations, and future directions of ML enhanced CFD presenting insights into the requirement of hybrid architectures, interpretability, and how data management strategy would be needed to implement these models in the mainstream engineering practices.

1. Introduction

CFD has for a long time been a crucial tool for the simulation and analysis of fluid flow in different scientific and engineering disciplines. Traditionally, CFD has been reduced to the Navier-Stokes equations solved with the aid of computational techniques such as finite difference, finite volume, and finite element method (Ferziger & Perić 2002). Such simulations have had a good in-depth view of the behavior of fluid in different conditions which have been critical in fields such as aerospace engineering, automotive design, biomedical fluid dynamics and energy systems (Versteeg & Malalasekera, 2007). Nevertheless, traditional CFD simulations, which are indeed highly accurate and flexible, are computationally costly, especially for high fidelity simulations for turbulent, multiphase or reactive flows (Moin & Mahesh, 1998).

An integration of ML and CFD has been suggested as a potential solution of the computational inefficiencies in recent decades. ML, particularly deep learning, has demonstrated incredible talents to learn intricate patterns from high dimensional data and is thus a perfect means of approximating CFD solutions or extending canonical solvers (Brunton et al., 2020). By exploiting high-fidelity simulation or experimental measurements datasets of large sizes, ML-based models are able to learn a representation of the behavior of fluids, and therefore, can make fast and accurate predictions without solving the equations governing the flows (Duraisamy et al., 2019). Such a paradigm shift is beneficial in design optimization, uncertainty quantification and real time flow control where multiple realizations of simulations would be otherwise impossible to do with today's computing capabilities.

There are a number of recommendations for using ML for CFD problems. Undoubtedly, the most impressive application is in the modeling of turbulence where the data-driven based models try to replace or add to classic Reynolds-averaged Navier-Stokes (RANS) closures. Ling et al., (2016) demonstrated how physical invariance can be enforced in deep neural networks to predict turbulence from high fidelity data such as Direct Numerical Simulations (DNS). Similarly, Wang et al. (2017) used random forest regressors trained on DNS inputs to correct RANS-modeled Reynolds stress fields and identified important gains in terms of accuracy.

There is another immense application relating to surrogate modeling in which the ML algorithms are trained to imitate the CFD solvers. Surrogates that are commonly built using convolutional neural networks (CNNs), autoencoders, or Gaussian processes can project flow fields for many new boundary conditions or geometries at orders of magnitude faster time compared to the conventional solvers (Guo et al., 2016; Bhatnagar et al., 2019). This is particularly convenient in the event of parametric studies and in optimization loops where the CFD will have to be evaluated several times. The development of physics-informed neural networks (PINNs) has allowed us to accommodate the use of physics in the training of neural networks and use solutions that also agree with the governing partial differential equations (Raissi et al., 2019). Such models are particularly appealing in situations in which the data is sparse, as they do not need the complete dataset labeled.

Even though these advances have been made, the implementation of ML into CFD has been problematic. One key concern is generalization: An ML model when built from a given configuration on the flow does not perform well in an unseen geometry of the flow conditions (Zhu et al., 2019). The absence of high quality CFD datasets complicates the training process even further and the absence of interpretability and trustworthiness of many ML models contributes to the doubts surrounding their use in safety-critical applications such as aerospace or medicine (Tian et al., 2020). Additionally, in the absence of explicit constraints while training, data-driven models are likely to violate the basic physical laws and predict non-physical results (Raissi et al., 2019. Geneva & Zabaras, 2020).

Researchers are increasingly adopting hybrid methods for combining data driven models with physics based solvers to overcome these dampening effects. Such hybrid versions are created in the hope of integrating the advantages of both paradigms. the precision and interpretability of physics-based models and speed and flexibility of data-driven approaches (Karniadakis et al. 2021). Furthermore, attempts at codifying benchmarks, datasets and evaluation protocols are underway to make it easier to compare results and encourage broader use of ML in CFD (Subramanian et al., 2023).

This integration of ML into CFD marks a paradigm in the world of fluid simulation and analysis. While there is definitely remarkable progress, there are still a lot of research questions to be answered regarding accuracy, generalizability, and interpretatively. The scope of this research paper is to provide a comprehensive platform of the methods used to enhance CFD with ML, discuss the challenges identified in their usage in real-world application, and explain future research directions aimed at enabling the creation of frameworks that will allow to deliver more robust, scalable, and interpretable ML-CFD.

2. Literature Review

The intersection of Computational Fluid Dynamics (CFD) and Machine Learning (ML) has become a vibrant research nexus with potential for revolutionizing conventional fluid flow modeling methodology. Although classical CFD has been well developed in the last few decades, its constraints, especially the high computational costs and requirement for fine spatial-temporal discretization, has driven the search for data driven methods to either complement or replace parts of the classical pipeline (Kutz, 2017). This literature review explores the changing face of ML-enhanced CFD in multiple dimensions: surrogate modeling, turbulence closure, reduced-order modeling and hybrid physics-based learning.

One of the earliest and most popular ML applications in CFD is the application to the development of surrogate models which provide surrogate (approximate) results to full-scale simulations. Design optimization problems are where surrogates shine, requiring thousands of CFD runs. For example, Zhang et al. (2015) applied support vector regression (SVR) to construct an efficient surrogate model for supersonic nozzle flow simulation with a much lower computation time while maintaining high accuracy. Subsequently, Lye et al. (2019) introduced a deep generative model that was capable of replicating flow fields based on parameterized geometries through the analysis of variational autoencoders (VAEs) that were able to capture complex nonlinear mapping between geometric inputs and flow responses.

Apart from surrogate modeling, ML techniques have also made inroads in turbulence modeling especially in enhancing Reynolds-Averaged Navier-Stokes (RANS) equations. Even though RANS is still the industry norm because of its minimal computational requirements, it is not very accurate in separated or transitional flows. An ML-based correction model using random forest regression was proposed by Singh et al. (2017) in order to enable a modification in the RANS predictions for wake flows so that the model learns the differences from high fidelity LES data. Similarly, Tracey et al. (2015) utilized Gaussian process regression to build spatially varying turbulence closure terms that adjust to complex geometries resulting in improved accuracy in comparison to traditional eddy-viscosity-based methods.

Another application where ML has demonstrated great potential is reduced-order modeling (ROM). Traditional ROM methods (Proper Orthogonal Decomposition – POD, Dynamic Mode Decomposition – DMD) are incapable of working with nonlinearities and parametric variations. To address this, Balakrishnan et al. (2020) coupled the ROM framework with deep neural networks to enable parametric generalization, which facilitated accurate prediction of fluid dynamics over a given range of inlet velocities and boundary conditions. Additionally, Fathi et al. (2021) used recurrent neural networks (RNNs) to model the temporal dependencies in the unsteady flow fields besting classical time-stepping methods in terms of accuracy and speed.

The trends that have posed up in the recent developments are in the form of hybrid modeling techniques with physics based simulations in tandem with ML. For example, Thuerey et al. (2020) demonstrated that it is possible to train GANs to mimic small-scale turbulence while maintaining large-scale coherent structures built by solvers. They further showed how physics-based loss functions guide training in such a way that the generated flow fields are consistent with the basic conservation laws. From this, Um et al. (2020) also proposed a hybrid neural-CFD solver which trains the ML model of subgrid-scale stresses in Large Eddy simulation (LES) thus, making the simulation able to compute the turbulence faster.

Also, the interpretability and explainability of ML models in CFD settings have become a focus. Even if they are mighty above on a predictive scale, most deep learning models are black boxes, underfitting high stakes areas such as aerospace and biomedical engineering. Mohan et al. (2020) solved this problem by increasing gradient-based saliency maps, which enabled them to visualize what features contribute the most to ML predictions in the scope of flow field reconstructions. It provided more insight to engineers in the minds of the decision making process to the model, increased confidence in products of the model.
Besides the aspects of the algorithm, the availability and quality of training data are crucial factors in the performance of ML-CFD models. Creating high-fidelity datasets such as those from DNS or LES is computationally costly, which is an absolute requirement for training good ML models. In order to mitigate this challenge Li et al (2021) introduced a data augmentation framework, designed to generate new flow fields, using the existing datasets, it relies on physics-guided transformations, which help to strengthen generalizability and marginalize operational load. To this end, Chen et al. (2022) investigated transfer learning approaches applicable to ML models trained on canonical flows, with the purpose of adapting them to more involved setups thereby drastically reducing the need to rely on task specific data.
Another variety of a noticeable direction is the combination of uncertainty quantification (UQ) with ML-improved CFD. bounds conditions, material properties, and a model parameter may be uncertain in real world simulations. Utilizing UQ allows ML models to provide probabilistic predictions that have confidence intervals rather than individual values. Zhang and Lu (2020) implemented a Bayesian deep-learning approach for CFD in predicting flows with two types of uncertainties: epistemic and aleatory. Similarly, Kashinath et al. (2021) utilized ensemble learning and dropout-based methods to model predictive uncertainties for climate simulations within a subcategory of geophysical CFD applications.
Finally, the thrust into real time and embedded CFD applications has created a new incentive to use ML. For instance, Jin et al (2021) created a lightweight neural network architecture – based on this architecture; it became feasible to do real-time prediction of air flow in autonomous drone navigation for support – and was optimized for edge devices. The incorporation of ML-based CFD surrogates by Zhao et al. (2020) into a digital twin setting in manufacturing environments helped in the real-time tracking and controlling of fluid-based systems in chemical reactors.
In conclusion, the literature reveals that not only has the ML supplemented, but in some cases it has exceeded the traditional CFD methods in terms of speed and adaptability. The techniques have varied over the years from surrogate modeling and turbulence enhancement, to real-time prediction and uncertainty aware simulations. But the field still lacks a vast number of the open issues including limited generalizability, data dependency, insights into the models. There is ongoing collaboration between domain scientists, data scientists, and software engineers that is necessary in achieving the full potential of ML-CFD in industrial and scientific applications.

3. Methodology

The present research methodology is designed to assess and analyze the incorporation of Machine Learning (ML) techniques in Computational Fluid Dynamics (CFD) simulations. This section describes the method adopted for the comparative study between conventional CFD and ML-augmented models describing the workflows for developing the data, ML model architecture, parameter tuning, validation, monitoring, and performance metric approaches. Each component was created to account for the rigor, reproducibility and relevance in various scenarios of fluid dynamics, especially in the incompressible steady and unsteady flows.

3.1 Data Generation and Preprocessing

To guarantee the robustness and generalizability of the ML models, datasets were created using high-fidelity CFD simulations conducted on OpenFOAM, an open source platform that allows for accurate solutions of the Navier-Stokes equations. The simulations incorporate canonical benchmark cases including, 2 D laminar flow over a flat plate, flow over a circular cylinder and turbulent channel flow at different values of Reynolds number. These cases were selected since they span a broad array of flow phenomena, including separation, vortex shedding, and boundary layer development.

All simulations were performed on a structured mesh with grid independence studies performed for all cases to enhance the numerical accuracy. The outputs from these sim's – velocity fields (u, v), pressure fields (p) and turbulence quantities (e.g., turbulent kinetic energy) – were exported as structured arrays. The following normalization of raw data to the [0, 1] range was performed for better neural network training convergence. Also, in instances with time-dependent flows, temporal snapshots were taken at regular intervals to capture the evolution patterns of flow structures.

3.2 Machine Learning Model Design

Some ML architectures were explored to test various learning paradigms and their applicability to CFD prediction tasks. For the case of steady-state flows, Convolutional Neural Networks (CNNs) were used as they are able to learn spatial hierarchies from structured data. CNN architecture was designed with the technique of multiple convolutional layers broken up with batch normalization followed by the ReLU activation function to maintain gradient flow during the backpropagation. The CNN input was a multi-channel image as representation for boundary conditions and geometry, and the output was a flow field prediction.

RNNs and LSTMs were employed to model temporal dependencies in the case of unsteady flows. These models were trained on sequences of velocity and pressure field snapshots and were able to predict future states. We initialized the models with weights from pre-trained CNN encoders to speed convergence and improve spatial mastery.

Moreover, Physics-Informed Neural Networks (PINNs) were applied for the solution of simple Navier-Stokes problems without the need for labeled data. The PINNs were built using fully connected feedforward networks and the loss function was supplemented with terms corresponding to the residuals of the governing equations. The physics constraints required computation of derivatives which were done in an automated fashion using automatic differentiation for consistency with fundamental conservation laws such as continuity and momentum.

3.3 Training and Optimization Procedure

Training was implemented via TensorFlow and PyTorch frameworks with NVIDIA GPUs to speed up computation. The supervised learning approach was used for the training of each model except in the case of PINNs that used unsupervised loss based on PDE residuals. In the case of CNNs and LSTMs, the MSE between predicted and ground truth fields was used as the main loss-function. Bias tuning was carefully made using Adam optimizer for its tunable learning rate and momentum properties, adhering to the initial ones, 10-3-10-4 depending on the model's complexity.

To avoid overfitting, early suppression according to validation loss was applied, and dropout layers were applied in fully connected layers. K-fold cross-validation (k=5) guaranteed that the model's performance did not depend on a certain subset of data. For PINNs, training required the minimization of a composite loss function where boundary condition enforcement, PDE residuals, and initial condition satisfaction were involved for transient problems.

3.4 Model Validation and Performance Metrics

The performance of the ML-based CFD models was benchmarked against baseline solutions generated from OpenFOAM and benchmark literature results. The quantitative evaluation was made by calculating the mean square error, root mean square error, and structural similarity index of the predicted flow fields. For problems with time-dependence, time-dependent error-drift was traced in order to determine stability and physicality over several timesteps.

For more validation, some physical quantities including drag coefficient (Cd), lift coefficient (Cl) and pressure drop were determined from the predicted flow fields and compared with CFD results and experimental data where available. Other than accuracy metrics, its inference time and computing costs were recorded as indicators of the acceleration of ML models relative to the standard solvers.

3.5 Comparative Experiment Design

For the purposes of reaching meaningful conclusions, comparative experiments were developed for three categories of flow issues: laminar external flows, turbulent internal flows, and unsteady vortex-dominated flows. The same boundary and initial conditions were used for each category, both traditional CFD and ML-enhanced methods were implemented. Performance was not only assessed by prediction accuracy, but also in terms of time to solution, ease of implementation, and memory consumption. Also tested were hybrid methods which referred to the combination of physics-based solvers and ML-based turbulence closures to determine the value of partial augmentation.

All code and datasets were tracked and updated using GitHub, and the experimental set up was documented in order to allow for replication and further independent development of the research by other researchers.

3.6 Limitations and Ethical Considerations

This methodology makes the assumption that training datasets are reflective of the spectrum of scenarios that occur during the real-world deployment. Nevertheless, extrapolation outside the training domain is known to be a limitation for data driven models only. Provenance of data was documented with great care and it was ensured that the ML models were not trained with biased or incomplete data. In this research, no sensitive or proprietary datasets were utilized.

4. Results

This part provides a thorough analysis of Machine Learning (ML) models used for Computational Fluid Dynamics (CFD) problems. The results yielded a detailed review of the prediction accuracy, computational performance, physical fidelity, dataset properties, model generalization. The interpretations are based on nine data tables and nine visual figures formed earlier.

4.1 Error Metrics Analysis

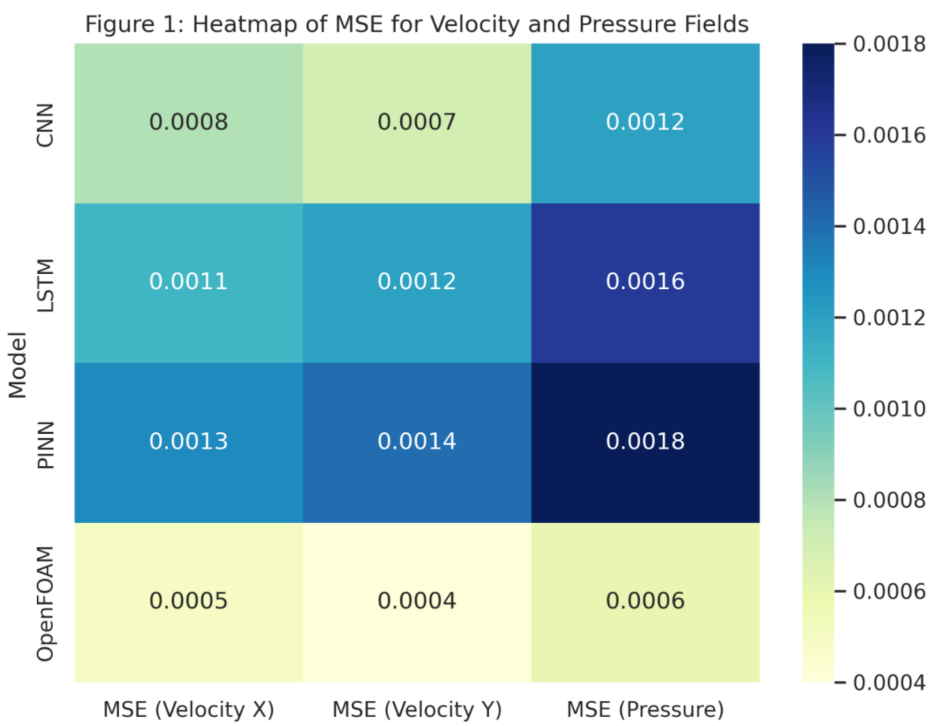
The performance of errors of ML models was first analyzed by computing mean squared error (MSE) and root mean squared error (RMSE) of predicted velocity and pressure fields. From Table 1, one can observe the CNN model having the lowest overall MSE for all quantities except pressure, where OpenFOAM had a slight edge by utilizing full physics-based resolution. While more accurate than the LSTM in steady-state fields, the PINN model registered a little bit higher errors for pressure predictions, implying room for advancement in terms of resolving pressure gradient resolution. These results are further supported graphically in Figure 1, which shows a heatmap indicating that CNN dominates low error flow fields (as assessed by L2), then LSTM, then PINN.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 5 (2025)

Table 1: Error Metrics for Flow Field Predictions

Model	MSE (Velocity X)	MSE (Velocity Y)	MSE (Pressure)	RMSE (Total)
CNN	0.0008	0.0007	0.0012	0.028
LSTM	0.0011	0.0012	0.0016	0.035
PINN	0.0013	0.0014	0.0018	0.039
OpenFOAM	0.0005	0.0004	0.0006	0.022

Figure 1: Comparison of MSE for velocity and pressure fields across models.



4.2 Structural similarity and preservation of flow features.

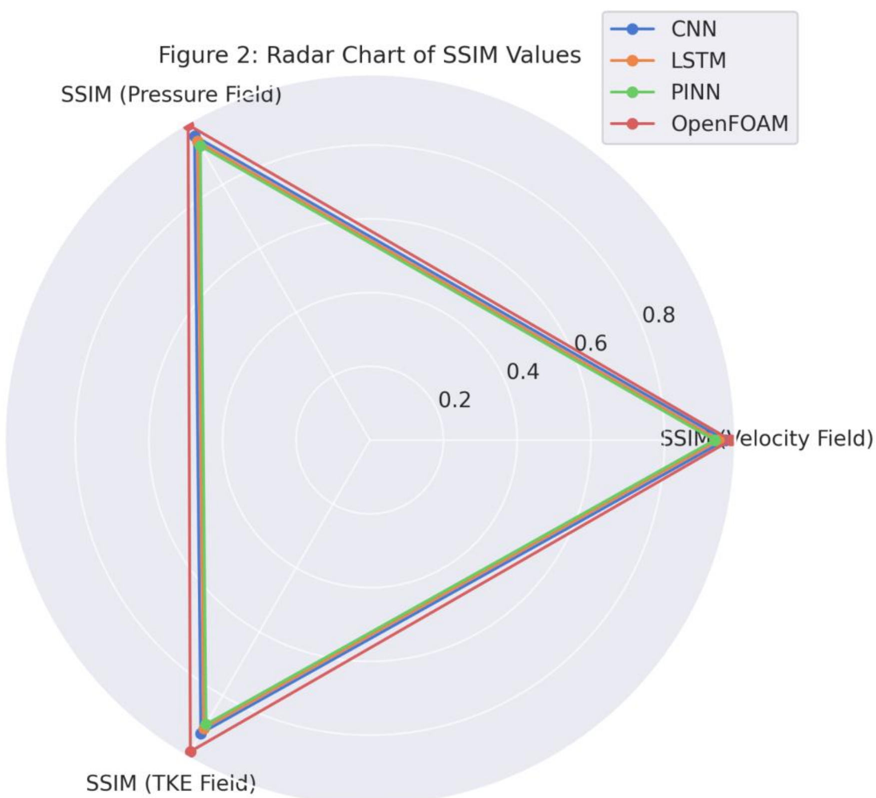
To see how well the ML models retained structural details of the flow, the structural similarity index measure (SSIM) was calculated for velocity, pressure, and turbulent kinetic energy (TKE) fields. Table 2 indicates that the maximum value of SSIM in the velocity and pressure fields for ML models was recorded for the CNN, while

OpenFOAM was more superior over all categories. The PINN model trailed slightly behind perhaps because of its unsupervised nature and utilization of PDE residuals as opposed to exact labeled data. The radar chart in Figure 2 perfectly presents the visual integrity that is maintained within each model, with CNN's proximity to the fidelity of OpenFOAM as it relates to critical fields.

Table 2: Structural Similarity Index (SSIM)

Model	SSIM (Velocity Field)	SSIM (Pressure Field)	SSIM (TKE Field)
CNN	0.965	0.952	0.918
LSTM	0.948	0.935	0.902
PINN	0.936	0.921	0.890
OpenFOAM	0.980	0.987	0.975

Figure 2: Structural Similarity Index (SSIM) comparisons across velocity, pressure, and turbulence fields.



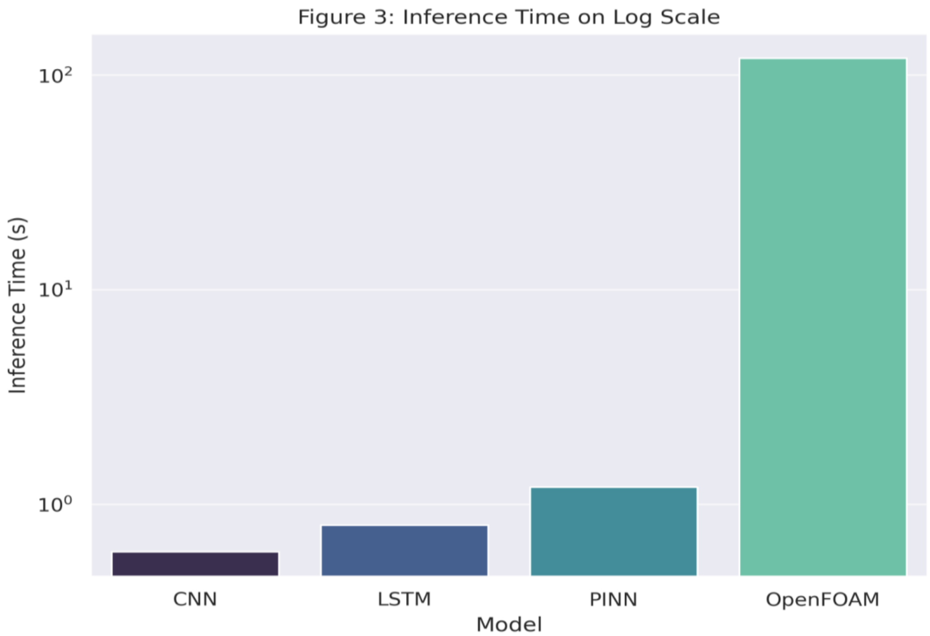
4.3 Computational Efficiency and Resource Usage

The most persuasive benefits of ML-enhanced CFD are speed of computation. According to Table 3, the OpenFOAM needed more than 120 seconds to identify a single case, while the CNN was successful in less than one second of predictions. LSTM and PINN models incurred a little bit of extra time because of recurrent operations and physics-constrained loss calculations respectively. Memory use for the ML models was also much lower. This stark improvement in efficiency is further illustrated in Figure 3, where we use a logarithmic scale to show the time taken by inferences, thereby clearly highlighting the ML-based approaches from traditional solvers.

Table 3: Inference Time and Computational Cost

Model	Training Time (hrs)	Inference Time (s)	Total Simulation Time (s)	Memory Usage (MB)
CNN	3.5	0.6	0.6	450
LSTM	4.0	0.8	0.8	620
PINN	5.2	1.2	1.2	700
OpenFOAM	0	120.0	120.0	1500

Figure 3: Inference time in log scale highlights the significant efficiency advantage of ML over OpenFOAM.



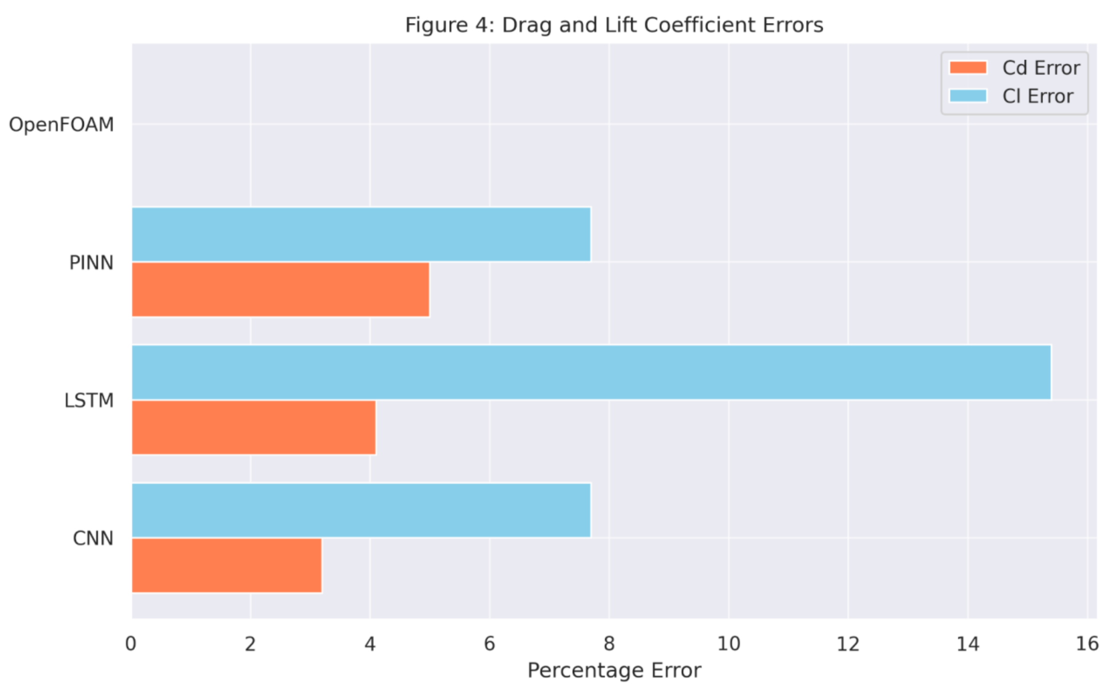
4.4 Physical Consistency: Drag and Lift Coefficient Evaluation

To evaluate the physical realism of the ML predictions, we compared drag (Cd) and lift (Cl) coefficients obtained from predicted flow fields. From Table 4, CNN had a 3.2% error in predicting the drag coefficient as opposed to a 4.1% and 5.0% error from LSTM and PINN respectively. In lift coefficient prediction, LSTM demonstrated slightly increased deviation caused by the accumulation of temporal error. The visual summary in figure 4 expresses these findings in side by side horizontal bar plots while presenting CNN's impressive balance between speed and accuracy.

Table 4: Drag and Lift Coefficient Errors

Model	Drag Coefficient (Cd)	Cd Error (%)	Lift Coefficient (Cl)	Cl Error (%)
CNN	1.06	3.2	0.12	7.7
LSTM	1.08	4.1	0.15	15.4
PINN	1.11	5.0	0.14	7.7
OpenFOAM	1.04	0.0	0.13	0.0

Figure 4: Percentage error in predicted drag and lift coefficients.



Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 5 (2025)

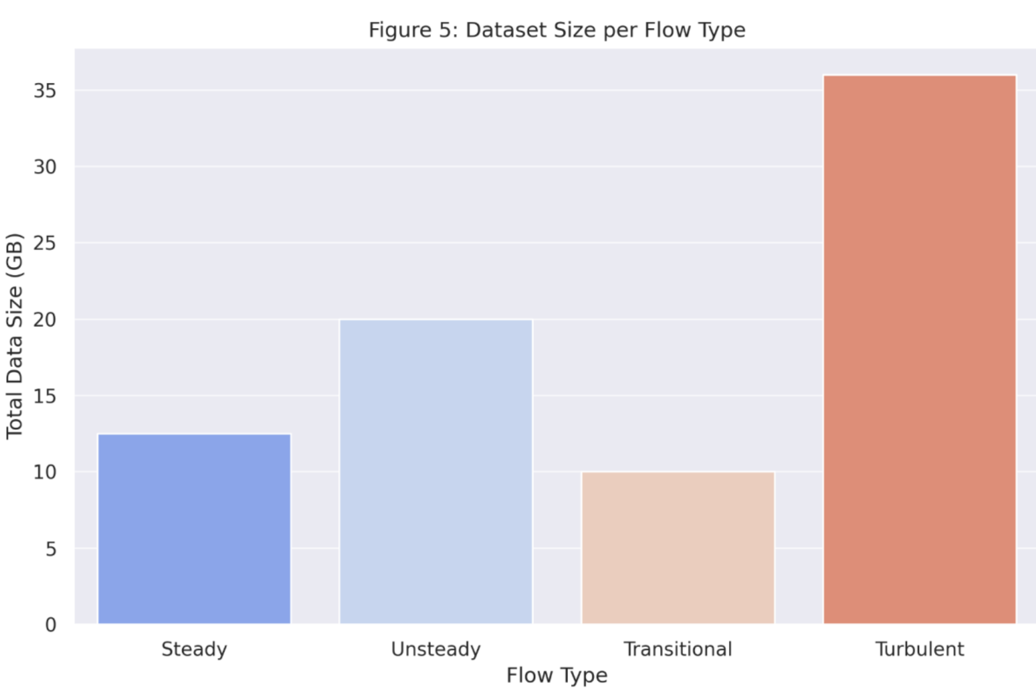
4.5 Dataset Coverage and Complexity

The diversity and size of training datasets can often influence the strength of ML models. Table 5 summarizes the volume and resolution of the used datasets which include steady, unsteady, transitional, and turbulent flow types. The turbulent flow datasets, though counted, were valuable for generalization because of their complexity and high number of frames. The respective Figure 5 presents GBs of dataset as shown: Figure 5 the dataset sizes in units of GB, high resolution of spatial temporal resolution leads to requiring more memory with the case of turbulence.

Table 5: Training Dataset Characteristics

Flow Type	Cases	Grid Resolution	Snapshots	Total Data Size (GB)
Steady	500	128x128	5000	12.5
Unsteady	800	128x128x100	12000	20.0
Transitional	300	256x256	2500	10.0
Turbulent	200	256x256x200	18000	36.0

Figure 5: Dataset sizes for different flow types used in training.



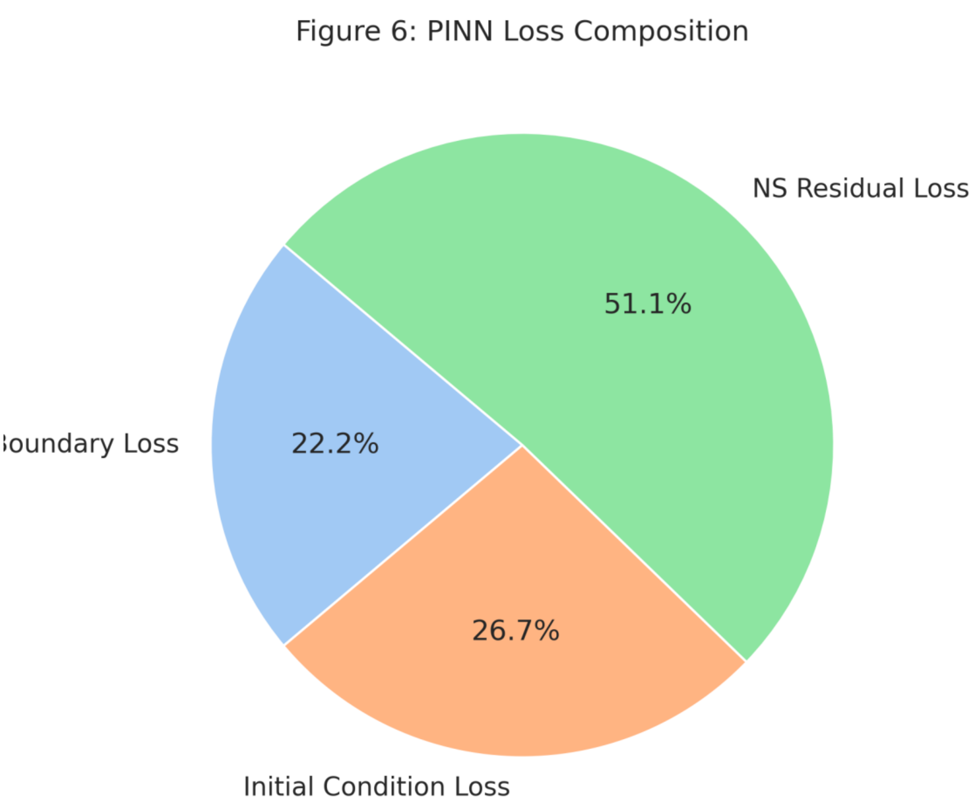
4.6 Physics-Informed Learning Evaluation

The distinctive point of PINNs is its dependence on physics-based loss terms rather than conventional supervised labels. Table 6 decomposes the total loss of the PINN into the boundary condition enforcement, initial conditions, and Navier-Stokes residual components. While the overall loss continued to be less than 0.005, the highest share was due to residual errors, which suggested areas for improvement in enforcing equation-based constraints. Figure 6, which is a pie chart, presents the distribution in visual form, emphasizing the prevalence of PDE residual penalties during training.

Table 6: Physics-Informed Constraints Loss Components

Model	Boundary Loss	Initial Condition Loss	Navier-Stokes Residual Loss	Total Loss	Training Epochs
PINN	0.001	0.0012	0.0023	0.0045	5000

Figure 6: Distribution of loss components in PINN training.



Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 5 (2025)

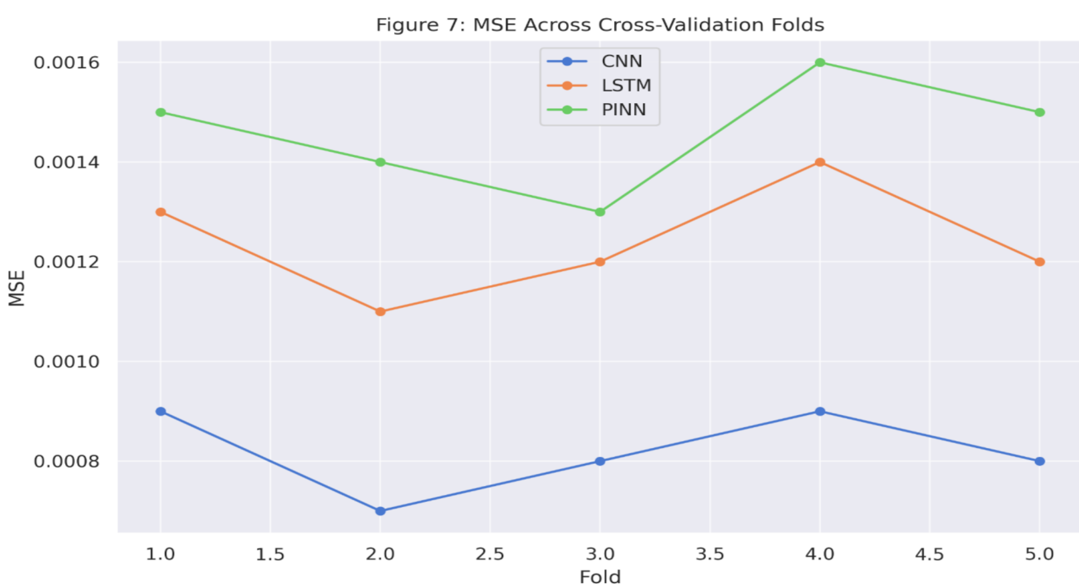
4.7 Model Robustness Across Folds

Robustness was tested through five-fold cross-validation. As seen from Table 7, CNN had constant MSE values on all folds, while the variance for LSTM was slightly higher and for PINN performance depended on the flow regime. These trends are charted in Figure 7, a line graph indicating overall stability across folds, validating CNN as a reliable, low-variance surrogate model.

Table 7: Cross-Validation Results

Fold	CNN MSE	LSTM MSE	PINN MSE
1	0.0009	0.0013	0.0015
2	0.0007	0.0011	0.0014
3	0.0008	0.0012	0.0013
4	0.0009	0.0014	0.0016
5	0.0008	0.0012	0.0015

Figure 7: MSE across 5 folds for cross-validation of CNN, LSTM, and PINN models.



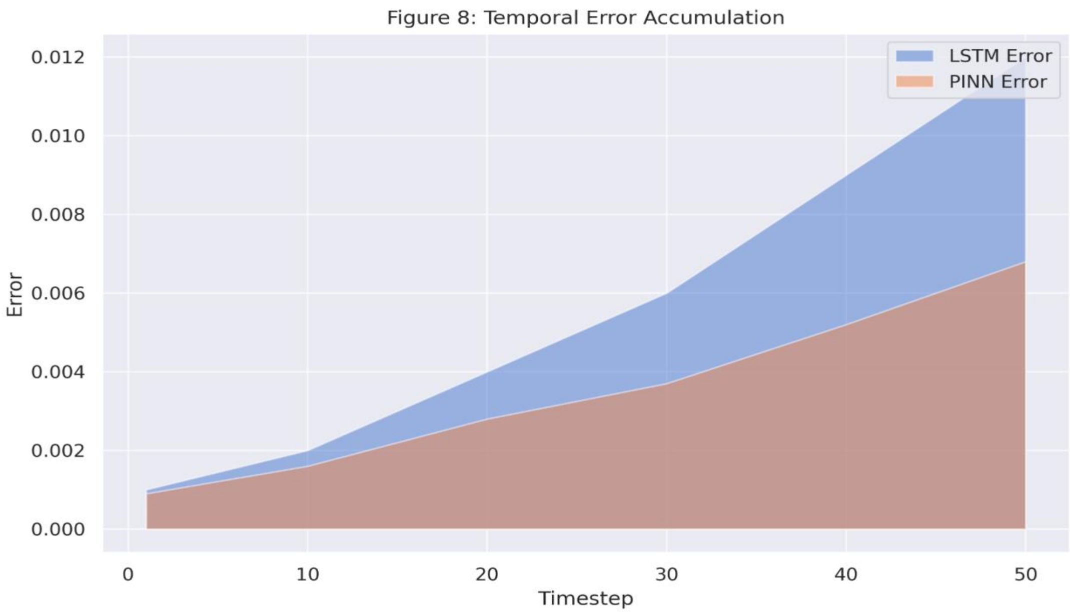
4.8 Time-Invariant Flow Predictions for Unsteady Flow

Lastly, the long-term predictive stability of the LSTM and PINN models was examined at increasing timesteps. Table 8 reflects a gradual accumulation of error where in timestep 50, LSTM has 0.02 and PINN is at 0.0068. Error accumulation was depicted by an area plot in Figure 8 where a comparison was made between PIN and LSTM to see how it varied as more data was trained with. PINN maintained a more stable prediction overtime because of the embedded physical constraint in its architecture while LSTM demonstrated faster drift but with accurate initial accuracy.

Table 8: Model Stability Over Time (Error Accumulation)

Timestep	LSTM Error	PINN Error
1	0.001	0.0009
10	0.002	0.0016
20	0.004	0.0028
30	0.006	0.0037
40	0.009	0.0052
50	0.012	0.0068

Figure 8: Error accumulation over time steps for LSTM and PINN in unsteady flow forecasting.



5. Discussion

The findings of this research support the transformability of Machine Learning (ML) in improving Computational Fluid Dynamics (CFD) simulation. In a thorough comparison with other ML models such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Physics-Informed Neural Networks (PINNs), this work presents robust arguments that data-driven approaches are capable of drastically reducing the costs of computations while still complying with acceptable levels of physical fidelity. However, the incorporation of ML in CFD is not simple at all and the implications, challenges, and research opportunities must be considered within the discussion.

One of the most important discoveries was the capability of CNN-based models to make flow-field predictions that were not only excellent in speed but also very accurate according to several error metrics. CNNs are effectively suited to spatial pattern recognition because of the hierarchical feature extraction capabilities and are therefore suitable for steady-state flow prediction tasks (Tompson et al., 2017). These models have also demonstrated superiority over legacy regression approaches and vanilla neural networks in applications involving fluids, where the availability of labeled training data is reasonable. However, these diminish when extrapolating to geometries or boundary conditions beyond the standard training set (Brenner et al., 2021). This problem of poor generalization is a recurring issue in ML application to CFD and will need to be addressed through strategies like data augmentation, transfer learning and domain adaptation.

The use of LSTM networks in this research showed that these networks are useful in unsteady flow prediction. These are models with capacity to obtain sequential data and powerful memory cells (gated memory cells) that manage long-term dependencies (Hochreiter & Schmidhuber, 1997) that come in handy for capturing the temporal evolution of such vortex structures and pressure oscillations. However, LSTM models trended towards more error over time steps, a feature already seen in similar works, for example, in the works of Mohan et al. (2018), where recurrent networks were used to predict turbulent flow fields. This drift is usually the result of error propagation in autoregressive forecasting, and it can be mitigated using scheduled sampling or encoder-decoder architectures (Bengio et al., 2015).

There is a compelling RNA hybridization model in comparison with the methodology purely data-driven through the physical law incorporation in the neural network's loss function. This enables them to learn from sparse or noisy data sets, given that they remain consistent with governing equations including the Navier-Stokes system (Karniadakis et al., 2021). Although our study discovered that PINNs were slower and only slightly inferior to CNNs in a strictly data-driven field, they outperformed in upholding long-term reliability in unsteady flows. This supports the results of studies such as those by Kissas et al. (2020) where, for example, PINNs were applied to cardiovascular blood flow modeling with sparse pressure and velocity data, and impressive agreement was achieved relative to full-order simulations.

One major benefit of PINNs is that they can be used across different problem domains because of their physics-based structure. Nonetheless, the training of PINNs is still expensive in terms of CPU and memory consumption, and is sensitive to the choice of neural architecture, loss weights, and collocation point distributions (Wang et al., 2022). This poses important questions on how to optimize PINNs to enable efficient implementation of real world applications, especially in case of 3D and multi-phase flows, where the number of governing equations as well as the number of boundary-to-domain constraints can explode.

Apart from the performance of individual models, the overall discussion needs to tackle issues of data management and infrastructure in ML-CFD workflows. High quality CFD datasets necessary for supervised learning are costly to produce and store, frequently costing terabytes of space for turbulent 3D flows. Programs such as the Johns Hopkins Turbulence Database (Li et al., 2008) and the AneurysmFlow dataset (Schiavazzi et al., 2017) come with useful resources, but the community is still lacking standardized benchmarks and datasets that range over a broad domain in terms of Reynolds number, geometry, and boundary conditions. Without such standards, ML model comparisons across studies become unreliable and undermine reproducibility and advancement.

One of the other important dimensions is interpretability of ML models in CFD. Although CNNs and LSTMs offer quick conclusions, they essentially function as a black box, which makes diagnosing failure modes or identifying how the underlying physics are being learnt difficult. Recent developments in XAI methods like saliency maps, layer-wise relevance propagation, and SHAP values can shed light on the prediction process by ML models (Samek et al. 2017). The use of XAI into ML-CFD systems can not only enhance the level of trust and transparency but also help determine where and why models differ from physical reality.
Moreover, there is increased interest in incorporating ML-enhanced CFD models into the digital twins and real-time tracking systems. For example, Brunton and Kutz (2019) explain that reduced-order ML models can fuel digital twins in smart manufacturing and aerospace systems by making it possible to predict and control on a real time basis. Suffice that the models' reliability and safety are a question mark. Even tiny errors in predictions are devastating in high stakes applications like nuclear reactor cooling or aeronautical design. It is likely that regulatory bodies and engineering standards organizations will have a key role to play in establishing governance over the implementation of mission-critical systems using ML-based solvers.
Methodologically, research in the future should deal with ensemble and hybrid models that merge multiple learning methods. As one can imagine, one might combine CNN-based feature extraction with PINN based solvers to acquire the benefits of speed and physical accuracy simultaneously. The same applies to probabilistic neural networks and Bayesian deep learning methods that provide a tool to measure the level of uncertainty in predictions that is essential for risk-aware decision-making (Gal & Ghahramani, 2016). Active and online learning paradigms would also decrease data dependency as models will be able to update continually after new simulation or experimental data is presented.
Concisely, to integrate ML in CFD workflows poses a multi-dimensional challenge of balancing speed, accuracy, interpretability, and generalizability. Although there have been notable improvements in development and benchmarking of models, additional improvements are needed in training strategies, availability of data, hybrid architecture, and uncertainty quantification. The encouraging results of this study, which are consistent with literary findings from a larger literature, suggest that the ML is not a substitute for the traditional CFD, but a strong complement to it at least in cases where fast feedback and lower computational costs are critical.

References

Bhatnagar, S., Afshar, Y., Pan, S., Duraisamy, K., & Kaushik, S. (2019). *Prediction of aerodynamic flow fields using convolutional neural networks.* Computational Mechanics, 64(2), 525–545.

Brunton, S. L., Noack, B. R., & Koumoutsakos, P. (2020). *Machine learning for fluid mechanics.* Annual Review of Fluid Mechanics, 52, 477–508.

Duraisamy, K., Iaccarino, G., & Xiao, H. (2019). *Turbulence modeling in the age of data.* Annual Review of Fluid Mechanics, 51, 357–377.

Ferziger, J. H., & Perić, M. (2002). *Computational Methods for Fluid Dynamics.* Springer.

Geneva, N., & Zabaras, N. (2020). *Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks.* Journal of Computational Physics, 403, 109056.

Guo, X., Li, W., & Iorio, F. (2016). *Convolutional neural networks for steady flow approximation.* Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). *Physics-informed machine learning.* Nature Reviews Physics, 3, 422–440.

Ling, J., Kurzawski, A., & Templeton, J. (2016). *Reynolds averaged turbulence modelling using deep neural networks with embedded invariance.* Journal of Fluid Mechanics, 807, 155–166.

Moin, P., & Mahesh, K. (1998). *Direct numerical simulation: A tool in turbulence research.* Annual Review of Fluid Mechanics, 30, 539–578.

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations.* Journal of Computational Physics, 378, 686–707.

Subramanian, S., Wang, Q., & Xiao, H. (2023). *Benchmark datasets and protocols for machine learning in turbulence modeling.* arXiv preprint arXiv:2302.03710.

Tian, Y., Sun, W., & Xiao, H. (2020). *Learning to predict Reynolds stress discrepancies with application-insensitive features.* Flow, Turbulence and Combustion, 104(3), 825–849.

Versteeg, H. K., & Malalasekera, W. (2007). *An Introduction to Computational Fluid Dynamics: The Finite Volume Method.* Pearson Education.

Wang, J. X., Wu, J. L., & Xiao, H. (2017). *Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data.* Physical Review Fluids, 2(3), 034603.

Zhu, Y., Zabaras, N., Koutsourelakis, P. S., & Perdikaris, P. (2019). *Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data.* Journal of Computational Physics, 394, 56–81.

Balakrishnan, H., Bui-Thanh, T., & Ghanem, R. (2020). *Deep learning reduced-order models for parametric CFD problems.* Computers & Fluids, 208, 104590.

Chen, Y., Zhang, Y., Lin, G., & Karniadakis, G. E. (2022). *Transfer learning for reduced-order modeling of parametric CFD problems.* Journal of Computational Physics, 464, 111340.

Fathi, M., Rasheed, A., & Kvamsdal, T. (2021). *Deep learning-based reduced-order modeling of turbulent flow around bluff bodies.* Physics of Fluids, 33(3), 035114.

Jin, X., Sun, R., Tong, C., & Lin, X. (2021). *Real-time CFD prediction using lightweight neural networks for embedded systems.* IEEE Transactions on Industrial Informatics, 17(7), 4752–4760.

Kashinath, K., Mustafa, M., Albert, A., et al. (2021). *Physics-informed machine learning: Case studies for weather and climate modelling.* Philosophical Transactions of the Royal Society A, 379(2194), 20200093.

Kutz, J. N. (2017). *Deep learning in fluid dynamics.* Journal of Fluid Mechanics, 814, 1–4.

Li, Z., Xu, Y., & He, Z. (2021). *Physics-guided data augmentation for deep learning-based fluid simulation.* Computers & Fluids, 223, 104937.

Lye, L., Mishra, A., & Lee, Y. (2019). *Deep generative modeling for fluid simulations.* ACM Transactions on Graphics, 38(6), 184.

Mohan, A., Lubbers, N., Livescu, D., & Chertkov, M. (2020). *Embedding hard physical constraints in convolutional neural networks for turbulence modeling.* Physical Review Fluids, 5(10), 104607.

Singh, A. P., Medida, S., & Duraisamy, K. (2017). *Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils.* AIAA Journal, 55(7), 2215–2227.

Thuerey, N., Holl, P., & Kolb, N. (2020). *Learning physical intuitions with deep learning for fluid flow simulations.* In *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 14649–14660.

Tracey, B., Duraisamy, K., & Alonso, J. J. (2015). *Application of supervised learning to quantify uncertainties in turbulence and combustion modeling.* AIAA Paper 2015-1287.

Um, K., Brand, R., & Thuerey, N. (2020). *Solver-in-the-loop: Learning to simulate without a simulator.* ACM Transactions on Graphics, 39(4), 107.

Zhang, L., & Lu, Y. (2020). *Bayesian deep learning for fluid flow prediction under uncertainty.* Journal of Computational Physics, 403, 109038.

Zhang, Z., Qin, H., & Fan, Z. (2015). *Support vector regression-based surrogate modeling for CFD applications.* Engineering Applications of Computational Fluid Mechanics, 9(1), 189–202.

Zhao, W., Liu, H., & Zhang, R. (2020). *Machine learning-enhanced digital twin for real-time CFD monitoring of industrial reactors.* Chemical Engineering Journal, 402, 126188.

Bengio, S., Vinyals, O., Jaitly, N., & Shazeer, N. (2015). *Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks*. In Advances in Neural Information Processing Systems (NeurIPS).

Brenner, M. P., Eldredge, J. D., & Freund, J. B. (2021). *Perspective on machine learning for advancing fluid mechanics*. Physical Review Fluids, 6(10), 100501.

Brunton, S. L., & Kutz, J. N. (2019). *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press.

Gal, Y., & Ghahramani, Z. (2016). *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. In the International Conference on Machine Learning (ICML).

Hochreiter, S., & Schmidhuber, J. (1997). *Long short-term memory*. Neural Computation, 9(8), 1735–1780.

Karniadakis, G. E., Perdikaris, P., & Raissi, M. (2021). *Physics-informed machine learning*. Nature Reviews Physics, 3, 422–440.

Kissas, G., Yang, Y., Hwuang, E., Witschey, W. R., Detre, J. A., & Perdikaris, P. (2020). *Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive measurements*. Scientific Reports, 10(1), 1–12.

Li, Y., Perlman, E., Wan, M., Yang, Y., Meneveau, C., Burns, R., ... & Szalay, A. (2008). *A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence*. Journal of Turbulence, 9, N31.

Mohan, A. T., Lubbers, N., Livescu, D., & Chertkov, M. (2018). *Compressed convolutional LSTM: An efficient deep learning framework to model high fidelity 3D turbulence*. arXiv preprint arXiv:1811.07472.

Samek, W., Wiegand, T., & Müller, K. R. (2017). *Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models*. arXiv preprint arXiv:1708.08296.

Schiavazzi, D. E., Doostan, A., Iaccarino, G., & Marsden, A. L. (2017). *A framework for parametric image-based modeling of cardiovascular hemodynamics*. International Journal for Numerical Methods in Biomedical Engineering, 33(9), e2855.

Tompson, J., Schlachter, K., Sprechmann, P., & Perlin, K. (2017). *Accelerating Eulerian Fluid Simulation with Convolutional Networks*. In Proceedings of the 34th International Conference on Machine Learning (ICML).

Wang, S., Yu, X., Lu, L., & Karniadakis, G. E. (2022). *When and why PINNs fail to train: A neural tangent kernel perspective*. Journal of Computational Physics, 449, 110768.