Multi-Fidelity Artificial Neural Networks for Rotorcraft Airfoil Design

Apurva Anand, M Muneeb Safdar, Koushik Marepally, James D Baeder
University of Maryland, College Park, MD, USA

ABSTRACT

Rotor blade design is a fundamental problem in the field of rotorcraft aeromechanics. The conventional design methods using high-fidelity techniques (like Computational Fluid Dynamics (CFD)) are computationally expensive and often become impractical at the early design phase. The use of deep learning techniques for blade design is an efficient alternative to conventional design methods. However, these machine learning techniques often require a lot of training data which are generated using the CFD. The training data generation process becomes a bottleneck and requires a lot of computational effort, before being able to train the neural networks. In this paper, we propose and implement a unique multi-fidelity artificial neural network (MFANNs) architecture that takes low-fidelity (Xfoil) data as inputs and maps/transforms it to high-fidelity data, requiring 1/10 of high-fidelity training data as compared to the conventional high-fidelity artificial neural networks. A comprehensive study on the number of high-fidelity cases required for training MFANNs and the sensitivity analysis of data reduction is also presented in the paper. The multi-fidelity architecture is integrated with an evolutionary algorithm to optimize the airfoil.

INTRODUCTION

The design and optimization of rotor blades is an area of active research for many decades. An important aspect of blade design is the selection of airfoil geometries along the span of the blade which meets a certain performance criteria. Designing an airfoil geometry requires knowledge of multiple fields like aerodynamics, structures and acoustics (Ref. [1]). The traditional design techniques iteratively optimize the shape of the airfoil either by using a physics-based high-fidelity code or simplified low-fidelity codes. Most high-fidelity codes use gradient-based optimization and therefore makes use of adjoint-based CFD (Computational Fluid Dynamics) solvers which are very accurate but computationally expensive (Refs. [2, 3]). One more issue with such optimization techniques is that the process needs to be restarted for any change in the objective function. Unlike high-fidelity codes, the simplified low-fidelity solvers are computationally cheaper and can be used with an optimization algorithm (like Genetic Algorithms) (Ref. [4]). But low-fidelity codes aren't accurate in predicting drag coefficient (C_d), the details of which are also presented in this paper.

For airfoil optimization, an efficient and accurate mapping function (from airfoil geometry to lift coefficient (C_l), drag coefficient (C_d) and pitching moment coefficient (C_m)) is necessary. There has been recent developments in the field of machine learning (ML) architectures to map airfoil geometries with performance parameters (Refs. [**anand2024novel**, 1, 4, 5, 6]). Most of these architectures use artificial neural networks (ANNs) for function approximation and are very accurate. The ANNs are either trained on data generated by high-fidelity code or low-fidelity code. While the ANNs are very accurate, the fidelity of code used to generate data determines how accurate the predicted performance polar is. While the ANNs trained on high-fidelity data are very accurate, one major drawback of such architecture is the cost of data generation (4 order higher than Xfoil). A lot of training data is required to train the ANNs and cover a wider geometric design space. This necessitates the use of high-performance computing facility to generate data with high-fidelity codes.

To overcome the limitation of the cost associated with training ANNs on high-fidelity data, a multi-fidelity architecture is proposed. There has been some recent development of the multi-fidelity framework which combines data from both low-fidelity and high-fidelity codes. The major advantage of using such architecture is the reduction in the number of high-fidelity data to train the ANNs. Nagawkar et. al. [7] used multi-fidelity neural networks for airfoil shape optimization. Zhang et. al. [8] used a multi-fidelity deep learning model with various infilling strategies for aerodynamic optimization. A multi-fidelity convolution neural network for aerodynamic optimization which dynamically incorporates new data in each optimization loop is used in Reference [9]. A data fusion-based multi-fidelity architecture is used by Lei et. al. (Reference [10]) to predict aerodynamic coefficients using a deep neural network. Reference [11] uses a multi-fidelity convolution neural networks for high-dimensional data such as computational flow field. Tekaslan et. al (Reference [11]) used Euler solver as low-fidelity code and viscous code with Spallart-Allmaras (SA) turbulence model as high fidelity code.

In this paper, a simple yet effective prediction-correction

Presented at the 50th European Rotorcraft Forum, Marseille, France, September 10–12, 2024.

based multi-fidelity architecture is proposed to predict the aerodynamic quantities of interest (C_l, C_d and C_m). Xfoil (Reference [12]) is used as a low-fidelity code which is computationally faster and HAM2D (Reference [13]) is used as a high-fidelity code which is an in-house developed Reynolds-Averaged Navier-Stokes (RANS) based code. The low-fidelity code is used as a predictor which predicts aerodynamic coefficients (or performance polar) to a reasonable accuracy, and the high-fidelity code is then used to correct it to match HAM2D data. The hypothesis behind using a predictor-corrector architecture is that the predictor maps airfoil geometry to a reasonable value of C_l, C_d and C_m, and the corrector maps the predicted C_l, C_d and C_m to high-fidelity data. This way the number of high-fidelity data required to train the model is reduced significantly and therefore the cost of data generation goes down without compromising the accuracy of the model.

This work is organized in three parts. The first part pertains to airfoil parametrization and details of low and high-fidelity codes. The architecture of low and high-fidelity artificial neural networks is explained in the next section. The details of accuracy of both neural network architecture is also presented. At the end, multi-fidelity architecture is presented along with the comparisons of true versus predicted aerodynamic performance polar. The multi-fidelity neural networks is also integrated with genetic algorithm to optimize airfoil for improved performance. As remarked in the opening paragraph, airfoil polar prediction is essential for blade design and this work is expected to considerably enhance the confidence with which multi-fidelity artificial neural networks can be used for such problems.

TECHNICAL APPROACH AND METHODOLOGY

This section sequentially describes the methodology employed in generating a representative database of airfoil geometries using both low-fidelity (Xfoil) and high-fidelity (HAM2D) CFD codes, airfoil parametrization, design of experiments and reduced order surrogate models.

Airfoil Parametrization

This paper presents a detailed investigation of the application of multi-fidelity artificial neural networks (MFANNs) for the prediction of airfoil performance polar (lift coefficient (C_l), drag coefficient (C_d) and pitching moment coefficient (C_m)). For training the neural networks, airfoil geometries are given as inputs to the networks. Therefore, airfoil geometries are parametrized by class function - shape function transformation originally used by Kulfan (Ref. [14]). For our research, we use Chebyshev polynomials as a shape function instead of Bernstein polynomials which was used in Ref. [14].

There are multiple ways of parametrizing an airfoil geometry. One way is to use an airfoil's camber and thickness to parameterize its geometry (Ref. [6]). A major issue with this methodology is that the camber and thickness are not explicitly defined for an airfoil. Therefore, to determine the camber and thickness, an inverse-optimization problem needs to be solved and then the airfoil is parametrized using this camber and thickness distribution. While the parametrization is computationally inexpensive, solving inverse-optimization for determining camber and thickness is a computationally expensive process and takes up to a few minutes in a single core machine for one airfoil.

A simpler and more effective way of parametrizing the airfoil is by using the upper surface and thickness distribution. Here, thickness is the difference between the upper and lower surface and always remains positive. A negative thickness would infer that there are intersecting surfaces in the airfoil and therefore the airfoil isn't practical. This way of parametrizing airfoil is very quick and works like a push-button solution. More details of this is presented in Ref. [6].

Figure 1 shows the comparison of CST coefficients for airfoil thickness distribution for SC1095 airfoil using 8th order polynomial. It can be seen from Figure 1 that the Chebyshev polynomials carry maximum information in the first 4 coefficients whereas the Bernstein polynomials don't have a well-defined pattern. This is attributed to the near-orthogonality nature of Chebyshev polynomials. For this work, we use an 8th-order (9 coefficients) Chebyshev polynomial to parameterize upper surface and thickness distribution, and two extra coefficients are used to parameterize trailing edge (TE) thickness - totalling 20 coefficients.

The chord-wise coordinate (ψ) and the normal coordinate (ζ) is used to represent these surfaces. Equation 1 explains the parametrization of upper surface, while Equation 2 is used for thickness parametrization, where C and S represents airfoil class and shape functions as presented in Equations 3 and 4 respectively. The lower surface can be constructed by subtracting the thickness from the upper surface as in Equation 5.

$$\zeta_U(\psi) = C(\psi)S_U(\psi) + \psi\zeta_{TE,U} \tag{1}$$
$$\Delta\zeta(\psi) = C(\psi)S_{\text{thn}}(\psi) + \psi\Delta\zeta_{TE} \tag{2}$$
$$C(\psi) = \sqrt{\psi} \times (1-\psi) \tag{3}$$
$$S(\psi) = \sum a_i S_i \tag{4}$$
$$\zeta_L(\psi) = \zeta_U(\psi) - \Delta\zeta(\psi) \tag{5}$$

Low and High Fidelity Solver for Data Generation

Multi-fidelity framework uses data from both low-fidelity code and high-fidelity codes. For low-fidelity, we use Xfoil [12], which is an open-source panel method code with

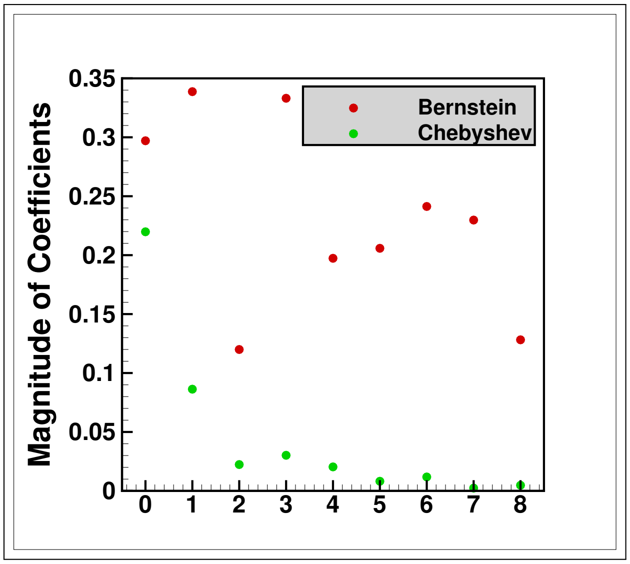


Figure 1: Comparison of the magnitude of 8th order CST coefficients for airfoil thickness parameterization using Bernstein and Chebyshev polynomials for SC1095 airfoil. Red: Bernstein Coefficients, Green: Chebyshev Coefficients

boundary layer corrections for viscous flows.

Low-Fidelity Code XFOIL: XFOIL is a specialized software tool used for analyzing and designing subsonic airfoils [12]. The core functionality of XFOIL revolves around its ability to compute the aerodynamic characteristics of airfoils. This is done by solving the potential flow over the airfoil using a panel method, which divides the airfoil surface into a series of panels and calculates the induced velocity at each panel. In addition to potential flow calculations, XFOIL incorporates boundary layer models to account for the transition from laminar to turbulent flow. This allows it to predict performance aspects such as stall characteristics and drag polar curves, which are essential for evaluating the efficiency and effectiveness of an airfoil design.

XFOIL is known for its user-friendly GUI and the capability to be controlled externally via text files. These text files allow users to specify various input parameters, such as flow conditions, the number of panels, Ncrit, and the range of angles of attack. This external control facilitates automation, enabling users to generate comprehensive aerodynamic data across different Mach numbers and Reynolds numbers efficiently. However, XFOIL sometimes faces convergence issues, particularly at high angles of attack and high subsonic Mach numbers. At elevated angles of attack, the flow over the airfoil can become highly nonlinear and complex, leading to significant flow separation and vortex shedding. These phenomena can challenge the panel method used by XFOIL, causing difficulties in convergence. Similarly, at higher subsonic Mach numbers, compressibility effects introduce additional complexities to the flow field, adding to convergence issues. To address these challenges, we have developed an automation script that interacts with XFOIL to ensure reliable convergence. The script initially sets the angle of attack (AoA) increment to a default value, typically 1°. If XFOIL encounters divergence or convergence issues, the script modifies the AoA increment and if the convergence issues persist after adjusting the AoA increment, the script incrementally increases the number of panels from the default value (usually 160) to higher values such as 180, 200, or more, as needed. The comparison of performance polar predicted by XFOIL for different number of panels is presented in Figure 2. From Figure 2, it can be inferred that there is no significant change in C_l, C_d and C_m when the number of panels varies between 120 to 240. Therefore, the step-by-step refinement of the computational grid helps address convergence problems by enhancing the resolution of the simulation without compromising on the consistency of the solution given by XFOIL. If convergence is still not achieved, the script can further adjust both the AoA increments and the number of panels iteratively, ensuring that XFOIL generates accurate aerodynamic data. In addition to managing convergence issues, XFOIL may not fully simulate turbulent flow conditions. However, it can approximate turbulent flow by adjusting Ncrit values and transition points. In our studies, we found that varying the Ncrit value from the default of 9 to lower values, such as 1, and setting the transition points to 5% of the chord length provided results that closely mimicked fully turbulent conditions. The values of Ncrit lower than 1 or transition points smaller than 5% led to divergence issues. Therefore, for relatively better turbulent flow predictions, we use a Ncrit of 1 and set the transition points to 5% of the chord length.

High-Fidelity Code HAM2D: Similarly, an in-house developed Reynolds-Averaged Navier-Stokes (RANS) based code, HAM2D is used for generating high-fidelity data for training. This solver is built on a Hamiltonian path-based solution methodology (Ref. [13]) and uses a fifth-order WENO (Ref. [15]) scheme for spatial reconstruction, with Roe's approximate Riemann solver for inviscid flux, second-order central differencing for viscous flux, and a preconditioned GMRES (Ref. [16]) for implicit time integration. The turbulent boundary layer is modelled using the one equation Spalart-Allmaras (SA) (Ref. [17]) model. HAM2D is extensively tested and validated with various airfoil data available in the literature (Ref. [18]). The data used for the study in this paper has been generated for standard rotorcraft airfoil geometries, the details of which are available in Reference [5, 6, 19]. For all airfoil geometries, unstructured all-quad grids are generated with a wall spacing of y+ = 1 using an in-house mesh generation tool (Ref. [20]). All stages of high-fidelity data generation has been automated using python scripts.

NEURAL NETWORKS

This section provides a detailed explanation of the neural network architecture used in this study.

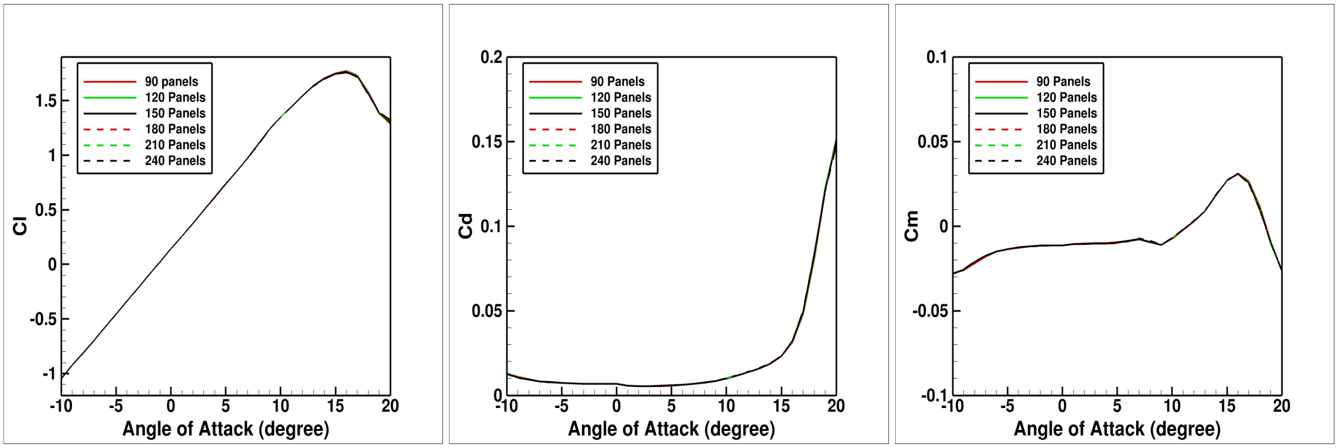


Figure 2: Comparison of aerodynamic coefficients predicted from Xfoil for different number of panels. From left to right; Comparison of C_l polar, Comparison of C_d polar, Comparison of C_m polar

Low-Fidelity Artificial Neural Networks (LFANNs)

The low-fidelity artificial neural networks (LFANNs) is trained on data generated from low-fidelity code (Xfoil). For this study, fourteen industrially relevant rotorcraft geometries are selected, the details of which are presented in Table 1. To generate the training data, the geometric design space is filled by perturbing the CST coefficients of each baseline airfoil using Latin-hypercube sampling. A total of 40 perturbations are generated for each baseline airfoil, and therefore 560 (14 $\times$ 40) airfoils are generated. Xfoil simulations are performed at 5 mach numbers (0.3, 0.4, 0.5, 0.6 and 0.7) and linearly varying corresponding Reynolds number at fully-turbulent condition, and 31 angles of attack (-10 to +20). More details of the design of experiments are presented in Refs. [5, 6, 19].

Table 1: List of representative databases of 14 baseline rotorcraft airfoils selected from industrially relevant geometries

Rotorcraft Airfoils	
Gluharoff/Sikorsky GS-1	NASA/Langley RC-SC2
NACA 0012	Sikorsky/Balch SC1095
NACA 0012-64	Sikorsky/Lednicer, Owen SC2110
NACA 0015	Sikorsky/Flemming SSC-A09
NACA 64A-015	Boeing-Vertol V23010-1.58
RC(3)-10	Boeing-Vertol V43015-2.48
RC(4)-10	Boeing-Vertol/Dadone VR-12

Since Xfoil is computationally cheaper, it takes 15 hours to run 86,800 (560 $\times$31 $\times$ 5) cases. Five neural networks are trained, one at each Mach number. The LFANNs take airfoil geometry as inputs and gives performance polar (C_l, C_d and C_m) as outputs. For each training data, the outputs are vectors of C_l, C_d and C_m, and therefore the angle of attack isn't used as input. It is to be noted that $\log(C_d)$ is used for training instead of C_d to ensure that the drag is always positive. More details on the use of vectorised implementation are presented in Refs. [5, 6, 19]. A comparison of error distribution of vectorized and non-vectorized implementation is also presented in the last section of this paper. Therefore, for each LFANNs architecture, the input is a 20-dimensional vector and the output is 93 (31 (angles of attack) $\times$ 3 (C_l, C_d and C_m)) dimensional vector. The architecture of LFANNs is presented in Figure 3. The loss function for LFANNs is presented by Equation 6, where P represents low-fidelity performance polar. Equation 7 shows the mapping of airfoil CST coefficients to performance polar, where O_{Low} represents the output from LFANNs, $func_{LFANN}$ represents the approximate function from trained LFANNs and CST represents airfoil CST coefficients.

$$J_{LFANN} = \frac{1}{\#\text{Train}} \sum_{\text{Train}} \left(\hat{\text{P}}_{\text{True}} - \hat{\text{P}}_{\text{Predicted}}\right)^2 \tag{6}$$

$$O_{Low} = func_{LFANN}(CST) \tag{7}$$

Figures 4 to 6 presents a comparison of predicted coefficients (from LFANNs) and true coefficients (Xfoil) for the test data which is not a part of the training. The comparison is provided for Mach 0.3. For all other Mach numbers, all three polar show similar trends. From Figures 4 to 6, it can be inferred that LFANNs is very accurate in predicting the polar of C_l, C_d and C_m.

High-Fidelity Artificial Neural Networks (HFANN)

Like LFANNs, high-fidelity artificial neural networks (HFANNs) is trained on data generated from high-fidelity code (HAM2D). The same cases that were used for training LFANNs are used to train HFANNs. Therefore, there are a total of 560 airfoils simulated at 31 angles of attack at 5 Mach numbers. The training methodology, neural network architecture and the cost function for HFANNs is the same as

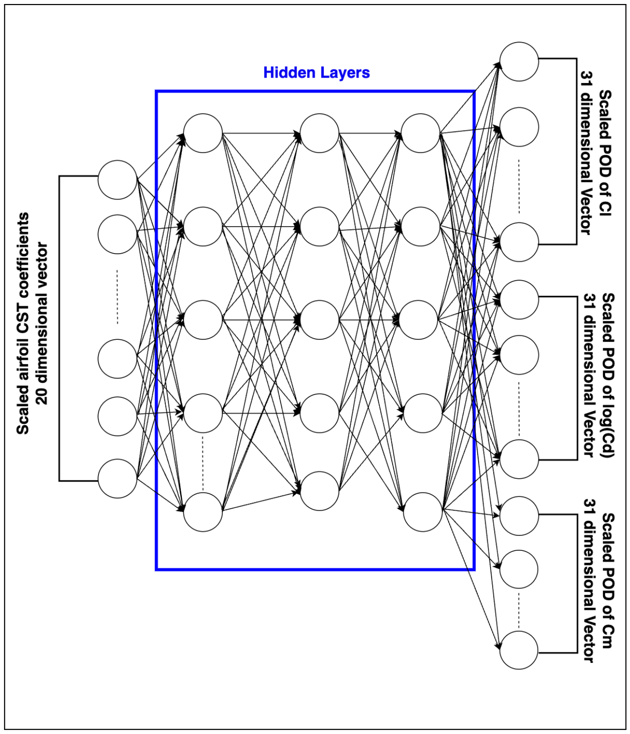


Figure 3: Architecture of low-fidelity Artificial Neural Network (LFANNs) with vectorized implementation of output polar. Inputs: Airfoil CST coefficients, Outputs: Polar of C_l, C_d and C_m

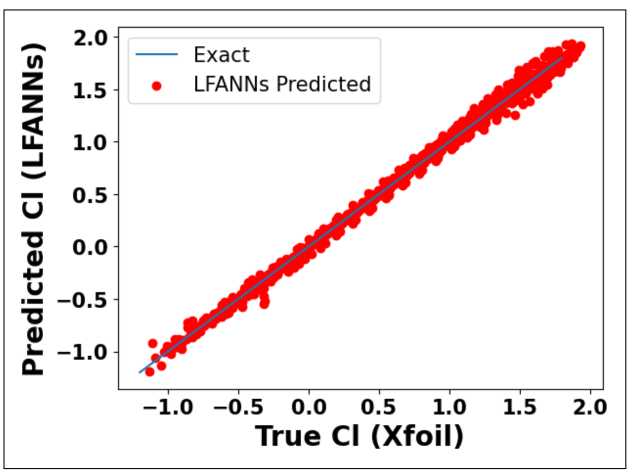


Figure 4: Accuracy of Low-Fidelity Artificial Neural Networks (LFANNs). Comparison of predicted lift coefficient from LFANNs with the true value (Xfoil). Mach number = 0.3, Reynolds number = 2 Million

that of LFANNs. The loss function for HFANNs is given by Equation 8, where P represents high-fidelity performance polar. Equation 9 shows the mapping of airfoil CST coefficients to performance polar, where O_{High} represents the output from HFANNs, $func_{HFANN}$ represents the approximate function from trained HFANNs and CST represents airfoil CST coefficients.

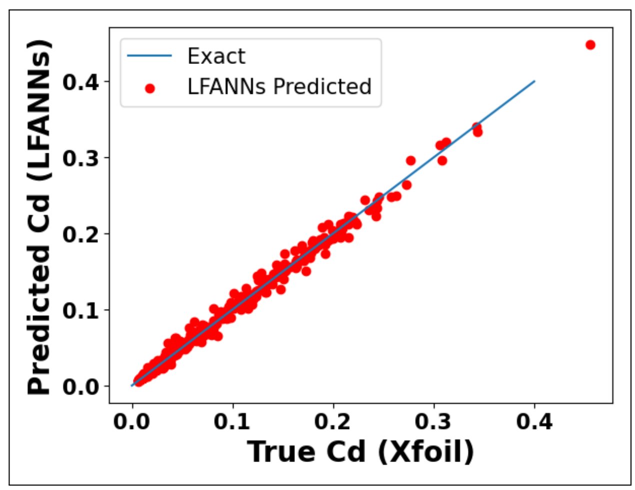


Figure 5: Accuracy of Low-Fidelity Artificial Neural Networks (LFANNs). Comparison of predicted drag coefficient from LFANNs with the true value (Xfoil). Mach number = 0.3, Reynolds number = 2 Million

$$J_{HFANN} = \frac{1}{\#\text{Train}} \sum_{\text{Train}} \left(\hat{\text{P}}_{\text{True}} - \hat{\text{P}}_{\text{Predicted}}\right)^2 \tag{8}$$

$$O_{High} = func_{HFANN}(CST) \tag{9}$$

Figure 7 presents a comparison of predicted C_l (from HFANNs) and true C_l (HAM2D) for the test data which is not a part of the training. The comparison is provided for Mach 0.3, and all other Mach numbers, all three polar show

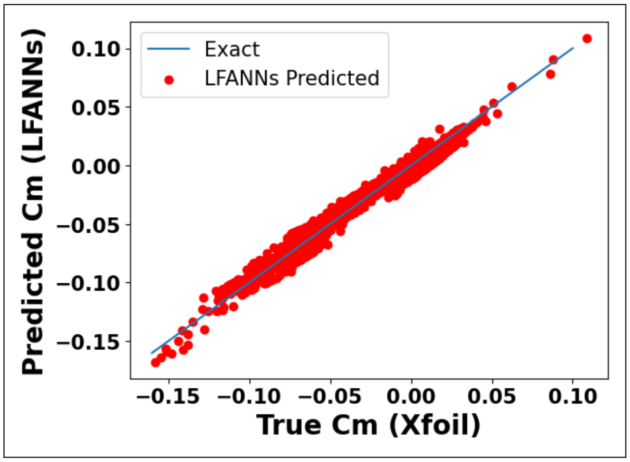


Figure 6: Accuracy of Low-Fidelity Artificial Neural Networks (LFANNs). Comparison of predicted pitching moment coefficient from LFANNs with the true value (Xfoil). Mach number = 0.3, Reynolds number = 2 Million

similar trends. Similarly, Figures 8 and 9 show a comparison of predicted drag and pitching moment coefficients (from HFANNs) with HAM2D data, and it can be seen that the accuracy of predicted coefficients from HFANNs has similar accuracy as that of LFANNs. This can also be inferred from Figure 10 (combined comparison of Xfoil and CFD). In Figure 10, the results from C_d comparison from both LFANNs and HFANNs are combined. The predicted C_d from LFANNs is compared with Xfoil data on the x-axis and the predicted C_d from HFANNs is compared with HAM2D on the x-axis. The accuracy and spread of data for both LFANNs predicted C_d and HFANNs predicted C_d are of similar order.

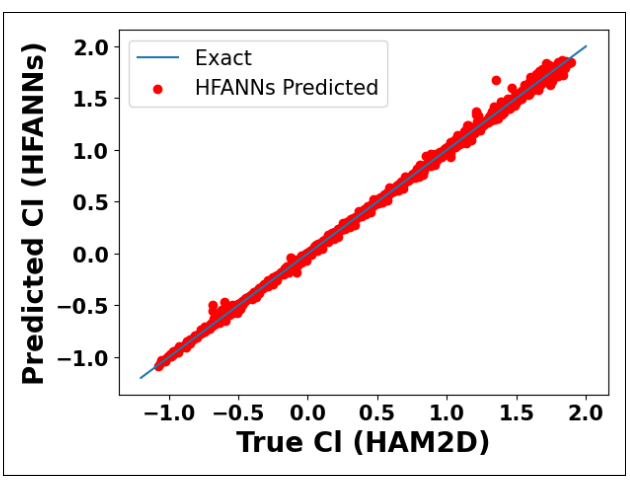


Figure 7: Accuracy of High-Fidelity Artificial Neural Networks (HFANNs). Comparison of predicted lift coefficient from HFANNs with the true value (HAM2D). Mach number = 0.3, Reynolds number = 2 Million

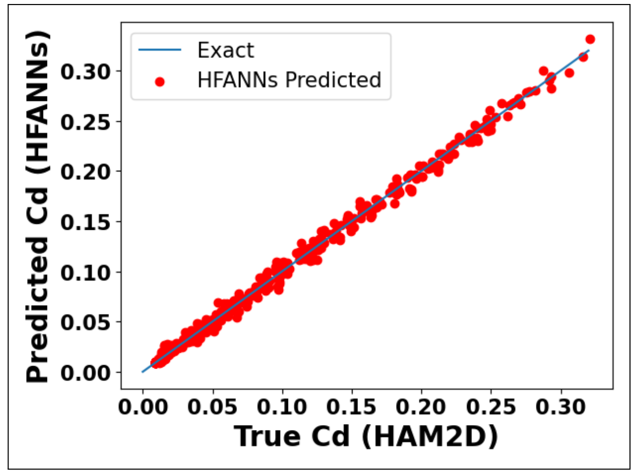


Figure 8: Accuracy of High-Fidelity Artificial Neural Networks (HFANNs). Comparison of predicted drag coefficient from HFANNs with the true value (HAM2D). Mach number = 0.3, Reynolds number = 2 Million

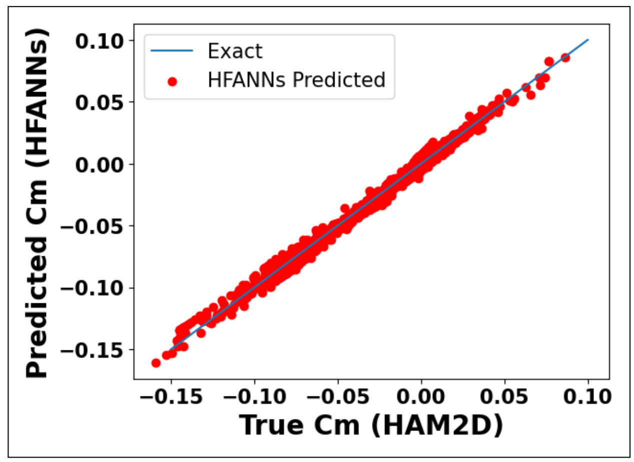


Figure 9: Accuracy of High-Fidelity Artificial Neural Networks (HFANNs). Comparison of predicted pitching moment coefficient from HFANNs with the true value (HAM2D). Mach number = 0.3, Reynolds number = 2 Million

MUTLI-FIDELITY ARTIFICIAL NEURAL NETWORKS (MFANNS)

This section presents a detailed explanation of the need for multi-fidelity artificial neural networks (MFANNs) and the implementation of the current methodology using both low-fidelity (Xfoil) and high-fidelity (HAM2D) data.

Need for Multi-Fidelity Architecture

The section above explained both LFANNs and HFANNs architecture for the same training data. From Figure 10, it was inferred that both LFANNs and HFANNs have the same order of accuracy. But it is to be emphasized that the predicted neural network values for both LFANNs and HFANNs (in Figures 5 and 8) are compared with the corresponding true values. i.e., Xfoil data from LFANNs and HAM2D data for HFANNs. As HAM2D is validated for a range of airfoil

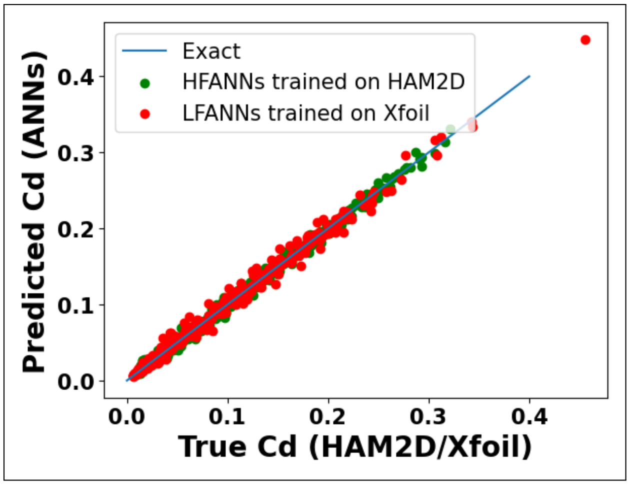


Figure 10: Accuracy of Low-Fidelity and High-Fidelity Artificial Neural Networks. Comparison of predicted drag coefficient from LFANNs with Xfoil and HFANNs with HAM2D. Mach number = 0.3, Reynolds number = 2 Million

geometries and freestream conditions [18], a comparison of Xfoil with HAM2D for the baseline airfoils is also presented in this paper to evaluate the deviation of Xfoil from HAM2D.

Figure 11 shows the comparison of the lift curve for the baseline airfoils using Xfoil and HAM2D simulations at Mach 0.3. We see a slight deviation of Xfoil data from HAM2D especially in the non-linear leg of the lift curve. But Xfoil overall captures the trend of the HAM2D data. Unlike the lift curve, the drag curve is significantly different from Xfoil as compared to HAM2D as shown in Figure 12. Xfoil underpredicts C_d for all the airfoil geometries for all values of angles of attack. The Xfoil prediction isn't accurate in the linear-leg and worsens in the near-stall and post-stall regions. A similar comparison of L/D is also presented in Figure 13.

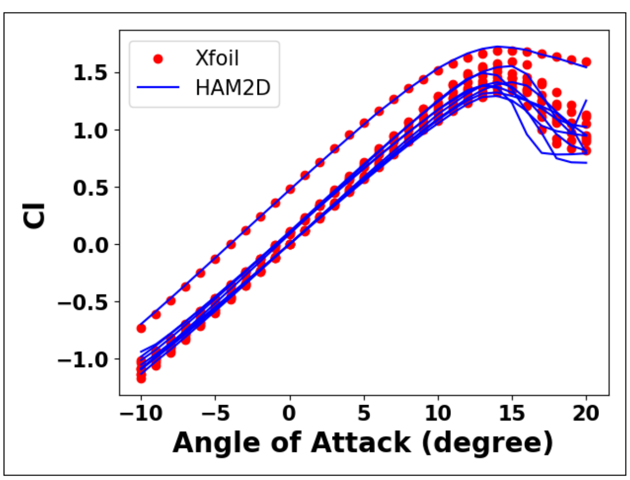


Figure 11: Comparison of lift curve of baseline airfoils generated from Xfoil and HAM2D. Mach number = 0.3, Reynolds number = 2 Million

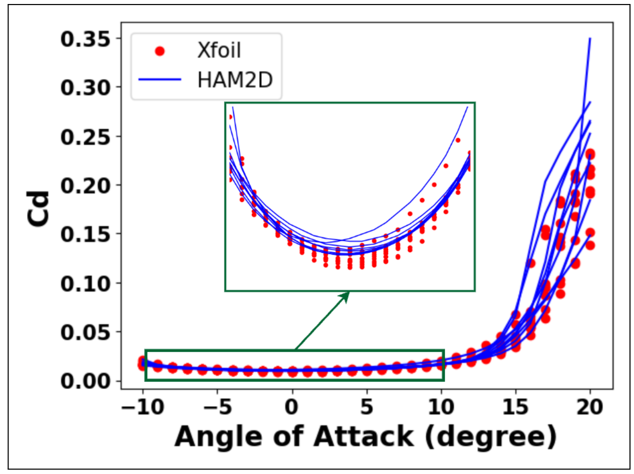


Figure 12: Comparison of drag curve of baseline airfoils generated from Xfoil and HAM2D. Mach number = 0.3, Reynolds number = 2 Million

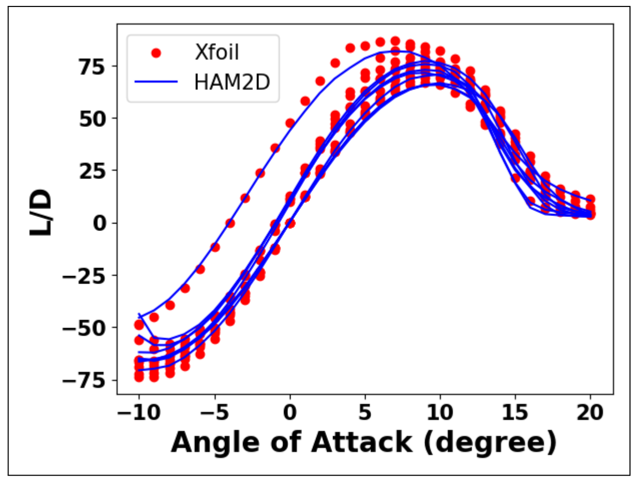


Figure 13: Comparison of lift-to-drag ratio of baseline airfoils generated from Xfoil and HAM2D. Mach number = 0.3, Reynolds number = 2 Million

A comparison of L/D is also presented in Figure 14 for SC1095 airfoil at Mach 0.3 for both Xfoil and HAM2D simulations. As expected, Xfoil overpredicts L/D for all angles of attack (as C_d is underpredicted by Xfoil). The maximum L/D for SC1095 airfoil as given by HAM2D is close to 75 whereas Xfoil predicts maximum L/D to be greater than 78. The angle of attack at maximum L/D also changes significantly for Xfoil as shown in Figure 14. From these studies, the following can be concluded.

1. Xfoil is significantly faster than HAM2D and predicts C_l with reasonable accuracy.
2. The C_d prediction by Xfoil is significantly different from HAM2D simulations and almost all the cases are underpredicted by Xfoil.
3. The comparison of L/D for Xfoil and HAM2D is significantly different.

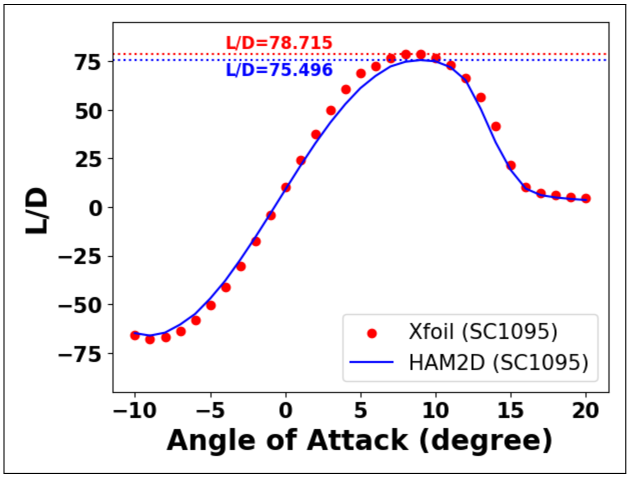


Figure 14: Comparison of lift-to-drag ratio of SC1095 airfoil generated from Xfoil and HAM2D. Mach number = 0.3, Reynolds number = 2 Million

4. Xfoil is orders of magnitude faster than HAM2D but has poor accuracy.

From the above conclusions, we can infer that Xfoil is faster but has poor accuracy and HAM2D is very accurate but is computationally expensive. Therefore, there is a need for a neural network architecture which uses a lot of data from Xfoil (computationally cheaper) and less data from HAM2D (accurate) such that the predicted performance polar from the neural network is close to HAM2D.

Predictor-Corrector Multi-fidelity Neural Network Architecture

In this section, we present a novel way of combining data from low-fidelity (Xfoil) and high-fidelity (HAM2D) codes and use them to train a multi-fidelity (combination of low and high fidelity data) neural networks to predict aerodynamic quantities of interest (C_l, C_d and C_m).

Figure 15 presents the overall working of MFANNs architecture. As shown in Figure 15, there are two separate artificial neural networks (ANNs) in this framework. These two ANNs act as predictor and corrector algorithms. The first network, highlighted in red is low-fidelity artificial neural networks (LFANNs) trained using Xfoil data for 560 airfoil geometries. Since Xfoil predictions aren't very accurate, a corrector ANNs is used in series with the predictor. MFANNs is used as a corrector network which takes airfoil geometry and performance polar predicted from LFANNs as inputs and gives polar closer to HAM2D as output. MFANNs is highlighted in green in Figure 15. LFANNs is pre-trained and frozen such that the weights of LFANNs don't change while training MFANNs. The loss function of MFANNs is explained by Equation 10, where P represents high-fidelity polar. Equation 11 shows the mapping of airfoil CST coefficients to performance polar, where O_{MFANN} represents the output from MFANNs, $func_{corrector}$ represents the corrector function from trained MFANNs. It is to be emphasized that $func_{LFANN}$ is the predictor function. The hypothesis behind a predictor-corrector architecture is that XFOIL predicts the performance polar with some reasonable accuracy, which is done by LFANNs. The corrector part of the MFANNs architecture then maps the XFOIL predicted polar to HAM2D predicted polar, and this requires a lesser number of training airfoils from high-fidelity code.

$$J_{MFANN} = \frac{1}{\#\text{Train}} \sum_{\text{Train}} (\hat{\text{P}}_{\text{True}} - \hat{\text{P}}_{\text{Predicted}})^2 \tag{10}$$

$$O_{MFANN} = func_{corrector}(CST + func_{LFANN}(CST)) \tag{11}$$

To validate the hypothesis, a comprehensive study on the number of airfoil geometries required for MFANNs is also performed. From the existing 560 airfoil geometries, a reduced number of high-fidelity cases are randomly chosen to train MFANNs. The objective of this exercise is to determine the least number of airfoils required to train MFANNs without compromising the accuracy of the model. Therefore, 1/2nd, 1/3rd, 1/4th, 1/5th, 1/6th, 1/10th, 1/15th and 1/20 cases are randomly chosen to train the MFANNs from the pool of 560 airfoil geometries at each Mach number. The comparison of predicted drag polar for all these cases is presented in Figure 16. As seen in Figure 16, the accuracy of all the cases looks similar. To quantitatively illustrate the number of reduced cases required for MFANNs, the root mean square error of C_d for all 8 cases is compared in Table 2. As can be seen from Table 2, the accuracy decreases with a reduction in the number of cases. The RMSE error of minimum C_d for 1/15th and 1/20th cases increases significantly as shown in the table, and therefore, 1/10th of high-fidelity (HAM2D) cases is considered optimal, which are used to train the MFANNs for all further studies.

The envelope of airfoil geometries used to train LFANNs is presented in Figure 17 which has 560 perturbation airfoils. The envelope of one-tenth airfoil geometries used to train MFANNs is also presented in Figure 18. From these Figures, it can be inferred that the number of cases as well as the envelope of airfoil geometries for training multi-fidelity architecture is much smaller than the total number of cases used to train LFANNs, which is essential to reduce the computational cost of data generation using high-fidelity code (HAM2D). Similarly, the envelope of C_l, C_d and C_m for training the LFANNs and MFANNs (with one-tenth cases) are presented in Figures 19, 20 and 21.

The comparison of predicted C_d from MFANNs for Mach number of 0.7 is presented in Figure 22. From this figure, it can be seen that MFANNs overall capture the drag polar very accurately even for higher Mach number. This is particularly interesting because many cases are transonic for Mach

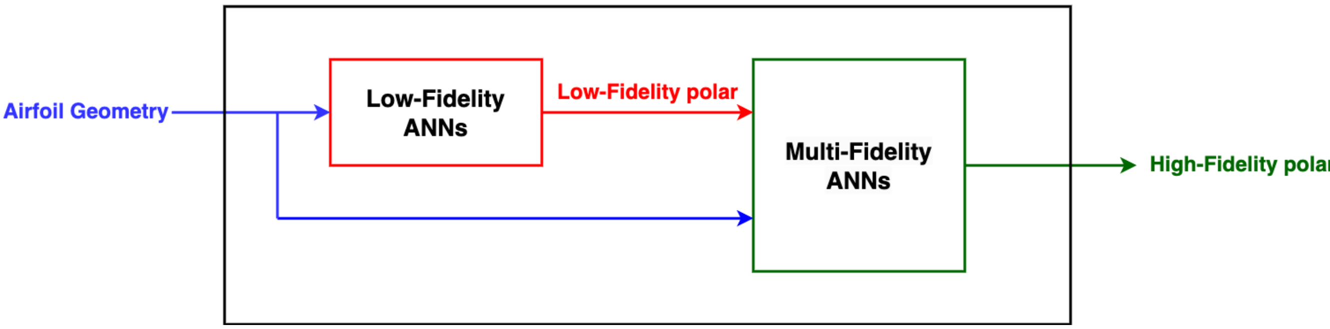


Figure 15: Architecture of Multi-Fidelity Artificial Neural Network (MFANNs). Inputs: Airfoil geometry and low-fidelity (Xfoil) polar; Outputs: High-fidelity (HAM2D) polar

Table 2: Comparison of accuracy of MFANNs for varying numbers of high-fidelity cases

Number of High-Fidelity Cases per Mach number	RMSE of C_d	RMSE for minimum C_d in drag counts
17360 (Total)	1.57×10^{-3}	0.441
8680 (1/2nd)	2.41×10^{-3}	0.532
5797 (1/3rd)	2.78×10^{-3}	0.711
4340 (1/4th)	2.89×10^{-3}	0.944
3720 (1/5th)	3.21×10^{-3}	1.910
2790 (1/6th)	3.35×10^{-3}	2.212
1736 (1/10th)	3.56×10^{-3}	2.184
1157 (1/15th)	5.65×10^{-3}	3.672
868 (1/20th)	6.41×10^{-3}	10.567

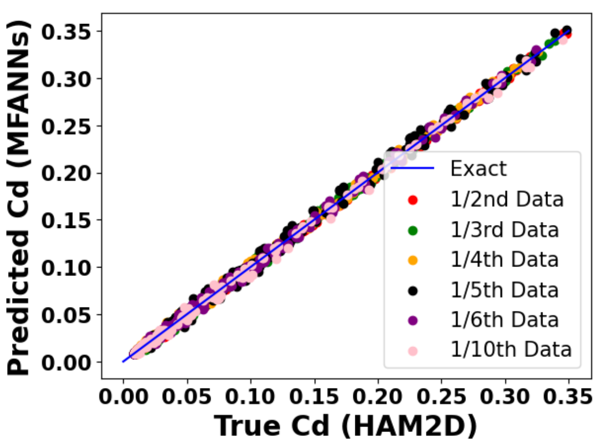


Figure 16: Comparison of accuracy of Multi-Fidelity Artificial Neural Networks (MFANNs) for varying number of high-fidelity cases. Comparison of predicted drag coefficient from MFANNs with the true value (HAM2D). Mach number = 0.5, Reynolds number = 3.333 Million

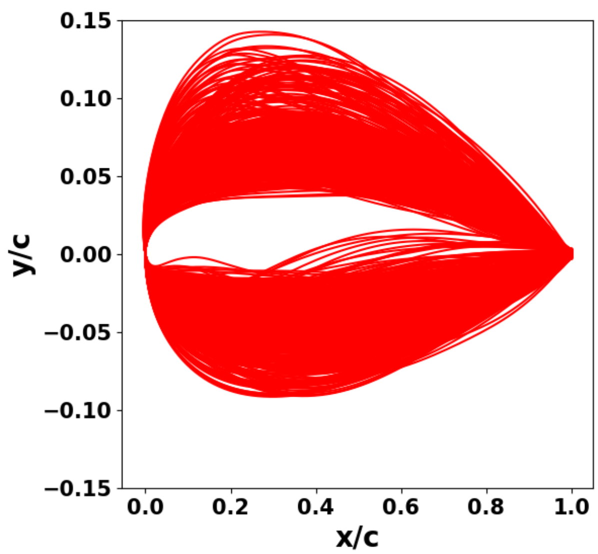


Figure 17: Visualization of the spread of airfoil geometries for 560 airfoils used to train low-fidelity artificial neural networks (LFANNs)

0.7 and the use of predictor-corrector type multi-fidelity architecture ensures that the model is accurate even for higher Mach numbers. The predictions for lower Mach numbers show a similar trend. Figure 23 shows a comparison of the predicted drag coefficient from HFANNs with 1/10th cases (same as MFANNs) and the RMSE of C_d is more than twice than that of MFANNs (Table 2). It is inferred from Figure 23 that 1/10th high-fidelity cases aren't good enough to train the HFANNs and have a much higher mean error as well as a wider spread of error. This exercise re-validates our hypothesis of using a predictor-corrector-based MFANNs architecture to reduce the number of high-fidelity cases without compromising the accuracy of the model. The

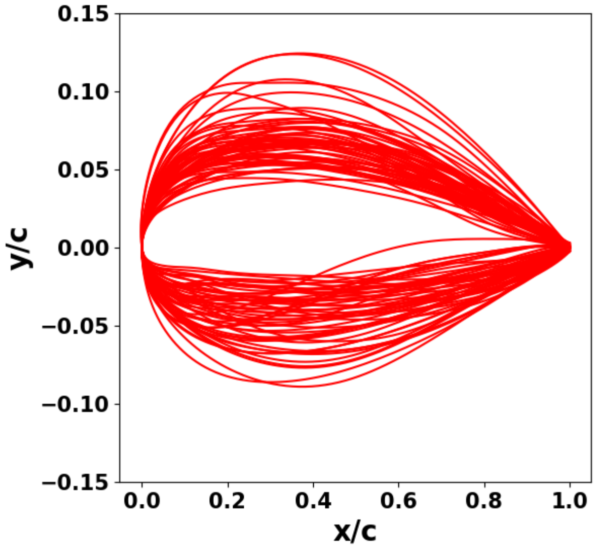


Figure 18: Visualization of the spread of airfoil geometries for 56 airfoils (1/10 of 560) used to train multi-fidelity artificial neural networks (MFANNs)

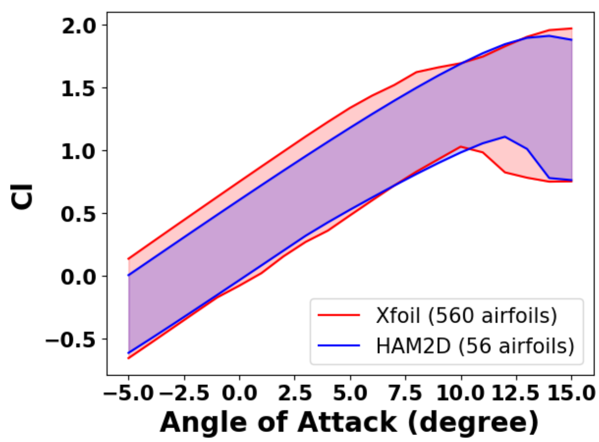


Figure 19: Visualization of variation in the envelope of lift coefficient. Red: 560 airfoils simulated with Xfoil, Blue 56 airfoils simulated with HAM2D

comparison of performance polar for baseline SC1095 airfoil is also presented in Figures 24, 25 and 26, which shows the accuracy of MFANNs architecture for standard rotorcraft airfoil. Some observations from the MFANNs architecture are listed below:

1. A predictor-corrector type multi-fidelity framework works efficiently and accurately to predict airfoil performance polar with significantly lesser high-fidelity training data (1/10th).

2. Both LFANNs and HFANNs have the same order of accuracy as that of MFANNs.

3. Xfoil predicts C_l with reasonable accuracy but the prediction of C_d has a higher error and therefore requires a correction.

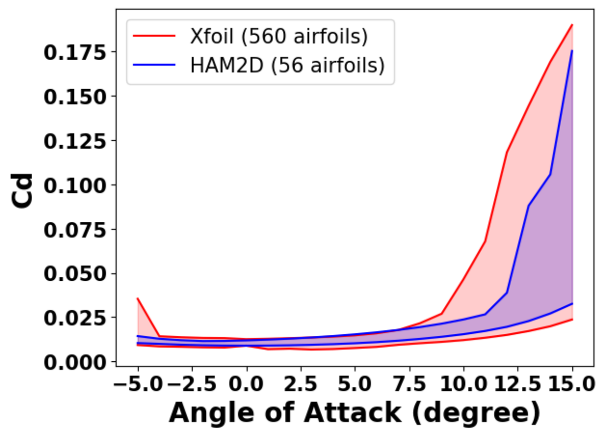


Figure 20: Visualization of variation in the envelope of drag coefficient. Red: 560 airfoils simulated with Xfoil, Blue 56 airfoils simulated with HAM2D

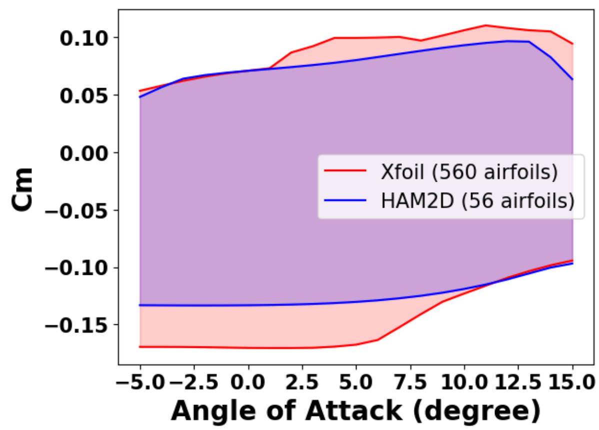


Figure 21: Visualization of variation in the envelope of pitching moment coefficient. Red: 560 airfoils simulated with Xfoil, Blue 56 airfoils simulated with HAM2D

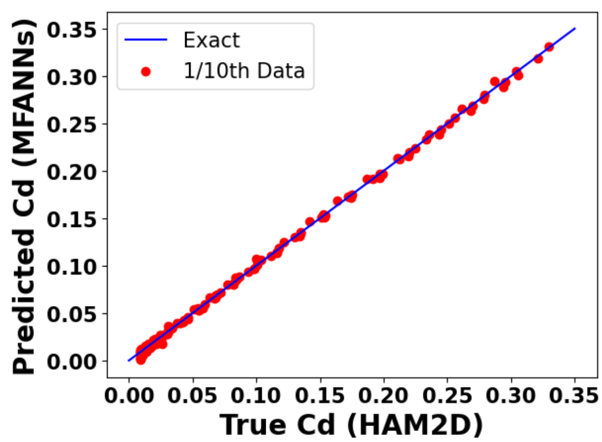


Figure 22: Comparison of accuracy of Multi-Fidelity Artificial Neural Networks (MFANNs) with 1/10th high-fidelity cases. Comparison of predicted drag coefficient from MFANNs with the true value (HAM2D). Mach number = 0.7, Reynolds number = 4.667 Million

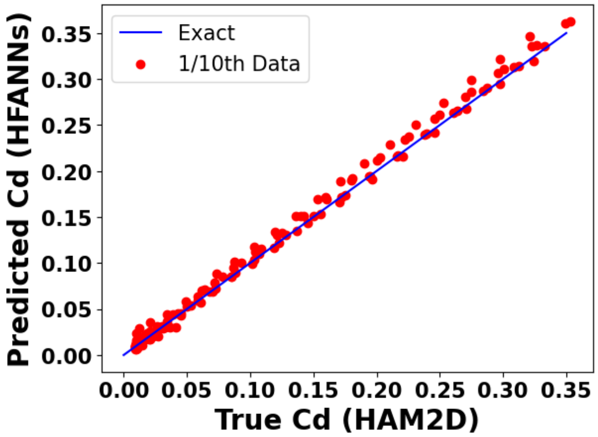


Figure 23: Comparison of accuracy of High-Fidelity Artificial Neural Networks (HFANNs) with 1/10th high-fidelity cases. Comparison of predicted drag coefficient from HFANNs with the true value (HAM2D). Mach number = 0.7, Reynolds number = 4.667 Million

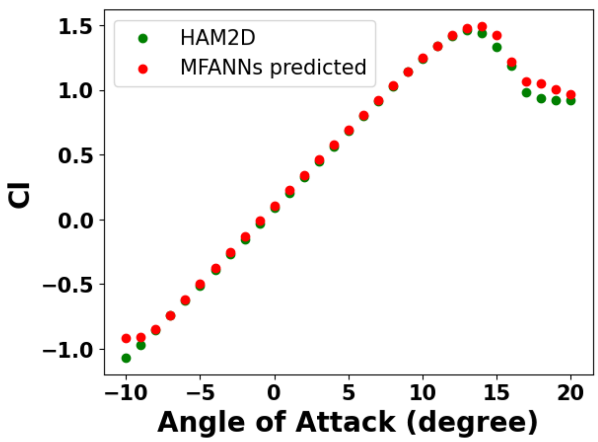


Figure 24: Comparison of predicted lift coefficient for SC1095 airfoil from MFANNs with the true value (HAM2D). Mach number = 0.3, Reynolds number = 2 Million

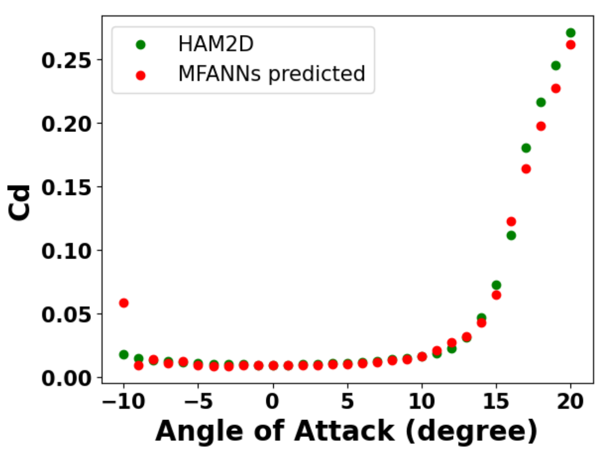


Figure 25: Comparison of predicted drag coefficient for SC1095 airfoil from MFANNs with the true value (HAM2D). Mach number = 0.3, Reynolds number = 2 Million

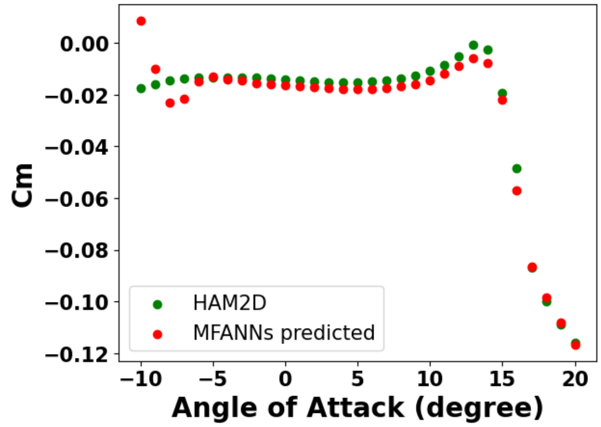


Figure 26: Comparison of predicted pitching moment coefficient for SC1095 airfoil from MFANNs with the true value (HAM2D). Mach number = 0.3, Reynolds number = 2 Million

Sensitivity Analysis of Multi-fidelity Neural Network Architecture

As explained in the section above, 1/10th of the high-fidelity data is used for training multi-fidelity neural networks (MFANNs). In this section, a comprehensive study of the number of high-fidelity cases and the corresponding error distribution of predicted drag coefficient is presented. As shown in Figure 16, the accuracy of MFANNs with 1/10th high-fidelity case is the same as that of all other cases (1/2nd to 1/6th), and therefore, for all further studies, 1/10th high-fidelity cases were used to train MFANNs. The root mean square error (RMSE) for all 6 cases is presented in Table 2 which also demonstrates the usability of 1/10th high-fidelity cases for MFANNs training. A comparison of the mean error of C_d for these 6 cases is presented in Figure 27. As expected, the maximum error is in the extreme angles of attack (both positive and negative). The error is of the same order for all the cases but increases as the number of high-fidelity cases is reduced. Figure 28 also presents a comparison of mean error in C_d for non-vectorized implementation. A non-vectorized implementation corresponds to the neural networks architecture where angle of attack is taken as input and single values C_l, C_d and C_m (at each angles of attack) are used as outputs instead of the entire performance polar. As seen in Figure 28, the error distribution for non-vectorized implementation is highly non-uniform even in the linear leg. The magnitude of error is also higher than the vectorized implementation. This also validates the use of vectorized implementation for MFANNs. Similarly, Figure 29 presents a comparison of the envelope of mean +/- standard deviation for three cases (1/2nd, 1/5th and 1/10th). It can be seen from Figure 29 that the envelope of 1/5th and 1/10th cases is similar with the same area under the envelope. This also demonstrates that the reduction of the number of high-fidelity cases from 1/5th to 1/10th doesn't impact the accuracy of MFANNs.

The 1/10th high-fidelity cases to train the MFANNs are

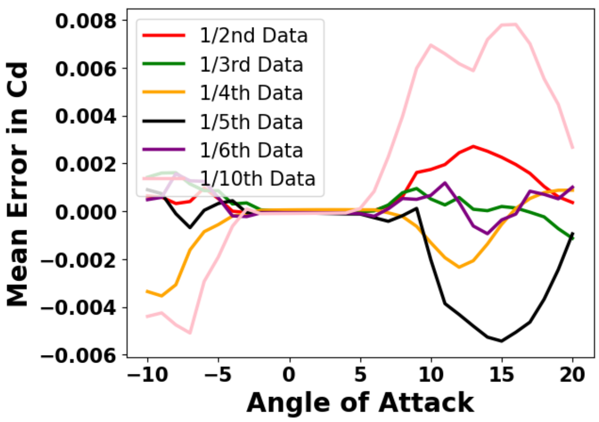


Figure 27: Mean Error distribution of predicted drag coefficients with MFANNs for a varying number of training data with vectorized implementation. Mach number = 0.5, Reynolds number = 3.333 Million

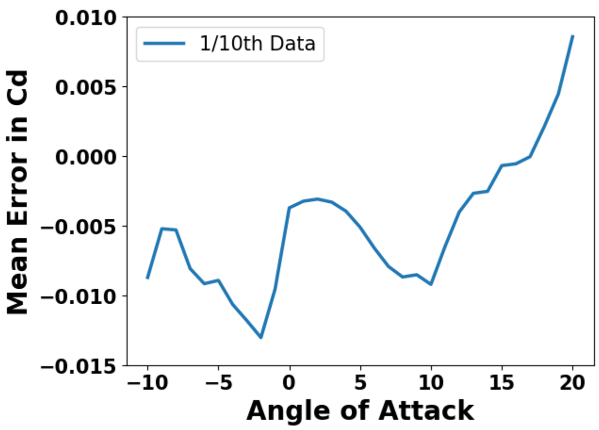


Figure 28: Mean Error distribution of predicted drag coefficients with MFANNs for one-tenth training data with non-vectorized implementation. Mach number = 0.5, Reynolds number = 3.333 Million

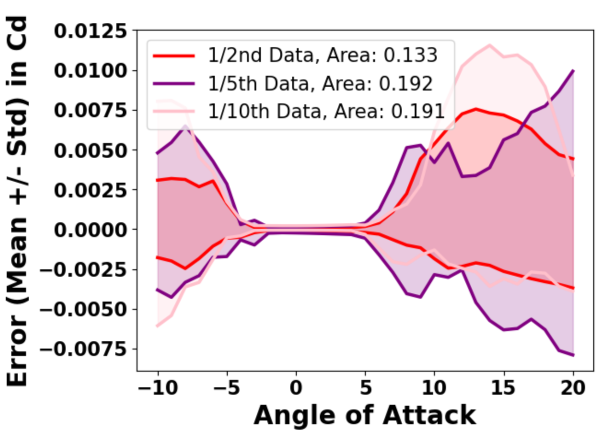


Figure 29: Envelope of Mean Error and standard deviation distribution of predicted drag coefficients with MFANNs for a varying number of training data with vectorized implementation. Mach number = 0.5, Reynolds number = 3.333 Million

chosen randomly from the pool of 560 airfoil geometries. To verify that there is no bias while selecting the high-fidelity cases from 560 airfoil geometries, 5 random seeds are used to randomly select 1/10th high-fidelity cases. The MFANNs is trained for all these cases separately and the predicted drag coefficient is compared with the true drag coefficient (HAM2D) in Figure 30. As seen in Figure 30, 5 samples represent 5 different random selections of 1/10th high-fidelity cases from 560 airfoils. All 5 samples have the same accuracy and similar scatter. The comparison is shown from Mach 0.5, but the results are similar for all other Mach numbers.

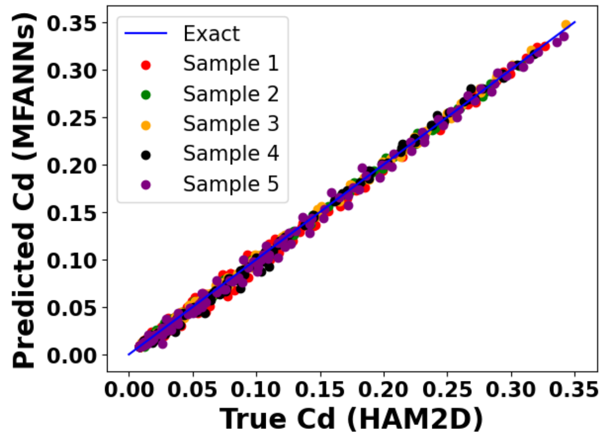


Figure 30: Comparison of accuracy of Multi-Fidelity Artificial Neural Networks (MFANN) for one-tenth training data with multiple combinations of randomly selected data. Comparison of predicted drag coefficient from MFANNs with the true value (HAM2D). Mach number = 0.5, Reynolds number = 3.333 Million

DATA PREPROCESSING AND NEURAL NETWORK TRAINING

Data preprocessing is an important step before training the neural networks. For LFANNs, airfoil CST coefficients are inputs and low-fidelity (Xfoil) performance polar are outputs, and similarly, for MFANNs, airfoil CST coefficients and low-fidelity performance polar are inputs and high-fidelity (HAM2D) performance polar are outputs. Therefore, there are three sets of data, i.e., airfoil CST coefficients, low-fidelity performance polar and high-fidelity performance polar.

As discussed in the airfoil parametrization section, the airfoil is parametrized using the Chebyshev polynomials as shape functions which are near-orthogonal. Therefore, airfoil CST coefficients are only scaled using mean and standard deviation without having to perform Proper Orthogonal Decomposition (POD). Unlike airfoil CST coefficients, the performance polar (both low-fidelity and high-fidelity) are non-orthogonal and are transformed using POD after scaling. The dimension of transformed performance polar is reduced ensuring that 99 % of the explained variance is retained.

After reducing the dimension, C_l is transformed from 31 dimensions to a 10-dimensional vector, C_d from 31 to an 18-dimensional vector and C_m from 31 to an 8-dimensional vector. The transformed CST coefficients and performance polar are used to train both LFANNs and MFANNs, and the loss function is defined using the transformed polar to minimize bias.

Both neural networks are trained using Keras which is a high-level API of TensorFlow (Ref. [21]). The hyperparameters of both LFANNs and MFANNs are optimized using a grid-search algorithm. The hyperbolic tangent activation function is used in all neurons. The data is split into training (54 %), test (30 %) and validation sets (16 %). Both networks (LFANNs and MFANNs) are trained to 5000 epochs and convergence of loss is monitored to ensure that there is no overfitting or underfitting. The comparison of loss with epochs is presented in Figures 31 and 32. As seen in Figures 31 and 32, the validation loss is very close to the training loss and there is no overfitting. Standard L2 regularization is also implemented to minimize overfitting, the details of which is explained in Equation 12. The details of the number of neurons and hidden layers are mentioned below:

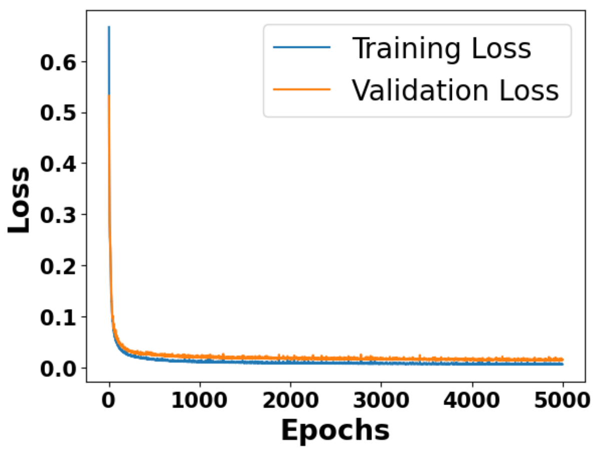


Figure 31: Comparison of convergence of loss for both training and validation data for Multi-Fidelity Artificial Neural Networks (MFANN)

$$Regularized \quad Loss = Original \quad Loss + \lambda \quad \times \quad \sum_{\#n} w^2 \tag{12}$$

1. **LFANNs:** 20(Inputs)-20-30-50-70-36(Outputs)
2. **MFANNs:** 56(Inputs)-20-20-20-36(Outputs)

AIRFOIL OPTIMIZATION USING GENETIC ALGORITHM WITH MFANNS

The MFANNs architecture can be used as an approximate function evaluation and can be combined with an optimization algorithm to optimize airfoil geometry for a given target performance polar. In this section, we use Genetic Algorithms (GA) [22] with multi-fidelity artificial neural networks (MFANNs) at Mach 0.3 to optimize airfoil geometry to improve lift-to-drag ratio (L/D) of the airfoil without changing the lift coefficient.

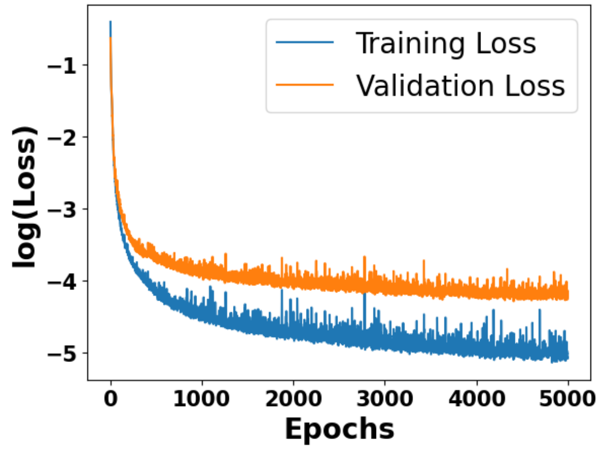


Figure 32: Comparison of convergence of log(loss) for both training and validation data for Multi-Fidelity Artificial Neural Networks (MFANN)

Figure 33 shows the steps involved in the optimization process using GA, an evolutionary global optimization algorithm inspired by the process of natural selection. The input design variables are evaluated for a defined objective function and these design variables iteratively evolve through selection, crossover and mutation to explore the solution space. The best-performing design survives in each generation to find a global optimum in the design space.

Equation describes a least squares objective function which is used to arrive at the best-performing airfoil geometry using GA and MFANNs. It is again emphasized that the pre-trained MFANNs architecture is used as a function evaluation in the optimization loop. The values of (λ_i) are used to tune the relative importance of each parameter and remove any bias by scaling all three parameters (C_l, C_d and C_m) to be in the same order of magnitude. There are 20 design variables (airfoil CST coefficients). For each of them, GA arrives at an optimal airfoil with an initial randomly distributed population of size 32, continuous random variables for crossover and a 10% mutation rate to ensure diversity in the design space. A total of 500 generations are performed and the best-performing design is chosen at the end of the last generation.

$$J_{\text{GA}} = \lambda_1 \left\| c_l - c_{l,\text{target}} \right\|_2^2 + \lambda_2 \left\| c_d - c_{d,\text{target}} \right\|_2^2 + \lambda_3 \left\| c_m - c_{m,\text{target}} \right\|_2^2$$

The objective of the airfoil optimization problem is to improve the L/D of SC1095 airfoil by 5% for all angles of attack without changing the lift and moment polar. The maximum thickness of the airfoil is allowed to vary between 8.5% - 10.5%. The GA algorithm explained above is used with MFANNs as an evaluation function to optimize the airfoil. Figure 34 shows

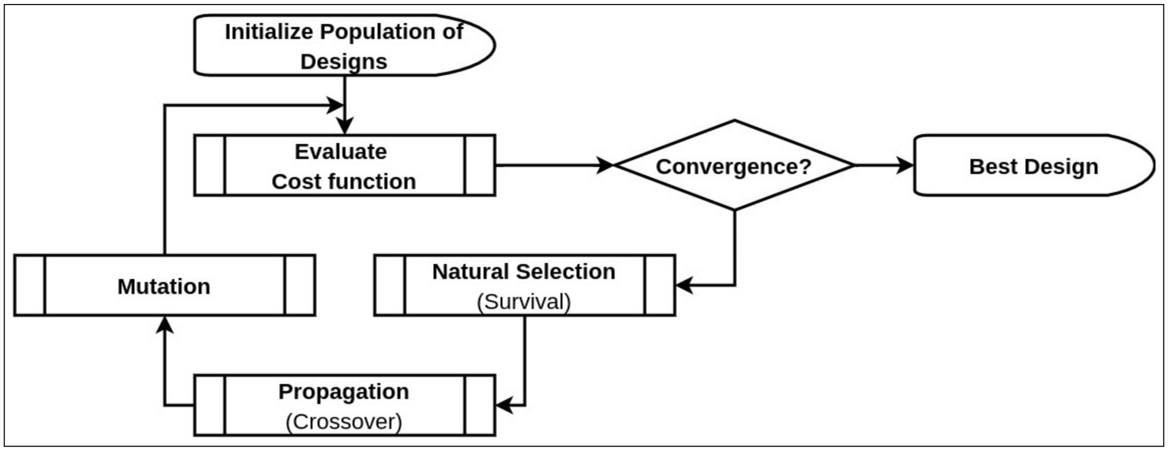


Figure 33: Genetic algorithms methodology and sequence of steps

the comparison of optimized airfoil geometry and the corresponding L/D polar. As seen in Figure 34a, the optimized airfoil becomes thicker (t/c = 10.3%) but remains within the defined thickness bound. There is an increase of maximum L/D by 4% than the baseline as seen in Figure 34b. This optimization problem is a sample test case to demonstrate the applicability of MFANNs for solving the airfoil optimization problem in addition to the airfoil polar prediction.

COMPUTATIONAL COST

The main objective of using a multi-fidelity framework is to reduce the computational cost of data generation without compromising on accuracy. The details of the computational cost associated with generating both high and low-fidelity data at multiple Mach numbers and training both LFANNs and MFANNs are presented in Table 3. It can be seen from Table 3 that the total time taken to generate low-fidelity performance polar for 560 airfoils at 31 angles of attack and 5 Mach number (86,800 cases) is close to 15 hours on a single core machine, which is less than 1 second per case. Unlike Xfoil, it takes HAM2D around 338,000 core hours to generate the training data for 560 airfoils at 5 Mach number and 31 angles of attack. This is a significantly larger number and requires a lot of computational resources, and can sometimes become impractical. Using MFANNs reduces the computational cost by 1/10th for generating high-fidelity (HAM2D) performance polar and it takes about 33,800 core hours to generate high-fidelity (HAM2D) data for training MFANNs. Once training data is generated, it takes about 12 minutes to train LFANNs and 1.2 minutes to train MFANNs.

CONCLUSION AND FUTURE WORKS

Conclusions

The current study provides a novel multi-fidelity artificial neural network architecture which works like a predictor-corrector algorithm to map airfoil geometry with performance polar. A neural network trained on low-fidelity (Xfoil) data is treated as a predictor which gives an initial prediction of performance polar. This network is trained on a lot of low-fidelity data which are computationally cheaper to generate (4 orders cheaper than high-fidelity (HAM2D)). A multi-fidelity artificial neural network (MFANNs) is trained, which works like a corrector algorithm and maps low-fidelity data to high-fidelity (HAM2D) data with a much lesser number of training data (1/10th) without compromising on the accuracy of the model (RMSE of minimum C_d = 2.184 drag counts).

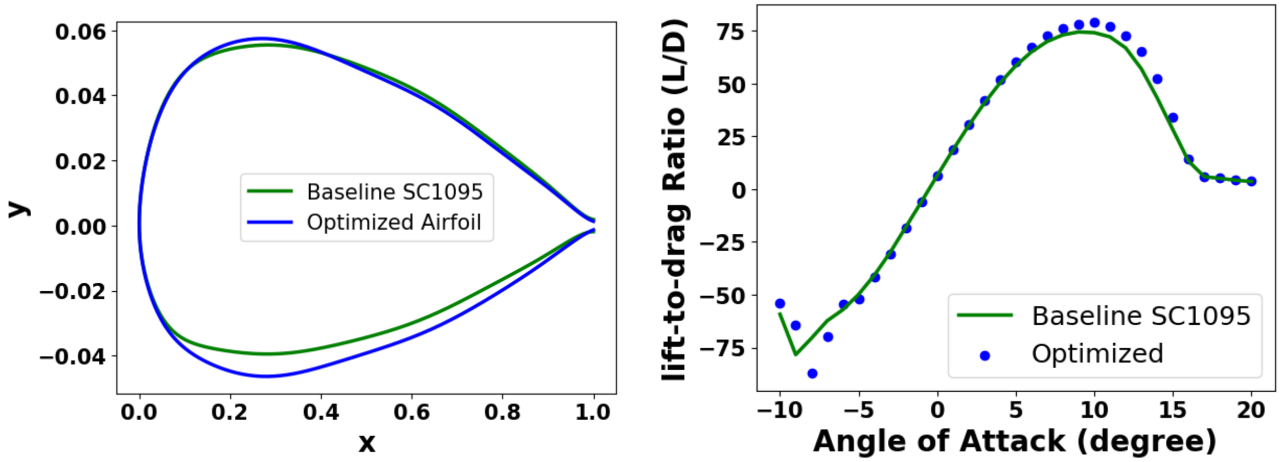
The current study also presents a comparison of the aerodynamic performance polar for low-fidelity (Xfoil) and high-fidelity (HAM2D) codes. This comparison shows the limitation of low-fidelity (Xfoil) code in predicting drag coefficient at all Mach and Reynolds numbers.

A detailed study on the number of high-fidelity (HAM2D) data required for training MFANNs is also presented. This allows using much lesser (1/10th) high fidelity (HAM2D) data without compromising on the accuracy of the model. The computational cost of generating data for the multi-fidelity network is reduced by one-tenth as compared to the high-fidelity trained neural networks.

The multi-fidelity artificial neural network (MFANNs) is combined with Genetic Algorithms (GA) to optimize SC1095 airfoil geometry to improve the lift-to-drag ration without changing lift and moment polar. This exercise demonstrates the applicability of MFANNs architecture for solving airfoil optimization problems.

Potential Extensions of the Current Study

The current multi-fidelity architecture can be extended to perform inverse design of airfoils. This can be a robust multi-fidelity inverse-design framework to design airfoils for a required performance polar. The inverse design architecture



(a) Airfoil geometry comparison

(b) L/D polar comparison

Figure 34: Comparison of optimized airfoil geometry (from GA + MFANNs) and the corresponding L/D polar with the baseline SC1095 airfoil.

Table 3: Comparison of computational cost for generating performance parameters using CFD, Neural Networks training and evaluation

Computational Exercise	Computational Cost (hours)
Xfoil Training Data Generation for LFANN	15
HAM2D Training Data Generation for HFANN	$338,000$
HAM2D Training Data Generation for MFANN	$33,800$
Train LFANN	0.20
Train MFANN	0.02
CFD Simulation for Single Airfoil at 1 Mach number	1.30
Evaluate Single Airfoil using ANNs	1.34×10^{-7} (milliseconds)

can then be integrated with Blade Element Momentum Theory (BEMT) code to optimize rotor blades.

The current multi-fidelity framework is only trained on fully turbulent cases. As a continuation of this study, transition cases and low Reynolds number cases can also be included. A detailed study on uncertainty quantification for predictor and corrector architecture can also be performed.

The current MFANNs architecture allows for the addition of higher fidelity in series, and therefore, some experimental data can also be included as corrector-2 in the MFANNs architecture. This will ensure that the output from MFANNs isn't only close to the high-fidelity (HAM2D) data, but also close to the experiments.

Author Contact: Apurva Anand *apurva01@umd.edu*

ACKNOWLEDGEMENTS

This research is conducted with support from the Vertical Lift Research Center of Excellence (VLRCOE) and Army Research Laboratory (ARL) under the technical monitoring of Dr. Mahendra Bhagwat and Dr. Phuriwat Anusonti-Inthra. The authors also acknowledge the University of Maryland super-computing resources, Zaratan HPC cluster, made available for conducting the research in this paper.

*

REFERENCES

[1] Glaws, A., King, R. N., Vijayakumar, G., and Ananthan, S. "Invertible neural networks for airfoil design". In: *AIAA Journal* 60.5 (2022), pp. 3035–3047.

[2] Jameson, A. "Re-engineering the design process through computation". In: *Journal of Aircraft* 36.1 (1999), pp. 36–50.

[3] Anand, A., MV Shesha S.and V, Chandrasekaran, VVS, Narayan, and Narayanarao, B. "Aerodynamic Interaction and Analysis of Nose-Mounted Propeller Aircraft". In: *AIAA AVIATION 2023 Forum*. 2023, p. 4080.

[4] Shalu, H., Govindarajan, B., Sridharan, A., and Singh, R. "Blade shape optimization of rotors using neural networks". In: *VFS Forum*. Vol. 79. 2023.

[5] Anand, A., Marepally, K., Lee, B., and Baeder, J. D. "Development and Application of Improved Tandem Neural Networks for Inverse Design of Rotorcraft Airfoil". In: *49th European Rotorcraft Forum* (2023).

[6] Anand, A., Marepally, K., Safdar, M. M., Lee, B., and Baeder, J. D. "Generalizable Deep Learning Module for Rotorcraft Inverse Design Applications". In: *AIAA SCITECH 2024 Forum*. 2024, p. 0460.

[7] Nagawkar, J. R., Leifsson, L. T., and He, P. "Aerodynamic shape optimization using gradient-enhanced multifidelity neural networks". In: *AIAA SciTech 2022 Forum*. 2022, p. 2350.

[8] Zhang, X., Xie F.and Ji, T., Zhu, Z., and Zheng, Y. "Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization". In: *Computer Methods in Applied Mechanics and Engineering* 373 (2021), p. 113485.

[9] Tao, G., Fan, C., Wang, W., Guo, W., and Cui, J. "Multi-fidelity deep learning for aerodynamic shape optimization using convolutional neural network". In: *Physics of Fluids* 36.5 (2024).

[10] He, L., Qian, W., Zhao, T., and Wang, Q. "Multi-fidelity aerodynamic data fusion with a deep neural network modeling method". In: *Entropy* 22.9 (2020), p. 1022.

[11] Tekaslan, H. E. and Nikbay, M. "A Multi-fidelity Prediction with Convolutional Neural Networks Using High-Dimensional Data". In: *AIAA Aviation 2022 Forum*. 2022, p. 3719.

[12] Drela, M. "XFOIL: An analysis and design system for low Reynolds number airfoils". In: *Low Reynolds Number Aerodynamics: Proceedings of the Conference Notre Dame, Indiana, USA, 5–7 June 1989*. Springer. 1989, pp. 1–12.

[13] Sitaraman, J. and Roget, B. "Solution algorithm for unstructured grids using quadrilateral subdivision and hamiltonian paths". In: *52nd Aerospace Sciences Meeting*. 2014, p. 0079.

[14] Kulfan, B. M. "Universal parametric geometry representation method". In: *Journal of aircraft* 45.1 (2008), pp. 142–158.

[15] Jiang, G. and Shu, C. "Efficient implementation of weighted ENO schemes". In: *Journal of computational physics* 126.1 (1996), pp. 202–228.

[16] Saad, Y. and Schultz, M. H. "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems". In: *SIAM Journal on scientific and statistical computing* 7.3 (1986), pp. 856–869.

[17] Spalart, P. and Allmaras, S. "A one-equation turbulence model for aerodynamic flows". In: *30th aerospace sciences meeting and exhibit*. 1992, p. 439.

[18] Jung, Y. S., Vijayakumar G.and Ananthan, S., and Baeder, J. D. "Local correlation-based transition models for high-Reynolds-number wind-turbine airfoils". In: *Wind Energy Science* 7.2 (2022), pp. 603–622.

[19] Anand, A., Marepally, K., and Baeder, J. D. "Deep Learning Framework for Design and Optimization of Rotor Blades". In: *VFS Forum*. Vol. 80. 2024.

[20] Costenoble, M., Baeder, J. D., and Jung, Y. S. "Automated mesh generation and solution analysis of arbitrary airfoil geometries". In: *Journal of Aircraft* 59.5 (2022), pp. 1120–1136.

[21] Chollet, François et al. *Keras*. https://keras.io. 2015.

[22] Haupt, R.L. and Haupt, S.E. *Practical Genetic Algorithms*. Wiley InterScience electronic collection. Wiley, 2004. ISBN: 9780471671756. URL: https://books.google.com/books?id=k0jFfsmbtZIC.