DEVELOPMENT AND APPLICATION OF IMPROVED TANDEM NEURAL NETWORKS FOR INVERSE DESIGN OF ROTORCRAFT AIRFOIL

Apurva Anand, Koushik Marepally, Bumseok Lee, James D Baeder

University of Maryland, College Park, MD, USA

Abstract

Designing an airfoil shape with specific performance characteristics is a fundamental problem in the field of rotorcraft blade design. Traditional aerodynamic design methodologies often involve iterative optimization of the shape using low-fidelity modeling techniques during the early design phase, owing to the demanding computational costs of high-fidelity adjoint-CFD based optimization. In this study, an efficient accurate approach using Deep Neural Networks for the inverse design of rotorcraft airfoils is investigated. We leverage the Tandem Neural Network (T-NN) architectures to design the airfoil for a required performance curves (lift, lift-to-drag ratio, and pitching moment) in a novel and efficient way. The T-NN architecture allows for a modified and flexible cost function making them highly efficient and accurate for the inverse design of airfoils. Three different improvements to the T-NN architecture are proposed in this work to allow for improved accuracy and better constraint handling capabilities, enabling a paradigm shift in the rotorcraft component design methodologies.

1 NOTATION

Symbols:

C : CST coefficients

C : Class function

S : Shape function

C_d : Drag Coefficient

C_l : Lift Coefficient

C_m : Pitching Moment Coefficient

P : Performance Polar

ψ : Normalized airfoil chord-wise coordinate

ζ : Normalized airfoil surface coordinate

$\mathcal{O}()$: Order of computational complexity

$\#()$: Cardinality operator (number of elements)

Acronyms:

ANN : Artificial Neural Network

BEMT : Blade Element Momentum Theory

CFD : Computational Fluid Dynamics

I-ANN : Inverted Artificial Neural Network

TE : Trailing Edge

T-NN : Tandem Neural Network

2 INTRODUCTION

The optimization of rotorcraft blades has been a focus of research for many years. Designing an airfoil shape with specific performance characteristics is a fundamental problem in the field of blade design which plays a crucial role in improving the performance of rotorcraft. Airfoil design is a complex optimization problem that involves multiple fields of study, such as aerodynamics, structures, and acoustics (Ref. [1]). Conventional design methods involve iterative optimization of the shape using either less accurate low-fidelity modelling techniques which don't take into account the intricate flow interactions, or adjoint-based CFD (Computational Fluid Dynamics), which is more accurate (Refs. [2, 3]) but demanding in terms of computational cost and storage (Ref. [4]) and they must be restarted for any change in the objective function. Additionally, the use of multi-disciplinary optimisation for full blade design across various operating conditions with adjoint-based CFD methods becomes impractical.

This work focuses on the development and implementation of Tandem Neural Networks (T-NNs) architecture (Ref. [5]) for the inverse design of rotorcraft airfoils. T-NNs is a new development (Ref. [6]) in the field of aerodynamic design. It is an extension of Artificial Neural Networks (ANNs) with a modified cost function. While the standard ANNs can be efficiently and accurately used to determine the load coefficients for given airfoil geometry (Refs. [1, 6, 7, 8]), they exhibit loss of accuracy when used for inverse design problems, due to inverse scattering. This problem arises due to the non-invertibility of the aerodynamics, i.e., similar load characteristics can be achieved by different airfoil geometries. T-NNs overcomes this limitation by defining the cost function such that the Neural Networks is not forced to predict the same airfoil geometry, rather it predicts an airfoil geometry which outputs the required load coefficients. This is also the novelty associated with this inverse design architecture, where the performance polar of an airfoil is fed as input and the T-NNs outputs the airfoil geometry. T-NNs architecture allows for the optimization of the airfoil geometries for the entire performance polar for a range of Mach and Reynolds numbers. One necessary requirement for airfoil design is the ability to use multiple design constraints (maximum thickness, trailing edge thickness, stall margin, etc.) during optimization. T-NNs architecture makes the inverse design process flexible enough to include multiple design constraints which makes the inverse design process more practical for designers. The details of these constraints are explained in relevant sections.

The standard T-NNs architecture is deterministic, and therefore, for a given design objective, only one airfoil shape is generated. This paper also focuses on statistical analysis with the T-NNs module that can predict a family of airfoil geometries for a given design objective. The details of statistical analysis along with the implementation of optimized airfoil for hover performance computation using an in-house developed BEMT (Blade Element Momentum Theory) code is also included in the paper.

3 METHODOLOGY

This section sequentially describes the methodology employed in generating a representative database of airfoil geometries, training data and reduced order surrogate models.

3.1 Airfoil Parametrization

The current study relies on CFD-trained neural networks for the inverse design of rotorcraft airfoils for various operating conditions. The airfoil geometries are parameterized by the class function - shape function transformation (CST) (Ref. [9]), where Chebyshev polynomials are used as shape functions in this work instead of Bernstein polynomials proposed in the original CST methodology (Refs. [6, 10]).

There are multiple ways of parametrizing the airfoils; for instance, the American convention using camber and thickness (Ref. [6]), and parametrizing upper and lower surfaces separately (Refs. [1, 8]). The airfoil parametrization using camber and thickness is associated with an inverse problem and requires an iterative optimization to determine the defining geometry parameters. Addition of this iterative step increases the computational cost ($\mathcal{O}(10^1$ seconds$)$ for each airfoil) making it difficult to be used as push-button solution, compromising the major objective of this work.

Parametrizing the airfoils using upper and lower surfaces solves the problem associated with higher computational cost. Andrew (Ref. [1]) shows such parametrization using Bernstein polynomial of 8th order. In recent work, Hrithwik (Ref. [8]) shows the parametrization of airfoil with upper and lower surfaces using Chebyshev polynomials instead of Bernstein polynomials as the Chebyshev polynomials allow for a well-conditioned representation problem due to their near orthogonality.

From these studies, it can be inferred that the airfoil parametrization using Chebyshev polynomials has advantages over Bernstein polynomials. This is further demonstrated in Table 1, where the L1 norm of difference of leading coefficient from 5th - 7th order with the 8th order coefficient for both Bernstein and Chebyshev polynomials clearly indicates that the Chebyshev coefficients don't change significantly with order whereas Bernstein coefficients change considerably since they are non-orthogonal.

However, the parametrization using upper and lower surface requires a non-trivial constraint on the

Table 1: Comparison of Sensitivity of Bernstein and Chebyshev coefficients representing airfoil geometry with higher order polynomials

Delta	Bernstein Polynomials	Chebyshev Polynomials
8th - 7th order coefficient	$+50.281\%$	-5.625%
8th - 6th order coefficient	$+54.428\%$	-3.831%
8th - 5th order coefficient	$+66.239\%$	-9.198%

thickness to ensure the resulting geometry is an airfoil. Instead, for the ease of implementation, a modified airfoil parametrization technique comprising of upper surface and thickness distribution is implemented. It is similar to the modified CST, but the lower surface is replaced by thickness, and the airfoil geometry constraint of non-negative thickness ensures no intersecting surfaces in the airfoil during the inverse design. This way of airfoil parametrization is also done using 8th-order Chebyshev polynomials for both upper surface and thickness. This effectively gives 9 polynomial coefficients each for upper surface and thickness; totalling to 20 parameters including the TE (Trailing Edge) thickness to describe the airfoil geometry.

Mathematically, each of these surfaces is represented using the chord-wise coordinate (ψ) and the normalized surface coordinate (ζ). Eq. 1 and 2 respectively explain the parametrization of the upper surface and the thickness distributions. Here, C is the airfoil class function (describing the rounded leading edge and sharp trailing edge characteristics) and S represents the respective shape functions. This S is represented as a combination of Chebyshev polynomials, where the coefficients are the design parameters. Eq. 3 shows the lower surface distribution obtained from the parametrization.

$$\zeta_U(\psi) = C(\psi)S_U(\psi) + \psi\zeta_{TE,U} \tag{1}$$

$$\Delta\zeta(\psi) = C(\psi)S_{\text{thn}}(\psi) + \psi\Delta\zeta_{TE} \tag{2}$$

$$\zeta_L(\psi) = \zeta_U(\psi) - \Delta\zeta(\psi) \tag{3}$$

To generate a representative database of rotorcraft airfoils, fourteen airfoil sections defining industrially relevant rotorcraft blades as well as reference rotorcraft airfoil sections are selected. A total of 40 permutations per airfoil were generated using Latin-Hypercube sampling, resulting in 560 total perturbation airfoils in the geometric design space. Figure 1 shows the baseline set of 14 airfoils. Figure 2 show all the generated perturbations of baseline airfoil geometries, which form the training dataset of airfoil geometries. More details of airfoil database are available in Ref. [6].

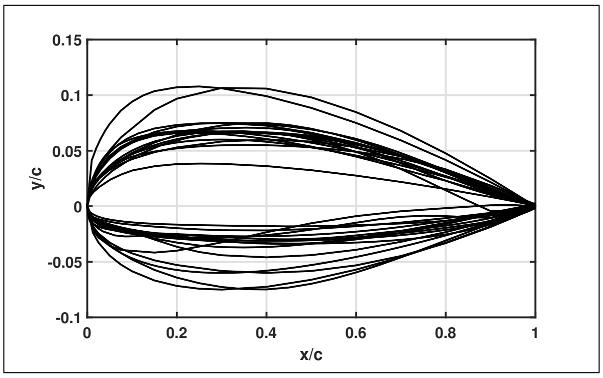

Figure 1: Airfoil Geometry Database; Baseline Airfoils

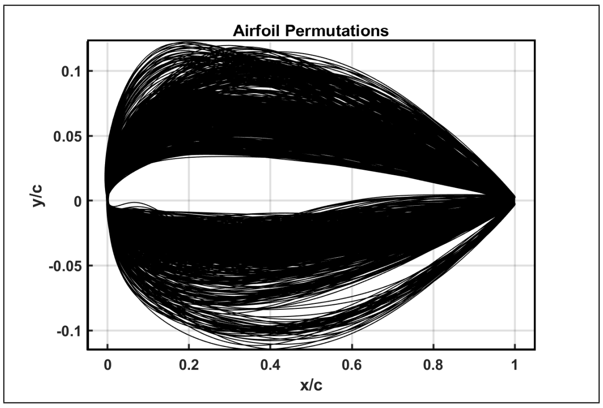

Figure 2: Airfoil Geometry Database; Perturbed Airfoils

3.2 Deterministic CFD solver and validation

To obtain the performance profiles of the database of airfoils, an in-house solver HAM2D is used to perform CFD simulations. This solver is built on a Hamiltonian path-based solution methodology (Ref. [11]) and uses a fifth-order WENO scheme (Ref. [12]) for spatial reconstruction, with Roe's approximate Riemann solver for inviscid flux, second-order central differencing for viscous flux, and a preconditioned GMRES

(Ref. [13]) for implicit time integration. The turbulent boundary layer is modelled using the one equation Spalart-Allmaras (SA) model (Ref. [14]). HAM2D is extensively tested and validated with various airfoil data available in the literature (Ref. [15]). 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 [6]. All stages of CFD simulations, including mesh generation (Ref. [16]), CFD flow solver setup, CFD runs, and post-processing, are automated using Python scripts to improve efficiency and reduce the potential for human error. CFD simulations for the training data generation are conducted for a range of angles of attack spanning from -10° to 20°, with a range of Mach numbers from 0.3 to 0.5 and corresponding Reynolds numbers, spaced linearly, ranging from 2.0 million to 3.333 million. These flow conditions cover typical low-speed operating conditions of rotorcraft vehicles.

3.3 Artificial Neural Networks (ANNs)

The objective of developing the surrogate model is to predict the aerodynamic quantities of interest for design applications, without having to perform a full-order CFD computation. The aerodynamic performance characteristics, specifically, lift coefficient, drag coefficient and pitching moment coefficient are fundamental design quantities. ANNs architectures, due to their highly accurate non-linear function approximation capabilities (Ref. [17]), are used for the surrogate model development in this study. The ANNs computes the entire performance curve instead of the performance at each angle of attack for a given geometry. Figure 4 shows the forward surrogate model architecture used in this study. The inputs of the surrogate model represent the CST parameters that define the airfoil geometry denoted by C, and the outputs are the performance parameters denoted by P (Cl, Cd and Cm), and this is referred to as the forward problem in this paper. For training the neural networks, log(C_d) is used instead of C_d to ensure the drag predicted by the neural networks is always positive. This approach is consistent with all the studies presented in this paper. As presented in Figure 3, the forward ANNs are very accurate, the details of which are also explained in Reference. [6].

Since the ANNs are trained to predict vectors of performance polar rather than single points, feature engineering becomes crucial. Both input and output features are scaled using mean and standard deviation. The airfoil characterization using Chebyshev polynomials ensures that the input coefficients are near-orthogonal, and therefore, Proper Orthogonal Decomposition (POD) of input features isn't necessary. But the POD of the output features (performance polar) becomes essential. For a given training data, the output is a 279-dimensional vector (31 values each of Cl, Cd and Cm at 3 Mach numbers), which makes it even more necessary to reduce the output dimensions. This is done by systematic POD on individual polar at each Mach number, resulting in a reduction of the 279-dimensional output vector to an 84-dimensional vector while retaining more than 99% of the information of each polar. It is to be noted that the Mach numbers are not specified in the inputs, rather the polar at individual Mach numbers are used as vectors in the output. For inverse design, inverting the forward ANNs as shown in Figure 5, to predict an airfoil geometry for a given performance polar isn't accurate as presented in Figure 6. This is mainly attributed to the inverse scattering problem. While the forward problem is well defined, the inverse problem is subjected to an inverse scattering problem i.e., the same performance polar can be given by multiple airfoil geometries and therefore the one-to-one mapping of the inverse problem isn't physical. This non-unique mapping makes it difficult for the inverted artificial neural networks (I-ANNs) to converge on a solution, and hence cannot be directly used for airfoil design. This is demonstrated in Figure 7 where two completely dif-

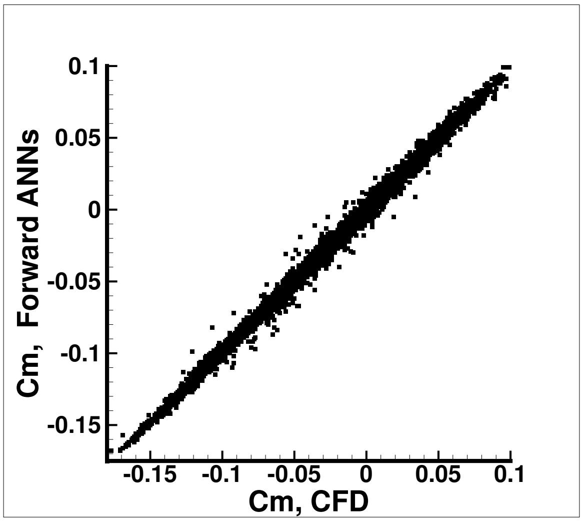

Figure 3: Accuracy of Forward Artificial Neural Networks (I-ANNs); Comparison of True C_m (CFD) with Forward ANNs predicted C_m

ferent airfoil geometries have similar performance parameters. The cost function for I-ANNs is defined by Equation 4.

$$J = \frac{1}{\#\text{Train}} \sum_{\text{Train}} (\hat{\text{C}}_{\text{True}} - \hat{\text{C}}_{\text{Predicted}})^2 \tag{4}$$

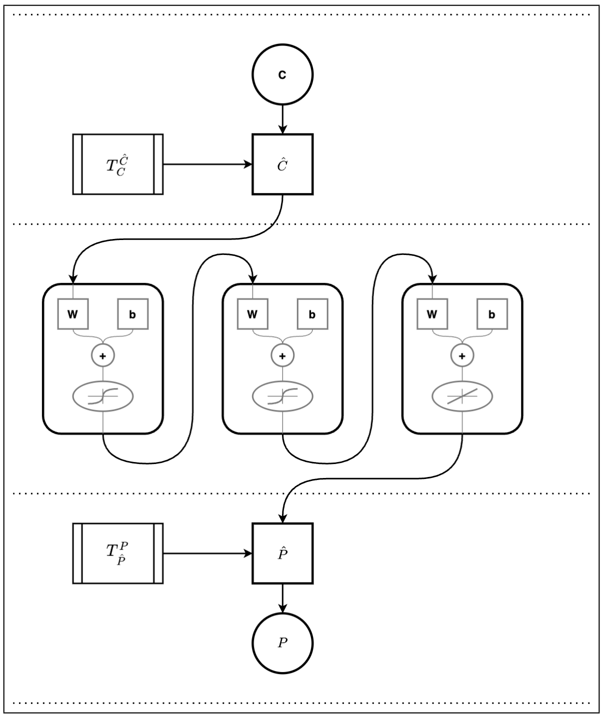

Figure 4: Architecture of Forward Artificial Neural Networks (ANNs)

3.4 Tandem Neural Networks (T-NNs)

The problem of inverse scattering is solved using a special type of deep learning module, Tandem Neural Networks. T-NNs architecture has two ANNs working in series as presented in Figure 8. A pre-trained ANNs is used for the forward problem (bottom portion in Figure 8). Taking Chebyshev coefficients as inputs and outputting performance polar (vectors of Cl, Cd and Cm) makes the second part of T-NN, while the first part (highlighted in green in Figure 8 (top portion)) is the same as the inverted ANNs used for inverse design. Performance polar are inputted to the first part of T-NNs, which out-puts intermediate layer data that represents airfoil geometry. The intermediate layer data is then inputted into the pre-trained forward ANNs with frozen weights and outputs the performance polar. In this way, the T-NNs takes performance polar as input and returns performance polar

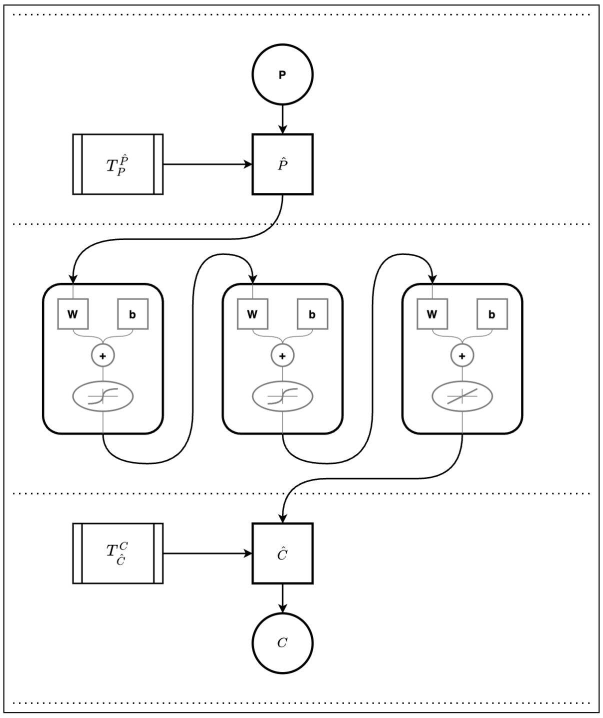

Figure 5: Architecture of Inverted Artificial Neural Networks (ANNs)

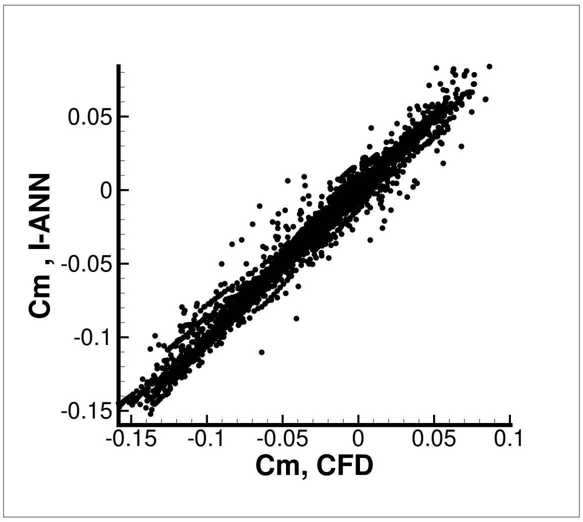

Figure 6: Accuracy of Inverted Artificial Neural Networks (I-ANNs); Comparison of True C_m (CFD) with I-ANNs predicted C_m

as output, with the airfoil geometry being extracted from the intermediate layer. The weights of the forward ANNs are fixed, while the weights of the inverse ANNs are trained to minimize the cost function defined by Equation 5. It is to be emphasized that the cost function for the T-NNs is different from the cost function for the I-ANNs (Equation 4). This approach overcomes the problem of non-uniqueness in the inverse

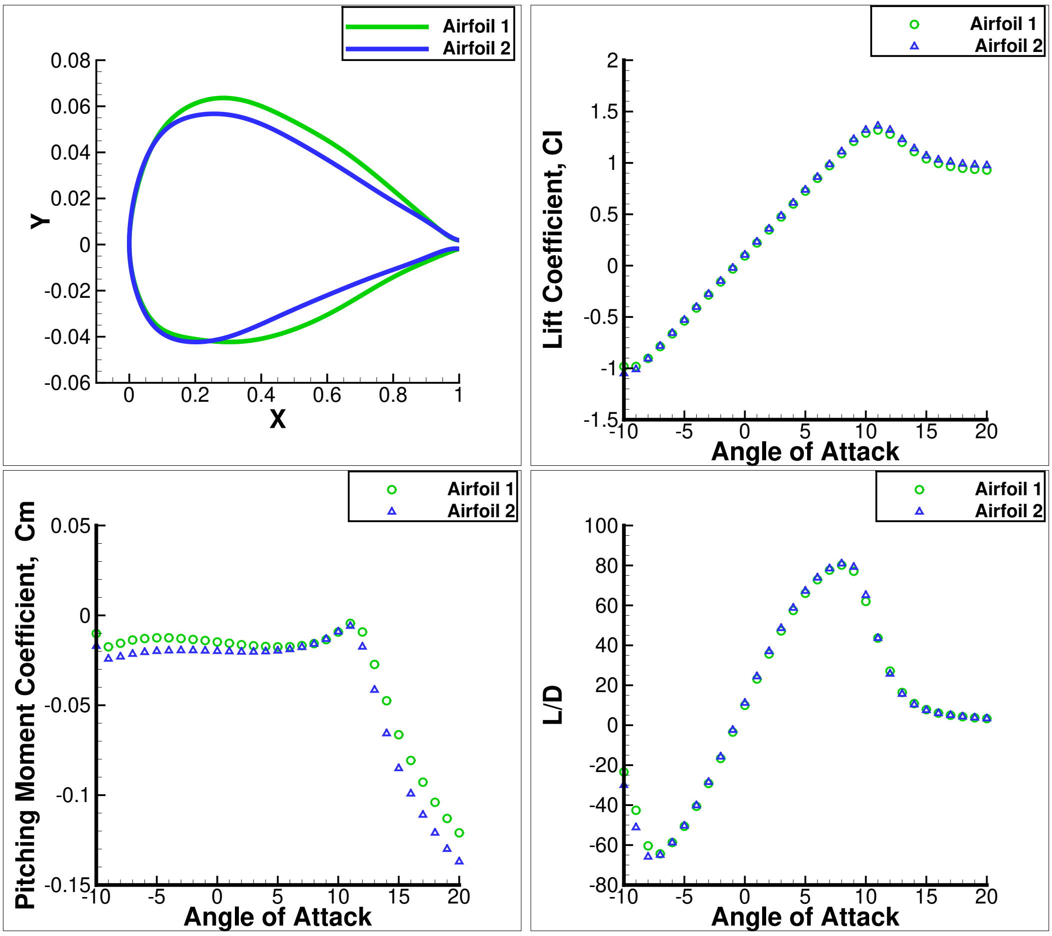

Figure 7: Inverse Scattering Problem. Clockwise from top-left: Airfoil Geometry, Comparison of Cl polar, Comparison of L/D polar, Comparison of Cm polar

design process, as the T-NNs is not required to predict the same airfoil geometry but rather ensure that the same performance polar is computed. It is also noteworthy that the cost function for T-NNs (Equation 5) is defined using the scaled polar. This is particularly important because the range of values for each performance polar is different in the orders of magnitude of 10 to 1000. The L2 norm error of predicted and original aero-dynamic coefficients for the I-ANNs and T-NNs are presented in Table 2. The L2 norm is computed for baseline airfoils which are not a part of training for both I-ANNs and T-NNs. The L2 norm for I-ANNs is significantly higher than that of T-NNs. This also quantitatively validates the use of T-NNs for inverse design. The accuracy of T-NNs as presented in Table 2 and Figure 9 is better than that of I-ANNs (Figure 6). The comparison of the performance of T-NNs and I-ANNs is done using the vector plots of pitching moment coefficients as shown in Figures 9 and 6 due to the fact that the pitching moment coefficient is more sensitive to a small change in airfoil geometry. The vector plots of the lift coefficient and lift-to-drag ratio follow similar trends as those of the pitching moment coefficient.

$$J = \frac{1}{\#\text{Train}} \sum_{\text{Train}} \left(\hat{\mathrm{P}}_{\text{True}} - \hat{\mathrm{P}}_{\text{Predicted}}\right)^2 \tag{5}$$

4 NEURAL NETWORK TRAINING AND COMPUTATIONAL COST

The hyperparameters for both forward ANNs and T-NNs are tuned using Random Search. The search space is bounded by a pre-defined domain of hyperparameter values and points are randomly searched in the domain. The details of the number of hidden layers and neurons in each layer are mentioned below.

1. Forward ANNs: 20-20-30-50-70-90

Table 2: Comparison of quantitative error estimates of the I-ANNs and T-NNs for baseline airfoil geometries.

Aerodynamic Coefficients	L_2 norm of I-ANNs	L_2 norm of TNNs
C_l	3.45×10^{-2}	1.97×10^{-2}
L/D	2.46	1.57
C_m	7.20×10^{-3}	3.90×10^{-3}

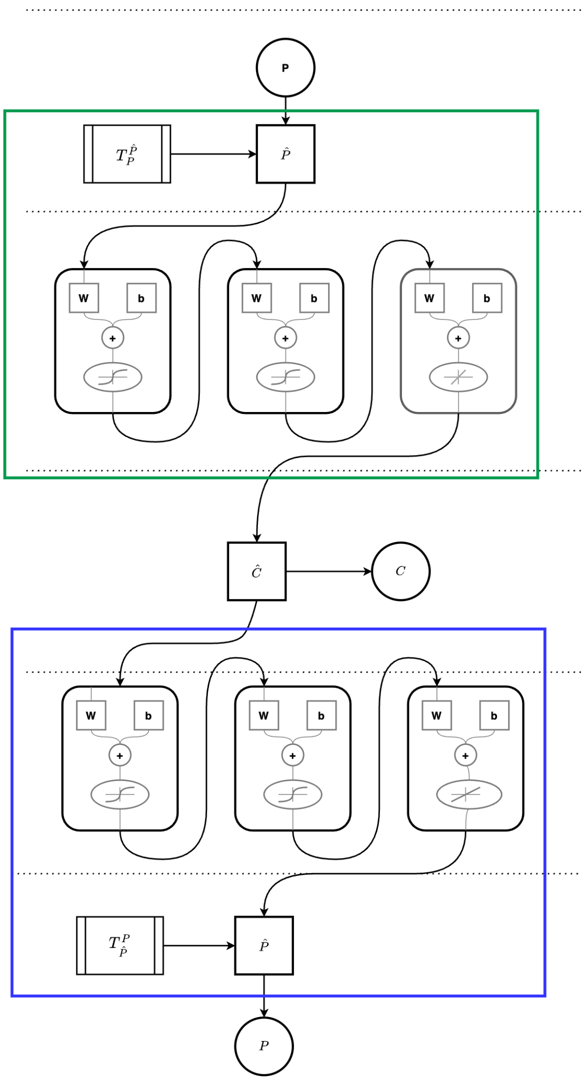

Figure 8: Architecture of Tandem Neural Networks (T-NNs)

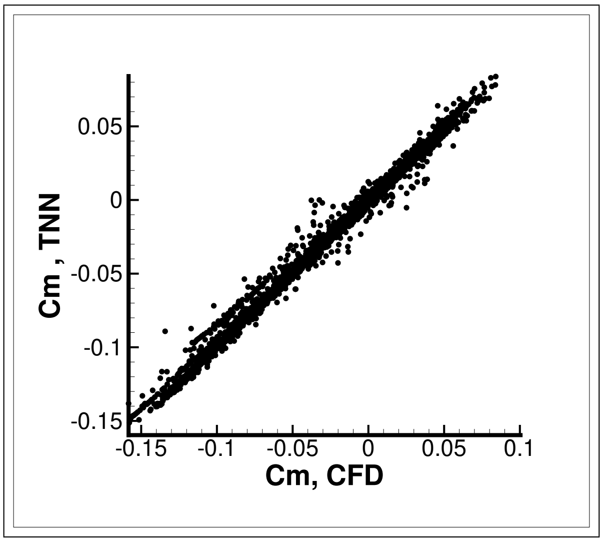

Figure 9: Accuracy of Tandem Neural Networks (T-NNs); Comparison of True C_m (CFD) with I-ANNs predicted C_m

2. T-NNs: 90-75-75-50-50-20-20 + Forward ANNs

For the forward problem with a single Mach number as input, a two-hidden layers with 20 neurons each are sufficient to predict polar with 99% accuracy (Ref. [6]). For the problem under consideration in this paper, multiple Mach numbers with additional design constraints make both forward ANNs and T-NNs have deeper layers with a higher number of neurons.

For both ANNs and T-NNs, the hyperbolic tangent activation function is used in all layers except the last layer where the linear activation function is used. The Huber Loss function is used for training. The entire database is split into training and test data in the ratio of 70:30. From the training data, 20% of the data is further split as validation data to monitor the loss to ensure there is no overfitting, the details of which is also presented in Figure 10. Deep networks are prone to overfitting, and therefore, standard regularization is also used to prevent overfitting. Figure 10 shows the comparison of Loss with epochs and the log(Loss) with epochs. It can be seen that the validation loss is very close to the training loss and there is no increasing trend of the validation loss with epochs, which validates that there is no overfitting. The right side of Figure 10 shows the comparison of log(Loss) for both training and validation data, which also shows that there is no overfitting. The ANNs and T-NNs are trained for 5000 epochs.

The comparison of computational cost using CFD and Neural Networks is presented in Table 3. The difference in computational cost with CFD and Neural Networks is very significant, and it demonstrates that Neural Networks can be used as push-button solutions for forward polar prediction and inverse design.

5 IMPROVEMENT OF T-NNS ARCHITECTURE

5.1 Improved Loss Function

A detailed study on the loss function of T-NNs is done to ensure that the overall accuracy of the predicted parameters remains as good as the forward ANNs. As explained in the section above, the loss function of T-NNs is defined by the performance polar and not the airfoil geometry parametrization coefficients. To study this in detail, a T-NNs is trained with an improved loss function which includes both airfoil geometries and performance polar as mentioned in Equation 6, where a and b represent the fraction of weights added to the loss function. The values of a and b are varied in such a way that their sum is always equal to 1. The improved loss function is simply a transition from I-ANNs (b = 0) to T-NNs (a = 0). Table 4 shows the comparison of L2 norm for different combinations of a and b, and it can be seen that the conventional loss function of T-NNs (a = 0) is the most accurate which also validates our hypothesis of T-NNs solving the inverse scattering problem. The same loss function is used in all further studies.

$$
\begin{aligned}
J = \frac{a}{\#\text{Train}} \sum_{\text{Train}} (\hat{\text{C}}_{\text{True}} - \hat{\text{C}}_{\text{Predicted}})^2 + \\
\frac{b}{\#\text{Train}} \sum_{\text{Train}} (\hat{\text{P}}_{\text{True}} - \hat{\text{P}}_{\text{Predicted}})^2
\end{aligned} \tag{6}
$$

5.2 Multiple Design Constraints

The architecture of T-NNs provides flexibility in terms of multiple design constraints. For instance, if we want to optimize an airfoil with the objective of reducing drag, the airfoil predicted by T-NNs can be optimized in such a way that its maximum thickness, trailing edge (TE) thickness and stall margin are pre-specified by the designer. This is done by adding the design constraints (for instance maximum thickness and TE thickness) at both inputs and outputs of the T-NNs. This is demonstrated in Figure 11 which has polar, Mach number, max. thickness and TE thickness as inputs and outputs, and the intermediate layer has CST coefficients representing airfoil geometry, Mach number, max. thickness and TE thickness. As explained in section 4, the hyperbolic tangent activation function is used in all layers except the output layer which has a linear activation function. Adding TE thickness as an additional constraint has the risk of neural networks predicting an airfoil geometry with negative TE thickness given that TE thickness values are very small and for some airfoils these are zero. A negative TE thickness gives airfoil with intersecting surfaces which is impractical. To ensure that the TE thickness is always positive, a non-negative weight constraint is added to the output layer that predicts TE thickness in addition to having linear activation function. The non-negative weight constraint ensures that the TE thickness is always greater than or equal to zero. Adding such design constraints in the optimization loop ensures that the airfoil geometry is more practical. All further studies in this paper have been done using both maximum thickness and TE thickness as additional constraints in the T-NNs.

6 RESULTS

This section discusses the results of the design problem of SC1095 airfoil using T-NNs and its impacts on the performance of a rectangular blade.

6.1 Design Problem 1: Increasing L/D of SC1095 airfoil with similar lift and stall characteristics independent of Mach numbers.

The objective of this design problem is to increase the lift-to-drag ratio of SC1095 airfoil by 5% for all angles of attack at all three Mach numbers (0.3, 0.4 and 0.5) while retaining the Cl and Cm characteristics as that of baseline airfoil, i.e., reducing the drag of the airfoil. Therefore, the design objective is to optimize the airfoil such that the lift-to-drag ratio at all three Mach numbers increases without changing the lift and moment polar. The lift-to-drag ratio predicted from the forward ANNs is modified as per the design objective and the Cl and Cm polar are unchanged to define the target polar. The target polar along with the maximum thickness and TE thickness is inputted to the T-NNs which outputs the optimized airfoil geometry from the intermediate layer. Maximum thickness and TE thickness are inputted to both inputs and outputs of the neural networks to enforce additional constraints. It is also remarked that the Mach numbers aren't used as

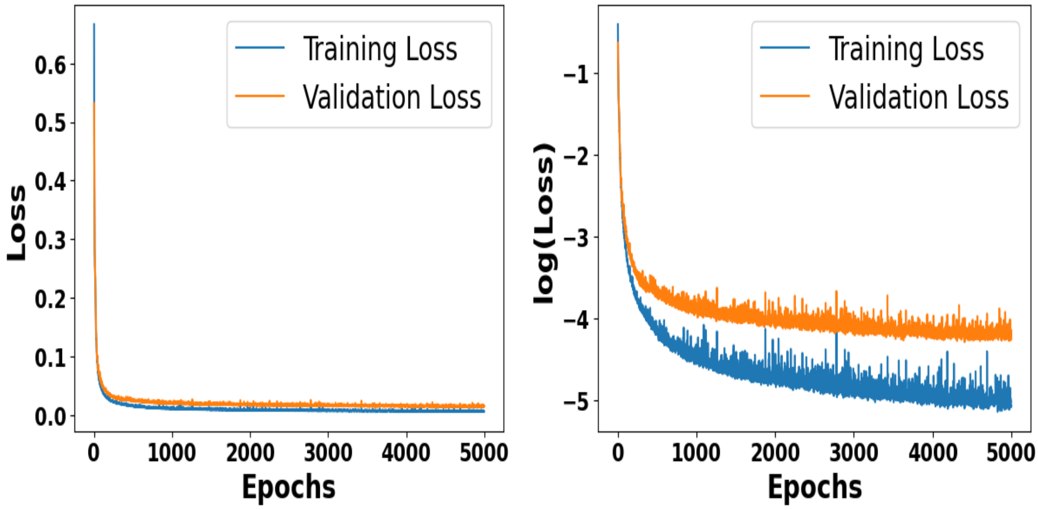

Figure 10: Comparison of Loss for both training and validation data for Tandem Neural Networks (T-NNs). Left Figure: Loss v/s Epochs, Right Figure: Log(Loss) vs Epochs

Table 3: Comparison of Computational Cost for generating performance parameters using CFD and Neural Networks

Computational Exercise	Computational Cost (hours)
CFD Training Data Generation per Mach number	$67,704$
Train Forward ANNs	0.20
Train TNNs	0.30
CFD Simulation for Single Airfoil at 1 Mach number	1.30
Evaluate Single Airfoil using ANNs	1.50×10^{-7}

Table 4: Comparison of L2 norm for Improved Loss Function for T-NNs.

a	b	L_2 norm of C_l	L_2 norm of L/D	L_2 norm of C_m
0	1	1.97×10^{-2}	1.57	3.90×10^{-3}
0.2	0.8	2.60×10^{-2}	1.80	5.80×10^{-3}
0.3	0.7	2.80×10^{-2}	2.02	5.90×10^{-3}
0.5	0.5	2.30×10^{-2}	1.80	5.00×10^{-3}
0.7	0.3	3.10×10^{-2}	2.23	6.40×10^{-3}
0.8	0.2	3.10×10^{-2}	2.21	6.70×10^{-3}
1	0	3.45×10^{-2}	2.46	7.20×10^{-3}

inputs to the network, rather the polar at each Mach number are vectorized (as explained in section 3.3) which ensures that one single airfoil geometry is optimized for all three Mach numbers. A comparison of the baseline SC1095 airfoil and the optimized airfoil for Mach 0.4 can be seen in Figure 12. The increase in max. L/D for Mach 0.4 is 4.067%, which is close to the target. The results for Mach 0.3 and 0.5 follow a similar trend. For this design problem, we don't exactly meet the target, and it may be attributed to the difference in aerodynamic properties of the airfoil at different Mach numbers. One single airfoil might not be optimum at multiple Mach numbers, and it is consistent with the aerodynamic properties of airfoils at higher speeds. The airfoil geometry predicted by T-NNs has a relatively higher thickness as compared to SC1095 airfoil, but the position of maximum thickness moves towards the leading edge which in turn reduces drag and therefore increases the L/D. The comparison of polar computed from forward ANNs for the baseline airfoil and the optimized airfoil at Mach 0.4 is also presented in Figure 12. The polar of the optimized airfoil

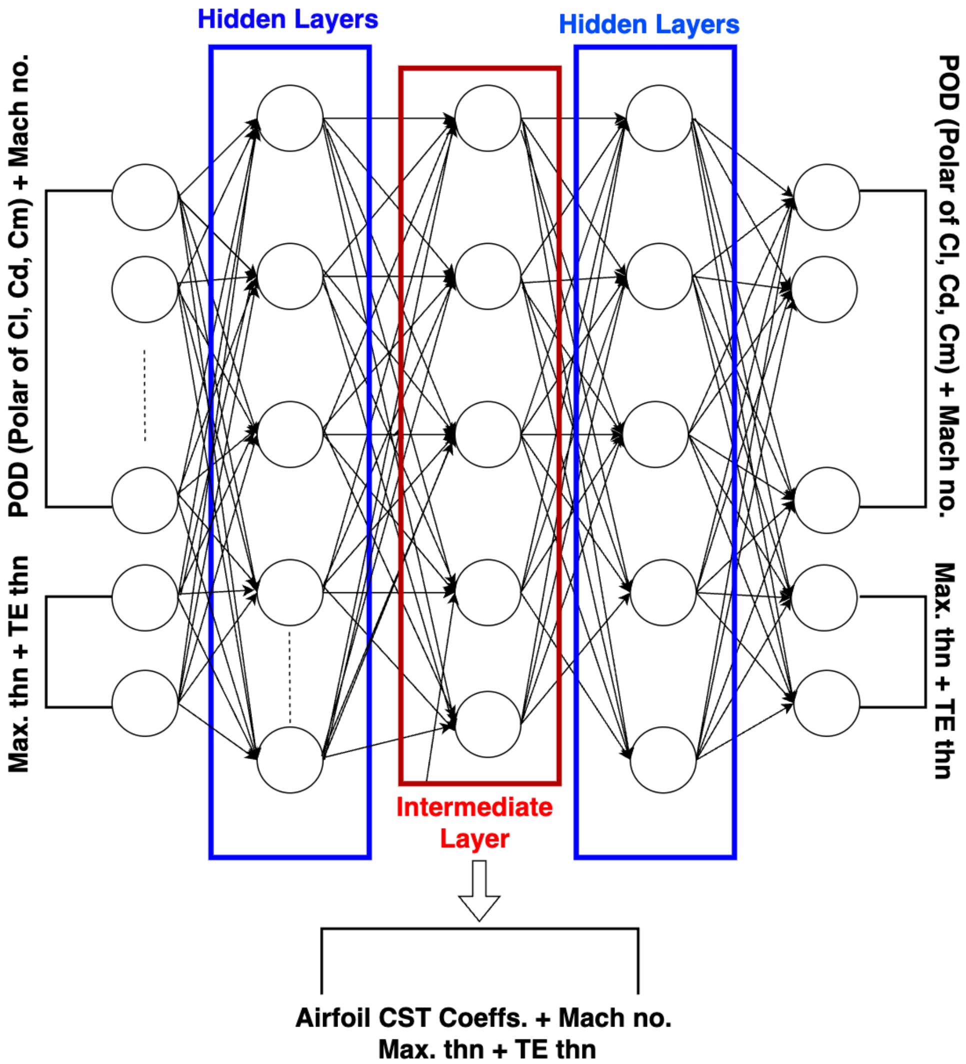

Figure 11: Tandem Neural Networks (T-NNs) architecture for inverse design with multiple design constraints

computed from forward ANNs shows a good comparison with the target polar for lift, lift-by-drag and moment coefficients.

6.2 Design Problem 2: Increasing L/D of SC1095 airfoil with similar lift and stall characteristics at individual Mach numbers.

Unlike section 6.1, the design objective of this section is to optimize the airfoil geometry at individual Mach numbers, i.e., three airfoils for three Mach numbers (0.3, 0.4 and 0.5). The design criteria remain the same, that is to increase lift-to-drag ratio by 5% while retaining the Cl and Cm characteristics. T-NNs allow for such design constraint by including Mach numbers as both inputs and outputs of the T-NNs. It means that the output has (31 $\times 3$ = 93 dimension of polar (Cl, Cd and Cm)), and the database for individual Mach numbers is stacked horizontally. This effectively gives three airfoil geometries at three Mach numbers. Like section 6.1, both maximum thickness and TE thickness are also added as constraints (to both inputs and outputs of the Neural Networks) to ensure that the optimized airfoils have similar maximum thickness and TE thickness as that of baseline SC1095 airfoil. 5 shows the comparison of maximum thickness for all three optimized airfoils, and the maximum thickness for these airfoils is closer to the baseline SC1095 airfoil (max. thickness is 9.5 %) because of the additional thickness constraint. Figure 13 shows the optimized airfoil geometries at all three Mach numbers and the corresponding L/D polar. All three optimized airfoils have similar lower surfaces but the difference in their aerodynamics is majorly due to the difference in their upper surfaces. The position of maximum thickness

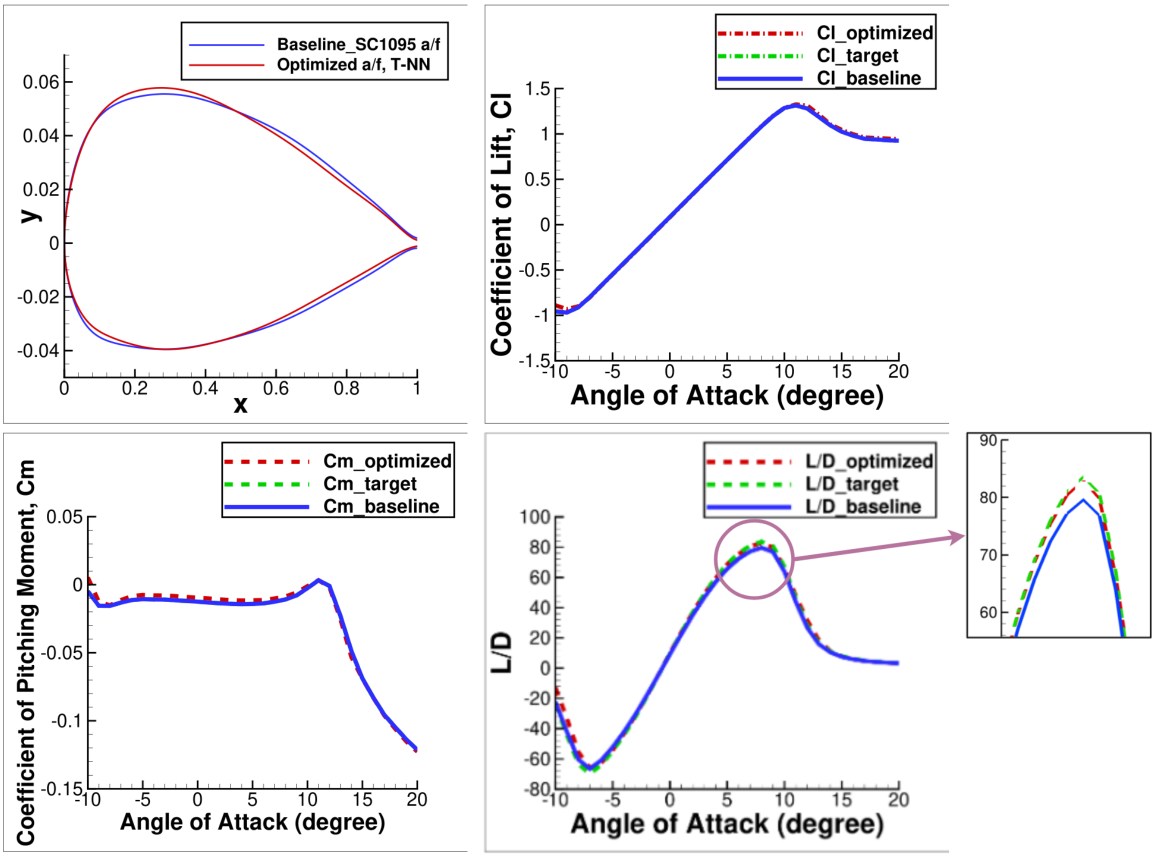

Figure 12: Comparison of airfoil geometry and performance polar for design problem 1 at Mach 0.4. Clockwise from top-left: Airfoil Geometry, Comparison of C_l polar, Comparison of L/D polar along with the zoomed view, Comparison of C_m polar

changes for all three airfoils as compared to the baseline. It is to be noted that the Cl and Cm polar follow a similar trend. The angle of attack at which L/D is maximum changes for both baseline airfoil and optimized airfoils as shown in Table 5. It is interesting to note that the stall margin for the optimized airfoil is better at Mach 0.3 and 0.5 than the baseline airfoil at the corresponding Mach numbers. Table 5 also shows the increase in maximum L/D for each Mach number, and it is to be emphasized that the optimized airfoil at Mach 0.5 has a higher max. L/D than the target. This also demonstrates that additional design constraints can be included in the Neural Networks to have more control over the predicted airfoil geometry and the aerodynamic performance.

6.3 Statistical Analysis: Generating a family of Airfoils

The current T-NNs architecture is deterministic and therefore for a given design objective, only one airfoil can be generated. For instance, in section 6.2, there are three design objectives, one at each Mach number and therefore we got three airfoils. To generate a family of airfoils for a single design objective i.e., multiple airfoils all of which give similar performance, a statistical analysis is performed where a range of maximum thickness and TE thickness are chosen randomly from a Gaussian distribution and for each of these combinations of variables, an airfoil is generated. The up-

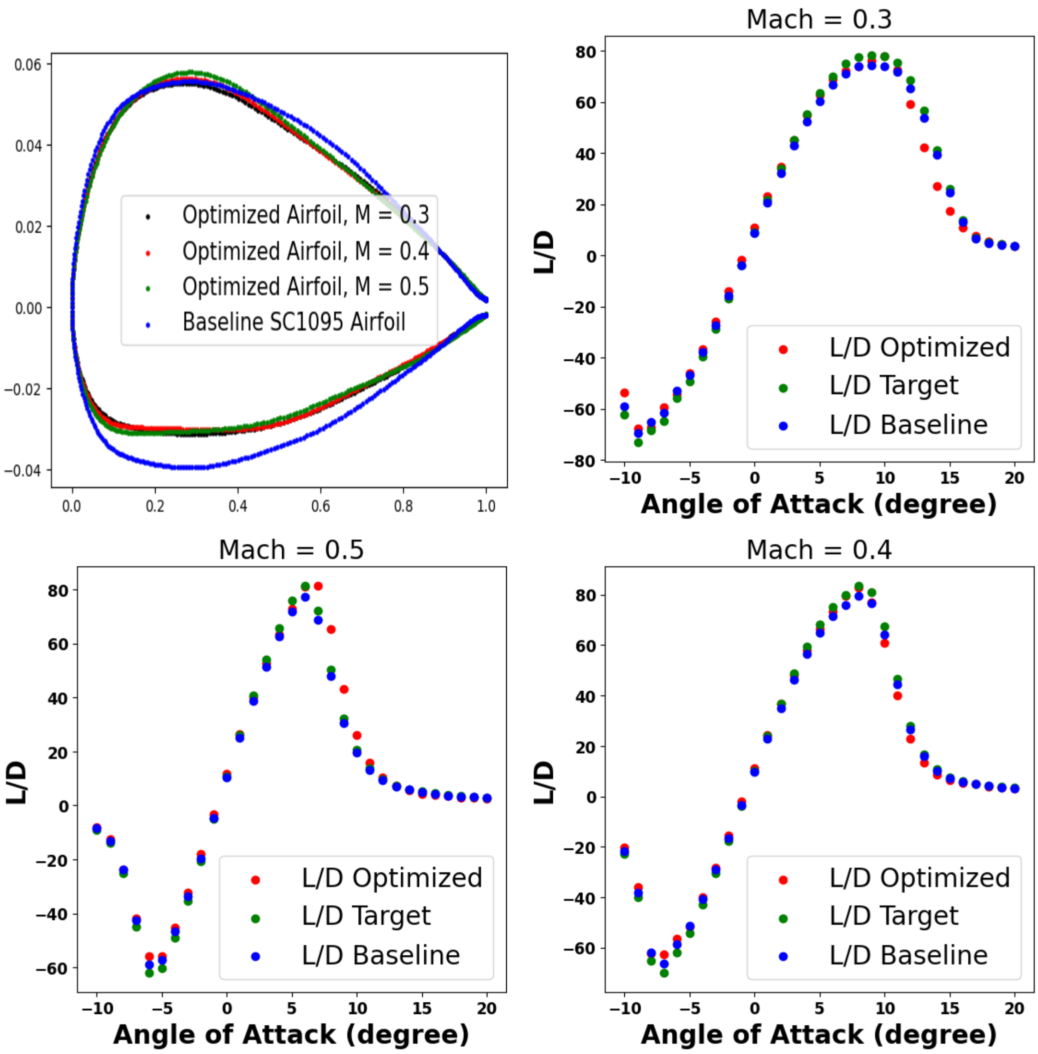

Figure 13: Comparison of Optimized airfoil geometries and the corresponding L/D polar at individual Mach numbers for Design Problem 2. Clockwise from top-left: Airfoil Geometry, L/D polar at Mach 0.3, L/D polar at Mach 0.4 and L/D polar at Mach 0.5

Table 5: Comparison of performance of optimized airfoils at individual Mach numbers

Mach Number	$\alpha_{\text{max}L/D}$ Baseline airfoil	$\alpha_{\text{max}L/D}$ Optimized airfoil	% Increase in Max. L/D	Max. Thickness
0.3	9 degree	10 degree	4.53	9.05%
0.4	8 degree	8 degree	4.11	9.30%
0.5	6 degree	7 degree	5.20	9.64%

per and lower limits for these variables (max. thickness and TE thickness) are pre-specified by the designer. For the problem under consideration, the max. thickness and TE thickness are varied between 60% to 110% of the baseline SC1095 airfoil. This way of inverse design allows the designer to generate multiple airfoil geometries within pre-specified criteria, and then choose airfoils of interest from those multiple airfoils. This approach of inverse design is a novel contribution to this work and is also a more practical way of designing airfoils.

To have constraints on both max. thickness and TE thickness, both these quantities are used as inputs, outputs and intermediate layer data while training the T-NNs as explained in sections 6.1 and 6.2. For this problem, Mach number is also specified as an input similar to section 6.2. Figure 14 shows the workflow of statistical analysis combined with T-NNs to generate a family of airfoils. The combination of max. thickness and TE thickness along with the target polar and target Mach number are given as inputs to the pre-trained T-NNs, and multiple airfoil geometries are extracted as outputs from the T-NNs. The design objective for Statistical Analysis is the same as that of section 6.2. The results are demonstrated in Figure 15. The top left of Figure 15 shows 50 airfoil geometries predicted by the T-NNs for 50 combinations of max. thickness and TE thickness as shown in the bottom left of Figure 15. The corresponding lift and lift-to-drag polar is also presented in Figure 15. It is interesting to note that the C_l polar for all predicted airfoils matches with that of the baseline SC1095 airfoil, whereas some airfoils reach the target L/D polar while others don't. This is the major objective of this analysis, i.e., to predict multiple airfoils for the same design objective and then choose the predicted airfoils based on the requirement. Figure 16 shows the distribution of max. thickness and TE thickness which are input design constraints, and the corresponding distribution of maximum C_l and maximum L/D for 50 airfoils predicted by the T-NNs. From Figure 16, it can be seen that the maximum C_l for the predicted airfoils vary between 1.07 to 1.13, and the maximum L/D vary between 76.34 to 78.31. From these observations, it can be concluded that our hypothesis for Statistical Analysis and inverse design remains valid such that multiple airfoil geometries are generated all of which have similar performance polar. For this work, 50 airfoils are used as a demonstration, but since this architecture is developed as a push-button solution, the generation of hundreds of thousands of airfoils using T-NNs can be done in a few seconds. The key outcomes of Statistical Analysis are:

1. A family of airfoil geometries can be designed for the same design objective using Statistical Analysis

2. Multiple combinations of design variables can be used as inputs. For instance, in addition to max. thickness and TE thickness, stall margin and position of max. thickness can also be specified to have more control of the predicted airfoil geometries.

6.4 Comparison of performance of rectangular blade with optimized airfoil

A simple rotor with 4 rectangular blades and a single airfoil with a linear twist is considered for the analysis at hover. The rotor radius is 26.8333 feet with a tip Mach number of 0.5. The rotor performance is computed using in-house developed Blade Element Momentum Theory (BEMT) code. The optimized airfoil improves the rotor performance which is shown in Figure 17. The baseline corresponds to SC1095 airfoil, the performance polar for which is generated using the CFD. The optimized airfoil is generated from the T-NNs, the details of which are explained in section 6.1. The performance polar of the optimized airfoil is generated using both CFD (green line in Figure 17) and T-NNs (blue line in Figure 17). The modification in L/D slightly improves the performance of the rotor which is shown in Figure 17. The performance of rotor computed using both CFD and T-NNs are comparable as shown in Figure 17 which also validates the use of T-NNs for inverse design of airfoil. While the improvement in Figure of Merit (FM) isn't significantly higher as shown in the bottom right of Figure 17, this increase in FM is by only a small improvement in the L/D. For instance, if the gross weight of a helicopter is around 20,000 pounds, then there would be roughly 100 pounds more of payload through just this airfoil shape optimization. It is also remarked that the airfoil optimization using the T-NNs is like a push-button solution, and it takes milliseconds to optimize one single airfoil. Therefore, a highly cost-effective and accurate surrogate model is developed and implemented for the inverse design of the airfoil.

7 CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

A Neural Networks based surrogate model is developed to perform inverse design of airfoils. The modified cost function for such architecture ensures that the inverse scattering problem is avoided. The following conclusions are derived from this study.

1. Representation of airfoil geometry using upper surface and thickness with Chebyshev polynomials ensures that there are no overlapping surfaces during the inverse design process.

2. The study of improved loss function highlights the advantage of using polar in the loss function for

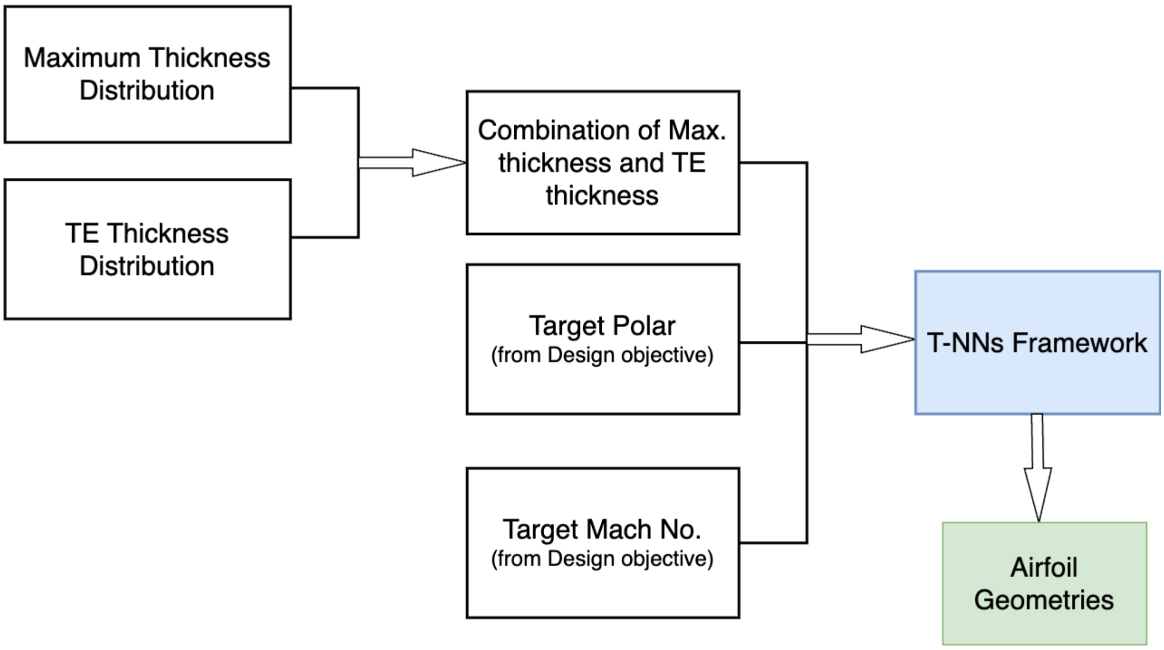

Figure 14: Flowchart explaining the working of Statistical Analysis with T-NNs; Generating a family of airfoils with variable max. thickness and TE thickness

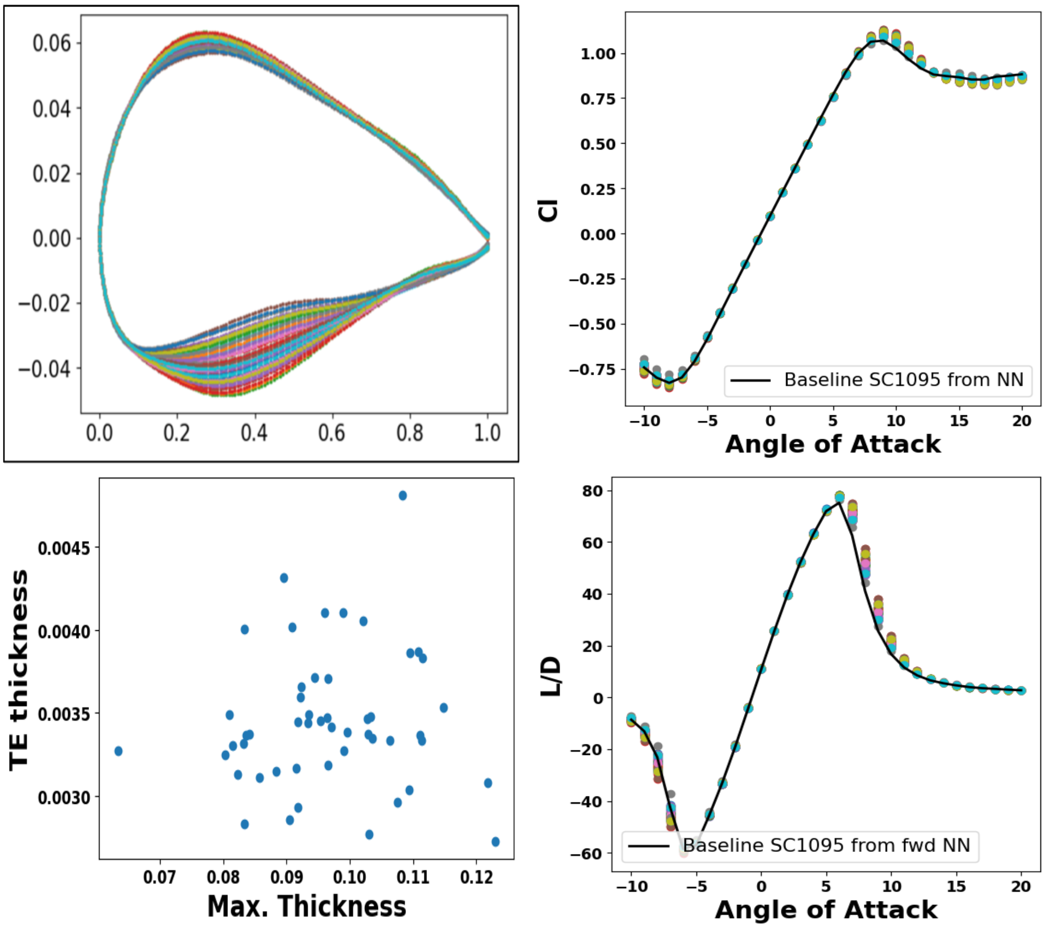

Figure 15: Statistical Analysis to generate a family of airfoils. Clockwise from top-left: Airfoil Geometries, Comparison of Cl polar, Comparison of L/D polar, Variation of TE thickness and maximum thickness

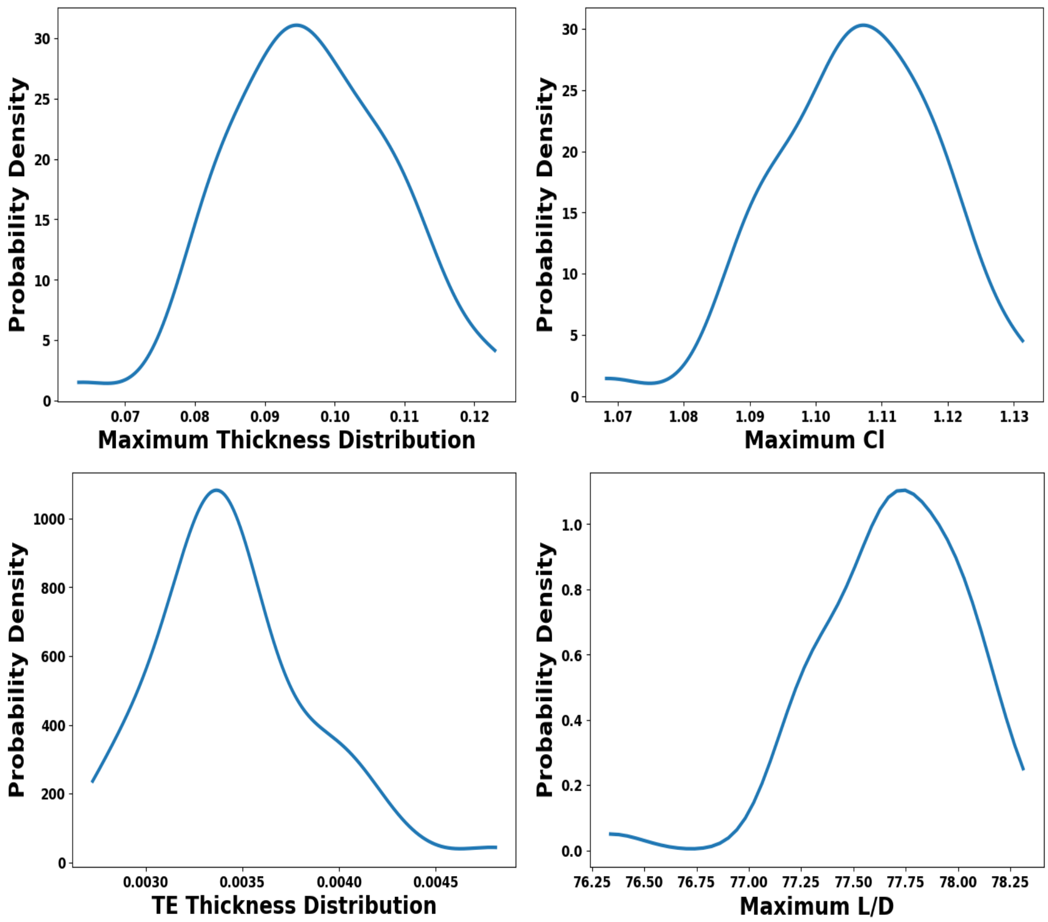

Figure 16: Comparison of the distribution of geometric constraints and the corresponding distribution of performance parameters. Clockwise from top-left: Distribution of max. thickness, Distribution of max. C_l, Distribution of max. L/D, Distribution of TE thickness

T-NNs.

3. Statistical Analysis generates a family of air-foils for a given design constraint which can be used for further study of multiple airfoils all of which give similar performance.

4. BEMT validates that the rotor blade with optimized airfoil gives better performance.

7.2 Future Work

The possible improvements and future works adding to this study are listed below:

1. Expanding the training database to higher Mach numbers, and varying Reynolds number independently of Mach number, and including variable stall margin with inverse design framework.

2. Improving CFD data to match with the experiments using Field Inversion Machine Learning (FIML).

3. Including both fully turbulent and free transition conditions in the database.

4. Training ANNs to predict boundary layer thickness at the trailing edge.

5. Blade optimization including multiple airfoils along the span with optimization of chord, ta-per and twist.

6. Study on uncertainty quantification for both airfoil parametrization and inverse design.

Author Contact: Apurva Anand apurva01@umd.edu

8 ACKNOWLEDGEMENTS

This research is conducted with support from the Vertical Lift Research Center of Excellence (VLRCOE) under the technical monitoring of Dr. Mahendra Bhagwat. The authors also acknowledge the University of

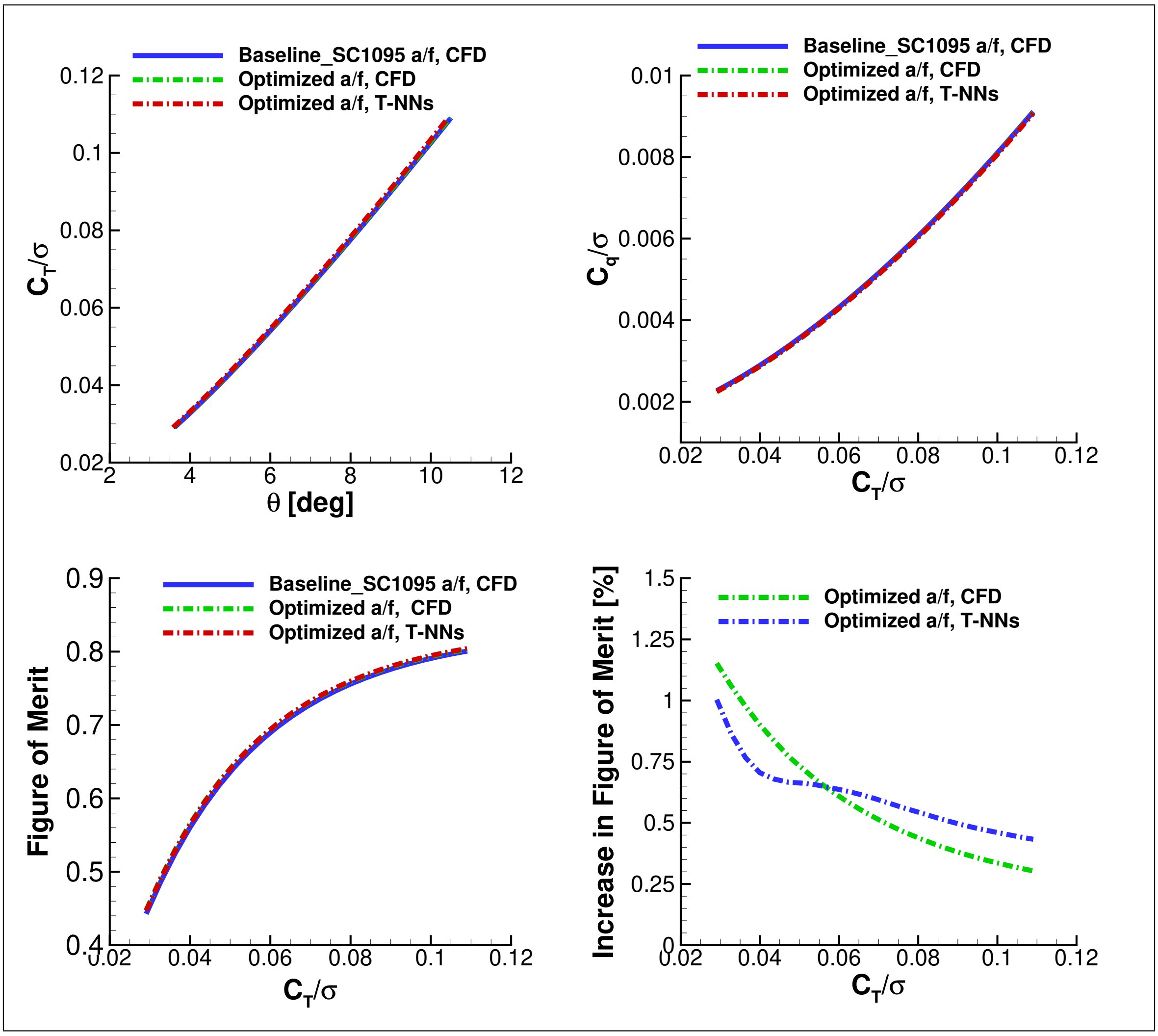

Figure 17: Comparison of rotor performance for baseline and optimized airfoil. Clockwise from top-left: Comparison of Thrust Coefficient, Comparison of Torque Coefficient, Comparison of increase in Figure of Merit, Comparison of Figure of Merit

Maryland super-computing resources, Zaratan HPC cluster, made available for conducting the research in this paper.

REFERENCES

[1] Andrew Glaws, Ryan N King, Ganesh Vijayakumar, and Shreyas Ananthan. “Invertible neural networks for airfoil design”. In: *AIAA Journal* 60.5 (2022), pp. 3035–3047.

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

[3] Jeffrey P Slotnick, Abdollah Khodadoust, Juan Alonso, David Darmofal, William Gropp, Elizabeth Lurie, and Dimitri J Mavriplis. *CFD vision 2030 study: a path to revolutionary computational aerosciences*. Tech. rep. 2014.

[4] M Costes, T Renaud, and B Rodriguez. “Rotorcraft simulations: a challenge for CFD”. In: *International Journal of Computational Fluid Dynamics* 26.6-8 (2012), pp. 383–405.

[5] Dianjing Liu, Yixuan Tan, Erfan Khoram, and Zongfu Yu. “Training deep neural networks for the inverse design of nanophotonic structures”. In: *Acs Photonics* 5.4 (2018), pp. 1365–1369.

[6] Nick Paternostro, Koushik Marepally, Apurva Anand, Bumseok Lee, and James Baeder. “Application of CFD-Trained Neural Networks on the Rotorcraft Airfoil Design Process”. In: *VFS Annual Electric VTOL Symposium* (2023).

[7] Koushik Marepally, Yong Su Jung, James Baeder, and Ganesh Vijayakumar. “Uncertainty quantification of wind turbine airfoil aerodynamics with geometric uncertainty”. In: *Journal of Physics: Conference Series*. Vol. 2265. 4. IOP Publishing. 2022, p. 042041.

[8] Hrithwik Shalu, Bharath Govindarajan, Ananth Sridharan, and Rajnesh Singh. “Blade Shape Optimization of Rotors using Neural Networks”. In: *79th Annual Forum of the Vertical Flight Society* (2023).

[9] Brenda M Kulfan. “Universal parametric geometry representation method”. In: *Journal of aircraft* 45.1 (2008), pp. 142–158.

[10] Koushik Marepally. “A Meta learning based aerodynamic analysis framework for wind turbine design applications”. PhD thesis. University of Maryland, College Park, 2023.

[11] Jayanarayanan Sitaraman and Beatrice Roget. “Solution algorithm for unstructured grids using quadrilateral subdivision and Hamiltonian paths”. In: *52nd Aerospace Sciences Meeting*. 2014, p. 0079.

[12] Guang-Shan Jiang and Chi-Wang Shu. “Efficient implementation of weighted ENO schemes”. In: *Journal of computational physics* 126.1 (1996), pp. 202–228.

[13] Youcef Saad and Martin H Schultz. “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.

[14] Philippe Spalart and Steven Allmaras. “A one-equation turbulence model for aerodynamic flows”. In: *30th aerospace sciences meeting and exhibit*. 1992, p. 439.

[15] Yong Su Jung, Ganesh Vijayakumar, Shreyas Ananthan, and James Baeder. “Local correlation-based transition models for high-Reynolds-number wind-turbine airfoils”. In: *Wind Energy Science* 7.2 (2022), pp. 603–622.

[16] M Costenoble, B Govindarajan, Y Jung, and J Baeder. “Automated Mesh Generation and Solution Analysis of Arbitrary Airfoil Geometries”. In: *Journal of Aircraft* 59.5 (2022), pp. 1–16.

[17] Eduardo Paluzo-Hidalgo, Rocio Gonzalez-Diaz, and Miguel A Gutiérrez-Naranjo. “Two-hidden-layer feed-forward networks are universal approximators: A constructive approach”. In: *Neural Networks* 131 (2020), pp. 29–36.