Received: 29 October 2023 | Revised: 21 February 2024 | Accepted: 7 May 2024
DOI: 10.1002/we.2918

RESEARCH ARTICLE

WILEY

A novel approach to inverse design of wind turbine airfoils using tandem neural networks

Apurva Anand | **Koushik Marepally** | **M Muneeb Safdar** | **James D. Baeder**

Department of Aerospace Engineering, University of Maryland, College Park, Maryland, USA

Correspondence
Apurva Anand, Department of Aerospace Engineering, University of Maryland, College Park, MD, USA.
Email: apurva01@umd.edu

Funding information
This work was funded by Maryland Energy Innovation Institute, University of Maryland.

Abstract

The performance of a wind turbine and its efficiency majorly depends on wind-to-rotor efficiency. The aerodynamic design of the wind turbine blades using high-fidelity tools such as adjoint-computational fluid dynamics (CFD) is accurate but computationally expensive. It becomes impractical when the number of design variables increases for multidisciplinary optimization (MDO). Low-fidelity tools are computationally cheaper but are not accurate, especially in regions of adverse pressure gradient and reverse flows. Surrogate modeling has been used in many aerodynamic problems. We develop and apply a recent architecture of the deep learning module, tandem neural networks (T-NNs) for the inverse design of wind turbine airfoils. The T-NNs trained on CFD data for fully turbulent cases predict not only the performance parameters for the given airfoil geometry but also the airfoil geometry for a given design objective. This framework uses the entire performance polar for inverse design which ensures that the airfoil optimization is not a single-point optimization problem which is essential for practical design problems. The T-NNs are also optimized to include multiple constraints like maximum thickness and trailing edge (TE) thickness which is a novel contribution in the field of inverse design using surrogate models. A statistical analysis is also performed to predict a family of airfoil geometries.

KEYWORDS
airfoil optimization, artificial neural networks, inverse design, surrogate modeling, tandem neural networks

1 | INTRODUCTION

The optimization of wind turbine rotors 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 wind turbine blade design and plays a crucial role in improving the performance of wind turbines. The blade design is a complex optimization problem that involves multiple fields of study, such as aerodynamics, structures, and acoustics.[1] High power output and low thrust are design requirements for wind turbine blades. The standard wind turbine blade design involves designing the blades using predefined airfoil geometries to capture the required power under steady-state conditions.[2] It is, therefore, imperative that the airfoils of wind turbine blades should be optimized in such a way that these design requirements are satisfied for higher efficiency and lower operational cost. Conventional airfoil design methods involve iterative optimization of the shape using less accurate low-fidelity modeling techniques which do not account for certain physical phenomena, especially in the regions of adverse pressure gradient and reverse flows. Some design processes also involve adjoint-based computational fluid dynamics (CFD), which is more accurate but demanding in terms of computational cost. The high-fidelity CFD simulations are very exhaustive computationally and can take up to thousands of core hours.[3,4] Additionally, using multidisciplinary optimization (MDO) for full blade design across various operating conditions with adjoint-based CFD methods becomes impractical.

To accelerate the design process, there have been several studies in the field of data-driven surrogate modeling which approximate the CFD models (Ref.[1,5–7]). With lower computational costs, these models also explore a wide range of design spaces. Some common surrogate models include polynomial expansions,[8] kriging,[9] radial basis functions,[10] support vector machines,[11] and artificial neural networks (ANNs).[12]. Among these, ANNs have become one of the most popular surrogate models as these are highly flexible and have been shown to perform exceptionally well for highly nonlinear function approximations. ANNs are capable of recovering smooth functions with a sufficient number of hidden layers, neurons, and activation functions. ANNs have also been coupled with standard optimization algorithms like genetic algorithms (GA),[12–14] where ANNs model the forward relationship of airfoil geometry to performance polar computation, and GA does the inverse design.

This work focuses on the development of a novel approach for the inverse design of wind turbine airfoils with practical design constraints using a special type of deep learning modules, tandem neural networks (T-NNs).[15] T-NN is a new development[6,13] in the field of aerodynamic design. It is an extension of 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,[1,5,12,13] they exhibit loss of accuracy when used for inverse design problems, due to inverse scattering. This problem arises due to the noninvertibility of the aerodynamics, that is, similar load characteristics can be achieved by different airfoil geometries. T-NNs overcome this limitation by defining the cost function such that the neural network 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 output 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 [TE] thickness, stall margin, etc.) during optimization. T-NNs architecture makes the inverse design process flexible enough to include multiple design constraints, making the inverse design process more practical for designers. The details of these constraints are explained in Section 5.2.

It is also remarked that the T-NNs architecture for inverse design uses the entire performance polar which also ensures that the inverse design is not a single-point optimization problem which is practical way of designing airfoils. 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. These models can additionally be used to create accurate databases for a wide variety of wind-turbine airfoils as a push-button solution. T-NN has both forward and inverse design capabilities and once trained can give polar as well as the airfoil design. The details of statistical analysis are included in Section 6.2.

The remainder of the paper is structured as follows: Section 2 provides the technical background and methodology employed in generating airfoil database and design space exploration techniques. Section 3 provides a comprehensive study in neural networks training and inverse design framework, while the details of neural network training and computational cost are presented in Section 4. A detailed study in improvement of T-NNs architecture is presented in Section 5, and results of design problems and statistical analysis are explained in Section 6 followed by conclusion and future works in Section 7.

2 | TECHNICAL APPROACH AND METHODOLOGY

This section sequentially describes the technical approach and methodology employed in airfoil geometry parametrization, design space exploration employed for generating airfoil database, and deterministic CFD solver to generate the training data.

2.1 | Airfoil geometry parametrization

The current study relies on CFD-trained neural networks for the inverse design of wind turbine airfoils for a range of operating conditions. The airfoil geometries are parameterized by the class function - shape function transformation (CST),[16] where Chebyshev polynomials are used as shape functions in this work instead of Bernstein polynomials proposed in the original CST methodology.[13]

There are multiple ways of parametrizing the airfoils; for instance, the American convention using camber and thickness,[13] and parametrizing upper and lower surfaces separately.[1,12]. The airfoil parametrization using camber and thickness is associated with an inverse problem and requires an iterative optimization to determine the defining geometry parameters. The addition of this iterative step increases the computational cost ($\mathcal{O}(10^1$ seconds$)$ for each airfoil) making it difficult to be used as a 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 costs. Glaws et al[1] show such parametrization using Bernstein polynomial of eighth-order. In recent work, Shalu et al[12] show 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 Figure 1, which clearly indicates that the first few coefficients of Chebyshev polynomials carry the maximum information unlike Bernshtein polynomials since Chebyshev polynomials are near-orthogonal.

However, the parametrization using upper and lower surfaces requires a nontrivial constraint on the thickness to ensure the resulting geometry is an airfoil. Instead, for ease of implementation, a modified airfoil parametrization technique comprising 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 nonnegative thickness ensures no intersecting surfaces in the airfoil during the inverse design. This way of airfoil parametrization is also done using eighth-order Chebyshev polynomials for both upper surface and thickness. This effectively gives nine polynomial coefficients each for upper surface and thickness, totalling 20 parameters including the TE 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). Equations (1) and (4), 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 TE 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. Equation (5) shows the lower surface distribution obtained from the parametrization.

$$\zeta_U(\psi) = C(\psi)S_U(\psi) + \psi\zeta_{TE,U} \qquad \textit{class} - \textit{shape function} \tag{1}$$

$$C(\psi) = \sqrt{\psi} \times (1-\psi) \qquad \textit{class} - \textit{function} \tag{2}$$

$$S(\psi) = \sum a_i S_i \qquad \textit{shape} - \textit{function} \tag{3}$$

$$\Delta\zeta(\psi) = C(\psi)S_{\text{thn}}(\psi) + \psi\Delta\zeta_{TE} \tag{4}$$

$$\zeta_L(\psi) = \zeta_U(\psi) - \Delta\zeta(\psi) \tag{5}$$

2.2 | Design space exploration

To generate a representative database of wind turbine airfoils, airfoil sections defining industrially relevant wind turbine blades as well as reference wind turbine airfoil sections of varying thickness are selected. Ultimately, we downsampled this to a representative baseline set of eleven airfoils, listed in Table 1. These airfoils have thickness distributions ranging from 18%–36%, encompassing a wide spread of potential geometries.

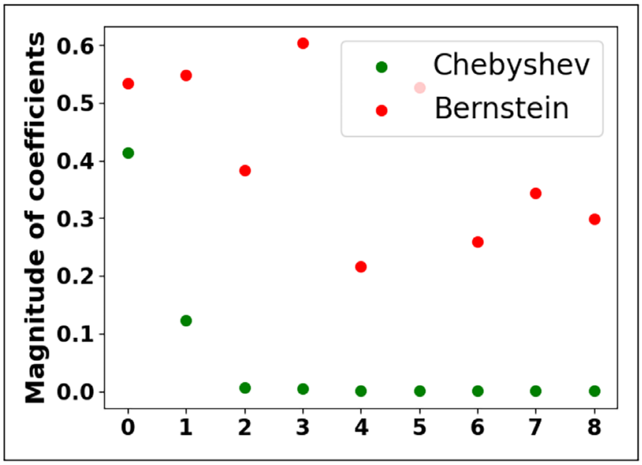

FIGURE 1 Comparison of magnitude of class function - shape function transformation (CST) coefficients for upper surface parametrization using Bernstein and Chebyshev polynomials.

These baseline airfoils are shown in Figure 2 which are represented by black lines. To train the neural networks and fill the geometric design space, the design parameters representing the upper surface coefficients and thickness coefficients are perturbed, up to ±20% and ±10%, respectively. A total of 40 permutations per baseline airfoil were generated using Latin-Hypercube sampling,[17] resulting in 440 (11 × 40) total perturbation airfoils in the geometric design space. The left of Figure 2 shows all perturbations of baseline airfoil geometries using Chebyshev coefficients along with the baseline airfoils, which completes the geometry input for training data. It is emphasized that the airfoil is parametrized using Chebyshev coefficients which are near-orthogonal. This way of parametrizing the airfoils also ensures that a lesser number of perturbations per baseline airfoil are required to fill the geometric design space. Glaws et al[1] and Marepally et al[5] use the training data that were generated by perturbing the Bernstein coefficients to 80 perturbations per baseline airfoils as shown in the right of Figure 2. From Figure 2, it can be inferred that the 40 perturbations of the baseline airfoils using Chebyshev polynomials cover similar geometric space as the 80 perturbations of Bernstein polynomials for the same baseline airfoils. Having a lesser number of perturbations also reduced the computational cost of data generation by half (33,000 h for this study).

Therefore, an efficient and accurate perturbation strategy is employed to generate the training data. The details of the accuracy of training data are explained in Section 3.

2.3 | 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[18] and uses a fifth-order WENO scheme[19] for spatial reconstruction, with Roe's approximate Riemann solver for inviscid flux, second-order central differencing for viscous flux, and a preconditioned GMRES[20] for implicit time integration. The mesh is generated using an automated grid generation tool for airfoils.[21] The mesh for all the cases are unstructured with All-Quad elements. The initial wall normal spacing of y+=1 is used and the far-field boundary at 300c. Figure 3 shows the sample grid generated on DU97-W-300LM airfoil.

HAM2D is extensively tested and validated with various airfoil data available in the literature.[22] For the previous studies, the turbulent boundary layer was modeled using the Spalart–Allmaras (SA) model,[23] which is a one equation model that solves for modified eddy viscosity ν_t. For this paper, we use strain adaptive linear Spalart–Allmaras (SA-noft2-salsa)[24] turbulent model with low Reynolds number correction. The combination of two variants of SA showed significant improvement in poststall regions. The details of these turbulence models and the comparison with the experimental data are presented in Appendix A.

TABLE 1 List of baseline wind turbine airfoils.

Wind turbine airfoils	
DU99-W-350LM	DU97-W-300LM
DU91-W2-250LM	DU93-W-210LM
NACA64-618	FFA-W3-211
FFA-W3-241	FFA-W3-270blend
FFA-W3-301	FFA-W3-330blend
FFA-W3-360	

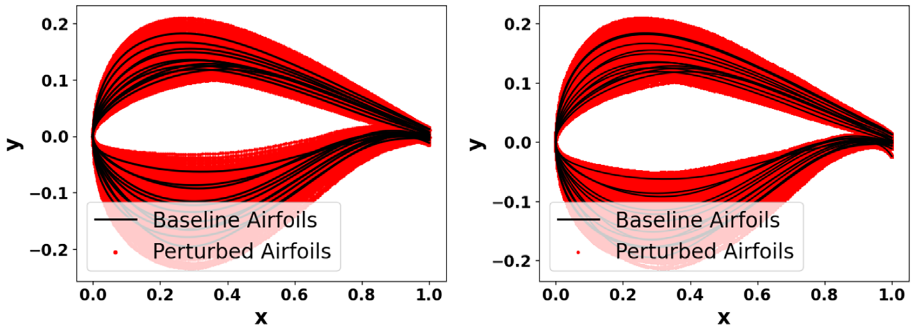

FIGURE 2 Baseline and perturbed airfoils, left: perturbation of Chebyshev polynomials, right: perturbation of Bernstein polynomials.

3 | MACHINE LEARNING FRAMEWORKS

This section presents the details of deep learning frameworks used for both the forward computations (airfoil geometry is input, and performance polar is output) and inverse design (performance polar is input, and airfoil geometry is output).

3.1 | ANNs

One of the objectives of developing the deep learning module 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, lift-to-drag ratio, and pitching moment coefficients are fundamental design quantities. ANNs architectures, due to their highly accurate nonlinear function approximation capabilities,[25] are used for the surrogate model development in this study.

There are two ways of training the ANNs. One is to use CST coefficients and angle of attack (α) as inputs with C_l, C_d, and C_m as outputs. Most papers[1,5] on surrogate modeling use this strategy for training the ANNs. The second way of training the ANNs is by vectorizing the polar such that the entire polar of C_l, C_d, and C_m is vectorized as output and α is not a part of training.[13] It is emphasized that the proper orthogonal decomposition (POD) of C_l, C_d, and C_m is used in training, and the cost function is also defined for the POD values. Since the POD makes the individual polar orthogonal, the first few modes are the most dominant and majorly contribute to the loss function. A comparison of estimates of accuracy for ANNs using the nonvectorized and vectorized polar is presented in Figures 4 and 5, respectively. This way of training not only ensures higher

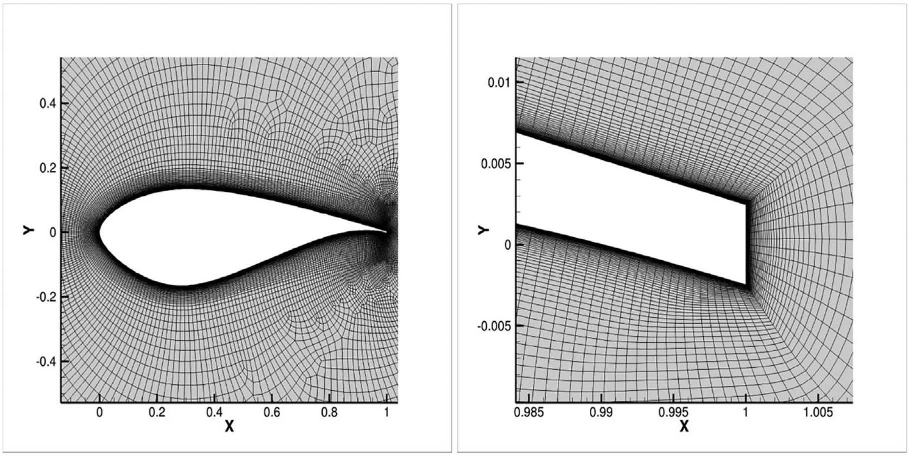

FIGURE 3 Sample DU97-W-300LM airfoil mesh. Left: mesh around DU97-W-300LM airfoil. Right: mesh at finite thickness trailing edge.

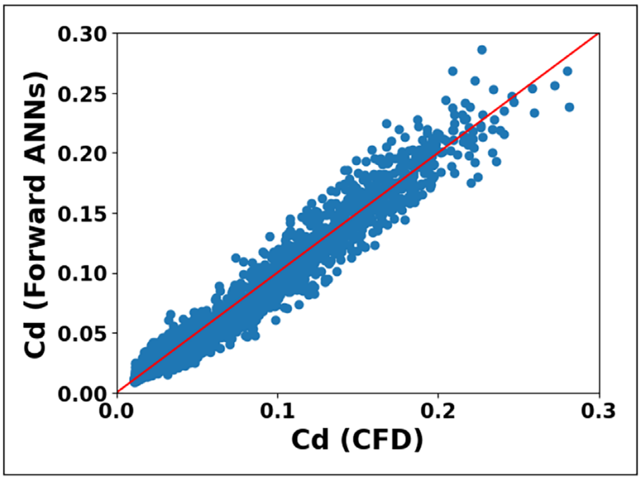

FIGURE 4 Accuracy of forward artificial neural networks (ANNs) trained with nonvectorized polar; comparison of true C_d (computational fluid dynamics [CFD]) with FWD. ANNs predicted C_d

accuracy but also a necessity for the inverse design, the details of which are explained in Section 3.2. Therefore, for all further studies, the vectorized polars are used in the ANNs without having to use the angles of attack as inputs. Figure 6 shows the forward surrogate model architecture used in this study. The inputs of the surrogate model represent the CST coefficients that define the airfoil geometry denoted by C, and the outputs are the performance parameters denoted by P (C_l, log(C_d), and C_m), 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 that the drag predicted by the neural networks is always positive. This approach is consistent with all the studies presented in this paper.

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 is not necessary. However, the

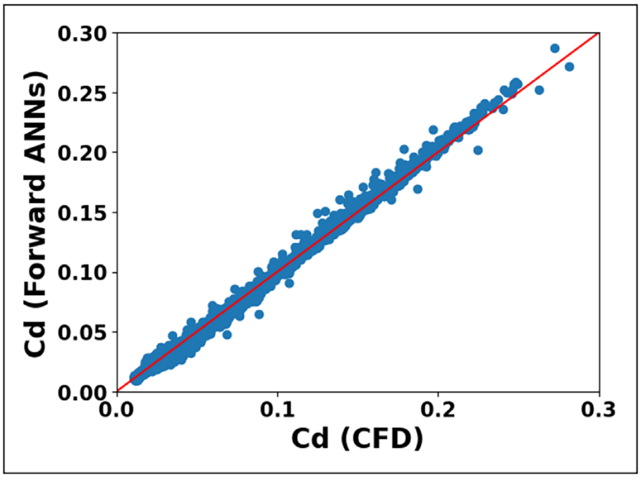

FIGURE 5 Accuracy of forward artificial neural networks (ANNs) trained with vectorized polar; Comparison of true C_d (computational fluid dynamics [CFD]) with fwd. ANNs predicted C_d

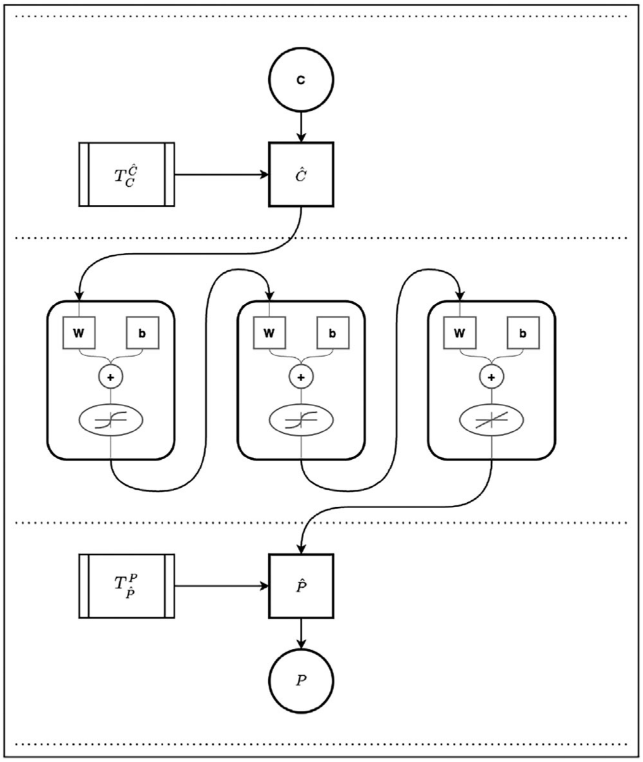

FIGURE 6 Architecture of artificial neural networks (ANNs) for forward problem.

POD of the output features (performance polar) becomes essential. For a given training data, the output is a 75-dimensional vector (25 values each of C_l, $\log(C_d)$ and C_m), which makes it even more necessary to make them orthogonal. This is done by systematic POD on individual polar such that the first five modes carry more than 99% of the information. This is quantitatively estimated using percentage explained variance for each polar. As shown in Figures 7–9, first five POD values of each polar hold more than 99% of the information, and therefore, these first five mode contributes majorly to the loss function. The ANN, once trained, gives POD of the polar as outputs which is then retransformed to the original dimension to extract original polar. It is emphasized that the POD is done after scaling the polar of C_l, $\log(C_d)$, and C_m. The cost function for the ANNs is defined by Equation (6). The forward ANNs are very accurate as presented in Figure 5. The quantitative estimate of error is presented in Section 3.2.

$$J=\frac{1}{\#\,\mathrm{Train}}\sum_{\mathrm{Train}}\left(P_{\mathrm{True}}-P_{\mathrm{Predicted}}\right)^2 \tag{6}$$

For inverse design, inverting the forward ANNs as shown in Figure 10, to predict an airfoil geometry for a given performance polar, is not accurate as presented in Figure 11. 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, that is, the same performance polar can be given by multiple airfoil geometries and therefore the one-to-one mapping of the inverse problem is not physical. This nonunique 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. The cost function for I-ANNs is defined by Equation (7).

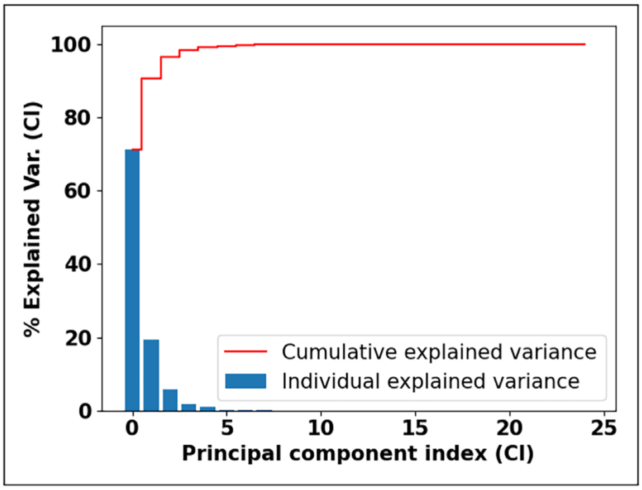

FIGURE 7 % Explained variance ratio after proper orthogonal decomposition (POD) for C_l for training dataset.

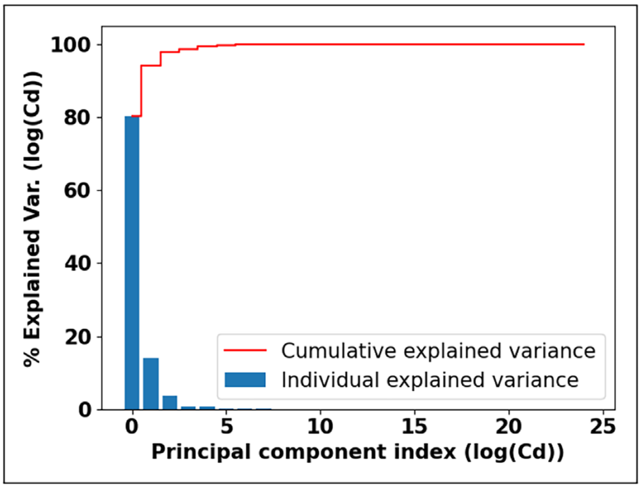

FIGURE 8 % Explained variance ratio after proper orthogonal decomposition (POD) for $\log(C_d)$ for training dataset.

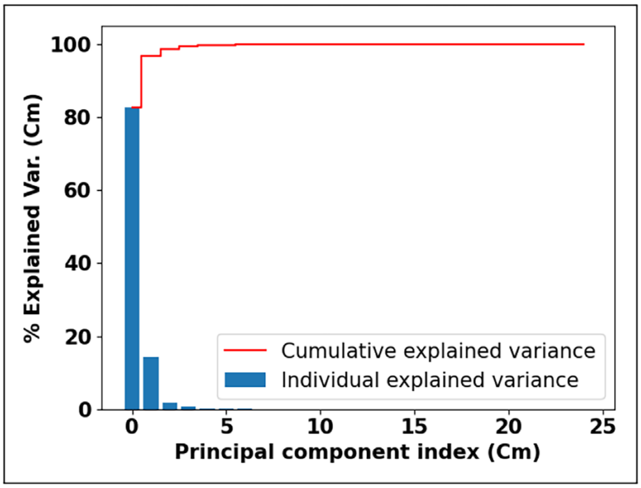

FIGURE 9 % Explained variance ratio after proper orthogonal decomposition (POD) for C_m for training dataset.

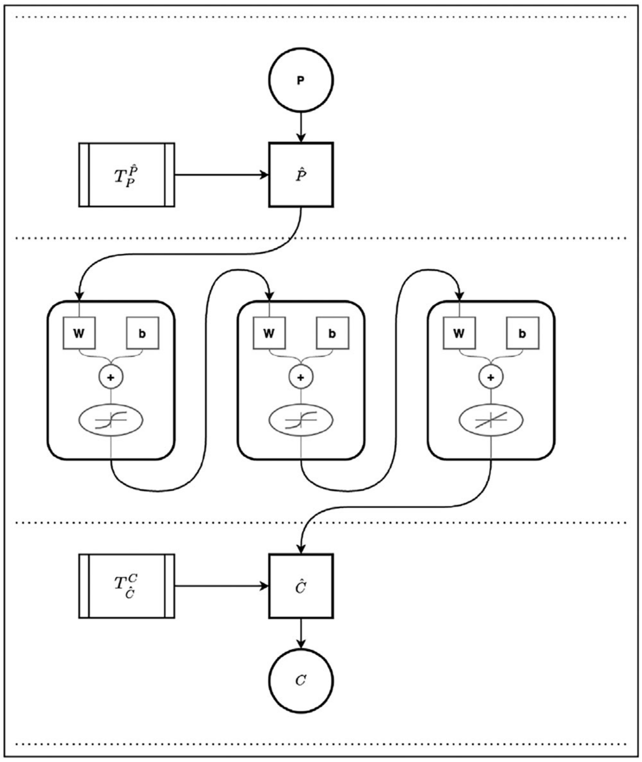

FIGURE 10 Architecture of inverted artificial neural networks (I-ANNs) for inverse problem.

$$J = \frac{1}{\#\,\mathrm{Train}} \sum_{\mathrm{Train}} \left(C_{\mathrm{True}} - C_{\mathrm{Predicted}} \right)^2 \tag{7}$$

3.2 | T-NNs

The problem of inverse scattering is tackled using a special type of deep learning module, T-NNs. T-NNs architecture has two ANNs working in series as presented in Figure 12. The bottom part of the T-NNs is the pretrained forward ANNs (highlighted by blue in Figure 12). These forward ANNs take airfoil CST coefficients and predict performance polar as explained in Section 3.1. It is emphasized that the forward ANN is pretrained, that is, the weights of the forward ANNs are frozen. The top portion of the T-NNs has an inverse ANNs (I-ANNs) which takes performance polar as inputs and gives airfoil CST coefficient as outputs. The entire architecture of the T-NNs is trained taking performance polar as input to the first

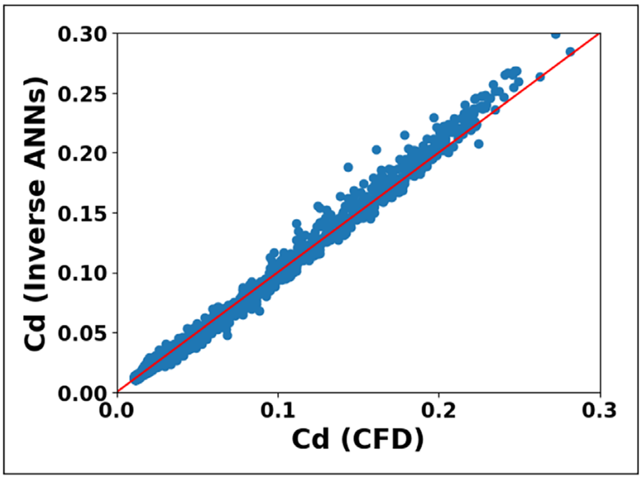

FIGURE 11 Accuracy of inverted artificial neural networks (I-ANNs) trained with vectorized polar; comparison of true C_d (computational fluid dynamics [CFD]) with forward ANNs predicted C_d

network (highlighted in green) which gives CST coefficients as outputs in the intermediate layer. This CST coefficient is fed as input to the second network (highlighted by blue), which gives performance polar as output. During training, the weights of the first network (highlighted in green) are updated but the weights of the second network (highlighted in blue) are frozen. The cost function of the T-NNs is defined by Equation (8). From Equation (8), it can be seen that the cost function is defined based on the performance polar and not the airfoil CST coefficients. This is particularly important to minimize the inverse-scattering problem because this way of defining the cost function does not require the T-NNs to predict the same airfoil geometry, rather it generates an airfoil geometry which gives the same performance polar. It is also noteworthy that the cost function for T-NNs (Equation (8)) 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 errors of predicted and original aerodynamic coefficients for the I-ANNs and T-NNs are presented in Table 2. The L2 norm is computed for the 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 13 is better than that of I-ANNs (Figure 11). The comparison of the performance of T-NNs and I-ANNs is done using the vector plots of drag coefficients as shown in Figures 11 and 13. The vector plots of the lift coefficient and pitching moment coefficient follow similar trends as those of the drag coefficient.

$$J=\frac{1}{\mathrm{Train}}\sum_{\mathrm{Train}}\left(P_{\mathrm{True}}-P_{\mathrm{Predicted}}\right)^2 \tag{8}$$

4 | NEURAL NETWORK TRAINING AND COMPUTATIONAL COST

The neural networks are trained using the TensorFlow library. 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 (number of neurons, number of layers, etc.), and points are randomly searched in the domain. The number of neurons in each hidden layer of forward ANNs and T-NNs are mentioned below:

1. **Forward ANNs:** 20(Inputs)-25-25-50-75-75(Outputs)
2. **T-NNs:** 75(Inputs)25-20-20-30-20 + Forward ANNs

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 that there is no overfitting, the details of which is also presented in Figure 14. Deep networks are prone to overfitting, and therefore, standard L2 regularization is used during training to penalize large weights by adding a penalty term in the loss function of the neural networks to prevent overfitting. Equation (9) represents the regularized loss after adding the regularization term.

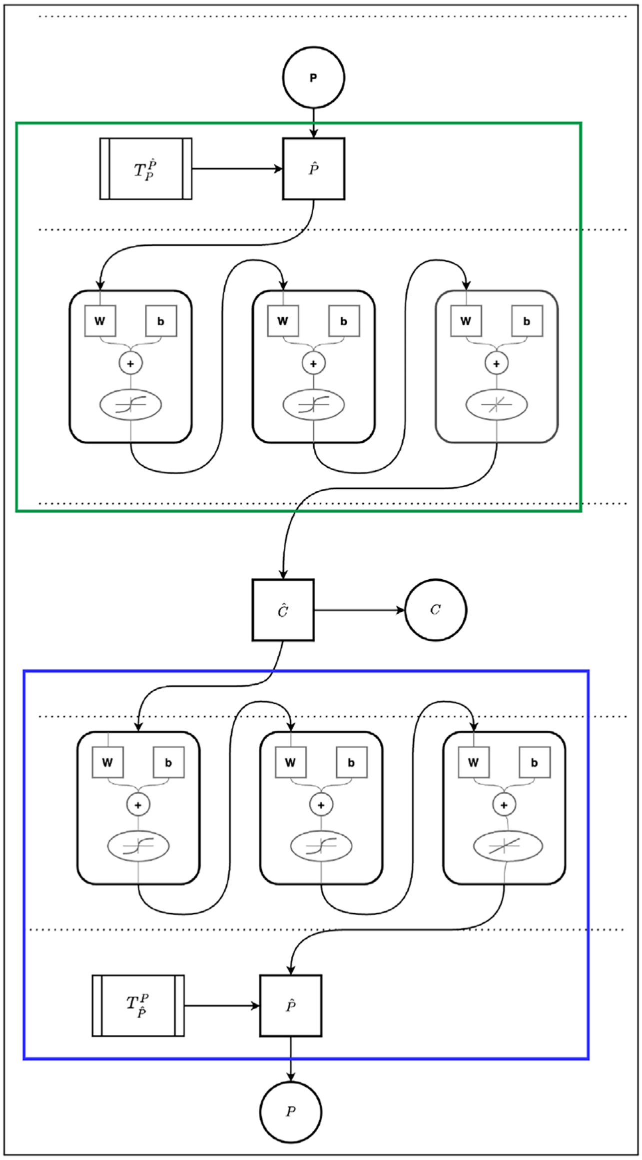

FIGURE 12 Architecture of tandem neural networks (T-NNs) for forward and inverse problem.

TABLE 2 Comparison of quantitative error estimates of the I-ANNs and T-NNs for test data.

Aerodynamic coefficients	L_2 norm of I-ANNs	L_2 norm of TNNs
C_l	5.51×10^{-2}	2.50×10^{-2}
L/D	2.78	1.11
C_m	8.44×10^{-3}	3.49×10^{-3}

Abbreviations: I-ANN, inverted artificial neural networks; T-NN, tandem neural networks.

$$\textit{Regularized Loss} = \textit{Original Loss} + \lambda \times \sum_{n} w^2 \tag{9}$$

In Equation (9), λ is the strength of the regularization parameter. This ensures that during the optimization process, smaller weights are favored leading to a more simple model. For our ANNs, a range of λ varying from 0.1 to 0.0001 is tested, and the best among them is chosen for all the cases. Adam optimizer is used to optimize the weights as it's an adaptive learning rate optimization algorithm and is efficient for deep learning modules.

Figure 14 shows the comparison of 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 ANNs and T-NNs are trained for 10,000 epochs.

The comparison of computational cost using CFD and neural networks is presented in Table 3. The difference in computational cost of CFD with 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.

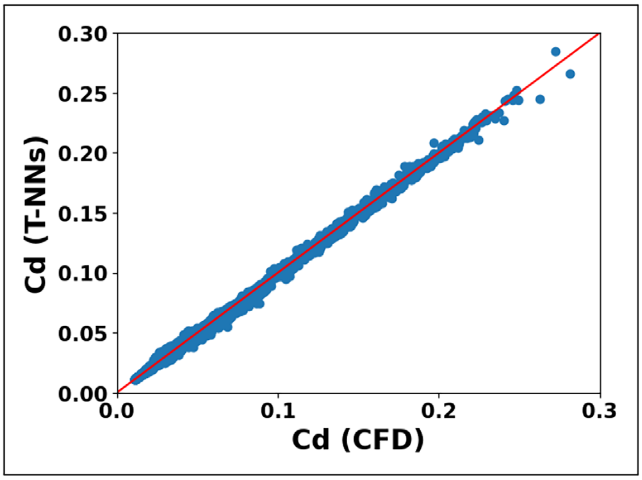

FIGURE 13 Accuracy of tandem neural networks (artificial neural networks [ANNs]); comparison of true C_d (computational fluid dynamics [CFD]) with fwd. Tandem neural networks (T-NNs) predicted C_d

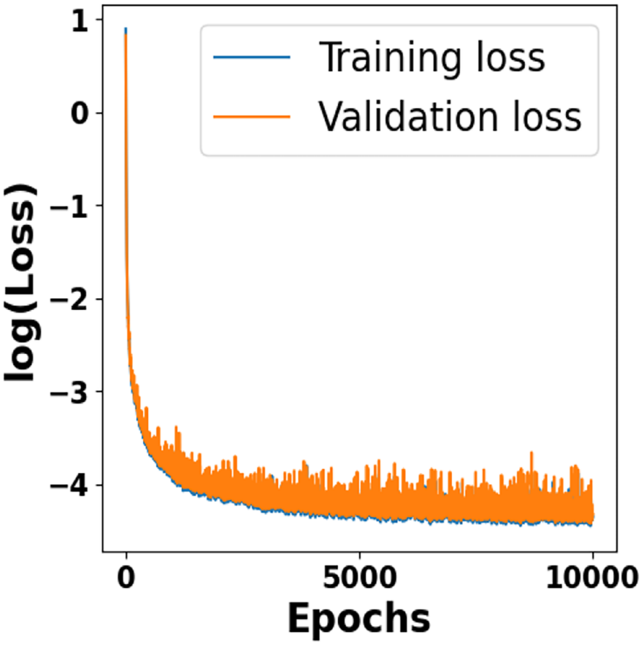

FIGURE 14 Comparison of log(Loss) versus epochs for both training and validation data for tandem neural networks (T-NNs)

5 | IMPROVEMENT OF T-NNS ARCHITECTURE

5.1 | Improved loss function

To validate our hypothesis that T-NNs solves inverse scattering problem by defining the loss function interms of polars rather than airfoil CST coefficients, a comprehensive study of the loss function is performed. The improved loss function has both CST coefficients as well as performance polar with a fractional weight added to both. This fractional weight is represented by a and b as presented in Equation (10). 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 for test data, and it can be seen that the conventional loss function of T-NNs (a = 0) is the most accurate for the test data which is not a part of training. And therefore, it validates our hypothesis of T-NNs solving the inverse scattering problem. The same loss function is used in all further studies.

$$J = \frac{a}{\mathrm{Train}} \sum_{\mathrm{Train}} \left(C_{\mathrm{True}} - C_{\mathrm{Predicted}}\right)^2 + \frac{b}{\#\,\mathrm{Train}} \sum_{\mathrm{Train}} \left(P_{\mathrm{True}} - P_{\mathrm{Predicted}}\right)^2 \tag{10}$$

5.2 | Multiple design constraints

The architecture of T-NNs provides flexibility in terms of multiple design constraints. Without design constraints, a designer has no control over the predicted airfoil geometries and the airfoils can be either much thicker or thinner than the baseline, which is not a very practical way of solving the inverse-design problem. The additional design constraints allow the designer to pre-specify the geometric constraints like maximum thickness, TE thickness, and so on.

A noble contribution of this work is the implementation of realistic geometric design constraints for the inverse design problem. As a test case, we use maximum thickness and TE thickness as design constraints. Implementing TE thickness constraint is straightforward as this is one of the CST coefficients. However, the implementation of the maximum thickness constraint is relatively complicated as this is not directly included in the CST parametrization. To implement maximum thickness as a geometric constraint, a function approximation is introduced in the intermediate layer as shown in Figure 15. This function approximation can either be an inverse CST transformation (which computed maximum thickness from CST coefficients) or a separate neural network. For ease of implementation, we use ANNs as a function approximation which relates CST coefficients to maximum thickness and TE thickness. The neural network for function approximation is trained offline (i.e., pretrained with frozen

TABLE 3 Comparison of computational cost for generating performance parameters using CFD and neural networks.

Computational task	Computational cost (approx. hours)
CFD training data generation per Re	33,325
Train forward ANNs	0.20
Train TNNs	0.30
CFD simulation for single airfoil at 1 Re	3.00
Evaluate single airfoil using ANNs	8.33×10^{-8} (0.3 ms)

Abbreviations: ANN, artificial neural network; CFD, computational fluid dynamics; T-NN, tandem neural networks.

TABLE 4 Comparison of L2 norm for improved loss function for T-NNs for test data.

a	b	L_2 norm of C_l	L_2 norm of L/D	L_2 norm of C_m
0	1	2.50×10^{-2}	1.11	3.49×10^{-3}
0.2	0.8	3.02×10^{-2}	1.56	3.53×10^{-3}
0.3	0.7	3.23×10^{-2}	1.71	5.13×10^{-3}
0.5	0.5	3.52×10^{-2}	1.84	4.11×10^{-3}
0.7	0.3	4.12×10^{-2}	2.25	5.10×10^{-3}
0.8	0.2	4.32×10^{-2}	2.37	5.15×10^{-3}
1	0	7.48×10^{-2}	3.41	11.3×10^{-3}

Abbreviation: T-NN, tandem neural networks.

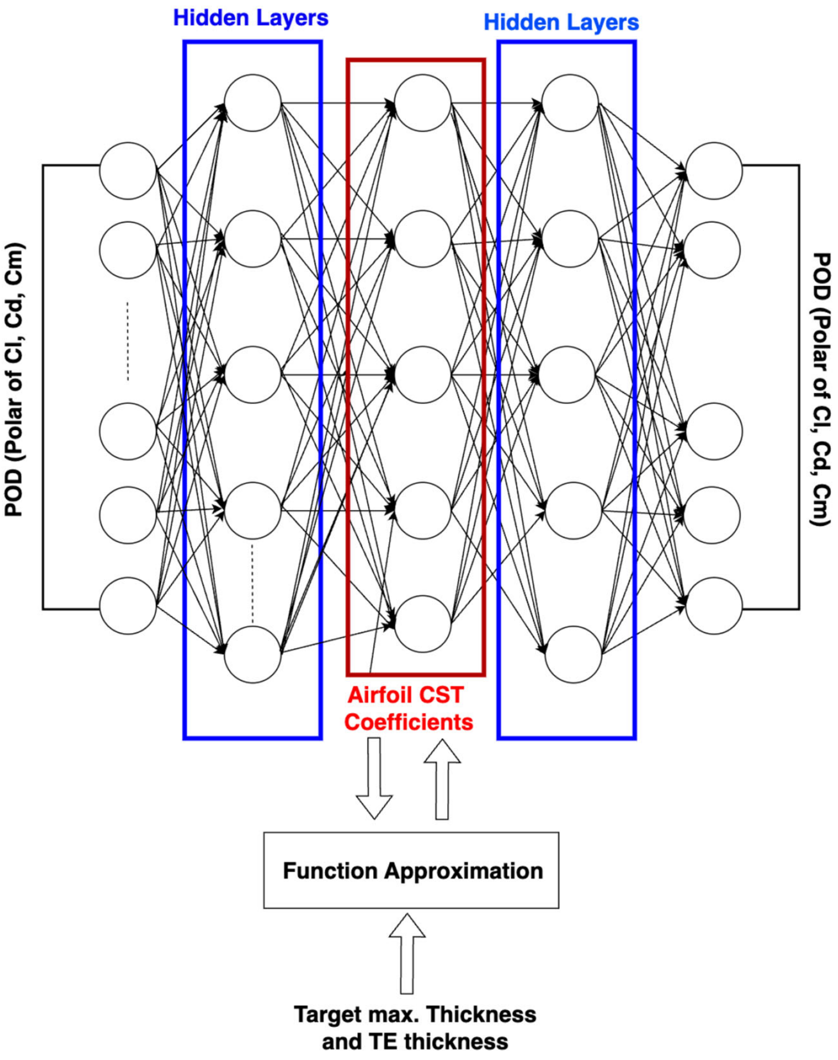

FIGURE 15 Tandem neural networks (T-NNs) architecture for inverse design with maximum thickness and trailing edge (TE) thickness design constraints in input, intermediate and output layers.

weights) which makes the constraint implementation more rigorous and practical. Despite a rigorous constraint on thickness, the neural networks have some associated errors, and therefore, the inverse-designed airfoil can have a slightly different maximum thickness than the target. The details of the implementation of these constraints and the difference in target thickness with the baseline are explained in Section 6.1.2. 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 nonnegative weight constraint is added to the layer that predicts TE thickness in addition to having a linear activation function. The nonnegative 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 AND DISCUSSIONS

6.1 | Design problems

6.1.1 | Increasing L/D of DU93-W-210LM airfoil from 5-15% with similar lift and stall characteristics

The objective of this design problem is to increase the lift-to-drag ratio of DU93-W-210LM airfoil from 5%–15% for all angles of attack while retaining the C_l and C_m characteristics as that of baseline airfoil, i.e., decreasing the drag coefficient of the airfoil. Therefore, the design objective is to optimize the airfoil such that the lift-to-drag ratio increases without changing the lift and moment polar. The L/D of the baseline DU93-W-210LM airfoil is modified such that the target L/D is changes as shown in Table 5.

For these five cases, the T-NNs predict five different airfoil geometries all of which are shown in top left of Figure 16. It is to be noted that, for this design problem, no thickness constraint is enforced, that is, the maximum thickness of predicted airfoils is allowed to change. From Figure 16, it can be seen that the five predicted airfoil geometries not only have slightly lesser thickness (varying from 20.5% to 19.4%) than the baseline, but their camber also changes considerably. It is interesting to note that the lower surface of all these airfoils is different from the baseline, and the position of maximum thickness moves towards the leading edge for all five cases. The TE thickness for all these airfoil geometries is positive. While the maximum thickness for these airfoils is slightly lesser (between 19.4% and 20.5% for all five cases) than the baseline, there are no overlapping surfaces.

The corresponding lift polar is presented in top right of Figure 16, and it can be observed that the lift coefficient for all the design cases are very close to the baseline. Similarly, the lift-to-drag ratio is presented in the bottom right of Figure 16, and it can be seen that the lift-to-drag ratio increases for all optimized airfoils. Table 5 shows the comparison of target polar and the corresponding true polar for all five design cases. For all five cases, the actual improvement in maximum L/D is either close to the target or higher than the target. There is no significant difference in α at which L/D is maximum for all optimized airfoil geometries, and this is consistent with the inputs to the T-NNs where the stall margin is not modified. bottom left of Figure 16 shows the corresponding drag polar in the log scale.

6.1.2 | Increasing L/D of DU93-W-210LM airfoil by 10% with maximum thickness and TE thickness constraints

The objective of this design problem is to increase the lift-to-drag ratio of DU93-W-210LM airfoil by 10% for all angles of attack while retaining the C_l and C_m characteristics as that of baseline airfoil, that is, decreasing the drag coefficient of the airfoil. The difference between this design problem with the one explained in Section 6.1.1 is the imposition of the maximum thickness and TE thickness constraints as explained in Section 5.2. The major objective of this study is to understand the impact of imposing constraints on the inverse design framework.

Two different neural networks are trained, one with constraints and one without constraints. To impose the constraints for the inverse design problem, the target is defined in such a way that the optimized airfoil has the same maximum thickness and TE thickness as that of the baseline airfoil. Figure 17 presents the comparison of baseline airfoils with optimized airfoils, one with constraints and one without constraints as highlighted in red and yellow colors, respectively. The top left of Figure 17 shows the comparison of the airfoil geometries. The airfoil geometry with the constraints has the same thickness as that of the baseline (Δ t/c $= 0.01$) but the airfoil geometry without the constraint is much thinner than the baseline (Δ t/c $= 1.46$). It is also interesting to note that the inverse-designed airfoil with thickness constraint has a different camber as well as a different position of maximum thickness as can be seen in Figure 17. This is because these parameters are not included in the constraints. Similarly, the C_l polar for both cases are comparable with the baseline as shown in the top right of Figure 17. The bottom of the Figure 17 shows the comparison of $\log(C_d)$ polar and L/D polar. The improvement in L/D of the inverse-designed airfoil with and without the constraints are comparable as shown in the bottom right of Figure 17. This demonstrates the importance of using design constraints in the inverse-design architecture. The resultant inverse-designed airfoil has maximum thickness and TE thickness comparable to the baseline and has higher L/D, and therefore better performance without compromising on the geometric constraints. This way of inverse-design in noble and an important contribution of this paper.

6.1.3 | Observations from design problems

The following can be inferred from the design problems:

1. T-NNs architecture can be used for both forward predictions of polar and inverse design of airfoil.

TABLE 5 Comparison of target improvement and actual improvement for design problem 1.

Case	Target % improvement of max. L/D	Actual % improvement of max. L/D
Design 1	5	6.35
Design 2	8	9.82
Design 3	10	11.97
Design 4	12	13.97
Design 5	15	16.66

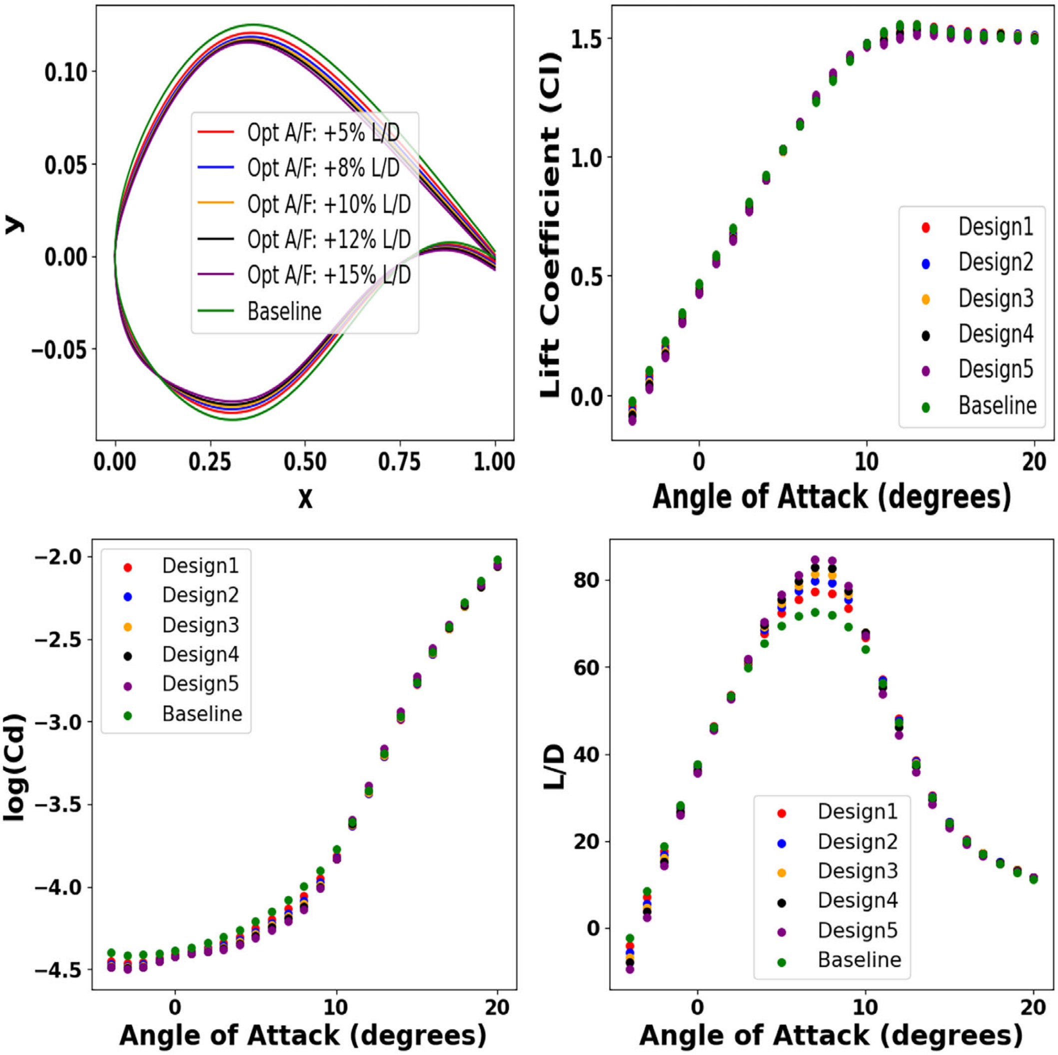

FIGURE 16 Comparison of airfoil geometries and the corresponding polar for design problem 1. Clockwise from top left; airfoil geometries, C_l polar, L/D polar, log(C_d) polar. Design 1 = 5% improvement in L/D, design 2 = 8% improvement in L/D, design 3 = 10% improvement in L/D, design 4 = 12% improvement in L/D, design 5 = 15% improvement in L/D.

2. T-NNs architecture is like a push-button solution which takes milliseconds to design one single airfoil and generate the entire polar.
3. T-NNs architecture is flexible to be used with and without the constraints
4. Addition of design constraints ensures that the predicted airfoil geometry is closer to the target, but the corresponding polar is relatively off-target.

6.2 | 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 6.1.1, there are five design objectives, and therefore, we got five airfoils. To generate a family of airfoils for a single design objective, that is, multiple airfoils all of which give similar performance, a statistical analysis is performed where a range of maximum thickness and TE thickness is chosen randomly from a Gaussian distribution, and for each of these combinations of variables, an airfoil is generated. The distribution of max. thickness and TE thickness is presented in Figure 18. The upper and lower limits for these variables (max. thickness and TE thickness) are prespecified by the designer. For the problem under consideration, the max. thickness and TE thickness are varied between 60% and 110% of the baseline DU93-W-210LM airfoil. This way of inverse design allows the designer to generate multiple airfoil geometries within prespecified 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.

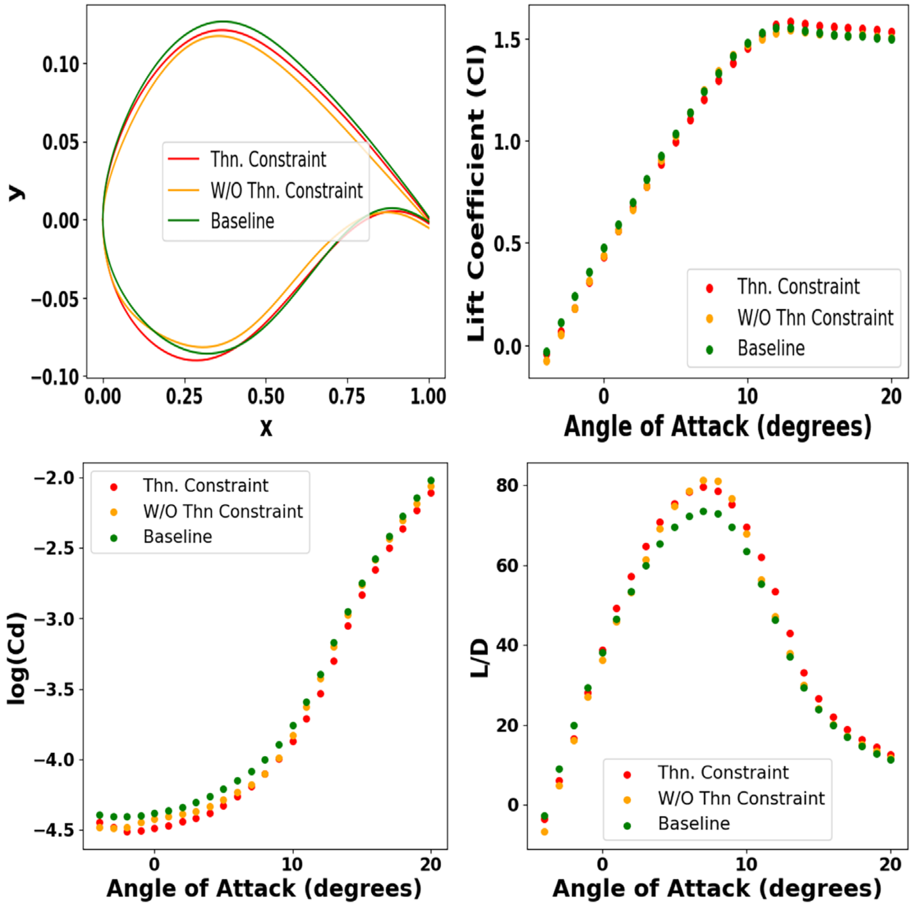

FIGURE 17 Comparison of airfoil geometries and the corresponding polar for design problem 2 (with and without thickness constraints). Clockwise from top left; airfoil geometries, C_l polar, L/D polar, log(C_d) polar.

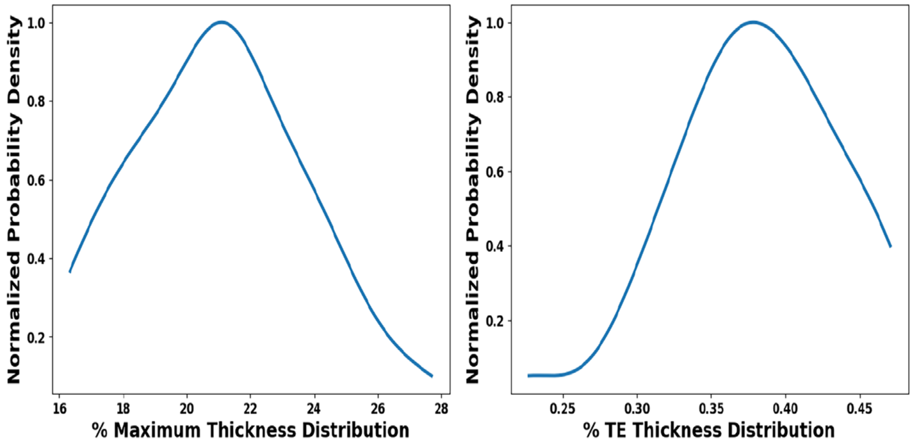

FIGURE 18 Comparison of the distribution of geometric constraints for statistical analysis. Left: distribution of max. thickness. Right: distribution of trailing edge (TE) thickness.

The geometric constraints on the maximum thickness and TE thickness are imposed, the details of which is explained in Section 6.1.2. Figure 19 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 (which is constant for this problem) are given as inputs to the pretrained 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 6.1.2. The results are demonstrated in Figure 20. The top left of Figure 20 shows 50 airfoil geometries predicted by the T-NNs for

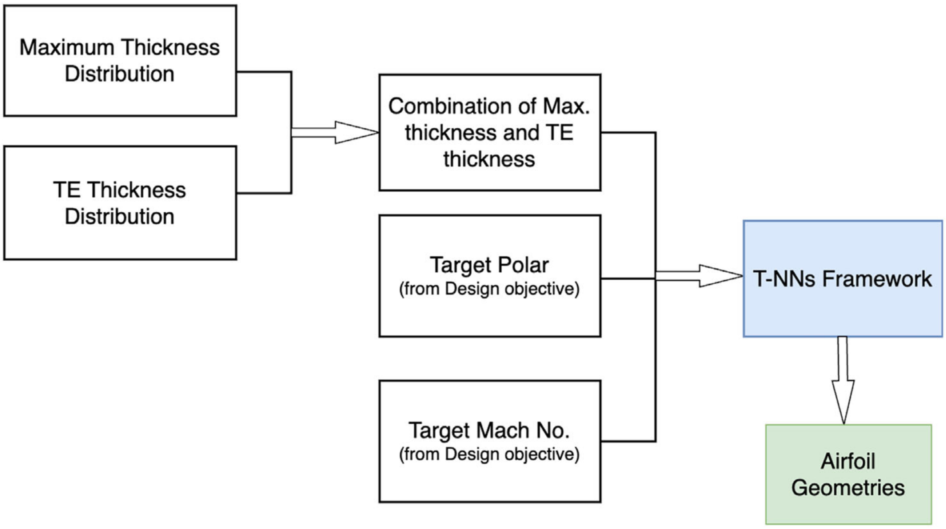

FIGURE 19 Flowchart explaining the working of statistical analysis with T-NNs; generating a family of airfoils with variable max. thickness and trailing edge (TE) thickness.

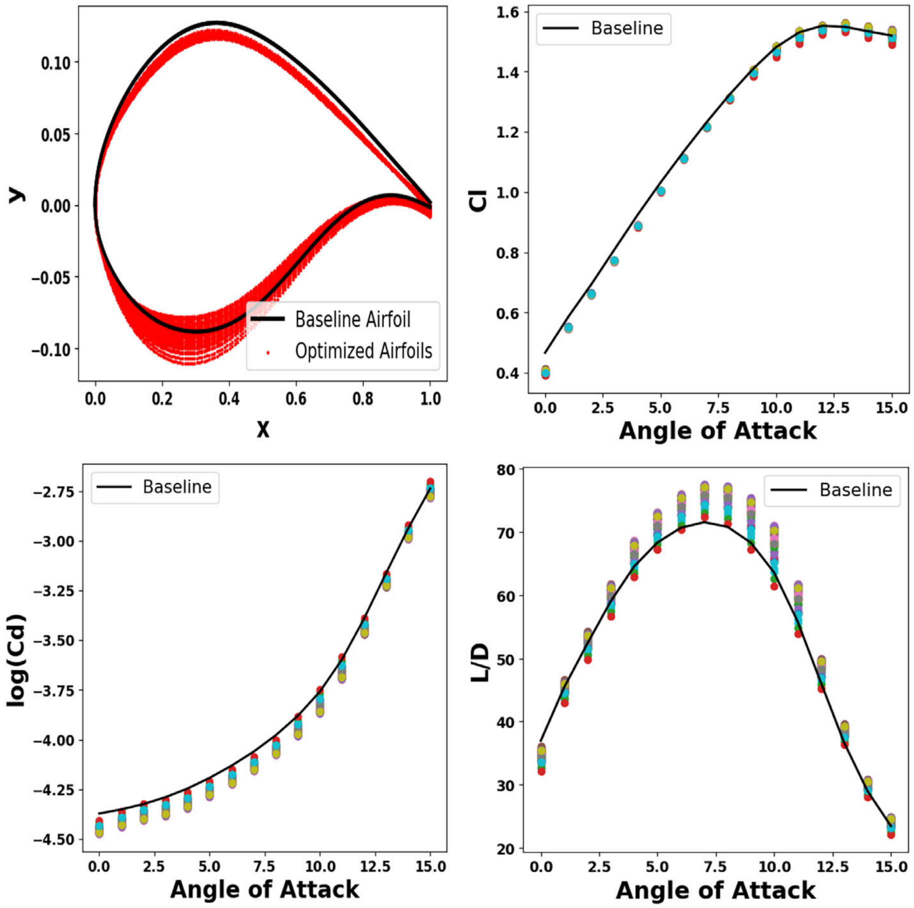

FIGURE 20 Statistical analysis to generate a family of airfoils with distribution of max. thickness and trailing edge (TE) thickness. Clockwise from top-left: Airfoil geometries, comparison of C_l polar, comparison of L/D polar, comparison of log(C_d) polar.

50 combinations of max. thickness and TE thickness as shown in Figure 18. The corresponding lift and lift-to-drag polar is also presented in Figure 20. It is interesting to note that most of the C_l polar for all predicted airfoils are close to that of the baseline airfoil, whereas some airfoils reach the target L/D polar while others do not. This is the major objective of this analysis, that is, to predict multiple airfoils for the same design

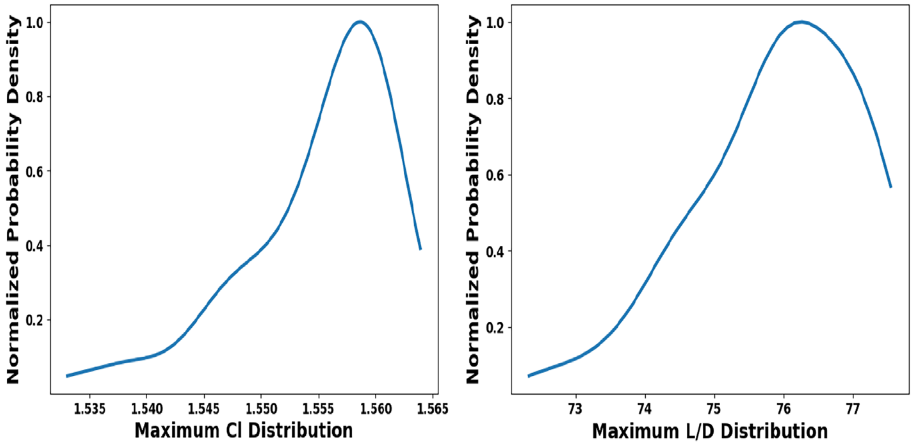

FIGURE 21 Comparison of the distribution of performance parameters of predicted airfoils. Left: distribution of max. C_l. Right: distribution of max. L/D.

objective and then choose the predicted airfoils based on the requirement. Figure 21 shows the distribution of maximum C_l and maximum L/D for 50 airfoils predicted by the T-NNs. From Figure 21, it can be seen that the maximum C_l for the predicted airfoils varies between 1.53 and 1.56, and the maximum L/D varies between 72.3 and 77.5. The change in maximum C_l for the baseline and optimized airfoil geometries lies between -1.29% and $+0.64\%$. Similarly, the change in maximum L/D for the baseline and optimized airfoil geometries is $+1.10\%$ to $+8.40\%$. It is imperative that the maximum C_l of all optimized airfoils are closer to the maximum C_l of the baseline airfoil as the target was to retain the C_l values. But the maximum improvement in L/D is close to 8.40%. For all airfoil geometries, the maximum L/D is higher than the baseline. 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 as follows:

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.

7 | CONCLUSION AND FUTURE WORKS

7.1 | Conclusions

The current study introduces a paradigm-shift in the exploration of wind turbine airfoil design, with the integration of T-NNs. This novel application leverages the inherent capabilities of neural networks and also mitigates the inverse scattering problem, serving as a versatile tool for both forward aerodynamic polar predictions and inverse airfoil design. The utilization of T-NNs in our design framework along with the methodology for enforcing geometric constraints allows for adaptability to the intricacies of aerodynamics, providing versatility in airfoil shape optimization.

A notable contribution of this work is the successful integration of design constraints within the T-NNs optimization framework. This allows designers to enforce certain geometric requirements, such as maximum thickness and TE thickness, steering the inverse design process towards practical outcomes that meet real-world engineering requirements. This not only enhances the precision of airfoil designs but also facilitates a smoother transition from theoretical models to tangible applications.

Diverging from conventional ML approach to predict the performance at a single angle of attack, the current methodology distinguishes itself by considering the entire range of polar data. This nuanced approach captures the aerodynamic behavior across the operating envelope, and the inverse design using the entire polar stands closer to the conventional airfoil design philosophy. Further, the adoption of entire polars circumvents the constraints associated with shape optimization performed at isolated angles of attack.

Another significant advancement in the T-NN methodology is the introduction of statistical analysis into airfoil design, allowing for the rapid generation of a family of airfoils for a specific design objective. Leveraging the deterministic nature of T-NNs, this approach introduces variability in design parameters, resulting in the swift generation of multiple airfoil geometries with similar performance characteristics. The statistical perspective adds a layer of flexibility to the design process, enabling exploration of diverse design possibilities and providing valuable insights for designers.

Applications of T-NNs specifically to aerodynamic design problems are presented, demonstrating a more practical and controlled inverse design process with the inclusion of design constraints, the adaptability of the framework to a spectrum of geometric constraints, and the efficiency in generating diverse airfoil shapes with similar performance characteristics. Each design problem serves as a case study, illustrating the versatility and applicability of our methodology in addressing real-world engineering challenges.

In conclusion, the current study represents a significant stride in wind turbine airfoil design methodologies. The amalgamation of T-NNs, design constraints, comprehensive polar utilization, and statistical analysis not only contributes to advancements in each specific area but also collectively propels the field towards more efficient and practical airfoil design solution methodologies.

7.2 | Future work

The possible improvements to enhance the current study are listed below:

1. Expansion of training database and design of experiments:
   The current database can be expanded to include a more comprehensive range of Mach and Reynolds numbers. Further, the effects of different aerodynamic and geometric operating parameters such as freestream turbulence intensity and surface roughness can be included. With the increasing number of design and operating parameters, suitable design of experiment strategies must be employed to control the computational requirements associated with training data generation.
2. Towards 3D - blade optimization:
   The study can be expanded to optimizing the blade design with design variables including local airfoil geometries as well as the variation of chord, twist, and sweep along the span of the blade. Suitable low-fidelity high-fidelity blending strategies become important in this scenario, owing to the computational cost associated with a single 3D CFD simulation as well as the total number of simulations required to build a training database.
3. Uncertainty quantification:
   With the T-NNs being relatively new methodology, methods for estimating the uncertainties associated with T-NN predictions must be developed. This, in the context of airfoil design problem, may translate to the sensitivity of geometric variables corresponding to a required change in performance. This becomes valuable in terms of developing more robust and novel control mechanisms to achieve a desired fluctuation in the aerodynamic performance.

ACKNOWLEDGMENTS

This research is conducted with support from the Maryland Energy Innovation Institute (MEI2). The authors would like to acknowledge Dr. Bumseok Lee from National Renewable Energy Laboratory (NREL) for creating automated python scripts for generating CFD data. The authors would like to acknowledge the University of Maryland super-computing resources, Zaratan HPC cluster, made available for conducting the research in this paper.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1002/we.2918.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Apurva Anand https://orcid.org/0000-0003-3746-0249
Koushik Marepally https://orcid.org/0000-0001-6913-0872

REFERENCES

1. Glaws A, King RN, Vijayakumar G, Ananthan S. Invertible neural networks for airfoil design. *AIAA J*. 2022;60(5):3035-3047.
2. Burton T, Jenkins N, Sharpe D, Bossanyi E. Component design. In: Burton T, Jenkins N, Sharpe D, Bossanyi E, eds. *Wind Energy Handbook*: John Wiley & Sons, Ltd; 2011:383-473.

3. Anand A, MV SS, V C, VVS N, Narayanarao B. Aerodynamic interaction and analysis of nose-mounted propeller aircraft. In: AIAA Aviation 2023 Forum; 2023:4080.
4. Anand A. Aerodynamics of nose mounted propeller aircraft. *Master's Thesis*: Indian Institute of Science; 2021. https://etd.iisc.ac.in/handle/2005/5213
5. Marepally K, Jung YS, Baeder J, Vijayakumar G. Uncertainty quantification of wind turbine airfoil aerodynamics with geometric uncertainty. In: Journal of Physics: Conference Series, Vol. 2265 IOP Publishing; 2022:42041.
6. Anand A, Marepally K, Safdar MM, Lee B, Baeder JD. Generalizable deep learning module for rotorcraft inverse design applications. In: AIAA SCITECH 2024 Forum; 2024:460.
7. Safdar MM, Anand A, Marepally K, Lee B, Baeder JD. A surrogate modeling framework for airfoil design and optimization. In: AIAA SCITECH 2024 Forum; 2024:458.
8. Haghi R, Crawford C. Surrogate models for the blade element momentum aerodynamic model using nonintrusive polynomial chaos expansions. *Wind Energy Sci*. 2022;7(3):1289-1304.
9. Han Z-H, Görtz S, Hain R. A variable-fidelity modeling method for aero-loads prediction. In: New Results in Numerical and Experimental Fluid Mechanics VII: Contributions to the 16th STAB/DGLR Symposium Aachen, Germany 2008 Springer; 2010:17-25.
10. Hsu M-H. Dynamic analysis of wind turbine blades using radial basis functions. *Adv Acoust Vibr*. 2011;2011:2-9.
11. Hu C, Albertani R. Wind turbine event detection by support vector machine. *Wind Energy*. 2021;24(7):672-685.
12. Shalu H, Govindarajan B, Sridharan A, Singh R. Blade shape optimization of rotors using neural networks. In: 79th Annual Forum of the Vertical Flight Society; 2023;1-8.
13. Paternostro N, Marepally K, Anand A, Lee B, Baeder J. Application of CFD-trained neural networks on the rotorcraft airfoil design process. In: VFS Annual Electric VTOL Symposium; 2023;6-8.
14. Marepally K. A meta-learning based aerodynamic analysis framework for wind turbine design applications. *Ph.D. Thesis*: University of Maryland, College Park; 2023.
15. Liu D, Tan Y, Khoram E, Yu Z. Training deep neural networks for the inverse design of nanophotonic structures. *Acs Photon*. 2018;5(4):1365-1369.
16. Kulfan BM. Universal parametric geometry representation method. *J Aircraft*. 2008;45(1):142-158.
17. Tang B. Orthogonal array-based latin hypercubes. *J Am Stat Assoc*. 1993;88:1392-1397.
18. Sitaraman J, Roget B. Solution algorithm for unstructured grids using quadrilateral subdivision and hamiltonian paths. In: 52nd Aerospace Sciences Meeting; 2014:79.
19. Jiang G-S, Shu C-W. Efficient implementation of weighted ENO schemes. *J Comput Phys*. 1996;126(1):202-228.
20. Saad Y, Schultz MH. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J Scientif Stat Comput*. 1986; 7(3):856-869.
21. Costenoble M, Baeder J, Jung YS. Automated mesh generation and solution analysis of arbitrary airfoil geometries. *J Aircraft*. 2022;59(5):1120-1136.
22. Jung YS, Vijayakumar G, Ananthan S, Baeder J. Local correlation-based transition models for high-Reynolds-number wind-turbine airfoils. *Wind Energy Sci*. 2022;7(2):603-622.
23. Spalart P, Allmaras S. A one-equation turbulence model for aerodynamic flows. In: 30th Aerospace Sciences Meeting and Exhibit; 1992:439.
24. Rung T, Bunge U, Schatz M, Thiele F. Restatement of the Spalart-Allmaras eddy-viscosity model in strain-adaptive formulation. *AIAA J*. 2003;41(7): 1396-1399.
25. Paluzo-Hidalgo E, Gonzalez-Diaz R, Gutiérrez-Naranjo MA. Two-hidden-layer feed-forward networks are universal approximators: A constructive approach. *Neural Netw*. 2020;131:29-36.
26. Spalart PR, Garbaruk AV. Correction to the Spalart–Allmaras turbulence model, providing more accurate skin friction. *AIAA J*. 2020;58(5):1903-1905.
27. Costenoble M, Govindarajan B, Jung Y, Baeder J. Automated mesh generation and solution analysis of arbitrary airfoil geometries. *J Aircraft*. 2022; 59(5):1-16.

How to cite this article: Anand A, Marepally K, Muneeb Safdar M, Baeder JD. A novel approach to inverse design of wind turbine airfoils using tandem neural networks. *Wind Energy*. 2024;1-22. doi:10.1002/we.2918

APPENDIX A: ADDITIONAL DETAILS

A.0.1 | Limitation of SA model

The SA model performs appropriately for attached boundary layer flows; however, it suffers in the regions of strong adverse pressure gradient. The reason for this discrepancy is that the model was formulated based on empiricism and dimensional analysis. The model is calibrated in zero adverse pressure gradient in accordance with the logarithmic law of the wall. The formulation of SA model allows for modification of defect layer through calibration of its coefficients for a specific flow. Two terms of relative more importance are c_{w1} and c_{w2} which affect the rate of dissipation of turbulent kinetic energy while eddies transport in the flow. Researchers have proposed a few modifications to these terms for accurate modeling of turbulent flows.

A.0.2 | Low Reynolds number correction (SA-noft2-LRe)

The low Reynolds number correction to the baseline SA models leads to more accurate prediction of skin friction.[26] In the baseline SA model, c_{w2} is a constant value, that is, 0.3 that was calibrated corresponding to momentum thickness Reynolds number of 10^4. This constant affects the strength of near-wall destruction which in turn controls the skin friction. In the SA model with Reynolds number correction, c_{w2} is no longer constant rather it is calculated using χ parameter—a ratio of proxy of eddy viscosity to laminar viscosity. This parameter is representative of turbulence Reynolds number. The c_{w2} term is calculated by the following expression.

$$c_{w2,LRe} = c_{w4} + \frac{c_{w5}}{\frac{\chi}{40}+1^2} \tag{A1}$$

where $c_{w4} = 0.21$, $c_{w5} = 1.5$ and χ is the ratio of a working variable of turbulent viscosity to laminar viscosity.

The value of c_{w2} varies between 0.21 and 1.4. As the thickness of the boundary layer increases along the chord length, the turbulence viscosity increases which in turn increases χ, decreases c_{w2}, increases wall damping function f_w, increases turbulence destruction, thus increases the skin friction at low Reynolds number.

A.0.3 | SA-noft2-salsa

The strain adaptive formulation of SA model[24] was developed primarily to extend the predictive capability of the model for nonequilibrium conditions like flows with unsteadiness, strong pressure gradients, and/or shock waves. This form invokes various modifications. The term C_{b1} is no longer a constant and is sensitized to variable strain rates. This coefficient is crucial to the model's predictive performance and is defined by Equations (A2) to (A4).

$$c_{w1} = \frac{c'_{b1}}{k^2} + \frac{1+c_{b2}}{\sigma} \tag{A2}$$

$$c'_{b1} = 0.1355\sqrt{\Gamma} \tag{A3}$$

$$\Gamma = min[1.25, max(\gamma, 0.75)] \tag{A4}$$

$$\gamma = max[\alpha_1, \alpha_2] \tag{A5}$$

$$\alpha_1 = \left[1.01\left(\frac{\tilde{\nu}}{\tilde{S}k^2d^2}\right)\right]^{0.65} \tag{A6}$$

$$\alpha_2 = max\left[0, 1 - tanh\left(\frac{\chi}{68}\right)\right]^{0.65} \tag{A7}$$

where γ, α_1 and α_2 are empirical constants.

Two-dimensional steady flow is studied for NACA0012 and DU-91-W2-300 airfoil. NACA0012 airfoil is primarily used to choose turbulence models as the NASA-certified experimental data is readily available. The DU-91-W2-300 airfoil has been chosen as it not only represents our wind turbine airfoil but also gives a rigorous comparison of turbulence models because of its higher thickness to chord ratio, that is, 30%. The results of the test cases are explained in the following paragraphs.

A. **NACA0012 Airfoil:** The lift and drag predictions from the NACA0012 airfoil at Reynolds number of 6 million and Mach number of 0.15 are shown in Figure A1. All the models show close agreement of lift and drag coefficients with experimental data in the pre-stall region. However, deviations can be observed at α 15 ° or higher. The LRe as well as salsa improves the lift coefficient predictions in comparison with the baseline SA-noft2 model in the stall range. However, LRe is observed to increase skin friction more specifically in the stall range. The combination of LRe and Salsa performs even superior to using separately thus proving it a suitable combination for generating CFD data.

B. **DU-91-W2-300 Airfoil:** The simulations for DU-91-W2-300 airfoil are performed at Reynolds number of 3 million and Mach number of 0.22. The lift and drag predictions are shown in Figure A2. All the candidate models predicted the same lift coefficient as the experimental data in the linear range. The deviations between CFD and experimental lift coefficient in the stall range of DU-91-W2-300 airfoil is more than

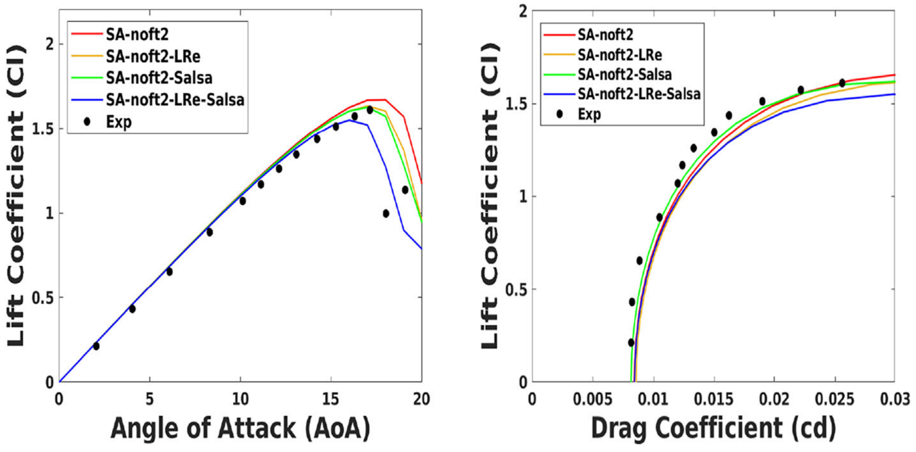

FIGURE A1 Comparison of accuracy of variants of Spalart–Allmaras (SA) turbulent model for NACA0012 airfoil; Re = 6 million, Mach = 0.15

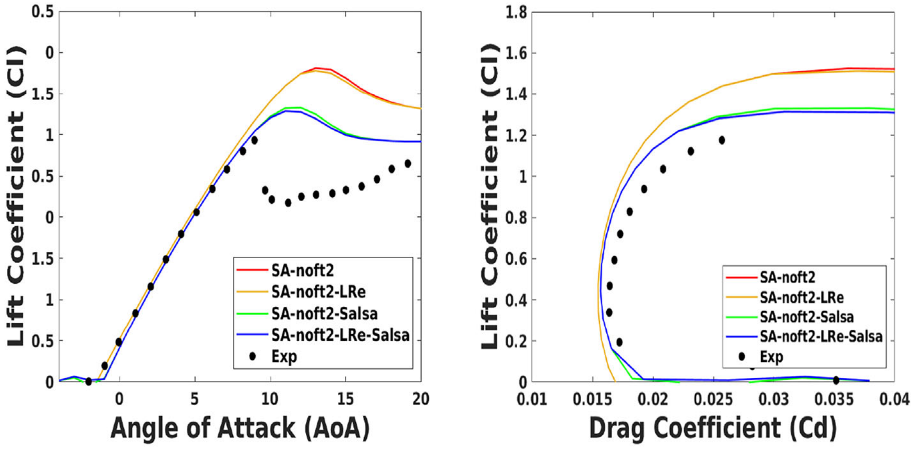

FIGURE A2 Comparison of accuracy of variants of Spalart–Allmaras (SA) turbulent model for DU-91-W2-300 airfoil; Re = 3 million, Mach = 0.22

NACA0012 airfoil. This large difference is attributed to stronger adverse pressure gradient experienced by the boundary layer as well as three-dimensional effects which are more pronounced on thicker airfoils. Such effects have not been accounted for in these turbulence models. The models show a slight difference in CFD and experimental drag coefficient which grows with an increase in angle of attack. Besides this discrepancy, the combination of LRe and salsa model still provides relatively more accurate predictions further corroborating their feasibility for generating high-fidelity data.

Based on these studies, the salsa model with LRe correction is used to generate data for this study. All stages of CFD simulations, including mesh generation (Ref.[27]), CFD flow solver setup, CFD runs, and postprocessing, 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 −4° to 20°, Reynolds numbers of 3 million and Mach number of 0.1. These flow conditions cover typical operating conditions of wind turbines.