A Machine Learning-Based Approach for Predicting Aerodynamic Coefficients Using Deep Neural Networks and CFD Data

Mara-Florina NEGOITA*[,2], Mihai-Vladut HOTHAZIE[1,2]

*Corresponding author
[1]INCAS – National Institute for Aerospace Research “Elie Carafoli”,
B-dul Iuliu Maniu 220, Bucharest 061126, Romania,
hothazie.mihai@incas.ro
[2]“POLITEHNICA” University of Bucharest, Faculty of Aerospace Engineering,
Str. Gheorghe Polizu 1-7, 011061, Bucharest, Romania,
mara.negoita@stud.aero.upb.ro*

DOI: 10.13111/2066-8201.2024.16.4.9

Received: 06 November 2024/ Accepted: 25 November 2024/ Published: December 2024

Abstract: *Artificial Intelligence (AI) and Machine Learning (ML) are increasingly being adopted across various fields, including aerodynamics, exhibiting impressive results in complex computational processes and improving prediction accuracy. This study introduces a novel method for airfoil performance assessment through the development and training of a deep Artificial Neural Network (ANN), used for predicting aerodynamic coefficients and pressure distributions, leveraging comprehensive data obtained by using a Computational Fluid Dynamics (CFD) solver. First, an automated CFD solver was developed for obtaining the extensive dataset needed for the effective training of the ANN. The automation process consisted in the generation of a geometry and a mesh, along with the successful integration of the open-source SU2 solver for conducting the aerodynamic simulations, chosen for its versatility and straightforward integration. Once various airfoil analyses were performed and a comprehensive dataset was obtained, data was normalized and the model was trained. Throughout the training process, several model configurations were tested, varying different architectures, hyperparameters and layer settings, until the best-performing layout was chosen. After broad testing and validation, the optimal configuration was identified as being the one to demonstrate the lowest error rates and the most accurate predictions on both training and unseen data, highlighting the model’s generalization capabilities. This Machine Learning-based approach, used as a substitute for traditional methods, provides remarkable accuracy and robustness, capturing complex behaviors and significantly reducing the computational costs associated with CFD simulations.*

Key Words: *artificial neural network, machine learning, computational fluid dynamics, automation, aerodynamic coefficients*

1. INTRODUCTION

Accurate airfoil performance assessments are crucial for optimizing designs and understanding aerodynamic behaviors, leading to overall improved efficiency and safety across various engineering applications. Precise results and detailed interpretations of physical processes around airfoils, operating at different conditions, offer essential understanding into how lift and drag generation and pressure distributions alike influence the aircraft's stability and overall performance.

Several techniques have been developed and used for airfoil analysis, ranging from theoretical models, such as those based on potential flow or thin airfoil theory, to numerical methods and wind tunnel testing, with the latter offering the most realistic results and highly accurate data on aerodynamic parameters. Wind tunnel testing serves not only as a flow visualization tool, but also as a validation procedure, confirming accuracy of other models used in design processes. Datasets obtained through experimental testing provide reliable, empirical results that represent benchmarks meant to validate and refine theoretical and numerical models, such being the case for [1], [2]. Regardless, dependency on costly equipment and elaborate data-acquisition systems, along with constrained testing ranges and capabilities, makes wind tunnel testing less feasible for frequent use and typical configurations.

Alternatively, numerical methods provide accurate, feasible and flexible simulation techniques, capable of capturing complex aerodynamic behavior without using extensive experimental setups. As showed in [3], J.G. Coder and M.D. Maughmer compared two low-fidelity solvers, based on panel methods and integral boundary-layers methods, XFOIL [4] and PROFIL [5], and one high-fidelity solver, OVERFLOW 2.2e [6], with wind tunnel results, providing valuable insights on the solvers' capabilities and limitations. Whereas all methods exhibit accurate results in low-drag regions with limited computational effort and reasonable processing time, they fail at capturing maximum lift, with the RANS solver outperforming low-fidelity methods at higher angles of attack.

Computational Fluid Dynamics (CFD) methods generate exceptional results, by capturing true physical behavior and phenomena even in more complex cases or at higher angles of attack, where flow separation effects become more pronounced. Still, CFD simulations are time-consuming, computationally expensive and highly reliant on mesh quality and the turbulence models used [7]. Consequently, attempts to reduce computational time and effort in high-fidelity simulations without lowering accuracy of the obtained results have led researchers to explore alternative solutions, leading to the development of Machine Learning (ML) based approaches.

Several studies have shown promising results in predicting aerodynamic coefficients using Artificial Neural Networks (ANN) [8], [9] and Convolutional Neural Networks (CNN) [10], [11]. As efficient training and accurate predictions of ANNs are highly dependent on the size, consistency and quality of the training data provided to the model, the use of an automation process for extracting CFD results has been developed in this study. The airfoil geometry is generated by using the NACA 4 digits parametrization technique, not only for ensuring aerodynamic feasibility, but also for reducing the number of input parameters of the neural network. The given geometry is then automatically replaced into the mesh generator and the resulted mesh is exported. The simulation is being performed using the opensource CFD solver SU2 (Standford University Unstructured) [12], chosen due to its versatility, effortless integration with pre and post-processing tools and parallel computing capabilities.

The extensive database obtained through the automation represents the input fed to the ANN for the training process. The generated predictions of the trained model present highly accurate results in substantially reduced time, with lower computational effort, further proving the efficiency of ML based models. Proper training of neural networks offers extensive use for real-time applications, such as flight control or on-board systems [13] , as well as optimization methods, allowing for more feasible airfoil geometries [14].

2. METHODOLOGY

The comprised methodology in this paper focuses on the development of the CFD automation needed for obtaining an extensive database of airfoil coefficients, consisting of the geometry and mesh generation, coupled with an opensource high-fidelity CFD solver used for extracting the lift and drag coefficients, as well as the pressure and skin friction coefficient distribution. The resulting data serves as the training dataset for a deep neural network (DNN), capable of predicting the airfoil characteristics, based on a series of input parameters. Both the DNN architecture and the influence of hyperparameters and training settings are presented in this section.

2.1 CFD Automation

The implemented CFD automation links the chosen parametrization method for creating the airfoil surface, with the generation of an unstructured mesh, integrating SU2 as the high-fidelity solver. The workflow is integrated through MATLAB, creating the airfoil shape, based on the three input parameters, M, P and T, that is automatically passed to Ansys Design Modeler for creating the geometric domain. The resulting geometry is then imported in Ansys Mechanical and an unstructured, high-quality mesh is generated. The open-source CFD solver SU2 is responsible for solving the governing fluid dynamics equations and predicting flow characteristics such as lift, drag, and pressure distribution.

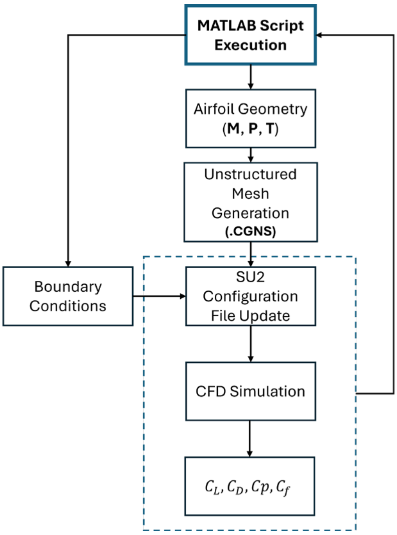

MATLAB Script Execution
Airfoil Geometry (M, P, T)
Unstructured Mesh Generation (.CGNS)
Boundary Conditions
SU2 Configuration File Update
CFD Simulation
C_L, C_D, Cp, C_f

Figure 1. Logical scheme of the automation process

After the development of the automation loop, the aerodynamic coefficients database was obtained by analyzing several NACA 4-digits airfoils, with a maximum camber (M) and the position of the maximum camber (P) varying from 1 to 6, and the maximum thickness (T) from 8 to 16. Each airfoil was evaluated at 12 angles of attack, from -4 to 7 degrees. For optimizing computational time and effort, the simulations were performed using the parallel computing capabilities of SU2, running on 16 cores.

2.1.1 Airfoil Parametrization Technique

Various airfoil parametrization methods are currently in use, ranging from NACA 4, 5 or 6 digits, to Bézier Curves [15] and Class-Shape Transformation (CST) [16]. Since the main objective was the efficient training of a neural network, the number of input parameters is crucial for accurate results. Therefore, the NACA 4-digit parametrization technique, defined by only three parameters: the maximum camber in tenths of chord, (M), the position of the maximum camber in tenths of chord (P) and the maximum thickness as a fraction of the chord (T), was considered the most suitable.

Based on parameters M and P, the camber line of the airfoil is defined as:

$$y_c = \begin{cases} \frac{M}{P^2}(2Px - x^2), & 0 \le x < P \\ \frac{M}{(1-P)^2}(1 - 2P + 2Px - x^2), & P \le x \le 1 \end{cases} \tag{1}$$

Then, based on the maximum thickness (T), the half thickness is written as:

$$y_T = \frac{T}{0.2}(a_0 x^{0.5} + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4) \tag{2}$$

where the coefficients a_0, a_1, a_2, a_3, a_4 have the values of:

$$a_0 = 0.2969, \quad a_1 = -0.126, \quad a_2 = -0.3516, \quad a_3 = 0.2843, \quad a_4 = -0.1036 \tag{3}$$

Thus, the coordinates for the upper and lower surface of the airfoil are obtained using the following relations:

$$\begin{array}{lll} Upper\ Surface & x_u = x_c - y_t sin\theta & y_u = y_c + y_t cos\theta \\ Lower\ Surface & x_l = x_c + y_t sin\theta & y_l = y_c - y_t cos\theta \end{array} \tag{4}$$

where:

$$\theta = \operatorname{atan}\left(\frac{dy_c}{dx}\right) \tag{5}$$

2.1.2 Automated Mesh Generation

Once the airfoil is created and imported into the geometry using Ansys Design Modeler, the computational domain and mesh are created using Ansys Meshing. The two-dimensional discretized volume consists of an unstructured, circular mesh, with domain boundaries placed 250 chord lengths from the airfoil surface, for reducing the effect of the domain boundary on the obtained solution.

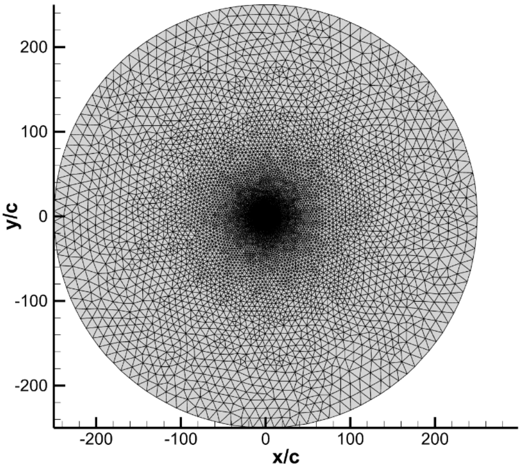


Figure 2. Computational domain around the airfoil

Triangular elements with the maximum size of 15 meters have been used for the mesh near the outer domain, whereas quadrilaterals elements were chosen for the region adjacent to the airfoil surface. Mesh spacing at the solid surface was 1×10^{-6}, along with a growth rate of 1.2 and a total number of 40 quadrilateral element layers ensured a $y^{+} < 1$ over the airfoil surface. This configuration proved itself to be both well-performing and computationally inexpensive, allowing the generation of a high-quality dataset in a reasonable calculation time.

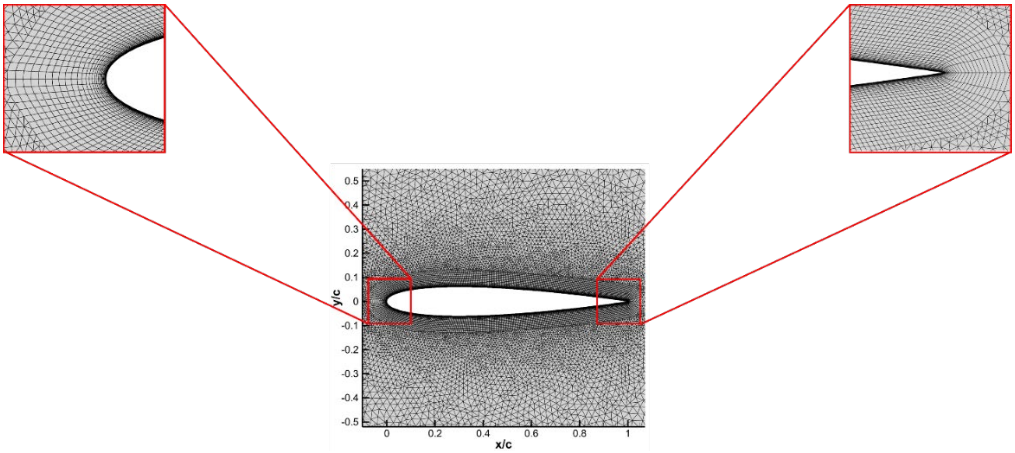


Figure 3. Mesh around the airfoil

2.1.3 Solver Settings and Simulation

In order to ensure proper results and overall accurate aerodynamic predictions, the chosen flow conditions represent the classic NACA 0012 validation case [17]. This case is widely used for benchmarking CFD solvers due to its availability of high-quality experimental data. Before proceeding with the automated simulation process, an extensive validation study was conducted, comparing the results obtained with SU2 with the outputs from CFL3D. This comprehensive validation confirmed the accuracy of SU2 in predicting key aerodynamic parameters, providing confidence in the automated simulation workflow.

Table 1. NACA 0012 free-stream conditions

$\boldsymbol{M_{inf}}$	0.15
$\boldsymbol{Re_{inf}}$	6×10^{6}
$\boldsymbol{T_{inf}}$	300 K
α	10°, 15°

In accordance with the validation procedures in [18], the simulation was preformed using the Spalart-Allmaras turbulence model, for capturing turbulent flows in aerodynamic cases, paired with a first-order scalar upwind method. Additionally, the default Verkatakrishnan slope limiter [19] in SU2 was employed to effectively dampen any non-physical oscillations that typically arise in regions with sharp gradients, thus improving both numerical stability and overall solution accuracy. This combination of turbulence modeling, numerical methods, and slope limiting is essential for maintaining the reliability of the simulation, particularly in cases with high flow gradients, ensuring accurate and consistent aerodynamic predictions.

2.2 Artificial Neural Network

Neural networks (NN) are a class of machine learning models designed to approximate complex, non-linear relationships between inputs and outputs. Structurally, they consist of multiple layers of interconnected nodes called neurons, each representing a mathematical function. The neurons are organized in layers: an input layer, one or more hidden layers, and an output layer. An artificial neural network consisting of multiple hidden layers is referred to as a Deep Neural Network (DNN). In order to achieve accurate predictions of aerodynamic coefficients, based on the previously generated training database, several deep neural networks have been trained using the Deep Learning Toolbox in MATLAB, varying architectures and hyperparameters. Each of them was checked according to the efficiency and time of the training process and the accuracy of the output results, determining one optimal configuration for every aerodynamic coefficient.

For each individual combination of geometric parameters (M, P, T) and angle of attacks in the database, the preferred DNNs are capable of predicting aerodynamic coefficients as well as the distribution of the pressure coefficient (C_P) and the skin friction coefficient $\left(C_f\right)$. Regarding the first two coefficients, the optimal neural network configuration consists of four input parameters and two hidden layers of neurons, for outputting the given coefficients. Due to the extensive database and training process, for the two distributions of pressure and skin friction coefficients, multiple architectures were tested. In order to reduce training time and optimize predictions, the training has been divided into two distinctive datasets, representing the distributions of upper and lower surface. Moreover, the data has been sorted by the value of the maximum camber (M), resulting in 6 different trained models, each having 4 input parameters: the two remaining geometry defining ones, P and T, the angle of attack AoA and the position x from the leading edge along the chord line.

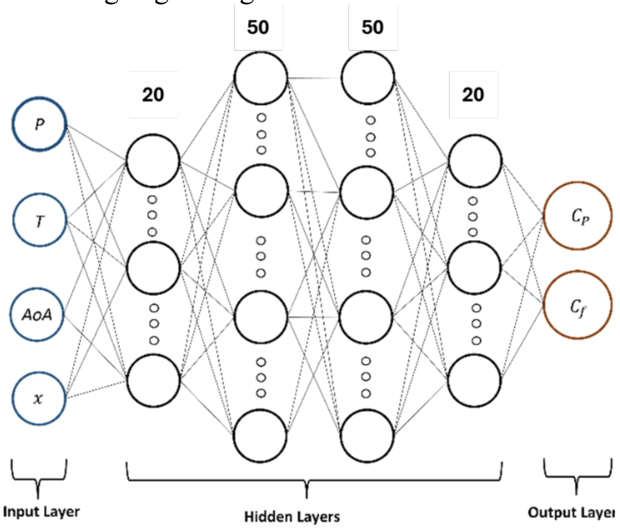


Figure 4. Deep Neural Network architecture for pressure and skin friction coefficients distributions

Once the network's architecture has been established, the data was normalized using Z-score normalization, improving the performance and accuracy of the machine learning algorithm by ensuring that all features contribute equally to the model. Standardization improves gradient descent convergence, prevents feature domination, eliminating any biased model learning caused by the larger magnitudes in the dataset. Z-score normalization is obtained by adjusting data (X) to a normal distribution, with a mean (μ) of 0 and a standard deviation (σ) of 1.

$$X' = \frac{X - \mu}{\sigma} \tag{6}$$

After the normalization, the data is split into a training and a validation dataset, to ensure the generalization capabilities of the model and prevent overfitting. The training process is done using a technique known as backpropagation in conjunction with gradient descent. Each neuron in a layer processes information by multiplying the vector of inputs (X) with a vector of weights (W) and passing the result through an activation function (f), introducing non-linearity, optimally and iteratively adjusting the weights and biases (B) to improve given predictions.

$$y = f\left(\sum_{i=1}^{n} w_i \cdot x_i + b_i\right) = f(W^T \cdot X + B) \tag{7}$$

For evaluating the model's efficiency, the training process has been closely supervised, analyzing the loss, gradient value and overall performance of the network. The mean squared error (MSE) was calculated to measure the difference between the predicted $(\widehat{y_i})$ and the actual value (y_i), with respect to the number of points (n):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \tag{8}$$

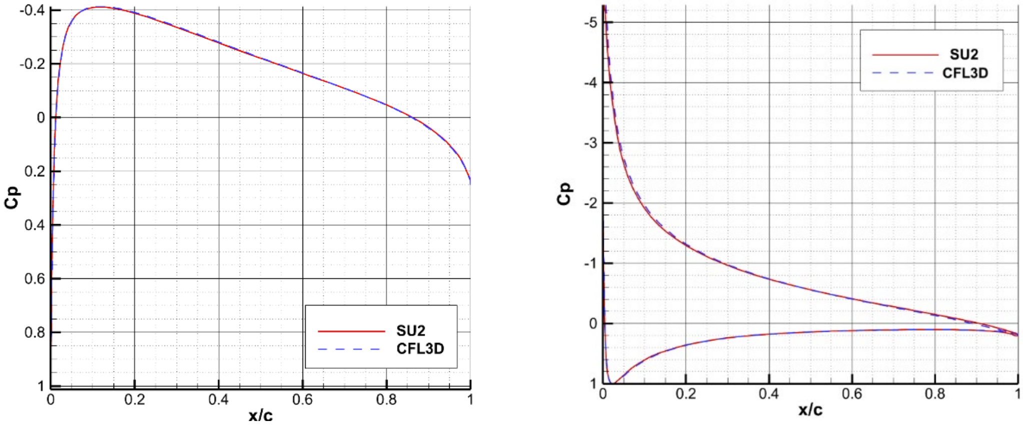
Furthermore, another metric used is R^2, representing the proportion of the variance in the independent variable from the mean of the actual values $(\bar{y})$

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2} \tag{9}$$

3. RESULTS

3.1 CFD Validation

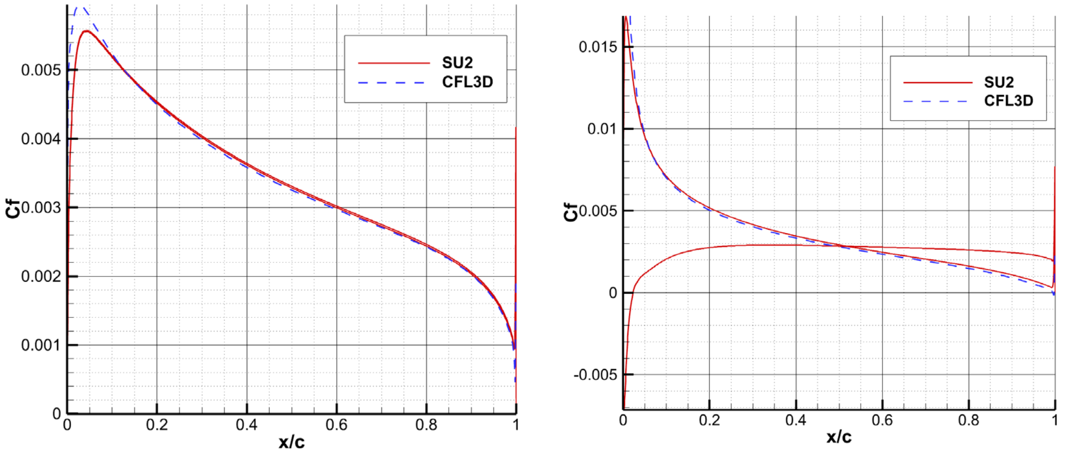
The computed results for the pressure coefficient distributions for NACA0012 are presented in *Figure 4.* The solutions obtained with SU2 are in good agreement with those obtained with CFL3D, for both angles of attack investigated. The stagnation point, as well as the recovery region are correctly captured in both cases, proving the accuracy of the run simulation.



(a) (b)

Figure 5. Pressure Coefficient Distribution versus Normalized Chord Length at α = 0° (a) and α = 10° (b)

The skin friction coefficient distribution obtained on the upper surface through the simulation with SU2 matches the aspect of the CFL3D solution, as was the case with the coefficient of pressure. Although a certain inconsistency appears around the leading edge for the zero angle of attack, the distribution recovers its aspect along the chord and the mismatch disappears, leading to overall good accuracy of the solution. Furthermore, no such nonconformance is present at 10° angle of attack.



(a) (b)

Figure 6. Skin Friction Coefficient Distribution versus Normalized Chord Length at α = 0° (a) and α = 10° (b)

3.2 Neural Network Results

Before investigating the displayed predictions of the DNN, it is crucial to carefully monitor the training process to ensure the model's performance and reliability. By observing training and validation loss, accuracy, and learning curves, issues can be detected and corrected early, preventing misleading results, ensuring the model generalizes well and performs optimally on later applications.

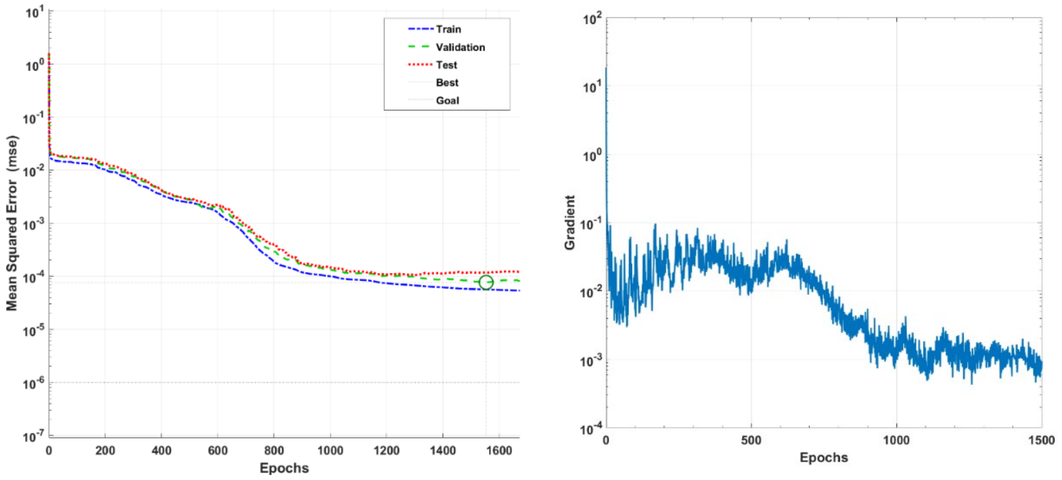
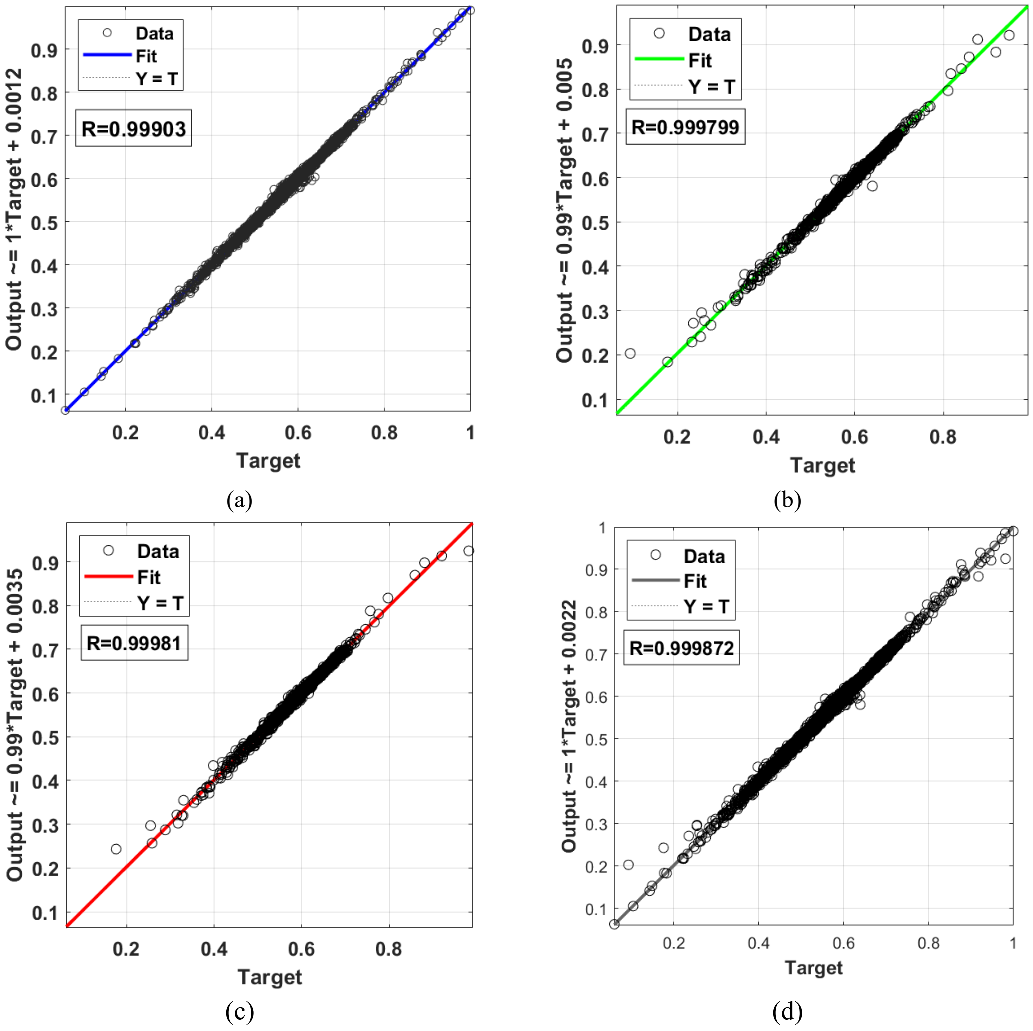


Figure 7. MSE versus Number of Epochs (a) and Gradient versus Number of Epochs (b)

The MSE versus Number of Epochs graph proves a successful training process, with error rates decreasing as the weights get optimally adjusted and the model learns data behavior.

During initial training, the error rate decreases rapidly, suggesting a quick improvement in prediction capabilities. After around 1000 epochs, the MSE continues declining at a slower rate, later hitting a plateau where further training yields less significant improvements. At approximately 1600 epochs, the model reaches the best point, marked by the circle, where the MSE for the validation and test sets is minimized. This is a crucial indicator of the optimal model performance in terms of generalization, as it represents the point in training where the validation error starts increasing. The model is stopped early, in order to prevent the overfitting phenomenon, where the neural network only replicates previously learned training data, without correctly capturing behaviors of unseen data.

The effective learning of the model is furthermore indicated by the gradual gradient decrease, dictating the direction and magnitude of updates made to the model's weights. Gradients have a descending trend along the training process, until around 1500 epochs, where values set to a constant mean that indicates the convergence of the neural network. The absence of sudden spikes or large fluctuations in the gradient towards the end of the training further suggests that the model is not encountering major instability, being the indicative of a smooth learning process that results in a well-trained network.



(a)

(b)

(c)

(d)

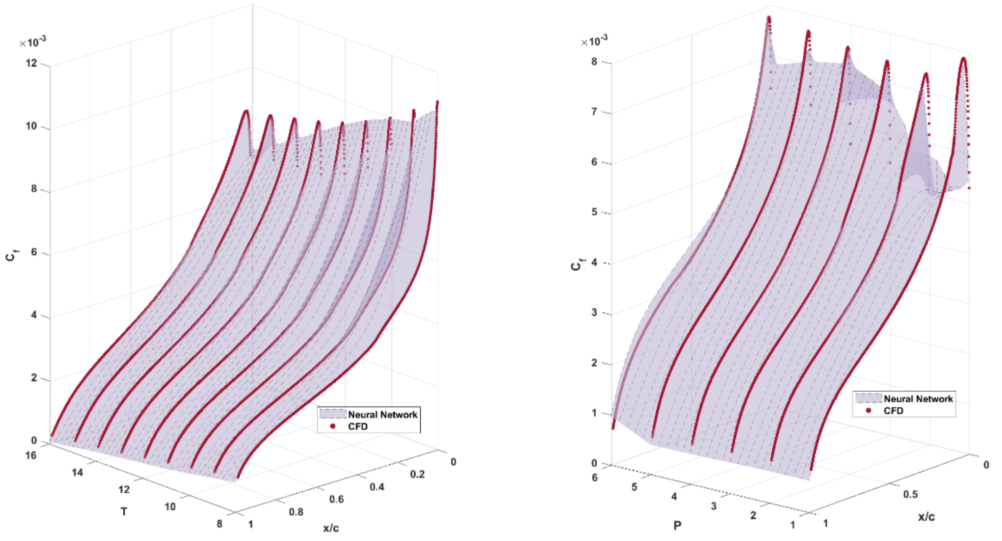
Figure 8. R Values for Training (a), Validation (b), Test (c) and overall fitting (d)

Presented regression plots assess the model's performance by comparing the predicted values (outputs) with the true values (targets). The correlation coefficient (R) measures how well the outputs correlate with the targets, values closer to 1 indicating better fitting of predicted data. A value of R of 0.999799 on the validation dataset indicates remarkable generalization, whereas high values, of 0.99903 and 0.99981 on training and testing data suggests effective training of the DNN, represented by the close fitting to the ideal values, displayed by the dotted line (Y=T). The chosen model architecture has shown exceptional correlation values, as well as low error rates on all subsets of data after 2000 epochs, where the model has been considered converged and training was stopped for preventing overfitting.

Table 2. Training parameters on the upper-surface datasets after 2000 epochs

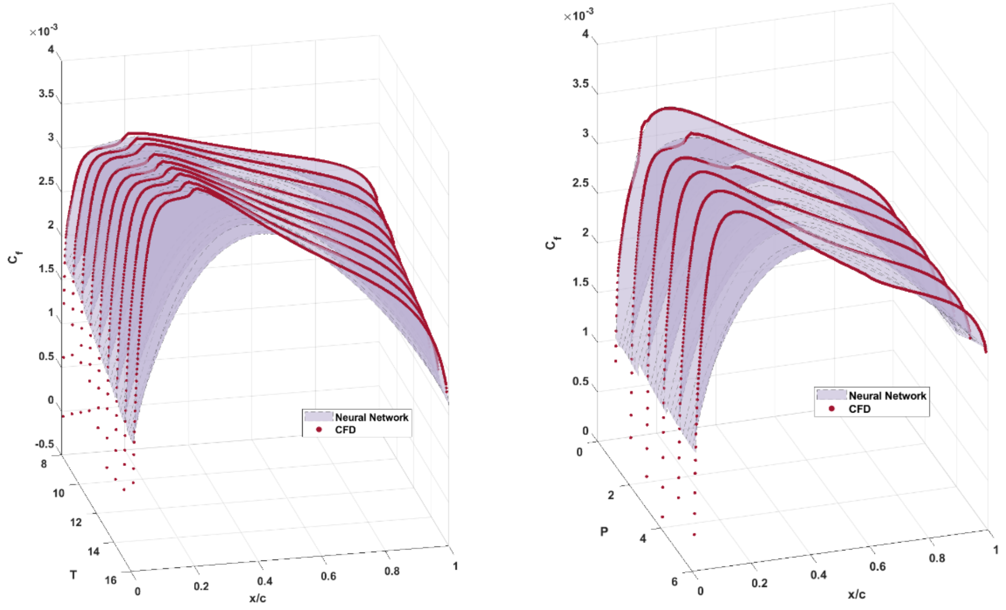
Maximum Camber	Aerodynamic (Coefficient)	R-Value (Training)	R-Value (Testing)	R-Value (Validation)	R-Value (Entire Dataset)	MSE
$M = 1$	C_P	0.99915	0.99725	0.99901	0.99847	0.0041
	C_f	0.999481	0.99921	0.99942	0.99937	0.00374
$M = 2$	C_P	0.99981	0.99903	0.999799	0.999872	0.0032
	C_f	0.999499	0.99941	0.999451	0.99945	0.0039
$M = 3$	C_P	0.99923	0.99914	0.99914	0.99917	0.00412
	C_f	0.99936	0.99931	0.999355	0.99934	0.00402
$M = 4$	C_P	0.99929	0.99917	0.99914	0.9992	0.00406
	C_f	0.99971	0.99964	0.99968	0.99968	0.00343
$M = 5$	C_P	0.99941	0.999451	0.99947	0.99944	0.00385
	C_f	0.9995	0.99953	0.99951	0.99952	0.00287
$M = 6$	C_P	0.99936	0.99936	0.999345	0.99935	0.00341
	C_f	0.999405	0.99939	0.99941	0.9994	0.0029

Furthermore, a comparative parametric analysis has been performed for investigating the performance of the trained DNN with respect to the generated results. Geometric parameters P and T have been varied for a fixed M and the resulting distributions of pressure and skin friction coefficient have been plotted for analyzing the difference between the CFD and the predicted data.



(a) C_f vs. T vs. x/c for M=2, P=4 and α = 4°

(b) C_f vs. P vs. x/c for M=2, T=12 and α = 4°

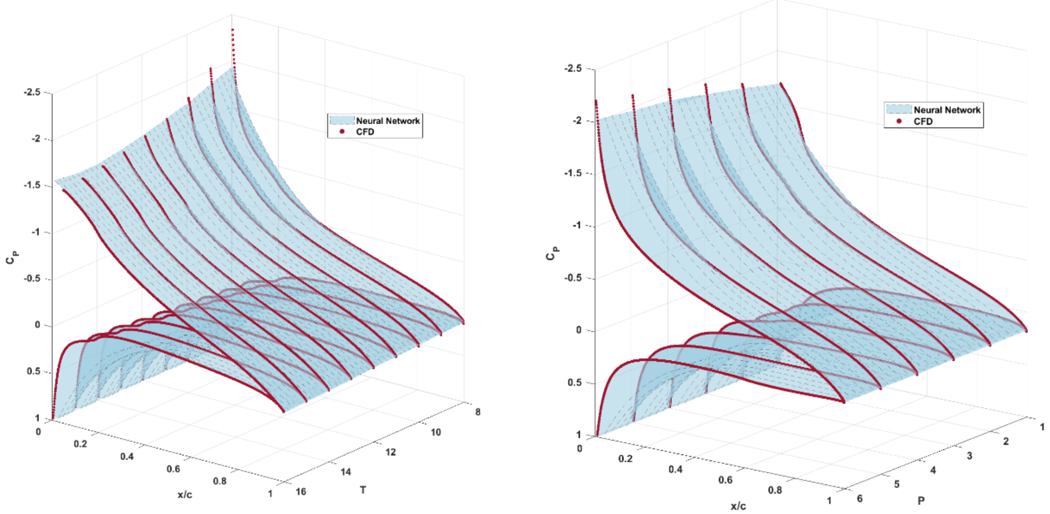


(c) C_f vs. T vs. x/c for M=2, P=4 and $\alpha = 4°$

(d) C_f vs. P vs. x/c for M=2, T=12 and $\alpha = 4°$

Figure 9. Comparative parametric analysis between CFD and AI predicted skin friction coefficient for the upper (a-b) and lower surface (c-d)

The C_f distributions reflect the high accuracy of the trained neural network, predictions being in good agreement with CFD data for both upper and lower surfaces. Maximum values near leading edge, as well as the aspect along the chord and near the trailing edge are correctly captured for all displayed distributions. Certain inconsistencies appear on the lower surface, for values close to the leading edge, as the model fails to capture extreme points, despite data normalization. Still, such points do not yield any relevant physical relevance, the neural network correctly predicting relevant aerodynamic behavior.



(a) C_P vs. T vs. x/c for M=2, P=4 and $\alpha = 4°$

(b) C_P vs. P vs. x/c for M=2, T=12 and $\alpha = 4°$

Figure 10. Comparative parametric analysis between CFD and AI predicted pressure coefficient

In accordance with the skin friction coefficient, the pressure coefficient distribution predicted by the DNN reflects the true physical data behavior, matching CFD data for both upper and lower surface.

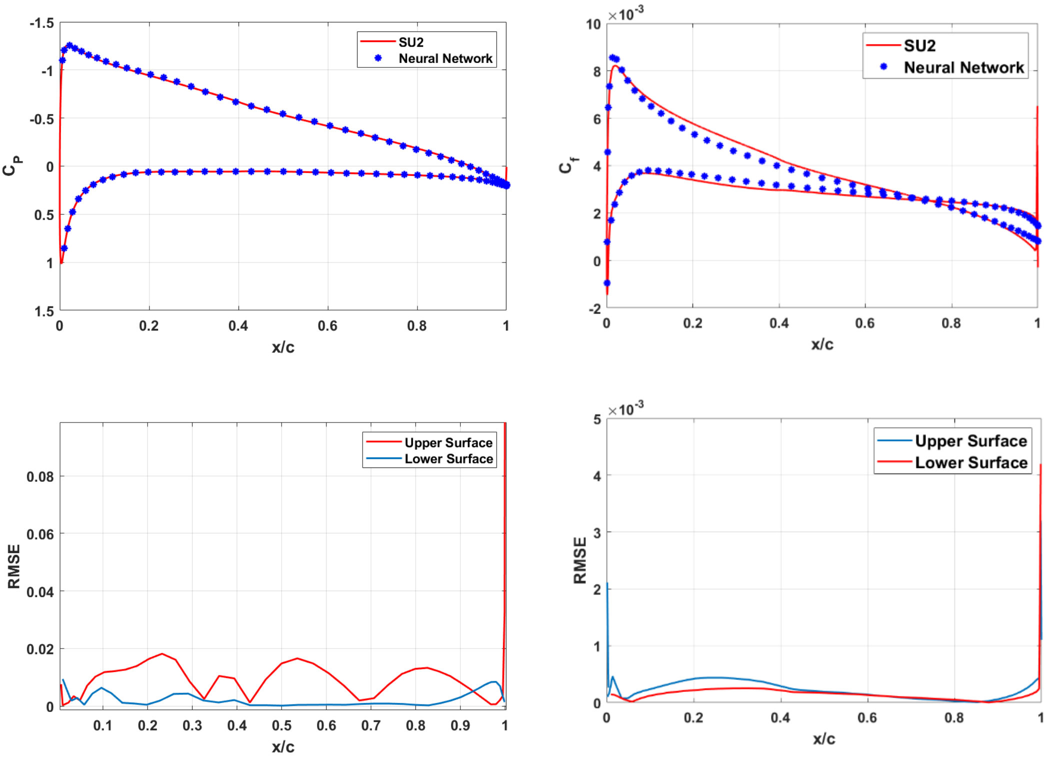
The main discrepancies between computed and predicted data lays in the representation of the stagnation point on certain airfoils, especially for lower thickness values, where minimal C_P values are reached, inconsistencies are due to the model's incapacity of capturing extreme values in the training set.

For better visualization, the NACA2412 at an angle of attack of 4 degrees was chosen for comparison. For both C_P and C_f, the model has successfully replicated CFD data with minimal error rates.

The DNN accurately captures both extreme values and distributions along the chord and close to the leading and trailing edge.

Table 3. Mean RMSE values for NACA2412, at α = 4°

Aerodynamic Coefficient	Surface	RMSE
C_P	Upper Surface	0.00185
	Lower Surface	0.00312
C_f	Upper Surface	0.00032
	Lower Surface	0.000305



(c) C_P vs. x/c

(d) C_f vs. x/c

Figure 11. Comparative analysis between CFD and DNN predicted data for NACA2412 at α = 4°

4. CONCLUSIONS

The main purpose of this study was the development of a Machine Learning approach for predicting aerodynamics coefficients. Firstly, for generating the extensive dataset needed for the training process, an automation process was created, coupling the geometry replacement with the mesh generation and the CFD simulation run with the open-source solver SU2. Then, after a solution validation was performed, confirming the fidelity of the obtained results, a database was constructed by investigating a series of NACA 4-digit airfoils at various angles of attack.

The resulting dataset was normalized and divided before being fed as input to the neural network. After the training process has proved its effectiveness, the predicted results were closely investigated, demonstrating the algorithm's high accuracy and low error rates. This approach was able to accurately replicate the original dataset in a significantly reduced computational time, based on only four input parameters, without the need for mesh generation, solving of mathematical equations or turbulence models.

Still, the approach's success is highly dependent on the size and quality of the training set, any inconsistencies leading to alternating gradient values during training, poor fitting and inaccuracies in the results. Moreover, no optimal neural network configuration exists, the iterative adjustment of hyper-parameters being the only method in identifying the best performant architecture.

To conclude with, this approach has shown promising results, its success laying the foundation for further analysis and development, exploring with different parametrization techniques and datasets, and allowing for the training of Convolutional Neural Networks, capable of predicting pressure or Mach contours trained on visual representations of aerodynamic coefficients.

ACKNOWLEDGEMENTS

This article is an extension of the paper presented at *The 11th International Conference of Aerospace Sciences, "**AEROSPATIAL 2024**"*, 17 – 18 October 2024, Bucharest, Romania, Section 1. Aerodynamics – Oral Presentation.

REFERENCES

[1] I. Abbot and A. von Doenhoff, *Theory of Wing Sections*, 1949.

[2] C. L. Ladson, *Effects of Independent Variation of Mach and Reynolds Numbers on the Low-Speed Aerodynamic Characteristics of the NACA 0012 Airfoil Section*, 1988.

[3] J. D. Coder and M. D. Maughmer, Comparisons of Theoretical Methods for Predicting Airfoil Aerodynamic Characteristics, *Journal of Aircraft,* vol. **51**, no. 1, 2014.

[4] M. Drela, XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils, in *Lecture Notes in Engineering*, 1989.

[5] R. Eppler, *Airfoil Program System, 'PROFIL07'*, User's Guide, Stuttgart, Germany, 2007.

[6] P. B. R. Nichols, *User's Manual for OVERFLOW 2.1*, NASA Langley Research Center, Hampton, VA, 2008.

[7] D. Dimitris, Computational aerodynamics: Advances and challenges, *The Aeronautical Journal,* vol. **120**, no. 1223, pp. 13-36, 01/2016.

[8] K. H. Z. I. Moin Hassan, Airfoil's Aerodynamic Coefficients Prediction using Artificial Neural Network, in *19th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, 2022.

[9] D. R. K. D. H. T. M. Y. U. K. A. Teimourian, Airfoil aerodynamic performance prediction using machine learning and surrogate modeling, *Heliyon,* vol. **10**, no. 8, 2024.

[10] R. Garcia-Fernandez, K. Portal-Porras and A. Zugatz, CNN-based flow field prediction for bus aerodynamics analysis, *Scientific Reports,* vol. **13**, no. 1, 2023.

[11] H. Chen and S. Wang, Multiple Aerodynamic Coefficient Prediction of Airfoils Using a Convolutional Neural Network, *Symmetry,* vol. **12**, no. 4, 2020.

[12] T. Economon, F. Palacios, T. Lukaczyk and J. Alonso, SU2: An Open-Source Suite for Multiphysics Simulation and Design, *AIAA,* vol. **54**, no. 3, 2015.

[13] D. Zhu, J. Peng and C. Ding, A Neural Network with Physical Mechanism for Predicting Airport Aviation Noise, *Aerospace,* vol. **11**, no. 9, 2024.

[14] M. Negoita, D. Crunteanu, M. V. Hothazie and M. V. Pricop, Enhancing Airfoil Performance through Artificial Neural Networks and Genetic Algorithm Optimization, *INCAS Bulletin,* (print) ISSN 2066–8201, (online) ISSN 2247–4528, ISSN–L 2066–8201 vol. **15**, no. 4, https://doi.org/10.13111/2066-8201.2023.15.4.17, 2023.

[15] T. Melin, *Parametric Airfoil Catalog, An Aerodynamic and Geometric Comparison Between Parametrized and Point Cloud Airfoils*, Linkoping, Sweden: Fluid and Mechatronic Systems, 2013.

[16] B. M. Kulfan, Universal parametric geometry representation method, *Journal of Aircraft,* vol. **45**, no. 1, pp. 142-158, 2008.

[17] * * * turbomodels.larc.nasa.gov, [Online], Available: https://turbmodels.larc.nasa.gov/naca0012_val.html.

[18] F. Palacios, T. D. Economon, A. C. Aranake and S. Copeland, Stanford University Unstructured (SU2): Open-source Analysis and Design Technology for Turbulent Flows,"in *52nd Aerospace Sciences Meeting*, National Harbor, Maryland, 2014.

[19] V. Venkatakrishan, "onvergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters, *Journal of Computational Physics,* vol. **118**, no. 1, 1995.