Heliyon 10 (2024) e29377

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Airfoil aerodynamic performance prediction using machine learning and surrogate modeling

Amir Teimourian [a], Daniel Rohacs [b], Kamil Dimililer [c], Hanifa Teimourian [c], Melih Yildiz [d,e], Utku Kale [b,*]

[a] Department of Aeronautical Engineering, University of Kyrenia, Northern Cyprus, Via Mersin 10, Turkey
[b] Department of Aeronautics, and Naval Architecture, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, Budapest, Hungary
[c] Department of Electrical & Electronic Engineering, Faculty of Engineering, Near East University, Northern Cyprus, Via Mersin 10, Turkey
[d] Department of Aeronautical Engineering, Faculty of Aviation Sciences, Erciyes University, Kayseri, Turkey
[e] Aviation Research and Application Center, Erciyes University, Kayseri, Turkey

A R T I C L E I N F O

Keywords:
Machine learning
Prediction
Train/test ratio
Aerodynamic design
Lift-to-drag ratio
Airfoil

A B S T R A C T

In recent times, machine learning algorithms have gained significant traction in addressing aerodynamic challenges. These algorithms prove invaluable for predicting the aerodynamic performance, specifically the Lift-to-Drag ratio of airfoil datasets, when the dataset is sufficiently large and diverse. In this paper, we delve into an exploration of five machine learning algorithms: Random Forest, Gradient Boosting Regression, Decision Tree Regressor, AdaBoost Algorithm, and Linear Regression. These algorithms are scrutinized within the context of various train/test ratios to predict a crucial aerodynamic performance metric—the lift-to-drag ratio—for different angle of attack values. Our evaluation encompasses an array of metrics including R^2, Mean Square Error, Training time, and Evaluation time. Upon analysis, the Random Forest Method, with a train/test ratio of 0.2, emerges as the frontrunner, showcasing superior predictive performance when compared to its counterparts. Conversely, the Linear Regression algorithm distinguishes itself by excelling in training and evaluation times among the algorithms under scrutiny.

1. Introduction

Machine learning (ML) has found widespread application across various scientific disciplines. Recent advancements in computer science and the emergence of efficient ML algorithms have aided scientists in analysing vast datasets to address myriad engineering challenges [1]. For example, ML has made significant inroads into the wind energy sector, where it plays a pivotal role in analysing data collected during energy production to enhance efficiency. Aksoy and Selbas [2] developed an ML model based on the MLR algorithm to estimate wind turbine energy production, achieving a remarkable 90 % accuracy in their predictions.

Another noteworthy application of ML is in the field of unmanned aerial vehicles (UAVs) and communication. ML serves as a valuable tool for optimizing UAV-based communication systems [3] and devising UAV trajectory plans that consider factors like power consumption and wireless communication optimization [4].

A distinctive and promising domain for ML application is aerodynamics, particularly in predicting aerodynamic performance.

* Corresponding author.
E-mail address: kale.utku@kjk.bme.hu (U. Kale).

https://doi.org/10.1016/j.heliyon.2024.e29377
Received 21 October 2022; Received in revised form 29 February 2024; Accepted 7 April 2024
Available online 9 April 2024

Abbreviations

ANN	Artificial Neural Network
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
DT	Decision Tree
DTR	Decision Tree Regressor
GBR	Gradient Boosting Regressor
LDR	Lift-To-Drag Ratio
LR	Linear Regression
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Square Error
PDE	Partial Differential Equations
RANS	Reynolds-Averaged Navier Stokes
RF	Random Forest
UAV	Unmanned Air Vehicle

Letters

α	Angle of Attack

Extensive research has been conducted on employing ML, big data analytics, and deep learning techniques in airfoil design, where the aerodynamic properties of airfoil play a pivotal role. Traditional methods, such as theoretical calculations involving computational fluid dynamics (CFD) and experimental measurements, have historically been employed to obtain airfoil aerodynamic coefficients. However, these approaches are encumbered by drawbacks like high costs and time-intensive processes, especially when dealing with extensive sets of precise aerodynamic data. Consequently, there has been a growing focus on investigating the aerodynamic efficiency of airfoils, given their critical significance in designing aircraft, wind turbines, and related applications. While wings exhibit a three-dimensional flow phenomenon, airfoils are indispensable in examining fundamental aerodynamic properties. Despite an airfoil's basic representation as a wing cross-section, modified airfoil properties can provide insights into finite wing aerodynamic performance, even extending to wing or wind turbine blade assessments. Notably, one of the paramount airfoil properties is the lift-to-drag ratio (LDR), a pivotal metric for evaluating aerodynamic performance and guiding airfoil optimization efforts.

2. Literature review

Artificial neural network (ANN) surrogate modelling possesses the advantages of computational efficiency and accurate generalization, making it a valuable tool for addressing aerodynamic design challenges. This approach leverages data-driven and bottom-up principles to achieve outcomes that are often challenging to attain using conventional methods. ANN has found application in aerodynamic design, particularly in domains such as aircraft and turbomachinery design, where optimal shapes are sought. Additionally, ANN has demonstrated its potential in surrogate modelling for optimization purposes [5]. For instance, Nowruzi et al. [6] explored the use of artificial neural networks to predict the hydrodynamic performance of hydrofoils. Their approach involved CFD simulations to establish a robust database, and they employed feed forward ANNs with an iterative algorithm, employing transfer functions for backpropagation. Portal-Porras et al. [7] discussed the application of Convolutional Neural Networks (CNNs) in predicting turbulent flow magnitudes based on a Reynolds-Averaged Navier-Stokes (RANS)-based turbulence model. Their study revealed that advanced network structures outperformed basic ones, particularly in scenarios involving uncertain flows. Furthermore, Chen et al. [8] investigated a graphical prediction method for multiple aerodynamic coefficients of airfoils, employing CNNs to process composite airfoil images and demonstrating high accuracy in predicting coefficients such as pitch-moment, drag, and lift.

Sekar et al. [9] delved into the inverse design of airfoils using a deep convolutional neural network, where a novel pressure-coefficient distribution guided the generation of airfoil shapes. However, these methods often relied on time-consuming CFD modelling to evaluate aerodynamic coefficients. While some intelligent methods aimed to reduce computational time, they frequently lacked enhancements and introduced graphical representations that could distort aerodynamic curves and data. Moreover, efforts to learn the relationship between airfoil shape and aerodynamic coefficients sometimes disregarded the influence of initial inflow conditions [10]. Portal-Porras et al. [11] concentrated on enhancing wind turbine performance by implementing flow control devices on airfoils. They conducted numerous CFD simulations, employing Convolutional Neural Networks (CNNs) to predict velocity, pressure fields, and aerodynamic coefficients. Deep learning presents an alternative approach, eliminating the need for computationally expensive physical equations while approximating input-output mappings accurately [12]. Their data-driven approach, employing CNNs to predict pressure distributions over airfoils, achieved remarkable accuracy with a Mean Squared Error (MSE) of less than 2 %. Furthermore, as numerical aerodynamic models for wind turbine blades entail substantial computational costs, Lalonde et al. [13] explored six neural networks as surrogate models. Among the models examined, CNN demonstrated the highest accuracy, while Full multilayer perceptron (MLP) excelled in combining accuracy with a simpler network structure.

In the context of this research, five ML algorithms—Random Forest, Gradient Boosting Regression, Decision Tree Regressor, AdaBoost Algorithm, and Linear Regression—were implemented to predict the Lift-to-Drag Ratio of airfoils. Various ML algorithms were employed to develop the suggested framework. The evaluation of these ML networks involved assessing MSE, the coefficient of determination (R^2), and testing time across different train-test ratios.

3. Methods and materials

In this study, the TensorFlow Python library for machine learning has played a pivotal role in constructing models for various ML algorithms. The foundation of these ML models heavily relies on the quality and diversity of the dataset used for training and evaluation. The dataset employed in this research is a valuable resource, acquired from a prior study, and is openly accessible, exemplifying the collaborative and open nature of scientific research [14]. This dataset has been an asset in the scientific community, serving as a fundamental building block for numerous investigations in the field.

The importance of the dataset cannot be overstated, as it serves as the bedrock upon which the ML algorithms are trained and subsequently evaluated. The dataset's robustness, encompassing a wide range of airfoil shapes and configurations, is instrumental in ensuring the ML models' adaptability and generalization capabilities. This diversity ensures that the neural network can effectively learn intricate relationships between airfoil shapes and their corresponding aerodynamic performance, such as the Lift-to-Drag Ratio. Moreover, this open dataset fosters transparency and reproducibility in research, allowing others in the scientific community to validate and build upon the findings presented in this study.

3.1. *Artificial Intelligence*

Artificial Intelligence (AI), particularly Machine Learning, constitutes a multifaceted domain that encompasses diverse methodologies catering to various data-driven challenges. In broad strokes, ML can be dichotomized into two principal classes: supervised learning and unsupervised learning, each addressing distinct problem-solving paradigms. The hallmark of supervised learning hinges on its reliance on labelled data, necessitating input data explicitly paired with corresponding target outcomes, enabling the model to discern patterns and relationships. In contrast, unsupervised learning harnesses unlabelled data, where the model endeavours to unveil inherent structures or groupings within the data autonomously.

To elucidate the fundamental tenets of ML further, consider the overarching architecture depicted in Fig. 1, serving as a blueprint for ML systems. The figure illustrates the fundamental structure of a machine learning system, encapsulating key phases: data foundation, model mastery, and the iterative process of training and evaluation. In the context of this study, we delve into the realm of supervised learning, a cornerstone of ML, where our models are meticulously trained using datasets consisting of parametrized airfoils meticulously paired with their associated Lift-to-Drag Ratio values. This supervised approach empowers the ML models to harness historical data patterns and make predictions based on newly encountered input.

Within the expansive landscape of ML methodologies, a plethora of algorithms and techniques exist to address a myriad of problems. The literature abounds with noteworthy contributions, encompassing gradient boosting regression [15], random forest [16], AdaBoost [17], decision tree [18], linear regression, and radial basis function neural networks, each offering its unique strengths and suitability for diverse applications.

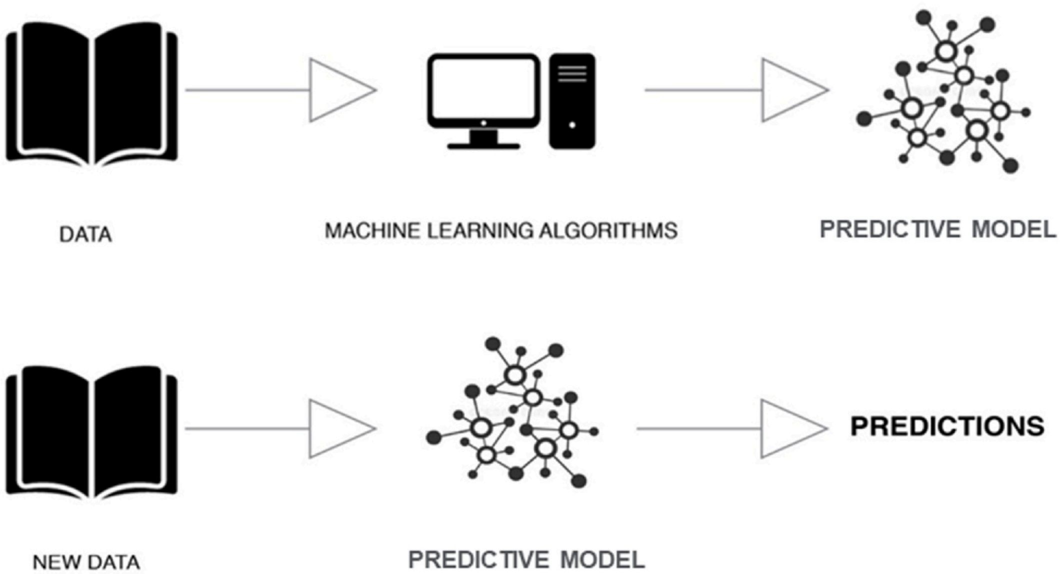


Fig. 1. General structure for machine learning system.

The choice of algorithm becomes pivotal, as it profoundly influences the model's performance and its ability to discern intricate relationships between airfoil parameters and the consequential aerodynamic metric, the LDR.

3.1.1. Gradient boosting regressor

Boosting is a technique of combining multiple simple models into a single composite model. However, as all weak learner models are added over time, the structure of the existing trees will not be affected, and the final model will yield a robust predicted result. The gradient algorithm is also designed to minimize the loss in the data, unlike linear regression. It is one of the most popular machine learning algorithms for tabular datasets, as it finds a nonlinear relationship between the model targets and the features, even identifying missing values. Gradient boosting is an algorithm that produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees, for regression and classification problems. It constructs the model in a stage-wise manner, like other boosting methods, and generalizes them by allowing optimization of any differentiable loss function [19].

3.1.2. Random forest

The random forest (RF) algorithm aims to amalgamate multiple random trees within a bagging ensemble model, a technique that often delivers excellent outcomes across a range of classification problems (as demonstrated in Fig. 2). In an RF model, the output variable is typically an average of the predictions from regression trees. Node splitting in this model is conducted using a finite subset of randomly selected features [16]. RF modelling involves creating trees through various training tests based on a random vector that remains independent of any other data within the distribution. However, to analyse errors effectively, it becomes essential to compute two critical parameters: (i) accuracy and (ii) interdependence among individual classifiers [20].

In the classification process, the RF algorithm accounts for the common distribution of errors resulting from the contributions of each base classifier. To mitigate these errors, the RF algorithm employs a final reduction step to enhance the classification effect, as outlined by Sulaiman [21].

The prediction in the RF algorithm is achieved through the adaptation of the test characteristic and the application of rules for each random decision tree. These trees collectively predict the result through a voting mechanism, where prediction goals are tallied, and the most highly voted outcome is selected as the final prediction, as discussed by Kolhe et al. [22].

The training process of the Random Forest model is visually depicted in Fig. 3. Initially, one RF formation must be defined, and a classifier for the formation is estimated. Many researchers have employed a formula to elucidate the final decision calculation by considering a test model and applying the rules of each decision tree created within the RF model, thereby determining the forecasted outcome.

$$H(x) = arg\max_{y} \sum_{i=1}^{k} I(h_i(x) = Y) \tag{1}$$

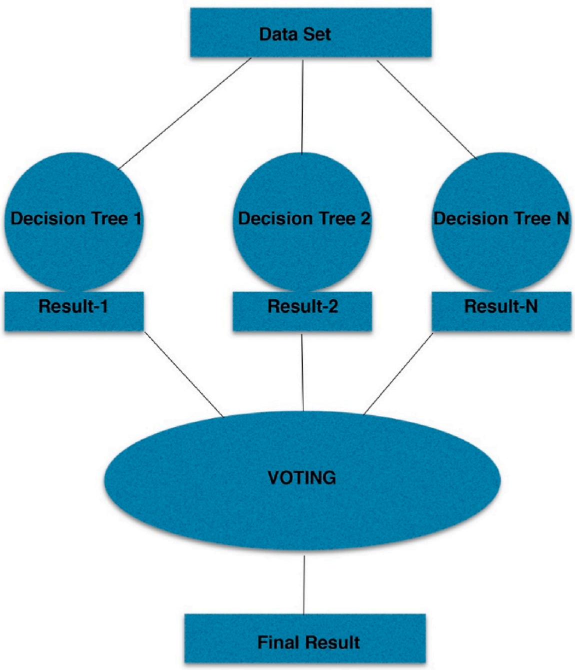


Fig. 2. Flow chart of random forest.

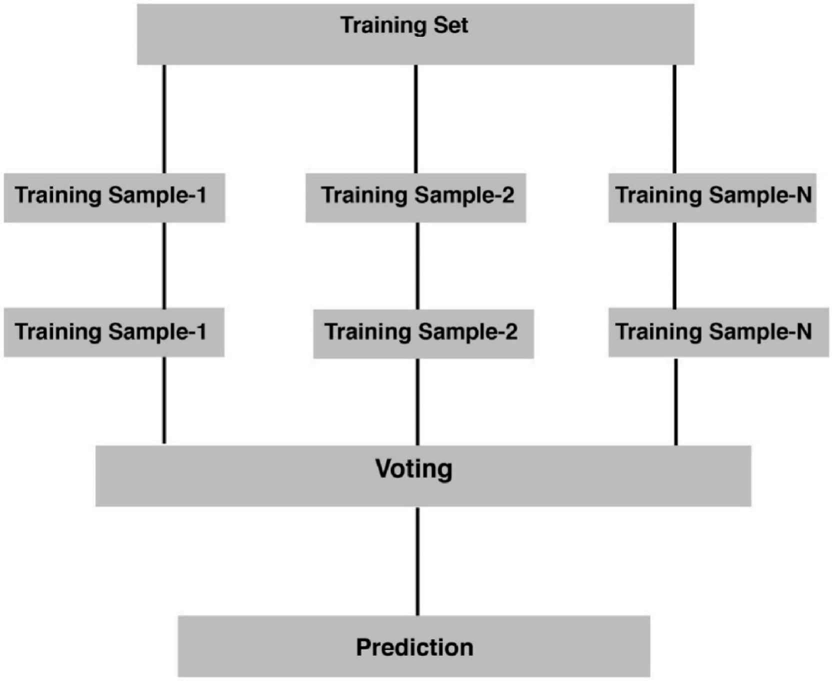


Fig. 3. Random Forest training flow chart.

Where x is the test sample, h_i is the single Decision Tree, y is the targeted data, and I is the indicator function that leads to the RF model of H. It is also the classification outcome of each test tree for the test sample with the maximum number of votes [23–25].

3.1.3. AdaBoost prediction

RF and AdaBoost are both examples of tree-based ensemble regressors that utilize individual regression trees' predictions to formulate their overall predictions. However, they employ fundamentally different approaches to assemble these tree ensembles.

RF, for instance, follows the Bootstrap Aggregating (bagging) concept, where each tree is developed using a distinct subset of the available training data. This approach aims to enhance the diversity of individual predictions, and the ensemble's prediction is derived by averaging the predictions made by the individual trees.

In contrast, AdaBoost adopts a sequential tree growth strategy, where each new tree focuses on rectifying the errors or shortcomings of the preceding one. AdaBoost's predictions are constructed based on this sequential refinement process [26]. These

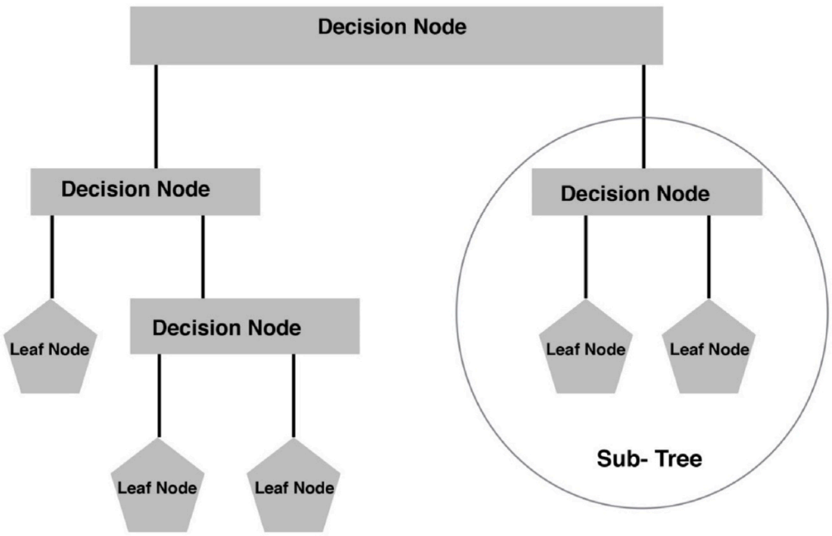


Fig. 4. Decision Tree training flow chart.

distinctions in their ensemble construction approaches contribute to their varying characteristics and performance in predictive modelling.

3.1.4. Decision tree

Decision tree (DT) models are versatile non-parametric algorithms suitable for both classification and regression tasks. These models tackle these tasks by acquiring straightforward if-then-else decision rules based on the available features. One notable characteristic of DT models is their transparency, often referred to as white-box models, as they allow explanations for predictions through Boolean logic.

However, the complexity of the decision rules grows as the tree's depth increases, potentially leading to overfitting, where the model fits the training data too closely. To mitigate this, it is crucial to carefully select and fine-tune the maximum depth hyperparameter before training and during subsequent model refinements. Additionally, DTs can exhibit bias towards dominant classes or densely populated numerical regions within the training dataset, emphasizing the importance of achieving proper class balance in the training data [27].

The decision tree algorithm involves constructing a tree-shaped model and partitioning information into subsets to assess various courses of action and determine predictive probabilities [28]. Each response is explained by a different branch of the decision tree. Several algorithms can be applied within the decision tree technique to determine how to split a node into two sub-nodes, with the goal of enhancing the homogeneity of resulting sub-nodes [26]. The process of finding the most homogeneous sub-nodes entails dividing the nodes based on all variables and then selecting the optimal split [29]. For a visual representation of the training flow chart for the Decision Tree model utilized in this study, please refer to Fig. 4.

3.1.5. Linear regression

Linear regression is a statistical method employed to establish a connection between the value of a dependent variable and that of an independent variable. This algorithm serves to enhance the understanding of the relationship between the outcome and the calculated values of predictor variables while also accounting for the influence of covariates or confounding factors. As elucidated by Azman et al. [30], linear regression is a modelling approach used to characterize the association between a single scalar variable, denoted as Y, and one or more explanatory variables represented by X. This relationship can be expressed through the following equation:

$$Y = mX + a \tag{2}$$

Where the values of the independent variable "*X*" will be defined, and for each value of "*X*" there is a subpopulation of "*Y*" variables that are homogeneous, "*m*" is the slope, and "*a*" is the intercept.

3.2. Implementation of algorithms

Once the ML models are developed, evaluating their performance becomes crucial. The intricacies of these methods can be scrutinized and assessed using various error metrics and accuracy measures as proposed in the literature. It is worth noting that the choice of performance evaluation criteria in ML applications can vary depending on the specific context. This variability arises because there is no universal criterion that can comprehensively capture all the nuances of errors within a given prediction method. For instance, in the realm of regression analysis, performance evaluation often relies on metrics such as Mean Squared Error and the coefficient of determination. Therefore, in this study, several statistical techniques have been employed to gauge the performance effectiveness of the investigated models [31]. The following equations outline the calculation of these evaluation criteria.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2 \tag{3}$$

Where n is the total number of samples, and y_i and x_i are the ground truth and predicted aerodynamic performance value, respectively.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \overline{Y})^2} \tag{4}$$

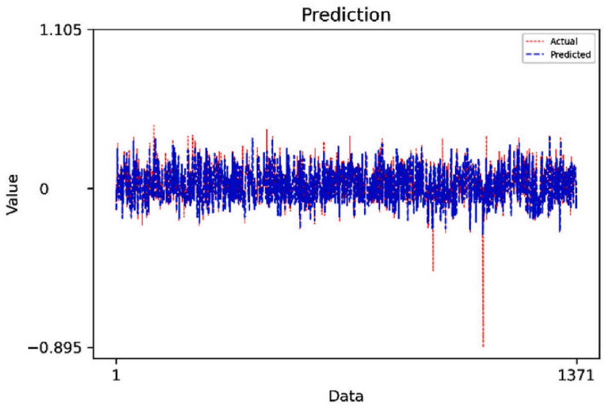
Where Y_i is the actual value, $\widehat{Y}_i$ is the predicted output value, and n is the total number of samples.

While the value of R^2 approaches unity for a suitable model and demonstrates better prediction, low values of MSE represent a better prediction of a model.

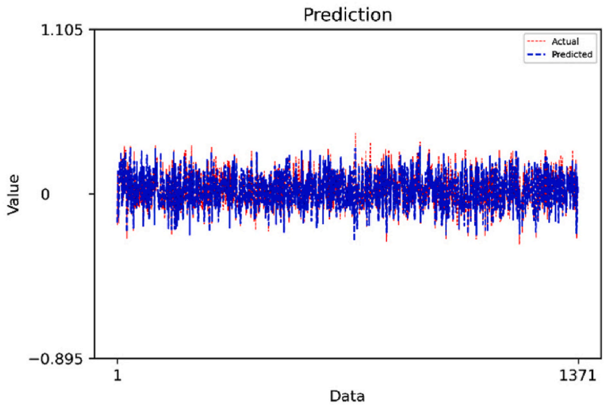
3.3. Airfoil data

The dataset used in this research was originally compiled in a previous study and is accessible as an open source on [14]. This dataset serves as the foundation for implementing an airfoil lift-to-drag ratio prediction method using Convolutional Neural Networks (CNNs). The objective of this approach is to model the contour of airfoils and, subsequently, employ CNNs to predict their lift-to-drag ratios, a pivotal aerodynamic parameter.

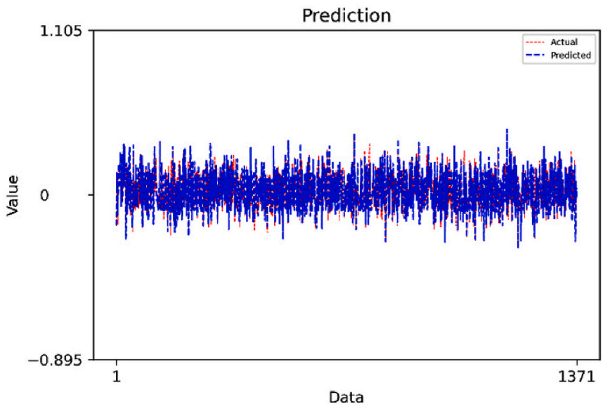
The data preparation and processing methodology adopted in this dataset creation commence with the acquisition of raw airfoil coordinates, sourced from the UIUC Airfoil Data Site. To ensure uniformity and compatibility as input data, these coordinate pairs are meticulously transformed into high-resolution grayscale images, each boasting dimensions of 128x128 pixels. This choice of image resolution stems from the recognition of the airfoil's extreme sensitivity to even minor shape variations. In a bid to maximize the neural network's capacity to learn, contour-filling techniques are judiciously applied to preserve valuable information within the contour images. Furthermore, the dataset accounts for a range of angles of attack through the rotation of the filled-in contour images, yielding twenty-five distinct images per original airfoil sample. Post-binarization and flattening, the data from all these samples are consolidated into a single input matrix, with values ranging from 0 to 1.



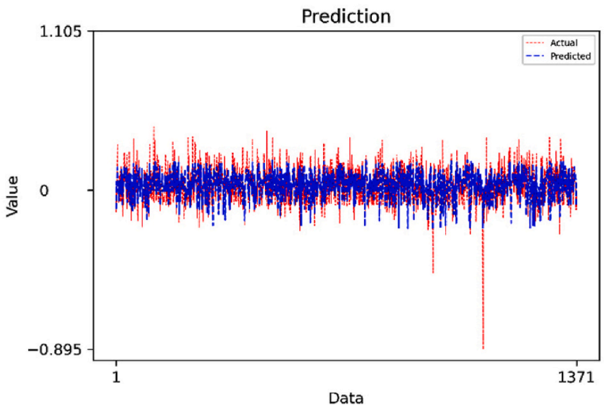
(a) Random Forest



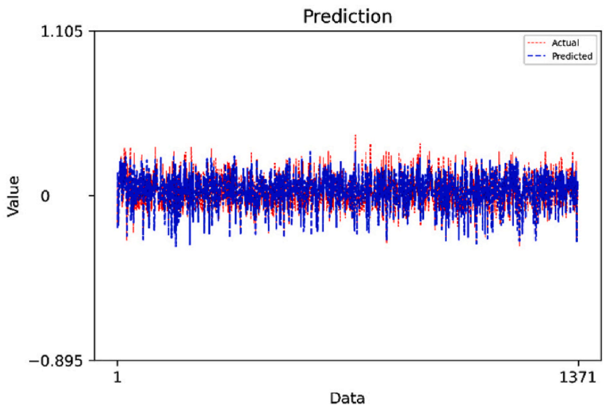
(b) Gradient Boosting Regressor



(c) Decision Tree Regressor



(d) AdaBoost Prediction



(e) Linear Regressor

Fig. 5. Prediction plots of the employed Machine Learning algorithms.

To ascertain the ground truth values, the researchers utilize Computational Fluid Dynamics software, specifically xflr5, to compute lift-to-drag (LDR) ratios. This involves importing airfoil data and generating corresponding contour figures. The software calculates LDR across a spectrum of angles of attack, automatically considering the influence of α. The results, accompanied by their respective angles of attack, are meticulously recorded, and exported as individual samples. This meticulous data preparation and ground truth calculation procedure lays the foundation for training the neural network model on a diverse and highly accurate dataset, enabling it to make precise predictions of lift-to-drag ratios based on airfoil geometry and angle of attack.

The process of input data preparation is pivotal in training the ML model to predict an airfoil's aerodynamic performance, specifically its lift-to-drag ratio, contingent on the airfoil's geometric properties and angle of attack. The dataset pairs parametrised airfoil geometry data with corresponding LDR values, which are computed using computational methods. Each airfoil is evaluated across a range of angles of attack (α), spanning from -5° to $+20^\circ$ with an increment of 0.25°. This incremental approach results in the generation of one hundred images for each airfoil, contributing to a comprehensive and versatile dataset for training and evaluating the ML model.

4. Results and discussion

The achievement of the study's objectives hinged on the meticulous implementation and comprehensive assessment of ML models, all executed through the TensorFlow Python library. Five distinct ML algorithms, namely Random Forest (RF), Gradient Boosting Regression (GBR), Decision Tree Regressor (DTR), AdaBoost, and Linear Regression (LR), were thoughtfully selected for the crucial task of predicting the Lift-to-Drag Ratio. This aerodynamic performance indicator is of paramount importance and is dependent on the precisely curated input data detailed in the preceding section. The choice of these algorithms was underpinned by their versatility and aptitude for unravelling the intricate relationship between airfoil shapes and LDR.

The rigorous evaluation process applied to these machine learning models encompassed a multifaceted analysis. It entailed a meticulous examination of key evaluation metrics, including the Coefficient of Determination (R^2), which serves as a barometer of predictive accuracy, and the MSE, a vital indicator of prediction precision. Beyond these critical metrics, the computational facets of training time and evaluation time were subject to thorough scrutiny. These assessments were conducted across a spectrum of train/test ratios, ranging from 0.2 to 0.8, mirroring diverse data split scenarios. This comprehensive evaluation framework aimed to gauge the models' efficacy and performance under various conditions, ensuring a robust analysis of their capabilities.

Fig. 5 presents a comparative analysis between actual and predicted aerodynamic performance, offering a compelling visual representation of the models' predictive ability. This visualization focuses on the outcomes derived from a train/test ratio of 0.2. The results, as illustrated in the figure, clearly affirm the models' exceptional ability to accurately forecast LDR, underscoring their robustness and reliability in capturing intricate relationships within airfoil data. In summary, this study underscores the profound potential of machine learning algorithms to effectively model and predict critical aerodynamic parameters, thereby laying a solid foundation for enhanced airfoil design and optimization across a spectrum of engineering applications.

Table 1 presents a comprehensive overview of the accuracy achieved by all five ML models across various train/test ratios, spanning from 0.2 to 0.8. For clarity, the highest and lowest accuracy values attained by each model are highlighted in the table. Notably, the Random Forest (RF) prediction model outshines the others with the highest R^2 score of 0.944, demonstrating a robust correlation between predicted and actual values. In contrast, the AdaBoost model lags, yielding the lowest R^2 score of 0.689 among the investigated ML models.

Upon closer inspection of the table, it becomes evident that the RF, Gradient Boosting Regression (GBR), and Decision Tree Regressor (DTR) models exhibit their peak accuracy at a train/test ratio of 0.2. Conversely, the Linear Regression (LR) and AdaBoost models achieve their highest accuracy at train/test ratios of 0.3 and 0.7, respectively. Notably, an increase in the train/test ratio corresponds to a reduction in the accuracy of the RF and GBR models, while the AdaBoost model excels at higher train/test ratios, specifically 0.7 and 0.8.

Transitioning to Table 2, we delve into the performance evaluation of the ML models, focusing on the train/test ratio that yielded the highest accuracy. The table provides a comprehensive view of the evaluation metrics such as R^2, MSE, training time and evaluation time for each machine learning algorithm. While RF and GBR demonstrate top-tier accuracy, they demand a substantial training time compared to the other models. In stark contrast, LR boasts significantly shorter training times, making it a viable choice for projects with strict computational constraints. In terms of evaluation time, AdaBoost, despite its lower accuracy among the ML algorithms, stands out with a noteworthy duration of 17.35 s for result prediction, while LR boasts the shortest evaluation time. These insights shed

Table 1
Accuracy of Predicted Results for various train/test ratios.

	Train/Test Ratio						
	0.2	0.3	0.4	0.5	0.6	0.7	0.8
ML Algorithm	R^2						
RF	**0.944**	0.93	0.918	0.91	0.892	0.88	**0.863**
GBR	**0.896**	0.881	0.871	0.868	0.859	0.853	**0.848**
DTR	**0.857**	0.823	0.828	0.812	0.793	0.775	**0.757**
AdaBoost	0.709	**0.689**	0.697	0.721	0.72	**0.747**	0.743
LR	0.831	**0.832**	0.797	0.791	**0.772**	0.775	0.776

Table 2
Performance evaluation metrics of the ML algorithms.

ML Algorithm	train/test ratio	R^2	MSE	Training time	Evaluation time
RF	0.2	0.944	0.0008	271.98s	0.23s
GBR	0.2	0.896	0.0014	197.30s	0.16s
DTR	0.2	0.857	0.0020	4.43s	0.10s
AdaBoost	0.7	0.747	0.0037	43.41s	17.35s
LR	0.3	0.832	0.0027	0.33s	0.01s

light on the trade-offs between accuracy and computational efficiency, allowing for informed selection of ML algorithms tailored to specific project needs and resource limitations.

5. Conclusion

Performance prediction plays a pivotal role in the advancement of airfoil design, particularly in applications such as aviation and wind energy. Presently, machine learning algorithms have emerged as powerful tools, offering superior efficiency and feasibility for rapid investigations in the realm of aerodynamics. This research delved into the application of ML algorithms for predicting aerodynamic performance, leveraging the TensorFlow Python library for comprehensive model development, training, and testing. The outcomes of this investigation unveiled commendable levels of accuracy across all proposed models. Notably, most of the examined algorithms achieved their peak performance, as indicated by the highest R^2 scores, with a test size of 0.2. The assessment of these models encompassed a range of evaluation metrics, including R^2, MSE, Training time, and Evaluation time. It came to light that Random Forest, Gradient Boosting Regression, and Decision Tree Regressor exhibited superior accuracy, specifically at a train/test ratio of 0.2, while AdaBoost and Linear Regression excelled at train/test ratios of 0.7 and 0.3, respectively. In the context of R^2 as an evaluation metric, the Random Forest prediction model emerged as the most accurate, while the AdaBoost prediction model showcased slightly lower accuracy. Furthermore, concerning training time and evaluation time, Linear Regression exhibited a substantial advantage over the other ML algorithms investigated in this study.

Future research endeavours will focus on refining these models to achieve even higher accuracy levels. Additionally, there is a keen interest in exploring the application of these well-suited models in predicting flow field properties, such as velocity and pressure fields, further expanding their utility and impact in the field of aerodynamics.

Data availability

Data will be made available on request.

Funding

Not applicable.

Replication of results

Results can be replicated by the formulas provided in the methods section.

CRediT authorship contribution statement

Amir Teimourian: Investigation, Data curation, Conceptualization. **Daniel Rohacs:** Validation, Supervision, Funding acquisition. **Kamil Dimililer:** Validation, Resources, Formal analysis, Conceptualization. **Hanifa Teimourian:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis. **Melih Yildiz:** Writing – review & editing, Writing – original draft, Visualization, Validation. **Utku Kale:** Writing – review & editing, Writing – original draft, Supervision, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by Project No. 2022–2.1.1-NL-2022-00012, which has been implemented with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the National Laboratories funding scheme.

References

[1] B. Tracey, K. Duraisamy, J. Alonso, Application of supervised learning to quantify uncertainties in turbulence and combustion modeling, in: The *51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2013.
[2] B. Aksoy, R. Selbaş, Estimation of wind turbine energy production value by using machine learning algorithms and development of implementation program. Energy sources, Part A: recovery, Utilization, and Environmental Effects 43 (6) (2021) 692–704.
[3] P.S. Bithas, E.T. Michailidis, N. Nomikos, D. Vouyioukas, A.G. Kanatas, A survey on machine-learning techniques for UAV-based communications, Sensors 19 (23) (2019) 5170.
[4] S. Aslan, D. Rohacs, M. Yildiz, K. Utku, A three-dimensional UCAV path planning approach based on immune plasma algorithm, Int. J. Comput. Intell. Syst. 16 (1) (2023), https://doi.org/10.1007/s44196-023-00284-9.
[5] G. Sun, S. Wang, A review of the artificial neural network surrogate modeling in aerodynamic design, in: Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 2019, pp. 5863–5872, 233(16).
[6] H. Nowruzi, H. Ghassemi, M. Ghiasi, Performance predicting of 2D and 3D submerged hydrofoils using CFD and ANNs, J. Mar. Sci. Technol. 22 (4) (2017) 710–733.
[7] K. Portal-Porras, U. Fernandez-Gamiz, A. Ugarte-Anero, E. Zulueta, A. Zulueta, Alternative artificial neural network structures for turbulent flow velocity field prediction, Mathematics 9 (16) (2021) 1939.
[8] H. Chen, L. He, W. Qian, S. Wang, Multiple aerodynamic coefficient prediction of airfoils using a convolutional neural network, Symmetry 12 (4) (2020) 544.
[9] V. Sekar, Q. Jiang, C. Shu, B.C. Khoo, Fast flow field prediction over airfoils using deep learning approach, Phys. Fluids 31 (5) (2019) 057103.
[10] Y. Wang, L. Deng, Y. Wan, Z. Yang, W. Yang, C. Chen, D. Zhao, F. Wang, Y. guo, An intelligent method for predicting the pressure coefficient curve of airfoil-based conditional generative adversarial networks, IEEE Transact. Neural Networks Learn. Syst. (2021), https://doi.org/10.1109/TNNLS.2021.3111911.
[11] K. Portal-Porras, U. Fernandez-Gamiz, E. Zulueta, et al., CNN-based flow control device modelling on aerodynamic airfoils, Sci. Rep. 12 (2022) 8205.
[12] X. Hui, J. Bai, H. Wang, Y. Zhang, Fast pressure distribution prediction of airfoils using deep learning, Aero. Sci. Technol. 105 (2020) 105949, https://doi.org/10.1016/j.ast.2020.105949.
[13] E.R. Lalonde, B. Vischschraper, G. Bitsuamlak, K. Dai, Comparison of neural network types and architectures for generating a surrogate aerodynamic wind turbine blade model, J. Wind Eng. Ind. Aerod. 216 (2021) 104696, https://doi.org/10.1016/j.jweia.2021.104696.
[14] Github, https://github.com/ziliHarvey/CNN-for-Airfoil (Accessed 26.June.2022).
[15] P. Friederich, M. Krenn, I. Tamblyn, A. Aspuru-Guzik, Scientific intuition inspired by machine learning-generated hypotheses, Mach. Learn.: Sci. Technol. 2 (2) (2021) 025027.
[16] I.A. Guedes, A.M.S. Barreto, D. Marinho, E. Krempser, M.A. Kuenemann, O. Sperandio, L.E. Dardenne, M.A. Miteva, New machine learning and physics-based scoring functions for drug discovery, Sci. Rep. 11 (1) (2021) 1–19.
[17] D.P.P. Meddage, I.U. Ekanayake, A.U. Weerasuriya, C.S. Lewangamage, Tree-based regression models for predicting external wind pressure of a building with an unconventional configuration, in: 2021 Moratuwa Engineering Research Conference (MERCon), IEEE, 2021, https://doi.org/10.1109/MERCon52712.2021.9525734.
[18] B. Yoon, A machine learning approach for efficient multi-dimensional integration, Sci. Rep. 11 (1) (2021) 1–8.
[19] S. Fafalios, P. Charonyktakis, I. Tsamardinos, Gradient Boosting Trees, 2020.
[20] B.O. Yigin, O. Algin, G. Saygili, Comparison of morphometric parameters in prediction of hydrocephalus using random forests, Comput. Biol. Med. 116 (2020) 103547.
[21] M.A. Sulaiman, Evaluating data mining classification methods performance in internet of things applications, Journal of Soft Computing and Data Mining 1 (2) (2020) 11–25.
[22] M.L. Kolhe, M.C. Trivedi, S. Tiwari, V.K. Singh, in: Advances in Data and Information Sciences Proceedings of ICDIS-2017, Proceedings of ICDIS, 2019 vol. 2.
[23] V.Y. Kulkarni, P.K. Sinha, Pruning of random forest classifiers: a survey and future directions, in: *2012 International Conference on Data Science & Engineering* (ICDSE), IEEE, 2012.
[24] K. Das, RN Behera, A survey on machine learning: concept, algorithms and applications, International Journal of Innovative Research in Computer and Communication Engineering 5 (2) (2017) 1301–1309.
[25] Y. Sun, Y. Li, Q. Zeng, Y. Bian, Application research of text classification based on random forest algorithm, in: *2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE*), IEEE, 2020, https://doi.org/10.1109/AEMCSE50948.2020.00086.
[26] G. Vergos, S.P. Sotiroudis, G. Athanasiadou, G.V. Tsoulos, S.K. Goudos, Comparing machine learning methods for air-to-ground path loss prediction, in: 2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST), IEEE, 2021, https://doi.org/10.1109/MOCAST52088.2021.9493374.
[27] P. Vicente-Valdez, L. Bernstein, M. Fratoni, Nuclear data evaluation augmented by machine learning, Ann. Nucl. Energy 163 (2021) 108596.
[28] D. Zebari, D. Zeebaree, J.N. Saeed, N.A. Zebari, A.A. Al-zebari, Image steganography based on swarm intelligence algorithms: a survey, Test Eng. Manag. 7 (8) (2020) 9.
[29] G.K. Kumar, P. Viswanath, A.A. Rao, Ensemble of randomized soft decision trees for robust classification, Sādhanā 41 (3) (2016) 273–282.
[30] A. Azman, F.A. Abdullah, S. Yogarayan, S.F.A. Razak, H. Azman, K.S. Muthu, H. Suhalia, Measuring driver cognitive distraction through lips and eyebrows, Int. J. Electr. Comput. Eng. 12 (1) (2022), 2088-8708).
[31] A. Radonjic, K. Vukadinovic, Application of ensemble neural networks to prediction of towboat shaft power, J. Mar. Sci. Technol. 20 (1) (2015) 64–80.