engineering proceedings

MDPI

Proceeding Paper

Comparative Study of Random Forest and Gradient Boosting Algorithms to Predict Airfoil Self-Noise †

Shantaram B. Nadkarni, G. S. Vijay and Raghavendra C. Kamath *

Department of Mechanical and Manufacturing Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, Karnataka, India; nadkarni1.mitmpl2022@learner.edu (S.B.N.); vijay.gs@manipal.edu (G.S.V.)

* Correspondence: cr.kamath@manipal.edu

† Presented at the International Conference on Recent Advances on Science and Engineering, Dubai, United Arab Emirates, 4–5 October 2023.

Abstract: Airfoil noise due to pressure fluctuations impacts the efficiency of aircraft and has created significant concern in the aerospace industry. Hence, there is a need to predict airfoil noise. This paper uses the airfoil dataset published by NASA (NACA 0012 airfoils) to predict the scaled sound pressure using five different input features. Diverse Random Forest and Gradient Boost Models are tested with five-fold cross-validation. Their performance is assessed based on mean-squared error, coefficient of determination, training time, and standard deviation. The results show that the Extremely Randomized Trees algorithm exhibits the most superior performance with the highest Coefficient of Determination.

Keywords: airfoil self-noise; random forest; extra trees; gradient boosting; XGBoost; feature importance

check for updates

Citation: Nadkarni, S.B.; Vijay, G.S.; Kamath, R.C. Comparative Study of Random Forest and Gradient Boosting Algorithms to Predict Airfoil Self-Noise. *Eng. Proc.* **2023**, *59*, 24. https://doi.org/10.3390/engproc2023059024

Academic Editors: Nithesh Naik, Rajiv Selvam, Pavan Hiremath, Suhas Kowshik CS and Ritesh Ramakrishna Bhat

Published: 12 December 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The sound pollution caused by airplanes, windmills, and other sources has been a matter of interest for the research community for many years. Airfoil self-noise arises from the interaction between the blade of an airfoil and the turbulence generated in the boundary layer near the disturbed flow. The obstruction of uniform and steady airflow by the airfoil creates turbulence, leading to the generation of eddies that fluctuate in nature and make noise. The noise becomes louder as the eddies interact with the airfoil surface. The mechanisms of noise generation are illustrated in Figure 1.

Airfoil self-noise can have an impact on the system's overall efficiency. Excessive turbulence in the airflow increases the drag, which can decrease the lift and make aircraft engines work harder to maintain the desired speed and elevation. Added to this is the environmental impact of airfoil self-noise; it causes noise pollution that can disturb wildlife and humanity. Airfoil noise is one reason that limits the use of helicopters and even wind turbines in urban areas. Airfoil self-noise is influenced by various factors with complex interactions, such as the shape of the airfoil, characteristics of the flow characteristics, etc., as shown in Figure 1. Constructing prototypes for physically testing these airfoil designs is very expensive and laborious.

Hence, applying machine learning (ML) to build predictive models enables researchers to explore various design possibilities while saving significant resources. Machine learning helps to extract the necessary information from existing data and generates clear-cut simulations during the early design stages.

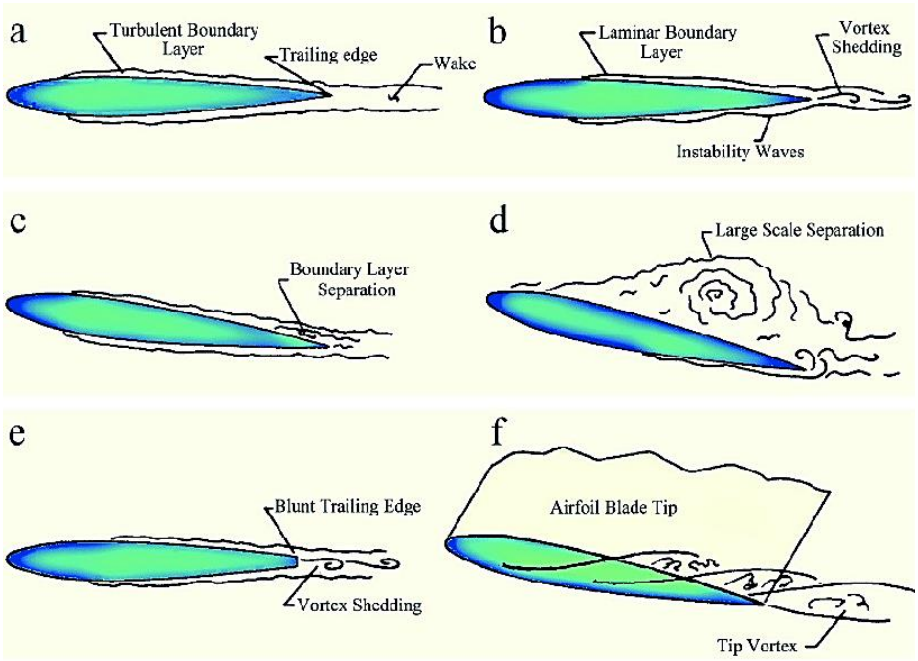


Figure 1. Various self-noise generation mechanisms: (**a**) noise at trailing edge due to turbulent boundary layer; (**b**) noise due to vortex-shedding with laminar boundary layer; (**c**) separation-stall noise with a slight angle of attack; (**d**) separation-stall phenomenon with a large angle of attack; (**e**) noise due to vortex-shedding and trailing-edge bluntness; (**f**) tip vortex noise [1].

More than 30 years ago, Brooks et al. [2] presented the development of a comprehensive prediction approach for airfoil self-noise using semi-empirical mechanisms based on various theoretical investigations and data gathered from a series of aerodynamic experiments on isolated airfoil sections. The results demonstrated a precise predictive capability for turbulent boundary layer noise and separation noise, but there was uncertainty in applying this method for diverse airfoil geometries. Moreau et al. [3] found that in wind tunnel testing, the effects of jet flow cause discrepancies in examining airfoil self-noise; hence, jet interference effects should be incorporated.

The potential of applying neural networks in modeling airfoil noise was tested when authors used polynomial approximation to parameterize the shape of the airfoils [4]. The neural network proved to be more accurate than empirical ways but less sensitive to boundary layer values as input. Symbolic regression for airfoil noise prediction was demonstrated to enhance existing modeling strategies [5,6]. Symbolic regression, unlike traditional regression, aims to discover mathematical expressions or equations that best relate to a given dataset.

The genetic algorithm was applied to optimize the self-noise produced by a 10 kW windmill [7]. The validation experiment revealed that the airfoil noise was significantly reduced despite maintaining the same aerodynamic performance. Another group of researchers attempted to predict the scaled sound pressure levels using a linear regression model [8]. They found that although linear regression is a simple and robust algorithm, there is a need to explore more advanced modeling techniques. The effect of airfoil curvature and thickness on noise production was examined using semi-empirical methods to find that the relationship between noise and lift is independent of camber [9]. A similar work was published to evaluate the displacement thickness of the turbulent boundary layer using semi-empirical methods [10].

Random Forest (RF) and Stochastic Gradient Tree Boosting (SGTB) for predicting airfoil self-noise were proposed where models used the NASA airfoil dataset [1]. Results showed that SGTB outperformed RF specifically for this dataset; however, RF seemed more convenient due to its significantly lesser computational time. A comparison of the speed

and accuracy of XGBoost and the traditional Gradient Boost Algorithm was conducted, which discovered that XGBoost enabled the creation of individual trees using multiple cores, reducing training time [11,12]. Another group of authors proposed a weighted KNN approach that assigned weights to the attributes obtained through feature selection methods and performed better than traditional KNN [13].

Genetic programming combined with adaptive regression showed promising results at low-mach-number turbulent flows for predicting airfoil noise [14]. Principal Component Analysis (PCA) integrated with a neural network or using quasi-Newtonian parameter optimization enhanced the prediction of self-noise, and the model performed better than quadratic or cubic regression [15,16]. A CatBoost Algorithm combined with Arithmetic Optimization reduced computational times and was a cost-effective method [17]. In another study, a decision tree was used to partition the training samples, and then support vector regression was trained to assess the regression relationship; this led to improved prediction accuracy [18]. The latest research study highlights a comprehensive evaluation of machine learning algorithms across multiple aerospace applications, showing the highest accuracy for KNN, Decision Tree, and Histogram Gradient Boosting (GB) for airfoil self-noise prediction [19].

In this paper, the airfoil dataset NACA 0012 is considered to envisage the scaled sound pressure using five different input features. Various RF and GB models are tested with five-fold cross-validation. Their performance is assessed based on mean-squared error, coefficient of determination, training time, and standard deviation. The results show that the Extremely Randomized Trees algorithm exhibits the most superior performance with the highest Coefficient of Determination. In contrast, the GB Regressor offers an advantage in terms of the least training time for the given dataset.

2. Materials and Methodology

2.1. Dataset Description

The airfoil self-noise data used in the study were compiled by a sequence of aerodynamic and acoustic examination of blade sections (two and three-dimensional) of an airfoil. These tests were carried out by NASA in 1989, and they included different sizes of NACA0012 airfoils, with a specific airfoil shape positioned with a fixed span against different wind speeds at varied angles. When subjected to smooth airflow, these tests measured the scaled sound pressure in dB or noise produced by the airfoils. The NASA data set includes a total of 1503 entries and consists of the following parameters as shown in Table 1, Source: "https://www.kaggle.com/datasets/fedesoriano/airfoil-selfnoise-dataset (accessed on 25 June 2023)".

Table 1. Attributes of NACA 0012 airfoils.

Features	Range of Values
Frequency (Hz)	21 discrete values between 200 and 20,000
The angle of attack (o)	27 discrete values between 0 and 22.2
Chord length (m)	Six discrete values between 0.0254 and 0.3048
Free stream velocity (m/s)	Four discrete values between 31.7 and 71.3
Boundary layer thickness (m)	Continuous values between 0.0004 and 0.058
Sound pressure (dB)	Continuous values between 103.3 and 140.9

The physical relevance of inputs is crucial for accurate predictions. Frequency is the rate of pressure fluctuation around the test object that creates disturbances and propagates as sound waves. The intensity of the resulting sound or noise corresponds to the original pressure fluctuation frequency. At the same time, varying the angle of attack or inclination between the chord and the incoming airflow changes the turbulence levels around the airfoil. Higher grades may disrupt the flow, leading to more turbulence and more noise. Free stream velocity defines the speed of air moving freely before interacting with an airfoil. Higher velocity means more momentum in the airflow and more noise generation. The

boundary layer is a critical region where the airflow velocity changes from zero at the airfoil surface to the free stream. Its thickness depends on airfoil shape characteristics. A thicker boundary layer can have more significant pressure gradients and is more prone to turbulence, hence more noise. Similarly, chord length decides the overall size of the airfoil and boundary layer, which affects the noise.

2.2. Dataset Preprocessing and Visualization

The data did not have any missing values and were used without undergoing any normalization process. However, data randomization was carried out to omit any biases in the original order of data, and five-fold cross-validation was employed to assess the model's performance. Since there are five input features and only one output feature (scaled sound pressure), a pair plot data visualization was used to explore their relationships; the pair plot creates a matrix of scatterplots, where each scatterplot represents the relationship between two given variables, as shown in Figure 2.

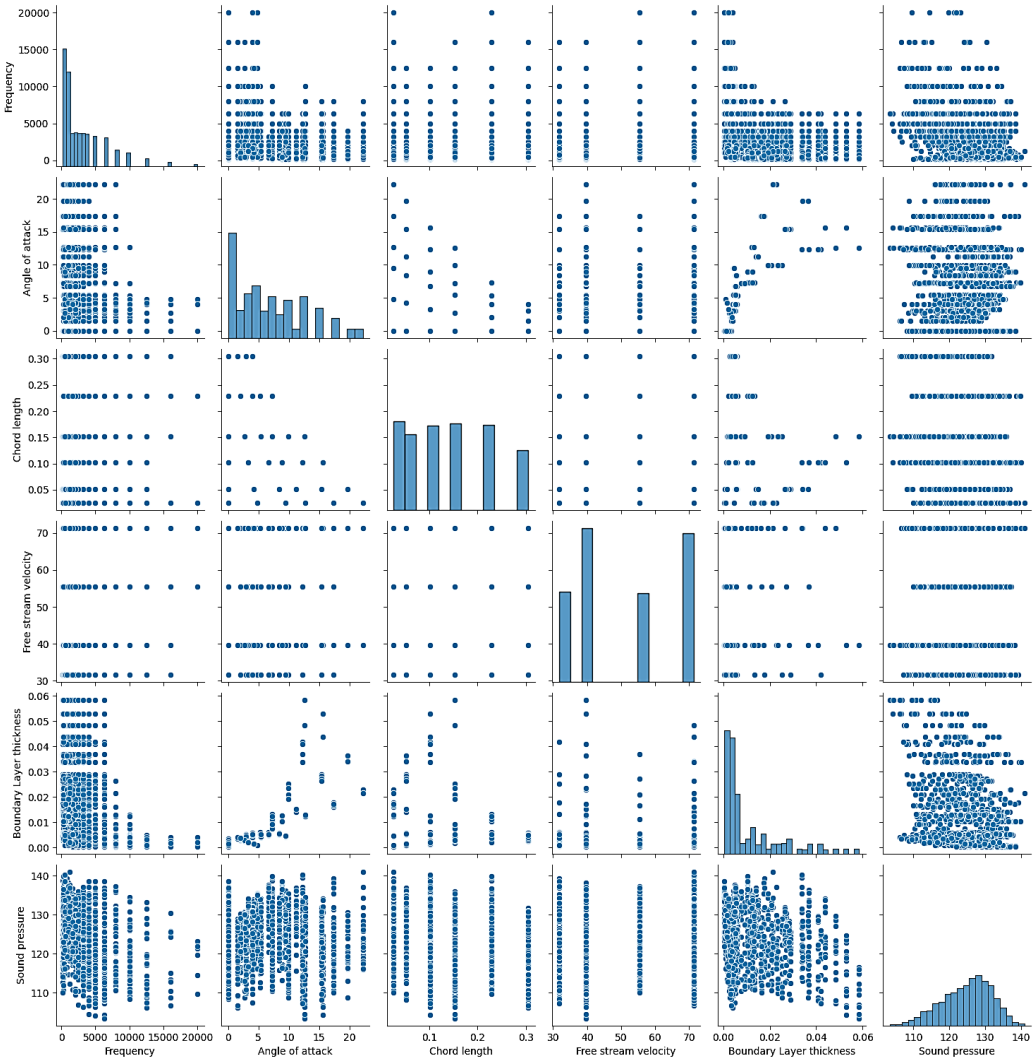


Figure 2. Pair-plot of inputs and outputs for NACA 0012 airfoil noise dataset.

Since all five input variables are categorical, their relationship revealed scattered clusters. The analysis becomes difficult since there are more than two dimensions involved.

Further, it shows that variables, viz., angle of attack, chord length, and free stream velocity, exhibit slight skewness with a longer right tail. Further, the dataset's frequency and boundary layer thickness have very high positive skewness, indicating that data are

not normally distributed. Hence, models like the RF or other ensemble methods such as GB must be used for predictive modeling that can handle skewed data reasonably well against other models like linear regression. RF and GB are popular ML techniques for handling high-dimensional datasets. Also, their interpretability and transparency with feature importance analysis make them a better choice than artificial neural networks or other ML algorithms.

2.3. Model Description

- GB Regressor:

GB is a robust boosting algorithm that utilizes multiple weak learners to create a strong learner. It works iteratively, wherein each new model is trained to minimize the cost function or error based on the predictions of the prior model. The main idea is to gradually improve the ensemble's predictive ability by addressing the mistakes made by the previous models. A weak learner is a model that performs marginally better than random guessing on a given task. The ensemble comprises M trees, as shown in Figure 3.

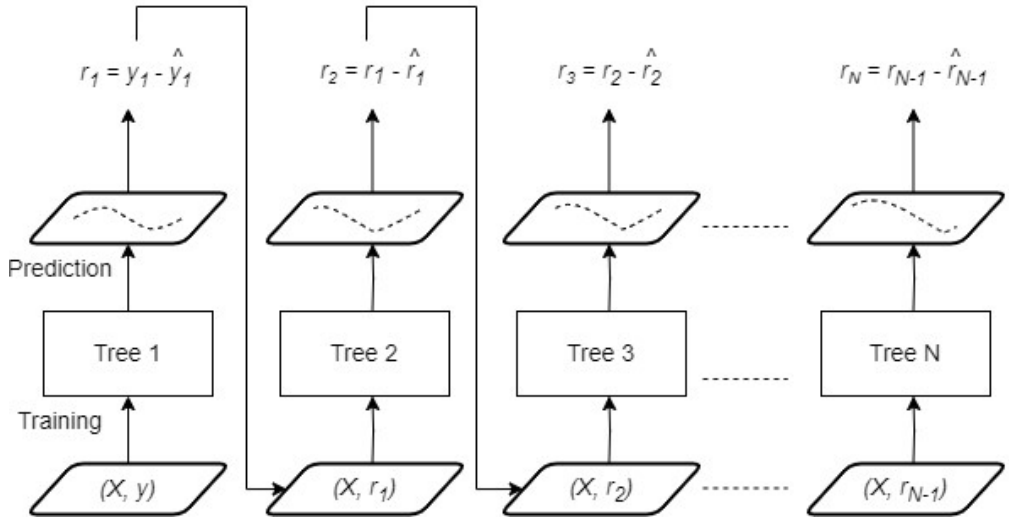


Figure 3. The mechanism behind the working of GB Regressor [20].

Tree 1 is tutored with the feature matrix X and the corresponding outputs Y. The prognosis, denoted as Y_1, is employed to compute the residual errors r_1 of the tutoring set. The final prediction is given in Equation (1).

$$Y(predicted) = Y_1 + (\eta \times r_1) + (\eta \times r_2) + \ldots\ldots + (\eta \times r_N), \quad (1)$$

where η is the learning rate.

- XGBoost Regressor:

XGBoost utilizes an objective function that consists of a loss function and a regularization term. The loss function quantifies the deviation of the model's results from the actual values. The regularization term helps control the model's complexity and prevents overfitting. XGBoost includes parallel processing techniques, allowing for faster computation, whereas, in traditional GB, each new model is trained sequentially based on the previous model's errors. XGBoost is an optimized form of GB that introduces advanced regularization techniques and efficient tree construction.

- Light Gradient Boost Regressor:

Light Gradient Boost is a gradient-boosting framework designed to enhance the model's efficiency and reduce memory usage. It differentiates from traditional GB and XGBoost through its leaf-wise tree growth strategy and memory efficiency. It focuses on

finding the best splits by focusing on leaves with the highest loss reduction, resulting in faster training and improved accuracy. Its memory efficiency is based on gradient-based one-side sampling (GOSS)-ranking the data instances based on their gradients (importance) and exclusive feature bundling (EFB) techniques- identifying groups of features that never occur together and bundling these features.

- CatBoost Regressor:

Catboost, also known as Categorical Boosting, is an open-source library created by Yandex, designed to tackle problems like regression and classification involving many independent features. It has a unique ability to handle categorical and numerical features without requiring feature encoding techniques, saving time and effort in preprocessing the data. Additionally, CatBoost automatically scales all the features internally to a reasonable range, unlike traditional boosting algorithms, which help in faster convergence and enhance the overall performance of the trained model.

- RF Regressor:

RF is an ensemble learning method suitable for regression and classification work. It combines the power of multiple decision trees and a technique known as Bootstrap and Aggregation, or bagging, as shown in Figure 4. It involves randomly sampling rows and features from the dataset to create sample datasets for each tree; this process is called Bootstrap. The Aggregation step combines the predictions of all the individual trees to produce the final output. While RF constructs multiple decision trees and averages their predictions, GB and XGBoost build models sequentially to rectify the mistakes of previous models. RF performs well with unseen data, is less prone to overfitting, and is computationally efficient. The final prediction is given in Equation (2).

$$Y(predicted) = \frac{1}{N} \sum_{i=1}^{N} Mi, \quad (2)$$

where Mi is the prediction of the decision tree and N is the total number of decision trees.

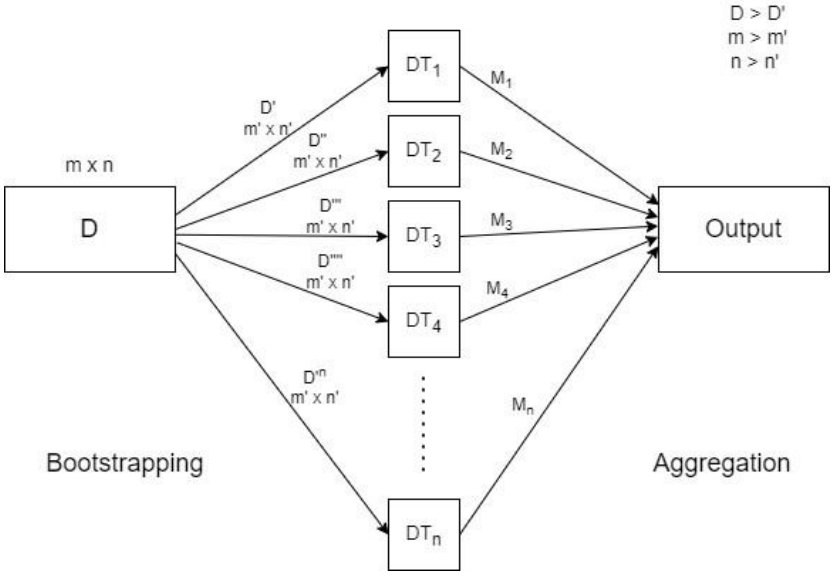


Figure 4. The working mechanism of RF Regressor [21].

- Extra Tress Regressor:

Extra Trees or Extremely Randomized Trees are also a type of ensemble learning technique that, like the RF, aggregates the outputs of various uncorrelated decision trees. Extra Trees can frequently outperform the RF algorithm. The main distinction between RF and the Extra Trees Regressor is that Extra Trees does not carry out Bootstrap Aggregation as

RF does. It constructs decision trees using the entire training dataset. Instead of considering all features and finding the best split point, the Extra Trees Regressor randomly selects a subset of features and a random split point. This extra randomness helps to reduce the variance and overfitting of the model. The Extra Trees Regressor can be helpful for high-dimensional datasets or when computational efficiency is a priority.

2.4. Model Training and Testing

Gradient Boosting Regressor, Random Forest Regressor, and Extra Trees Regressor are implemented in scikit-learn in Python. Scikit-learn, or sklearn, is a well-known open-source machine-learning library for Python that provides diverse tools and algorithms for various machine-learning models, including classification, regression, clustering preprocessing, etc. It is a comprehensive and powerful tool for ML in Python.

Present work involves the using XGBRegressor, LGBMRegressor, and CatBoostRegressor, which require a separate library designed explicitly for its functioning, such as XGBoost library, LightGBM library, and CatBoost library. These are extensively applied in various ML tasks and have proven to be a reliable choice.

During model training, 30% of the data were allocated to the testing set, while the remaining 70% were allocated for the activity. The model learns from the training data's features and target values to establish relationships. The model is trained and tested five times, with each of the five folds used as the testing set once, while the remaining four folds were used as the training set, respectively. A more reliable estimate of the model's effectiveness is obtained by averaging the performance metrics across all five folds.

3. Results

3.1. Performance Metrics

To compare GB and RF models, several performance metrics can be used. The commonly used metrics in regression tasks include Mean Squared Error (MSE), Coefficient of Determination (R^2) score, Mean Absolute Percentage Error (MAPE), Standard Deviation, and training time. MSE measures the mean of the squared discrepancies involving the predicted and actual values. A lower MSE denotes better model performance, whereas zero MSE indicates perfect fit. R^2 score or coefficient of determination measures the goodness of fit of a model and ranges from 0 to 1. A higher R^2 score implies a better fit, where 1 denotes a perfect fit.

MAPE calculates the mean of the percentage difference between the predicted and actual values, which is primarily useful when the scale of the data varies significantly. Training time refers to the time the machine learning model takes to train on the given dataset. It measures the computational resources needed to train a model. Standard deviation is a computation of the spread of a group of data points. In machine learning, it provides insights into the stability and consistency of the model's performance. Also, RF and GB models were used to extract feature importance.

3.2. Results Visualization

The results from Table 2 show that the Extra Trees or Extremely Randomized Trees algorithm exhibits superior performance with the highest Coefficient of Determination and lowest mean-squared error. In contrast, the GB Regressor shows an advantage in terms of the lowest standard deviation and training time for the given dataset. Light GBM and CatBoost Algorithm also achieved very high Coefficient of Determination, and CatBoost exhibited the most extended training duration.

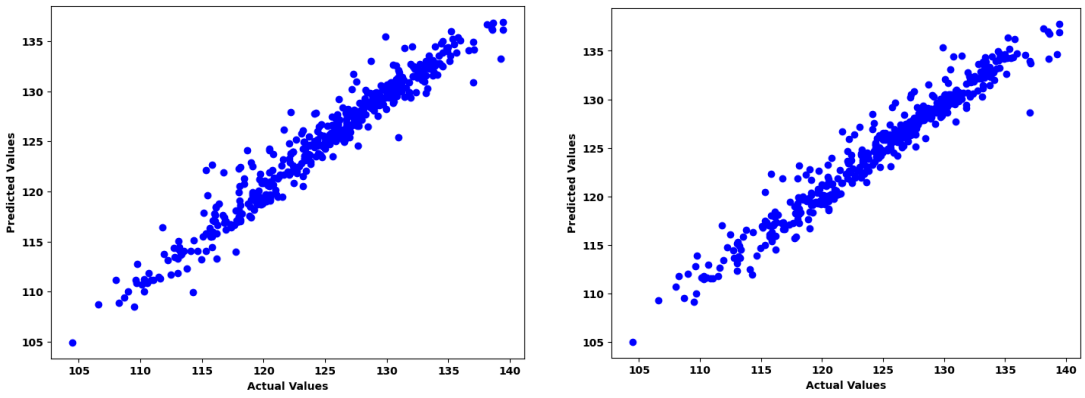
The Extra Trees algorithm is known for adding more randomness in constructing individual decision trees than RF. Also, due to this randomness, they perform better in noisy datasets by ignoring less informative features, which is the reason for its highest Coefficient of Determination. Gradient Boost is designed to build weak learners in a sequence by learning from its previous predictions, which helps in faster convergence.

Also, the depth of its decision trees is often smaller than that of a random forest, which imparts the shortest training time.

Table 2. Performance metrics obtained using Python version 3.10.9.

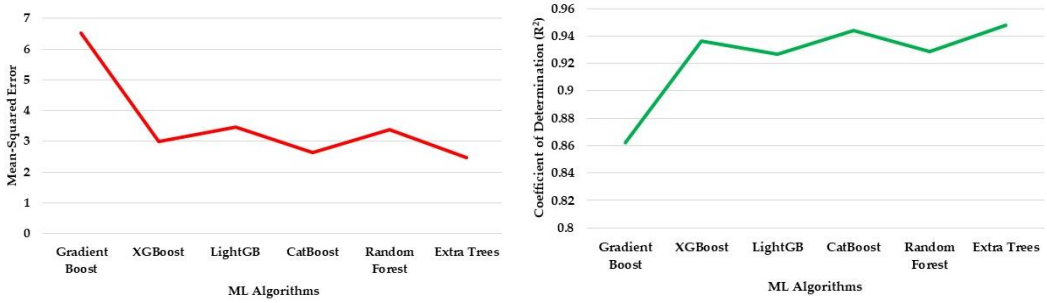
ML Model	MSE	R^2	MAPE	SD	Training Time (s)
Gradient Boost	6.5337	0.8620	0.0159	5.8407	0.1250
XGBoost	3.0026	0.9365	0.0095	6.4960	0.1258
LightGB	3.4792	0.9265	0.0109	6.4331	0.1366
CatBoost	2.6331	0.9443	0.0092	6.4633	1.5546
RF	3.3765	0.9287	0.0107	6.1658	0.5082
Extra Trees	2.4631	0.9479	0.0090	6.3133	0.3750

The visual demonstration of how well the model captures the underlying patterns in the data is shown in Figure 5a,b, wherein the predicted values can closely align with the actual values, forming a linearly fitted curve. However, in some cases, outliers may exist, which indicates the model's potential areas of improvement. Figure 6a,b show the graphically comparing mean-squared error and coefficient of determination for different machine learning algorithms.



(**a**) (**b**)

Figure 5. Predicted versus actual values for (**a**) CatBoost Regressor, (**b**) Extra Trees Regressor.



(**a**) (**b**)

Figure 6. Algorithm performance comparison using (**a**) Mean Squared Error, (**b**) Coefficient of Determination.

The relative importance of all five features is determined using both RF and GB models, yielding similar results as shown in Figure 7a,b. Frequency and boundary layer thickness are the topmost influential features, and the other three features, although individually less critical, contribute to the overall predictive performance.

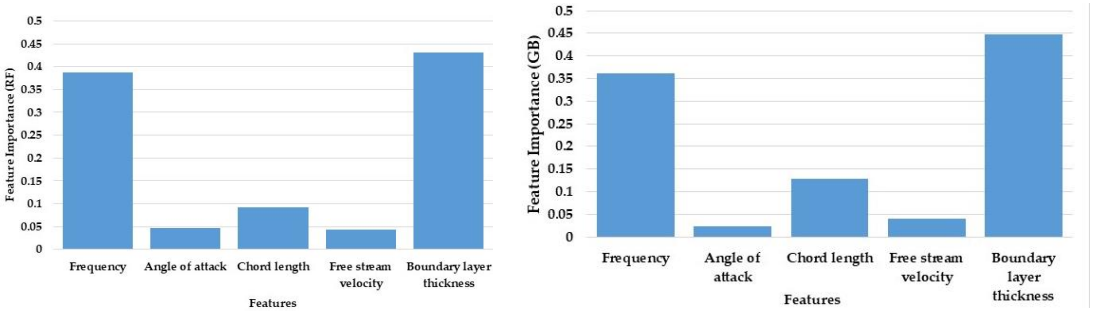


Figure 7. Feature importance using (**a**) RF Regressor (**b**) Gradient Boost Regressor.

A simpler model can be created by focusing only on influential features to help mitigate the risk of overfitting. Reduced feature data can lead to faster model training, which is especially beneficial when dealing with large datasets. However, expert domain knowledge is required to identify influential features accurately.

4. Discussion

The prediction and optimization of airfoil self-noise is significant in aeronautical engineering. Traditional methods of predicting airfoil noise rely on empirical or numerical models and experimental testing. However, the advancements in machine learning techniques have opened up new possibilities for accurate and effective prediction of airfoil self-noise.

The results show that Gradient Boost has a higher MSE when evaluated on the test data since it is more prone to overfitting. CatBoost outperforms other boosting algorithms regarding R^2 metric because it is specifically designed to handle categorical variables more effectively, and its advanced regularization techniques prevent overfitting. The Gradient Boost algorithm trains an ensemble of weak learners sequentially, with each weak learner trying to correct the errors made by the previous learners. This process allows GB to converge quickly and requires fewer iterations, reducing training time. Also, by averaging the predictions of multiple weak learners, GB minimizes the uncertainty in the projections, leading to a lower standard deviation.

It is essential to acknowledge that the findings presented in this study are subject to certain limitations and assumptions that may influence its generalizability. The performance of the models depends on the choice of features. Although we identified top features, we incorporated all of them for our analysis. In the future, omitting some less relevant features could improve accuracy.

Also, for this research, we have used 1503 data instances, out of which 70% were allocated to training, and all algorithms showed remarkable robustness with five-fold cross-validation. As we look towards the future, expanding the dataset with a broader representation of airfoil instances, multiple analyses with hyperparameter tuning can further enhance the reliability of our results.

5. Conclusions

The use of machine learning in the aerospace industry is rapidly evolving. Prediction of airfoil self-noise finds implications in improved design reduced development cost of wind tunnel testing, and it can help manufacturers comply with the noise regulations. However, a high quality and substantial amount of data is required to ensure consistency of the results. Training time for large datasets may depend on the model and computational power of the hardware, but we have to strike a balance between time and accuracy. This study found Extra Trees Regressor and CatBoost to be the best performing but at the cost of more training time. Gradient Boost with the least training time suffered at maintaining accuracy. XGBoost could achieve both high accuracy and less training time, mainly due to

its ability to parallel processing on multiple CPU cores and its regularization techniques to avoid overfitting.

Machine learning can use large amounts of data to identify complex patterns that may be difficult to capture by traditional methods. Future work could include exploring physics-based models that can work on the underlying mechanism of airfoil noise generation. Large amounts of diversified data on self-noise can be generated using extensive wind tunnel experiments or computational fluid dynamics. Hybrid models with further hyperparameter tuning can enhance the model's performance. Efforts can be made to develop real-time airfoil noise prediction systems using machine learning to facilitate the design of quieter and more efficient aircraft.

Author Contributions: Conceptualization, S.B.N. and G.S.V.; methodology, S.B.N.; software, S.B.N.; validation, S.B.N.; formal analysis, R.C.K. and G.S.V.; resources, R.C.K. and G.S.V.; writing—original draft preparation, S.B.N.; writing—review and editing, R.C.K. and G.S.V.; visualization, G.S.V.; supervision, R.C.K.; project administration, R.C.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data analyzed in this research work is accessible from Kaggle at "https://www.kaggle.com/datasets/fedesoriano/airfoil-selfnoise-dataset (accessed on 25 June 2023)". Kaggle is a well-known platform with a wide variety of datasets and is publicly accessible. We appreciate Kaggle and the original dataset provider (NASA) for sharing this valuable information and significantly advancing our knowledge.

Acknowledgments: The authors would like to sincerely thank NASA for their time, effort, and expertise in conducting wind tunnel experiments for various airfoil dimensions and sharing the collected dataset that paved the way for our research work. We would also like to thank Kaggle for providing the platform for data enthusiasts and their commitment to open data access, without which this project would not have been feasible.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Patri, A.; Patnaik, Y. Random forest and stochastic gradient tree boosting based approach for the prediction of airfoil self-noise. In Proceedings of the International Conference on Information and Communication Technologies (ICICT), Kochi, India, 3–5 December 2014.
2. Brooks, T.F.; Stuart, D.; Marcolini, M.A. Airfoil Self-Noise, and Prediction. In *NASA Reference Publication 1218*; NASA: Hampton, VA, USA, 1989.
3. Moreau, S.; Henner, M.; Iaccarino, G.; Wang, M.; Roger, M. Analysis of Flow Conditions in Freejet Experiments for Studying Airfoil Self-Noise. *AIAA J.* **2003**, *41*, 1895–1905. [CrossRef]
4. Errasquin, L.A. Airfoil Self-Noise Prediction Using Neural Networks for Wind Turbines. Doctoral Dissertation, Virginia Tech. University, Blacksburg, VA, USA, 2009.
5. Sarradj, E.; Geyer, T. Symbolic regression modeling of noise generation at porous airfoils. *J. Sound Vib.* **2014**, *333*, 3189–3202. [CrossRef]
6. Sarradj, E.; Geyer, T. Airfoil noise analysis using symbolic regression. In Proceedings of the 19th AIAA/CEAS Aeroacoustics Conference, Berlin, Germany, 27–29 May 2013.
7. Lee, S.; Lee, S.; Ryi, J.; Choi, J.S. Design optimization of wind turbine blades for reduction of airfoil self-noise. *J. Mech. Sci. Technol.* **2013**, *27*, 413–420. [CrossRef]
8. Sathyadevan, S.; Chaitra, M.A. Airfoil self-noise prediction using linear regression approach. *Comput. Intell. Data Min.* **2014**, *2*, 551–561.
9. Marks, C.R.; Rumpfkeil, M.P.; Reich, G.W. Predictions of the effect of wing camber and thickness on airfoil self-noise. In Proceedings of the 20th AIAA/CEAS Aeroacoustics Conference, Atlanta, GA, USA, 16–20 June 2014.
10. Saab, J.Y.; de Mattos Pimenta, M. Displacement thickness evaluation for semi-empirical airfoil trailing-edge noise prediction model. *J. Braz. Soc. Mech. Sci. Eng.* **2016**, *38*, 385–394. [CrossRef]
11. Santhanam, R.; Uzir, N.; Raman, S.; Banerjee, S. Experimenting XGBoost algorithm for prediction and classification of different datasets. *Int. J. Control Theory Appl.* **2017**, *9*, 651–662.

12. Bentejac, C.; Csorgo, A.; Martínez-Munoz, G. A comparative analysis of gradient boosting algorithms. *Artif. Intell. Rev.* **2021**, *54*, 1937–1967. [CrossRef]
13. Chen, Z.; Li, B.; Han, B. Improve regression accuracy by using an attribute-weighted KNN approach. In Proceedings of the 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Guilin, China, 29–31 July 2017.
14. Tahmassebi, A.; Gandomi, A.; Meyer-Baese, A. A Pareto front-based evolutionary model for airfoil self-noise prediction. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018.
15. Pal, P.; Datta, R.; Rajbansi, D.; Segev, A. A neural net-based prediction of sound pressure level for the design of the aerofoil. In Proceedings of the Swarm, Evolutionary, and Memetic Computing and Fuzzy and Neural Computing: 7th International Conference, SEMCCO 2019, 5th International Conference, FANCCO 2019, Maribor, Slovenia, 10–12 July 2019.
16. Radha Krishnan, N.S.; Uppu, S.P. A novel approach for noise prediction using Neural network trained with an efficient optimization technique. *Int. J. Simul. Multidiscip. Des. Optim.* **2023**, *14*, 3. [CrossRef]
17. Rastgoo, A.; Khajavi, H. A novel study on forecasting the airfoil self-noise using a hybrid model based on the combination of CatBoost and arithmetic optimization algorithm. *Expert Syst. Appl.* **2023**, *229*, 120576. [CrossRef]
18. Naik, N.; Kowshik, S.; Bhat, R.; Bawa, M. Failure analysis of governor in diesel engine using Shainin System™. *Eng. Fail. Anal.* **2019**, *101*, 456–463. [CrossRef]
19. Jain, I.; Manikandan, J. Study and Evaluation of Machine Learning algorithms for Aerospace applications. In Proceedings of the IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES), Yogyakarta, Indonesia, 24–25 November 2022.
20. Gradient Boosting in ML, GeeksforGeeks. Available online: https://www.geeksforgeeks.org/ml-gradient-boosting/ (accessed on 10 July 2023).
21. Random Forest Regression in Python, GeeksforGeeks. Available online: https://www.geeksforgeeks.org/random-forest-regression-in-python/ (accessed on 10 July 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.