aerospace

MDPI

Article

Composite Fins Subsonic Flutter Prediction Based on Machine Learning

Mirko Dinulović, Aleksandar Benign and Boško Rašuo *

Aerospace Department, Faculty of Mechanical Engineering, University of Belgrade, 11120 Belgrade, Serbia; mdinulovic@mas.bg.ac.rs (M.D.); abengin@mas.bg.ac.rs (A.B.)
* Correspondence: brasuo@mas.bg.ac.rs

Abstract: In the present work, the potential application of machine learning techniques in the flutter prediction of composite materials missile fins is investigated. The flutter velocity data set required for different fin aerodynamic geometries and materials is generated using a hybrid data collection method: from the wind tunnel experiments at flows ranging from 5 to 30 m/s at Re = 300,000 to 500,000, whereas synthetic data is collected using modified NACA flutter boundary model. Once the flutter data are collected, different regression algorithms were investigated, and the results were compared in terms of accuracy (when compared to the experimentally obtained results and the numerical flutter models), training time (minimization), and R-squared values (maximization). The algorithms investigated and their performance analyzed are fast forest regression, SDCA regression (stochastic dual coordinate ascent), and the light GBM (light gradient boosting machine) regression algorithm that belongs to the gradient boosting regression algorithm family. It was found that the light GBM algorithm renders the most accurate results. Based on this research, it can be concluded that artificial intelligence (machine learning) techniques can be successfully deployed in the analysis of complex flutter phenomena.

Keywords: machine learning; regression; wind-tunnel; flutter; composite materials

Citation: Dinulović, M.; Benign, A.; Rašuo, B. Composite Fins Subsonic Flutter Prediction Based on Machine Learning. *Aerospace* **2024**, *11*, 26. https://doi.org/10.3390/aerospace11010026

Academic Editor: Daochun Li

Received: 13 November 2023
Revised: 18 December 2023
Accepted: 21 December 2023
Published: 27 December 2023

1. Introduction

The human desire to fly has been receiving interest from the beginning of time. From ancient legends to the present day, people have long longed to resist gravity and take off through the skies. From Icarus in Greek mythology to the flying carpets of Middle Eastern stories, these accounts reflect mankind's well-established yearning to rise above natural restrictions. As human evolution advanced, endeavors to imitate the trip of birds arose. Early creators and visionaries, like Leonardo da Vinci, planned flying machines and concentrated on the mechanics of flight. While their plans frequently stayed on paper, these innovators laid the preparation for future aeronautics leap forwards. The desire to rise to the skies pressed forward in the late eighteen hundreds when designers and pilots like the Wright brothers proved that controlled flight in a machine heavier than air was possible by performing the first controlled flight in December 1903. With their momentous accomplishments in controlled flight, they showed the way that people could take off like birds and open the huge capability of aeronautics. From that point forward, the longing to fly has become profoundly imbued in human culture. Air travel has changed the world, interfacing individuals across landmasses and societies. The capacity to load up a plane and navigate huge distances surprisingly fast has turned into a basic piece of our globalized society. Past pragmatic travel and the human longing to fly have likewise filled the improvement of sporting aeronautics and experience sports. Paragliding, skydiving, hang floating, and wingsuit flying are only a couple of instances of exercises that permit people to encounter the thrill and opportunity of flight firsthand. These pursuits catch the quintessence of human interest, the yearning to investigate new frontiers and push the limits of what is believed to be conceivable.

Since the first successful flight in 1903, aviation as an industry has been developing at one of the fastest rates. Furthermore, the peculiarities of flight have tracked down the use in military applications, like those of warplanes, missiles, drones, and UAVs.

However, flying, like any form of transportation, comes with inherent risks (even if it is generally considered to be a safe mode of transportation). Accordingly, designers, manufacturers, avionics controllers, organizations, pilots, and carriers are continually attempting to develop security conventions further and advance airplane innovation.

In this paper, and in the realm of fly-worthy structure design improvement, new techniques based on machine learning (ML) for phenomena known as flutter are calculated, and the results are compared to experimental wind tunnel data and discussed. More explicitly, in this work, AI (ML) procedures are utilized to predict the flutter speeds (loss of stability) of missile composite materials fins. Lifting surface or control surface flutter refers to the self-excited oscillations or vibrations that can occur due to the interaction of aerodynamic, inertial, and elastic forces. It potentially can lead to loss of stability and hence control, or in certain situations, to catastrophic structural failure and loss of flying vehicle. Even though many computational techniques have been developed over time since the discovery of the flutter, the problem still persists. To emphasize this fact, a brief historical overview of crashes due to flutter is presented: As per a few verifiable pieces of information, the absolute first plane accident occurred in 1903, even before the authentic flight of the Wright brothers. On 17 October 1903, Samuel Langley, an American researcher, encountered flutter on his airplane, known as the "Langley Aerodrome". The occasion, frequently alluded to as the "Langley first plane accident", occurred on the Potomac Stream close to Washington, D.C. As per some computer simulations performed at a later date involving present-day strategies for flutter estimate, loss of stability could have been one of the reasons for the accident of Langley's plane (Figure 1).

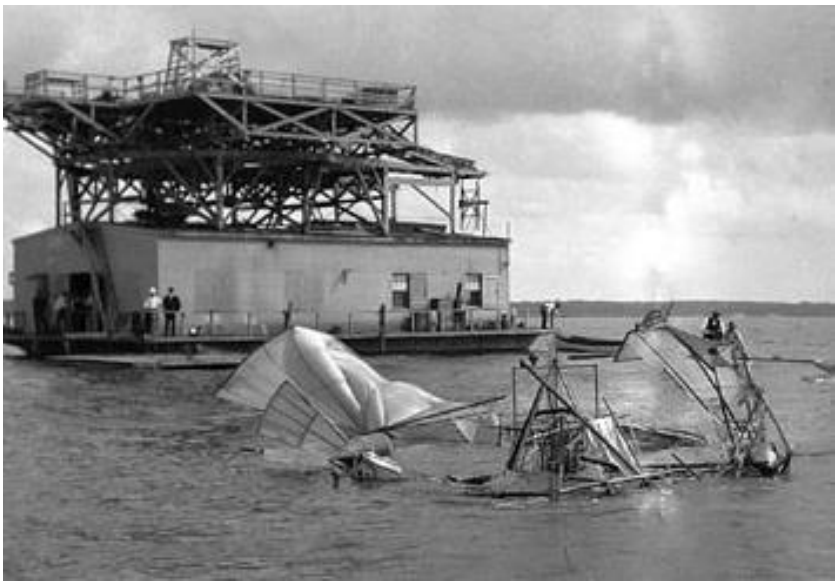


Figure 1. Langley plane "Aerodrome" crash 1903 (http://www.unmuseum.org/flyers1.htm, accessed on 22 Decemeber 2023).

Almost one century after the Langley plane crash, the F-117 Nighthawk, a prestigious military airplane used by the US, experienced an elevon malfunction that led to a crash. The airplane in question, known for its precise shape and radar-sidestepping capacities, utilized an exceptional flight control framework that used elevons—experienced elevon flutter at particular flight conditions (Figure 2). These elevons were basic in-flight controls, rendering the airplane able to fly and play its main role successfully. This episode highlighted the basic significance of a thorough understanding of flutter and the further development of flutter prediction techniques.

Figure 2. F-117 crash due to elevon flutter (https://crewdaily.com/well-known-jet-ejections-of-the-aviation-history-part-1/, accessed on 22 Decemeber 2023).

Unfortunately, many other failures due to flutter have been reported throughout aviation history. An exhaustive survey of this issue, according to verifiable resources, is given in a paper by I.E Garrick and Wilmer H Reed the III, named "Historical Development of Flutter" (1981).

It can be concluded, from this introduction, that in order to prevent flutter, further research is required, coupled with the improvement of computation and computing methods. In the present method, using the ML approach, the flutter speed of composite materials missile fins is presented. The results obtained using this approach are compared to the experimental ("wind tunnel testing").

The motivation for this research is driven by the fact that the flutter phenomenon represents a potentially disastrous occurrence that may lead to the loss of the flying vehicle. Present flutter speed prediction methods are under constant development and improvement in order to achieve better efficiency and accuracy. The application of ML techniques in flutter speed prediction is investigated in this paper. In recent years, the flutter analysis, using different artificial intelligence approaches, has been the focus of many researchers [1–4].

2. ML Flutter Model Development Methodology

ML is revolutionizing many fields, and one of the areas in which it is making a significant impact is aerospace engineering, specifically in the analysis of composite materials fin flutter. Composite materials fin flutter is a complex phenomenon that can lead to catastrophic failure in missiles or control loss, and it is essential to understand and mitigate this risk. ML offers powerful tools to enhance our understanding and prediction of fin flutter, ultimately leading to safer and more efficient aircraft design [5,6].

Different algorithms can process vast amounts of data from various sources, including wind tunnel tests, flight data, and computational simulations. By analyzing this data, ML models can identify patterns and correlations that humans might miss. These insights can help engineers better understand the conditions that lead to wing flutter and develop more effective mitigation strategies.

ML can automatically extract relevant features from raw data, such as airspeed, altitude, and wing shape parameters. This feature extraction process can reveal critical factors contributing to flutter and improve the accuracy of predictive models.

ML algorithms can be used to build predictive models that estimate the likelihood of wing flutter under different conditions. These models can consider multiple variables and provide early warnings or recommendations for avoiding flutter, enabling real-time decision-making during flight.

ML can be employed to identify abnormal behavior in an aircraft's wing dynamics. By continuously monitoring the aircraft's response to various conditions, ML models can detect subtle changes that might indicate the onset of flutter. This early warning system can be crucial for pilot safety and aircraft maintenance.

ML algorithms can assist in the optimization of wing designs to minimize the risk of flutter. Engineers can use ML-based optimization techniques to find the best combination of wing parameters that enhance flutter resistance while maintaining aerodynamic performance [7].

Traditional methods of analyzing wing flutter often involve complex and computationally expensive simulations. Machine learning can offer more efficient ways to approximate flutter behavior, reducing the computational cost and accelerating the design process [8,9].

ML can integrate data from multiple sources, including physical testing, simulation, and historical flight data. This holistic approach enables a more comprehensive understanding of wing flutter, leading to a more robust analysis.

ML models can continually learn and adapt to new data, making them capable of improving over time as more information becomes available. This adaptability is particularly valuable in the aerospace industry, where conditions and technologies are constantly evolving.

Based on this, ML has the potential to revolutionize the field of wing flutter analysis in aerospace engineering. By leveraging the power of data-driven insights, predictive modeling, and optimization, engineers can better understand, predict, and mitigate the risk of wing flutter. As a result, aircraft can be designed with improved safety, performance, and efficiency.

In this research, with the objective of developing a subsonic flutter model for composite materials missile fins based on ML, the following key steps are deployed:

1. Problem definition:
   The first step involves clearly articulating the problem that needs to be solved using ML. Understanding what needs to be predicted or classified is crucial.
2. Data gathering and data preparation:
   Data collection is undertaken to acquire the necessary dataset for training and testing the model. It is important that the data is representative of the problem at hand. The data is cleaned and preprocessed to eliminate inconsistencies and noise.
3. Exploratory data analysis (EDA):
   This step entails analyzing the data to comprehend its characteristics, relationships, and patterns. It aids in identifying correlations and making informed decisions regarding feature selection and engineering.
4. Conducting feature engineering:
   Features are chosen or generated to train the model. This process involves transforming or extracting relevant information from the raw data.
5. Splitting data into training and testing sets:
   The data is divided into two segments: one for training the model and the other for evaluating its performance. The training set imparts knowledge to the model, while the testing set is employed to assess its generalization.
6. Selecting a model:
   The appropriate ML algorithm is chosen based on the nature of the problem (classification, regression, etc.), dataset size, and other relevant factors.
7. Training the model:
   The model is educated using the training data to make predictions or classifications. It learns to recognize patterns and relationships within the data.
8. Evaluating the model:
   The testing set is used to gauge the model's performance. Common metrics include accuracy, precision, recall, F1-score (for classification), mean absolute error (MAE), mean squared error (MSE) (for regression), etc.

9. Tuning hyperparameters:
   Hyperparameters of the model are adjusted to optimize its performance. These parameters govern aspects like the learning rate, regularization strength, depth of a decision tree, etc.
10. Implementing cross-validation:
    Techniques such as k-fold cross-validation are applied to assess the model's performance robustly.
11. Deploying the model:
    Once content with the model's performance, it is deployed to a production environment where it can make predictions on new, unseen data.
12. Monitoring and maintenance:
    Regular monitoring of the model's real-world performance is essential. The model may need to be retrained or updated to maintain its accuracy.
13. Interpreting and explaining predictions (optional):
    Depending on the context, understanding how the model arrives at its predictions may be important. Techniques like SHAP values or LIME can be used for interpretation.
14. Iterating and improving:
    ML often involves an iterative process. As new data becomes available or as the problem evolves, the model may need to be reevaluated and retrained.

It is important to note that these steps are not always strictly sequential, and there may be some back-and-forth between them. Additionally, the specifics of each step can vary depending on the complexity of the problem, the type of data, and the choice of algorithms.

3. Data Collection Methods

Data collection is a pivotal step in the ML development process. It begins with a clear definition of the type of data required for the specific machine learning task, including the identification of relevant features or attributes. Next, data sources need to be identified, ranging from databases and APIs to web scraping, surveys, sensors, and logs. Obtaining the necessary permissions or rights to access and use the data is crucial, especially if others own it. Depending on the nature of the problem, data can be collected through various methods such as random sampling, stratified sampling, systematic sampling, or snowball sampling.

Ensuring data quality is paramount. This involves a thorough check for completeness, accuracy, consistency, and relevance. Any errors or inconsistencies are rectified through data-cleaning techniques.

For dynamic datasets, implementing version control is recommended. Depending on the ML task, some preprocessing steps may be applied at this stage, including feature scaling, normalization, and one-hot encoding. In cases where more data is needed, data augmentation techniques can be applied to generate additional training samples. The dataset is then divided into training, validation, and testing sets to evaluate the model's performance.

In the present work, data required for the ML model are collected using a hybrid approach. Data collection can employ a hybrid approach, combining real-world experimental data with synthetic data generated from mathematical models. This approach is valuable when acquiring large amounts of real data is challenging, costly, or time-consuming. Real-world experimental data is gathered through controlled experiments, observations, or measurements, providing insights into actual conditions or phenomena in specific contexts. However, this data may be limited in quantity or diversity. Complementing the experimental data, mathematical models are employed to simulate and generate additional data. These models rely on mathematical equations and algorithms that mimic the behavior of the system or phenomenon under study. This synthetic data significantly augments the dataset [5].

The benefits of this hybrid approach are substantial. Firstly, it increases the quantity of data beyond what can be achieved through experimental collection alone. Additionally, it allows for the simulation of a wide array of scenarios and conditions, offering a more comprehensive understanding of the system. Moreover, it reduces costs associated with

data collection, especially in cases involving expensive equipment or logistical challenges. The approach also provides control over input parameters, enabling the exploration of conditions that may not be feasible in actual experiments. Lastly, synthetic data can be used to test and validate ML models, algorithms, or analytical techniques prior to application on real-world data. However, it is imperative to consider a few key points. The accuracy of the mathematical model is paramount; it must faithfully represent the underlying system. Additionally, it is crucial to be aware of the assumptions and simplifications that underlie mathematical models. Finally, the integration of real and synthetic data requires careful attention to ensure that the combined dataset accurately reflects the system under study.

In summary, a hybrid approach that combines experimental data with synthetic data generated from mathematical models can be a powerful strategy for enhancing the quantity, diversity, and comprehensiveness of a dataset for ML or research purposes.

4. Materials and Methods

4.1. Experimental Data

In the present work, the experimental data were generated from wind tunnel testing on previously manufactured composite materials fin samples of different geometries and composite materials.

The experimental setup with DAQ (data acquisition) is presented in the following figures (Figures 3 and 4):

Figure 3. Wind tunnel test section (**top**) and mounted composite materials fin (**bottom**).

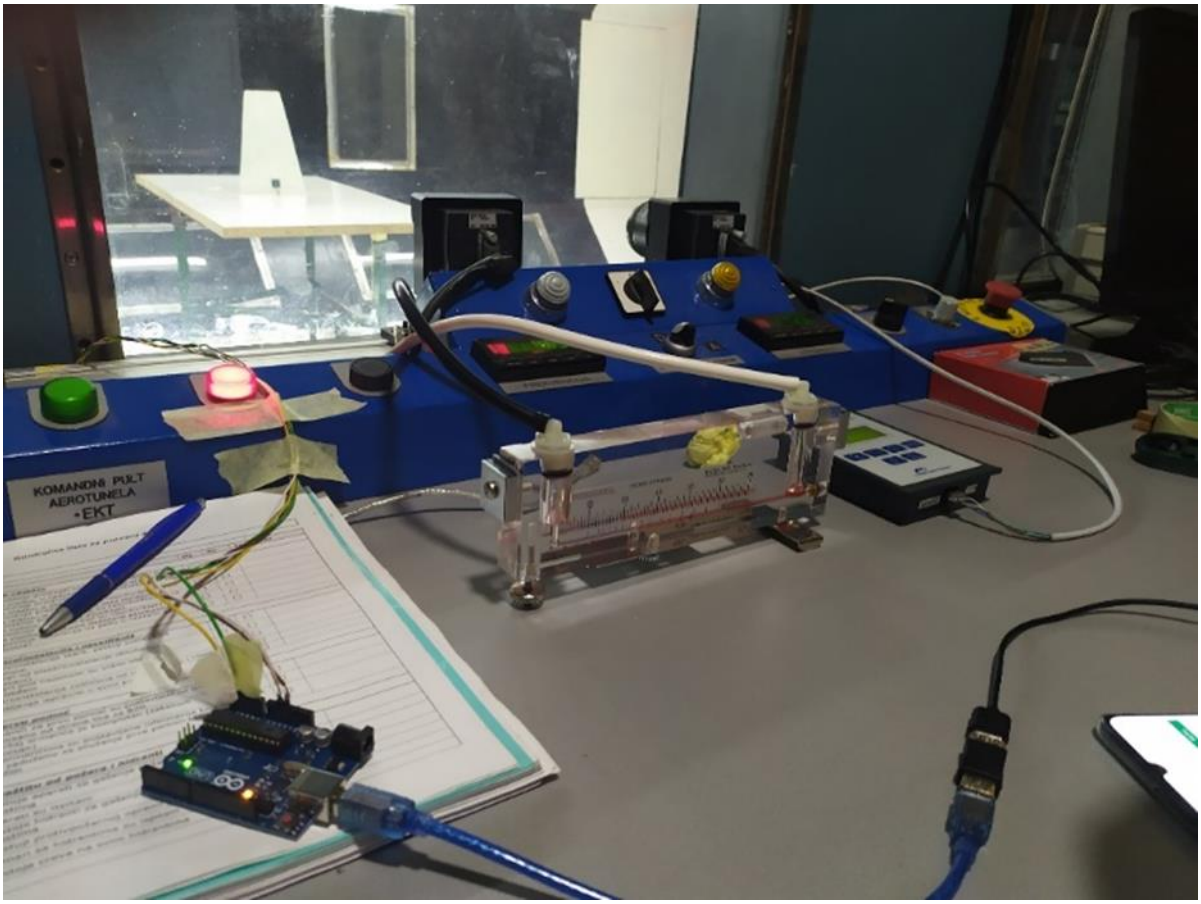


Figure 4. Wind tunnel DAQ.

A specially designed support structure is used to support the test samples and is placed in the test section of the tunnel. Test samples are presented in Figure 5 The support structure with a clamped sample at the root chord, positioned in the test section of the wind tunnel, is presented in Figure 3 In order to monitor the magnitude of the amplitudes during an oscillation cycle, accelerometers connected to the DAQ system were mounted on the test plate at the location of the root chord. The PCE PFM2 micro manometer with a pitot tube is used to determine the airflow velocity. The system is capable of increasing or decreasing airflow velocities in the wind tunnel by 0.5 m/s increments, with acceptable airflow stabilization times. To ensure the testing repeatability of the tests, a total of five samples were manufactured using the same material (same material supplier) and exposed to the same airflow conditions. Since low Mach number flows are of interest in this research, a subsonic closed return type wind tunnel, capable of generating axial flows of up to 30 m/s, is used. The DAQ system is capable of registering axial flow speeds in the wind tunnel along with frequencies of sample oscillations throughout the entire duration of the test (per run). The system registers all the acquired data (axial speed vs. frequency of oscillations). When a sudden increase in frequencies is observed for a corresponding speed, this particular boundary is assumed to be the flutter boundary for the sample tested.

The sample parameters were determined based on past experiences in flutter analysis, literature overview, and design of experiments (DOE) based on the Taguchi method. In the DOE, the process begins with clearly defining the experiment's objective and identifying critical factors. These factors, both controllable and uncontrollable, are tested at predetermined levels using orthogonal arrays, which systematically organize combinations of factor levels [10].

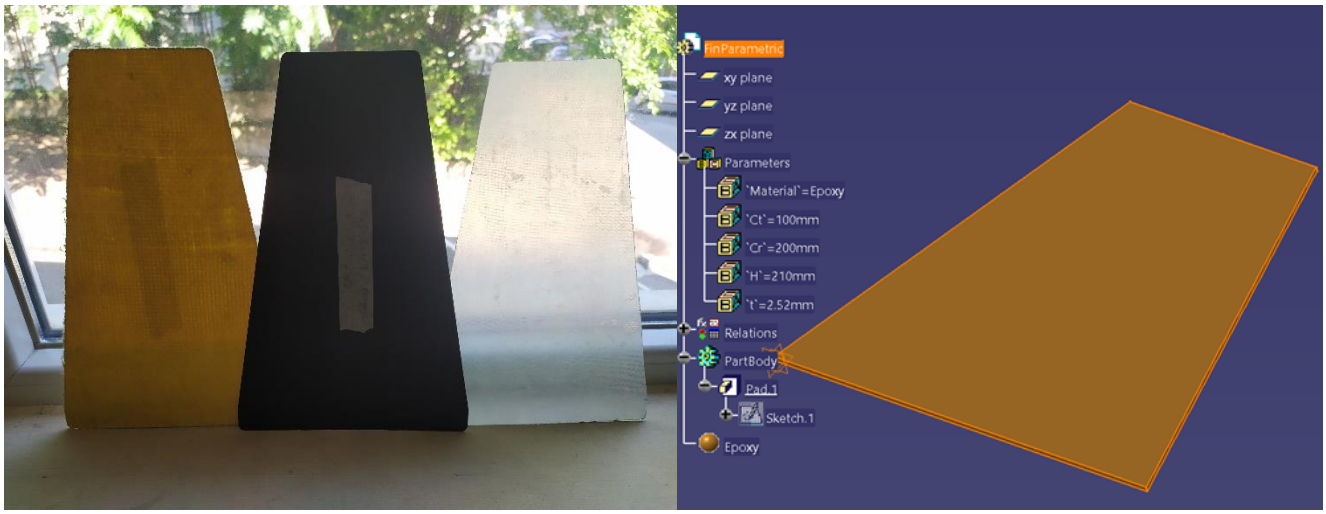


Figure 5. Composite materials fin(s) test samples (E-Glass/Epoxy, Kevlar/Epoxy, Carbon T-300/Epoxy).

In the particular research, the factors that are initially considered, with their respective levels and values, are summarized in Table 1.

Table 1. DOE factors and levels.

Factor	Level	
	H	**L**
Material	6.5 GPa	4.7 GPa
Thickness	1 mm	2.5 mm
Aspect ratio = b^2/S	2.5 -	1 -
Taper ratio = C_T/C_R	1 -	0.5 -

The orthogonal array used in this analysis is presented in the following table (Table 2).

Table 2. The DOE orthogonal array L9.

Run	Thickness	Material	Taper	Aspect
1	H	L	L	L
2	L	H	L	L
3	H	H	H	L
4	L	L	H	L
5	H	L	H	H
6	L	H	H	H
7	H	H	L	H
8	L	L	L	H
9	H	L	L	H

As can be seen, a total of nine experimental runs are required to perform the DOE analysis using these factors (with their H and L levels) by means of an L9 orthogonal array. To obtain the flutter speeds, and in accordance with the DOE setup, computational flutter analysis is deployed using the commercial NASTRAN software. A total of 9 models were constructed, and flutter analysis was performed.

Quality is assessed using signal-to-noise ratios (SN Ratios), aiming to maximize robustness. The experiment's runs are recorded, and SN Ratios are calculated for each

combination. By analyzing these ratios, the optimal settings that minimize sensitivity to variation are identified. In summary, the recognized geometrical parameters are fin thickness (t), fin root chord (C_R), fin tip chord (C_T), and fin span (H) (Figure 5). On the material side, material types (fibers and matrix) and their respective volume fractions in composite materials were examined. The results obtained were in very good correlation with the results reported in the literature [11].

Furthermore, a parametric CAD model was developed (Figure 6), which enabled faster mold production for different fin geometries and materials for producing the samples used. The samples were produced using the wet lay-up technique [12,13].

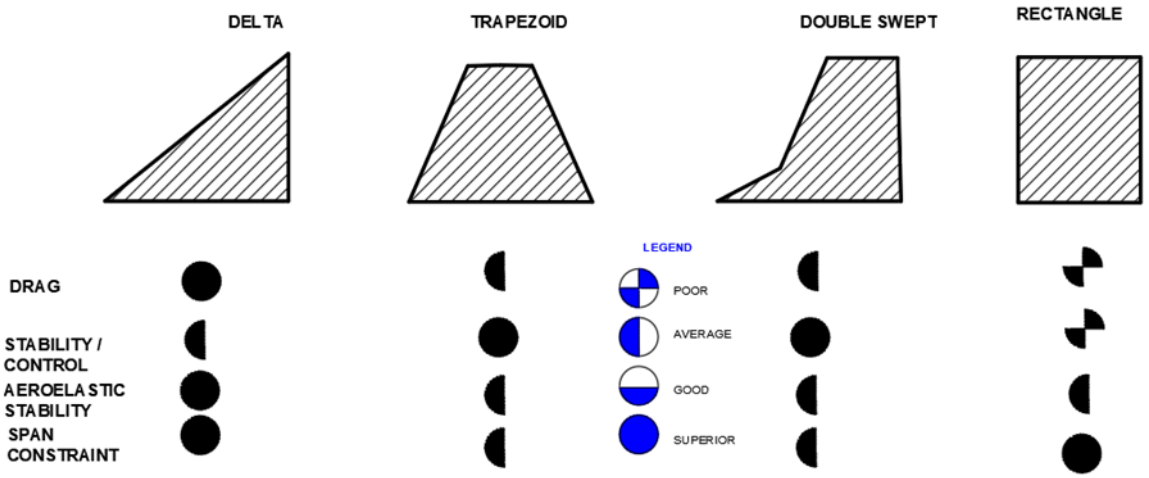


Figure 6. Typical missile fins geometries characteristics [12,13].

Using this approach, a portion of the required data was collected. Even though the experimental data were considered the most reliable and accurate, it should be noted that experiments in the wind tunnel are expensive and require long setup times and manpower. Furthermore, sample production (composite materials of different phases, fibers, and matrix, with different volume fractions and different geometries) also requires long manufacturing times and economic resources.

4.2. Synthetic Data

Bearing in mind that wind tunnel tests are expensive and time-consuming, a part of the synthetic data was required. Several flutter mathematical models were investigated for accuracy and calculation performance to create an enhanced dataset. Synthetic data generally refers to artificially generated data that mimics real-world data but is not obtained from actual observations or measurements. It is created using algorithms, mathematical models, or other computational techniques. Synthetic data is designed to have statistical properties, distributions, and relationships similar to real data, making it useful for various applications, particularly in situations where access to real data is limited or restricted [14–21].

4.3. CFA Model

The first model analyzed in this part of the research for synthetic data creation was the computational flutter analysis (CFA) approach. Computational flutter analysis is a sophisticated engineering technique used to predict and analyze the dynamic stability of structures, particularly in aerospace engineering. The CFA approach requires the design of fin aerodynamic, structural, and spline models. This model renders the most accurate results (when compared to experiments). For the fin(s) analyzed in this research, the CFA model is presented in the following picture (Figure 7):

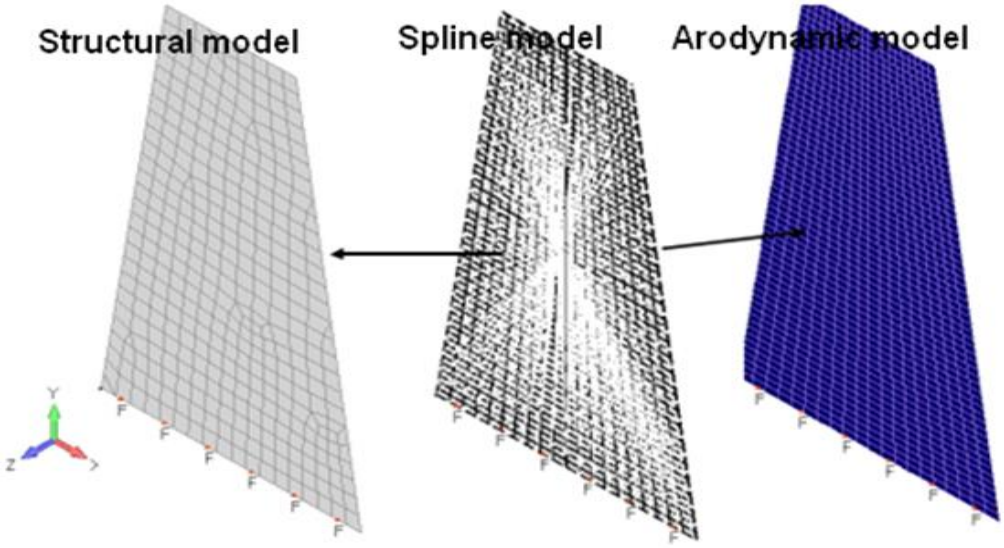


Figure 7. The CFA model schema (aerodynamic, spline, and structural flutter model of generic composite materials fin).

To discretize the aerodynamic model, aeropanel elements are used (often referred to as aerodynamic panels). These elements are typically used to define the aerodynamic properties of a structural model. In the present model, the aerodynamic theory used in aeropanel analysis within a finite element analysis (FEA) is based on the boundary element method (BEM): This method divides the aerodynamic surfaces into small panels and uses the source and doublet distributions on these panels to model the flow. BEM can handle both subsonic and supersonic flow and can be used for a wide range of aerodynamic analyses. Spline elements are used to connect the structural and aerodynamic models. Specifically, these spline elements are employed to define the coupling between the structural degrees of freedom (DOFs) and aerodynamic degrees of freedom. This enables the simulation of the interaction between the structural and aerodynamic responses in a coupled aeroelastic analysis. The spline element connects grid points (structural points) with aerodynamic points and allows for the transfer of displacements and rotations between them. This facilitates the exchange of information between the structural and aerodynamic models during the analysis. The element allows for the transfer of translational and rotational displacements between the structural and aerodynamic grids.

The structural model in the CFA model uses plate finite elements based on the Kirchhoff–Love theory. This theory assumes that the plate is thin enough such that transverse shear deformations are negligible. It is based on the Kirchhoff–Love hypothesis, which neglects the transverse shear strains and assumes that the normal to the mid-surface remains normal after deformation.

The complete CFA model is presented in Figure 7.

4.4. Binary Flutter Model (Typical Section)

In this analysis, the known binary flutter model is investigated for potential use in synthetic dataset creation. The binary model, based on the "typical section", is presented in the following picture (Figure 8):

This model represents the 2DOF model in twisting and bending, and based on the energy principles, the airfoil (typical section) equations of motion can be obtained in the following form:

$$\begin{aligned} \ddot{h} + \frac{S_k}{M_k}\ddot{\alpha} + g_h\frac{\omega_h^2}{\omega}\dot{h} + \omega_h^2 h &= -\frac{R_z(t)}{M_k} \\ \ddot{\alpha} + \frac{S_k}{I_k}\ddot{h} + g_\alpha\frac{\omega_\alpha^2}{\omega}\dot{\alpha} + \omega_\alpha^2\alpha &= \frac{M_T(t)}{I_k} \end{aligned} \quad (1)$$

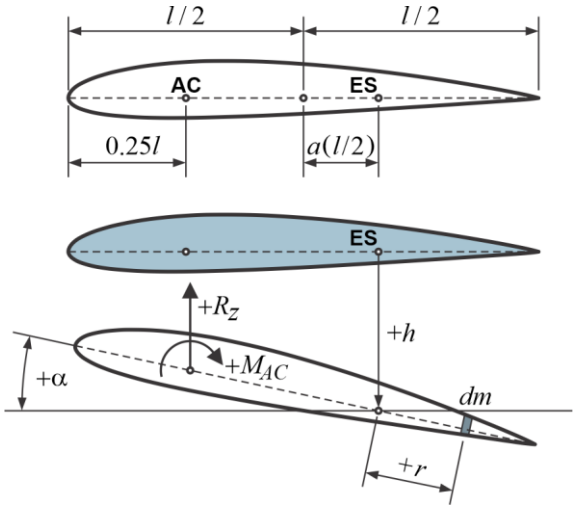


Figure 8. Typical section, binary flutter model.

In the application of this model for flutter analysis, the choice of adequate aerodynamic theory is of great importance. Using steady aerodynamics for lifting force and aerodynamic moment (Figure 9), the previous set of equations of motion (Equation (1)) can be solved and yield a closed-form solution.

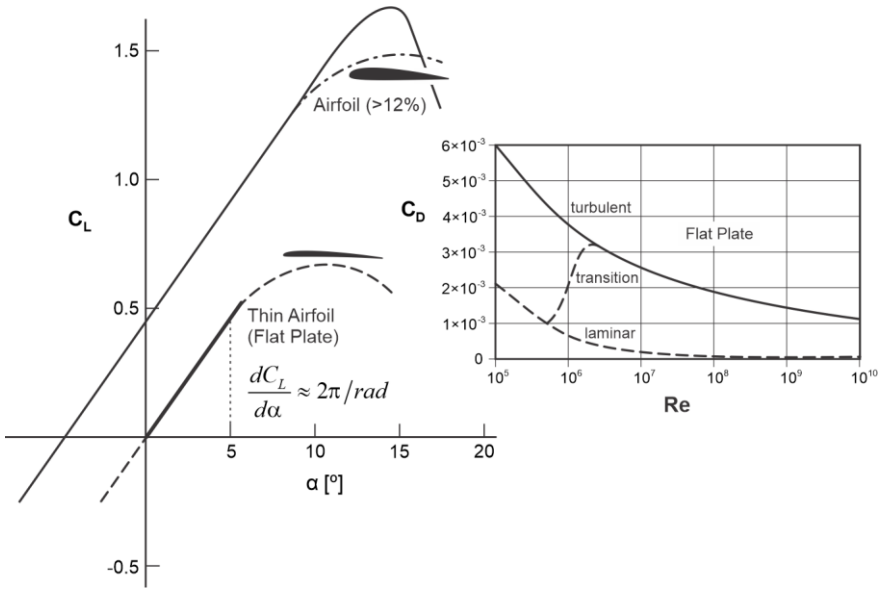


Figure 9. Thin airfoil lift coefficient.

This approach renders the least accurate results.

$$R_z(t) = -\frac{1}{2} C_L \cdot \alpha \cdot \rho \cdot V_0^2 \cdot l \tag{2}$$

In the previous equation, C_L is the lift coefficient assumed to be the function of the airfoil shape and the angle of attack α. Using the Prandtl–Glauert correction (Equation (3)) for the lift coefficient, in the form where A is the fin aspect ratio,

$$C_L = \frac{2\pi \cdot A}{2 + \sqrt{A^2 + 4 - \frac{A^2 \cdot V_0{}^2}{a_0{}^2}}}, \tag{3}$$

the quasi-steady flutter speed can be expressed as (Equation (4)):

$$V_F = \sqrt{\frac{\pi l^2 r_\alpha^2 \mu \left(\omega_\alpha^2 - \omega_h^2\right)}{8 C_{L\alpha} x_a}} \tag{4}$$

In Relations (4), ω_h and ω_α are the bending and twisting natural frequencies of the fin, r_α is the radius of gyration about the mid-chord, and μ is the dimensionless plate airstream mass ratio.

The unsteady aerodynamics approach is more accurate but requires numerical solvers. The Theodorsen unsteady model for aerodynamic force and moment is used to investigate the effectiveness of this model in the synthetic flutter data set generation. The Theodorsen relation for the unsteady aerodynamic force and moment reads (Equation (5)):

$$\begin{aligned} R_z(t) &= \pi\rho\tfrac{l}{2}V_0{}^2\left\{\tfrac{l}{2V_0{}^2}\ddot{h} + \tfrac{2}{V_0}C(k)\dot{h} - \tfrac{l^2 a_0}{4V_0{}^2}\ddot{\alpha} + \left[1 + (1-2a_0)C(k)\right]\tfrac{l}{2V_0}\dot{\alpha} + 2C(k)\alpha\right\} \\ M_T(t) &= -\pi\rho\tfrac{l^2 V_0{}^2}{4}\left\{-\tfrac{a_0 l}{2V_0{}^2}\ddot{h} - (1+2a_0)\tfrac{1}{V_0}C(k)\dot{h} + \left(\tfrac{1}{8} + a_0^2\right)\tfrac{l^2}{4V_0{}^2}\ddot{\alpha}\right\} \\ &\quad -\pi\rho\tfrac{l^2 V_0{}^2}{4}\left\{-\left[a_0 - \tfrac{1}{2} + 2\left(\tfrac{1}{4} - a_0^2\right)C(k)\right]\tfrac{l}{2C(k)}\dot{\alpha} - (1+2a_0)C(k)\alpha\right\} \end{aligned} \tag{5}$$

The good approximation of the circulation function $C(k)$ (with an error less than 4%) can be expressed as (Equation (6)):

$$C(k) = \frac{(1 + i10.6k)\cdot(1 + i1.774k)}{(1 + i13.51k)\cdot(1 + i2.745k)} \tag{6}$$

where k is reduced frequency, which can be expressed as a function of the frequency of vibration in the following form (Equation (7)):

$$k = \frac{\omega \cdot l}{2 \cdot V_0} \tag{7}$$

4.5. NACA Boundary Flutter Model

National Advisory Committee for Aeronautics, in technical note No 4197, has defined the flutter boundary velocity for the preliminary design of thin plates in the following form:

$$\left(\frac{V_F}{a_0}\right)^2 = \frac{G_E}{\frac{39.3\cdot A^3}{\left(\frac{t}{l}\right)^3\cdot(A+2)}\left(\frac{\lambda+1}{2}\right)\cdot\left(\frac{p}{p_0}\right)} \tag{8}$$

In the previous equation (Equation (8)), V_F represents flutter boundary speed, a_0 is the speed of sound, G_E is the equivalent shear modulus of elasticity, t is the lifting surface average thickness, p/p_0 is the ratio of the fluid pressure to standard pressure, and A and λ are aspect and taper ratio, respectively.

In this research, the composite materials (fin materials building blocks) are of prime interest. The previous equation (NACA flutter boundary equation) is derived for the isotropic materials, where the elastic coefficient G_E corresponds to the fin's shear modulus of elasticity. Hence, to verify the potential use of this model for synthetic data generation, a variation of the original equation is used to adopt the same approach for orthotropic structures (composite materials).

Furthermore, symmetric and quasi-isotropic laminates are of prime interest since, using this stack-up, unfavorable twisting-bending coupling is avoided in these types of laminates. This type of stack-up sequence is often used in composite materials missile fin design. The quasi-isotropic (QI) laminates satisfy the following relation (Equation (9)), (in

terms of the number of different ply orientations in the laminate, denoted with "m," and the number sequence repetitions "n") [22–24].

$$\left[0^0/\frac{180^0}{m}\ldots(m-1)\frac{180^0}{m}\right]_n \tag{9}$$

Laminate analysis is well established by means of the classical lamination theory (CLT) or first shear deformation theory. The first-order shear deformation theory (FSDT) is a theory used in the analysis of laminated composite materials. It is an extension of the classical laminate theory (CLT) that incorporates the effects of transverse shear deformation. The CLT theory is based on Kirchoff's hypothesis of thin plates bending with integration over separate layers leading to a well-known ABD matrix which relates forces and moments per unit width N and M, respectively, to the central midplane curvatures and strains (ε^0, κ) [14]. The ABD matrix is a matrix for a generic laminate and is usually presented in the following form (Equation (10)):

$$\begin{bmatrix} N \\ M \end{bmatrix} = \begin{bmatrix} A & B \\ B & D \end{bmatrix} \begin{bmatrix} \varepsilon^0 \\ \kappa \end{bmatrix} \tag{10}$$

The A, B, and D coefficients in the ABD matrix can be computed based on the following (Equation (11)):

$$\begin{aligned} A_{i,j} &= \sum_{\kappa=1}^{n} Q_{i,j}^{k}(h_k - h_{k-1}) \\ B_{i,j} &= \tfrac{1}{2}\sum_{\kappa=1}^{n} Q_{i,j}^{k}\left(h_k^2 - h_{k-1}^2\right) \\ D_{i,j} &= \tfrac{1}{3}\sum_{\kappa=1}^{n} Q_{i,j}^{k}\left(h_k^3 - h_{k-1}^3\right) \end{aligned} \tag{11}$$

where forces and moments are given (Equation (12)):

$$\begin{aligned} \begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} &= \int_{-t/2}^{t/2} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix} \cdot dz \\ \begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} &= \int_{-t/2}^{t/2} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix} \cdot z\, dz \end{aligned} \tag{12}$$

Properties are required for the formation of the ABD matrix thicknesses and principal lamina. The principal lamina properties for the thin lamina are the lamina shear modulus G_{12}, principal Young's module E_1 and E_2, and the major Poisson ratio ν_{12}. These properties are usually obtained through experimentation, OEM data, or composite materials material micromechanics theories. Micromechanics composite materials theories are well established, like the rule of mixtures and Chamis's theory, to name a few, and they are well explained in the literature. The required lamina elastic coefficients (E_1, E_2, G_{12}, and ν_{12}) as a function of phase characteristics (E_f, E_m, G_m, G_f, and volume fractions) can be computed using the following relations (Equation (13)):

$$\begin{aligned} E_1 &= V_f \cdot E_f + V_m \cdot E_m, \\ \nu_{12} &= V_f \cdot \nu_f + V_m \cdot \nu_m, \\ E_2 &= \frac{E_m}{1-\sqrt{V_f}\left(1-\frac{E_m}{E_2^f}\right)}, \\ G_{12} &= \frac{G_m}{1-\sqrt{V_f}\left(1-\frac{G_m}{G_{12}^f}\right)} \end{aligned} \tag{13}$$

After computation of the ABD matrix, the required shear modulus, for orthotropic and symmetric laminate, in the NACA boundary flutter equation can be computed from the following relation (Equation (14)):

$$G_{eq} = \frac{12}{t^3 \cdot D_{66}^{inv}} \tag{14}$$

And in the cases of QI stack-ups, the required equivalent shear modulus and flutter speed are expressed as (Equations (15) and (16)):

$$G_{xy} = \frac{1}{2} \cdot G_{12} + \frac{1}{8} \frac{E_1 \cdot (E_1 + E_2 - 2\nu_{12}E_2)}{E_1 - 2\nu_{12}^2 E_2} \tag{15}$$

$$V_F = a_0 \cdot \left[\frac{(A+2)}{78.6 \cdot A^3} \cdot \left(\frac{t}{\frac{2}{S} \cdot \int_0^{b/2} l(y) \cdot dy} \right)^3 \cdot \frac{\left(G_{12} + \frac{1}{4} \frac{E_1 \cdot (E_1 + E_2 - 2\nu_{12}E_2)}{E_1 - 2\nu_{12}^2 E_2} \right)}{\left(\frac{C_T/C_R + 1}{2} \right) \cdot \left(\frac{p}{p_0} \right)} \right]^{1/2} \tag{16}$$

All phase parameters (fiber and matrix) for all composite materials analyzed in this research are reported in the literature [15]. A diagrammatic representation of the synthetic data generation is presented in the following flowchart (Figures 10 and 11):

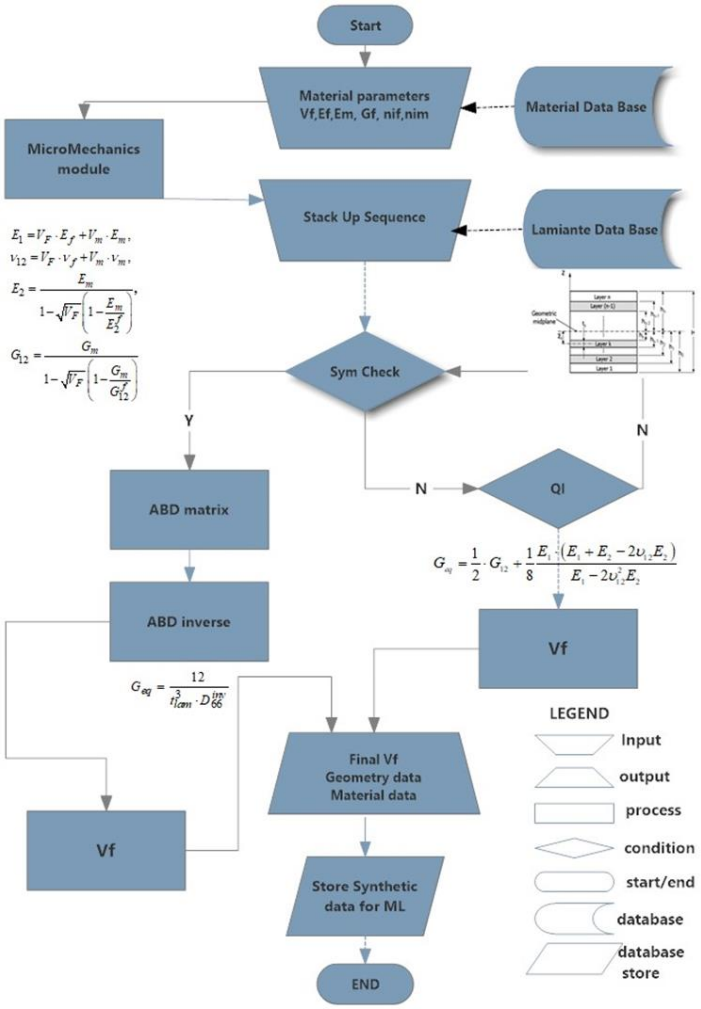


Figure 10. A diagrammatic representation of the synthetic data generation.

```
run:184671              step: 5
Data received in class to work on:

t=      0.10999999999999999
H=      9.5
CR=     9.5
CT=     9.499999999999998
G=      3480000
AR=     1
lambda= 0.9999999999999998

vf=             1059.7639133858206

run:184672              step: 5
Data received in class to work on:

t=      0.11999999999999998
H=      9.5
CR=     9.5
CT=     9.499999999999998
G=      3480000
AR=     1


        Completed Step: 5/4.6

        summary:

CSV File size :         14955[kb]       14[mb]
Data Collected:         184.68E3        [lines]
Task completed:         108.7           [%]


Completed !
Execution Time:  103952 ms
```

Figure 11. Synthetic data generation (last steps).

Based on this approach, synthetic data, based on the NACA boundary equation modified for orthotropic structures, was generated. More than 200×10^3 data entries were generated with acceptable computing times and generated file sizes (Figure 12).

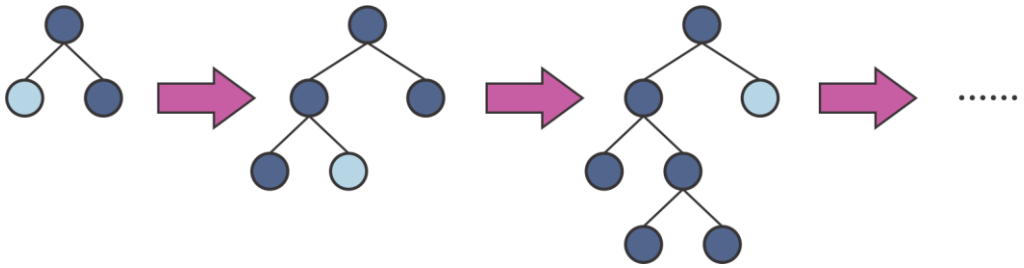


Figure 12. Figure LightGBM regression schema.

4.6. ML Algorithms Analyzed

Based on past experience and the application of metaheuristic genetic optimization algorithms, one gets the impression that the best results of the researched problem of aeroelasticity are obtained by applying the differential evolution method. This evolutionary method tunes or selects a partial space search algorithm that can provide a good enough solution to an optimization problem or ML problem, especially with incomplete or imperfect information and databases of wind tunnel experiment results or limited computer capacity [15–27].

Genetic algorithms use principles that are inspired by natural genetic populations in an attempt to evolve solutions to the analyzed problem. The basic idea is to maintain a population of chromosomes representing candidate solutions for a specific problem that develops over time through competition and controlled variation depending on changing

environmental conditions. Differential evolution, like some other genetic algorithms, operates with large design vectors of the collection, the initial population. It interprets the value of the vector function as a measure of that person's fitness as an optimum. Then, guided by the principle of survival of the fittest, the initial vector population is transformed, generation by generation, into the solution vector. The overall structure of the differential evolution algorithm resembles most other population-based searches. Two arrays are maintained, each containing a population of n-dimensional real-valued vectors. The primary array contains the current population, while the secondary array accumulates the vectors to be selected for the next generation, based on which the generation network can be formed. Selection is done by competition between existing vectors and test vectors. Test vectors used by differential evolution are formed by the mutation and recombination of vectors in the primary sequence. Mutation is an operation that makes small random changes to one or more parameters of an existing population vector. Mutation is crucial for maintaining diversity in the population and is usually performed by perturbation, which enables the avoidance of local minima and transition to a new population (generation). This significantly speeds up the search process, but we must keep in mind that it is not entirely certain that we will get the global minimum, that is, the optimal solution to the optimization problem. We must also keep in mind that the experience of the research team is extremely important and crucial for the efficiency of solving such complex problems of aeroelasticity.

In ML, the process of fitting a model to the data requires solving an optimization problem. The difficulty resides in the fact that this optimization quickly becomes very complex when dealing with real problems.

The algorithms used on previously created datasets in order to create an ML model for the flutter of composite materials material fin, with the R squared and training duration parameters, are summarized in the following table:

The R^2 (R-squared) value is a statistical measure that represents the proportion of the variance in the dependent variable that is predictable from the independent variables in a regression model. In the context of the LightGBM regression (or any regression model), R^2 is used to assess how well the model fits the data.

Stochastic dual coordinate ascent (SDCA) is a ML algorithm for solving large-scale linear regression problems. It is a stochastic optimization algorithm, which means that it updates the model parameters one at a time using a small subset of the training data. This makes the SDCA very efficient for training large models on large datasets. SDCA works by minimizing the regularized loss function, which is a combination of the data loss and the regularization term. The regularization term penalizes the model complexity, which helps to prevent overfitting.

Fast tree regression (FTR) is an efficient implementation of the gradient boosting machines (GBM) algorithm for regression tasks. It is a sequential ensemble learning method that builds multiple decision trees in a greedy fashion, where each tree is trained to correct the errors of the previous tree. FTR works by first initializing a weak learner, such as a decision tree with a single leaf node. This tree is trained on the training data to predict the target variable. The residuals between the predicted and actual target values are then calculated. Next, a new decision tree is trained to predict the residuals from the previous tree. This process is repeated until a predefined number of trees have been trained. To make predictions, FTR averages the predictions of all the trees in the ensemble. This helps to improve the accuracy of the model. Here are some of the advantages of FTR: fast training and prediction, accurate predictions, robustness to outliers and noise, and versatility. It can be used for a wide variety of regression tasks. Here are some of the disadvantages of FTR: It can be difficult to interpret the model, and it can be sensitive to the hyperparameters.

The L-BFGS Poisson regression is a supervised learning algorithm used for regression tasks with count data. It is a specialized type of Poisson regression that uses the L-BFGS algorithm to optimize the model parameters. The L-BFGS algorithm is a quasi-Newton method that is very efficient for optimizing large-scale models. It uses a limited amount of

memory to approximate the Hessian matrix, which allows it to train models quickly and efficiently. L-BFGS Poisson regression is a good choice for regression tasks with count data where accuracy and speed are important. It is also robust to outliers and noise in the data.

L-BFGS is an optimization algorithm widely employed in ML, including Poisson regression. Poisson regression is a type of regression analysis used when the dependent variable is a count, often applied to count data like the number of occurrences of an event in a fixed interval of time or space. In L-BFGS Poisson regression, the algorithm seeks to find the optimal parameters (coefficients) of the Poisson regression model. Specifically, it minimizes the negative log-likelihood function, which represents the probability of observing the given counts based on the model and parameters. L-BFGS operates efficiently even in scenarios with large-scale optimization problems, as it is designed to be memory-efficient. It stores a limited history of past gradients and updates the Hessian approximation accordingly. Depending on the variant of Poisson regression used, you may also incorporate regularization terms like L1 (Lasso) or L2 (Ridge) penalties to prevent overfitting. The optimization process continues until a stopping criterion is met, such as a specified level of convergence or a maximum number of iterations. This approach is widely employed in statistics and ML when dealing with count data. It efficiently finds the optimal parameters that best fit the data according to the Poisson distribution. Here are some of the advantages of L-BFGS Poisson regression: accurate predictions, fast training and prediction, and robustness to outliers and noise. The disadvantage is that the model interpretation can be somewhat difficult.

LightGBM regression trains a model to predict a continuous numeric output given a set of input features. The model is built by iteratively adding decision trees that minimize the loss function. At each iteration, the model computes the slope of the loss function with respect to predictions and updates the model parameters to minimize the loss. LightGBM uses a special technique called "Gradient-based One-Sampling" (GOSS) to speed up the training process. GOSS samples instances with large gradients with higher probability to reduce the number of instances considered when computing gradients. This technique helps speed up training while maintaining accuracy. LightGBM is a popular open-source gradient boosting framework that uses decision tree algorithms to perform regression and classification tasks. "Light" in LightGBM refers to the fast performance achieved through multiple techniques, such as gradient-based single-sided sampling and exclusive feature bundling. LightGBM works by building a series of decision trees, each of which is trained to predict the residual error of the previous tree. The residual error is the difference between the actual target value and the predicted target value of the previous tree (Figure 12).

This algorithm uses a number of optimizations to make it faster and more efficient than other gradient-boosting algorithms. For example, LightGBM uses a histogram-based algorithm to split the data at each node of the decision tree. This is much faster than the traditional greedy algorithm, which searches through all possible split points.

It is also very scalable. It can be used to train models on very large datasets, even on distributed systems.

4.7. Algorithms Analyzed Comparative Analysis

SDCA regression is an optimization algorithm used for solving regularized empirical risk minimization problems, including regression tasks. It employs stochastic dual coordinate ascent for efficient optimization in the dual space, making it suitable for large datasets. Additionally, it can be parallelized for distributed computing and efficiently handles sparse data, contributing to its versatility across various ML tasks. However, it may require careful hyperparameter tuning for optimal performance [28].

Fast tree regression is an ensemble learning algorithm primarily used for regression tasks. It is based on decision trees, which enables it to capture complex relationships in the data. By building an ensemble of decision trees and aggregating their predictions, fast tree regression achieves efficient training, especially compared to traditional gradient-boosting methods. This algorithm can also be parallelized for further efficiency.

L-BFGS Poisson regression, on the other hand, is a specialized optimization algorithm designed for Poisson regression tasks, which are essential for modeling count data. It utilizes the limited-memory Broyden–Fletcher–Goldfarb–Shanno method for efficient optimization in regularized empirical risk minimization problems. L-BFGS Poisson regression is known for its efficiency in solving these specific types of regression problems.

LightGBM regression is an efficient gradient-boosting framework that uses tree-based learning algorithms. It is particularly known for its speed and efficiency in handling large datasets and high-dimensional feature spaces. LightGBM utilizes a histogram-based approach for training decision trees, which significantly speeds up the training process. This makes it a popular choice for various ML tasks, including regression [8].

Comparing the algorithms, the SDCA regression exhibits low complexity, making it suitable for large-scale problems, while fast tree regression, though moderately complex, is capable of handling complex relationships in the data. L-BFGS Poisson regression falls in the same complexity range and is tailored for Poisson regression tasks. LightGBM Regression stands out for its efficiency, especially in handling large datasets and high-dimensional feature spaces.

In terms of speed and efficiency, the SDCA Regression is known for its fast convergence rate and efficiency, especially with large datasets. Fast tree regression demonstrates efficient training, particularly when compared to traditional gradient boosting methods. L-BFGS Poisson regression is efficient in solving Poisson regression problems, while LightGBM Regression is particularly acclaimed for its speed and efficiency in handling large datasets.

Considering versatility, SDCA Regression is highly versatile and applicable to various ML tasks beyond regression. Fast tree regression is primarily designed for regression tasks, but with some modification, it can be extended to classification problems. L-BFGS Poisson regression is specialized for Poisson regression tasks and is particularly well-suited for modeling count data. LightGBM regression is versatile and widely used across a range of ML tasks due to its efficiency and speed.

When it comes to data requirements, the SDCA Regression is adept at handling large datasets and is particularly efficient with sparse data. Fast tree regression is flexible in handling various types of data but may require a sufficient number of samples for capturing complex relationships. L-BFGS Poisson regression is specialized for count data, making it suitable for Poisson regression tasks. LightGBM regression is well-suited for large datasets and high-dimensional feature spaces due to its efficiency.

In conclusion, the choice between these algorithms should consider factors such as the characteristics of the dataset, the nature of the prediction task, available computational resources, and the level of interpretability required for the results (Table 3).

Table 3. Characteristics of the analyzed regression algorithms for the ML model.

Algorithm	Complexity	Speed/Efficiency	Versatility	Data Requirements
SDCA Regression	Low	Fast convergence	Highly versatile, sparse data handling	Large datasets, sparse data
Fast Tree Regression	Moderate	Efficient	Flexible, Extendable to Classification	Various data types, sufficient samples
LBFGS Poisson Regression	Moderate	Efficient	Specialized for Poisson regression	Suitable for count data
LightGBM Regression	Moderate	Very efficient	Versatile, high-dimensional data handling	Large datasets, high-dimensional data

The analysis data flow is presented in the Figure 13.

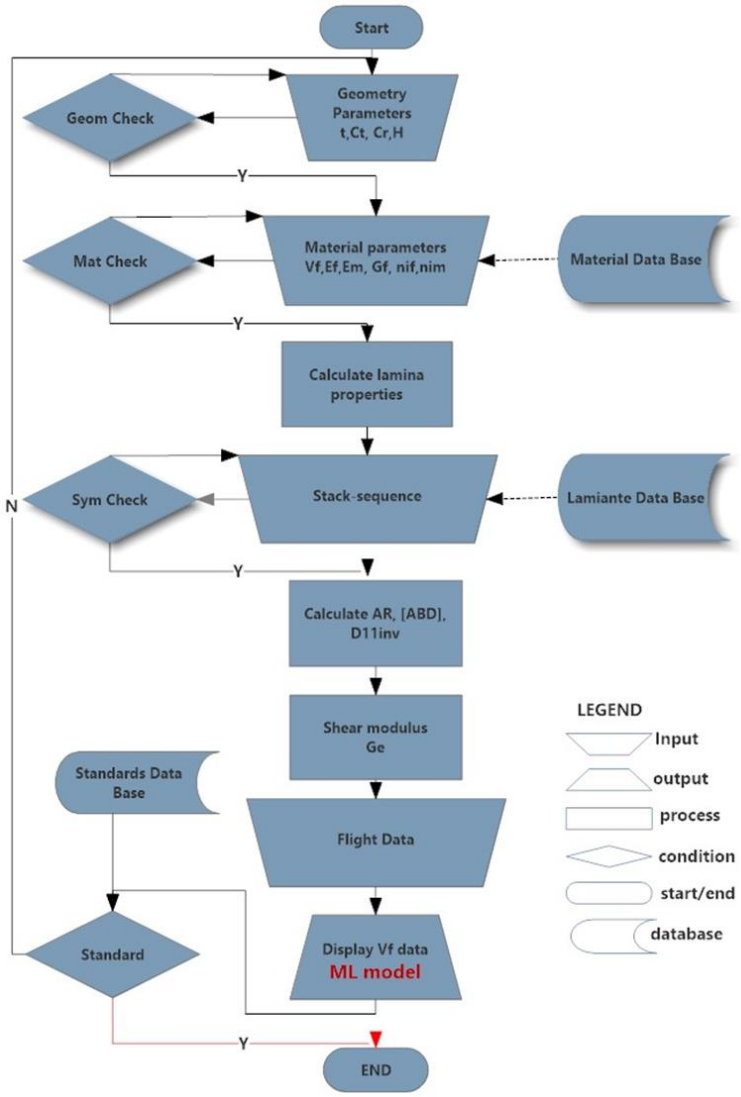


Figure 13. Analysis process flow.

5. Results and ML Model Evaluation

The results for a particular flutter data set generated and regression algorithms analyzed in this research are summarized in Table 4. The Light GBM regression algorithm was chosen for the ML model since it managed to achieve the highest correlation coefficient (or coefficient of determination) of $R^2 = 0.9988$. This coefficient is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable and directly shows the data fit to the regression model. In general, regression algorithms are specifically designed for value prediction tasks in ML. In regression, the goal is to predict a continuous numerical value rather than a categorical label. Value prediction problems are prevalent in various domains, and regression algorithms are well-suited for addressing such tasks. The choice of a regression algorithm depends on factors such as the nature of the data, the complexity of the relationships, and the interpretability of the model. It is common to experiment with multiple algorithms and select the one that performs best on the specific task at hand.

Table 4. Regression algorithms analyzed results for the flutter dataset.

	Trainer	R^2	Duration
	[name]	[-]	[h]
1	SdcaRegression	0.0083	2.5980
2	FastTreeRegression	0.0706	0.3430
3	LbfgsPoissonRegression	−0.3708	0.3800
4	LightGbmRegression	0.6611	0.9340
5	LightGbmRegression (≈200 × 10^3 size)	0.9988	6.3280

Duration is another parameter of great importance in algorithm selection for a particular problem. The term "duration" typically refers to the temporal aspect of the data, indicating the time span or period over which the ML model is trained and evaluated. The selection of the most suitable algorithm for fin flutter analysis in this research was not based on the minimization of duration but on the maximization of the correlation coefficient. Among the algorithm candidates, the selected Light GBM regression algorithm has the highest duration time for the coefficient R^2.

Having a very small dataset may lead to overfitting, where the model memorizes the training data instead of learning generalizable patterns. In a particular problem, the size of the data set was chosen by initially setting the constraint on R^2 to be equal to or greater than 0.9 and increasing the number of data points in the set until the desired R^2 value was achieved (per algorithm). The selected algorithm (Light GBM regression) managed to satisfy this constraint with the data set of approximately 200 × 10^3 fin data points (Table 4, Figure 13).

The flutter speeds, calculated using the ML approach proposed in this research, are verified on an E-glass composite materials fin of an arbitrary geometry and material composition that was not included in the process of dataset generation. The test geometry and material characteristics are presented in the following figure (Figure 14):

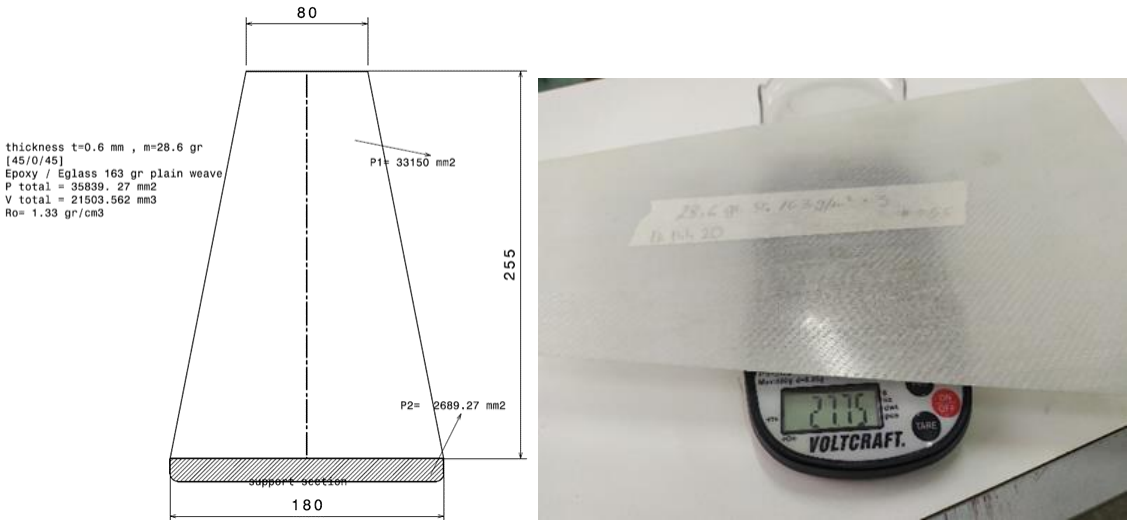


Figure 14. Geometry of the fin test sample used for the ML flutter model verification.

The ML flutter model was implemented in a bespoke flutter software named CODE F, and the input module interface (geometry and material data, with the data for the test fin) is presented in the following pictures (Figure 15):

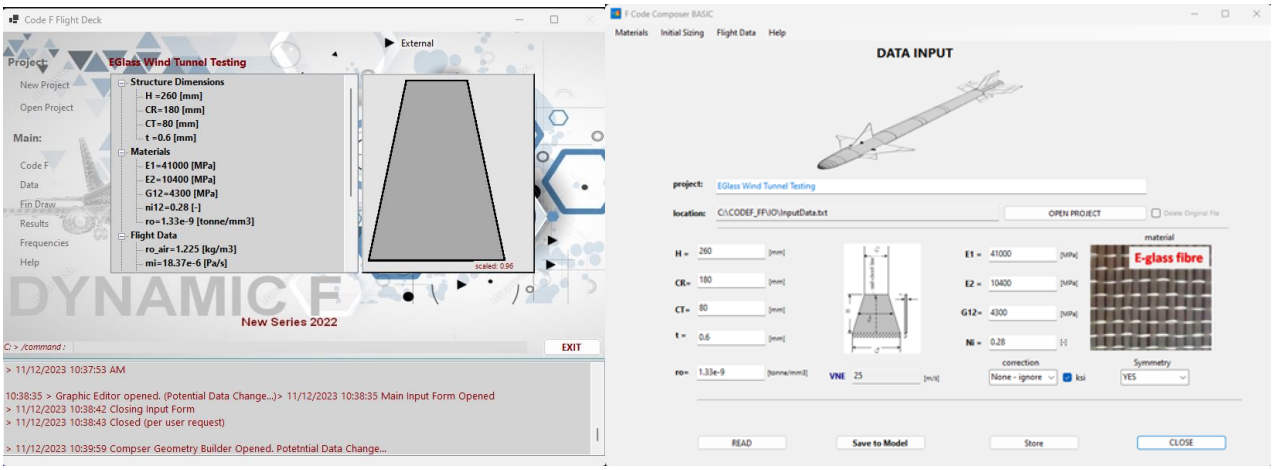


Figure 15. Geometry of the fin test sample used for the ML flutter model verification.

Based on the dimensions and material characteristics (Figure 14), test samples were manufactured for the purpose of experimental testing in the wind tunnel. The test setup and experimental results are presented in the following figure (Figure 16).

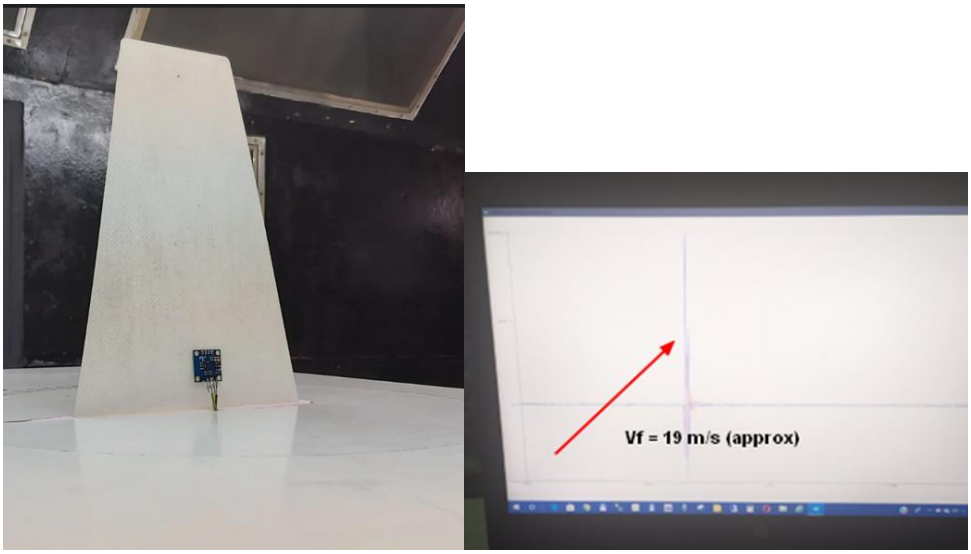


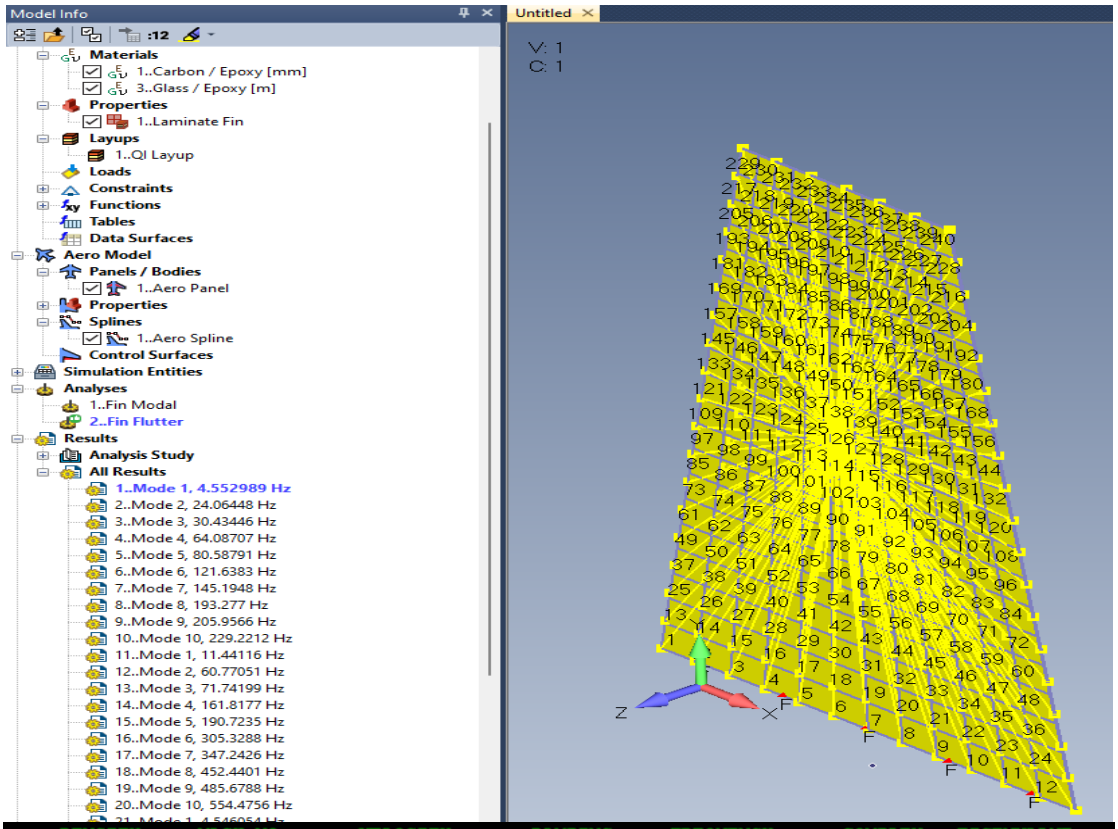
Figure 16. Geometry of the fin test sample used for the ML flutter model verification.

For the fin test sample, flutter speeds were calculated based on the CFA model, the binary flutter model (unsteady aerodynamics), and the NACA boundary flutter model.

The CFA (computational flutter) approach based on the commercial NASTRAN software model (structural, aerodynamic, and spline), with frequencies in bending and torsion, is given in the following pictures (Figure 17):

Frequencies in bending and torsion obtained using CODE F software (Figure 18):

The results of the calculated flutter speed for the test fin analyzed are presented in the following figure (Figure 19):



DENSITY	MACH NO.	VELOCITY	DAMPING	FREQUENCY	COMPLEX	EIGENVALUE
1.00000E+00	0.00000E+00	5.00000E+00	-2.13893E-01	4.34045E+00	-2.91663E+00	2.72718E+01
1.00000E+00	0.00000E+00	5.83000E+00	-2.57220E-01	4.30762E+00	-3.48090E+00	2.70656E+01
1.00000E+00	0.00000E+00	6.67000E+00	-3.04273E-01	4.26710E+00	-4.07893E+00	2.68110E+01
1.00000E+00	0.00000E+00	7.50000E+00	-3.54737E-01	4.21877E+00	-4.70156E+00	2.65073E+01
1.00000E+00	0.00000E+00	8.33000E+00	-4.10184E-01	4.16109E+00	-5.36212E+00	2.61449E+01
1.00000E+00	0.00000E+00	9.17000E+00	-4.72771E-01	4.09124E+00	-6.07652E+00	2.57060E+01
1.00000E+00	0.00000E+00	1.00000E+01	-5.42801E-01	4.00978E+00	-6.83772E+00	2.51942E+01
1.00000E+00	0.00000E+00	1.08300E+01	-6.23496E-01	3.91275E+00	-7.66418E+00	2.45845E+01
1.00000E+00	0.00000E+00	1.16700E+01	-7.19746E-01	3.79475E+00	-8.58050E+00	2.38431E+01
1.00000E+00	0.00000E+00	1.25000E+01	-8.34939E-01	3.65245E+00	-9.58052E+00	2.29490E+01
1.00000E+00	0.00000E+00	1.33300E+01	-9.79115E-01	3.47739E+00	-1.06964E+01	2.18491E+01
1.00000E+00	0.00000E+00	1.41700E+01	-1.17119E+00	3.25281E+00	-1.19684E+01	2.04380E+01
1.00000E+00	0.00000E+00	1.50000E+01	-1.44000E+00	2.96133E+00	-1.33967E+01	1.86066E+01
1.00000E+00	0.00000E+00	1.58300E+01	-1.86991E+00	2.55947E+00	-1.50356E+01	1.60816E+01
1.00000E+00	0.00000E+00	1.66700E+01	-2.78706E+00	1.93622E+00	-1.69532E+01	1.21657E+01
1.00000E+00	0.00000E+00	1.75000E+01	-1.23708E+01	4.92864E-01	-1.91547E+01	3.09676E+00
1.00000E+00	0.00000E+00	1.83300E+01	-9.32304E-02	0.00000E+00	-8.77419E+00	0.00000E+00
1.00000E+00	0.00000E+00	1.91700E+01	-4.88414E-02	0.00000E+00	-4.80725E+00	0.00000E+00
1.00000E+00	0.00000E+00	2.00000E+01	-1.90912E-02	0.00000E+00	-1.96043E+00	0.00000E+00
1.00000E+00	0.00000E+00	2.08300E+01	3.67544E-03	0.00000E+00	3.93085E-01	0.00000E+00
1.00000E+00	0.00000E+00	2.16700E+01	2.23533E-02	0.00000E+00	2.48706E+00	0.00000E+00
1.00000E+00	0.00000E+00	2.25000E+01	3.79048E-02	0.00000E+00	4.37889E+00	0.00000E+00
1.00000E+00	0.00000E+00	2.33300E+01	5.14199E-02	0.00000E+00	6.15932E+00	0.00000E+00
1.00000E+00	0.00000E+00	2.41700E+01	6.36101E-02	0.00000E+00	7.89387E+00	0.00000E+00
1.00000E+00	0.00000E+00	2.50000E+01	7.46151E-02	0.00000E+00	9.57754E+00	0.00000E+00
1.00000E+00	0.00000E+00	2.58300E+01	8.49337E-02	0.00000E+00	1.12640E+01	0.00000E+00
1.00000E+00	0.00000E+00	2.66700E+01	9.49952E-02	0.00000E+00	1.30080E+01	0.00000E+00
1.00000E+00	0.00000E+00	2.75000E+01	1.04877E-01	0.00000E+00	1.48081E+01	0.00000E+00
1.00000E+00	0.00000E+00	2.83300E+01	1.15059E-01	0.00000E+00	1.67361E+01	0.00000E+00
1.00000E+00	0.00000E+00	2.91700E+01	1.26148E-01	0.00000E+00	1.88931E+01	0.00000E+00
1.00000E+00	0.00000E+00	3.00000E+01	1.38580E-01	0.00000E+00	2.13456E+01	0.00000E+00
1.00000E+00	0.00000E+00	3.08300E+01	1.53647E-01	0.00000E+00	2.43212E+01	0.00000E+00
1.00000E+00	0.00000E+00	3.16700E+01	1.73840E-01	0.00000E+00	2.82673E+01	0.00000E+00
1.00000E+00	0.00000E+00	3.25000E+01	2.03310E-01	0.00000E+00	3.39258E+01	0.00000E+00
1.00000E+00	0.00000E+00	3.33300E+01	2.49834E-01	0.00000E+00	4.27538E+01	0.00000E+00
1.00000E+00	0.00000E+00	3.41700E+01	3.10877E-01	0.00000E+00	5.45407E+01	0.00000E+00

Figure 17. The Nastran code flutter results for the composite materials test geometry. The output file (partial) is presented. Using the PK flutter model for the flutter (calculated, CFA, approach V_F obtained is 20.8 m/s).

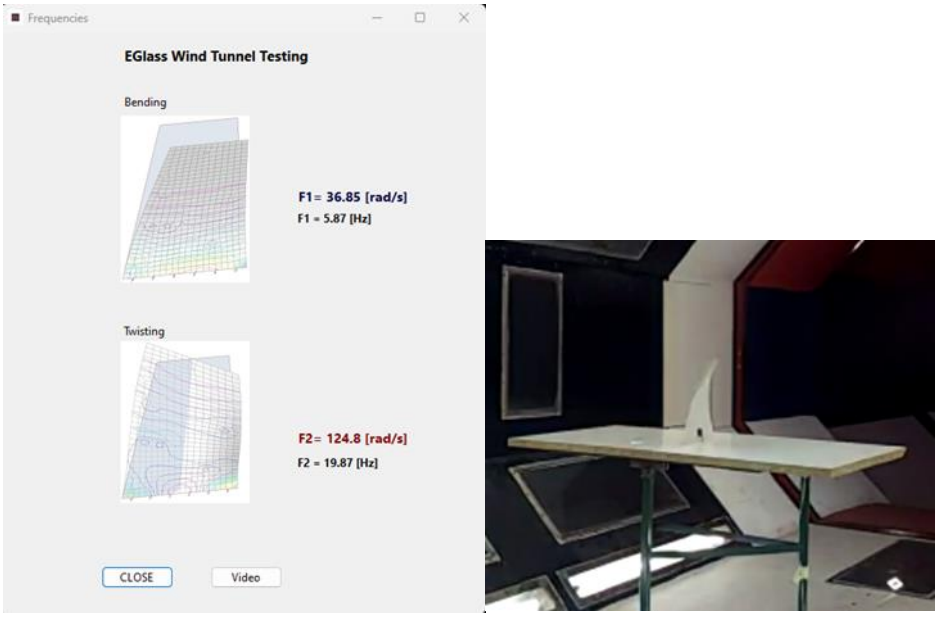


Figure 18. The Nastran code flutter results for the composite materials test geometry. The output file (partial) is presented. Using the PK flutter model for the flutter (calculated, CFA, approach V_F obtained is 20.8 m/s).

EGlass Wind Tunnel Testing

Calulcationg Unsteady flutter speed using K-Method...

	k	g1	V1 [m/s]	g2	V2 [m/s]
1.	0.001	−0.012	12.12	+0.005	3531.64
2.	0.002	−0.025	12.12	+0.011	1765.95
3.	0.003	−0.037	12.13	+0.016	1177.44
4.	0.004	−0.050	12.13	+0.021	883.23
5.	0.005	−0.062	12.14	+0.026	706.74
6.	0.006	−0.074	12.14	+0.032	589.10
7.	0.007	−0.086	12.15	+0.037	505.10
8.	0.008	−0.099	12.16	+0.042	442.12
9.	0.009	−0.111	12.17	+0.047	393.16
10.	0.010	−0.123	12.18	+0.053	354.00
11.	0.020	−0.240	12.33	+0.103	178.27
12.	0.030	−0.350	12.51	+0.151	120.23
13.	0.040	−0.451	12.67	+0.195	91.57
14.	0.050	−0.545	12.79	+0.234	74.63
15.	0.060	−0.632	12.85	+0.266	63.49
16.	0.070	−0.712	12.87	+0.292	55.61
17.	0.080	−0.785	12.83	+0.311	49.74
18.	0.090	−0.850	12.75	+0.323	45.17
19.	0.100	−0.906	12.64	+0.330	41.49
20.	0.110	−0.954	12.49	+0.331	38.44
21.	0.120	−0.992	12.31	+0.327	35.87
22.	0.130	−1.021	12.10	+0.319	33.66
23.	0.140	−1.041	11.88	+0.308	31.72
24.	0.150	−1.053	11.64	+0.294	30.02
25.	0.160	−1.058	11.39	+0.278	28.51
26.	0.170	−1.055	11.13	+0.260	27.15
27.	0.180	−1.047	10.87	+0.241	25.93
28.	0.190	−1.034	10.60	+0.220	24.83
29.	0.200	−1.016	10.32	+0.199	23.83
30.	0.210	−0.994	10.05	+0.178	22.92
31.	0.220	−0.970	9.78	+0.156	22.09
32.	0.230	−0.943	9.52	+0.134	21.33
33.	0.240	−0.915	9.25	+0.113	20.64
34.	0.250	−0.886	8.99	+0.091	20.00
35.	0.260	−0.856	8.74	+0.069	19.42
36.	0.270	−0.825	8.50	+0.048	18.89
37.	0.280	−0.795	8.26	+0.028	18.39
38.	0.290	−0.766	8.03	+0.008	17.94
39.	0.300	−0.736	7.81	−0.012	17.52
40.	0.310	−0.708	7.59	−0.031	17.13
41.	0.320	−0.680	7.38	−0.049	16.77
42.	0.330	−0.654	7.18	−0.066	16.44

Figure 19. The unsteady aerodynamics, based on the Theodorsen model and K-flutter algorithm flutter results from the Code F software v.1.0, for the test fin geometry (V_F obtained is 17.52 m/s).

The results from the developed ML model based on the Light GBM retrogression for the test fin are given in Figure 20:

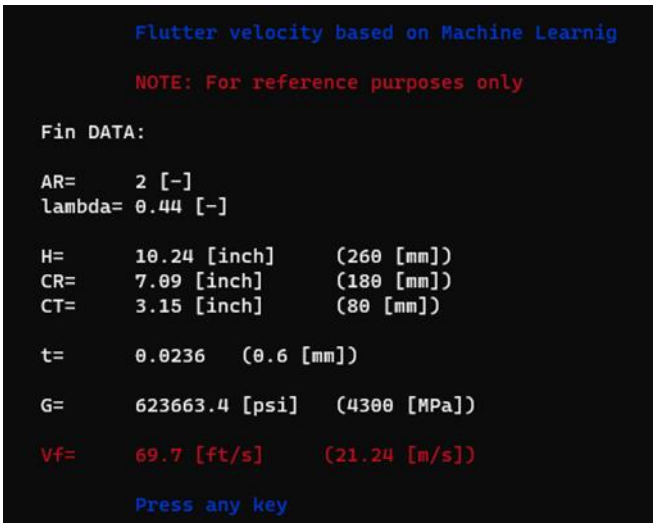


Figure 20. ML, flutter speed prediction from the Code F software, for the test fin geometry (V_F obtained is 21.24 m/s).

The test results (from the experiment to all flutter models analyzed) are summarized in the following table (Table 5).

Table 5. Test fin sample comparative flutter speed results.

Model	V_F [m/s]
Experiment	19.0
Typical section binary model (unsteady aerodynamics)	17.52
CFA (NASTRAN, p-k algorithm)	20.8
Modified NACA boundary flutter model (Equations (8)–(16))	18.11
ML model Light GBM regression	21.24

Depending on the approach, the flutter estimates can vary. In this particular research, one of the objectives was to find (and modify) the most suitable model that can be efficiently used in ML model generation. The first two models (both binary models in terms of DOFs) are based on the widely accepted typical section model. The first typical section model assumes stationary aerodynamics; on the other hand, based on this model, the equations of motion can be solved in a closed form. Expanding this model and using unsteady aerodynamics (based on the Theodorsen model), the binary model becomes more complex and cannot be solved in a closed form, requiring numerical solutions; however, it renders better results when compared to the wind tunnel tests. The NACA boundary model, apart from all flight conditions and fin geometric parameters, requires only the calculation of the fin's material equivalent shear modulus without the calculation of frequencies in bending and torsion. Finally, the CFA model (aerodynamic, structural, and spline) is the most accurate model; however, its development, computation time, and expensive software require immense resources.

6. Conclusions

In the present work, the potential application of ML methodology for the analysis of the flutter of composite materials missile fins is investigated. Flutter is known to be a potentially catastrophic occurrence for almost all flying vehicles. On the other hand, even throughout history, researchers have struggled to develop efficient and precise flutter models to prevent flutter from occurring to minimize the costs of experimental analysis and speed up the design process.

The main conclusions obtained during this research are as follows:

1. Machine learning techniques can be successfully deployed in general flutter analysis, however, with caution. Model training requires substantial resources and time.
2. For the flutter analysis of composite materials missile fins, and creation of synthetic data, the NACA flutter boundary model is the most convenient in terms of programmatic application (code development); however, it is not the most accurate when compared to CFA (computational flutter analysis, based on NASTRAN code). Furthermore, if composite materials are in question, they have to be modified to incorporate the orthotropic nature of these materials.
3. To develop the ML flutter model, several regression algorithms were analyzed from the aspect of accuracy, required dataset size, and precision. It was found that in terms of data correlation (R^2 values), the LightGBM regression is the most accurate but requires large data sets and long processing times.
4. Once the model based on the LightGBM regression algorithm is trained, the ML model renders accurate flutter speed results in a subsonic regime for composite materials missile fins (within 12% when compared to wind tunnel experiments and the CFA approach).
5. This methodology can be deployed in the preliminary stages of design when large potential fin configurations have to be analyzed (different geometries and different composite materials).

Finally, the complete process (from dataset generation to computer code development is presented in the following picture (Figure 21):

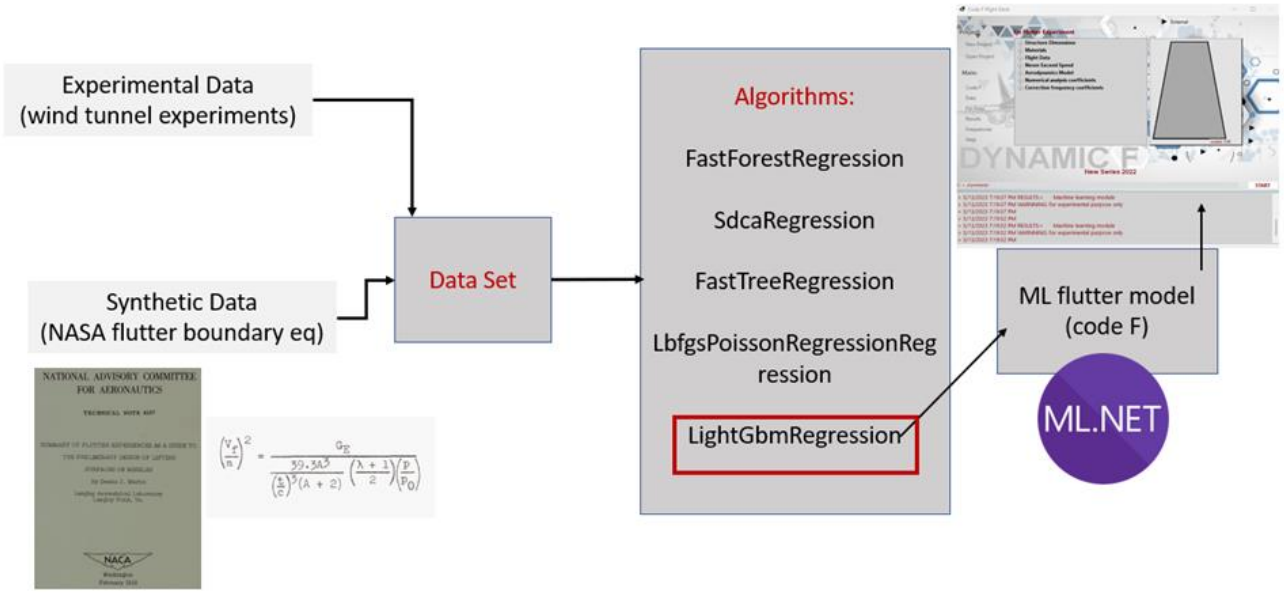


Figure 21. Development process schematic (summary).

Author Contributions: Conceptualization, M.D., A.B. and B.R.; methodology, M.D.; software, M.D.; validation., A.B. and B.R.; formal analysis, M.D.; investigation, M.D. and A.B.; resources, M.D.; data curation, M.D.; writing—original draft preparation, M.D.;.writing—review and editing, A.B. and B.R.; visualization, M.D.; supervision, B.R.; project administration, B.R.; funding acquisition B.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

Abbreviations

k	Reduced frequency	V_f	Fiber volume coefficient
ω	Frequency of vibration	V_m	Matrix volume coefficient
V_0	Linear flight speed	a_0	Speed of sound at H = 0
l	Plate reference chord	ω_h, ω_α	Natural frequencies (bending, torsion)
C_R	Root chord	V_F	Speed at flutter
C_T	Tip chord	g_h, g_α	Damping
H	Tapered plate span	t	Plate thickness
h	Plunging degree of freedom	μ	Plate to airstream mass ratio
α	Rotational degree of freedom	S_k	Static moment of inertia
$M(t)$	Aerodynamic moment	E_1, E_2	Moduli of elasticity (fiber and across fiber direction)
$R_z(t)$	Aerodynamic force	E_{45}	Young's modulus of elasticity in 45° directions
M_w	Fin mass	G_{12}	In-plane shear modulus

References

1. Wang, Y.-R.; Wang, Y.-J. Flutter speed prediction by using deep learning. *Adv. Mech. Eng.* **2021**, *13*, 16878140211062275. [CrossRef]
2. Liao, H.; Mei, H.; Hu, G.; Wu, B.; Wang, Q. Machine learning strategy for predicting flutter performance of streamlined box girders. *J. Wind. Eng. Ind. Aerodyn.* **2021**, *209*, 104493. [CrossRef]
3. Tinmitondé, S.; He, X.; Yan, L.; Hounye, A.H. Data-driven prediction of critical flutter velocity of long-span suspension bridges using a probabilistic machine learning approach. *Comput. Struct.* **2023**, *280*, 107002. [CrossRef]
4. Casoni, M.; Benini, E. A Review of Computational Methods and Reduced Order Models for Flutter Prediction in Turbomachinery. *Aerospace* **2021**, *8*, 242. [CrossRef]
5. Rizzo, F.; Caracoglia, L. Artificial Neural Network model to predict the flutter velocity of suspension bridges. *Comput. Struct.* **2020**, *233*, 106236. [CrossRef]
6. AE, H.; Darwish, A.; El-Askary, H. *Machine Learning and Data Mining in Aerospace Technology*; Springer: Berlin/Heidelberg, Germany, 2020.
7. Brunton, S.L.; Nathan Kutz, J.; Manohar, K.; Aravkin, A.Y.; Morgansen, K.; Klemisch, J.; Goebel, N.; Buttrick, J.; Poskin, J.; Blom-Schieber, A.W.; et al. Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning. *AIAAJ Aeronaut. Astronaut.* **2021**, *59*, 2820–2847. [CrossRef]
8. Shalev-Shwartz, S.; Ben-David, S. *Understanding Machine Learning from Theory to Algorithms*; Cambridge University Press: Cambridge, UK, 2014.
9. Alpaydin, E. *Introduction to Machine Learning*; The MIT Press: Cambridge, MA, USA, 2004.
10. Roy, R.K. *Design of Experiments Using the Taguchi Approach*; John Wiley & Sons: Hoboken, NJ, USA, 2001.
11. Tola, C.; Nikbay, M. Investigation of the effect of thickness, taper ratio and aspect ratio on fin flutter velocity of a model rocket using response surface method. In Proceedings of the 2015 7th International Conference on Recent Advances in Space Technologies (RAST), Istanbul, Turkey, 16–19 June 2015; pp. 27–32.
12. Fleeman, L.E. *Tactical Missile Design*; AIAA Education Series; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2001.
13. Fleeman, L.E. *Missile Design and System Engineering*; AIAA Education Series; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2012.
14. *Simcenter NASTRAN Aeroelastic Analysis User's Guide*; Siemens Product Lifecycle Management Software Inc.: Munich, Germany, 2019.
15. Bisplinghoff, R.L.; Ashley, H.; Halfman, R.L. *Aeroelasticity (Dover Books on Aeronautical Engineering)*; Dover Publications: New York, NY, USA, 2013.
16. Meijer, M.-C. Aeroelastic Prediction for Missile Fins in Supersonic Flows. In Proceedings of the 29th Congress of the International Council of the Aerospace Sciences, St. Petersburg, Russia, 7–12 September 2014.

17. Yuan, W.; Zhang, X.; Poirel, D.C. Flutter Analysis Solution Stabilization for the PK-Method. In Proceedings of the AIAA Aviation 2021 Forum, Virtual, 2–6 August 2021.
18. Dinulović, M.; Rašuo, B.; Slavković, A.; Zajić, G. Flutter Analysis of Tapered Composite materials Fins: Analysis and Experiment. *FME Trans.* **2022**, *50*, 576–585. [CrossRef]
19. Hodges, D.; Pierce, G.A. *Introduction to Structural Dynamics and Aeroelasticity*; Cambridge Aerospace Series; Cambridge University Press: Cambridge, UK, 2011.
20. Martin, D. *Summary of Flutter Experiences as a Guide to the Preliminary Design of Lifting Surfaces on Missiles*; NACA TN 4197; National Advisory Committee for Aeronautics: Hampton, VA, USA, 1958.
21. Tanga, D.M.; Yamamoto, H.; Dowell, E.H. Flutter and limit cycle oscillations of two-dimensional panels in three-dimensional axial flow. *J. Fluids Struct.* **2003**, *17*, 225–242. [CrossRef]
22. Eloy, C.; Lagrange, R.; Souilliez, C.; Schouveiler, L. Aeroelastic instability of cantilevered flexible plates in uniform flow. *J. Fluid Mech.* **2008**, *611*, 97–106. [CrossRef]
23. Rahtika, I.P.G.S.; Wardana, I.N.G.; Sonief, A.A.; Siswanto, E. Experimental Investigation on Flutter Similitude of Thin-Flat Plates. *Adv. Acoust. Vib.* **2017**, *2017*, 7091425. [CrossRef]
24. Dodic, M.; Krstic, B.; Rasuo, B.; Dinulovic, M.; Bengin, A. Numerical Analysis of Glauert Inflow Formula for Single-Rotor Helicopter in Steady-Level Flight below Stall-Flutter Limit. *Aerospace* **2023**, *10*, 238. [CrossRef]
25. Akkerman, R. On the properties of quasi-isotropic laminates. *Compos. Mater. Part B Eng.* **2002**, *33*, 133–140. [CrossRef]
26. Daniel, I.; Ishai, O. *Engineering Mechanics of Composite Materials*, 2nd ed.; Oxford Press: Oxford, UK, 2006.
27. Tsai, S.W.; Hahn, H.T. *Introduction to Composite Materials*; Lancaster Techtonic: Maidenhead, UK, 1980.
28. Shalev-Shwartz, S.; Zhang, T. Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization. *J. Mach. Learn. Res.* **2013**, *14*, 567–599.