Aerodynamic Shape Optimization Using Machine Learning and Evolutionary Algorithms

Jonathan A. Sánchez-Muñoz *, Christian Lagarza-Cortés, Jorge Ramírez-Cruz, Juan Manuel Silva-Campos, Gustavo Flores-Eraña *

Posted Date: 26 February 2026

doi: 10.20944/preprints202602.1777.v1

Keywords: drag reduction; aerodynamic shape optimization; machine learning; evolutionary algorithms; blunt body; supersonic flows; gradient boosting; differential evolution; surrogate model

 doi:10.20944/preprints202602.1777.v1

Article

Aerodynamic Shape Optimization Using Machine Learning and Evolutionary Algorithms

Jonathan A. Sánchez-Muñoz [1,*], Christian Lagarza-Cortés [2], Jorge Ramírez-Cruz [3], Juan Manuel Silva-Campos [1] and Gustavo Flores-Eraña [1,*]

[1] School of Engineering and Sciences, Tecnologico de Monterrey, Av. Eugenio Garza Sada 300, San Luis Potosí S.L.P 78211, México

[2] Department of Industrial and Mechanical Engineering, Universidad de las Américas Puebla, Ex Hacienda Sta. Catarina Mártir S/N, San Andrés Cholula Puebla 72810, México

[3] División de Ciencias Básicas, Facultad de Ingeniería, Universidad Nacional Autónoma de México, Av. Universidad 3000, Ciudad Universitaria, Coyoacán, Ciudad de México 04510, México

* Correspondence: jonathan.sanchez@tec.mx (J.A.S.-M.); gustavo.flores@tec.mx (G. F.-E.)

Abstract

This study proposes a surrogate-assisted evolutionary optimization framework for small dataset that integrates machine learning–based surrogate models with evolutionary algorithms for the aerodynamic optimization of a spiked blunt body in supersonic flow. A database of simulated cases covering a range of Mach numbers, spike length ratios (L/D), and diameter ratios (d/D) was used to train regression models and identify optimal geometries. Among the tested algorithms, the Gradient Boosting Regressor (GBR) achieved the best predictive performance (R^2 = 0.8909, RMSE = 0.00775), accurately capturing the nonlinear dependencies of the drag coefficient (C_d). Evolutionary optimization methods, including Differential Evolution (DE), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and Genetic Algorithm (GA), consistently converged to near-optimal configurations, with DE exhibiting the most stable behavior across Mach regimes. SHapley Additive exPlanations (SHAP) analysis revealed that (L/D) is the most influential parameter on C_d, followed by Mach number and (d/D), highlighting the dominant effect of geometric slenderness in drag reduction. The integration of data-driven modeling with evolutionary computation provides a robust framework for aerodynamic optimization, offering both predictive accuracy and physical interpretability. These results demonstrate the potential of hybrid Machine Learning-Evolutionary algorithms and CFD approaches to accelerate the design of high-performance configurations in supersonic applications.

Keywords: drag reduction; aerodynamic shape optimization; machine learning; evolutionary algorithms; blunt body; supersonic flows; gradient boosting; differential evolution; surrogate model

1. Introduction

For decades, experimental techniques such as wind tunnels—sometimes using scaling laws—have predominated across various sectors, including the automotive, naval, and aeronautical industries. Nevertheless, technological development has enabled more efficient analysis and design techniques. Computational fluid dynamics (CFD) plays a central role in the analysis and design of high-speed aerodynamic systems, particularly in supersonic and hypersonic regimes [1,2]. However, the application of CFD to engineering optimization is often limited by its high computational cost, especially when transient simulations, complex geometries, shock-dominated flows, or high-dimensional design spaces are involved [3,4]. These requirements demand fine spatial and temporal resolutions, advanced shock-capturing schemes, and long computational times, making direct CFD-based optimization impractical for large-scale parametric or design studies [2,5–8].

 doi:10.20944/preprints202602.1777.v1

To mitigate the computational cost associated with CFD design process, surrogate models have been used as fast approximation that emulate complex flow solvers [7–11]. Machine Learning (ML) has enabled the possibility of developed surrogate models capable of addressing some of the limitations of traditional trial and error methods for CFD design optimization [12–19]. Surrogate models based on artificial neural networks[19,20], deep learning[21,22] or ensemble methods [23] have shown improved generalization capabilities and scalability, making them suitable for complex flow–geometry interactions, for example aerodynamic and Multiphysics optimization problems, although their performance remains sensitive to the quality of the training data and the strategy used to sample the design space.

In parallel with advances in surrogate modelling, the optimization of CFD-driven problems has highlighted the need for robust global optimization strategies [3,6,7]. Many aerodynamic design problems are inherently nonconvex and multimodal, with objective functions that may be discontinuous or noisy due to numerical effects associated with shock capturing and turbulence modeling [2,5]. Under these conditions, gradient-based optimization methods may become unreliable or inefficient [4,6], motivating the use of evolutionary and population-based algorithms that do not rely on gradient information.

Genetic algorithms (GAs) have emerged as complement to machine learning surrogate models for CFD design optimization [29–32]. By coupling surrogate models with evolutionary algorithms, it becomes possible to evaluate candidate solutions at a reduced computational cost. This hybrid methodology enables the possibility of global exploration of complex design spaces and has proven effective by characterizing multiple local optima and nonlinear interactions between design variables.

An interesting application for surrogate-evolutionary optimization not well studied is the aerodynamic design of spiked blunt bodies in supersonic flow [20–24]. Blunt bodies operating at supersonic speeds generate strong bow shocks that result in high aerodynamic drag and significant thermal loads due to intense gas compression at the shock front [1,2,8]. The introduction of a spike extending from the nose of the body modifies the shock structure by generating a primary oblique shock that weakens the secondary shock formed on the blunt body, thereby reducing drag and heat transfer [8]. The effectiveness of this mechanism depends on multiple geometric and flow parameters, making the problem highly nonlinear and multi-parametric [9,12].

Although extensive experimental and numerical studies have contributed to the understanding of drag-reduction mechanisms in spiked blunt bodies [8–10], high-fidelity investigations remain computationally and experimentally demanding. CFD simulations of supersonic spike–body configurations require fine grids, accurate turbulence and shock-capturing models, and long computational times, particularly when parametric or optimization studies are conducted [2,5,6]. Similarly, experimental investigations rely on specialized wind tunnel facilities and precise instrumentation, limiting the feasibility of large-scale, systematic exploration of the design space [9,11].

Motivated by these challenges, the present study proposes a surrogate-assisted evolutionary optimization framework for small dataset that integrates machine learning–based surrogate models with evolutionary algorithms for the aerodynamic optimization of a spiked blunt body in supersonic flow. Several machine learning techniques, including Gradient Boosting and LightGBM are employed as surrogate models and optimized through hyperparameter tuning. These surrogates are coupled with multiple evolutionary algorithms—namely the Genetic Algorithm (GA), Differential Evolution (DE), and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES)—to solve the inverse design problem of identifying geometric parameters that minimize aerodynamic drag at a fixed Mach number.

2. Materials and Methods

The methodology is divided into three key stages: dataset generation through CFD, training and validation of ML models, and design optimization using surrogate models for a fixed condition. Figure 1 presents a roadmap of methodology used in this work.

 doi:10.20944/preprints202602.1777.v1

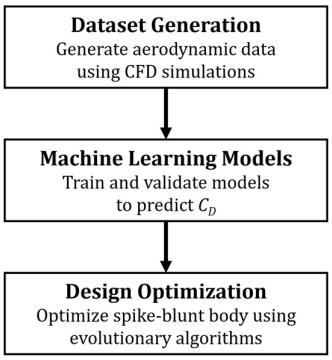

Figure 1. Methodological workflow of the study. Aerodynamic data are generated via CFD simulations. Machine learning models are then trained and validated to predict the drag coefficient $\boldsymbol{C_D}$. Finally, the spike–blunt body geometry is optimized using evolutionary algorithms.

In next sections are described, CFD dataset generation.

It is important to mention that machine learning models and the evolutionary algorithms used in this work use Python libraries were employed to ensure efficiency and reproducibility. For machine learning, the Scikit-learn library was used to implement models such as Gradient Boosting and LightGBM. For evolutionary algorithms, DEAP (Distributed Evolutionary Algorithms in Python) and pycma were used to run Genetic Algorithms, Differential Evolution, and CMA-ES, respectively.

2.1. Dataset Generation

The process for obtaining aerodynamic a non-dimensional dataset is presented in [25] using information from [26], (see appendix) , a flat-face spike blunt body is analyzed considering diameter (d) and length (l) and a blunt body diameter (D) of 1 inch; see Figure 2. These parameters are treated as dimensionless.

Simulation was carried out using FloWorks module integrated into SolidWorks 2023 CAD software. The compressible Navier–Stokes equations, including continuity, momentum, and energy conservation, were solved using the Finite Volume Method (FVM) in a Cartesian coordinate system [27]. Solver employs structured Cartesian mesh, and geometric bodies implemented using the cut-cell method[28], which allows mesh generation independent of complex geometrical features.Turbulent flow behavior was modeled using the Reynolds-Averaged Navier–Stokes (RANS) equations with a modified k–ε turbulence model that incorporates the Lam and Bremhorst damping function to accurately capture laminar, transitional, and turbulent regimes of homogeneous fluids.

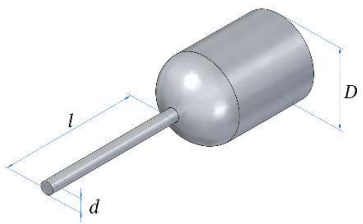

Figure 2. Blunt body scheme with main dimensions.

Validation process was carried out with a smooth sphere model at different Ma numbers as well as spiked blunt body [25]. A database with 230 registers were obtained considering drag coefficient as function of Mach number, $\boldsymbol{L/D}$ and $\boldsymbol{d/D}$.All simulations were performed on the same computer (Dell Precision 7530 wkst intel core Xeon e-2176m/memory 16 GB).

 doi:10.20944/preprints202602.1777.v1

2.2. Machine Learning Models

To predict Cd from the geometric and flow parameters, several supervised ML regression models were developed. These included Gradient Boosting Regressor (GBR) and Light Gradient Boosting Machine (LightGBM).

Gradient Boosting Regressor (GBR) and Light Gradient Boosting Machine (LightGBM) were selected as surrogate models due to their strong performance in nonlinear regression problems under limited data availability, which is a common constraint in computational fluid dynamics (CFD) studies where each simulation is computationally expensive. Unlike deep learning approaches, which typically require large datasets to generalize effectively, boosting-based ensemble methods can achieve high predictive accuracy with relatively small sample sizes by sequentially correcting residual errors and capturing complex interactions among input parameters.

The performance of the prediction model was evaluated using the coefficient of determination (R^2) and the root mean square error (RMSE). The R2 and RMSE were calculated as shown in Equations (2) and (3):

$$R^2 = 1 - \frac{\sum_1^N (y_p - y_0)^2}{\sum_1^N (\bar{y} - y_0)^2}$$

$$RSME = \sqrt{\frac{\sum_1^N (y_p - y_o)^2}{N}}$$

where N is the sample size, y_p is the predicted value, y_0 is the test observation and $\bar{y}$ is the average of y_0.

In next sections a brief description of machine learning methods.

2.2.1. Gradient Boosting Regressor (GBR)

Gradient boosting regressor (GBR) is a sequential ensemble learning method that constructs a predictive model by combining multiple weal learners [33–36], typically decision trees. In this approach, the prediction at iteration *m*, denoted as $F_m(x)$, is obtained by adding a new tree $h_m(x)$ to the previous ensemble:

$$F_m(x) = F_{m-1}(x) + \eta h_m(x)$$

Where η is the learning rate controlling the contribution of each tree. The weak learner $h_m(x)$ is trained to fit negative gradient of the loss function $L(y, F(x))$ with respect to the current predictions:

$$r_i^{(m)} = \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1(x)}}$$

$$h_m(x) \approx argmin_h \sum_{i=1}^{N} \left(r_i^{(m)} - h(x_i)\right)^2$$

The final ensemble prediction after M iterations is expressed as:

$$\hat{y}(x) = F_M(x) = F_0(x) + \sum_{m=1}^{M} \eta h_m(x)$$

Where $F_0(x)$ is the initial prediction, often the meaning of the target variable in regression task. GBR is highly effective at modelling complex, non-linear relationships and achieves proper performance when hyperparameters-such as the number of estimators *M*, learning rate η, and maximum tree depth-are properly tuned. This makes it particularly suitable for regression tasks involving structured tabular data.

 doi:10.20944/preprints202602.1777.v1

2.2.2. Light Gradient Boosting Regressor (LGBR)

LightGBM is a gradient boosting framework optimized for high efficiency and scalability [37,39], particularly in large-scale machine learning tasks. Like traditional Gradient Boosting, it constructs an additive model of M weak learners (decision trees), where the prediction at iteration m is updated as:

$$F_m(x) = F_{m-1}(x) + \eta h_m(x)$$

with η representing the learning rate and $h_m(x)$ the newly fitted tree. LightGBM employs a histogram-based algorithm that discretizes continuous feature values into k bins, reducing memory usage and accelerating the computation of split gains. The optimal split for a feature j is determined by maximizing the information gain:

$$Gain(j,s) = \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}$$

where G and H are the first- and second-order gradients of the loss function with respect to the predictions,λ is the L_2 regularization parameter, and γ controls the minimum split gain.

A key innovation of LightGBM is its leaf-wise growth strategy, where each tree expands by splitting the leaf that yields the highest gain, rather than level-wise as in conventional gradient boosting. While this strategy enhances training efficiency and model expressiveness, it may increase the risk of overfitting on small datasets. Proper tuning of hyperparameters—such as maximum depth, number of leaves, and learning rate—ensures that LightGBM achieves predictive performance comparable to or exceeding traditional gradient boosting methods, while offering significantly faster training for large and high-dimensional datasets.

2.3. Design Optimization via Evolutionary Algorithms

Once the surrogate model was trained and validated, it was employed to solve the inverse design problem, which consists of identifying the optimal pair of geometric parameters (*L*/*D*, *d*/*D*) that minimizes the predicted drag coefficient *Cd* for a fixed Mach number *Ma*. Formally, the optimization problem can be expressed as:

$$min_x C_d(x) \ \ subject\ to \ \ x = \left[\frac{L}{D}, \frac{d}{D}\right]$$

Where $C_d(x)$ is evaluated through the surrogate model. Given the non-linear and multi-dimensional nature of the problem, three evolutionary algorithms were implemented: Genetic Algorithm (GA), Differential Evolution (DE), and Covariance Matrix Adaptation Evolution Strategy (CMA-ES).

Each algorithm was executed over a bounded search space derived from the CFD study: L/D ∈ [0.15, 1.95] and d/D ∈ [0.06, 0.18] and Ma ∈ [1.2, 3.6]. The ML surrogate model was embedded as the fitness function to evaluate candidate solutions iteratively across generations. Convergence behavior, optimal Cd values, and computation time were tracked and compared across the algorithms.

2.3.1. Genetic Algorith (GA)

The Genetic Algorithm (GA) is a population-based stochastic optimization method inspired by the principles of natural selection and biological evolution [40,41]. A GA operates on a population of P candidate solutions (chromosomes) $x_i^{(t)} \in R^n$ at generation t_i, where each candidate represents a potential solution to the optimization problem. Everyone is evaluated using a fitness function $f(x)$, which quantifies the quality of the solution:

$$f_i^{(t)} = f\left(x_i^{(t)}\right), i = 1,2, \dots \dots, P$$

The evolutionary cycle of GA comprises three main operators:

 doi:10.20944/preprints202602.1777.v1

- Selection: Individuals with higher fitness are probabilistically favored for reproduction. A common approach is the roulette-wheel selection, where the probability of selecting an individual is proportional to its fitness:

$$P_i^{(t)} = \frac{f_i^{(t)}}{\sum_{j=1}^{P} f_j^{(t)}}$$

- Crossover (Recombination): Two parent solutions x_p and x_q are combined to produce offspring x_{child} A simple arithmetic crossover for continuous variables is:

$$x_{child} = \alpha x_p + (1-\alpha)x_q, \quad \alpha \in [0,1]$$

- Mutation: A small random perturbation is applied to offspring to maintain diversity and explore new regions of the search space:

$$x_{mutated} = x_{child} + \epsilon, \epsilon \sim \mathbb{N}(0, \sigma^2)$$

After generating a new population through these operations, the process iterates for generations or until a convergence criterion is met. The best solution at generation *t* is tracked as:

$$x^* = \arg max_{x_i^{(t)}} f(x_i^{(t)})$$

Through successive generations, GA converges towards optimal or near-optimal solutions, making it a flexible and widely adopted approach for solving complex, multi-modal, and non-differentiable optimization problems in engineering and design applications.

2.3.2. Differential Evaluation (DE)

Differential Evolution (DE) is a robust and efficient population-based evolutionary algorithm designed for solving continuous optimization problems [42,43]. It operates through the differential mutation of individuals and combines exploration and exploitation effectively.

Each generation *g*, for each target vector $x_i^{(g)} \in \mathbb{R}^d$ in the population, a donor vector $v_i^{(g)}$ is generated using the differential mutation rule:

$$v_i^{(g)} = x_{r1}^{(g)} + F(x_{r1}^{g} - x_{r3}^{(g)})$$

Where:

- $x_{r1}^{(g)}, x_{r2}^{(g)}, x_{r3}^{(g)}$ are three distinct vectors randomly selected from the current population, different from $x_i^{(g)}$.
- $F \in [0,2]$ is scaling factor controlling the amplification of the differential variation.

Next, crossover is applied to produce a trial vector u_o^g binomial crossover:

$$u_{ij}^{(g)} = \begin{cases} v_{ij}^{(g)} & if\ rand_i \le C_r\ or\ j = j_{rand} \\ x_{ij}^{(g)} & otherwise \end{cases}$$

Where:

- $C_r \in [0,1]$ is the crossover probability,
- j_{rand} is the randomly chosen index to ensure that at least one component comes from the donor vector.

Finally, selection is applied based on the fitness function $f(\cdot)$:

$$x_i^{(g+1)} = \begin{cases} u_i^{(g)} & if\ f(u_i^{(g)}) \le f(x_i^{(g)}) \\ x_i^{(g)} & otherwise \end{cases}$$

This process repeats until a stopping criterion is met (e.g., a maximum number of generations or convergence threshold). Due to its few control parameters and strong global search capabilities, DE

 doi:10.20944/preprints202602.1777.v1

is well-suited for real-valued, non-linear, and non-convex optimization problems with limited computational budgets.

2.3.3. Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is a population-based evolutionary strategy that focuses on adapting the search distribution to solve continuous optimization problems [44,45]. The main idea is to sample candidate solutions from a multivariate normal distribution, whose mean and covariance matrix are iteratively updated to reflect the most promising search directions.

- Initialization

The algorithm starts by defining an initial distribution:

  - Initial mean: $m^{(0)} \in \mathbb{R}^n$
  - Initial standard deviation: $\sigma^{(0)} > 0$.
  - Initial covariance matrix: $C^{(0)} = I_n$ $(Identity\ matrix)$.
  - Population size: λ

- Sampling of individuals

At generation g, λ candidate solutions are sampled as:

$$x_k^{(g)} = m^{(g)} + \sigma^{(g)} \cdot N_k^{(g)}, \qquad where\ N_k^{(g)} \sim \mathbb{N}(0, C^{(g)})$$

That is, each individual is sampled from a multivariate Gaussian distribution centered at the current mean $m^{(g)}$.

- Evaluation and selection

Each candidate solution is evaluated using the objective function $f\left(x_k^{(g)}\right)$, and the top μ individuals with the best fitness values are selected:

$$x_{1:\lambda}^{(g)}, x_{2:\lambda}^{(g)} \dots\dots\dots\dots x_{\mu:\lambda}^{(g)}$$

- Update of the mean

The new mean is computed as a weighted sum of the selected top individual:

$$m^{(g+1)} = \sum_{i=1}^{\mu} \omega_i \cdot x_{i:\lambda}^{(g)}, \quad where \quad \sum \omega_i = 1$$

The weighs ω_i typically satisfy $\omega_1 > \omega_2 > \cdots > \omega_\mu$ assigning more importance to better-performing individuals.

- Evolution paths

Two evolution paths are updated:

  - Step-size control path: p_σ
  - Covariance matrix adaptation path: p_C

The paths help accumulate information about the direction and magnitude of the search over generations and influence subsequent adaptation steps.

- Covariance matrix update

The covariance matrix is updated to capture the shape and orientation of search landscape:

$$C^{(g+1)} = \left(1 + c_1 + c_\mu\right) \cdot C^{(g)} + c_1 \cdot p_c^{(g+1)} (p_c^{(g+1)})^T + C_\mu \sum_{i=1}^{\mu} \omega_i \cdot y_i^{(g)} (y_i^{(g)})^T$$

Where

$$y_i^{(g)} = \frac{x_{i:\lambda}^{(g)} - m^{(g)}}{\sigma^{(g)}}$$

 doi:10.20944/preprints202602.1777.v1

This update combines the rank-one and rank-μ updates based on evolution paths and individual contributions.

- Step size adaptation

The global step size σ is adapted based on the length of the evolution path p_σ:

$$\sigma^{(g+1)} = \sigma^{(g)} \cdot exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|p_\sigma^{(g+1)}\|}{E[\|N(0,1)\|]} - 1\right)\right)$$

This mechanism allows dynamic control of the search radius, promoting global exploration or local refinement depending on the search progress.

3. Results and Discussion

This section presents the results of simulations for drag coefficient for different L/D, d/D and fixed Mach numbers. Figure 3 shows heatmaps of the aerodynamic drag coefficient (Cd) at three different Mach numbers (1.2, 2.4, and 3.2), while varying the geometric ratios L/D and d/D. Cd values are color-mapped across a matrix of spike length ratios (L/D, y-axis) and spike diameter ratios (d/D, x-axis).

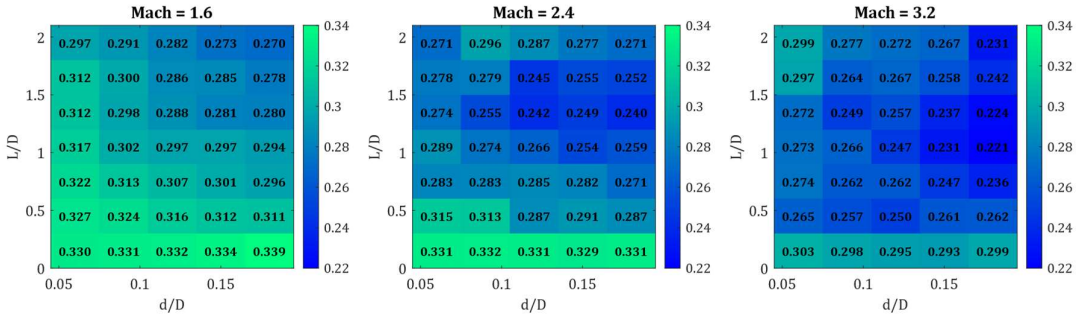

Figure 3. Heatmaps of drag coefficient (Cd) as a function of the spike length ratio (L/D) and spike diameter ratio (d/D) for three different Mach numbers: (a) Ma = 1.2, (b) Ma = 2.4, and (c) Ma = 3.2. The color scale indicates the magnitude of Cd, with cooler colors representing lower drag values. These plots highlight the influence of geometric parameters on aerodynamic performance, showing a consistent trend of drag reduction with increasing spike length across all Mach regimes.

For a fixed Mach number, there is a clear tendency for the drag coefficient (C_d) to decrease as the spike length ratio (L/D) increases. Longer spikes, corresponding to larger L/D values, consistently show lower drag, particularly for intermediate and high spike diameter ratios (d/D). The influence of the spike diameter ratio, however, appears to be more subtle. Although its effect is less pronounced than that of L/D, very small or very large diameter ratios do not consistently yield the lowest C_d values. Instead, optimal aerodynamic configurations tend to occur at intermediate d/D values (approximately 0.09–0.12), especially at higher Mach numbers.

The overall trend indicates a progressive reduction in Cd values as the Mach number increases, which is consistent with the literature. These findings align with prior research [46–51] on spike effectiveness in shock wave manipulation and drag reduction for blunt bodies.

This visualization supports that increasing the spike length is more effective than increasing the diameter in reducing drag, and that optimal performance shifts slightly depending on the Mach regime.

3.1. Machine Learning Models Performance

Table 1 presents the results of optimized Gradient Boosting Regressor (GBR) yielded the highest performance metrics, with a cross-validated R^2 of 0.8553, a test R^2 of 0.8909, and a test RMSE of

 doi:10.20944/preprints202602.1777.v1

0.00775. These results were obtained after tuning key hyperparameters using GridSearchCV, with the best configuration identified as: 100 estimators, maximum tree depth of 3, and a learning rate of 0.2. The relatively shallow tree depth helped prevent overfitting, while the moderately high learning rate ensured fast convergence.

In the case LGBM model, while also demonstrating good generalization capability, showed slightly lower performance with a test R^2 of 0.8459 and RMSE of 0.00922. The optimal hyperparameters included 100 estimators, maximum depth of 3, learning rate of 0.1, and 20 leaves per tree. Although LGBM benefits from its histogram-based training and leaf-wise tree growth—enabling faster training and scalability—these features may become limiting when the dataset is small, and the numerical precision of the features is critical. The discretization process inherent to histogram-based algorithms can potentially reduce the model's sensitivity to subtle variations in the input space.

Table 1. Performance comparison between optimized machine learning models.

Model	CV R^2 (mean)	Test R^2	Test RMSE	Optimized Parameters
Gradient Boosting	0.8553	0.8909	0.00775	n_estimators=100 max_depth=3 learning_rate=0.2
Light GBM	0.7252	0.8459	0.00922	n_estimators=100 max_depth=3 learning_rate=0.1 num_leaves=20

Figure 5 presents a comparison between the predicted and CFD generated values of the drag coefficient (C_d) for two optimized machine learning models: Gradient Boosting Regressor and LGBM. The scatter points represent the predictions made by Gradient Boosting and LGBM, respectively, plotted against the real values obtained from CFD simulations. The red dashed line corresponds to the ideal case where predicted values match the actual values perfectly. As observed, both models exhibit a strong correlation with the CFD data, with the Gradient Boosting model showing slightly higher accuracy. The proximity of the points to the ideal line indicates that both models can capture the underlying physical relationships in the dataset, with Gradient Boosting offering more precise predictions across the range of observed (C_d) values.

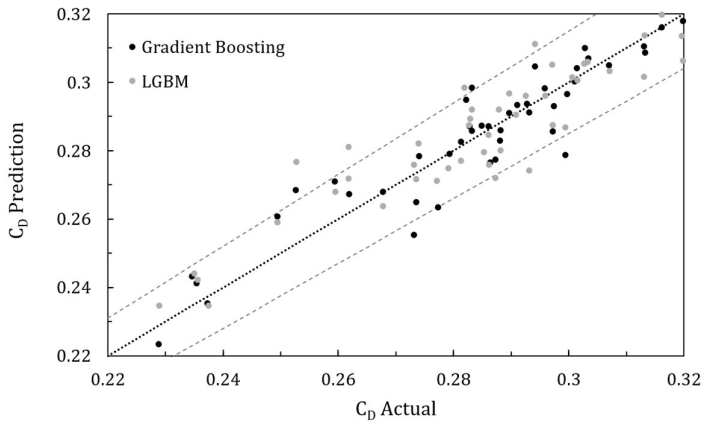

Figure 5. Computed drag coefficient vs. predicted drag coefficient for gradient boosting and LGBM.

 doi:10.20944/preprints202602.1777.v1

In summary, the GBR model demonstrated superior accuracy and consistency in comparation with LGBM. LGBM is an excellent alternative for cases where computational efficiency is prioritized, but some predictive power may be sacrificed.

3.2. Optimization Models Performance

Once the best machine learning model is selected, it is necessary to evaluate and compare the performance of the evolutionary algorithms under study, a fixed Mach number approach was adopted. For each selected Mach value within the supersonic range (from 1.2 to 3.6), the optimization problem was defined as minimizing the drag coefficient C_D by adjusting the spike geometry, specifically the parameters d/D and L/D. These parameters were bounded based on previous design constraints: $d/D \in [0.06,0.18]$ and $L/D \in [0.15,1.95]$. For optimization, Gradient Boosting Regressor will be used as a surrogate model, three optimization models were selected, Differential Evolution (DE), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and Genetic Algorithm (GA), the combination of geometrical parameters are ser for minimized C_D for the given flow condition.

Figure 5 presents the resulting minimum drag coefficients obtained by each algorithm across the range of Mach numbers. The curves show a general decreasing trend in Cd with increasing Mach number, reflecting the aerodynamic behavior of spike-blunt configurations at higher speeds. Notably, while all algorithms yielded consistent values for most cases, CMA-ES showed a slight deviation at Mach 1.6, suggesting possible convergence to a local minimum. Overall, the results confirm that all three algorithms can effectively explore the design space and provide nearly identical optimal solutions in terms of aerodynamic performance.

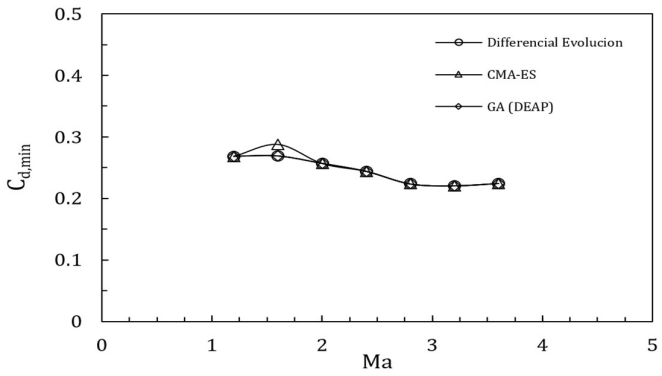

Figure 5. Minimum drag coefficients obtained by each algorithm across the range of Mach numbers.

 doi:10.20944/preprints202602.1777.v1

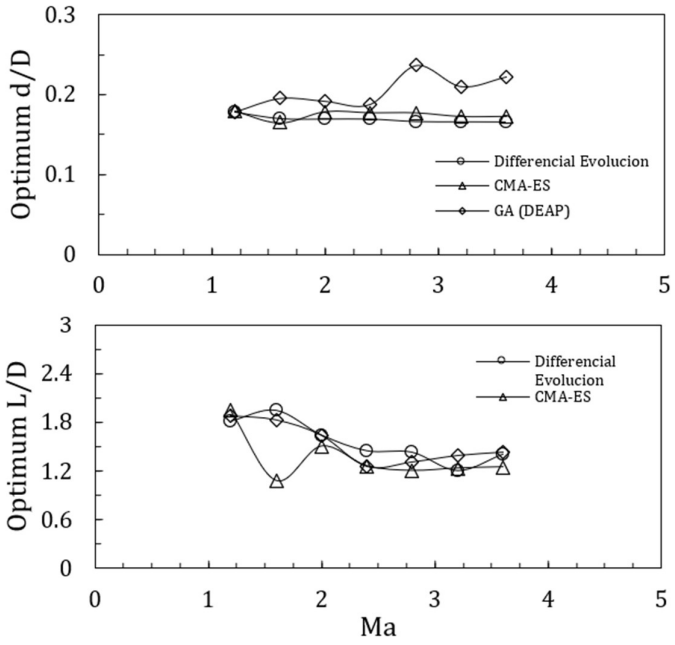

Figure 6. The figure illustrates the evolution of the optimal geometric ratios $\boldsymbol{d/D}$ (relative spike diameter) and $\boldsymbol{L/D}$ relative spike length) as a function of the Mach number for three different evolutionary algorithms: Differential Evolution (DE), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and Genetic Algorithm (GA, implemented via DEAP).

In the left plot, the behavior of the optimal d/D values are shown as Mach increases. Overall, DE and CMA-ES maintain relatively stable values around 0.17–0.18 across all speeds, whereas the GA algorithm exhibits more variability, particularly at higher Mach numbers. This greater fluctuation may reflect the sensitivity of GA to initial conditions or slower convergence compared to the other algorithms.

The right plot shows how the optimal L/D values evolve with increasing Mach number. A general decreasing trend is observed, indicating that shorter spikes relative to the main body become more effective in minimizing drag as flow speed increases. CMA-ES demonstrates more compact and consistent solutions, while GA again shows a wider spread, especially at higher Mach values.

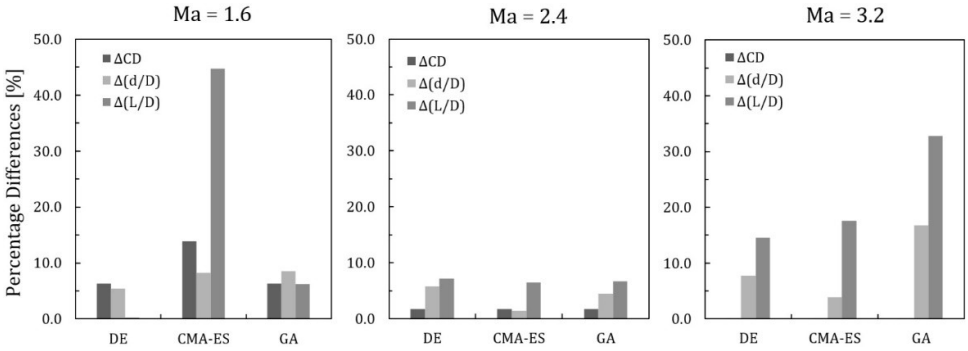

Figure 7. Comparison of percentage differences in the optimized aerodynamic parameters (ΔC_d, $\Delta(d/D)$, and $\Delta(L/D)$) for the *Differential Evolution*, *CMA-ES*, and *GA (DEAP)* algorithms at different Mach numbers.

For $Ma = 1.6$, the *CMA-ES* algorithm exhibits the largest variations, particularly in the L/D ratio, indicating less geometric convergence stability at lower supersonic speeds. In contrast, *Differential Evolution* and *GA (DEAP)* show more consistent results, with differences below 10%.

 doi:10.20944/preprints202602.1777.v1

At $Ma = 2.4$, discrepancies among the three algorithms decrease significantly, with all variations remaining below 8%, suggesting a more robust and stable convergence within this regime.

Finally, at $Ma = 3.2$, the differences increase again, especially for *GA (DEAP)*, which shows a pronounced rise in $\Delta(L/D)$, likely due to higher geometric sensitivity in hypersonic flow conditions. Overall, *Differential Evolution* demonstrates the most stable performance across all evaluated Mach numbers.

3.3. SHAP (SHapley Additive exPlanations)

The comparative analysis of optimization algorithms across different Mach regimes revealed consistent variations in the optimal geometric ratios and drag coefficients, highlighting the relative sensitivity of each parameter within the design space. To further quantify the influence of these input features on the predictive model, the SHAP (SHapley Additive exPlanations) analysis was conducted [52,53]. Figure 8 presents the corresponding SHAP summary plot, which provides a feature-level interpretation of the gradient boosting model, identifying the parameters that most strongly contribute to variations in the predicted drag coefficient.

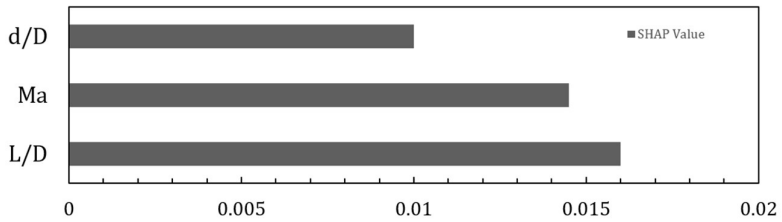

Figure 8. SHAP summary plot (bar) showing the average absolute SHAP values for each input feature. The L/D ratio exhibits the highest impact on the predicted drag coefficient, followed by the Mach number (Ma) and the d/D ratio.

Figure 8 presents the SHAP summary bar plot, which displays the average absolute SHAP value for each input feature, representing their mean impact on the model's output magnitude. Among the three nondimensional input parameters, L/D (length-to-diameter ratio) shows the highest influence on the predicted drag coefficient (Cd), followed closely by the Mach number (Ma). The parameter d/D (displacement-to-diameter ratio) has comparatively less impact. This indicates that geometric proportions, particularly L/D, play a more critical role in determining the aerodynamic behavior modeled by the gradient boosting algorithm. These insights help prioritize the most influential design variables in future analyses and optimization processes.

4. Conclusions

This study combined computational fluid dynamics (CFD), machine learning, and evolutionary optimization to investigate drag reduction mechanisms in spike-blunt configurations under supersonic flow conditions. The main findings can be summarized as follows:

1. Aerodynamic trends: Increasing the spike length ratio (L/D) leads to a consistent reduction in drag coefficient (Cd) across all Mach numbers. The diameter ratio (d/D) exerts a secondary influence, with optimal configurations occurring at intermediate values (≈0.09–0.12). Higher Mach numbers result in lower overall drag, confirming the stabilizing aerodynamic effects of elongated spikes at supersonic regimes.
2. Machine learning performance: Among the tested regression models, the Gradient Boosting Regressor (GBR) achieved the best predictive accuracy (R^2 = 0.8909, RMSE = 0.00775), outperforming LightGBM. The relatively shallow tree depth and optimized learning rate enabled the GBR to effectively capture nonlinear relationships in the aerodynamic dataset without overfitting.

 doi:10.20944/preprints202602.1777.v1

3. Optimization robustness: All three evolutionary algorithms—Differential Evolution (DE), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and Genetic Algorithm (GA)—successfully identified near-optimal geometries for drag minimization. DE demonstrated the most stable convergence across Mach regimes, whereas CMA-ES showed slightly higher variability at low Mach numbers and GA exhibited increased dispersion at high Mach numbers.
4. Feature importance (SHAP analysis): The SHAP results revealed that the spike length ratio (L/D) is the most influential parameter in determining the drag coefficient, followed by the Mach number (Ma) and the spike diameter ratio (d/D). This highlights the predominance of geometric slenderness over diameter effects in supersonic drag behavior.

Overall, the integration of machine learning with evolutionary algorithms proved to be an efficient framework for aerodynamic optimization. The results not only validate the predictive capability of surrogate modeling approaches but also provide physical insights into the dominant geometric factors influencing drag reduction. Future work will focus on extending the methodology to three-dimensional configurations and transient flow regimes to further generalize the optimization framework.

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

Funding: The APC was funded by Tecnologico de Monterrey, Vicerrectory of Research and Technology Transfer.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: During the preparation of this manuscript, the authors used OpenAI ChatGPT (GPT-5.2) for language review, clarity improvement, and verification of citation formatting. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

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

Appendix. Dataset Used for Prediction

d/D	l/D	CD comp Ma = 1.2	CD comp Ma = 1.6	CD comp Ma = 2.0	CD comp Ma = 2.4	CD comp Ma = 2.8	CD comp Ma = 3.2	CD comp Ma = 3.6
0.06	0.15	0.3304	0.353	0.348	0.3312	0.3459	0.3032	0.3229
0.06	0.45	0.3268	0.3442	0.333	0.3145	0.2887	0.2645	0.321
0.06	0.75	0.3217	0.3291	0.3088	0.2834	0.2828	0.2738	0.322
0.06	1.05	0.3165	0.3197	0.3133	0.2891	0.2749	0.2728	0.3027
0.06	1.35	0.3117	0.3213	0.3016	0.274	0.28	0.2719	0.3139
0.06	1.65	0.3122	0.3053	0.3028	0.2783	0.299	0.2973	0.3084
0.06	1.95	0.2971	0.2863	0.33	0.2711	0.2861	0.2993	0.3162
0.09	0.15	0.3307	0.3533	0.3483	0.3321	0.2992	0.2976	0.329
0.09	0.45	0.3236	0.3358	0.3213	0.313	0.3322	0.2569	0.3198
0.09	0.75	0.313	0.3152	0.2941	0.283	0.2714	0.2621	0.3133
0.09	1.05	0.3016	0.3007	0.2827	0.2744	0.2593	0.2658	0.2831
0.09	1.35	0.298	0.2898	0.2752	0.2554	0.267	0.2495	0.3072
0.09	1.65	0.3005	0.2913	0.2818	0.279	0.2862	0.2638	0.2898
0.09	1.95	0.2915	0.2925	0.3229	0.2961	0.2783	0.2772	0.3098
0.12	0.15	0.3322	0.3541	0.3479	0.331	0.297	0.295	0.3311
0.12	0.45	0.3161	0.321	0.3077	0.287	0.2791	0.25	0.3162

 doi:10.20944/preprints202602.1777.v1

0.12	0.75	0.307	0.3042	0.2821	0.285	0.2632	0.2617	0.2996
0.12	1.05	0.2969	0.2895	0.2851	0.266	0.2478	0.2469	0.2824
0.12	1.35	0.2883	0.2828	0.2683	0.2422	0.2546	0.257	0.2367
0.12	1.65	0.2861	0.2902	0.2535	0.2455	0.2743	0.2673	0.2617
0.12	1.95	0.2821	0.2881	0.3087	0.2874	0.2776	0.2718	0.3012
0.15	0.15	0.3337	0.3549	0.3464	0.3294	0.2938	0.2927	0.33
0.15	0.45	0.3122	0.3166	0.3017	0.291	0.2703	0.2614	0.3122
0.15	0.75	0.3007	0.2995	0.2896	0.2823	0.2478	0.2474	0.2964
0.15	1.05	0.2975	0.2864	0.2785	0.2541	0.2349	0.2309	0.2774
0.15	1.35	0.2813	0.275	0.2643	0.249	0.2406	0.2373	0.2414
0.15	1.65	0.2854	0.2843	0.2737	0.255	0.2569	0.2584	0.2731
0.15	1.95	0.2734	0.2785	0.299	0.2771	0.2678	0.2669	0.2746
0.18	0.15	0.3393	0.3578	0.3454	0.3306	0.2926	0.2994	0.3227
0.18	0.45	0.311	0.3139	0.3012	0.2869	0.2671	0.2623	
0.18	0.75	0.2958	0.2941	0.2881	0.2705	0.2392	0.2356	
0.18	1.05	0.2938	0.2851	0.2843	0.259	0.2233	0.2211	
0.18	1.35	0.2798	0.27	0.2617	0.24	0.2289	0.2244	
0.18	1.65	0.2779	0.2733	0.2436	0.2516	0.242	0.2421	
0.18	1.95	0.2702	0.2528	0.2931	0.2714	0.2522	0.2314	

References

1. Anderson, J. D. (2003). Modern compressible flow: With historical perspective (3rd ed.). McGraw-Hill Education. https://doi.org/10.1063/1.2756072
2. Toro, E. F. (2009). Riemann solvers and numerical methods for fluid dynamics: A practical introduction (3rd ed.). Springer. https://doi.org/10.1007/b79761
3. Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., et al. (2013). High-order CFD methods: Current status and perspective. International Journal for Numerical Methods in Fluids, 72(8), 811–845. https://doi.org/10.1002/fld.3767
4. Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., & Mavriplis, D. (2014). CFD vision 2030 study: A path to revolutionary computational aerosciences. NASA Technical Report, NASA/CR-2014-218178. https://doi.org/10.2514/6.2014-218178
5. Jameson, A. (2003). Aerodynamic shape optimization using the adjoint method. VKI LS 2003-04.
6. Halila, G. L. O., Ghazlane, I., Carrier, G., & Trepanier, J. Y. (2020). Adjoint-based aerodynamic shape optimization for compressible flows: A review. Aerospace Science and Technology, 106, 106112. https://doi.org/10.1016/j.ast.2020.106112
7. Mukhtar, A., Yasir, A. S. H. M., & Nasir, M. F. M. (2023). A machine learning-based comparative analysis of surrogate models for design optimisation in computational fluid dynamics. Heliyon, 9(8), e18674. https://doi.org/10.1016/j.heliyon.2023.e18674
8. Alizadeh, R., Allen, J. K., & Mistree, F. (2020). Managing computational complexity using surrogate models: a critical review. Research in Engineering Design, 31(3), 275–298. https://doi.org/10.1007/s00163-020-00336-7
9. Mao, R., Lan, Y., Liang, L., Yu, T., Mu, M., Leng, W., & Long, Z. (2025). Rapid CFD prediction based on Machine Learning surrogate model in built Environment: a review. Fluids, 10(8), 193. https://doi.org/10.3390/fluids10080193
10. Kirsch, J., Rider, W. J., & Fathi, N. (2024). Credibility Assessment of Machine Learning-Based Surrogate Model Predictions on NACA 0012 Airfoil Flow. https://doi.org/10.1115/vvuq2024-132964
11. Demystifying the Data Need of ML-surrogates for CFD Simulations. (2022). https://doi.org/10.48550/arxiv.2205.08355
12. Joly, M. M., Sarkar, S., & Mehta, D. (2018). Machine Learning Enabled Adaptive Optimization of a Transonic Compressor Rotor With Pre-Compression. Journal of Turbomachinery, 141(5), 051011. https://doi.org/10.1115/GT2018-77098

 doi:10.20944/preprints202602.1777.v1

13. El Zaatari, L., Leicht, T., Lang, S., Bekemeyer, P., & Görtz, S. (2022). A Machine Learning based Expert System for Optimizing CFD Solver Parameters. ECCOMAS 2022. https://doi.org/10.23967/eccomas.2022.013
14. Owoyele, O., Pal, P., Vidal Torreira, A., et al. (2021). AutoML-GA coupled with CFD for engine design optimization. International Journal of Engine Research, 23(9), 1586-1601. https://doi.org/10.1177/146808742110234
15. Niu, Y., Zhao, K., Yang, Y., Yao, M., Wu, Q., Bai, B., & Ma, L. (2024). Integration of deep learning and CFD for aerodynamic force prediction. Physics of Fluids, 36(10). https://doi.org/10.1063/5.0232956
16. Hwang, P., Wu, J., & Chang, Y. (2024). CFD and ML optimization for diffuser-augmented wind turbines. Sustainability, 16(9), 3648. https://doi.org/10.3390/su16093648
17. Cabrera-Escobar, J., Vera, D., Jurado, F., et al. (2024). CFD and deep learning for process optimization. Energy Sources Part A, 46(1), 4756-4776. https://doi.org/10.1080/15567036.2024.2331563
18. Wang, Y., Guzik, S., Zupanski, M., & Gao, X. (2021). Ensemble filter for computational flame and fluid dynamics. IMA Journal of Applied Mathematics, 86(4), 631-661. https://doi.org/10.1093/imamat/hxab010
19. Stavrakakis, G., Karadimou, D., Zervas, P., Sarimveis, H., & Markatos, N. (2010). CFD and neural networks for optimization. Building and Environment, 46(2), 298-314. https://doi.org/10.1016/j.buildenv.2010.07.021
20. Crawford, D. H. (1961). Flow over spiked-nose hemispherical cylinders. NASA TN D-118.
21. Feszty, D., Badcock, K. J., & Richards, B. E. (2004). Aerodynamic characteristics of spiked blunt bodies. AIAA Journal, 42(5), 879–887. https://doi.org/10.2514/1.12357
22. Panaras, A. G., & Drikakis, D. (2009). High-speed flows around spiked blunt bodies. Journal of Fluid Mechanics, 632, 69–96. https://doi.org/10.1017/S0022112009007088
23. Deng, F., Wen, C. Y., Tang, H., & Liu, J. (2017). Spike effects on drag reduction. Journal of Spacecraft and Rockets, 54(1), 36–47. https://doi.org/10.2514/1.A33567
24. Sarwar, M. D. G., Siddiqui, M. S., & Uddin, M. S. (2023). Aerodynamic performance of a spiked blunt body. Aerospace Science and Technology, 139, 108317. https://doi.org/10.1016/j.ast.2023.108317
25. Muñoz, J. A. S., Lagarza-Cortés, C., & Ramírez-Cruz, J. (2024). Drag coefficient prediction of a spike blunt body. Aerospace, 11(9), 757. https://doi.org/10.3390/aerospace11090757
26. Kalimuthu R, Mehta RC, Rathakrishnan E. Investigation of aerodynamic coefficients at Mach 6 over conical, hemispherical and flat-face spiked body. *The Aeronautical Journal*. 2017;121(1245):1711-1732. doi:10.1017/aer.2017.100
27. Numerical Basis of CAD-Embedded CFD: White Paper. Available online: https://www.solidworks.com/sw/docs/Flow_Basis_of_CAD_Embedded_CFD_Whitepaper.pdf (accessed on 1 June 2024).
28. Ingram, D., Causon, D., & Mingham, C. (2003). Developments in Cartesian cut cell methods. *Mathematics and Computers in Simulation*, *61*(3–6), 561–572. https://doi.org/10.1016/s0378-4754(02)00107-6
29. Beigzadeh, R., Soltanian, S., & Tofangchi, D. (2024). Enhancing heat transfer efficiency in heat exchangers: A fusion of computational fluid dynamics and genetic algorithm for circular baffle optimization. International Journal of Heat and Fluid Flow, 112, 109726. https://doi.org/10.1016/j.ijheatfluidflow.2024.109726
30. Yu, J., Tang, M., Cui, P., Deng, J., Mi, H., Zhang, L., Chen, L., Wu, P., Chao, Y., Zhu, W., & Liu, Z. (2024). Computational fluid dynamics and machine learning assisted Al-LDH adsorbent reactor design for lithium recovery from salt lakes. *Desalination*, *600*, 118396. https://doi.org/10.1016/j.desal.2024.118396
31. Barragan, G., Atarihuana, S., Cando, E., & Hidalgo, V. (2025). Enhancing hydraulic efficiency of pelton turbines through computational fluid dynamics and metaheuristic optimization. *Algorithms*, *18*(1), 35. https://doi.org/10.3390/a18010035
32. Samiani, O. S., & Boroushaki, M. (2024). Optimal design of Archimedes Wind Turbine using genetic algorithm. *Energy*, *314*, 134157. https://doi.org/10.1016/j.energy.2024.134157
33. Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2020). Comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, 54(3), 1937-1967. https://doi.org/10.1007/s10462-020-09896-5
34. Cha, G., Moon, H., & Kim, Y. (2021). Random Forest vs Gradient Boosting for prediction. IJERPH, 18(16), 8530. https://doi.org/10.3390/ijerph18168530

 doi:10.20944/preprints202602.1777.v1

35. Natekin A and Knoll A (2013) Gradient boosting machines, a tutorial. Front. Neurorobot. 7:21. doi: 10.3389/fnbot.2013.00021
36. GeeksforGeeks. (2025b, diciembre 3). Gradient Boosting in ML. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/ml-gradient-boosting/
37. A. A. Taha and S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," in *IEEE Access*, vol. 8, pp. 25579-25587, 2020, doi: 10.1109/ACCESS.2020.2971354.
38. Guo, J., Yun, S., Meng, Y., He, N., Ye, D., Zhao, Z., Jia, L., & Yang, L. (2023). Prediction of heating and cooling loads based on light gradient boosting machine algorithms. Building And Environment, 236, 110252. https://doi.org/10.1016/j.buildenv.2023.110252
39. GeeksforGeeks. (2025a, septiembre 5). Regression using LightGBM. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/regression-using-lightgbm/
40. Kramer, O. (2017). Genetic Algorithms. In: Genetic Algorithm Essentials. Studies in Computational Intelligence, vol 679. Springer, Cham. https://doi.org/10.1007/978-3-319-52156-5_2
41. PyGAD. (s. f.). *PyGAD documentation*. Read the Docs. https://pygad.readthedocs.io/en/latest/
42. Mayer, D., Kinghorn, B., & Archer, A. (2004). Differential evolution – an easy and efficient evolutionary algorithm for model optimisation. Agricultural Systems, 83(3), 315-328. https://doi.org/10.1016/j.agsy.2004.05.002
43. Cristina, S. (2021, 12 de octubre). *Differential evolution from scratch in Python*. Machine Learning Mastery. https://machinelearningmastery.com/differential-evolution-from-scratch-in-python/
44. Nikolaus Hansen, N., & Andreas Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation, 9(2), 159–195. https://doi.org/10.1162/106365601750190398
45. cma-es. (2021, 1 de diciembre). *cma-es (versión 1.5.0) [Paquete PyPI]*. Python Package Index. https://pypi.org/project/cma-es/
46. Esfeh, M.K.; Tajalli, S.M.; Liu, P. Evaluation of Aerospike for Drag Reduction on a Blunt Nose Using Experimental and Nu-merical Modeling. Acta Astronaut. 2019, 160, 656–671. https://doi.org/10.1016/j.actaastro.2019.03.010.
47. Tekure, V.; Pophali, P.S.; Venkatasubbaiah, K. Numerical Investigation of Aerospike Semi-Cone Angle and a Small Bump on the Spike Stem in Reducing the Aerodynamic Drag and Heating of Spiked Blunt-Body: New Correlations for Drag and Surface Temperature. Phys. Fluids 2021, 33, 116108. https://doi.org/10.1063/5.0066028.
48. Li, Z.; Sun, C.; Xia, X.; Li, X. Numerical Simulation of Aerodynamic Heating over Solid Blunt Configuration with Porous Spike. J. Aerosp. Eng. 2018, 31, 04018083. https://doi.org/10.1061/(ASCE)AS.1943-5525.0000901.
49. Asif, M.; Zahir, S.; Kamran, N.; Khan, M. Computational Investigations Aerodynamic Forces at Supersonic / Hypersonic Flow Past a Blunt Body with Various Forward Facing Spikes. In 22nd Applied Aerodynamics Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences, AIAA, Providence, Rhode Island, 16–19 August 2004.
50. Xue, Y.; Wang, L.; Fu, S. Drag Reduction and Aerodynamic Shape Optimization for Spike-Tipped Supersonic Blunt Nose. J. Spacecr. Rocket. 2018, 55, 552–560. https://doi.org/10.2514/1.A33621.
51. Hamza, M.; Khan, S.B.; Maqsood, A. Geometric Optimization of Blunt Bodies with Aerodisk and Opposing Jet for Wave Drag and Heat Reduction. Aerospace 2022, 9, 800. https://doi.org/10.3390/aerospace9120800.
52. Ponce-Bobadilla, A. V., Schmitt, V., Maier, C. S., Mensing, S., & Stodtmann, S. (2024). Practical guide to SHAP analysis: Explaining supervised machine learning model predictions in drug development. *Clinical And Translational Science*, *17*(11), e70056. https://doi.org/10.1111/cts.70056
53. Awan, A. A. (2023, 28 de junio). *An Introduction to SHAP values and machine learning interpretability*. DataCamp. https://www.datacamp.com/tutorial/introduction-to-shap-values-machine-learning-interpretability