

ENSEMBLE MACHINE LEARNING FOR UAV PROPELLER PERFORMANCE PREDICTION WITH UNCERTAINTY QUANTIFICATION

Mohammad Irfan Alam[1,2], Abrar H. Baluch[1,2] & Suhail Hyder Vattathurvalappil[2]

[1]Interdisciplinary Research Centre for Aviation and Space Exploration, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.
[2]Department of Aerospace Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

Abstract

Accurate propeller performance prediction is important for unmanned aerial vehicle (UAV) design and mission planning. Computational fluid dynamics (CFD) is accurate but expensive, while empirical models are fast but lack accuracy and give no estimate of reliability. This study presents a machine learning approach that combines gradient boosting ensembles with two uncertainty quantification methods for fast and accurate propeller performance prediction. We built a voting ensemble of XGBoost, LightGBM, and CatBoost models, trained on 30,188 experimental measurements from the UIUC propeller database covering more than 400 propeller geometries. A physics-informed feature set based on blade element theory gives 27 features that capture geometric, aerodynamic, and operational effects. The model reaches a coefficient of determination $R^2 = 0.9375$ for the thrust coefficient (C_T), $R^2 = 0.9217$ for the power coefficient (C_P), and $R^2 = 0.857$ for propulsive efficiency (η), with no measurable overfitting on a diameter-stratified hold-out set. A systematic comparison of 15 model configurations confirms that the ensemble is the best choice. Two uncertainty quantification methods, ensemble variance and quantile regression, describe the reliability of each prediction; the ensemble-variance 95% confidence intervals are well calibrated, with an empirical coverage of 94.7–95.1%. Typical prediction uncertainties are $\pm$1–3%, which allows uncertainty-aware design decisions. The model predicts about four orders of magnitude faster than CFD while keeping high accuracy, which supports rapid design iteration and propeller selection. Case studies show practical uses such as propeller selection, performance-curve generation, and uncertainty-informed design. To our knowledge, this is the first work to provide calibrated uncertainty estimates for data-driven propeller performance prediction, which is an important requirement for safety-critical UAV applications.

Keywords: Machine Learning, UAV Propellers, Uncertainty Quantification, Gradient Boosting, Surrogate Modeling, Ensemble Methods

1. Introduction

1.1 Background and Motivation

Unmanned aerial vehicles (UAVs) are now used widely in surveillance, delivery, agriculture, infrastructure inspection, and scientific work [1]. This has increased the need for efficient propulsion. The propeller is the central component of an electric UAV propulsion system: it converts shaft power into thrust, and its efficiency directly affects flight endurance, payload, and range. A small improvement in propeller selection can give a large gain in capability, while a poorly matched propeller can make an otherwise sound airframe impractical.

Predicting propeller performance, that is the thrust coefficient (C_T), power coefficient (C_P), and propulsive efficiency (η), across the full operating envelope is difficult [2]. Three classes of methods are in common use. First, high-fidelity computational fluid dynamics (CFD) resolves the flow around the rotating blades and gives accurate predictions [3]. However, it costs hours to days per operating point,

which makes it impractical for the wide design-space studies needed in early design. Second, lower-order analytical models based on blade element momentum theory (BEMT) [23] and vortex or panel methods [4] are fast, but they need detailed blade-geometry inputs (chord and twist distributions, airfoil polars) that are often not available for commercial off-the-shelf propellers, and their accuracy drops at the low Reynolds numbers ($Re \approx 5 \times 10^4$–4×10^5) typical of small UAVs. Third, wind-tunnel testing [5] gives ground-truth data but is slow and costly, and covers only a finite set of geometries and operating conditions.

These limitations motivate data-driven surrogate models that can give near-experimental accuracy at very low computational cost. Large experimental databases are now available, and machine learning (ML) for tabular regression has matured, so such surrogates are both practical and attractive. UAV design and mission planning, however, are increasingly safety-critical: a point prediction with no estimate of its reliability is of limited use when sizing battery margins or certifying endurance. A model that also reports how confident it is, through calibrated uncertainty bounds, allows risk-aware engineering decisions that point predictions alone cannot support. Recent work on machine learning for fluid mechanics has shown the potential of data-driven methods to complement CFD [6].

1.2 Related Work

Classical propeller design and analysis rests on momentum and blade element theory [23, 7], which couple a one-dimensional momentum balance with sectional aerodynamics to predict loads. Tools such as QPROP and XROTOR [8] remain widely used. Previous research [7, 9] extended BEMT to low-Reynolds-number small-UAS propulsion and highlighted the sensitivity of predictions to sectional airfoil data and Reynolds-number corrections, which are difficult to obtain for arbitrary commercial propellers.

The University of Illinois at Urbana-Champaign (UIUC) propeller database [11, 10] is the largest public source of small-scale propeller wind-tunnel data, spanning hundreds of geometries and a wide range of advance ratios and rotational speeds. Its breadth makes it a good basis for data-driven modeling, yet it is most often used as a lookup table rather than as the basis of a continuous, queryable predictive model.

Data-driven and ML methods are now widely used in fluid mechanics and aerospace [12, 13], including surrogate modeling, reduced-order modeling, and flow reconstruction. For tabular engineering data of the scale considered here, several studies have shown that tree-based gradient boosting methods match or beat deep neural networks while needing far less tuning and compute [14, 15]. Gradient boosting implementations such as XGBoost [16], LightGBM [17], and CatBoost [18], and their ensembles [19], are the standard tools for such problems.

The ML literature on uncertainty quantification (UQ) is large and includes deep ensembles [20], Bayesian methods, and quantile or distribution-free approaches [21], as surveyed by Abdar et al. [22]. Even so, UQ has not been applied in a systematic way to data-driven propeller performance prediction. Existing ML propeller studies usually report point accuracy only, and give the practitioner no principled measure of prediction reliability. This is the main gap addressed in this work.

1.3 Research Objectives and Contributions

The objectives of this study are to: (1) develop a high-accuracy ML model for predicting propeller C_T, C_P, and η from readily available geometry and operating parameters; (2) engineer a physics-informed feature set grounded in propeller aerodynamics; (3) systematically evaluate a broad set of model configurations to establish, rather than assume, the optimal architecture; (4) implement and validate two complementary uncertainty quantification methods that yield calibrated prediction intervals; and (5) demonstrate practical applicability through representative UAV propeller-selection case studies.

To the best of our knowledge, this is the first work to provide calibrated uncertainty estimates for data-driven propeller performance prediction. The specific contributions are:

1. We combine ensemble-variance (epistemic) and quantile-regression (combined epistemic/aleatoric) methods; the ensemble-variance 95% confidence intervals achieve 94.7–95.1% empirical coverage on held-out data.

2. We evaluate 15 distinct configurations in an AutoML-style comparison and show that the voting ensemble is the best choice rather than a lucky selection.

3. We train on 30,188 experimental measurements that span more than 400 geometries, one of the largest propeller datasets used for ML, which supports reliable interpolation across the sampled design envelope.

4. We derive 27 features from blade element and momentum theory, and validate their physical relevance through SHAP analysis.

5. The final model runs roughly four orders of magnitude faster than CFD on commodity hardware with no GPU, supporting rapid, uncertainty-aware design iteration.

The remainder of this paper is organized as follows. Section 2. describes the dataset, feature engineering, models, and uncertainty quantification methodology. Section 3. presents model comparison, feature importance, accuracy, uncertainty validation, and case studies. Section 4. interprets the results and discusses limitations, and Section 5. concludes.

2. Methodology

This section describes the full methodology, covering dataset preparation, physics-informed feature engineering, model development, and the two uncertainty quantification methods. Figure 1 gives an overview of the pipeline, from raw database processing through ensemble modeling to calibrated, uncertainty-aware predictions.

2.1 Dataset Description and Preparation

2.1.1 UIUC Propeller Performance Database

The UIUC propeller database [11, 10] provides experimental measurements for small-scale propellers typical of UAV applications. It contains wind tunnel test results for more than 400 propeller geometries from several manufacturers (APC, Master Airscrew, GWS, and others), tested in the UIUC low-speed subsonic wind tunnel.
Our analysis used 1,994 data files in three test categories:

- **Performance tests:** dynamic measurements at various advance ratios (J) and rotational speeds, providing thrust coefficient (C_T), power coefficient (C_P), and propulsive efficiency (η).
- **Static tests:** measurements at zero forward velocity ($J = 0$), capturing static thrust characteristics.
- **Geometry data:** blade chord and pitch angle distributions along the radius.

Table 1 summarizes the dataset. The combined dataset contains 30,188 experimental measurements that cover a wide range of operating conditions relevant to small UAVs. The geometry coverage and operating-condition distributions are shown in Figure 2.
The propeller diameter range of 5–18 inches covers typical small UAV applications including quadcopters, fixed-wing UAVs, and hybrid VTOL platforms. The advance ratio range $J = 0.1$–1.2 encompasses the complete operational envelope from static thrust to cruise flight conditions.

2.1.2 Data Processing and Quality Control

Raw data files were systematically parsed and processed through a multi-stage pipeline.
Step 1: Automated file parsing. Each filename encodes propeller geometry and test conditions following the convention `propname_diameterxpitch_testID_RPMrpm_testtype.txt`. A custom parser extracted metadata and associated performance measurements.
Step 2: Data cleaning. Quality control procedures removed physically unrealistic measurements: efficiency values outside $[-1, 1]$ were removed as unphysical; missing values in critical parameters (D, P, J, C_T, C_P) resulted in record exclusion; and outliers exceeding 3σ from the mean were flagged for inspection.

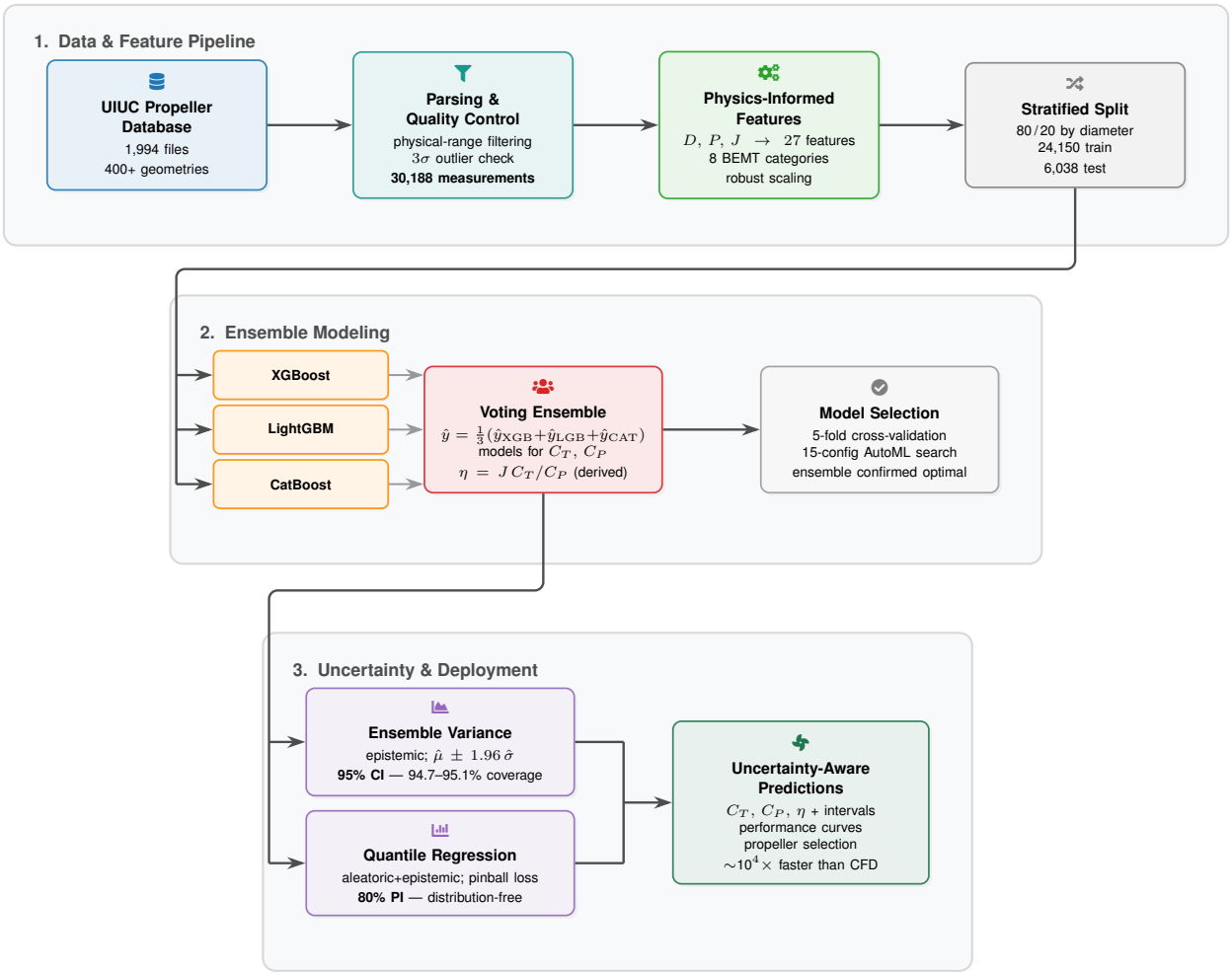


Figure 1 – Overview of the end-to-end methodology. Stage 1 parses and quality-controls the UIUC database and expands the three primary inputs (D, P, J) into 27 physics-informed features before a diameter-stratified split. Stage 2 trains three gradient-boosting base learners that are combined by simple averaging into a voting ensemble for C_T and C_P (with η derived), validated by 5-fold cross-validation and a 15-configuration AutoML search. Stage 3 adds two uncertainty quantification methods, ensemble variance (95% CI) and quantile regression (80% PI), to give calibrated, uncertainty-aware predictions.

Table 1 – UIUC propeller database summary statistics.

Category	Count	Range/Details
Total measurements	30,188	Performance + static tests
Unique propellers	400+	Multiple manufacturers
Diameter (D)	–	5.0–18.0 inches (127–457 mm)
Pitch (P)	–	2.5–14.0 inches (64–356 mm)
Advance ratio (J)	–	0 (static), 0.1–1.2 (dynamic)
Reynolds number (Re)	–	50,000–400,000
Test velocities	–	0–30 m/s
Rotational speeds	–	2,000–10,000 RPM
Performance records	26,011	Dynamic conditions
Static records	4,177	$J = 0$ conditions
Geometry records	2,364	Blade profiles

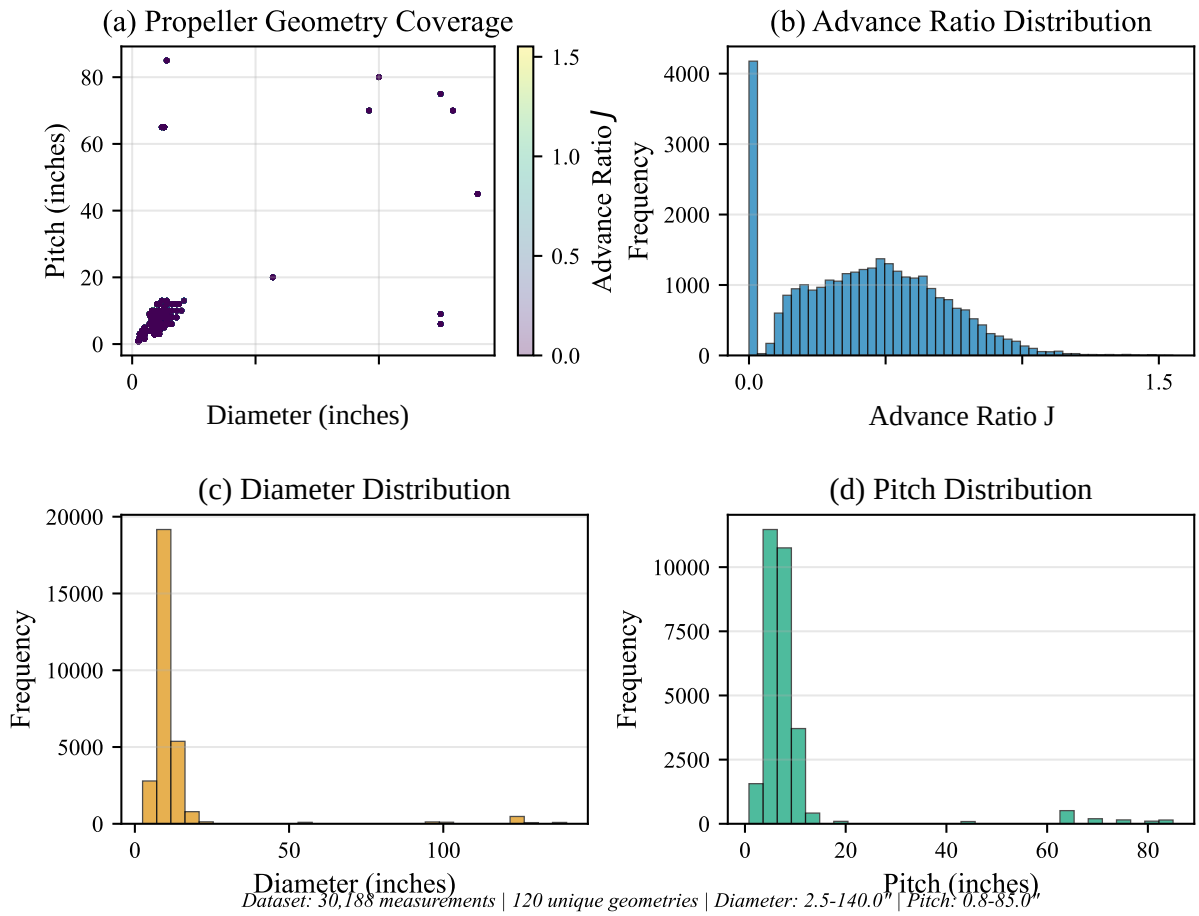


Figure 2 – Overview of the UIUC propeller dataset: (a) geometry coverage in the diameter–pitch plane colored by advance ratio, and (b–d) distributions of advance ratio, diameter, and pitch across the 30,188 measurements.

Step 3: Feature computation. All engineered features (Section 2.2) were calculated from basic parameters with appropriate handling of edge cases (e.g., $\log(J)$ with offset for small J values).
Step 4: Train–test splitting. The records were split into an 80% training set (24,150 samples) and a 20% held-out test set (6,038 samples), stratified by propeller diameter so that both sets span the full size range. The split is done at the level of individual measurements rather than whole propellers, so different operating points (advance ratios) of the same propeller can appear in both sets. The reported test metrics therefore measure interpolation within the sampled geometry envelope, not extrapolation to entirely unseen propeller designs; we make this scope explicit in Section 4.. The feature scaler and all model parameters were fitted on the training set only.

2.2 Physics-Informed Feature Engineering

Effective feature engineering is critical for machine learning model performance, particularly when incorporating domain knowledge from aerodynamic theory [30]. We developed 27 features organized into eight categories, each grounded in propeller aerodynamics and blade element momentum theory (BEMT). The complete framework is summarized in Figure 3.
The advance ratio, the principal dimensionless operating parameter, is defined as

$$J = \frac{V}{nD}, \tag{1}$$

where V is the freestream velocity [m/s], n is the rotational speed [rev/s], and D is the diameter [m]. The two predicted coefficients follow the standard propeller non-dimensionalization,

$$C_T = \frac{T}{\rho\, n^2 D^4}, \qquad C_P = \frac{P_{\text{sh}}}{\rho\, n^3 D^5}, \tag{2}$$

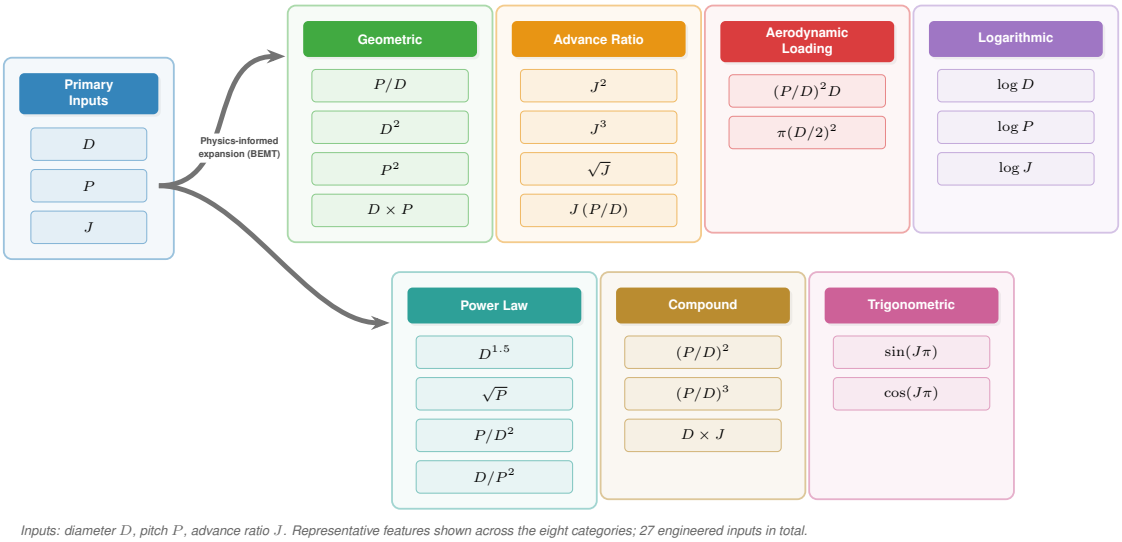


Figure 3 – Physics-informed feature engineering framework. Three primary inputs (D, P, J) are expanded into 27 features across eight physically motivated categories.

where T is thrust, P_{sh} is shaft power, and ρ is air density; shaft power is written P_{sh} to avoid confusion with the geometric pitch P. Beyond the three primary inputs (D, P, J), the engineered features fall into the following categories.

Geometric features: the pitch–diameter ratio P/D, diameter squared D^2, pitch squared P^2, and the diameter–pitch product $D \times P$. The ratio P/D is fundamental in propeller theory, governing blade loading and optimal operating conditions.

Advance-ratio features: J^2, J^3, $\sqrt{J}$, and the coupled term $J \times (P/D)$. These polynomial and root terms provide a flexible basis for the smooth, nonlinear dependence of C_T and C_P on advance ratio observed experimentally; they are motivated by dimensional and scaling arguments rather than by an exact closed-form law.

Aerodynamic loading features: the blade activity factor proxy $BAF = (P/D)^2 \times D$ and the disk area $A_{\text{disk}} = \pi(D/2)^2$. The BAF proxy is an empirical surrogate for a propeller's power-absorption capacity, whose rigorous definition depends on the spanwise chord distribution [24] that is unavailable for most catalogue propellers.

Logarithmic transformations: $\log(D+1)$, $\log(P+1)$, and $\log(J+0.01)$, which provide scale invariance and capture multiplicative relationships common in physical systems; the offset terms prevent singularities at zero.

Power-law features: $D^{1.5}$, $\sqrt{P}$, P/D^2, and D/P^2, capturing fractional power relationships not represented by integer exponents.

Compound interaction features: $(P/D)^2$, $(P/D)^3$, and $D \times J$.

Trigonometric features: $\sin(J\pi)$ and $\cos(J\pi)$, included to represent, in compact form, any smooth non-monotonic (quasi-periodic) variation of the coefficients with advance ratio.

In total the pipeline produces 27 engineered inputs. Table 2 summarizes representative features in each category together with their physical interpretation.

For consistency across the modeling pipeline we applied robust scaling to all features,

$$x_{\text{scaled}} = \frac{x - \text{median}(x)}{\text{IQR}(x)}, \tag{3}$$

where IQR denotes the interquartile range ($Q_{75} - Q_{25}$); robust scaling is less sensitive to outliers than standard normalization. Gradient-boosted trees are largely invariant to monotonic feature scaling, so this step mainly helps the auxiliary linear and neural-network baselines and aids reproducibility; it does not change the tree ensembles themselves. The scaler was fitted on the training set only and applied to the test set to prevent information leakage.

Table 2 – Representative physics-informed features by category, with physical interpretations. The full pipeline comprises 27 engineered inputs.

Category	Feature	Physical Interpretation
Primary	D	Propeller size, scale effect
	P	Blade pitch angle, loading
	J	Operating condition, flight regime
Geometric	P/D	Fundamental geometry ratio
	D^2	Area scaling, disk loading
	P^2	Pitch intensity
	$D \times P$	Combined size effect
Advance Ratio	J^2	Momentum theory thrust scaling
	J^3	Momentum theory power scaling
	$\sqrt{J}$	Velocity scaling
	$J \times (P/D)$	Coupled operation–geometry
Loading	$(P/D)^2 \times D$	Blade activity factor (proxy)
	$\pi(D/2)^2$	Disk area, reference area
Logarithmic	$\log(D)$	Scale invariance
	$\log(P)$	Pitch scaling
	$\log(J)$	Logarithmic velocity ratio
Power Law	$D^{1.5}$	Volume/strength scaling
	$\sqrt{P}$	Nonlinear pitch effect
	P/D^2	Pitch-to-area ratio
	D/P^2	Diameter-to-pitch intensity
Compound	$(P/D)^2$	Quadratic geometry effect
	$(P/D)^3$	Cubic geometry effect
	$D \times J$	Size-operation coupling
Trigonometric	$\sin(J\pi)$	Periodic flow behavior
	$\cos(J\pi)$	Phase-shifted periodicity

2.3 Machine Learning Models

2.3.1 Model Selection and Rationale

We used gradient boosting decision tree (GBDT) algorithms because of their strong performance on tabular data [26]. Recent studies show that tree-based methods consistently match or beat deep learning approaches, including transformers, on tabular datasets, especially for small to medium datasets ($<$100,000 samples) [15]. We selected three widely used GBDT implementations:

XGBoost (eXtreme Gradient Boosting) [16]: a highly optimized implementation with regularization to prevent overfitting, using a second-order gradient (Hessian) approximation for faster convergence, built-in handling of missing values, and column subsampling for variance reduction.

LightGBM (Light Gradient Boosting Machine) [17]: designed for efficiency on large datasets through leaf-wise tree growth (vs. level-wise in XGBoost), gradient-based one-side sampling (GOSS), and exclusive feature bundling (EFB) for dimensionality reduction.

CatBoost (Categorical Boosting) [18]: focuses on reducing overfitting through ordered boosting to eliminate prediction shift, a symmetric tree structure for fast scoring, and robustness to hyperparameter settings.

2.3.2 Hyperparameter Configuration

Each model was configured with hyperparameters balancing accuracy and generalization. Table 3 lists the optimal configurations determined through preliminary grid search with 3-fold cross-validation.

Table 3 – Optimal hyperparameter configuration for individual models.

Hyperparameter	XGBoost	LightGBM	CatBoost
Number of estimators	200	200	200
Maximum depth	5	7	7
Learning rate	0.1	0.1	0.1
Subsample ratio	0.9	1.0	–
Column subsample	0.8	1.0	–
Number of leaves	–	31	–
L2 regularization	1.0	0.0	3.0
Early stopping	50 rounds	50 rounds	50 rounds
Random seed	42	42	42

2.3.3 Voting Ensemble Architecture

Individual models capture different aspects of the underlying data structure. Ensemble methods combine multiple models to improve prediction accuracy and robustness [19]. We implemented a voting ensemble that averages predictions from the three GBDT models:

$$\hat{y}_{\text{ensemble}} = \frac{1}{3}\left(\hat{y}_{\text{XGB}} + \hat{y}_{\text{LGB}} + \hat{y}_{\text{CAT}}\right). \tag{4}$$

This simple averaging approach (as opposed to weighted voting) was chosen because individual model performances were comparable (within 0.5%), equal weights prevent overfitting to validation-set performance, and simplicity aids interpretability and reproducibility.
Separate ensemble models were trained for C_T and C_P. Propulsive efficiency was computed from the predicted coefficients,

$$\eta = \frac{J \cdot C_T}{C_P}. \tag{5}$$

As a more flexible alternative, we also evaluated a *stacking* ensemble, in which a linear meta-learner is trained on the out-of-fold predictions of the three base models in place of simple averaging; as reported in Section 3., it did not improve on the voting ensemble.

2.3.4 Model Comparison

To check that the voting ensemble is the best configuration and not a lucky choice, we ran a systematic evaluation of 15 model configurations in an AutoML-style study. The configurations include XGBoost variants (3, with different depth and estimator counts), LightGBM variants (3, with different leaf and depth settings), CatBoost variants (2, with different iteration counts and depths), scikit-learn Gradient Boosting variants (2), Random Forest variants (2), an Extra Trees ensemble (1), and voting ensembles (2, with different algorithm combinations). Each configuration used identical train–test splits and cross-validation for a fair comparison.

2.4 Uncertainty Quantification

Uncertainty quantification is essential for safety-critical applications such as UAV mission planning and propeller selection [22]. We implemented two complementary approaches to characterize prediction uncertainty: ensemble variance and quantile regression.

2.4.1 Ensemble Variance Method

The voting ensemble architecture naturally enables uncertainty estimation through prediction variance across individual models [29]. Each base model ($i \in \{\text{XGB, LGB, CAT}\}$) produces a prediction

$\hat{y}_i(x)$ for input x. The ensemble mean and standard deviation are

$$\hat{\mu}(x) = \frac{1}{3}\sum_{i=1}^{3} \hat{y}_i(x), \tag{6}$$

$$\hat{\sigma}(x) = \sqrt{\frac{1}{3}\sum_{i=1}^{3} (\hat{y}_i(x) - \hat{\mu}(x))^2}. \tag{7}$$

Assuming an approximately Gaussian error distribution, a 95% confidence interval is constructed as

$$\mathrm{CI}_{95\%}(x) = [\hat{\mu}(x) - 1.96\,\hat{\sigma}(x),\ \hat{\mu}(x) + 1.96\,\hat{\sigma}(x)]\,. \tag{8}$$

This approach captures *epistemic uncertainty* that arises from model disagreement; regions where the three models give divergent predictions indicate higher uncertainty in the learned function.

2.4.2 Quantile Regression Method

Quantile regression [27, 28] provides distribution-free prediction intervals without assuming Gaussian errors. Instead of predicting the conditional mean, quantile regression models the conditional quantiles of the target variable. We trained XGBoost models per target (C_T or C_P) with the quantile (pinball) loss objective

$$\mathscr{L}_\alpha(y,\hat{y}) = \sum_{i=1}^{n} \rho_\alpha(y_i - \hat{y}_i), \qquad \rho_\alpha(u) = \begin{cases} \alpha\, u & \text{if } u \geq 0, \\ (\alpha - 1)\, u & \text{if } u < 0, \end{cases} \tag{9}$$

for three quantile levels: $\alpha = 0.10$ (lower bound), $\alpha = 0.50$ (median), and $\alpha = 0.90$ (upper bound). The 80% prediction interval is then

$$\mathrm{PI}_{80\%}(x) = \left[\hat{Q}_{0.10}(x),\ \hat{Q}_{0.90}(x)\right]. \tag{10}$$

This method captures both epistemic and *aleatoric uncertainty* (inherent data variability from measurement noise, manufacturing tolerances, and environmental factors). Throughout, the reported point prediction remains the voting-ensemble mean of Eq. (4); the quantile models serve only to construct the interval bounds.

2.4.3 Uncertainty Calibration Assessment

To validate that predicted uncertainty estimates are reliable, we assess calibration through empirical coverage,

$$\mathrm{Coverage}_\alpha(\mathscr{D}_{\text{test}}) = \frac{1}{|\mathscr{D}_{\text{test}}|} \sum_{(x,y)\in\mathscr{D}_{\text{test}}} \mathbb{1}[y \in \mathrm{PI}_\alpha(x)]\,. \tag{11}$$

Well-calibrated uncertainty estimates should satisfy $\mathrm{Coverage}_\alpha \approx \alpha$; for example, a 95% confidence interval should contain approximately 95% of test samples. We also construct reliability diagrams that compare predicted confidence levels with empirical frequencies for a visual check of calibration quality.

2.5 Model Training, Validation, and Implementation

All models were trained using the following protocol. **Step 1 (data preparation):** load the processed dataset (30,188 samples), compute the 27 engineered features, and apply the stratified 80/20 train–test split by diameter. **Step 2 (feature scaling):** fit the RobustScaler on the training set only and transform both sets. **Step 3 (model training):** train individual XGB, LGB, and CAT models for C_T and C_P, then construct the voting ensembles by averaging. **Step 4 (uncertainty models):** obtain ensemble-variance intervals from the existing members and train the additional quantile models.

To get performance estimates that do not depend on a single split, we used 5-fold cross-validation during model selection: the training set was divided into five stratified folds; for each fold the model was trained on four folds and validated on the remaining one, with R^2, RMSE, and MAE computed on the validation fold, and the mean and standard deviation reported across folds. Final evaluation used the held-out test set (20% of the data), which was never used during training or hyperparameter

selection. Performance was assessed using the coefficient of determination R^2, root mean squared error (RMSE), and mean absolute error (MAE):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \qquad \text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \qquad \text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{12}$$

The train–test gap, $\text{Gap} = R^2_{\text{train}} - R^2_{\text{test}}$, measures overfitting; a negative gap indicates good generalization.

All models were implemented in Python 3.11 using XGBoost 2.0.3, LightGBM 4.1.0, CatBoost 1.2.2, scikit-learn 1.3.2 [31], pandas 2.1.3, and numpy 1.26.2. Training was performed on a consumer-grade workstation (Intel Core i7, 16 GB RAM) without GPU acceleration. Total training time for all models (including 15 configurations and uncertainty models) was approximately 15 minutes. Model persistence used Python's `pickle` serialization, yielding compact model files (12 MB total) suitable for deployment in embedded systems or web services.

3. Results

3.1 Model Performance and Comparison

Table 4 reports the held-out test performance of the individual GBDT models, the voting ensemble, and a stacking ensemble for all three targets. The three base learners perform almost identically, within 0.06% R^2 for C_T and within 0.1% for C_P, so no single algorithm dominates; this supports the equal-weight averaging of Eq. (4). The voting ensemble gives the best test accuracy for both directly predicted coefficients: $R^2 = 0.9375$ (RMSE $= 0.01084$) for C_T and $R^2 = 0.9217$ (RMSE $= 0.00724$) for C_P. Efficiency is a derived quantity (Eq. (5)) and is harder to predict, with $R^2 = 0.857$. The voting ensemble also slightly beats the more complex stacking ensemble on both coefficients, which supports the choice of simple averaging.

Table 4 – Held-out test performance (R^2) of individual models and ensembles. Best value per target in bold.

Model	C_T	C_P	η
XGBoost (tuned)	0.9367	0.9201	**0.8573**
LightGBM	0.9369	0.9211	0.8535
CatBoost	0.9363	0.9211	0.8448
Stacking ensemble	0.9364	0.9204	0.8547
Voting ensemble	**0.9375**	**0.9217**	0.8568

To show that the ensemble is the best option and not a chance result, we ran an AutoML-style search over 15 configurations. Figure 4 ranks the leading models by test R^2. The search confirms that the ensemble is best: the top AutoML configuration (a voting ensemble of XGBoost, LightGBM, and CatBoost) reached $R^2 = 0.9374$ for C_T and 0.9212 for C_P, just below the tuned ensemble (0.9375 and 0.9217). No configuration among the 15 beat the ensemble, and the strongest single model (LightGBM with 300 trees, depth 8, 50 leaves) trailed at $R^2 = 0.9372$ (C_T). These differences are within the noise floor of the data, which indicates that the approach is close to the best accuracy achievable on this dataset.

A baseline comparison shows the value of gradient boosting. A regularized linear (Ridge) model reached only $R^2 = 0.799$ (C_T), 0.685 (C_P), and 0.376 (η), and a multilayer-perceptron neural network reached 0.908, 0.869, and 0.795. Both are well below the ensemble. The poor linear-model result for efficiency confirms that the problem is strongly nonlinear.

Generalization. On this split the ensemble shows no sign of overfitting: averaged over targets, the voting ensembles have a near-zero, slightly *negative* train–test gap (≈ -0.0005), so training and test errors are statistically indistinguishable. Random Forest and Extra Trees show much larger positive gaps ($+0.041$ and $+0.056$), in line with their tendency to memorize the training data, which is why they rank below the regularized boosting methods. Because the split is done at the measurement

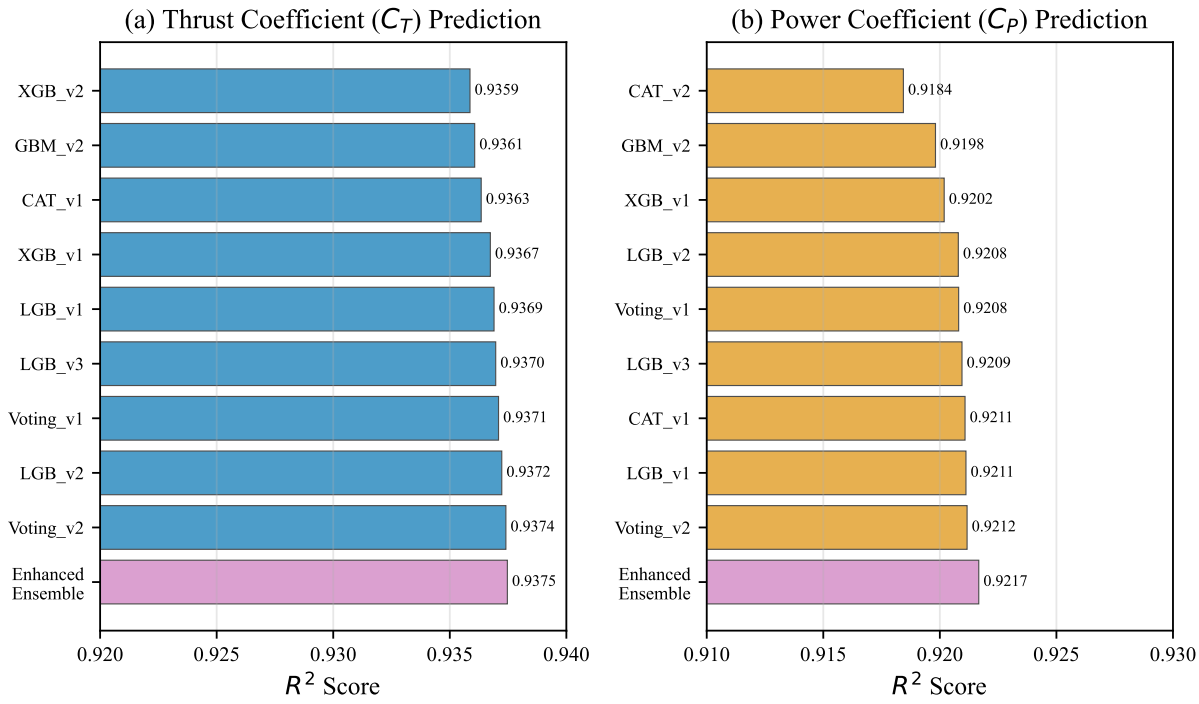


Figure 4 – Model comparison from the AutoML-style search. Top-ranked configurations by test R^2 for (a) C_T and (b) C_P; the ensemble (highlighted) is optimal for both targets.

level (Section 2.), this gap reflects interpolation rather than extrapolation to unseen geometries; we discuss the latter as a limitation in Section 4..

3.2 Feature Importance

We computed SHAP (SHapley Additive exPlanations) values on the trained models to interpret which physics-informed features drive predictions; the global rankings are shown in Figure 5. For C_T, the advance ratio J is the dominant feature (mean $|\text{SHAP}| = 0.0199$), followed by the pitch-to-diameter ratio P/D (0.0153) and the trigonometric term $\cos(J\pi)$ (0.0088). For C_P, the ordering shifts: P/D dominates (0.0122), ahead of J (0.0049) and $\cos(J\pi)$ (0.0047).

These rankings are physically meaningful. The strong dependence of C_T on J reflects the first-order effect of operating condition on thrust loading, and the prominence of P/D for C_P reflects the role of blade pitch in setting power absorption. This is consistent with blade element theory, in which power scales with the local angle of attack set by pitch. The repeated appearance of the engineered $\cos(J\pi)$ term among the top features suggests that the trigonometric encoding captures a smooth, non-monotonic variation of the coefficients with advance ratio that the raw features represent less efficiently. Because physically interpretable features appear at the top of the ranking, we are confident that the model has learned aerodynamically sensible relationships rather than spurious correlations.

3.3 Prediction Accuracy

Figure 6 shows actual-versus-predicted scatter plots on the held-out test set for all three targets. Predictions cluster tightly along the ideal diagonal across the full operating range, with no systematic bias. The thrust and power coefficients show the tightest agreement, while efficiency exhibits greater scatter, particularly at the extremes of its range, reflecting both its derived nature (errors in C_T and C_P propagate through Eq. (5)) and the amplification of relative error where C_P is small. Residuals are approximately symmetric and centered near zero, supporting the Gaussian assumption underlying the ensemble-variance intervals.

3.4 Uncertainty Quantification

Both uncertainty methods were checked for calibration on the held-out test set using the empirical coverage of Eq. (11). The ensemble-variance 95% confidence intervals are well calibrated, with an

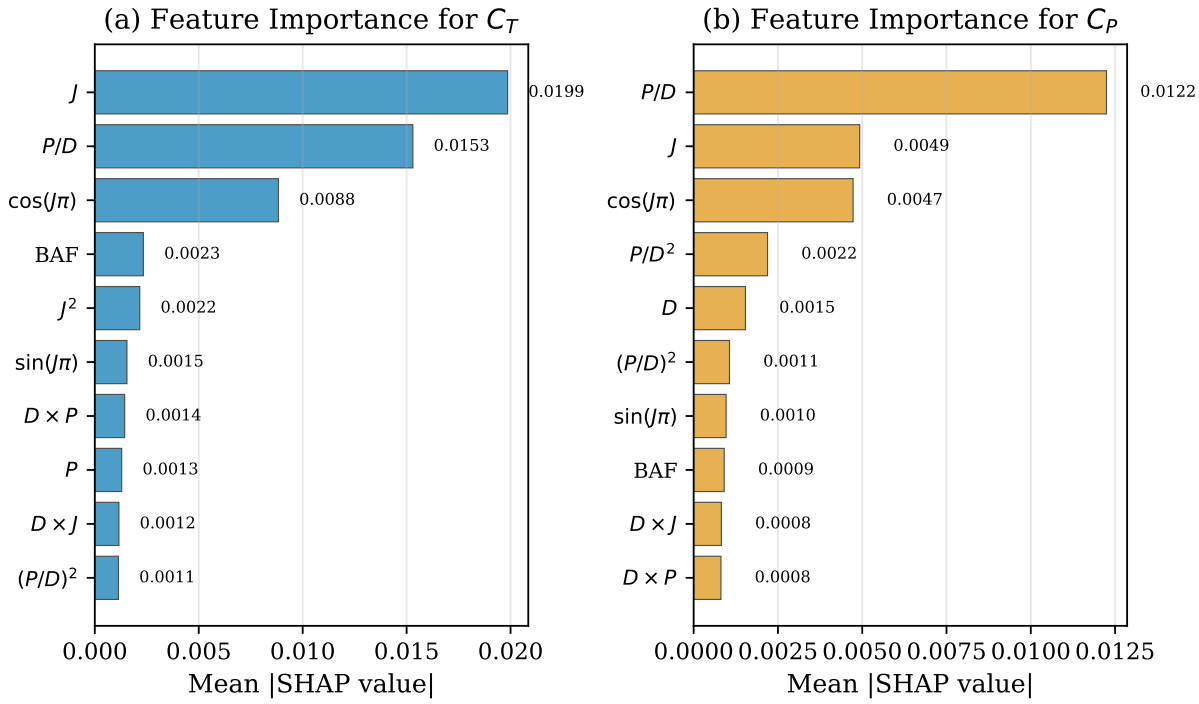


Figure 5 – SHAP global feature importance (mean |SHAP|) for the top features of (a) C_T and (b) C_P. Advance ratio dominates thrust prediction, while the pitch-to-diameter ratio dominates power prediction, consistent with propeller theory.

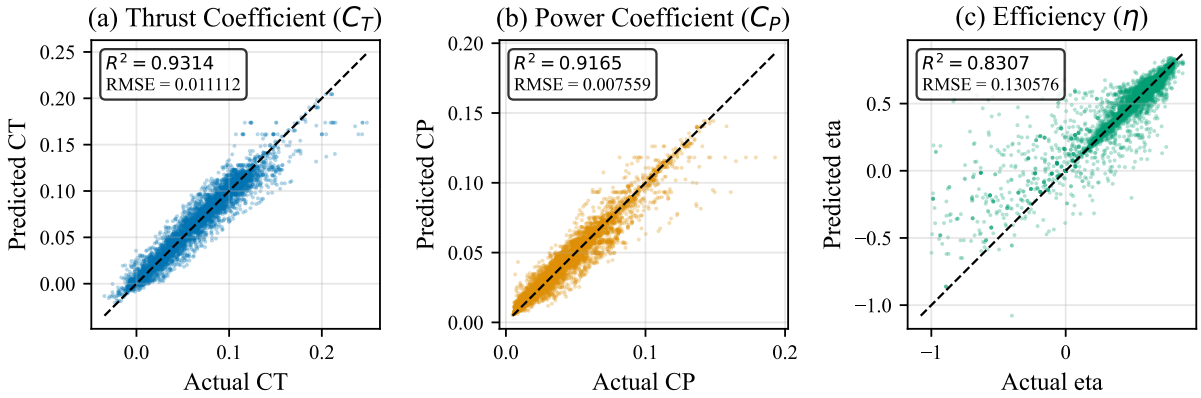


Figure 6 – Actual versus predicted values on the held-out test set for (a) C_T, (b) C_P, and (c) η. The dashed line denotes perfect prediction.

empirical coverage of 94.7–95.1% across C_T and C_P, within half a percentage point of the nominal 95% level. The quantile-regression 80% prediction intervals are mildly under-covered, with an empirical coverage of 71.8–73.3% against the nominal 80% level, so the learned 10th–90th-percentile band is a little narrower than ideal and would benefit from mild recalibration. Typical predicted uncertainties are about $\pm$1–3% of the coefficient value, small enough to support practical design decisions.

Figure 7 shows the practical output: full performance curves for a representative propeller, with C_T, C_P, and η plotted against advance ratio together with 95% confidence bands. The bands widen toward the extremes of the advance-ratio range, where training data are sparser and the flow physics is more complex, and narrow in the mid-J region where data are densest. This is the expected behavior of a well-calibrated estimate: the model reports lower confidence where it has seen less data.

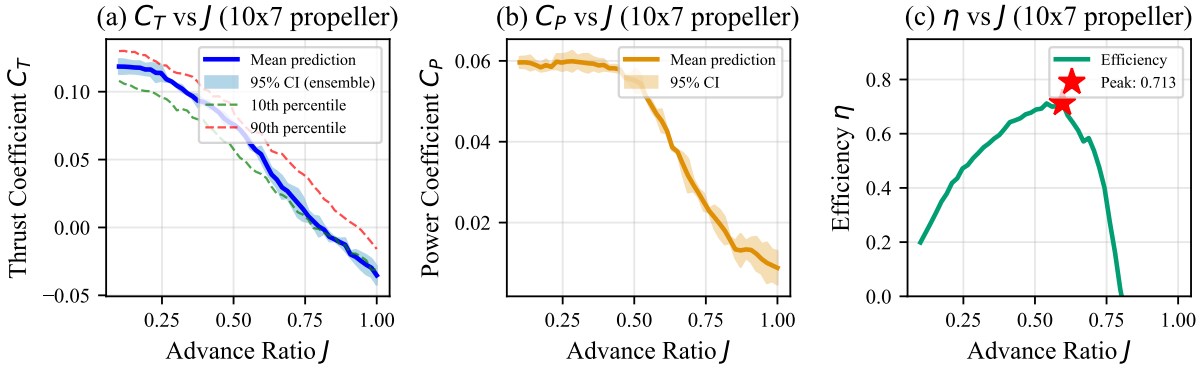


Figure 7 – Predicted performance curves with 95% confidence bands for a representative propeller: (a) C_T, (b) C_P, and (c) η versus advance ratio, with peak efficiency marked. Uncertainty grows at the extremes of the operating range.

3.5 Case Study: Uncertainty-Aware Propeller Selection

To show the method in a design setting, Figure 8 compares four candidate propellers (9×5, 10×7, 11×8.5, and 12×6) at a representative cruise condition. The model predicts C_T, C_P, and η for each candidate with 95% confidence intervals, which allows a direct, uncertainty-aware ranking. Two points stand out. First, the predictions separate the candidates clearly by efficiency, so a designer can identify the best performer in seconds instead of through successive wind-tunnel tests. Second, the error bars differ across candidates: geometries near the dense core of the training distribution get tighter intervals than those near its edge. A designer can therefore choose a slightly less efficient but more confidently predicted propeller when reliability is the priority, a decision that point-prediction tools cannot inform. This addresses a key need in UAV mission planning, where a reliable lower bound on performance is often more useful than an unqualified nominal value.

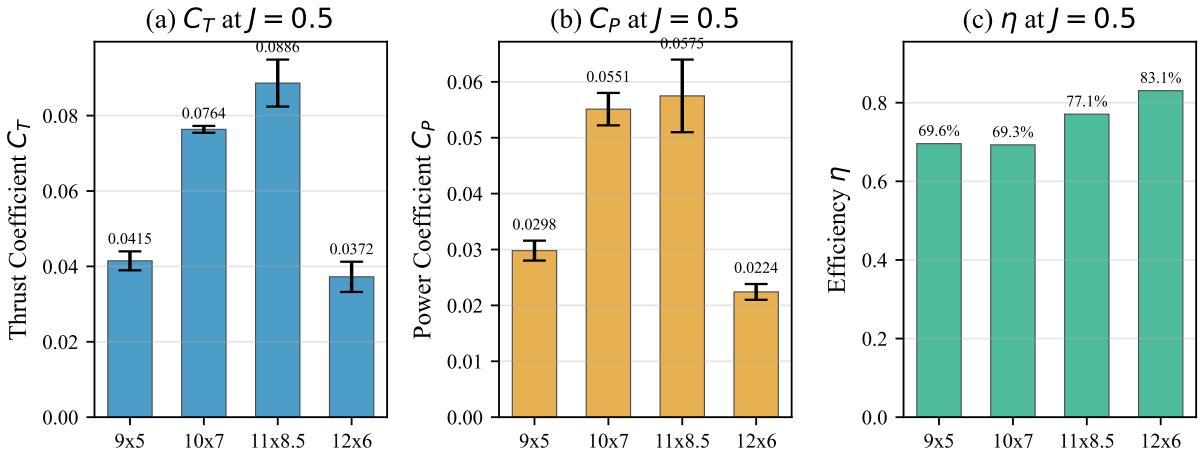


Figure 8 – Uncertainty-aware comparison of four candidate propellers at a representative cruise condition: predicted (a) C_T, (b) C_P, and (c) η with 95% confidence intervals.

4. Discussion

4.1 Interpretation of Predictive Performance

The ensemble explains about 94% of the variance in thrust coefficient and 92% in power coefficient on the held-out test set. Two observations suggest this is close to the practical ceiling for this dataset. First, a full AutoML-style search over 15 configurations did not improve on the ensemble by more than a few hundredths of a percent. Second, the residual error is of the same order as the expected experimental scatter in low-Reynolds-number wind-tunnel measurements, where repeatability and

manufacturing tolerances limit the attainable accuracy [25]. Further gains in accuracy are therefore more likely to come from more or better data than from more elaborate models.

Compared with traditional approaches, the data-driven model combines speed and convenience. Compared with CFD, it predicts about four orders of magnitude faster, turning an hours-long per-point computation into a sub-millisecond query and allowing hundreds of candidate geometries to be explored in early design. Compared with BEMT and related low-order methods [8], it needs only top-level geometry (D, P) and operating condition (J), not the detailed chord, twist, and airfoil-polar data that are rarely available for commercial propellers, and it avoids the Reynolds-number sensitivities that complicate low-order modeling in this regime. The trade-off is that, as a purely data-driven surrogate, the model is reliable only within the range of its training data, a limitation we state clearly and partly offset through uncertainty quantification.

4.2 The Value of Uncertainty Quantification

The main contribution of this work is a calibrated uncertainty estimate with every prediction. The ensemble-variance intervals are well calibrated (94.7–95.1% coverage for a nominal 95% level), and the performance-curve and case-study results show that the reported uncertainty behaves sensibly, growing where data are sparse and shrinking where they are dense. This turns the tool from a point estimator into a decision-support aid. In endurance estimation, for example, a calibrated interval lets a designer size battery margins against a reliable lower bound rather than an unqualified nominal value; in quality control, a measured performance that falls outside the predicted interval flags a possibly out-of-spec unit; and in test planning, wide intervals point to the geometries or operating points that need more experiments.

The two methods complement each other. Ensemble variance is almost free, since it reuses the existing base learners, and it captures epistemic (model) uncertainty, but it assumes approximate normality. Because it comes from only three strong and therefore highly correlated base learners, the raw ensemble variance is best read as a lower bound on the true epistemic uncertainty; its close-to-nominal coverage above should be read as an empirically calibrated result on this dataset, not a first-principles guarantee, and it should be re-checked whenever the model is used outside the training envelope. Quantile regression is distribution-free and captures aleatoric (data) variability, including possible asymmetry, at the cost of training extra models. The mild under-coverage of the 80% quantile intervals seen here is a known property of quantile-regression bands and could be corrected with a conformal recalibration step in future work. Used together, the two give a fuller picture of reliability than either alone.

4.3 Feature Engineering Insights

The SHAP analysis confirms that the physics-informed features carry real predictive signal and agree with aerodynamic theory: advance ratio governs thrust, while pitch-to-diameter ratio governs power. The consistent prominence of the engineered $\cos(J\pi)$ term is a useful empirical finding, and suggests that a smooth periodic encoding of advance ratio captures the non-monotonic variation of the coefficients more efficiently than polynomial terms alone. The feature ranking also shows which inputs most affect the predictions, and therefore where measurement precision matters most, which is useful guidance for instrumentation in experimental campaigns.

4.4 Limitations

Several limitations bound the use of the present model. (i) **Domain of validity.** The model is trained on conventional fixed-pitch propellers of 5–18 inch diameter at Reynolds numbers of 5×10^4–4×10^5; it should not be extrapolated to ducted, folding, variable-pitch, or much larger propellers, nor to compressible tip-speed regimes. (ii) **Targets.** Only C_T, C_P, and η are predicted; acoustic, vibratory, and structural quantities are out of scope. (iii) **Uncertainty sources.** The quantified uncertainty reflects model disagreement and measurement scatter in the training data, but does not explicitly capture manufacturing tolerances, blade wear, or in-service environmental effects. (iv) **Geometry abstraction.** Using only diameter and pitch as geometric inputs cannot distinguish propellers that share these parameters but differ in planform or airfoil; richer geometric descriptors would be required to resolve such cases. (v) **Static condition.** Predictions near $J = 0$ are inherently noisier, consistent with the

difficulty of static-thrust measurement. (vi) **Evaluation scope.** The train–test split is performed at the level of individual measurements rather than whole propellers, so different operating points of a given propeller can appear in both partitions. The reported metrics therefore quantify interpolation across operating conditions for geometries represented in the database; accuracy on entirely unseen propeller designs (a leave-propeller-out, or leave-geometry-out, setting) is expected to be lower and should be assessed before deployment to novel hardware.

4.5 Future Work

Promising extensions include rigorous leave-propeller-out (leave-geometry-out) validation to quantify true extrapolation performance; conformal calibration of the quantile intervals to guarantee coverage; incorporation of additional geometric descriptors (planform, airfoil family, blade count) to resolve geometries with identical D and P; physics-informed or multi-fidelity modeling that blends the surrogate with selective CFD to improve extrapolation; explicit modeling of manufacturing-tolerance uncertainty; and deployment as a web-based, uncertainty-aware propeller-selection tool integrated into UAV design workflows. The methodology is not specific to propellers and could be transferred to other low-Reynolds aerodynamic components such as rotors and ducted fans.

5. Conclusions

This work presented an uncertainty-aware machine learning method for predicting small-scale propeller performance from readily available geometry and operating parameters. Trained on 30,188 experimental measurements from the UIUC database covering more than 400 geometries, a voting ensemble of XGBoost, LightGBM, and CatBoost with 27 physics-informed features reaches held-out accuracies of $R^2 = 0.9375$ for thrust coefficient, 0.9217 for power coefficient, and 0.857 for efficiency, with essentially no overfitting. A systematic search over 15 configurations confirms that the ensemble is close to the best accuracy achievable on this dataset.

The main contribution is calibrated uncertainty quantification. Two methods, ensemble variance and quantile regression, provide prediction intervals; the 95% ensemble-variance intervals reach 94.7–95.1% empirical coverage, with typical uncertainties of $\pm$1–3%. The SHAP analysis confirms that the model behaves in line with propeller aerodynamics, with advance ratio governing thrust and pitch-to-diameter ratio governing power. The tool predicts about four orders of magnitude faster than CFD on commodity hardware, which supports rapid, risk-aware propeller selection and mission planning. To our knowledge, this is the first propeller performance predictor to provide calibrated uncertainty estimates; it addresses an important need for safety-critical UAV applications and transfers naturally to other aerodynamic components.

Acknowledgments

The authors gratefully acknowledge the support of the Interdisciplinary Research Centre for Aviation and Space Exploration and the Department of Aerospace Engineering at King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia. The authors also thank the University of Illinois at Urbana-Champaign for making the UIUC propeller database publicly available.

Contact Author Email Address

Corresponding author: Mohammad Irfan Alam (`mohammad.alam.1@kfupm.edu.sa`).

Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third-party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] Ji Y, Song K, Wen H, Xue X, Yan Y and Meng Q. UAV applications in intelligent traffic: RGBT image feature registration and complementary perception. *Advanced Engineering Informatics*, Vol. 63, 102953, 2025.

[2] Zhang M, Jiao J, Zhang J and Zhang Z. Aerodynamic performance modeling method of high-altitude propellers across the entire flight envelope. *Scientific Reports*, Vol. 15, No. 1, 11296, 2025.

[3] Zhang YX, Zhu YX, Zhang L, Yang ZT and Li YL. CFD study of propeller tip vortex cavitation. *International Journal of Multiphase Flow*, Vol. 182, 105020, 2025.

[4] Hsin CY. *Development and analysis of panel methods for propellers in unsteady flow*. PhD thesis, Massachusetts Institute of Technology, 1990.

[5] Chen M, Liu T and Fang T. Experimental and numerical analysis of a toroidal propeller. *Journal of Aerospace Engineering*, Vol. 38, No. 6, 04025093, 2025.

[6] Azizzadenesheli K, Kovachki N, Li Z, Liu-Schiaffini M, Kossaifi J and Anandkumar A. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, Vol. 6, No. 5, pp 320–328, 2024.

[7] MacNeill R and Verstraete D. Blade element momentum theory extended to model low Reynolds number propeller performance. *The Aeronautical Journal*, Vol. 121, No. 1240, pp 835–857, 2017.

[8] Drela M and Youngren H. XROTOR. `http://web.mit.edu/drela/Public/web/xrotor`, 2013.

[9] Zhao A, Zhang J, Li K and Wen D. Design and implementation of an innovative airborne electric propulsion measure system of fixed-wing UAV. *Aerospace Science and Technology*, Vol. 109, 106357, 2021.

[10] Brandt J, Deters R, Ananda G and Selig M. *UIUC propeller database*. University of Illinois at Urbana-Champaign, 2015.

[11] Brandt JB. *Small-scale propeller performance at low speeds*. PhD thesis, University of Illinois at Urbana-Champaign, 2005.

[12] Brunton SL, Kutz JN, Manohar K, Aravkin AY, Morgansen K, Klemisch J, Goebel N, et al. Data-driven aerospace engineering: reframing the industry with machine learning. *AIAA Journal*, Vol. 59, No. 8, pp 2820–2847, 2021.

[13] Nemati M and Jahangirian A. A data-driven machine learning approach for turbulent flow field prediction based on direct computational fluid dynamics database. *Journal of Applied Fluid Mechanics*, Vol. 17, No. 1, pp 60–74, 2023.

[14] Emami S and Martínez-Muñoz G. A gradient boosting approach for training convolutional and deep neural networks. *IEEE Open Journal of Signal Processing*, Vol. 4, pp 313–321, 2023.

[15] Shwartz-Ziv R and Armon A. Tabular data: deep learning is not all you need. *Information Fusion*, Vol. 81, pp 84–90, 2022.

[16] Niazkar M, Menapace A, Brentan B, Piraei R, Jimenez D, Dhawan P and Righetti M. Applications of XGBoost in water resources engineering: a systematic literature review (Dec 2018–May 2023). *Environmental Modelling & Software*, Vol. 174, 105971, 2024.

[17] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q and Liu TY. LightGBM: a highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, Vol. 30, 2017.

[18] Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV and Gulin A. CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, Vol. 31, 2018.

[19] Dietterich TG. Ensemble methods in machine learning. *International Workshop on Multiple Classifier Systems*, Springer, pp 1–15, 2000.

[20] Rahaman R. Uncertainty quantification and deep ensembles. *Advances in Neural Information Processing Systems*, Vol. 34, pp 20063–20075, 2021.

[21] Chechile RA. *Bayesian statistics for experimental scientists: a general introduction using distribution-free methods*. MIT Press, 2020.

[22] Abdar M, Pourpanah F, Hussain S, Rezazadegan D, Liu L, Ghavamzadeh M, Fieguth P, et al. A review of uncertainty quantification in deep learning: techniques, applications and challenges. *Information Fusion*, Vol. 76, pp 243–297, 2021.

[23] Adkins CN and Liebeck RH. Design of optimum propellers. *Journal of Propulsion and Power*, Vol. 10, No. 5, pp 676–682, 1994.

[24] Larrabee EE. Practical design of minimum induced loss propellers. *Business Aircraft Meeting and Exposition*, SAE Technical Paper 790585, 1979.

[25] Deters RW, Ananda GK and Selig MS. Reynolds number effects on the performance of small-scale propellers. *32nd AIAA Applied Aerodynamics Conference*, AIAA Paper 2014-2151, 2014.

[26] Grinsztajn L, Oyallon E and Varoquaux G. Why do tree-based models still outperform deep learning on

typical tabular data? *Advances in Neural Information Processing Systems*, Vol. 35, pp 507–520, 2022.

[27] Koenker R and Bassett G. Regression quantiles. *Econometrica*, Vol. 46, No. 1, pp 33–50, 1978.

[28] Meinshausen N. Quantile regression forests. *Journal of Machine Learning Research*, Vol. 7, pp 983–999, 2006.

[29] Lakshminarayanan B, Pritzel A and Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, Vol. 30, 2017.

[30] Zheng A and Casari A. *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, 2018.

[31] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, et al. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, Vol. 12, pp 2825–2830, 2011.