Comparisons of Performance Metrics and Machine Learning Methods on an Entry, Descent, and Landing Database

T.J. Wignall,* Tenavi Nakamura-Zimmerer,† James B. Scoggins,‡ Steven Snyder,§
Satvik G. Kumar,¶ and Brendon K. Colbert‖
NASA Langley Research Center, Hampton, Va, 23666

This work focuses on evaluating machine learning methods and their applicability to the generation of an aerodynamic database, particularly for trajectory analysis of a capsule during entry, descent, and landing with a focus on uncertainty quantification. The source data used are the wind tunnel and computational data for the Integrated Design Assessment Team (IDAT) configuration of the Orion Program, which has been publicly released. The methods used to generate the proposed databases are designed to naturally include a prediction interval, which will be evaluated both for their mean response as well as how well the prediction interval performs. These machine learning methods are compared to a traditionally generated database used by the Orion team as a baseline. It is found that while these machine learning methods perform well, the Orion database tends to still outperform, them showing that engineering experience and judgment is still needed to make the best database possible. However, these methods still provide comparable results with significantly less effort.

Nomenclature

BNN	=	Bayesian Neural Network
CDF	=	Cumulative Distribution Function
DropoutNN	=	Dropout Neural Network
EDL	=	Entry, Descent, and Landing
FMV	=	Function of Mach and Velocity
GP	=	Gaussian Process
GPR	=	Gaussian Process Regressor
IPM	=	Interval Predictor Model
KRR	=	Kernel Ridge Regressor
NARGP	=	Nonlinear Autoregressive Gaussian Process
NTF	=	National Transonic Facility
PI	=	Prediction Interval
RPM	=	Random Predictor Model
SCGN	=	Structured Covariance Gaussian Network
SNR	=	Sliced Normal Regressor
SPD	=	Symmetric Positive-Definiteness
SVR	=	Support Vector Regressor
C_D	=	Drag Coefficient
C_L	=	Lift Coefficient
$C_{m,cg}$	=	Pitching Moment Coefficient about the center of gravity
cal	=	Calibration
$\mathbb{F}$	=	A Given Cumulative Distribution Function

*Aerospace Engineer, Configuration Aerodynamics Branch, thomas.j.wignall@nasa.gov
†Research Aerospace Engineer, Flight Dynamics Branch
‡Research Aerospace Engineer, Aerothermodynamics Branch, AIAA Member
§Research Aerospace Engineer, Dynamic Systems and Control Branch, AIAA Member
¶Pathways Intern, Configuration Aerodynamics Branch, AIAA Student Member
‖Aerospace Engineer, Dynamic Systems and Control Branch

MAE	=	Mean Absolute Error
MSE	=	Mean Square Error
pin	=	Pinball loss
R^2	=	Coefficient of Determination
sha	=	Sharpness
α	=	Angle of Attack
σ^2	=	Variance

I. Introduction

As machine learning becomes more prevalent both in aerodynamics and in other fields, figuring out how to determine the best model for a project has become a more important and frequent topic of debate. One of the areas of interest for machine learning in aerodynamics is in the creation of aerodynamic databases (aerodatabases). Aerodatabases have a wide range of uses from controller design to design optimization. The present work considers aerodatabases for the use of trajectory analysis, which are usually generated using a relatively uniform density of data across a large parameter space. While modeling techniques have advanced significantly over the past few years thanks to advances in numerical methods and machine learning, there still remains significant engineering judgment in model creation, uncertainty quantification (UQ), and model performance evaluation.

This work is part of the Mars Lander Aerodynamic Model Data Fusion Using Machine Learning with Embedded Uncertainty Quantification (AeroFusion) project, which is a NASA project devoted to improving the creation of aerodynamic databases by prioritizing uncertainty quantification through all steps of the database generation process [1]. A piece of achieving the goal of better database generation, and the motivation of this work, is to determine how best to compare the different databases focusing on both the nominals and uncertainty quantification. The work uses a series of models generated using a variety of techniques including newer machine learning methods and looks at how different metrics help make comparisons between models possible, especially when they cover a large input parameter space. Traditional model evaluation metrics are agnostic to the type of data, so modifications to account for the problem specific concerns would be beneficial to help select best performing models. However, the data used here are already clustered in the area of interest due to experience, which gives an implicit weighting of performance in this region of interest.

Metrics evaluated on nominal or median predictions are commonplace, usually intuitive, and useful for quickly weeding out poor performing models. One of the requirements for the models under evaluation is that they are also able to report a prediction interval (PI) and so that is evaluated as well. While normally distributed uncertainty assumptions are useful and fairly common, a preference is given to metrics that do not make that assumption, which allows for greater flexibility in model selection.

The data used for this analysis were generated by the Orion Program [2] for one of their early aerodatabases for use in entry, descent, and landing (EDL) trajectory simulations. One of the goals of the AeroFusion project is to use all data generated over the lifetime of a project as opposed to just the most recent or highest fidelity collection because the heterogeneous nature of these data are considered beneficial. While multifidelity model generation techniques are not used here, the model evaluation tools used here would directly apply. A brief overview of the data and generated models looked at is provided, but the focus is on the metrics used to evaluate these models. The models are compared to the Orion v0.60 [3, 4] database as a baseline to show how these relatively hands-off methods compare to more traditional database generation techniques. While the Orion v0.60 database is over 10 years old, newer versions were developed using similar techniques, but with data accounting for updates to the flight geometry and improved aerodynamic modeling.

While the new models use the same data as were used for the creation of the nominal of the Orion V0.60 database, the Orion database has uncertainty terms that are not captured by the used aerodynamic data. Some of these terms are derived primarily from engineering judgment, while others are informed by older data not used here. We would expect the UQ performance of the data-driven methods would improve if these older data were included, but likely at the expense of the nominal prediction. This difference in uncertainty will be a major driver of the differences between the Orion database and the proposed model, but part of this work is motivated to see how well data-driven UQ terms are able to compare to more established methods.

A. Aerodynamic Data

The data used to generate our models come courtesy of the Orion Program. They are a collection of the previously publicly released data focused on the Integrated Design Assessment Team (IDAT) configuration, which is an axisymmetric capsule with a 30° backshell angle as described by Bibb et al. in Refs. [3, 4]. Figure 1 shows the definition of the IDAT geometry as well as the previous Orion baseline with a steeper backshell angle.

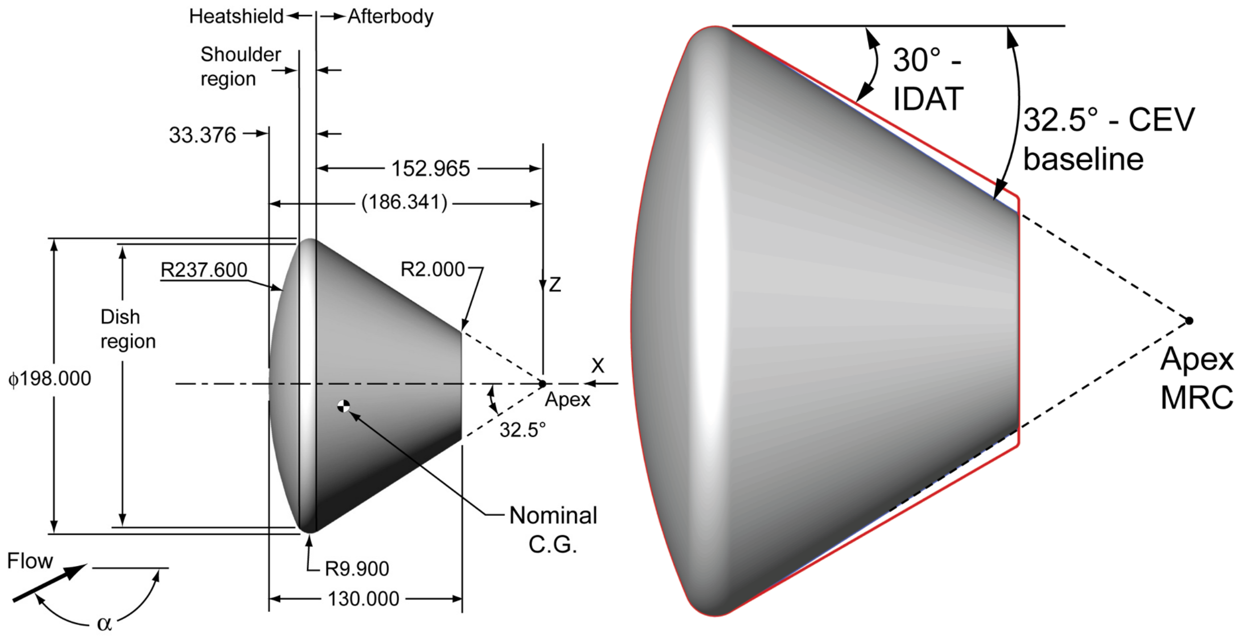


Fig. 1 Geometry definition of the Orion IDAT geometry used for this work. Linear dimensions in inches [Source: NASA Ref. [4]].

The dataset covers the data used in Refs. [3, 4], which covers a wide range of angles of attack, speeds, and Reynolds numbers from a variety of experimental and computational data sources. Computational flow solvers used include USM3D, OVERFLOW, LAURA, and DPLR. Experimental data included in this analysis were collected in the Langley Unitary Plan Wind Tunnel, the Ames Unitary Plan Wind Tunnel (both the 11-Foot Transonic Wind Tunnel (11-Ft TWT) and the 9- by 7- Foot Supersonic Wind Tunnel (9x7 SWT)), and the Langley 20-inch Mach 6 tunnel. The data have an increased resolution in the region of nominal flight, in this case corresponding to angle of attack, α, from 145° to 175° as can be seen in Fig. 2. For further details on this dataset, please refer to Refs. [3–5].

While all six force and moment coefficients are available in this dataset, the present work focuses on prediction of the coefficients of lift (C_L), drag (C_D) and pitching moment about the center of gravity ($C_{m,cg}$). The other force and moment coefficients are approximately zero due to the axisymmetric nature of the geometry. This also helps narrow the range of possible input parameters to α and the blended function of Mach and velocity (FMV). This FMV is equivalent to Mach number until the hypersonic regime where it is blended with kilofeet per second. This blending is done due to the databases needing to cover the entire entry, descent, and landing including the upper atmosphere where the speed of sound (and as a result Mach number) is less meaningful. The FMV is defined as

$$FMV = \begin{cases} M & \text{if } U \leq 8.8 \\ M(9.8 - U) + U(U - 8.8) & \text{if } 8.8 < U < 9.8 \\ U & \text{if } U \geq 9.8 \end{cases}, \tag{1}$$

where M is the Mach number and U is the dimensional velocity in kilofeet per second.

For the purpose of training and evaluating models, the dataset is randomly split into 80% training data and 20% testing data and each model uses the same split. A recent wind tunnel test performed in the National Transonic Facility (NTF) on the IDAT configuration will be used to see how these models perform on a new dataset. This dataset has a different underlying distribution than the data used to create the models and so perfect agreement is not expected; however, if the models are unable to capture this new data, it is unlikely they would successfully capture any flight data.

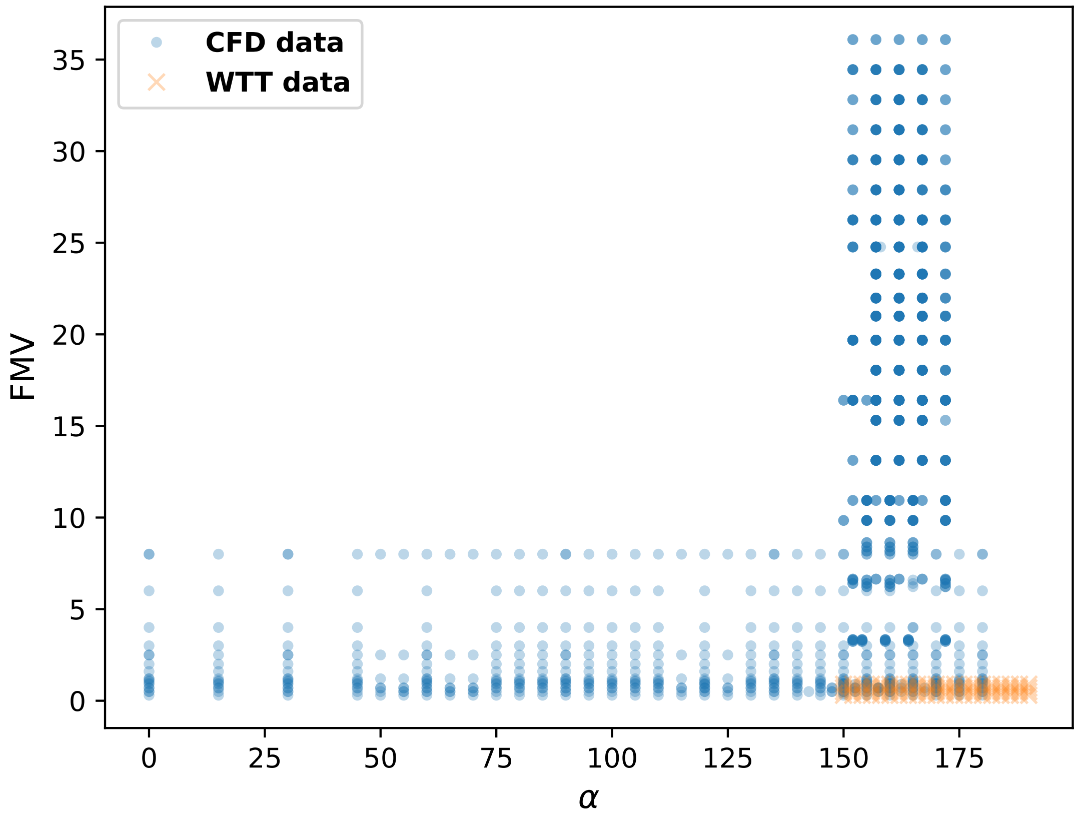


Fig. 2 FMV and α distribution of dataset used for training and testing.

There are some limitations on these data because it focuses on the subsonic regime, but will still be a useful comparison due to the complexity of aerodynamics on a blunt body like the IDAT capsule in this regime. These complications arise primarily from the increased sensitivity to the unsteady shedding that starts to dominate the forces and moments on the geometry. These new experimental data will come from a recently completed wind tunnel test at the National Transonic Facility, which focused on exploring subsonic Reynolds number effects. Since the models presented here are not functions of Reynolds number, data from the wind tunnel are downselected to minimize Reynolds number effects; however, some remains, which helps simulate increased measurement noise.

II. Model Evaluation

While aerodynamic databases have many uses, the focus for this work is the generation of databases to be used in trajectory analysis. This means trying to get good coverage across the whole domain of interest, but in particular what is expected to be flown. Consequently, the predicted "nominal" value and predicted *distribution* are both equally important. This means that the prediction interval (PI) is also of great concern, which should not be confused with a confidence interval. When it comes to judging resulting databases, this dual focus means that traditional metrics such as the coefficient of determination (R^2), which evaluates nominal prediction, need to be combined with metrics that evaluate the model prediction interval. A brief overview of the metrics used to evaluate nominal prediction is given followed by one for the metrics to cover the prediction interval.

A. Metrics on Nominal Value

In our experience, most databases are designed such that the median and mean values of the prediction interval are the same value. This is typically because a uniform prediction interval or some form of Gaussian distribution is assumed but holds valid for any symmetric distribution. However, there are cases, especially when nonlinear transformations are involved, that the median and mean value differ. For the purpose of this work, the nominal value is defined as the mean value of a distribution, and while not explicitly checked or required, most of the models evaluated in this paper return a symmetric distribution.

Three metrics are used to evaluate the prediction of the nominal value. The first is the R^2 value, the second is the mean squared error (MSE), and the third is mean absolute error (MAE). R^2 and MSE are selected to penalize outliers. Values for the MSE are dependent on specific problems and so judging a sufficiently satisfactory value requires domain specific knowledge. R^2 on the other hand is normalized, making it more universal across different problems. MAE is similar to MSE but does not penalize large errors to such a degree. While not the case in this dataset, there are situations where the expected value is close to zero across the domain and a few outliers can cause R^2 and MSE to give a worse view on the results. For example, if side force or yawing moment were included in these results, which are expected to be zero for this axisymmetric geometry, the R^2 metric could possibly be unsatisfactory despite the MAE being acceptable.

These three metrics are commonly used for model performance and easy to understand intuitively. The standard definitions are used, which are:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \tag{2}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{3}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{4}$$

where y_i is the observation at test point i, $\hat{y}_i$ is the predicted mean value at that point, and $\bar{y}$ is the dataset-wide mean value of the observation. N is the number of points being used.

The R^2 has a theoretical maximum of 1 and is the only term used here that should be maximized instead of minimized. The MSE and the MAE have ideal values of 0; however, these are limited by the inherent noise in the dataset. Ideally, this noise is properly characterized by the PI.

B. Metrics on Prediction Interval

Here we want to make clear why we want to evaluate PI and not confidence interval. While these are related terms, we are interested in being able to predict the range and distribution of possible future observations, which is represented by the PI, as opposed to distribution of predictions of the nominal, which would be represented by the confidence or credible interval. Because the use of the aerodynamic databases being considered will be used in trajectory analysis, all possible values need to be tested and not just the range of likely "truth" values. Different databases may have different needs resulting in the focus being on the confidence interval, but that is not the case here. The metrics selected are distribution independent and do not make any assumptions on their shapes.

Unsurprisingly, evaluating an interval is more difficult than evaluating a nominal prediction. This is because there are two competing goals for a prediction interval: a desire to cover all possible states and not being overly broad. In general, these metrics work by comparing the distribution of test points to the PI. There are three metrics that will be introduced; however, the primary metric used here will be calibration.

Calibration, sometimes referred to as fairness, directly compares the expected cumulative distribution function (CDF) with the observed CDF [6, 7]. Plotting these two gives a good understanding of how the test points are distributed throughout the PI. Ideally, the observed CDF matches the expectations, but that is rarely the case. We can quantitatively summarize the difference between the two CDFs by integrating their differences across the domain:

$$cal = \int_{r=0}^{r=1} |\mathbb{F}_{\text{obs}}(r) - \mathbb{F}_{\text{expected}}(r)|dr, \tag{5}$$

where $\mathbb{F}$ is either the observed CDF or the expected one and r is the quantile level. For all practical considerations, $\mathbb{F}_{\text{expected}}(r) = r$, which makes that portion of the equation simple. To calculate $\mathbb{F}_{\text{obs}}$, empirical frequency observations were used, which required taking model evaluations at each test point and finding the percentile for each observation; then overall at each percentile (or quantile), the number of points below the predicted output at the quantile of interest, $F(x, r)$, were counted and divided by the total number of points to get the observed value at that percentile. To put this in mathematical terms,

$$\mathbb{F}_{obs}(r) = \frac{1}{N}\sum_{i=1}^{N} G(x_i, y_i, r), \qquad G(x, y, r) = \begin{cases} 1 & \text{if } y \leq F(x, r) \\ 0 & \text{if } y > F(x, r) \end{cases}. \tag{6}$$

The next metric is called pinball loss or sometimes the check function, which is an extension of the mean absolute error to a quantile-based regression [8]:

$$pin_q = \frac{1}{N}\sum_{i=1}^{N} q \max(y_i - \hat{y}_i, 0) + (1 - q)\max(\hat{y}_i - y_i, 0), \tag{7}$$

where q is a selected quantile of interest from 0 to 1. It is a form of weighted averaging of the residuals that attempts to recenter the results at the quantile of interest. Like the MAE, the ideal value is 0 and the lower the better. The work will look at values for q of 0.025, 0.16, 0.84, and 0.975. These values are chosen to line up with the approximate quantiles of 1 and 2 standard deviations on a normal distribution. At $q = 0.5$, the term is identical to half the MAE described in equation 4.

The last metric that will be used is called sharpness. This is a measure of the average predicted variance and is intended to penalize models that have an overly broad distribution.

$$sha = \frac{1}{N}\sum_{i=1}^{N} \sigma_i^2, \tag{8}$$

where σ_i^2 is the variance predicted by a model at test point i. While calibration will show if the overall distribution is too spread out, it does not penalize being overly broad when there are few data in a region. Sharpness helps measure the overall spread of the PI at each test point. While variance does not completely define the range of possible values, it is a good single metric that encapsulates how spread out the data are.

C. On the Distribution of Data and Weighting Performance of the Trim Region

These metrics have a bias toward the model performance where there is a tighter cluster of data, and this is considered acceptable in this case. The data are clustered where experience has identified the most important region for these aerodynamic databases. However, if building a database without prior knowledge of the aerodynamic behavior or if data are more uniformly distributed across the parameter space, then variations on the above metrics can be made. If everything goes according to plan during EDL, a capsule will fly at the trim angle, α_0, which is the α where $C_{m,cg} = 0$. As a result, this α value should be considered more important when considering model comparisons. Many of the previous metrics have canonical "weighted" forms, but choosing how to calculate the weight is an open question. Some possible weighting terms for this problem are an inverse distance based on proximity to nominal trim or a function related to the probability that a given condition could return $C_{m,cg}$. These choices are a bit self-referential and could possibly lead to some unexpected results if used for optimization, but should pose no problem during model comparisons.

III. Overview of Models

So far, 11 different models have been implemented into the AeroFusion framework [1]. Through the use of common python libraries such as Scikit-learn [9] and TensorFlow [10], it is relatively straightforward to add already established methods to the framework while giving flexibility to add newer methods as well. Since the focus of the paper is on how to compare results of these models across a large parameter space, only a brief overview is provided. The four models selected for comparisons are given a slightly more detailed explanation. Some of these models require tuning through various hyperparameters, which were not explored in this work. Again, the focus of the paper is finding the best way to make these comparisons and which metrics are most useful when considering EDL aerodynamic databases. Table 1 has the list of models that are currently implemented with abbreviations used for convenience as well as references.

Table 1 List of Currently Implemented Models within AeroFusion.

Name	Abbreviation	Type	Predict Distribution?	Reference
Dropout Neural Network	DropoutNN	Neural Network	Yes	[11–13]
Bayesian Neural Network	BNN	Neural Network	Yes	[14]
Structured Covariance Gaussian Network	SCGN	Neural Network	Yes	[15, 16]
Support Vector Regressor	SVR	Kernel	No	[17–19]
Kernel Ridge Regressor	KRR	Kernel	No	[18]
Gaussian Process Regressor	GPR	Kernel	Yes	[20]
Nonlinear Autoregressive Gaussian Process	NARGP	Kernel	Yes	[21]
Polynomial Response Surface	Polynomial	Other	Yes	[22]
Random Predictor Model	RPM	Other	Yes	[23]
Interval Predictor Model	IPM	Other	Yes	[24–27]
Sliced Normal Regressor	SNR	Other	Yes	[28]

The models are also categorized for convenience. The first categorizations are variations of typical neural network machine learning methods that use a series of hidden layers, neurons, and activation functions that are trained via back propagation. The second categorizations are kernel methods, which find patterns in a high-dimensional implicit feature space. The third category serves as a catch-all for the remainder of the models.

A. Neural Network Methods

Neural network methods create a series of hidden layers with a set number of neurons per layer. Data are passed through these layers and between neurons through the use of activation functions. The weighting between all these parameters are typically trained via back propagation. The currently implemented neural network models are fairly typically structured being made up of a user defined number of hidden layers and neurons. It is important to note that neural network model performance is deeply tied to the architecture used and any limitations that are seen in the models used may be addressable by changes to the architecture; however, variations in architecture are not explored here.

What may be noticed is a lack of a baseline artificial neural network. This has not been implemented due to its inability to predict a distribution on its own. The three implemented methods are Bayesian Neural Networks (BNN) [14], Dropout Neural Networks (DropoutNN) [11–13], and Structured Covariance Gaussian Networks (SCGN) [15, 16]. While all three are able to predict a nominal value and a distribution, in this paper, we compare DropoutNN and SCGN. For the DropoutNN, we use a standard feedforward NN with 4 hidden layers, each with 128 neurons, swish activation function [29], and dropout rate $r = 0.1$. The NN weights are learned by minimizing the MSE with respect to the training data. SCGN is constructed from two NNs. The NN parameterizing the mean has 4 hidden layers, each with 16 neurons and $tanh$ activation functions; the covariance NN also has 4 hidden layers, each with 32 neurons and $tanh$ activation functions. The SCGN is trained by maximizing the log-likelihood of the training data as described in Refs. [15, 16]. For both DropoutNN and SCGN, we use the Adam optimizer [30] with a learning rate of $\lambda = 10^{-3}$ and minibatch sizes of 16, 32, 64, and 128 data points over 100, 150, 200, and 250 epochs, respectively.

B. Kernel Methods

Kernel methods work by finding an implicit mapping to a higher dimensional feature space and are common in machine learning. The implemented Kernel methods are the Kernel Ridge Regression (KRR) [18], Support Vector Regression (SVR) [17–19], Gaussian Process Regressor (GPR) [20], and the Nonlinear Autoregressive Gaussian Process (NARGP) [21]. Part of the work for AeroFusion is looking at introducing hierarchical datasets and models, so NARGPs have been implemented as a first step. Development work continues with a newly developed multihierarchy Gaussian process, which looks to expand NARGPs in situations where source data contains varying fidelity that are hard to categorize on one scale [31].

For the model comparison performed in this work, we consider only a single-fidelity GPR [20]. A Gaussian Process (GP) can be thought of as extending the Gaussian distribution for scalars to random functions. More specifically, we say that a real-valued random function is distributed according to a GP, if for any finite subset of inputs, the random vector formed by evaluating the function on those inputs is Gaussian, with a given mean and covariance, where the elements of the covariance matrix are determined by evaluating a kernel function on the corresponding pair of inputs. The covariance kernel must satisfy a symmetric positive-definiteness (SPD) property such that for any subset of function inputs, the resulting covariance matrix is SPD. Once a kernel function is supplied, the predictive distribution from a GP is found by conditioning over the set of training data. Unlike the neural network models, GPRs are considered to be nonparametric, as they do not have any parameters that must be learned in order to provide a predictive distribution. However, most kernel functions of interest do have some parameters (often called hyperparameters), which can be learned to maximize the marginal likelihood of generating the training data from the GP distribution. For this work, we use a Squared Exponential kernel function [32] with Automatic Relevance Determining weights. The weights and variance parameter for the kernel function are determined by maximizing the marginal likelihood via a gradient descent algorithm with restarts to ensure a global optimum is reached. In order to improve the predictions obtained from the GPR, the FMV inputs are first transformed into their natural logarithm before being supplied to the GPR during the training phase. Finally, in order to compare with the other models presented in this work, the observation noise present in the training data are added to the variance of the predictive distribution.

C. Other Methods

Not all models implemented fall under the previous two categories and so a catch-all category is included. Out of these, the polynomial response surface or just polynomial is used. This is a straight forward 6th-order polynomial regression fitted with least squares and is included as a sort of baseline [22]. The expectation is that the added complexity and flexibility of the other models will allow for better performance over this classic model. Other models that have been implemented are the Random Predictor Model (RPM) [23], Interval Predictor Model (IPM) [24–27], and the Sliced Normal Regressor (SNR) [28].

IV. Comparisons

While summary metrics are useful for objective decisions, data visualization is still necessary for human understanding and intuition. Before diving into the metrics, a quick exploration of how the highlighted models perform across the domain is presented. Following the graphical comparisons, the summary metrics are discussed. These are further broken down by nominal and distribution related metrics explored on both the test points and the NTF data.

A. Graphical Comparisons

This section uses plots of constant FMV or α to give a better understanding of the model predictions compared to the baseline Orion v0.60 database. The training, testing, and NTF test data are also plotted to give better context for model comparisons. These charts present the baseline Orion v0.60 database in red while the new model is in blue. For space reasons only C_D and $C_{m,cg}$ are plotted; however, C_L performance tends to be similar to C_D and so much of the analysis would be redundant. The shaded region represents the 90% coverage region (5% to the 95%). The training data are represented with small circles, the test data with Xs and the NTF test data with stars when present.

The first comparison is one of the more complicated regions with FMV = 0.5, which is shown across the whole α range in Fig. 3, and then the expected flight α range in Fig. 4. At this condition, there are no shocks or other temperature dependent aerodynamics, which dominate at the higher FMV values in the databases. However, it is at this condition where large unsteady wakes tend to dominate the forces and moments on the vehicle. This is also one of the easier conditions to do wind tunnel testing for similar reasons, and as a result, there are significant amounts of data present in this region including the data from the NTF testing. This goes back to how the summary metrics will be heavily influenced by this region due to the density of data.

One of the first things to note is the sharp discontinuity in the Orion database for $C_{m,cg}$ at $\alpha = 70°$. This represents a shift in the fluid dynamics as the stagnation point transitions from the heatshield shoulder to the backshell shoulder over a small and unstable α region. The data do capture this shift but most of the models are not able to capture this naturally. In Fig. 3d, the SCGN model is best able to capture this through an increase in the PI; however, the nominal value is similar to the rest of the models. While looking at the SCGN model, the C_D nominal predictions trend closely to the Orion database while the uncertainties grow significantly as α approaches 0°. The Orion database is developed using a term-based uncertainty model, which allows engineering judgment to apply similar distributions in regions with less data while the SCGN is predicting a narrow PI, which may not be providing the desired level of uncertainty.

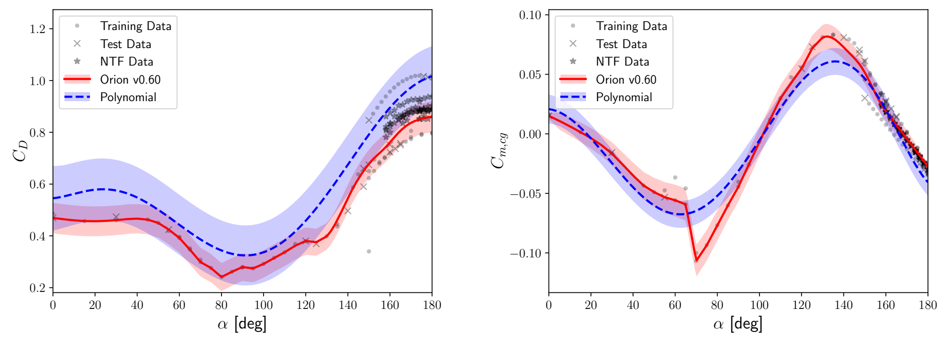
Moving up to the GPR and focusing on Fig. 4c, the nominal prediction falls in the center of the NTF data and the PI covers all of them. Based only on the NTF data and this condition, this would likely come out as the best model. The DropoutNN tends to be fairly similar to the Orion DB both in nominal and PI with a major difference being in the C_D predictions where the nominal splits between the two "lines" of training data unlike the Orion database, which sticks closer to the lower set. This slight shift causes it to fall right in the thick of the NTF data. As somewhat expected, the polynomial model does not have the best performance. It tends to round out the peaks of the data but its PI does tend to cover most of the data used outside those regions. As such, it may be a nice compromise when covering a wider range of data is necessary over matching data peaks.

The next regime looked at is a supersonic condition of FMV = 6.0, which can be found in Fig. 5. This is on the border of the hypersonic regime and so capturing the effects of reacting flow starts to become more important. The flow is dominated by the bow shock in front of the vehicle. While getting data at these conditions can be difficult, the aerodynamic coefficients tend to be well behaved making the model predictions better. Here, except for the polynomial model, all models tend to have similar nominal predictions but have significantly different PIs. The DropoutNN has the closest to the Orion database PI while the GPR has a slightly broader PI. The SCGN has a narrow PI, which is likely a reflection of the sparser data in this part of the database. While these qualitative comparisons of the PI can be useful, determining the proper distribution of the PI requires more quantitative metrics.

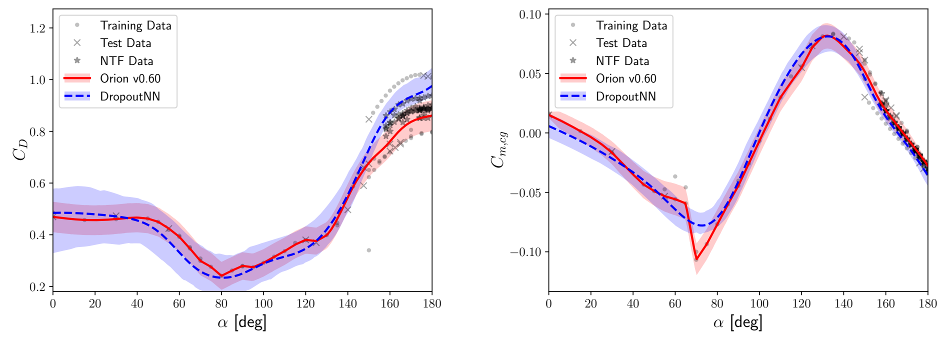
The last domain looked at is a constant α = 160° and the plots for it are in Fig. 6. This α was selected due to its proximity to trim. These constant α plots really highlight the sudden change in aerodynamics in the transonic region around FMV = 1. The Orion database is constructed in a method that takes this into account, but the other models do not. The polynomial model struggles most significantly here since it is a least squares fit across the whole domain. The rest of the models are better able to adapt to this discontinuity. The GPR tends to account for this discontinuity by increasing the PI around that region, which is probably overly broad when compared to the Orion database and the other models. The rest of the PI trends are similar to the FMV = 6.0 regime where the polynomial PI is very broad and the SCGN is very narrow.

Overall the models have similar capabilities and nominal performance based on these plots. The polynomial model tends to have the widest PI, which does not vary strongly with position in the database. The SCGN model is able to quickly respond to a local spread of data, which may be beneficial if there are certain conditions that should have a larger uncertainty; however, when the data get sparser, the PI narrows considerably. The GPR models tend to strike a good balance between increasing the range of the PI when there is a large spread of data while not overly narrowing. This is likely due to the addition of the observation noise to the predictions. In these plots, the DropoutNN tends to be the closest to the Orion database while having a slightly larger PI.

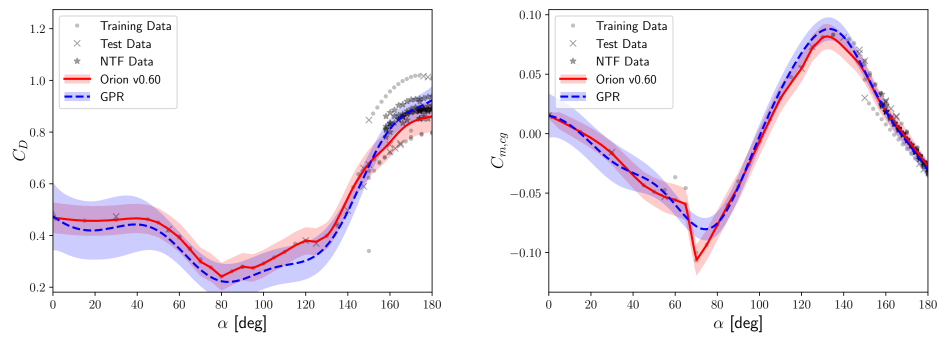
These plots give a general idea and may be sufficient for removing models from contention, but they only cover a small range of the entire database. To plot a single variable at a time, with FMV every 0.5 and α = 1°, would require over 10,000 sets of plots. While already impractical for this small case, the plotting method of model comparison becomes even more intractable as the number of inputs and outputs increase and the number of models in contention increase. Hence, summary metrics are used to compare performance over either the entire range of values or subregions of interest.



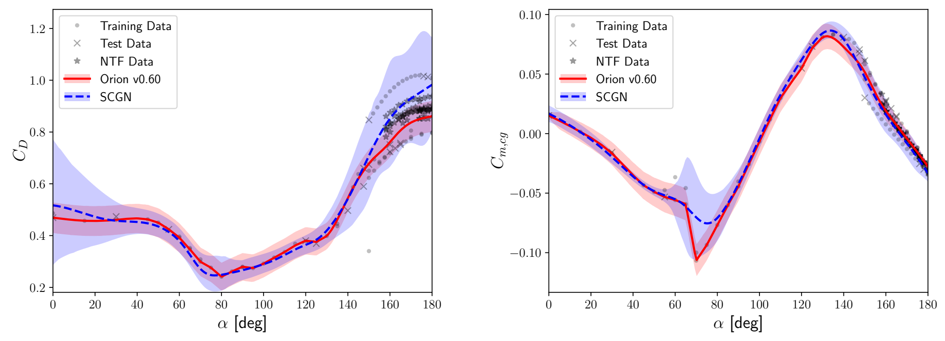
(a) Polynomial



(b) DropoutNN

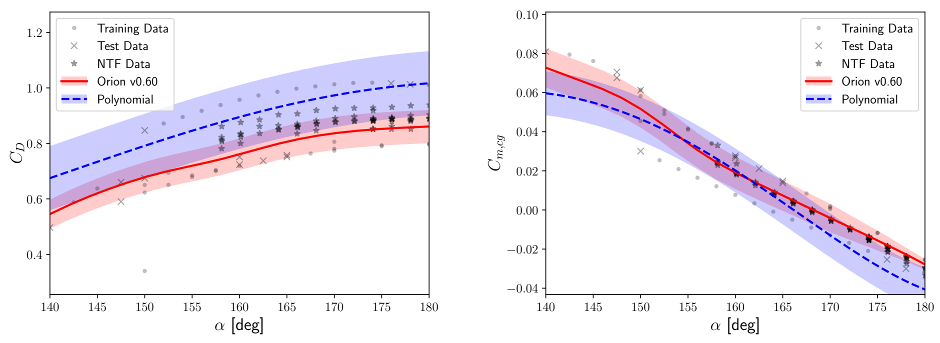


(c) GPR

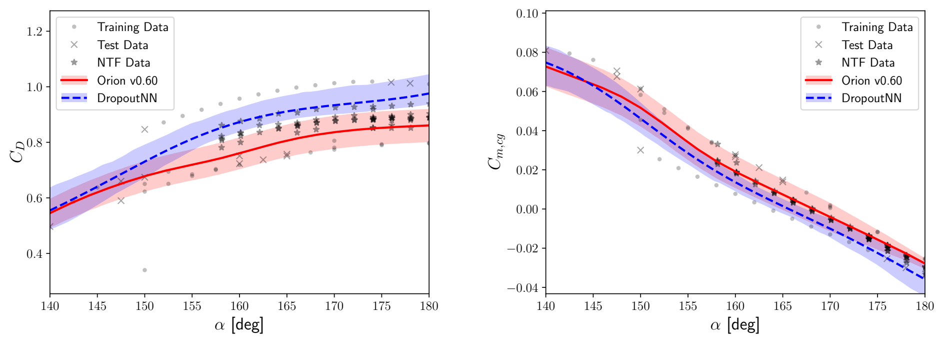


(d) SCGN

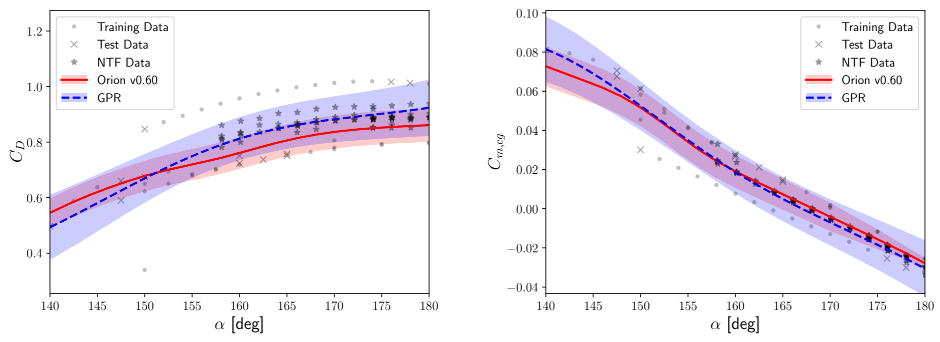
Fig. 3 Model predictions of C_D and $C_{m,cg}$ at a constant FMV of 0.5. Shaded region represents a 90% PI.



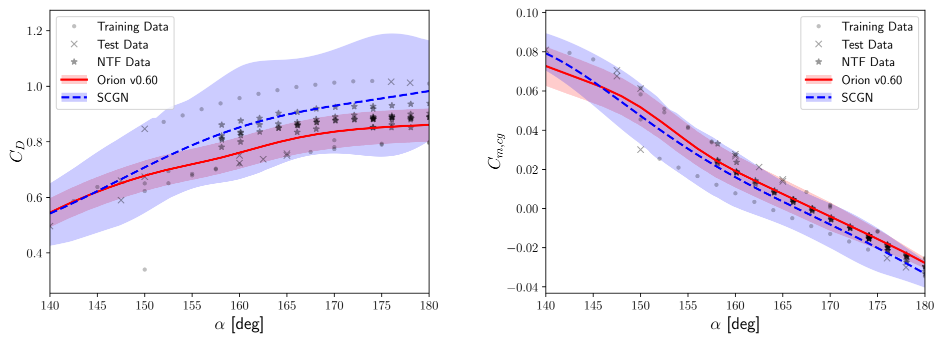
(a) Polynomial



(b) DropoutNN

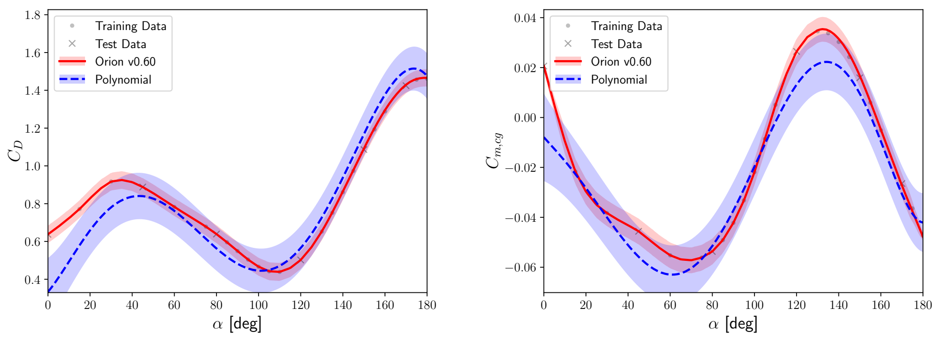


(c) GPR

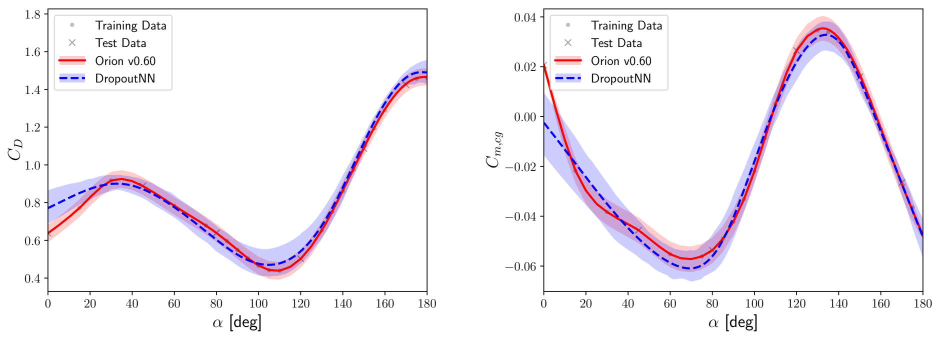


(d) SCGN

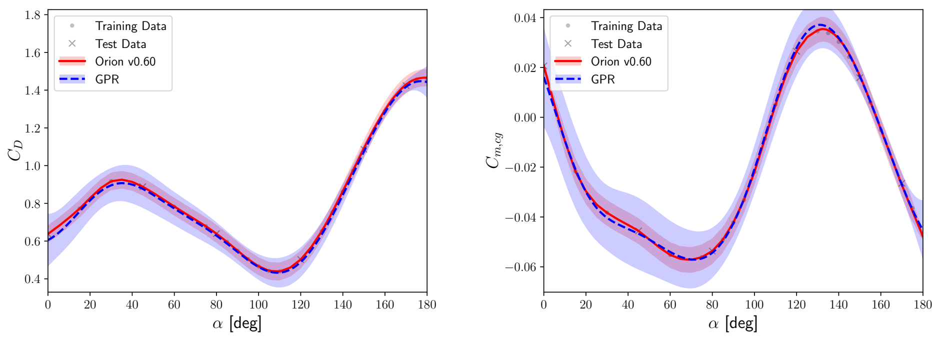
Fig. 4 Model predictions of C_D and $C_{m,cg}$ at a constant FMV of 0.5 from α = 140° to 180°. Shaded region represents a 90% PI.



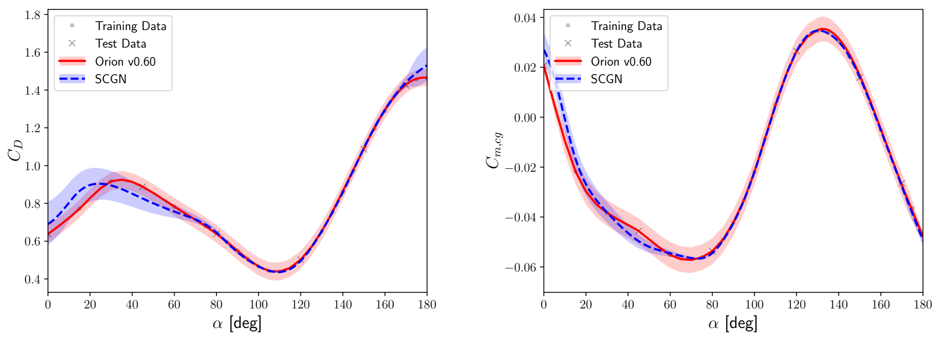
(a) Polynomial



(b) DropoutNN

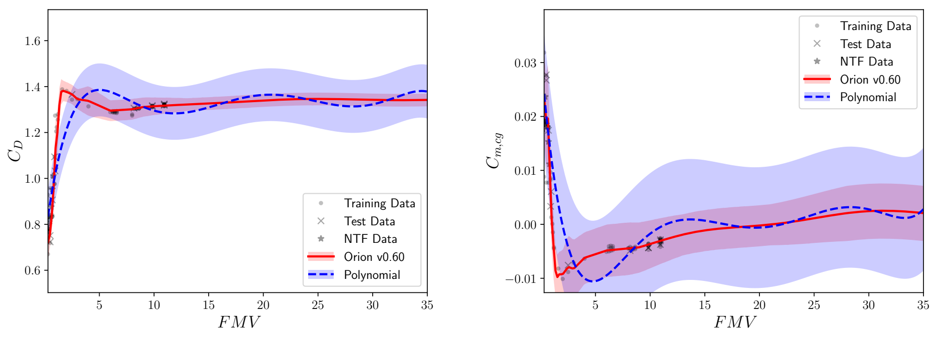


(c) GPR

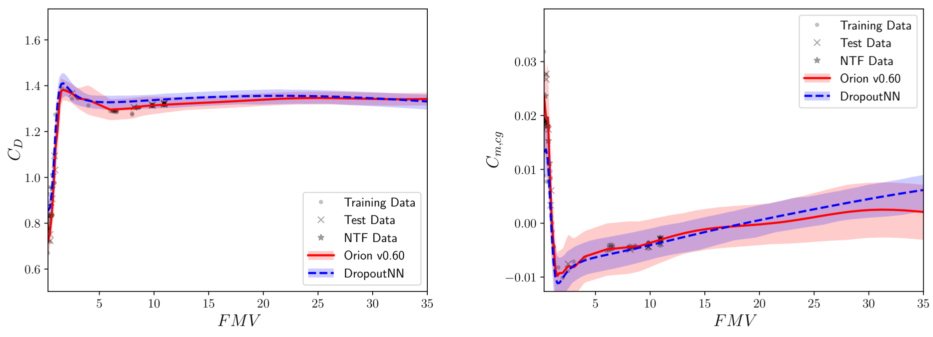


(d) SCGN

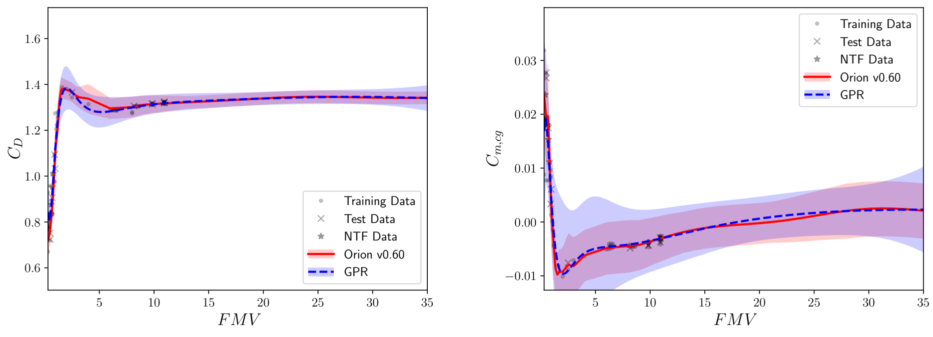
Fig. 5 Model predictions of C_D and $C_{m,cg}$ at a constant FMV of 6.0. Shaded region represents a 90% PI.



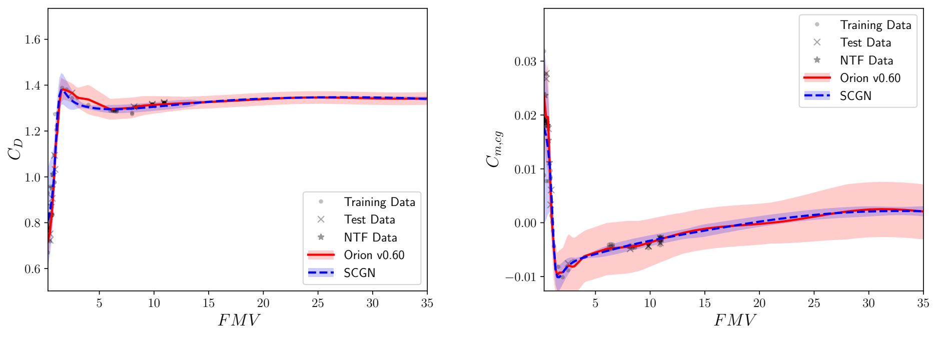
(a) Polynomial



(b) DropoutNN



(c) GPR



(d) SCGN

Fig. 6 Model predictions of C_D and $C_{m,cg}$ at a constant α = 160°. Shaded region represents a 90% PI.

B. Summary Metrics

The comparison by metrics is broken down into several tables. The first collection of tables looks at performance of the models on the 20% of the points left out of the training data, which is then further subdivided by the results on the nominal and the metrics on the PI. The second collection is these same summary metrics evaluated using the NTF data.

1. Summary Metrics on Test Data

The metrics in this section focus on model performance when using the 20% of the data withheld for testing beginning with the nominal predictions. Table 2 has the 3 nominal evaluation metrics based on the test data. It is divided into three subtables for each output value of interest. The R^2 values for all of them, except the polynomial model, tend to be clustered together showing that all models have similar ability to predict the nominal. This conclusion carries over to the MSE and MAE, which show similar trends. As an example of needing domain specific knowledge, the MAE for C_D for all models are on the order of 0.01, which is acceptable for capsule aerodynamics; however, if applied to a commercial airliner at cruise, these values would probably be high enough to disqualify the models due to performance needs to predict C_D on the order of 0.0001 (i.e., 1 count of drag). This is related to the inherent noise in the data, which is much higher for capsule aerodynamics than commercial airliners. Based only on these values, it would be difficult to determine if any one of these models are better than the others except for perhaps the polynomial model being the worst. Considering that R^2 and MSE penalize outliers, when combining with an evaluation of the PI, there is some redundancy in characterizing test data that are far from the nominal predictions. With this in mind, we consider the MAE to be the most informative metric on the nominal predictions in this scenario.

The next step is then to consider the quality of the PI, which can be found in Table 3. The pin scores are less intuitive than the MAE scores from the previous table but show similar trends. This metric seems best suited for when there is a critical quantile that needs to be considered as opposed to the overall PI. The next term is cal, which does give an idea of the overall PI. The $\mathbb{F}_{\text{obs}}$ of the different models are shown in Fig. 7 plotted against $\mathbb{F}_{\text{expect}}$. The area between the solid colored lines and the dashed lines is then summed to give cal. While the integrated value is important, these charts are also a quick way to confirm that not just the PI at certain bounds but the entire distribution is behaving as desired. The Orion v0.60 in blue can be picked out by its ‘S’ shape, which was originally believed to be a result of the uniform distribution assumptions. Once the GPR was plotted in red, it was clear something else causes this miscalibration since the GPR forces a normal distribution. It is believed that this is from having an overly broad distribution that accounts for uncertainties not present in the test data. The GPR has the same problem because the observation noise, which is added back into the GPR prediction, is assumed to be constant across the domain, which should not be the case with this dataset for best performance. Any mismatch at the ends implies that the PI does not cover all observed points, with larger mismatches implying more points not captured by the PI. Going back to cal itself, the results of all candidate models except the GPR and the DropoutNN for C_D are better than the Orion v0.60 database. The implies that these data-driven methods are better at capturing the distribution within the data as compared to the more tailored approach of the Orion database. The final metric is sha, which penalizes overly broad distributions, which in this work is a good companion to cal. Looking at the C_D metrics, the cal for the polynomial and SCGN model are fairly close but the lower sha value of the SCGN favors its selection.

2. Summary Metrics on NTF Test Data

This last section looks at model performance specifically when using the NTF test data, which are from a recent wind tunnel test. This test was designed to explore Reynolds number effects and while the data are limited to the higher Reynolds numbers, there is still significant variation in the C_D data. This is reflected in the metrics where the nominal C_D results are significantly worse than the other two coefficients as shown in Table 4. To begin with, the Orion database is the only one with an R^2 value for C_D greater than 0.5. However, for the other two metrics, the other models are much more competitive. Most of the models perform better than the Orion database for C_L.

Moving to the performance of the distribution in Table 5, the poor performance in C_D again shows up in the high value across the board. Unlike the nominal metrics, the GPR and SCGN models have better performance as compared to the Orion database. For the other coefficients, the results are better across the board; however, the Orion database still performs best.

Table 2 Metrics on performance of nominal prediction on the test data.

(a) C_D

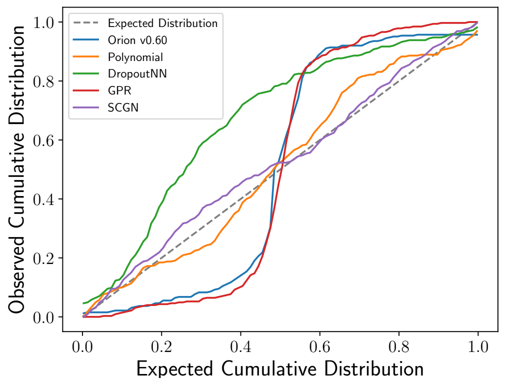
Model Name	R^2	MSE	MAE
Orion v0.60	0.988037	0.001447	0.015313
DropoutNN	0.980752	0.002328	0.031776
GPR	0.989522	0.001267	0.019904
Polynomial	0.951178	0.005905	0.053889
SCGN	0.986490	0.001634	0.019928

(b) C_L

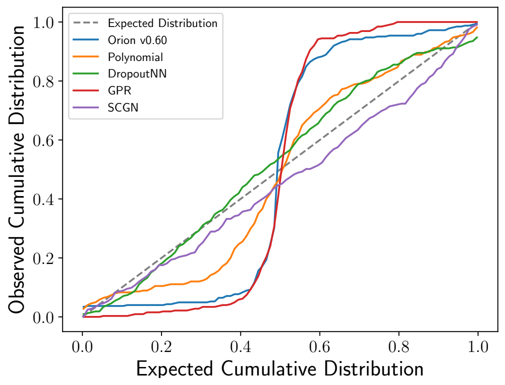
Model Name	R^2	MSE	MAE
Orion v0.60	0.996364	0.000160	0.005314
DropoutNN	0.984817	0.000667	0.016442
GPR	0.993766	0.000274	0.009178
Polynomial	0.951073	0.002150	0.024055
SCGN	0.994314	0.000250	0.008261

(c) $C_{m,cg}$

Model Name	R^2	MSE	MAE
Orion v0.60	0.985699	0.000010	0.001264
DropoutNN	0.959724	0.000029	0.003099
GPR	0.982959	0.000012	0.001792
Polynomial	0.928805	0.000052	0.004394
SCGN	0.982422	0.000013	0.001838



(a) C_D



(b) $C_{m,cg}$

Fig. 7 Calibration Curves.

Table 3 Metrics on performance of the PI on the test data.

(a) C_D

Model Name	$pin_{0.025}$	$pin_{0.16}$	$pin_{0.84}$	$pin_{0.975}$	cal	sha
Orion v0.60	0.001470	0.006954	0.009248	0.004720	0.155977	0.001088
DropoutNN	0.004632	0.010015	0.010547	0.003086	0.173067	0.001144
GPR	0.004130	0.014227	0.013435	0.004047	0.172515	0.010737
Polynomial	0.004426	0.018180	0.020523	0.006503	0.039716	0.005337
SCGN	0.001259	0.005266	0.006402	0.001637	0.026282	0.001743

(b) C_L

Model Name	$pin_{0.025}$	$pin_{0.16}$	$pin_{0.84}$	$pin_{0.975}$	cal	sha
Orion v0.60	0.000732	0.003386	0.003502	0.000914	0.174970	0.000457
DropoutNN	0.001550	0.005229	0.005257	0.001306	0.078506	0.000609
GPR	0.003220	0.010343	0.010810	0.003291	0.201895	0.006830
Polynomial	0.004073	0.010605	0.010479	0.003354	0.092193	0.001611
SCGN	0.000617	0.002446	0.002246	0.000607	0.039283	0.000436

(c) $C_{m,cg}$

Model Name	$pin_{0.025}$	$pin_{0.16}$	$pin_{0.84}$	$pin_{0.975}$	cal	sha
Orion v0.60	0.000315	0.000892	0.000750	0.000157	0.173669	0.000015
DropoutNN	0.000241	0.000897	0.001231	0.000538	0.039265	0.000016
GPR	0.000582	0.001929	0.001898	0.000584	0.196519	0.000218
Polynomial	0.000641	0.001902	0.001721	0.000531	0.077545	0.000045
SCGN	0.000120	0.000558	0.000552	0.000113	0.046217	0.000012

Table 4 Metrics on performance of nominal prediction on the NTF data.

(a) C_D

Model Name	R^2	MSE	MAE
Orion v0.60	0.847890	0.001223	0.029125
DropoutNN	-0.064418	0.008559	0.082206
GPR	0.436154	0.004534	0.049269
Polynomial	0.146157	0.006866	0.072278
SCGN	0.368781	0.005076	0.057819

(b) C_L

Model Name	R^2	MSE	MAE
Orion v0.60	0.966905	0.000812	0.024351
DropoutNN	0.987257	0.000313	0.013513
GPR	0.979902	0.000493	0.016815
Polynomial	0.947079	0.001298	0.028570
SCGN	0.987146	0.000315	0.013563

(c) $C_{m,cg}$

Model Name	R^2	MSE	MAE
Orion v0.60	0.990682	0.000004	0.001531
DropoutNN	0.911199	0.000041	0.006210
GPR	0.974085	0.000012	0.002909
Polynomial	0.855534	0.000067	0.007049
SCGN	0.975197	0.000011	0.002926

Table 5 Metrics on performance of the PI on the NTF data.

(a) C_D

Model Name	$pin_{0.025}$	$pin_{0.16}$	$pin_{0.84}$	$pin_{0.975}$	cal	sha
Orion v0.60	0.002642	0.013205	0.006159	0.001834	0.186066	0.002144
DropoutNN	0.026083	0.041848	0.018578	0.003850	0.454616	0.001293
GPR	0.002802	0.007084	0.018859	0.004825	0.145149	0.009259
Polynomial	0.001649	0.020161	0.022819	0.005271	0.305537	0.005119
SCGN	0.008471	0.018988	0.024416	0.006084	0.163609	0.011561

(b) C_L

Model Name	$pin_{0.025}$	$pin_{0.16}$	$pin_{0.84}$	$pin_{0.975}$	cal	sha
Orion v0.60	0.001801	0.007390	0.006077	0.001658	0.043165	0.000695
DropoutNN	0.001655	0.005737	0.005550	0.001844	0.091625	0.001130
GPR	0.002851	0.008674	0.011400	0.003275	0.156383	0.005824
Polynomial	0.002214	0.010274	0.006787	0.002006	0.054979	0.001683
SCGN	0.001244	0.004079	0.004393	0.001308	0.110029	0.000276

(c) $C_{m,cg}$

Model Name	$pin_{0.025}$	$pin_{0.16}$	$pin_{0.84}$	$pin_{0.975}$	cal	sha
Orion v0.60	0.000156	0.000661	0.000907	0.000185	0.156059	0.000018
DropoutNN	0.000371	0.001669	0.001883	0.000276	0.418899	0.000021
GPR	0.000603	0.002147	0.001387	0.000484	0.198180	0.000192
Polynomial	0.000519	0.002156	0.002099	0.000216	0.212730	0.000049
SCGN	0.000349	0.001395	0.000957	0.000483	0.195046	0.000041

V. Conclusions

A handful of models using machine learning methods are evaluated and compared to a more traditionally created database. These metrics include evaluation of mean prediction as well as predicted distribution. In general, the more traditional Orion database tends to perform better on mean prediction but the data-driven UQ of the machine learning models are more reactive to the spread in the training data. This data-driven UQ model has advantages such as reducing the need for engineering expertise and time to develop. However, the term-based UQ by inspection used in the Orion database still allows for more flexibility in including known unknowns not present in the training data. This is reinforced when the new NTF data are introduced and used to evaluate database performance, showing the Orion database does significantly better than the proposed models. One of the outstanding tasks for the AeroFusion project is finding a seamless way to combine this data-informed UQ with UQ by inspection to cover uncertainties not accounted for in the source data. In the meantime, the data-driven UQ can be computed very quickly and with very little effort on the engineer, which opens up the usage of UQ to help guide data collection during a wind tunnel test.

It was found that two metrics were most useful in characterizing model performance for aerodatabases. For nominal predictions, the R^2, MSE, and MAE all trended similarly across models and outputs; however, the accounting for large errors can be accomplished by evaluating the PI instead. We will be favoring MAE going forward since it does not stress these outliers as much, which will be done by cal. It was found that cal gives an intuitive understanding of how the PI of a model compares to the test data and allows both a numeric representation for quantitative analysis and input dimension insensitive plotting for qualitative analysis. While the MSE can be weighted to stress performance in a critical region, there is not a similar canonical form for a weighted cal, which may be a future avenue of research.

Out of the proposed models, the SCGN model has the best mix of nominal metrics and distributed metrics. It has the smallest PI in the high FMV range where data are sparse, which may not be a desired behavior. The other major task is finding a way to join multiple models trained on subregions, which would allow for different models in subsonic, supersonic, and hypersonic regimes if needed. The current machine learning methods are able to replicate some of this behavior, but could see a significant improvement if properly implemented. Naive implementations allow for discontinuities at the borders of the subregions and need to be addressed before being evaluated. When more mature, the domain knowledge of which subregions are important and where their boundaries are located will become a critical part of dividing the dataset.

The nominal metrics are common and behavior here was mostly as expected. The most surprising was the difficulty in of all models in predicting the C_D data from the NTF test; however, most of those points still fell within a 90% PI. When using new data, the PI could possibly be better evaluated using a pass/fail metric that checks to see if the new data are within some percentile of interest. For evaluating performance of the PI on test data, cal proved the most useful and intuitive, and since it is not dependent on output values, it is independent of problem. The other metrics on the distribution are useful supplements to the cal score, and in particular if a certain quantile is important, the pinball score allows a direct comparison at that quantile. In general, if a model performed well in one metric, it performed well on all metrics, which helps remove the question of how to weigh which metric is most important.

While most of the models here assume some normal or uniform distribution, future work should look at the performance of these metrics when the underlying distribution is known to be more complicated. It is not uncommon for aerodynamics to encounter bistable states such as the hysteresis seen during evaluating dynamic pitching rates. This would seriously affect the definition of nominal where a median or mean value would give a misleading notion of the underlying distribution. One thing to keep in mind is that these presented models have a number of hyperparameters that can be tuned in order to improve their predictions. While techniques like cross-validation allow for an automatic way to pick these parameters, normally a single metric is used to optimize. Continuing work would explore optimizing these hyperparameters based on different metrics to see how that affects the final predictions and performance.

References

[1] Snyder, S., Wignall, T., Green, J. S., Kumar, S. G., Lee, M. W., Nakamura-Zimmerer, T., Scoggins, J. B., and Williams, R. A., "AeroFusion: Data Fusion and Uncertainty Quantification for Lander Vehicles," AIAA Paper 2023-XXXX, AIAA SCITECH Forum, National Harbor, MD, 2023.

[2] "Orion Program," Available at `https://www.nasa.gov/exploration/systems/orion/index.html`, Accessed: 11/8/2022.

[3] Bibb, K., Brauckmann, G., Walker, E., and Robinson, P., "Development of the Orion Crew Module Static Aerodynamic Database, Part I: Hypersonic," *29th AIAA Applied Aerodynamics Conference*, AIAA, 2011. https://doi.org/10.2514/6.2011-3506.

[4] Bibb, K., Walker, E., Brauckmann, G., and Robinson, P., “Development of the Orion Crew Module Static Aerodynamic Database, Part II: Subsonic/Supersonic,” *29th AIAA Applied Aerodynamics Conference*, AIAA, 2011. https://doi.org/10.2514/6.2011-3507.

[5] CAP Aerodynamics Team, “CAP Orion MPCV Aerodynamic Database Development, Version 0.90,” NASA EG-CAP-13-21, December 2017.

[6] Zhao, S., Ma, T., and Ermon, S., “Individual Calibration with Randomized Forecasting,” arXiv, 2020. https://doi.org/10.48550/ARXIV.2006.10288.

[7] Tran, K., Neiswanger, W., Yoon, J., Zhang, Q., Xing, E., and Ulissi, Z. W., “Methods for comparing uncertainty quantifications for material property predictions,” IOP Publishing, 2020, p. 025006. https://doi.org/10.1088/2632-2153/ab7e1a.

[8] Koenker, R., and Bassett, G., “Regression Quantiles,” *Econometrica*, Vol. 46, No. 1, 1978, pp. 33–50. URL http://www.jstor.org/stable/1913643.

[9] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.

[10] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” 2015. URL https://www.tensorflow.org/, software available from tensorflow.org.

[11] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Mach. Learn. Res.*, Vol. 15, No. 56, 2014, pp. 1929–1958.

[12] Gal, Y., and Ghahramani, Z., “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” *33rd International Conference on Machine Learning (ICML)*, Vol. 48, 2016, pp. 1050–1059. https://doi.org/10.5555/3045390.3045502.

[13] Gal, Y., Hron, J., and Kendall, A., “Concrete Dropout,” *31st International Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 3584–3593. https://doi.org/10.5555/3294996.3295116.

[14] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D., “Weight Uncertainty in Neural Networks,” , 2015. https://doi.org/10.48550/ARXIV.1505.05424.

[15] Dorta, G., Vicente, S., Agapito, L., Campbell, N. D., and Simpson, I., “Structured Uncertainty Prediction Networks,” *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5477–5485. https://doi.org/10.1109/CVPR.2018.00574.

[16] Nakamura-Zimmerer, T., Stringer, M. T., Colbert, B. K., and Scoggins, J. B., “Structured Covariance Gaussian Networks for Orion Crew Module Aerodynamic Uncertainty Quantification,” AIAA Paper 2023-XXXX, AIAA SCITECH Forum, National Harbor, MD, 2023.

[17] Smola, A. J., and Schölkopf, B., “A tutorial on support vector regression,” *Stat. Comput.*, Vol. 14, No. 3, 2004, pp. 1990–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88.

[18] Murphy, K. P., *Machine Learning: A Probabilistic Perspective*, MIT Press, Cambridge, Massachusetts, 2012.

[19] Palar, P. S., Zakaria, K., Zuhal, L. R., Shimoyama, K., and Liem, R. P., “Gaussian Processes and Support Vector Regression for Uncertainty Quantification in Aerodynamics,” *AIAA Scitech Forum*, 2021, pp. 1–12. https://doi.org/10.2514/6.2021-0181.

[20] Rasmussen, C. E., and Williams, C. K. I., *Gaussian Processes for Machine Learning*, The MIT Press, 2006.

[21] Perdikaris, P., Raissi, M., Damianou, A., Lawrence, N., and Karniadakis, G. E., “Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 473, No. 2198, 2017, p. 20160751.

[22] Myers, R., Montgomery, D., and Anderson-Cook, C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley Series in Probability and Statistics, Wiley, 2009. URL https://books.google.com/books?id=89oznEFHF_MC.

[23] Crespo, L. G., Kenny, S. P., and Giesy, D. P., "Random predictor models for rigorous uncertainty quantification," *International Journal for Uncertainty Quantification*, 2015.

[24] Campi, M. C., Calafiore, G., and Garatti, S., "Interval predictor models: Identification and reliability," *Automatica*, Vol. 45, No. 2, 2009, pp. 382–392.

[25] Crespo, L. G., Giesy, D. P., and Kenny, S. P., "Interval predictor models with a formal characterization of uncertainty and reliability," *53rd IEEE conference on decision and control*, IEEE, 2014, pp. 5991–5996.

[26] Lacerda, M. J., and Crespo, L. G., "Interval predictor models for data with measurement uncertainty," *2017 American Control Conference (ACC)*, 2017, pp. 1487–1492.

[27] Crespo, L. G., Kenny, S. P., Giesy, D. P., Norman, R. B., and Blattnig, S., "Application of interval predictor models to space radiation shielding," *18th AIAA Non-Deterministic Approaches Conference*, 2016, p. 0431.

[28] Colbert, B., Crespo, L., and Peet, M., "A Sum of Squares Optimization Approach to Uncertainty Quantification," *American Control Conference*, 2019.

[29] Ramachandran, P., Zoph, B., and Le, Q., "Searching for Activation Functions," 2017.

[30] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," 2014.

[31] Scoggins, J. B., Wignall, T., Nakamura-Zimmerer, T., and Bibb, K., "Multihierarchy Gaussian Process Models for Probabilistic Aerodynamic Databases using Uncertain Nominal and Off-Nominal Configuration Data," AIAA Paper 2023-XXXX, AIAA SCITECH Forum, National Harbor, MD, 2023.

[32] Duvenaud, D., Lloyd, J., Grosse, R., Tenenbaum, J., and Zoubin, G., "Structure Discovery in Nonparametric Regression through Compositional Kernel Search," *Proceedings of the 30th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 28, edited by S. Dasgupta and D. McAllester, PMLR, Atlanta, Georgia, USA, 2013, pp. 1166–1174. URL https://proceedings.mlr.press/v28/duvenaud13.html.