Machine learning-based Hall Thruster scaling laws

Alfredo Marianacci, S Mazouffre

Machine learning-based Hall Thruster scaling laws

IEPC-2024-267

Presented at the 38th International Electric Propulsion Conference, Toulouse, France
June 23-28, 2024

A. Marianacci* and S. Mazouffre†
ICARE-CNRS, Orléans, 45100, France

The methodology for scaling the geometry and operating parameters of Hall Thrusters, as described in this paper, relies on supervised Machine Learning algorithms. This approach aims to determine the geometrical dimensions, propellant mass flow rate and discharge voltage, with thrust and specific impulse as input requirements. The magnetic field is also taken into account. Gradient Boosting Regression (GBR) is identified as the most appropriate algorithm for this purpose. The scaling process utilizes a specific database of 54 thrusters, incorporating measurements with xenon, krypton, and argon as propellants. An analytical method based on the GBR and optimization algorithms has been developed and validated to design a Hall thruster that meets space mission specifications. Additionally, the procedure is complemented by a more direct and intuitive graphical method.

Nomenclature

B	= magnetic field magnitude, G
d	= channel mean diameter, m
d_{ext}	= channel external diameter, m
d_{int}	= channel internal diameter, m
h	= channel width, m
I_d	= discharge current, A
I_{sp}	= specific impulse, s
L	= channel length, m
$\dot{m}_n$	= propellant mass flow rate, kg/s
n	= number of tests
P_d	= discharge power, W
T	= thrust, N
U_d	= discharge voltage, V
Δv	= increment of velocity, m/s
η	= thrust efficiency

*PhD Student, ICARE-CNRS, alfredo.marianacci@cnrs-orleans.fr.
†Research Director, ICARE-CNRS, stephane.mazouffre@cnrs-orleans.fr.

I. Introduction

In recent years, Hall Thrusters (HTs) have attracted increasing interest in the spacecraft propulsion field due to particular advantages such as high Δv and total impulse, high thrust-to-power ratio and thrust density, and an efficiency in excess of 50 %. In addition, HTs are versatile, operating across a broad range of power levels. This versatility enables their application in both microsatellities and large satellites orbiting Earth, as well as in deep space missions. Despite their widespread use, certain internal physical mechanisms remain unknown, such as anomalous electron transport and plasma-wall interactions[1,2]. This means that building and optimization of a new thruster is complex, long in duration and mostly empirical whatever the size and power level. Unfortunately, sophisticated 2D Particle-In-Cell or fluid codes are of little help in the development of Hall Thrusters, since they are not predictive and necessitate empirical validation. In fact, the construction of simple models helped with design and data interpretation the development of Hall thrusters from its inception. The gradual development and maturation of these thrusters over the past decades have enabled the creation of methodologies based on scaling laws derived from reference thrusters along with database[3,4]. The inherent limitations of these approaches lie mainly in the equations on which they rely. Although useful for obtaining an initial estimate of thruster parameters and performances, the potential of 0-D scaling relations remains limited. The latter are in fact too simple to correctly reproduce the physics at play in HTs. Simplifying assumptions are indeed made on loss factors, plume divergence, thermal load, magnetic field topology, presence of multiply-charged ions, to name just a few examples[5]. All these approximations often result in large differences between predictions and performance measurements.

In this work, the problem was addressed through an approach that was no longer purely statistical but based on Machine Learning (ML) algorithms. The field of ML is very broad and includes complex algorithms covering a wide range of topics. In this study, supervised machine learning regression algorithms were analysed with the aim of determining continuous output values from input data[6]. Such an approach makes it possible to avoid in principle any kind of initial assumption and is disconnected from the physical relationships between the parameters involved[7]. The objective of this study was therefore to improve the scaling methodologies with a non-statistical approach, in order to assess the size and properties of HTs from requirements such as thrust and specific impulse. To achieve this, a validation of our database was carried out with the principal scaling laws derived over the years, see section II. In section III, the Machine Learning algorithm that is most appropriate for exploring the database is selected. In section IV, the selected model was used in an optimization method to determine the characteristics of a generic new Hall Thruster. Intuitive and straightforward graphs are shown in section V to validate the method. Finally, conclusions and perspectives are given in section VI.

II. Scaling laws: the statistical approach

II.A. Database

The data used in this study came from databases consisting of data collected since 2012 and stored in our laboratory, combined with an extensive search for new tests in the literature, sometimes supplemented by extrapolation of data from graphs provided in various scientific journal papers, doctoral theses and conference articles. The data collected concern Hall thrusters firing from 50 W up to 70 kW[8–36]. The complete database features 54 thrusters, with a total number of 3323 different operating conditions. The channel mean diameter is the parameter $d = \frac{1}{2}(d_{ext} + d_{int})$, where d_{ext} and d_{int} are the external and internal channel diameter, and $h = \frac{1}{2}(d_{ext} - d_{int})$ is the channel width. L is the discharge channel length. The quantities included in the database are given in Table 1. The database only contains information on stationary-plasma-thruster type HTs, meaning that TAL are not considered. Very little data is available on magnetically shielded thrusters, as it is a fairly recent technology. Most of the tests were performed with xenon as propellant. There is however data with krypton and argon, which allows to study the characteristics of thrusters according to the type of propellant. Xenon-fueled HTs cover 82 % of the database and Kripton-fueled thrusters 14 %. Unfortunately, the full set of characteristics is not available for each condition, especially geometrical characteristics. In fact, leaving aside the magnetic field and considering only complete sets, 1570 points are available for xenon, 268 for krypton and 80 for argon.

The parameters we considered cover wide ranges, as the database includes small thrusters and large thrusters with powers ranging from a few watts up to 70 kW, as can be seen in Figure 1, where the thrust T is plotted as a function of the input power P_d for the three propellants. It is worth noting that the database has

Table 1: Database parameters

PARAMETER	SYMBOL	UNIT
Model	-	-
Developer	-	-
Type	HT(SPT or TAL), MS(Magnetic shielded)	-
Propellant	Xe,Ar,Kr	-
Diameter	d	m
Height	h	m
Length	L	m
Mass flow rate	$\dot{m}_n$	kg/s
Voltage	U_d	V
Current	I_d	A
Power	P_d	W
Thrust	T	N
Specific impulse	I_{sp}	s
Thrust Efficiency	η	-
Magnetic field	B	G

79% of the tests for powers below 5 kW, as illustrated in the inset plot in Fig. 1. In short our database is particularly focused on small and medium size Hall thrusters.

II.B. Standard scaling laws

Statistical scaling methodologies typically depend on well-established and extensively tested thrusters. By leveraging these methodologies, proportional relationships between intensive and extensive parameters that characterize thrusters can be identified. Assuming the validity of similarity criteria, these relationships allow us to determine how parameter values change with variations in thruster scale. Since the advent of Hall Thrusters (HTs) in the 1970s, extensive research has been conducted in this field, focusing on varying one or more parameters while maintaining others constant.[37–42]. Recent works have considered the extraction of scaling relations from physical processes validated against a database[5,43].

In this work, the same approach is applied to the new database described above. In order to obtain the design characteristics of a new thruster, basically its geometrical characteristics, it is necessary to consider assumptions on losses and proportionality between channel height and diameter, as described in reference 5. Starting from mission requirements such as thrust and specific impulse, it is possible to develop an iterative procedure to determine the characteristics of a new thruster. As explained, laws are determined from physical processes and coefficients are found using databases. An usual relation for the thrust is:

$$T \propto hd \cdot \sqrt{U_d} \tag{1}$$

where h, d are the channel height and mean diameter, respectively.
Figure 2 illustrates Eq. 1. The data spread is due to the different mass flow rate values for tests made on the same thruster. A scaling relationship must also be defined for the mass flow rate and the specific impulse[44]. Using the established laws and linear fitting coefficients obtained from available data, one can estimate the

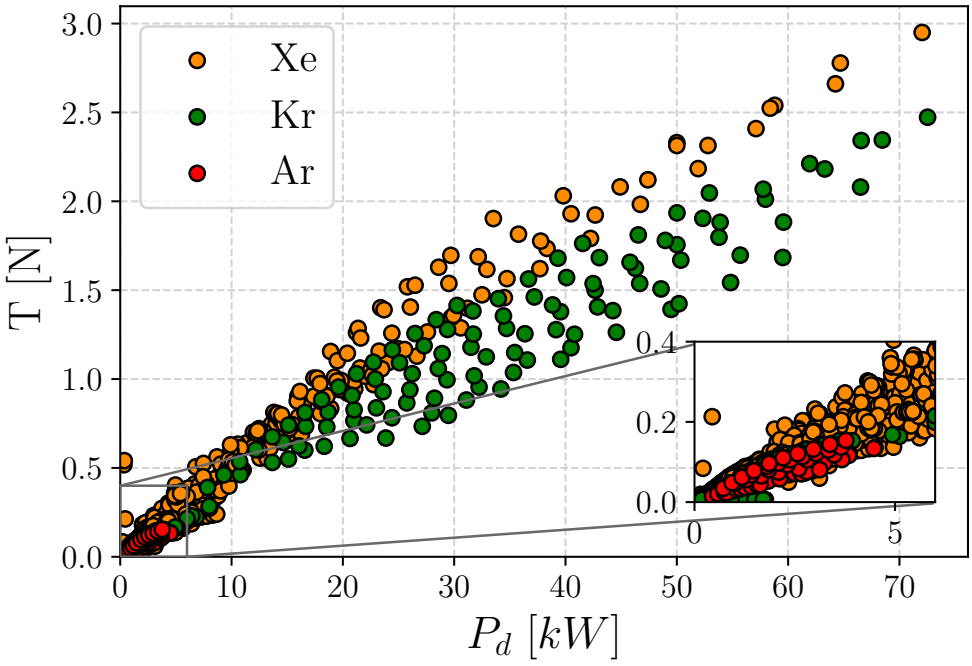

Figure 1: Thrust T as a function of the discharge power P_d. The zoom at the bottom represents the position of the majority of the data in the database

primary characteristics of a new thruster. This outlined procedure provides a preliminary insight into the necessary thruster geometry to achieve desired thrust and specific impulse parameters. However, a more

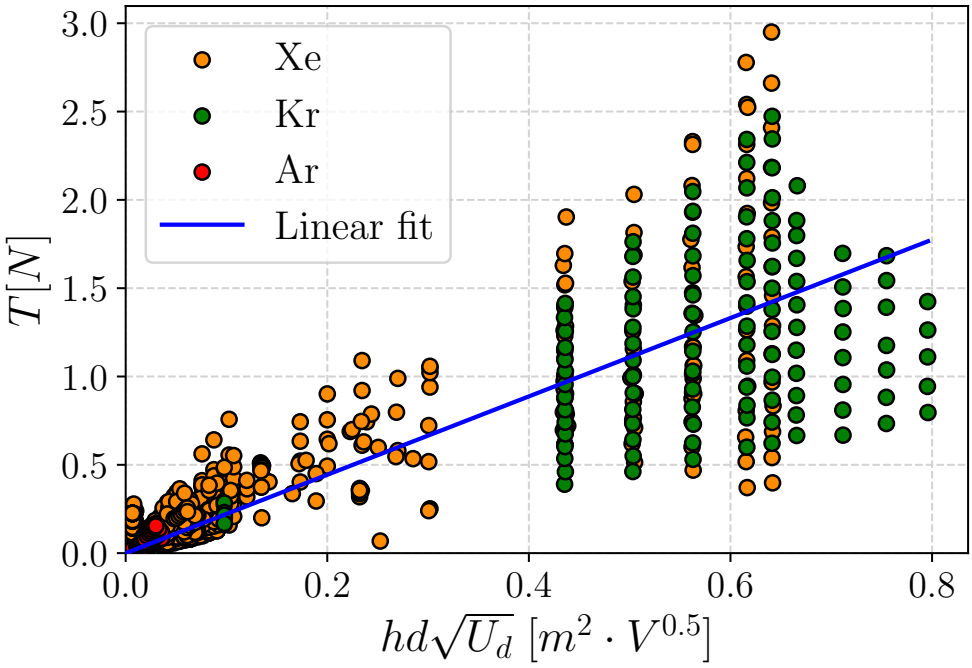

Figure 2: Thrust as a function of $hd \cdot \sqrt{U_d}$ and linear fit for Hall Thrusters in the database. Data are divided by type of propellant.

detailed analysis is needed, taking into account factors such as thermal stresses on walls, the presence of multiple charged ions, various forms of energy losses, and anomalous electron transport. The main drawbacks of the traditional approach lie in its neglect of magnetic field effects and its reliance on assumptions regarding geometric relationships.
In contrast, in the following approach which is based on machine learning algorithms applied to the database, the model will be left free of initial assumptions, although it still depends on the available data, and in the second analysis the magnetic field will be included in the fundamental parameters of the model.

III. Machine learning approach

III.A. Introduction

The machine learning approach is based on the determination of thrust and specific impulse by knowing the values of mass flow rate, discharge voltage, geometric dimensions (h,d,L) and, secondarily, magnetic field. The type of propellant was also added as a parameter in the model. Thrust and specific impulse are in fact the usual mission requirements, while all the others parameters are the targets. But algorithms are designed to obtain a continuous value from a set of inputs, usually greater in number than the desired outputs.

III.B. Data pre-processing

Prior to implementing machine learning algorithms, it is imperative to preprocess the available data. In fact, the range of values for the variables considered here is very different (e.g. mass flow rate in orders of 10−6 and voltage in orders of hundreds). Various methodologies exist to address this issue, necessitating careful consideration to select the most appropriate technique based on the data type. One prevalent method is standardization, which entails subtracting the mean value from each data point and scaling all values to achieve a mean of zero and a unit variance. However, it is important to note that standardization is generally suited for data with a Gaussian distribution, which may not correspond with the characteristics of our dataset. The standardization was therefore discarded. The method chosen is the normalization, where all the values are normalized inside a range from 0 to 1. Technically, between a value very close to 0 and a value very close to 1 to avoid singularity. The normalization law reads:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2}$$

where x is the actual value, x_{max} and x_{min} are the maximum and minimum value of each parameter. Scaling data exclusively using the training set is imperative in machine learning to ensure the integrity of the predictive model. Utilizing the entire dataset for scaling can lead to information leakage, where the model indirectly learns from the test data. This compromises the model's generalization capability, as predictions on unseen data are influenced by pre-exposed information, thus distorting the evaluation process.

Due to a relatively small amount of data in the database, it was not possible to use a simple splitting procedure between a large amount of data for training and a small amount for testing. The model selection was then performed using the Kfold-cross-validation procedure[45].

Table 2: Schematic representation of the work made by the cross-validation tool

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5		
TRAINING	TRAINING	TRAINING	TRAINING	TESTING	$\longrightarrow$ ERROR	
TRAINING	TRAINING	TRAINING	TESTING	TRAINING	$\longrightarrow$ ERROR	
TRAINING	TRAINING	TESTING	TRAINING	TRAINING	$\longrightarrow$ ERROR	$\rightarrow$ **MEAN ERROR**
TRAINING	TESTING	TRAINING	TRAINING	TRAINING	$\longrightarrow$ ERROR	
TESTING	TRAINING	TRAINING	TRAINING	TRAINING	$\longrightarrow$ ERROR	

Table 3: One-hot encoding for propellant

Type	O-HE	Xe	Kr	Ar
Xe	$\longrightarrow$	1	0	0
Kr	$\longrightarrow$	0	1	0
Ar	$\longrightarrow$	0	0	1

This technique, illustrated in Tab. 2, is based on dividing the available data into sub-folders, in this case five, where 4/5 of the data is used for training and 1/5 for testing, per each sub-folder. In this way, the model is trained five times instead of just one, and there are five mean errors between the target and the predicted values. Finally, a global mean error is calculated. There are several metrics to evaluate the performance of a model. In this work, the Mean Absolute Error (MAE) was chosen. It evaluates the modulus of the difference between the predicted value and the actual value, without taking direction into account. The choice was dictated by the fact that this type of metric is much less sensitive to outliers, i.e. data that shows a different trend from the general trend, usually due to measurement errors[7]. The expression of MAE[6] is the following:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

where n is the number of samples, y_i are the actual values and $\hat{y}_i$ are the predicted values. Furthermore, to enhance results clarity, percentage error was used as a comparative measure across different results, calculated according to Equation 4. This metric enables straightforward comparisons of predictive accuracy, offering a consistent and interpretable performance measure.

$$\%_{error} = \frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \tag{4}$$

A one-hot encoding (OHE) procedure[45] was applied to incorporate the propellant type into the model parameters, aiming to utilize this information for optimizing thruster characteristics. The propellant type was transformed into a three-dimensional vector, with the relevant component set to 1 and others to 0, as illustrated in Table 3.

III.C. Model selection

1. Introduction

The first step in determining a model capable of predicting thrust and specific impulse values by learning from the available data is to choose the most suitable machine learning algorithm. The selected algorithms are part of the supervised group, in the regression branch, as the aim is to determine continuous values[46]. The guide in determining the most suitable model is the database, as it is crucial to choose the algorithm that best fits the data. In fact, the outcomes can be quite different according to the model and to the hyperparameters inside each model.

The procedure started by considering different types of supervised machine learning algorithms, trying to cover the working range of the entire branch. Below are the algorithms initially chosen[45,46]:

- Linear Regression (LR)
- Decision Tree Regression (DTR)
- K-Nearest Neighbors Regression (KNR)
- Random Forest Regression (RFR)
- Gradient Boosting Regression (GBR)

- Bagging Regression (BR)

Each of these algorithms was tested on the database which, as previously explained, only has 1918 tests available due to missing data; 30% of data was set aside for the final validation of the chosen model, while 70% was used for training and testing with the cross validation procedure described above. Each model has its own hyperparameters, which are parameters of the algorithm class. They define its learning process. Algorithms have default hyperparameters, based on the most common functions used per each model[46]. In a first attempt, models were tested with them, to have an initial estimation of their performances. The results are presented in reference 44.

2. Model selection with hyperparameters tuning

A more efficient approach involves allowing the algorithm to determine the optimal hyperparameters for the specific dataset, rather than relying on default settings. This process, known as hyperparameter tuning[45], is commonly performed using GridSearchCV. GridSearchCV exhaustively tests all combinations of hyperparameters and evaluates their performance. However, this method is computationally intensive, especially when numerous hyperparameter combinations and models are involved, leading to exponential increases in computational time. Consequently, GridSearchCV was deemed impractical for our purposes.
To expedite the identification of the best model and hyperparameters, we employed the HalvingGridSearchCV tool. This innovative machine learning method evaluates hyperparameter combinations on a small subset of the data, referred to as resources. Iteratively, a subset of these combinations, called candidates, are selected for evaluation with an increasing amount of resources. The number of iterations is determined by the sample size and the number of parameters involved. The combination that progresses through all iterations with the smallest prediction error is considered optimal. HalvingGridSearchCV significantly reduces computational time compared to GridSearchCV. Throughout this process, the algorithm performs cross-validation for each combination, as detailed in Sec. III.B. This approach integrates fold-based evaluation using Mean Absolute Error (MAE) with the search for optimal hyperparameters, aiming to minimize prediction error. Table 4 shows the results obtained with the tuning procedure. KNR, RFR and GBR work in a high-performance manner, with a very low Mean Absolute Error. As a final step, all the models were evaluated on the test set left apart. Figure 3 shows the comparison between the models. Absolute errors are counted between each test and its corresponding predicted value. The performances of KNR, RFR and GBR are almost similar, but the distribution curve of the Gradient Boosting seems to drop first to null values of absolute errors.

Table 4: Hyperparameters tuning and evaluation of the six models

Model	Best hyperparameters	MAE
LR	(' fit_intercept ': True, ' positive ': False)	0.019
DTR	(' criterion ': 'absolute_error', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 3, 'random_state': 10, 'splitter ': 'best')	0.009
KNR	('algorithm': 'kd_tree', 'n_neighbors': 3, 'weights': 'distance')	0.007
RFR	(' criterion ': 'absolute_error', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200, 'random_state': 500)	0.007
GBR	(' criterion ': 'squared_error', 'learning_rate': 0.1, ' loss ': 'absolute_error', 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 200, 'random_state': 500)	0.006
BR	('estimator': DecisionTreeRegressor(), 'max_features': 8, 'max_samples': 10, 'n_estimators': 200, 'random_state': 500)	0.02

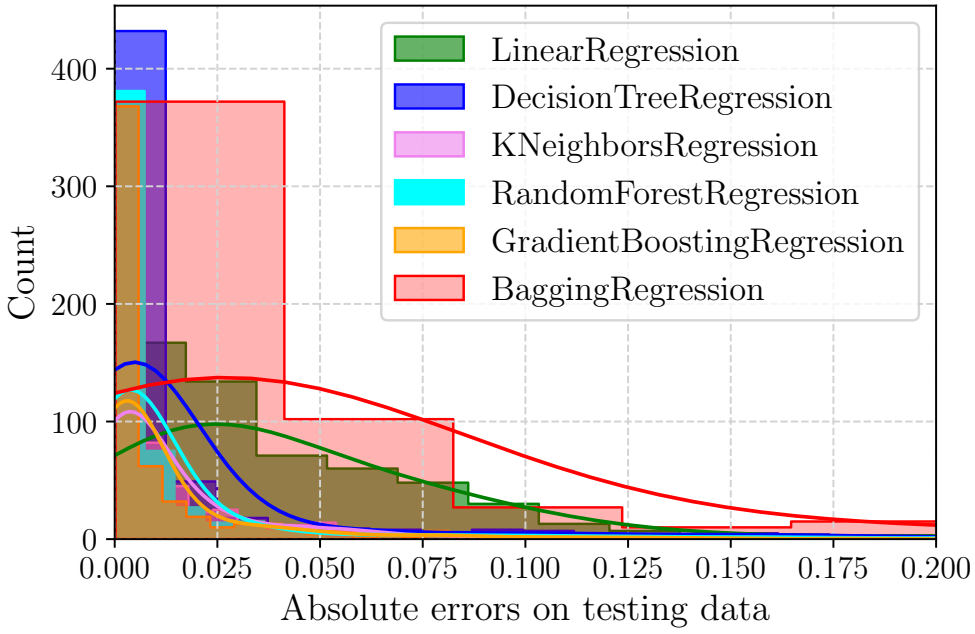

Figure 3: Models comparison in terms of absolute errors

Moreover its histogram is more concentrated on the extreme left, where the errors are close to zero. Thus the model chosen as the best one for our data is the Gradient Boosting Regression.

III.D. Gradient Boosting Regression

1. Building of trees and prediction

The Gradient Boosting Regression (GBR) is an ensemble algorithm that puts together weaker models to perform better as a whole[46]. It is based on decision trees, whose number is decided in the hyperparameters tuning phase. In this analysis, the model generates 200 sequential trees. Gradient Boosting works on the gradient of the Loss function, minimizing the errors between predicted and actual data[47]. The model is initialized with an initial value:

$$F_0(x) = \arg\min_{\gamma}[\sum_{i=1}^{n} L(y_i, \gamma)] \tag{5}$$

where F_0 is the initial value of the model and it is equal to the value of γ that minimizes the loss function $\sum_{i=1}^{n} L(y_i, \gamma)$, represented by the Mean Absolute Error, where y_i are the actual value, γ the predicted values and n the number of tests. To find therefore the value of γ, a derivative of the loss function is taken and is set to zero:

$$\frac{dL}{d\gamma} = \frac{d\sum_{i=1}^{n} |y_i - \gamma|}{d\gamma} = -\sum_{i=1}^{n} sign(y_i - \gamma) = 0 \tag{6}$$

The sign function is either -1, 0 or 1 and no matter how distant the target is from the current prediction. The model is trained on just the direction, without the magnitude. Considering the latter in fact the computations are easily skewed by outliers. Solving the equation above, we obtain that the value of γ that minimizes the loss function is the median of the output values in the training dataset. It is worth remind that we have both thrust and specific impulse as outputs of the model, therefore the GBR is trained on each output separately but the performance evaluation is a mean of the performance evaluation of the model on the two separate outputs. Once $F_0(x)$ is calculated, the GBR generates decision trees. Each leaf is created by splitting the training data through values greater or lower of a certain threshold.

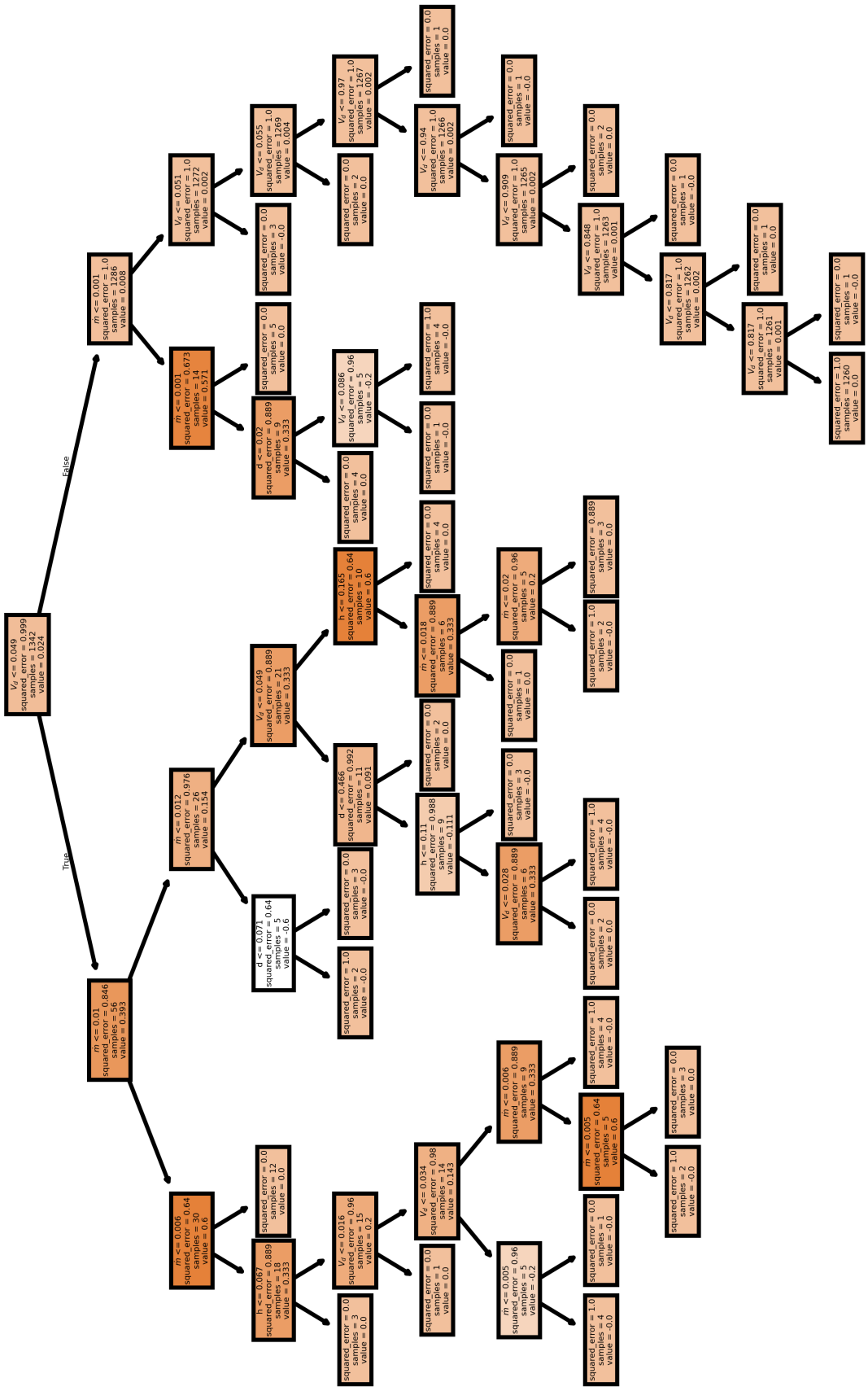

Figure 4: Last decision tree created by GBR

The algorithm itself is able to analyse the consequences of each splitting and from there to decide for which threshold value the results are best in terms of prediction. The evaluation of the split is made through the *criterion* function, see Table 4. Since the magnetic field is not considered in this first analysis, our inputs are eight. The GBR is capable of dividing the training set by acting on all parameters simultaneously, finding the optimal threshold values for best performance. The mechanism is therefore quite complex. The same formula is then applied for each leaf, where the output is in fact:

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma) \tag{7}$$

where j is the leaf, m is the tree and R_{ij} are the output of the leaves. For the same reason as before, the output in each leaf is exactly the median value. Finally, this value is first multiplied by a α, the learning rate, that controls the speed of the process and then is added to the previous one, that in this case was $F_0(x)$:

$$F_m(x) = F_{m-1}(x) + \alpha \sum_{j=1}^{J_m} \gamma_{jm} \tag{8}$$

Once the first decision tree is built on the training data, it has to be evaluated on the testing set through the MAE. It is important to remind that the Gradient Boosting Regression builds sequential decision trees, thus the first one gives the worst prediction and step by step, i.e. tree by tree, the performance is improved. Figure 4 shows the final decision tree generated by the GBR model in the training phase. Each box contains information about the splitting procedure, but the purpose is to present a complete tree to help the reader understand how the model works.

2. Evaluation of the model

Dividing the data based on specific parameter values can provide insights into the significance of these parameters within the model. Specifically, if the model's performance exhibits considerable variation when the threshold value of a parameter is altered, it underscores the parameter's importance. Figure 5 illustrates the relative importance of various parameters employed in the decision tree construction process. The histograms are normalized to a maximum value of 1. Among the inputs, the mass flow rate emerges as the most critical factor in the model: variations in this parameter result in significant changes in the model's predictions. This phenomenon can be attributed to the intrinsic relationship between the thruster's geometrical characteristics and the mass flow rate.

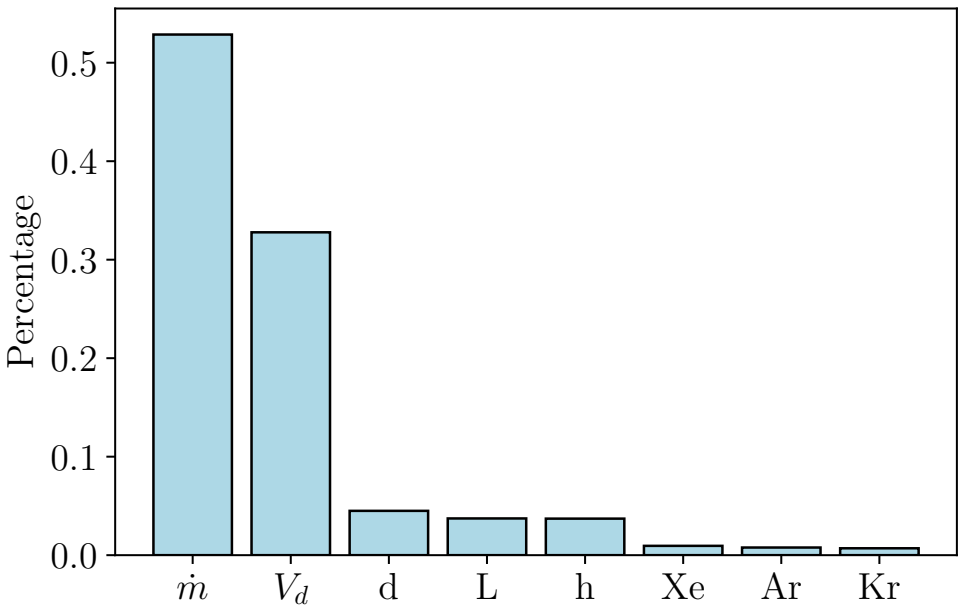

Figure 5: Features importance in the training phase of the Gradient Boosting

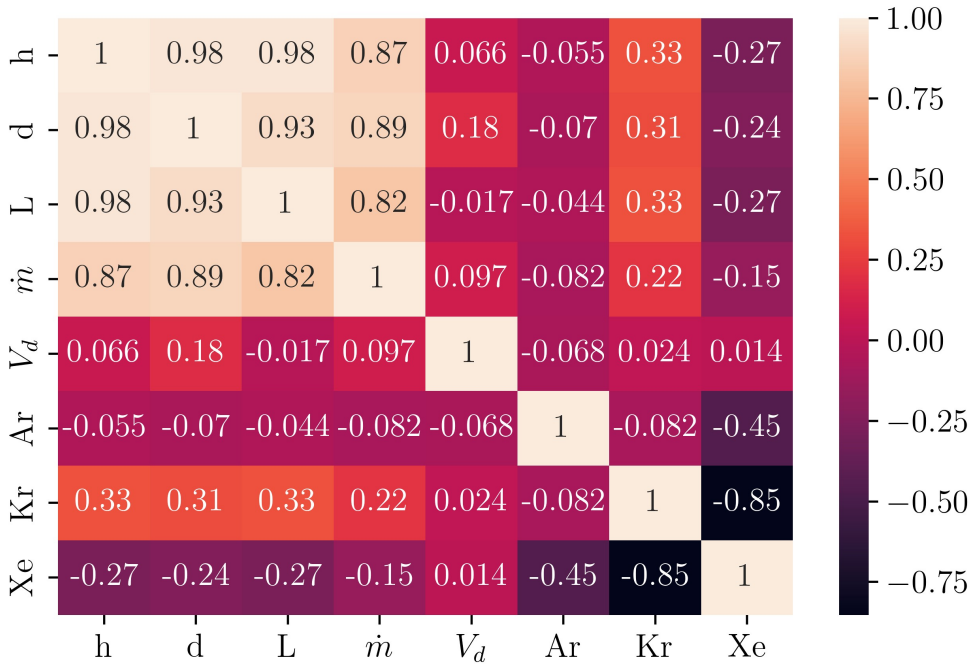

Figure 6: Correlation matrix between inputs parameters in GBR

Alterations in the mass flow rate inherently lead to modifications in other parameters. Additionally, the interdependence among the geometrical features complicates the model's ability to isolate their individual contributions to the prediction of thrust and specific impulse. Consequently, the model attributes greater importance to parameters with clearer independence, such as mass flow rate and voltage. This interdependence can be further elucidated through a correlation matrix, as shown in Figure 6. The correlation matrix helps in understanding the correlation between variables. It measures the covariance, i.e. how much a parameter varies if another one is changed. It is normalized between -1 and 1. Usually, if the value is between 0.3 and 1 there is a direct correlation between the two parameters. In fact, see Fig. 6, the geometrical features have an high value of correlation among themselves and with mass flow rate. If the values are between -0.3 and -1, there is an inverse correlation, while if it is between -0.3 and 0.3 there is no correlation. In our case, the only independent variable is the discharge voltage U_d and the propellants are dependent among themselves.

As explained previously, the model is evaluated on the testing dataset. In Figure 7 is represented the prediction of the two outputs made by GBR after all the 200 trees are built.
The collective behavior is good enough, since the model, represented by the blue crosses, is effective in prediction of thrust and specific impulse values in the entire range. In the upper right prediction is weaker due to the lack of data for high power thrusters in the database. To better understand the improvement in performance of the Gradient Boosting algorithm, it is useful to compare it to a thrust prediction model based on a scaling law, see Eq. (1), which is recalled for clarity:

$$T \propto hd \cdot \sqrt{U_d} \tag{9}$$

Figure 8 shows the thrust predictions of the Gradient Boosting model for the thrust as a function of diameter, while Figure 9 shows the predictions of the linear scaling law described by the Eq. (1). The vertical distribution of data for a given diameter value is linked to changes of mass flow rate values for tests performed with a given Hall thruster (d fixed then). It is worth noting that the two models were not tested on the same data, because the splitting of data in training and testing is done randomly. The GB model in Fig. 8 was tested on 576 samples, while the scaling law model in Fig. 9 on 493 samples. The difference in prediction quality is obvious. In Fig. 8 the model predicts thrust values very accurately, whereas in Fig. 9 agreement between data and calculation is poor. Moreover, the model follows a parabolic trend due to the quadratic dependence of thrust on diameter and fails to capture the punctual distribution of the test data. Figure 10

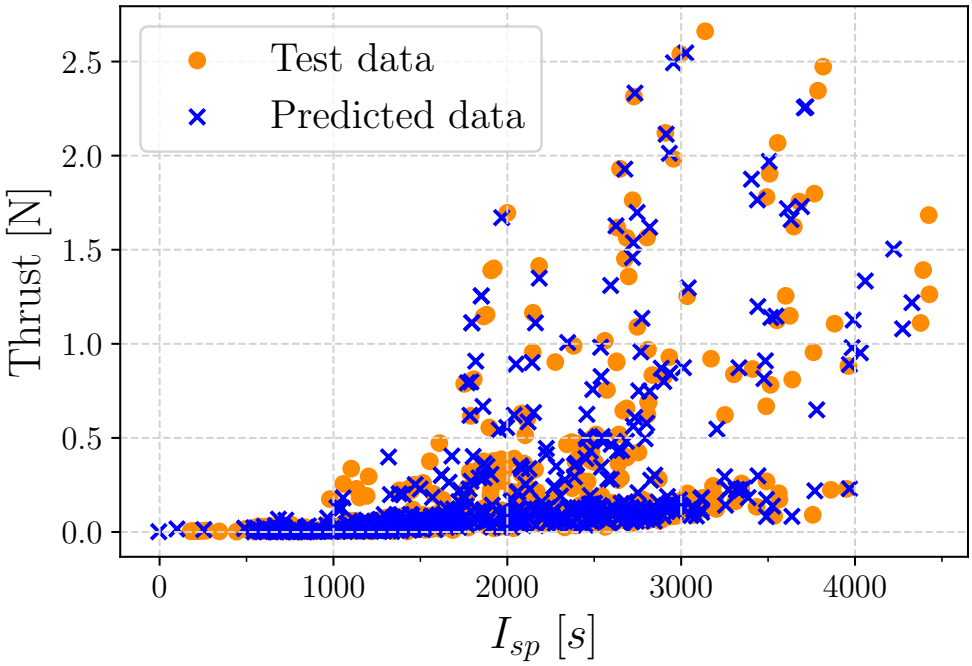

Figure 7: Model prediction of thrust and specific impulse on testing data

illustrates the absolute errors of the two models, in a graph similar to the one in Fig. 3. The errors of the Gradient Boosting are very small and the density curve goes fast to zero. On the contrary, the scaling law errors are almost two orders of magnitude greater.

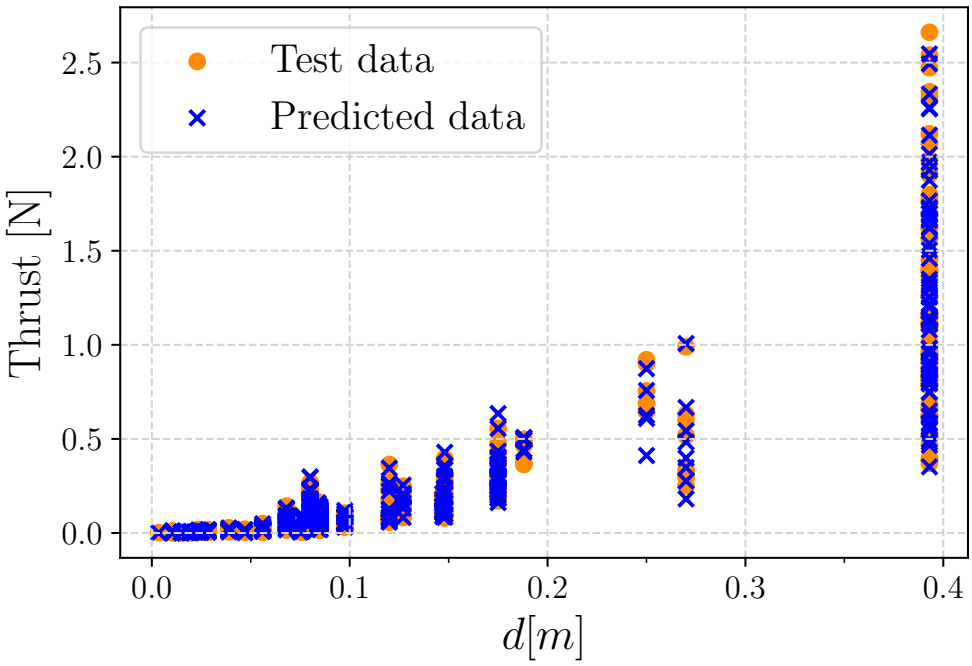

Figure 8: Gradient Boosting Regression predictions of the thrust against mean diameter for the testing set.

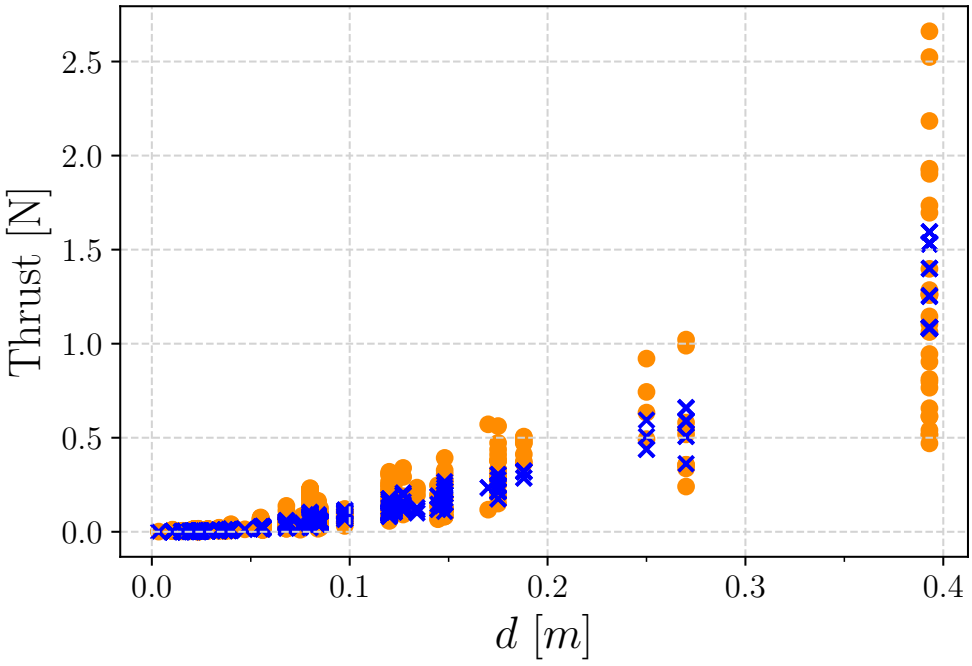

Figure 9: Scaling law predictions of the thrust with respect to mean diameter values for the testing set.

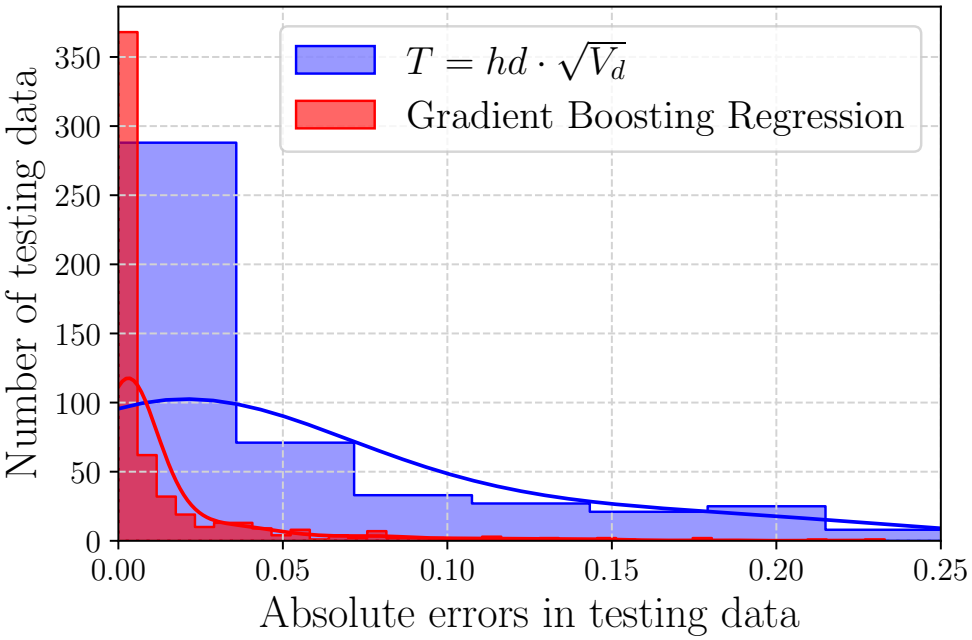

Figure 10: Comparison of absolute errors for the GBR model and the scaling relation when predicting the $T = f(d)$ behavior.

IV. Design of a new Hall thruster

IV.A. Introduction

As detailed in section III.A, the development of a new thruster begins with defining certain parameters based on mission requirements. Typically, thrust and specific impulse serve as the initial design parameters, which guide the determination of other thruster characteristics such as geometrical dimensions, voltage, mass flow rate, and magnetic field. The Gradient Boosting model demonstrates a high level of accuracy in predicting thrust and specific impulse values when given the other parameters. However, when the thruster design is considered, a back-propagation problem arises. It is therefore necessary to reconsider the relationship generation process underlying the creation of decision trees in order to extrapolate information about the various parameters when a particular output is desired. Given that Gradient Boosting is constructed with sequential decision trees, each of which splits the parameters in a different manner in order to identify relations between them, the challenge is practically unfeasible. Consequently, the problem must be addressed in a different way. The solution was find in an analytical approach based on an optimization problem. The database was initially divided based on the propellant type. Furthermore, the magnetic field was inserted secondarily in the model, as described in Sec. IV.D, as it plays a key role in the thrust process although very little data is available, resulting in reduced model performance.

IV.B. Analytical approach

The analytical approach to determining parameter values for achieving a specified thrust and specific impulse involves an optimization algorithm. The procedure begins with providing a desired output. Subsequently, the Gradient Boosting model utilizes initial random inputs and computes the corresponding thrust and specific impulse values. The error between the computed output and the desired output is then evaluated. The optimization algorithm's objective is to adjust the input parameters to minimize the loss function, defined as the absolute error between the two sets of output values. When the error is minimized, the model identifies the optimal parameters that yield the required outputs. This methodology is illustrated in Figure 11, where the box 'Differential Evolution' denotes the optimization algorithm.
The choice of the optimization algorithm is mainly based on the loss function minimization procedure. In fact, it is a stochastic algorithm that does not require the optimization problem to be differentiable, as is required by other classic algorithms as gradient descent [48]. Since the GBR consists of sequential decision trees, it is impossible to determine how the model depends on the various parameters individually and therefore it is not feasible to calculate their partial derivatives. Instead, Differential Evolution optimizes the problem iteratively, trying to improve a candidate solution[49]. The Differential Evolution (DE) algorithm operates in four main steps: initialization, mutation, crossover, and selection. Initially, random values are generated for each parameter to create vectors. In the mutation phase, mutant vectors are generated by combining three randomly chosen vectors with a scaling factor. During the crossover phase, the mutant vectors are evaluated, and if their performance exceeds a certain threshold, they replace the original vectors. This process continues iteratively until a specified threshold or a predefined number of iterations is reached [44,49,50]. Bounds within which the algorithm is guided in searching the optimal inputs are defined. The database was divided in subgroups and the DE was looking for parameters values only in the subgroup containing values able to generate that desired output. Different algorithm were developed for the three propellants and also for the consideration or not of the magnetic field.

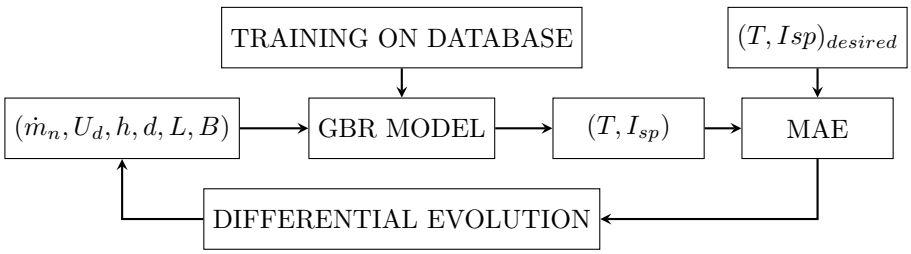

Figure 11: Flowchart of the analytical approach

Table 5: Desired outputs

	$T[N]$	$I_{sp}[s]$
Example A	0.04	1000
Example B	1.7	3200

Naturally, the more the outputs are far from values present in the database, the worst are the performances of Differential Evolution algorithm, struggling to converge. However, in the various cases we tried, the algorithm was performing quite well, as shown in section IV.C.

IV.C. Final example

To clarify the working of the approach, it is beneficial to provide examples of determining the characteristics of two new thrusters: a potential low-power Hall Thruster, designated as Example A, and a high-power Hall Thruster, designated as Example B. Both thrusters utilize xenon as the propellant. The requirements in terms of thrust and specific impulse are reported in Table 5. The characteristics of the thrusters are obtained with the analytical approach, determining the mass flow rate $\dot{m}_n$, discharge voltage U_d, diameter d, height h and length L of the channel. In section IV.D, the magnetic field B is also added. The approach is based on the application of the diagram shown in Fig. 11. It is noteworthy that the mean absolute errors are computed between scaled outputs, as the Gradient Boosting Regressor (GBR) operates on a scaled set of inputs. Subsequently, the parameters are unscaled to be presented in the appropriate format. Moreover, to enhance comprehension of the disparity between desired and predicted values, the error is expressed as a percentage and delineated separately for the two outputs, notwithstanding the algorithm's optimization of their average. In the case of Example A, the optimization procedure requires 82 iterations to ascertain the minimum of the loss function, as denoted by the mean absolute error. The attained results exhibit a high degree of accuracy, as evidenced by the negligible percentage errors observed in the Table 6, indicative of minimal disparity between the desired and actual values. Additionally, the inclusion of data pertaining to an existing thruster from the database within the Table 6 facilitates comparative analysis and aids in understanding their physical implications.
Regarding Example B, despite encountering slightly higher errors, particularly in specific impulse, the outcomes remain promising. The optimized input parameters exhibit values coherent with physical expectations. However, it is notable that the efficacy of the differential evolution algorithm is contingent upon data availability. When operating within regions abundant in data, the algorithm demonstrates robust generalization capabilities; conversely, its convergence falters in data-scarce domains, resulting in an amplification of the discrepancy between actual and desired outputs.

IV.D. Magnetic field

The magnetic field B has been addressed separately due to its pivotal role in thruster performance[1]. However, the notably constrained data availability precludes its inclusion as a primary parameter in the model. Specifically, the magnetic field consideration is based on a limited dataset, encompassing merely 1000 tests, in contrast to 1918 tests available when omitting this factor. Moreover, the provided value solely represents the maximum intensity along the channel axis, neglecting the comprehensive topology of the magnetic field. Consequently, the model's efficacy diminishes, as evident in Figure 12 and Figure 13. Notably, discrepancies between model predictions and actual data become apparent when compared to scenarios excluding B, as depicted in Fig. 7 and Fig. 8.

Table 6: Parameters values obtained with the analytical approach

	Example A	Real LP HT	Example B	Real HP HT
Prop.	Xe	Xe	Xe	Xe
$\dot{m}_n\ [mg/s]$	3.33	2.54	61.18	62.6
$U_d\ [V]$	202.66	200	710.37	650
$d\ [mm]$	60.44	75	393	393
$h\ [mm]$	10	10	59.95	64
$L\ [mm]$	14.61	15	73.98	83
$T\ \%_{err}$	0.64 %	-	1.44 %	-
$I_{sp}\ \%_{err}$	0.02 %	-	5.609 %	-
iter	82	-	28	-
$T_{desired}\ [N]$	0.04	-	1.7	-
$I_{sp\,desired}\ [s]$	1000	-	3200	-
$T_{actual}\ [N]$	0.0397	0.027	1.724	1.791
$I_{sp\,actual}\ [s]$	999.8	1004.49	3024.74	2917

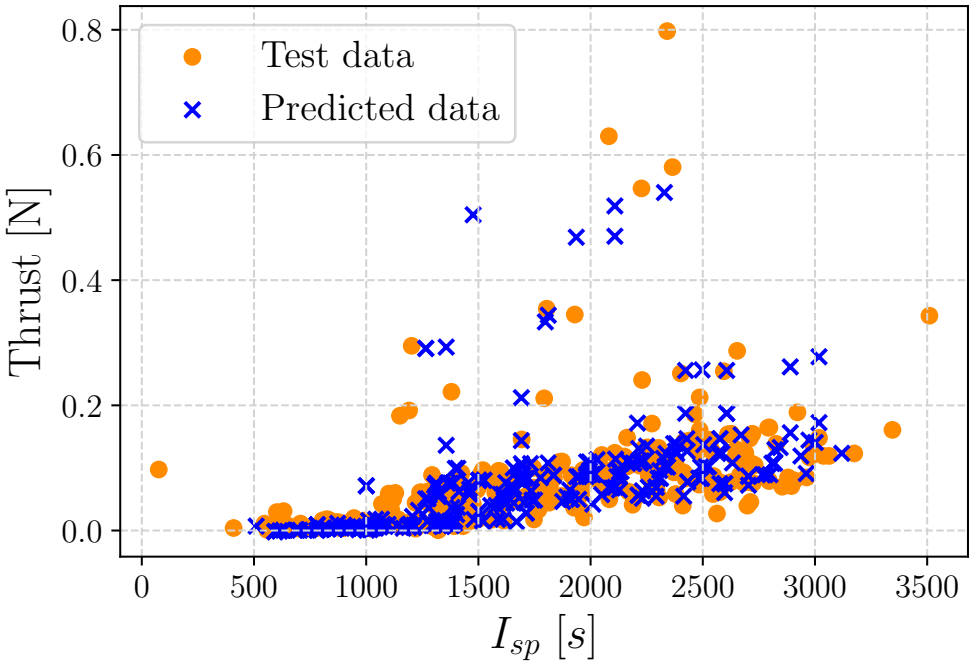

Figure 12: Model prediction of thrust and specific impulse on data including the magnetic field

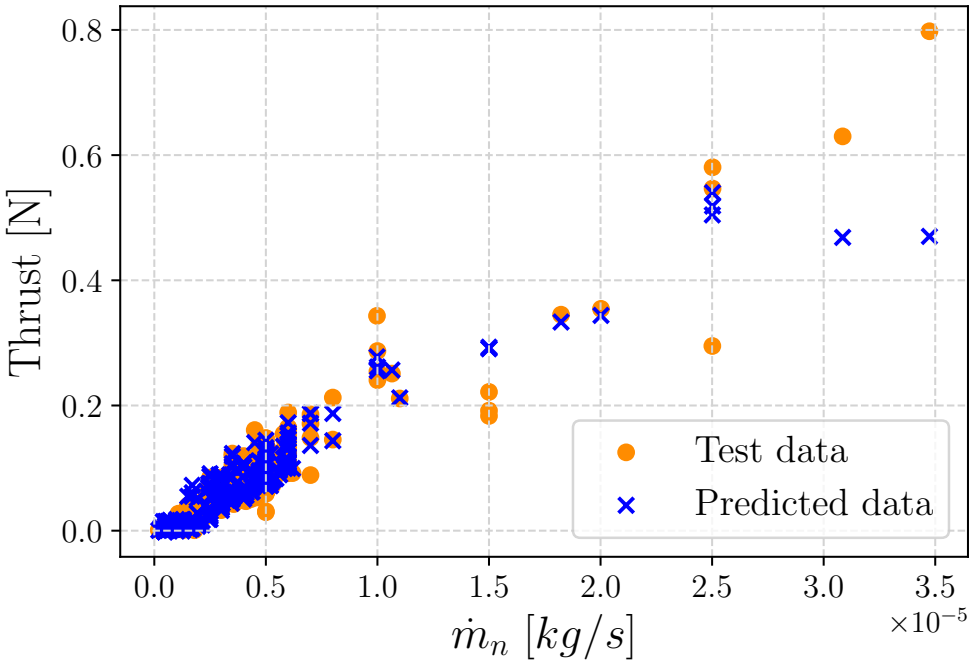

Figure 13: Predicted thrust as a function of the mass flow rate for data that include magnetic field values

However, an attempt was made introducing the magnetic field as a model parameter for examples A and B previously described. Table 7 shows the optimal parameters values found with the optimization. It can be observed that the errors committed by the algorithm are greater than in the case without B. This is to be expected, given the lack of data available for both training and the optimization itself.

Table 7: Parameters values obtained with the analytical approach including the magnetic field

	Example A	Real LP HT	Example B	Real HP HT
Prop.	Xe	Xe	Xe	Xe
$\dot{m}_n\ [mg/s]$	3.40	2.54	44.48	40.78
$U_d\ [V]$	208.39	200	613.29	500
$d\ [mm]$	84.86	75	270	270
$h\ [mm]$	9.7	10	50	50
$L\ [mm]$	14.87	15	70	70
$B\ [G]$	108.32	130	106.77	130
$T\ \%_{err}$	7.58 %	-	41.07 %	-
$I_{sp}\ \%_{err}$	0.42 %	-	13.53 %	-
iter	39	-	11	-
$T_{desired}\ [N]$	0.04	-	1.7	-
$I_{sp\,desired}\ [s]$	1000	-	3200	-
$T_{actual}\ [N]$	0.037	0.027	1.002	1.058
$I_{sp\,actual}\ [s]$	1003.9	1004.49	2777	2644.01

In order to facilitate a more accurate comparison between the two cases, a mean relative error has been calculated between the percentage errors on thrust and specific impulse. The approach that considers the magnetic field has therefore yielded a percentage increasing error of approximately 93.39 % for Example A, while for Example B it is approximately 77.52 %. The performance has deteriorated, particularly in relation to geometric dimensions. The optimization, constrained by imposed physical limits, lacks sufficient data to identify optimal inputs and, consequently, fails to generalize effectively.

V. Graphical approach

As a final proof, 2-D graphs containing information on the thrusters present in the database were extracted. In fact, data regarding mass flow rate, voltage discharge, diameter, thrust, specific impulse and magnetic field were gathered inside 2-D maps to help the validation of the procedure described in the section IV. The graphs were obtained dividing the database for propellant type first, and adding the magnetic field values in a second step, to follow the same approach as before. The database was divided in subgroups based on parameters values. Considering one parameter at a time, tests with neighbouring values of that parameter were combined into a group characterized by the average value of the parameter considered for the split, as we can see in Figure 14, where specific impulse and thrust are the two model's outputs and the mission requirements, on the x axis and y axis respectively, and data is subdivided based on mass flow rate values. The values indicated in the legend are the average ones for each subgroup. In the case of Fig. 14, a linear pattern is seen between thrust and specific impulse for constant mass flow rate values, confirming the theory. When considering the discharge voltage, the same subdivision procedure gives a parabolic trend for constant voltage values of thrust versus specific impulse. At this point, these trends can be extrapolated and replaced by straight lines for mass flow rate and parabolas for voltage. Figure 15 and Figure 16 represent in fact data on tests made with xenon as propellant and they give information about mass flow rate, discharge voltage, diameter, thrust and specific impulse. The diameter values are presented as scatter plot. The averages obtained with the procedure mentioned above are shown. Example A and Example B discussed in IV.C are presented as a red triangle. Figure 15 provides a detailed zoomed-in view of the lower section of Figure 16, where the low-power Hall thrusters (HT) are predominantly located. These figures offer valuable insights into the primary characteristics of Hall thrusters, facilitating an initial estimation based on thrust and specific impulse requirements.

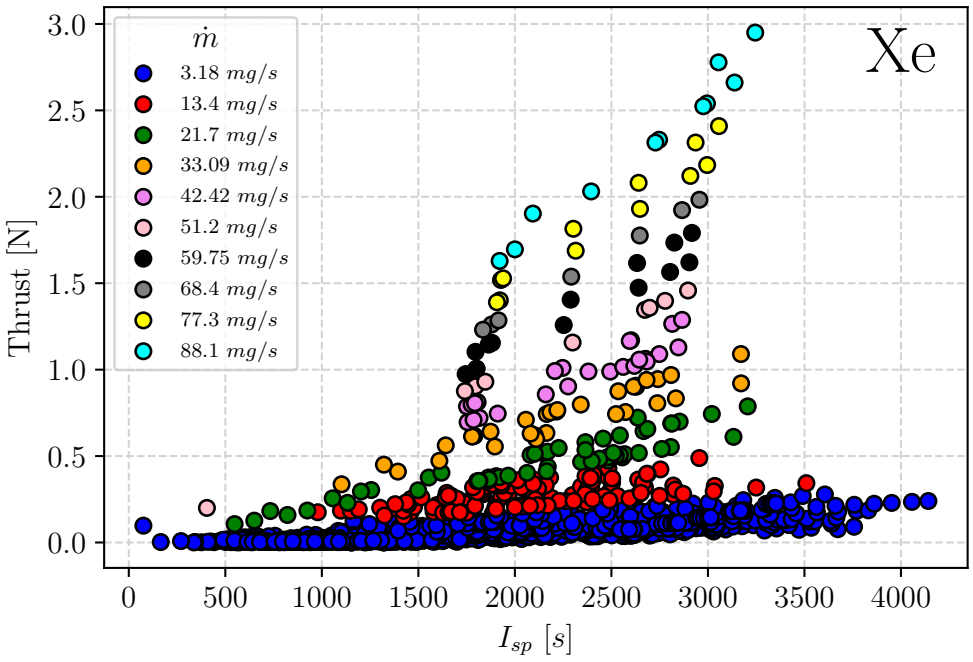

Figure 14: Thrust against specific impulse values grouped by mass flow rate values in tests made with xenon.

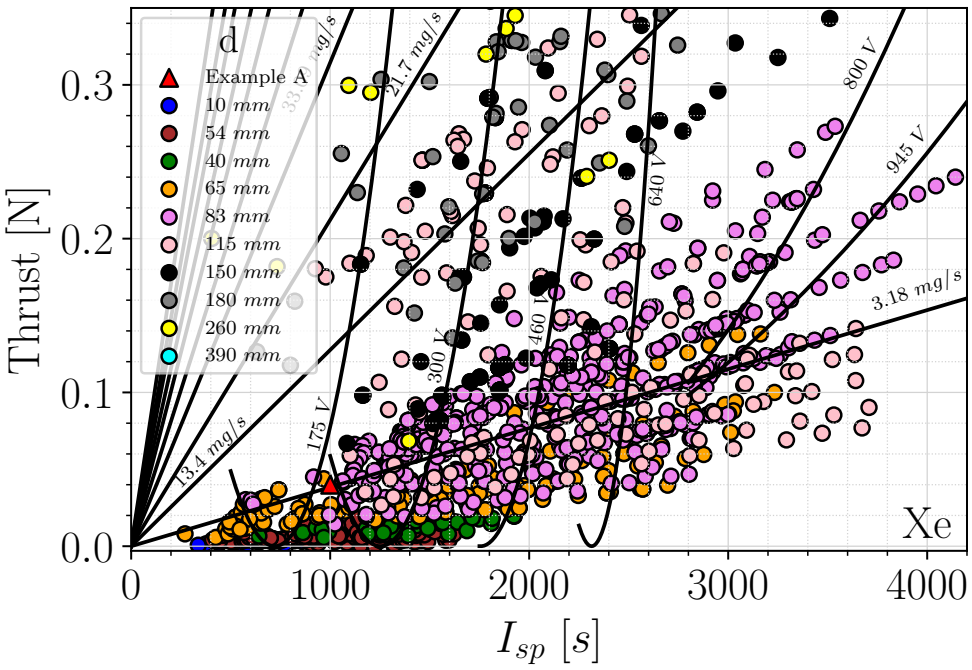

Figure 15: 2-D map of xenon tests present in the database with info on T, Isp, $\dot{m}_n$, U_d and d. Example A is also shown.

Examples A and B on the graphs serve as clear evidence of the previously explained procedure, demonstrating that the optimization-derived values align well with the test data available in the database.

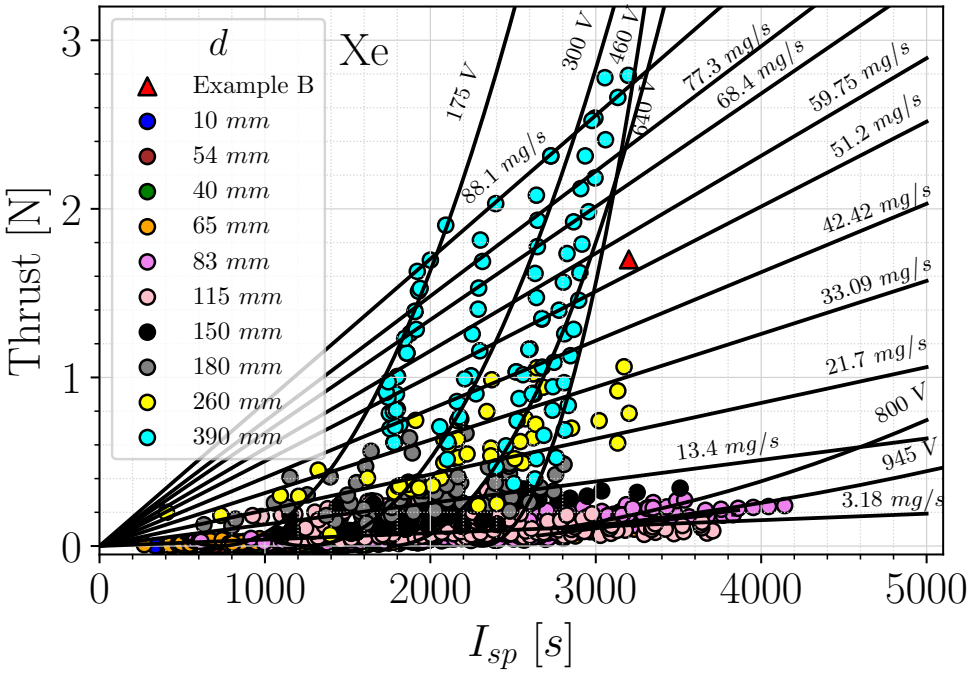

Figure 16: 2-D map of xenon tests present in the database with info on T, Isp, $\dot{m}_n$, U_d and d. Example B is also shown.

Similar graphical analyses can be generated for different propellants, such as Krypton and Argon, and can incorporate the magnetic field.

VI. Conclusion

The objective of this research was to develop a model capable of determining the geometrical characteristics and operating parameters of a novel Hall thruster, utilizing Machine Learning algorithms. A comprehensive database comprising data from 54 thrusters was established, incorporating information from existing databases and scholarly publications. The database was initially validated using a statistical methodology based on scaling laws, which have been thoroughly explored in previous literature. A machine learning-based model selection was then performed on the same dataset, resulting the Gradient Boosting Regressor being identified as the optimal algorithm in terms of mean absolute error. This model demonstrated high efficiency in predicting thrust and specific impulse values, particularly when the magnetic field was excluded from the input parameters. The inclusion of the magnetic field significantly reduced the available data, thereby diminishing the model's performance. By learning the intricate relationships among the various parameters, the Machine Learning model adeptly integrates all parameters, learning relationships that enable the accurate determination of thrust and specific impulse values for previously unseen data. Thus, the model showcases its potential for accurate and reliable performance in the design and analysis of Hall thrusters. The primary advantage of this model lies in its ability to eliminate the need for initial assumptions regarding loss factors or dependencies between geometric parameters, relying entirely on empirical data. Although the data used for the model is derived from tests on thrusters built using traditional approaches, the model effectively identifies dependencies between parameters, particularly geometric ones. The model performs well within the range of available training data; however, its extrapolation accuracy decreases in regions with sparse or absent data. The Gradient Boosting model was meticulously constructed and demonstrates robustness in predictions for unseen data. However, the optimization algorithm's heavy reliance on available data results in a significant increase in error when optimizing in data-sparse regions. Despite this limitation, the adoption of this algorithm was essential for achieving the objective of determining the characteristics of new Hall thrusters based on a supervised Machine Learning model. Future research should focus on reducing the constraints applied during optimization to enable the use of a broader dataset while ensuring that the solutions remain physically feasible. The incorporation of neural networks might also mitigate issues associated with backpropagation. This study demonstrates the significant potential of a Machine Learning-based approach in defining scaling laws and, uniquely, in predicting the geometric characteristics and operating points of a Hall thruster based on mission requirements. The approach developed and validated here, though complex, remains accessible and relatively straightforward to implement, given the availability of tools like Gradient Boosting Regression and tree generation in various versions and languages. Ultimately, this work underscores the importance of the quantity and quality of data in achieving accurate and reliable results. Thus, it is crucial to generate extensive databases covering a wide power range, either through long and costly measurements or through numerical simulations, to aid engineers in developing the next generation of Hall thrusters.

References

[1] Jean-Pierre Boeuf. Tutorial: Physics and modeling of hall thrusters. *Journal of Applied Physics*, 121(1), 2017.

[2] Stéphane Mazouffre. Electric propulsion for satellites and spacecraft: established technologies and novel approaches. *Plasma Sources Science and Technology*, 25(3):033002, 2016.

[3] Mariano Andrenucci, Leonardo Biagioni, Salvo Marcuccio, and Fabrizio Paganucci. Fundamental scaling laws for electric propulsion concepts. In *28th International Electric Propulsion Conference*, page 1721, 2003.

[4] Andrey A Shagayda. On scaling of hall effect thrusters. *IEEE Transactions on Plasma Science*, 43(1):12–28, 2014.

[5] Käthe Dannenmayer and Stéphane Mazouffre. Elementary scaling relations for hall effect thrusters. *Journal of Propulsion and Power*, 27(1):236–245, 2011.

[6] Andriy Burkov. *The hundred-page machine learning book*, volume 1. Andriy Burkov Quebec City, QC, Canada, n.d., 1 edition, 2019.

[7] Yegor V Plyashkov, Andrey A Shagayda, Dmitrii A Kravchenko, Alexander S Lovtsov, and Fedor D Ratnikov. On scaling of hall-effect thrusters using neural nets. *Journal of Propulsion and Power*, 38(6):935–944, 2022.

[8] James J. Szabo, Rachel Tedrake, Emily Metivier, Surjeet Paintal, and Zachary Taillefer. Characterization of a One Hundred Watt, Long Lifetime Hall Effect Thruster for Small Spacecraft. In *53rd AIAA/SAE/ASEE Joint Propulsion Conference*, Atlanta, GA, July 2017. American Institute of Aeronautics and Astronautics.

[9] W.A. Hargus and M.R. Nakles. Ion Velocity Measurements Within the Acceleration Channel of a Low-Power Hall Thruster. *IEEE Transactions on Plasma Science*, 36(5):1989–1997, October 2008.

[10] James J. Szabo, Rachel Tedrake, Emily Metivier, Surjeet Paintal, and Zachary Taillefer. Characterization of a One Hundred Watt, Long Lifetime Hall Effect Thruster for Small Spacecraft. In *53rd AIAA/SAE/ASEE Joint Propulsion Conference*, Atlanta, GA, July 2017. American Institute of Aeronautics and Astronautics.

[11] W.A. Hargus and M.R. Nakles. Ion Velocity Measurements Within the Acceleration Channel of a Low-Power Hall Thruster. *IEEE Transactions on Plasma Science*, 36(5):1989–1997, October 2008.

[12] Kristi De Grys, Benjamin Welander, John Dimicco, Scott Wenzel, Bob Kay, Vadim Khayms, and Jonathan Paisley. 4.5 Kw Hall Thruster System Qualification Status. In *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Tucson, Arizona, July 2005. American Institute of Aeronautics and Astronautics.

[13] K. De Grys, C. Rayburn, and J. Haas. Study of Power Loss Mechanisms in BPT-4000 Hall Thruster. In *39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Huntsville, Alabama, July 2003. American Institute of Aeronautics and Astronautics.

[14] Igal Kronhaus, Alexander Kapulkin, Vladimir Balabanov, Maksim Rubanovich, Moshe Guelman, and Benveniste Natan. Discharge Characterization of the Coaxial Magnetoisolated Longitudinal Anode Hall Thruster. *Journal of Propulsion and Power*, 29(4):938–949, July 2013.

[15] Igal Kronhaus, Alexander Kapulkin, Moshe Guelman, and Benveniste Natan. Investigation of two discharge configurations in the CAMILA Hall thruster by the particle-in-cell method. *Plasma Sources Science and Technology*, 21(3):035005, June 2012.

[16] Igal Kronhaus, Alexander Kapulkin, Vladimir Balabanov, Maksim Rubanovich, Moshe Guelman, and Benveniste Natan. Investigation of physical processes in CAMILA Hall thruster using electrical probes. *Journal of Physics D: Applied Physics*, 45(17):175203, May 2012.

[17] Lou Grimaud. *Magnetic shielding topology applied to low power Hall thrusters*. PhD thesis, Université d'Orléans, 2018.

[18] Stéphane Mazouffre and Lou Grimaud. Characteristics and performances of a 100-w hall thruster for microspacecraft. *IEEE Transactions on Plasma Science*, 46(2):330–337, 2018.

[19] AI Bugrova, AD Desiatskov, HR Kaufman, VK Kharchevnikov, AI Morozov, and VV Zhurin. Design and experimental investigation of a small closed drift thruster. In *Proceedings of the International Electric Propulsion Conference, Pasadena, CA, USA*, pages 15–19, 2001.

[20] Alejandro Lopez Ortega, Ioannis G Mikellides, Ryan Conversano, Robert B Lobbia, and Vernon H Chaplin. Plasma simulations for the assessment of pole erosion in the magnetically shielded miniature hall thruster (masmi). 2019.

[21] Ryan W Conversano, Dan M Goebel, Richard R Hofer, and Nitin Arora. Performance enhancement of a long-life, low-power hall thruster for deep-space smallsats. In *2017 IEEE Aerospace Conference*, pages 1–12. IEEE, 2017.

[22] Ryan W Conversano, Robert B Lobbia, Thomas V Kerber, Kimberlyn C Tilley, Dan M Goebel, and Sean W Reilly. Performance characterization of a low-power magnetically shielded hall thruster with an internally-mounted hollow cathode. *Plasma Sources Science and Technology*, 28(10):105011, 2019.

[23] Ryan W Conversano, Dan M Goebel, Richard R Hofer, Ioannis G Mikellides, and Richard E Wirz. Performance analysis of a low-power magnetically shielded hall thruster: Experiments. *Journal of Propulsion and Power*, 33(4):975–983, 2017.

[24] Vadim Khayms and M Martinez-Sanchez. Design of a miniaturized hall thruster for microsatellites. In *32nd Joint Propulsion Conference and Exhibit*, page 3291, 1996.

[25] Hani Kamhawi, Thomas Haag, David Jacobson, and David Manzella. Performance evaluation of the nasa-300m 20 kw hall thruster. In *47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, page 5521, 2011.

[26] Hani Kamhawi, Wensheng Huang, Thomas Haag, Rohit Shastry, George Soulas, Timothy Smith, Ioannis Mikellides, and Richard Hofer. Performance and thermal characterization of the nasa-300ms 20 kw hall effect thruster. In *International Electric Propulsion Conference (IEPC 2013)*, number GRC-E-DAA-TN11609, 2013.

[27] M Yu Bernikova and VV Gopanchuk. Parametric family of the plas-type thrusters: development status and future activities. In *35th International Electric Propulsion Conference (Georgia Institute of Technology, Atlanta, Georgia, USA*, pages 8–12, 2017.

[28] M Yu Potapenko, VV Gopanchuk, and SV Olotin. Plas-40 development status: New results. In *34th International Electric Propulsion Conference*, 2015.

[29] Antonio Piragino, Vittorio Giannetti, Maryam Reza, Farbod Faraji, Eugenio Ferrato, Alena Kitaeva, Daniela Pedrini, Tommaso Andreussi, Fabrizio Paganucci, and Mariano Andrenucci. Development status of sitael's 20 kw class hall thruster. In *AIAA Propulsion and Energy 2019 Forum*, page 3812, 2019.

[30] Antonio Piragino, Eugenio Ferrato, Farbod Faraji, Maryam Reza, Vittorio Giannetti, Alena Kitaeva, Daniela Pedrini, Mariano Andrenucci, and Tommaso Andreussi. Sitael's magnetically shielded 20 kw hall thruster tests. In *Proceedings of the 36th International Electric Propulsion Conference, Vienna, Austria*, pages 15–20, 2019.

[31] B Arkhipov, N Maslennikov, V Murashko, A Veselovzorov, A Morozov, I Pokrovsky, V Gavryushin, S Khartov, V Kim, and V Kozlov. Development and investigation of characteristics of increased power spt models. paper iepc-93-222. In *23rd International Electric Propulsion Conference. Seattle (USA), 13-16 September*, page 2087, 1993.

[32] David Manzella, David Jacobson, and Robert Jankovsky. High voltage spt performance. In *37th Joint Propulsion Conference and Exhibit*, page 3774, 2001.

[33] Tommaso Misuri and Mariano Andrenucci. Het scaling methodology: Improvement and assessment. In *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, page 4806, 2008.

[34] David Jacobson and Robert Jankovsky. Test results of a 200w class hall thruster. In *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, page 3792, 1998.

[35] P Saevets, D Semenenko, R Albertoni, and G Scremin. Development of a long-life low-power hall thruster. In *The 35th International Electric Propulsion Conference*, pages 1–11, 2017.

[36] Richard Hofer and Robert Jankovsky. A hall thruster performance model incorporating the effects of a multiply-charged plasma. In *37th Joint Propulsion Conference and Exhibit*, page 3322, 2001.

[37] AI Morozov and VV Savelyev. Fundamentals of stationary plasma thruster theory. *Reviews of plasma physics*, pages 203–391, 2000.

[38] Viacheslav V Zhurin, Harold R Kaufman, and Raymond S Robinson. Physics of closed drift thrusters. *Plasma Sources Science and Technology*, 8(1):R1, 1999.

[39] E Ahedo and JM Gallardo. Scaling down hall thrusters. In *28th Int. Electric Propulsion Conf., Toulouse, France*, pages 2003–104, 2003.

[40] Vadim Khayms and M Martinez-Sanchez. Design of a miniaturized hall thruster for microsatellites. In *32nd Joint Propulsion Conference and Exhibit*, page 3291, 1996.

[41] Yu Daren, Ding Yongjie, and Zeng Zhi. Improvement on the scaling theory of the stationary plasma thruster. *Journal of Propulsion and Power*, 21(1):139–143, 2005.

[42] Tommaso Misuri, Francesco Battista, Christian Barbieri, Enrico Alessio De Marco, Mariano Andrenucci, et al. High power hall thruster design options. In *Proceedings of the 30th International Electric Propulsion Conference (Florence), IEPC*, pages 07–311, 2007.

[43] Eunkwang Lee, Younho Kim, Hodong Lee, Holak Kim, Guentae Doh, Dongho Lee, and Wonho Choe. Scaling approach for sub-kilowatt hall-effect thrusters. *Journal of Propulsion and Power*, 35(6):1073–1079, 2019.

[44] Alfredo Marianacci and Stephane Mazouffre. Supervised machine learning-based hall thruster scaling. *To appear in Journal of Electric Propulsion*, 2024.

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

[46] Stuart J Russell and Peter Norvig. *Artificial intelligence a modern approach*. Pearson, London, 4 edition, 2010.

[47] Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002. Nonlinear Methods and Data Mining.

[48] Jarmo Ilonen, Joni-Kristian Kamarainen, and Jouni Lampinen. Differential evolution training algorithm for feed-forward neural networks. *Neural Processing Letters*, 17:93–105, 2003.

[49] Vitaliy Feoktistov. *Differential evolution*. Springer, New York, 1 edition, 2006.

[50] Mohamad Faiz Ahmad, Nor Ashidi Mat Isa, Wei Hong Lim, and Koon Meng Ang. Differential evolution: A recent review based on state-of-the-art works. *Alexandria Engineering Journal*, 61(5):3831–3872, 2022.