Empirical Models for Predicting Two-Stage Light Gas Gun Muzzle Velocity

Max Murtaugh[a], Jacob A. Rogers[a], Douglas Allaire[a], Thomas E. Lacy, Jr.[a,*]

[a] *J. Mike Walker '66 Department of Mechanical Engineering, Texas A&M University, College Station, Texas, 77843.*

Abstract

Two-stage light gas guns (2SLGGs) can reliably accelerate projectiles to velocities between 1.5-8.0+ km/s and are used in hypervelocity impact, aerospace, and hypersonic research. 2SLGG operation involves a variety of physical phenomena including combustion, gas-compression, heat-transfer, and friction. Due to the wide range of operational parameters and experimental uncertainty, accurate muzzle velocity predictions can be a serious challenge. In this paper, a series of regression models for predicting muzzle velocity were fitted to and validated against 171 2SLGG launches (projectile velocities, 1.5-6.8 km/s) performed at the Texas A&M University Hypervelocity Impact Laboratory (TAMU HVIL). Most of the regression models analyzed had minimal accuracy improvement compared to basic linear regression. However, a neural network model (RMSE = 0.234 km/s) utilizing several methods to combat over-fitting, showed consistent improvement over linear regression (RMSE = 0.260 km/s) and Gaussian process regression (RMSE = 0.240 km/s). Regression model projectile velocity estimates were compared to results from the classical Piston Compression Light Gas Gun Performance (PCLGGP) and state-of-the-art LGGUN 2SLGG performance prediction codes. All of the regression models demonstrated significantly better predictive capabilities than the PCLGGP model (RMSE = 0.597 km/s), particularly at lower velocities. The regression model absolute errors from the 2SLGG experiments also compared very favorably to general absolute error estimates obtained using LGGUN. These results suggest that easy-to-implement, maintain, and scalable regression models may provide a viable alternative to complex physics-based computational models for 2SLGG launch velocity predictions, particularly as the volume of available experimental data increases. Such regression models have the potential to markedly improve predictive capabilities, identify complex coupling between experimental parameters, and reduce uncertainty.

Keywords: two-stage light gas gun (2SLGG), aeroballistic range, hypervelocity impact, regression, neural network, empirical model

1. Introduction and Motivation

Hypersonic vehicles and spacecraft are subjected to extreme operating environments that can adversely affect mission performance. For hypersonic vehicles travelling well above the speed of sound, impacts with

*Corresponding author
Email address: TELacyJr@tamu.edu (Thomas E. Lacy, Jr.)

small, slow-moving atmospheric particles (rain, ice, dust, *etc.*) can be devastating. In space, micrometeoroid orbital debris (MMOD) can impact spacecraft and planetary structures with relative velocities ranging from 2–70 km/s [1], leading to catastrophic system failure or possible loss of life. Although the impact physics between the projectile and target are well-known at lower velocities, such knowledge does not directly transfer to impacts at velocities exceeding roughly 2.5–3.0 km/s since the material response generally transitions from strength-dominated (high-velocity) to shock/pressure-dominated (hypervelocity)[1] behavior [2]. Since the end of WWII, numerous laboratories have been established worldwide to conduct hypervelocity impact (HVI) research aimed at characterizing and mitigating HVIs. Many of these facilities employ a two-stage light gas gun (2SLGG) to accelerate projectiles to hypervelocities. An extensive review of such 2SLGG facilities, their capabilities, and research areas is presented in Rogers *et al.* [3].

In general, 2SLGGs can be used to efficiently accelerate projectiles to velocities between 1.5-8.0 km/s. Accurate and reliable prediction of 2SLGG muzzle velocities as a function of launch parameters can be extremely challenging but is essential for the execution of tightly controlled launches. Without reasonably accurate predictions of muzzle velocity, multiple launches may be necessary to achieve a desired projectile velocity. Moreover, replicate experiments performed at a specific velocity (with an acceptable tolerance) can be difficult to perform. This can dramatically increase experimental costs and turnaround times, as well as limit the utility of single experiments where repeated shots are not feasible (*e.g.,* there are a limited number of targets). Thus, incorporating robust and adaptable tools for predicting launch velocities is critical for the execution of a viable test plan. The actual 2SLGG muzzle velocity depends on the complex coupling between a multitude of operational parameters, including gunpowder type, gunpowder mass, piston mass, burst disk rupture pressure, light gas initial pressure, projectile package mass, pump tube geometry, launch tube geometry, frictional forces between components, *etc.* For example, the free volume of the powder chamber can profoundly affect the powder burn efficiency, compression piston acceleration profile, and energy transfer to the projectile.

Several methods have been developed to predict 2SLGG performance and muzzle velocity. Charters *et al.* [4] at the NASA Ames Research Center developed one of the earliest and simplest models, implemented in the Piston Compression Light Gas Gun Performance (PCLGGP) software package (*i.e.*, "Charters' code"). Charters' code has been widely adopted and is relatively easy to implement since it involves the simultaneous solution of a set of linear algebraic equations. The PCLGGP model, however, makes a number of simplifying assumptions that can limit accuracy (such as negligible friction, negligible heat loss, and no gas flow in the pump tube). In contrast, the Simple Isentropic Compression model [5] makes fewer assumptions (accounting for friction and simplified gas flow in the pump tube, as well as subsonic-to-sonic gas flow in the nozzle) but is slightly more challenging to implement than the PCLGGP model since it requires the numerical solution of a coupled set of nonlinear differential equations. The Richter-Von Neuman "q-method" [6] can account

[1] For simplicity, the transition from high-velocity to hypervelocity is loosely considered to occur over the range 2.5-3.0 km/s [2].

for supersonic flow and shock waves in the pump tube, resulting in improvements in the predicted launch velocities. The recently updated LGGUN program developed by Bogdonaff *et al.* [7, 8] at NASA Ames is arguably the most sophisticated 2SLGG prediction code. LGGUN is a quasi-one-dimensional Gudonov code that is second-order accurate in time and third-order accurate in space. Validation studies using LGGUN report high accuracy, but the code implementation and interpretation of results requires considerable expertise on the part of the user.

The limitations and difficulties associated with numerical 2SLGG performance prediction tools have motivated the development and implementation of a few empirical approaches. Fraunhofer EMI developed a neural network model to predict 2SLGG muzzle velocities in an effort to improve gun performance [9, 10]. However, limited information was provided regarding the neural network model's design and accuracy. Shojaei *et al.* [11] evaluated a series of regression methods for predicting 2SLGG muzzle velocities, with a focus on random forest regression [12]. In the current study, we dramatically extend the work of Shojaei *et al.* [11] by considering a more robust set of regression models. Random forest regression was not included since an initial screening of regression models suggested that it did not perform as well as basic linear regression. Apart from Shojaei *et al.* [11] and the current study, most techniques for predicting 2SLGG performance are physics-based numerical models with, at most, empirically derived parameters. To the best of the authors' knowledge, there are no other purely empirical 2SLGG prediction studies reported in the literature.

Since the advent of 2SLGGs (*circa* 1950) [13, 14], many computational resources and advanced regression techniques have been developed which can be used to predict launcher performance. For instance, artificial neural network regression [15], support vector regression [16], and Gaussian process regression with particular kernels [17] are often used for nonlinear regression since they serve as "universal approximators" [18, 19] able to approximate any deterministic relationship between inputs and outputs. The different regression methods vary widely in their implementation, accuracy, and unique strengths.

In this study, a series of regression methods (linear, LASSO, ridge, elastic-net, neural network, Gaussian process, and support vector) are used to predict 2SLGG performance. Specification of each regression method is first presented followed by a discussion of regression results. Concluding remarks are provided and future improvements to the approach are suggested.

2. Methodology: HVI Experiments and Regression Models for Predicting 2SLGG Muzzle Velocity

All regression models were developed using performance data from the powder-driven 12.7 mm bore 2SLGG used in the Texas A&M University Hypervelocity Impact Lab (TAMU HVIL) [2]. A brief overview of the 2SLGG operation and the regression models are included in the following discussion.

2.1. 2SLGG Operation and Experiments

In essence, a 2SLGG harnesses the energy generated by a single-stage launch system to compress a light working gas, which ultimately drives a projectile. Typically, 2SLGGs are comprised of seven primary structural and consumable elements: (1) a pressure breech, (2) pump tube, (3) central breech, (4) launch tube, (5) deformable compression piston, (6) burst disk (a.k.a. petal valve), and (7) projectile or projectile package (projectile + sabot). The pressure breech, pump tube, central breech, and launch tube are coaxially arranged and connected with rigid coupling mechanisms and interfacing O-rings to ensure gas-tight seals [3]. Initially, the piston occupies the uprange end of the pump tube, and the projectile is situated at the uprange end of the launch tube, just downrange of the burst disk (Figure 1a). A predetermined quantity of a low-molecular-weight light working gas (*e.g.*, hydrogen or helium) is injected into the pump tube. As an aside, gunpowder combustion generates the piston driver gas in most guns (˜80%), but a few guns use cold-gas (He, N_2, *etc.*) stored in high-pressure reservoirs, released through a fast-acting valve [3]. While cold-gas-driven 2SLGGs are not considered in this study, their inclusion in the future would be relatively straightforward. For the HVIL gun and most other powder-driven 2SLGGs, the main (secondary) powder charge is ignited in the pressure breech by a smaller, faster-burning primary charge. High-pressure combustion gases generated in the powder/pressure breech are used to drive a deformable (*e.g.* polyethylene) piston downrange within the pump tube (Figure 1b); this compression process rapidly elevates the pressure and temperature of the working gas. When a specific burst pressure is exceeded, the burst disk ruptures, exposing the projectile to the compressed working gas, which propels it down the launch tube (Figure 1c). Eventually, the projectile exits the muzzle at velocities up to $\sim$8 km/s (Figure 1d) [3]. Most 2SLGGs are accompanied by coaxial range tankage that collectively form an aeroballistic range. Tankage assemblies usually include one or more enclosed cylindrical tanks that contain the projectile during its free flight and/or impact. Range tankage configurations and applications are largely outside the scope of this work but are summarized in Rogers *et al.* [3]. For reference, key design parameters and capabilities of the HVIL 2SLGG are given in Table 1.

After the projectile package leaves the muzzle, the sabot generally must be stripped from the projectile by some means. For smooth bore guns, such as that in the HVIL, sabots are separated from the projectile *via* aerodynamic forces within one or more of the range tanks [3, 20, 21]. The tankage internal pressure can be varied to induce different degrees of sabot separation. Of course, this separation process can also decelerate the projectile. Hence, the range tankage "backfill" pressure, in addition to primary and secondary powder types/masses, piston mass, piston release pressure, working gas type and initial pressure, burst disk rupture pressure, and projectile package mass can be readily varied to tune 2SLGG launch conditions and performance. Accurately predicting the muzzle velocity given some or all of these variables is the primary goal of this work. The fundamental operation of most unaltered 2SLGGs is very similar [3], allowing this study's findings, interpretations, and predictive methods to be easily adapted to other gun configurations.

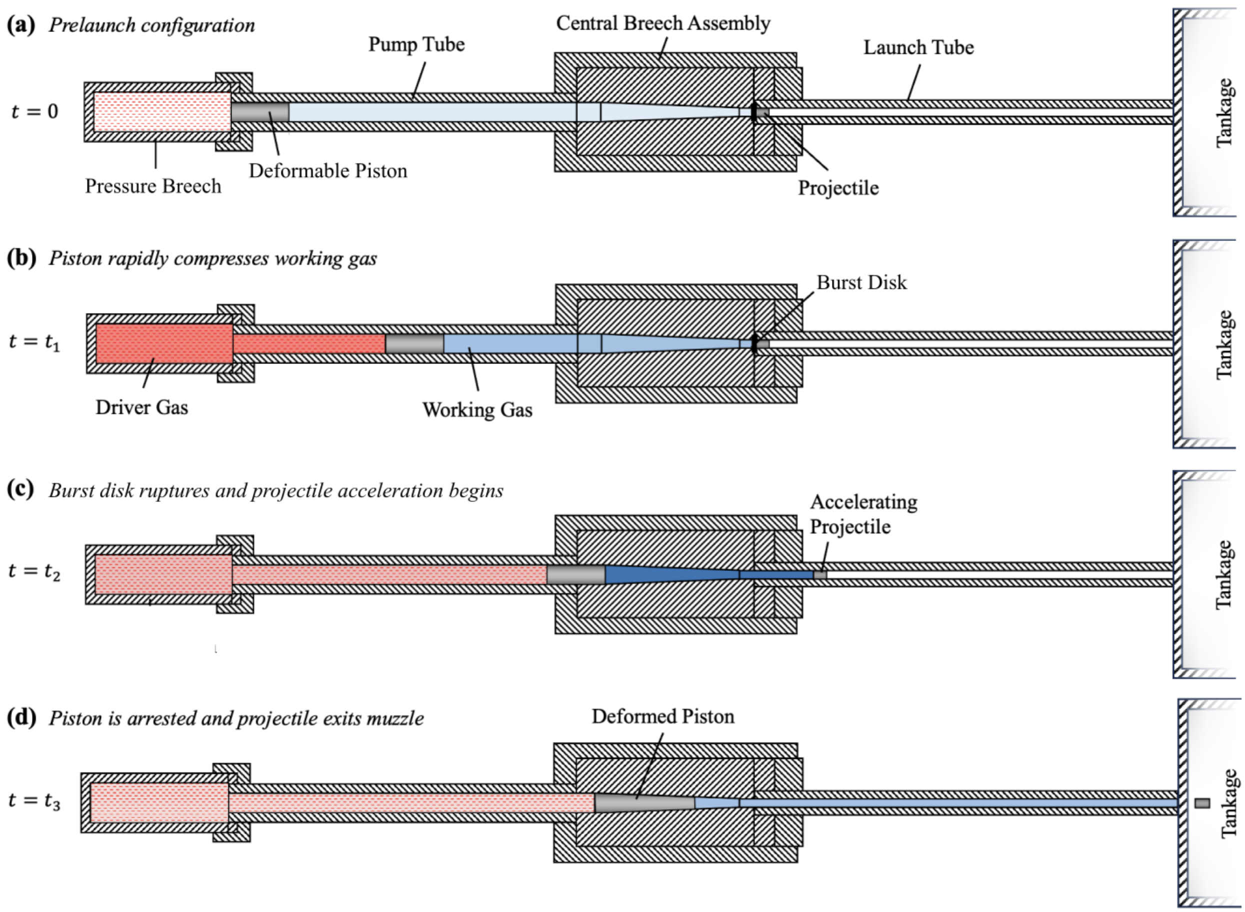

Figure 1: An illustrative guide to the operating principle of a 2SLGG, detailing (a) the moment prior to projectile launch, (b) the phase of working gas compression, (c) the instant the burst disk ruptures and acceleration of the projectile commences, and (d) immediately following the projectile's exit from the muzzle as it enters the aeroballistic range tankage. Figure reprinted with modification from Rogers *et al.* [3].

2.2. Variable/Feature Selection

In total, ten different variables (features) were used for predicting 2SLGG muzzle velocity. Most of the variables were continuous: projectile package mass, primary powder mass, secondary powder mass, pump-tube fill (working gas) pressure, and tankage (backfill) pressure. The steel burst disk rupture pressure was varied by changing the disk's score depth. This score depth was treated as quasi-discrete since only two depths were considered. The primary powder type and mass were not the only variables that influenced the pressure profile in the powder breech (and thus piston acceleration in the pump tube): changing the internal volume of the powder breech for a given powder mass resulted in more efficient powder combustion. Increasing the piston release pressure had a similar effect. The powder breech internal volume was varied by incorporating volume reducers. The number of volume reducers was treated as a discrete variable. Changes in piston release pressure were somewhat controlled by discrete variations in the piston frictional fit within the pump tube. The piston release pressure was treated as a *subjective* discrete variable characterized in terms of how "tightly" the piston fit into the uprange end of the pump tube; "low," "med," or "high" levels of tightness corresponded with parameter values of 0.0, 0.5, and 1.0, respectively. Finally, five categorical features were used to represent the five different gun powders employed in previous experiments: H4831 [22], H4831SC [22], IMR 4831 [23], 50BMG [22], and SW50BMG [24]. The 50BMG and SW50BMG powder burn data needed for PCLGGP predictions were not available at the time of this publication. Hence, to compare with PCLGGP predictions, data using these powders was omitted. However, the full-scale laboratory implementation of the regression model for 2SLGG muzzle velocity is trained on more of the data which includes all powder types.

2.3. Model Validation and Hyperparameter Tuning

A regression model is only useful so far as it can accurately predict new data. Since models can overfit data (accurately predicting given data but mispredicting new data) the only reliable metric for a regression model's performance (*i.e.,* accuracy) is how well it predicts "unseen" data that was not used to fit the model. In practice this means that some of the data must be withheld during the training process to validate model performance afterwards. A common method is holdout validation which involves partitioning the data into a training set (for fitting the model) and a testing set (for validating model performance) [25]. Determining how much data to withhold for testing presents a nuanced challenge, particularly for smaller datasets. The training set needs enough data to construct a robust model while the testing set needs enough data to give a reliable and representative estimate of model performance. [25]

During training, parameters (analogous to model coefficients) are fit to match the training data through some sort of optimization process (*e.g.*, gradient descent [26]). However, some regression models contain both parameters (optimized during the training process) and hyperparameters which affect the training process and must be decided beforehand. Examples of hyperparameters include the learning rate, number of training iterations, penalty terms, and the number of layers in a neural network. These hyperparameters can have a significant impact on a model's performance but must be determined outside of the model's typical training

Table 1: Key design and operational parameters for the TAMU HVIL 2SLGG [2]. Regression model parameters are indicated with an asterisk (*).

Parameter	Value
2SLGG total length (m)	13.70
Pump tube inner diameter (mm)	44.00
Pump tube length (m)	3.70
Launch tube inner diameter (mm)	12.70
Launch tube length (m)	3.70
Single-projectile diameter range (mm)	2.0–12.7
Achievable projectile velocity range (km/s)	1.5–8.0
Maximum rated kinetic energy (kJ)	80.00
Projectile package mass* (g)	2.0–6.1
Primary powder mass* (g)	1.0–2.0
Secondary powder mass* (g)	50–165
Pump tube fill (working gas) pressure* (MPa)	0.96–1.93
Tankage (backfill) pressure* (kPa)	0.0–66.0
Piston Mass* (g)	350-800
Burst disk score depth* (mm)	0.36, 0.51
Piston Fit Tightness*	low (0.0), medium (0.5), high (1.0)
Powder chamber volume* (cm^3)	172.6, 265.5, 424.8
Secondary powder type*	H4831, H4831SC, IMR 4831, 50BMG, and SW50BMG

process. To create better models, hyperparameters are often tuned algorithmically to achieve the best model performance on some unseen data set (tuning data). Like the testing set, the tuning set gives an estimate of how well the model predicts unseen data with specific hyperparameters. However, the tuning set and testing set must be disjoint to avoid "peeking:" including testing data in the hyperparameter tuning process. Once the testing data has been *peeked* it is no longer truly unseen data which risks overfitting and can bias the final results. A simple method for segmenting the testing, tuning, and training sets is to first partition the data into a training and testing set and then further partition the training set into a tuning set and final training set as demonstrated in Figure 2. Since hyperparameter tuning shrinks the training set, it makes training robust models on limited data more difficult. For this reason, using the final hyperparameters, the model is typically retrained on the combined training and tuning data before validating the model's performance on the testing data.

To alleviate the problems with holdout validation on small data sets, cross-validation can be used instead [25]. In k-fold cross-validation, the data is partitioned into k subsets (usually 5-10). For each subset, a model is trained on the other $k - 1$ subsets combined and tested on the original subset. If this is done for each subset, the aggregate results give a measure of model performance based on all data while still using $(k - 1)/k$ of the data to train each model. This gives a more reliable and representative measure of model performance while still using sufficient data to train robust models. A drawback of cross-validation is its computational cost, as it requires the model to be trained k times instead of just once. However, when several models are trained on slightly different data (as in each fold), the predictions can be averaged to give a result less prone to overfitting. This is referred to as k-fold averaging [27] and is one of many ensemble

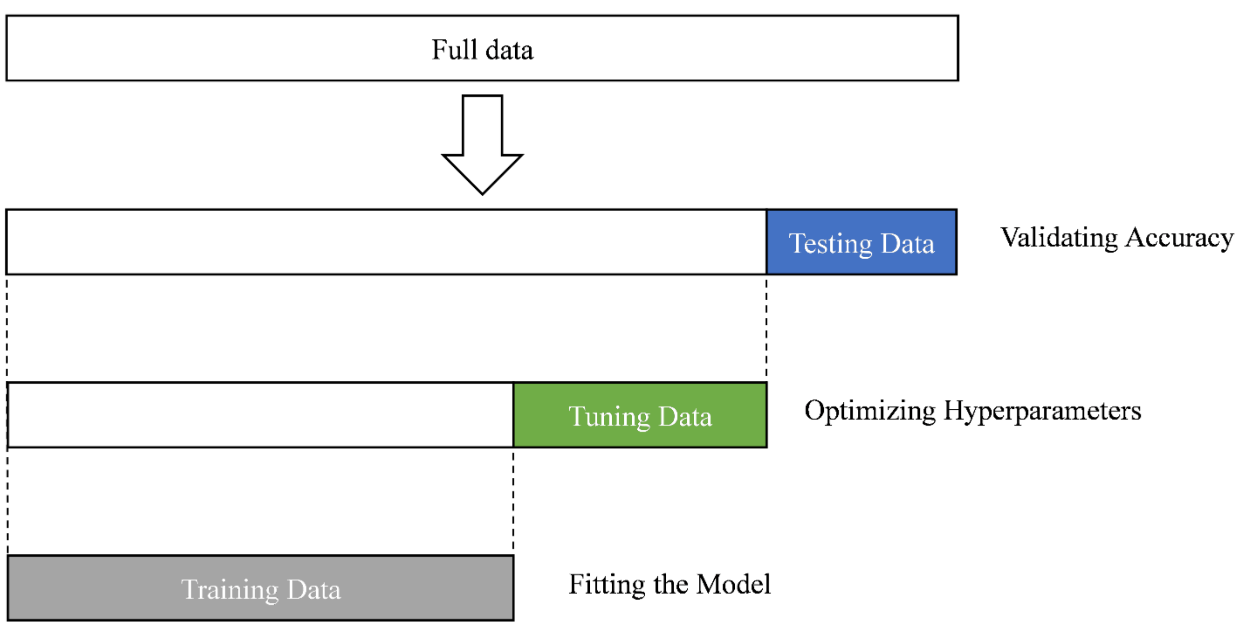

Figure 2: Illustration of data partitioning for hyperparameter tuning and model validation using basic holdout validation.

methods [28] used to combat overfitting in regression models.

Adding hyperparameter tuning to k-fold cross-validation can be accomplished in two different ways. The simplest option is to partition the training set into a single training and tuning set for each fold like in holdout validation. Another option is to split the training set into several subsets to perform *nested* cross-validation [29] which uses all the *training* data to optimize hyperparameters in the same way non-nested cross validation uses all the data to validate model performance. The recursive data partitioning involved in nested cross-validation is demonstrated in Figure 3 where an outer loop (a) is used to validate model performance and within each of these outer loops another inner loop (b) is used to optimize the hyperparameters. Nested cross-validation often leads to higher computational costs without added benefit [29], but was deemed necessary in this study to utilize k-fold averaging.

In this study, the regression model performance varied dramatically depending on which data points were used in the test set (some being easier or harder to predict than others) due to the limited number of data points (171, see Appendix A) and large number of variables (10). As such, five-fold cross-validation was used to give a more reliable estimate of model performance. For regression models needing hyperparameter tuning,

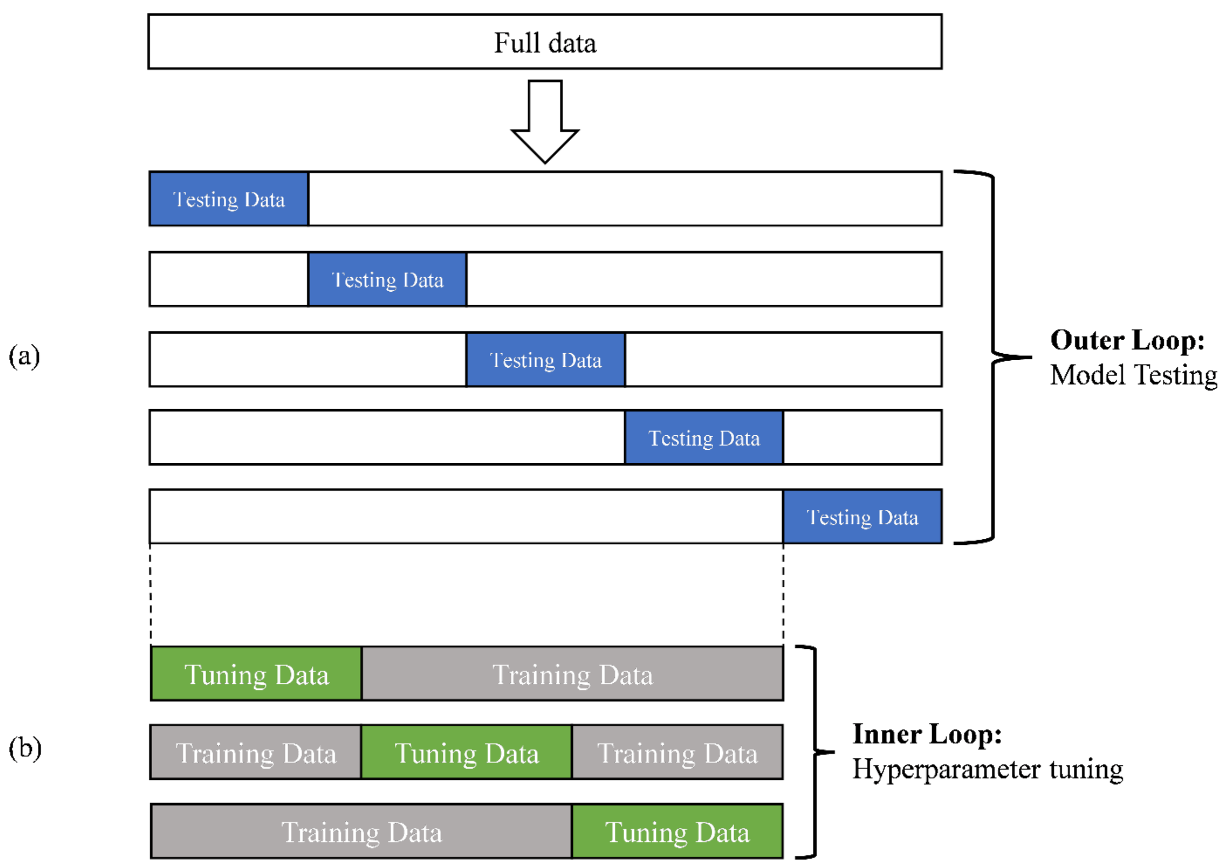

Figure 3: Pictorial demonstration of recursive data partitioning in nested cross-validation with five-fold cross-validation in the outer loop (a) and three-fold cross-validation in the inner loop (b).

hyperparameters were optimized using the Tree-Structured Parzen Estimator method [30] and nested cross-validation (five outer and inner folds) was employed to leverage k-fold averaging. This approach minimized overfitting by averaging predictions from all models in the inner loop, rather than retraining a single model with the combined training and tuning data.

2.4. Regression Methods

The most basic form of regression is linear least-squares regression, which determines the weights (coefficients) that minimize the squared residual error. LASSO [31], ridge [32], and elastic-net [33] regression function nearly identically to linear regression but add a penalty term (abbreviated as P) to the squared error that is related to the magnitude of the weights $\{w_i\}$:

$$\text{LASSO: } P = \lambda_L \sum |w_i| \,, \tag{1}$$

$$\text{ridge: } P = \lambda_R \sum w_i^2, \tag{2}$$

$$\text{elastic-net : } \; P = \lambda_L \sum |w_i| + \lambda_R \sum w_i^2, \tag{3}$$

where λ_L and λ_R are hyperparameters for scaling the penalty term. This encourages the regression model to assign zero weights to variables that only marginally improve results which reduces over-fitting. A penalty term of zero ($\lambda_L = \lambda_R = 0$) makes all three methods identical to basic linear regression.

These linear regression methods can be expanded to higher order polynomial regression (*e.g.,* quadratic) by using a higher order expansion of the independent variables. For example, a linear model with independent variables $x_1,\ x_2$ and dependent variable y can instead be expressed using a quadratic form:

$$z_1 = x_1, \quad z_2 = x_2, \quad z_3 = x_1 x_2, \quad z_4 = (x_1)^2\,, \quad z_5 = (x_2)^2\,. \tag{4}$$

Then the relationship $f\,(z_1,\ z_2,\ z_3,\ z_4,\ z_5) = y$ can be fit using linear regression (just with more variables) and is equivalent to a quadratic regression of the relationship $g\,(x_1,\ x_2) = y$. This is less useful for more sophisticated regression methods which can intrinsically derive higher order relationships (*e.g.,* neural network regression) but is essential for modeling more complex phenomena with linear regression methods.

Support vector regression [16] is more complicated than the LASSO, ridge, and elastic-net methods and seeks to find a curve that represents the data with maximal error (ϵ) between the curve and experimental values. Anything within $\pm\epsilon$ is treated the same whether the error is 0 or 0.99ϵ. This helps create a curve that fits the data within acceptable margins while avoiding over-fitting. Since ϵ is a user-specified hyperparameter, the problem can often be over-constrained at which point some "slack" beyond ϵ is allowed but minimized.

Gaussian process regression [17] is based on Bayesian statistics where prior "beliefs" (assumptions on probability) are updated based on new information. Gaussian process regression starts by assuming some prior distribution of functions and then updates this distribution based on given data. Instead of assuming a specific function for predicted model output, Gaussian process regression gives a distribution of functions with common properties (*e.g.,* differentiability, periodicity, and how close two points need to be to affect each other). In a simple case with no experimental error, giving the algorithm several data points updates the prior distribution of functions to a new (posterior) distribution where all functions pass through the given data points. From this new distribution, the expected value (or mean) at a specified point is the prediction of the model. Noisy inputs can be easily accounted for by adding a hyperparameter for the variance (or experimental error/noise) of the training data (σ_n^2) to the model. Unlike other regression methods, Gaussian process regression can give an estimate of the variance (*i.e.*, uncertainty) of a prediction using the variance of the posterior distribution at the specified point. However, Gaussian process regression can become computationally expensive on larger data sets due to the inversion of an $N \times N$ matrix, where N is the number of data points. In addition, difficulties may arise from an excessive number of features such as in image recognition. For this study, the Gaussian process regression model employs a radial basis function (RBF) kernel [34] with added noise and is allowed to update its hyperparameters and restart the process up to 200 times on each outer fold of the training data. Unlike most other models, it does not update these hyperparameters using nested cross validation nor the Tree-Structured Parzen Estimator method [30] but instead an internal process which maximizes the log-marginal likelihood [34]. The model's accuracy was determined using basic (non-nested) five-fold cross validation.

Neural network models are based on the structure of neural networks found in the brain [25]. They consist of layers of interconnected "neurons" which can process and transmit information. Each neuron receives the inputs from the previous layer and performs a weighted sum on these inputs, applies a bias, and passes it through a nonlinear activation function to produce an output for the next layer of neurons. Adjusting these weights and biases (through training the model) allows a sufficiently large neural network to approximate arbitrary functional relationships [18]. However, since neural networks use many weights and biases to fit the data, they are prone to over-fitting and lack interpretability. Many neural network architectures exist in the literature with a variety of strengths and weaknesses. After testing many different architectures and techniques, a simplified DenseNet architecture [15, 35] was used in this study due to its feature-reuse capabilities (*i.e.*, inputs are not "lost" as the depth of the neural network increases). The neural network model also employs dropout regularization [36] and k-fold averaging [27] to reduce overfitting.

3. 2SLGG Muzzle Velocity Prediction Results and Discussion

Linear least-squares (basic, LASSO, ridge, and elastic-net), support vector, Gaussian process, and artificial neural network regression were used to empirically model the 2SLGG's muzzle velocity. The independent variables (parameters) used were (1) projectile package mass, (2) primary powder mass, (3) secondary powder mass, (4) pump tube fill (working gas) pressure, (5) tankage (backfill) pressure, (6) piston mass, (7) burst disk score depth, (8) piston fit tightness, (9) number of volume reducers in the powder breech, and (10) gunpowder type. The results of each regression model and their comparison to those from two numerical models (PCLGGP and LGGUN) are presented in this section. The models are compared based on the root-mean-squared error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left|f\left(\vec{x}_i\right) - y_i\right|^2}, \tag{5}$$

where n is the number of $(\vec{x}_i,\ y_i)$ input(s)/output pairs to the regression model $f\left(\vec{x}_i\right)$. RMSE (Eq. 5) approximates the standard deviation of the errors (lower is better). All predictions and error measurements are based on data points withheld during the training process.

3.1. Regression Results for 2SLGG Muzzle Velocity Predictions

The predicted projectile velocity obtained using the most basic model (linear regression) was compared to that from PCLGGP (Charters' code) over the given 2SLGG velocity range 1.5-6.8 km/s. The linear model (RMSE = 0.260 km/s) matched the measured muzzle velocity values (diagonal dashed line) significantly better than the PCLGGP model predictions (RMSE = 0.597 km/s) especially at lower muzzle velocities (Figure 4a). The plot clearly shows that PCLGGP consistently over-predicted the actual values at lower launch velocities (*i.e.*, <5.0 km/s) but yielded more accurate results for velocities >5 km/s. Since Charter's code does not account for the effect of piston and projectile friction, the PCLGGP model has been reported to consistently over-predict muzzle velocities by 10–20% [37]. Since these effects become less significant for more energetic shots, the PCLGGP model error will tend to decrease with increasing muzzle velocity. The difference between PCLGGP and linear regression is clearer in Figure 4b, which shows the velocity normalized absolute error ($\bar{\epsilon} = \left|v_{measured} - v_{predicted}\right| / v_{measured}$) compared to the measured velocity. PCLGGP's relative error exceeded 70% at lower velocities ($\sim$ 2 km/s) but significantly improved with increasing projectile velocity. The linear model predicted a few outliers at lower velocities but provided more stable predictions throughout the entire velocity range. The dashed lines in Figure 5b demonstrate that the average relative error for the linear regression predictions (5.4%) was dramatically lower than those for the PCLGGP model (15.8%).

The results of LASSO (RMSE = 0.259 km/s), ridge (RMSE = 0.260 km/s), and elastic-net (RMSE = 0.260 km/s) regression were almost identical to the linear regression results. In all cases, the penalty terms were negligibly small. Recall that with a null penalty term the LASSO, ridge, and elastic-net regression predictions will be identical to that for basic linear regression with no reduction in overfitting. Hence, a penalty term near zero implies that, for a comparable linear model, there is little to no benefit in removing any independent variables. The support vector regression (RMSE = 0.263 km/s) predictions approach, but do not improve upon, the linear model results even with significant hyperparameter tuning. Hence, the support vector regression model was the only model considered that performed *worse* than basic linear regression. A summary of all model results is presented in order of descending RMSE values in Table 2.

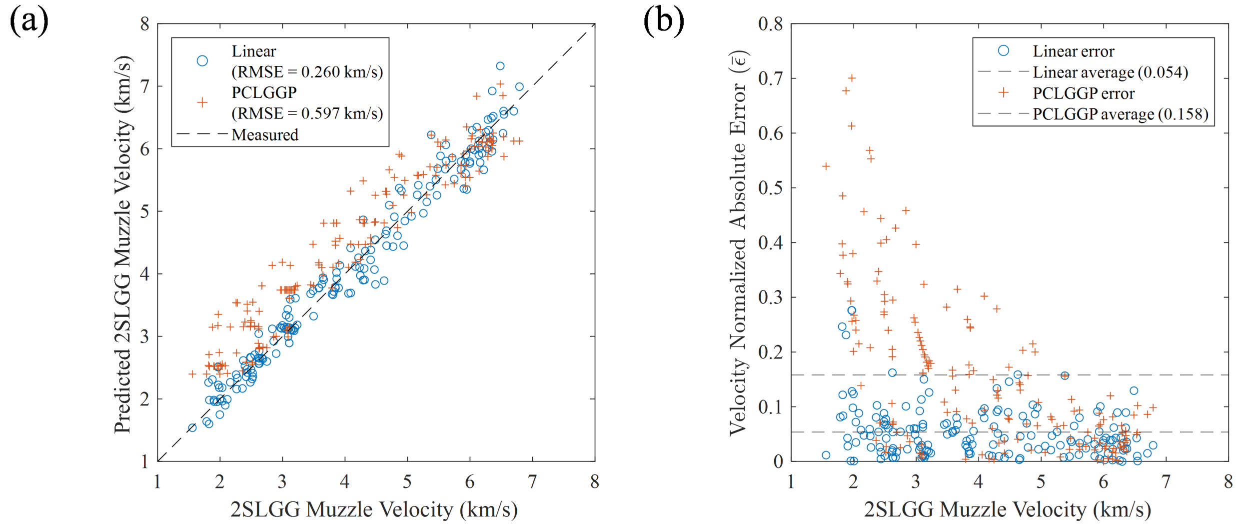

Figure 4: PCLGGP (Charters' code) results compared to linear regression results: (a) predicted muzzle velocity versus measured muzzle velocity and (b) normalized absolute error versus measured muzzle velocity.

A reasonable explanation exists as to why the LASSO, ridge, elastic-net, and support vector methods did not perform better than simple linear regression. While 171 2SLGG launches are significant from a cost and time standpoint, there are still relatively few data points for regression, especially when using 10 independent variables. Furthermore, significant random error between nearly identical shots makes it difficult to differentiate trends in the data from random noise. For example, when linear regression was performed with the secondary powder mass as the only independent variable, the model achieved an RMSE of 0.439 km/s and an R-squared value of 0.92 (*i.e.*, 92% of all the variance in the data can be explained solely by the secondary powder mass). The coupled effects of other variables are likely numerous, complex, and largely overshadowed by random error and the dominant effect of the secondary powder mass. This makes it more difficult to model weaker trends without overfitting the limited data.

Gaussian process regression showed a noticeable improvement (RMSE = 0.240 km/s) over linear regression (RMSE = 0.260 km/s). Moreover, it also gives an estimate of the variance in the prediction at any evaluated point— this is useful in determining the model's confidence in each prediction. The linear, LASSO,

Table 2: The root-mean-square error (RMSE) over holdout data for each model over the 2SLGG operational velocity range 1.5-6.8 km/s (The Quad. RMSE column represents the results when quadratic features were passed through the model). LGGUN results (*) correspond to a different set of experiments performed using multiple 2SLGGs [38, 39].

No.	Model	RMSE (km/s)	Quad. RMSE (km/s)
1	PCLGGP (Charters' code)	0.597	...
2	LGGUN [38, 39]	0.302*	...
3	Support Vector	0.263	...
4	Linear	0.260	1.445
5	Elastic-Net	0.260	0.255
6	Ridge	0.260	0.249
7	LASSO	0.259	0.248
8	Gaussian Process	0.240	...
9	Neural Network	0.234	...

ridge, and elastic-net methods were also expanded to quadratic regression by using a quadratic expansion of the variables. The basic quadratic model (linear regression model with quadratic variables) performed significantly worse (RMSE = 1.445 km/s) than basic linear regression. In contrast, the quadratic LASSO, ridge, and elastic-net models provided a *slight* improvement in prediction error (RMSE = 0.248-0.255 km/s) relative to basic linear regression (RMSE = 0.260 km/s; *cf.*, Table 2). The difference between the basic quadratic model results and the results of the quadratic LASSO, ridge, and elastic-net models is explained by the lack of weight (coefficient) regularization in basic polynomial regression, where the coefficients of all polynomial terms must be determined, leading to substantial overfitting in higher order regression as the number of terms increases. In contrast, the LASSO, ridge, and elastic-net regression techniques can eliminate excess (non-contributing) variables to avoid overfitting. Moreover, since the RMSE is calculated based on unseen data (instead of the training data), the overfitted basic quadratic model performed much worse than the basic linear model. Note that the RMSE values of the quadratic models were all *greater* than that for the Gaussian process model (RMSE = 0.240 km/s).

In contrast, the neural network model (RMSE = 0.234 km/s) was the most accurate of all regression models considered and exhibited reliable (albeit nominal) improvement over linear regression (RMSE = 0.260 km/s) and Gaussian process regression (RMSE = 0.240 km/s). Figure 5a compares the linear and neural network predictions to measured velocities, and Figure 5b compares the relative error of the linear and neural network predictions at different muzzle velocities. In both Figure 5a and Figure 5b, the modest improvements in predicted launch velocities obtained using the neural network model are readily apparent but are much less pronounced than the differences between PCLGGP and linear regression predictions shown in Figure 4. Moreover, error histograms corresponding to the neural network and PCLGGP results (Figure 6) clearly show that the neural network model's error distribution has a lower standard deviation and less bias towards over-prediction. As an aside, as the number of experiments is further increased, the accuracy of the neural network and Gaussian process models are expected to increase dramatically relative to the other regression models considered here since they are well suited for approximating the nonlinear relationships between independent variables that a larger data set would reveal. As the number of experiments increases, however, the neural network model will scale better with the size of the data set.

Despite the neural network model's relative increase in accuracy and potential for improved predictions on larger data sets, it has some unavoidable limitations relative to the other empirical models. For example, in traditional polynomial regression, the magnitude and sign of each coefficient in the fitted model defines the influence of each independent variable (projectile package mass, working gas pressure, *etc.*) on the predicted response (*i.e.,* projectile velocity), as well as how coupled interactions between independent variables affect the model outputs. The neural network model, however, lacks such intuition. Relationships between inputs and outputs can be estimated by evaluating the neural network model at different points, but not directly interpreted from the trained weights and biases. Similarly, while the Gaussian process model was slightly less accurate than the neural network model, it provides a prediction of uncertainty that the neural network

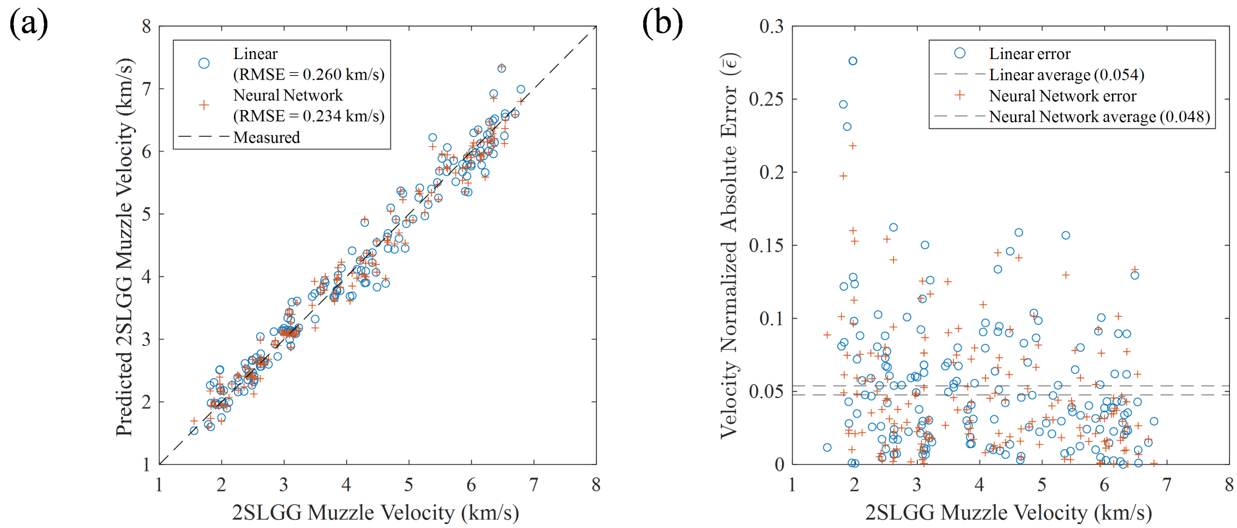

Figure 5: Neural network model results compared to linear results: (a) predicted muzzle velocity versus measured muzzle velocity and (b) velocity normalized absolute error versus measured muzzle velocity.

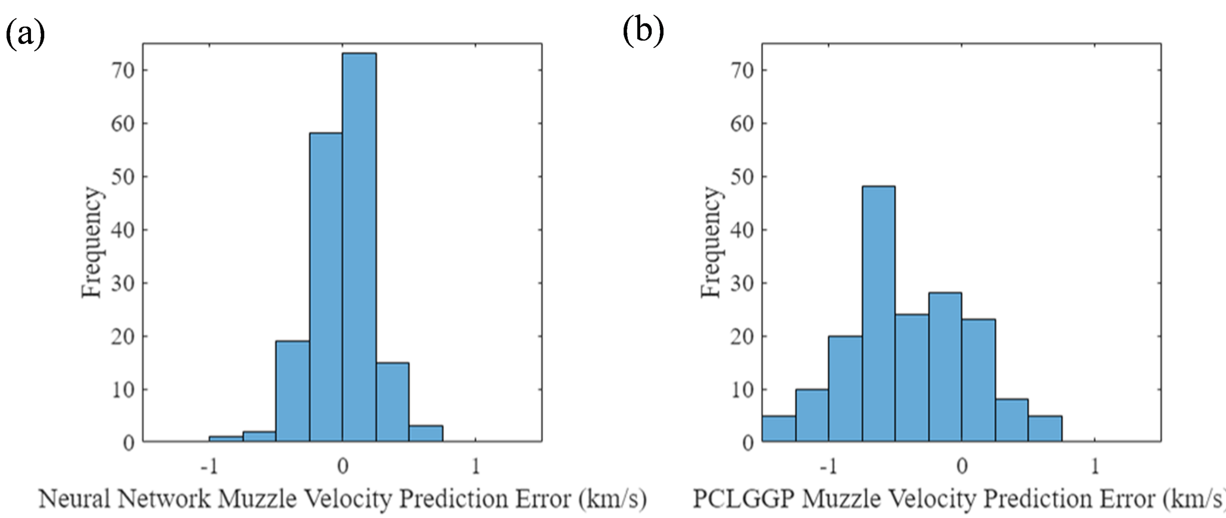

Figure 6: Histogram of errors for (a) the neural network model and (b) the PCLGGP predictions. Over-predictions by the model are represented as negative to match $y = \widehat{y} + \varepsilon$.

model cannot. The slight loss of accuracy associated with use of the Gaussian process model may be offset by the ability to estimate prediction uncertainty. Overall, most of the regression methods considered in this study had an RMSE value *less than one-half* that for the PCLGGP model. This demonstrates that even simple empirical models are well-suited for predicting 2SLGG muzzle velocity if sufficient experimental data exists.

3.2. Projectile Velocity Absolute Error Percentiles and Comparison to LGGUN

Using data from the 171 TAMU HVIL 2SLGG experiments with muzzle velocities in the range 1.5-6.8 km/s, projectile velocity error estimates obtained using the PCLGGP model and 11 regression models were expressed in terms of absolute error percentiles. For example, the 50$^{\text{th}}$ percentile (2$^{\text{nd}}$ quartile, Q2) would define the *median absolute error* associated with each model. Similarly, the absolute error associated with the 25$^{\text{th}}$ percentile (1$^{\text{st}}$ quartile, Q1) means that 25% of the predicted values would have absolute errors less than or equal to the specified value. Hence, absolute error percentiles can be used to assess how well a given model approximates the actual velocity over the entire range of experimental observations and estimate the likelihood of different magnitudes of error in future predictions. The absolute error estimates generated using the 171 TAMU 2SLGG experiments in this study were compared to published values predicted using LGGUN [8] for 52 experiments performed using five *different* 2SLGGs over a velocity range of 3-11 km/s [38, 39]. As mentioned previously, LGGUN is a leading edge 2SLGG performance prediction code.

Excluding the basic quadratic model (*i.e.*, linear regression model with quadratic variables), all of the regression models' root mean square error (RMSE) values (0.234-0.263 km/s) were *lower* than that for the LGGUN model (0.302 km/s; *cf.*, Table 2). Similarly, Table 3 includes a summary of predicted velocity absolute error estimates corresponding to the 25$^{\text{th}}$, 50$^{\text{th}}$, 75$^{\text{th}}$, and 100$^{\text{th}}$ percentiles (*i.e.*, quartiles Q1, Q2, Q3, Q4, respectively) for the PCLGGP and 11 regression models, as well as generic LGGUN values from data provided by D. W. Bogdanoff and shown in the literature [38, 39]. Figure 7 contains a graphical representation of these same data using "box and whisker" plots [40]. For a given model, the "box" provides the Q1, Q2 (median), and Q3 velocity absolute error estimates, as suggested in the figure. The "whiskers" associated with a given model defines the minimum (Q0) and maximum (Q4) values including outliers. Not surprisingly, the absolute errors associated with LGGUN predictions were over 40% lower than those for PCLGGP for all quartiles. Similarly, all of the regression models significantly outperformed the PCLGGP model. For example, the median absolute error (50$^{\text{th}}$ percentile; Q2) for each of the regression models was at least 62% lower than the PCLGGP value. With the exception of the 100$^{\text{th}}$ percentile (Q4) absolute error for the basic quadratic linear model, all of the regression models outperformed Charters' code over the entire range of experiments. The regression model Q1-Q3 absolute errors from the TAMU 2SLGG experiments also compared favorably to general absolute error estimates obtained using LGGUN. Excluding the basic quadratic linear model, the 25$^{\text{th}}$ (Q1), 50$^{\text{th}}$ (Q2, median), and 75$^{\text{th}}$ percentile (Q3) absolute errors for the regression models were at least 10%, 25%, and 30% *lower* than the corresponding LGGUN values, respectively. For comparison purposes, the neural network model's predicted 25$^{\text{th}}$, 50$^{\text{th}}$, and 75$^{\text{th}}$ percentile absolute errors were 18%, 39%, and 38% lower than the generic LGGUN values, respectively. All of the regression model absolute errors, however, *exceeded* the LGGUN 100$^{\text{th}}$ percentile (Q4) value. This latter difference may be attributable to a number of factors including: 1) the increased likelihood of encountering outliers within the larger TAMU dataset (171 experiments) compared to the dataset used to validate LGGUN (52 experiments); 2) the reliance of regression models on data points in the training set with similar or "nearby" inputs (*i.e.*, experiments conducted at the extreme range of independent variables will be more difficult to predict), and 3) better fidelity over the entire range of independent variables associated with the physics-based LGGUN numerical model. As an aside, if LGGUN was used to predict muzzle velocities

Table 3: Absolute error percentiles (in km/s) for all considered models. LGGUN results (*) correspond to a different set of experiments performed using multiple 2SLGGs [38, 39].

No.	Model	Absolute error Percentile (Quartile) 25$^{\text{th}}$ (Q1) (km/s)	50$^{\text{th}}$ (Q2) (km/s)	75$^{\text{th}}$ (Q3) (km/s)	100$^{\text{th}}$ (Q4) (km/s)
Numerical Models					
1	PCLGGP	0.185	0.501	0.677	1.381
2	LGGUN [38, 39]	0.083*	0.226*	0.395*	0.655*
Regression Models					
3	Linear	0.065	0.168	0.250	0.844
4	Quad. Linear	0.083	0.192	0.377	11.686
5	Ridge	0.065	0.168	0.247	0.839
6	Quad. Ridge	0.058	0.139	0.269	1.162
7	LASSO	0.068	0.155	0.261	0.826
8	Quad. LASSO	0.065	0.142	0.246	1.168
9	Elastic-Net	0.065	0.163	0.256	0.833
10	Quad. Elastic-Net	0.073	0.151	0.257	1.062
11	Support Vector	0.074	0.156	0.264	0.903
12	Gaussian Process	0.069	0.145	0.255	0.992
13	Neural Network	0.068	0.138	0.244	0.865

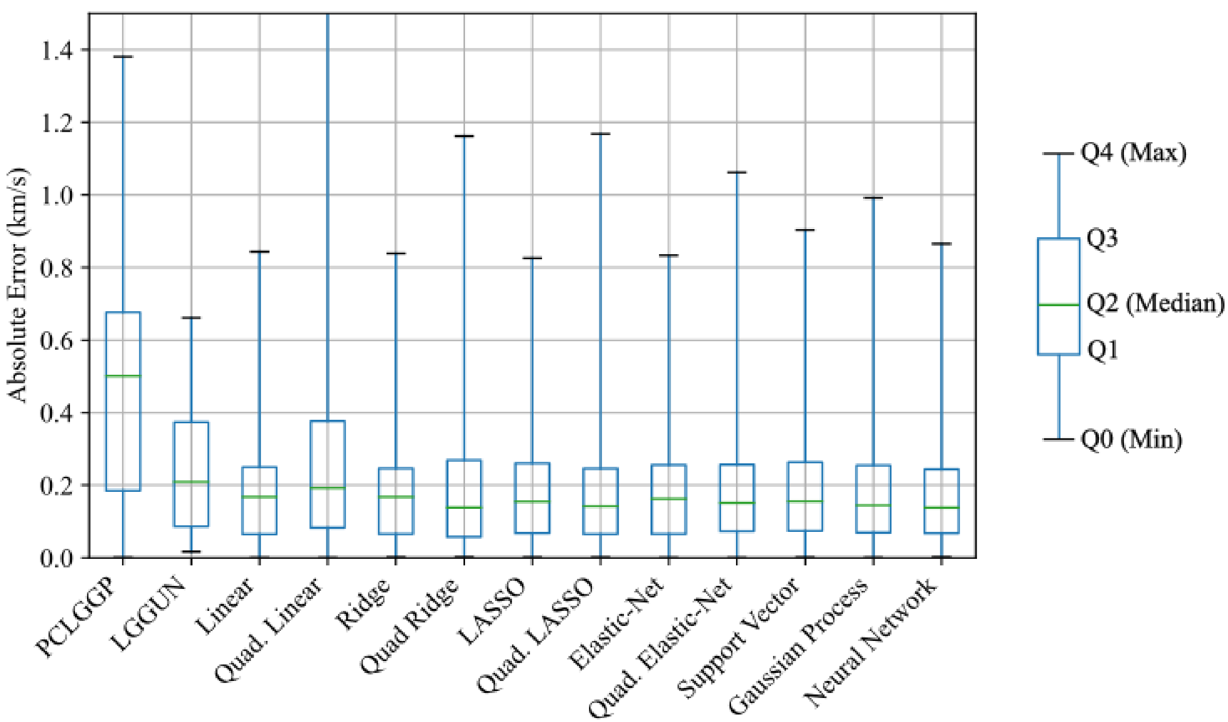

Figure 7: Box and whisker plots of the prediction absolute error for PCLGGP, LGGUN, and all considered regression models. Note that the Q4 whisker plot for the Quad. Linear model was cutoff since the Q4 value was much larger than that for any other model. LGGUN results correspond to a different set of experiments performed using multiple 2SLGGs [38, 39].

for the 171 experiments using TAMU 2SLGG launch parameters, the predicted absolute error values would likely improve upon the LGGUN values reported here. Nonetheless, these results suggest that easy-to-implement, maintain, and scalable regression models may provide an attractive alternative to more complex computational models such as PCLGGP and LGGUN for 2SLGG launch velocity predictions. While LGGUN can provide crucial information about shock formation, bore erosion, pump tube pressure-time histories, and other key aspects of specific 2SLGG performance, model specification and interpretation of results requires considerable expertise. Clearly, regression models that are amenable to scaling across different 2SLGG platforms can augment physics-based numerical models, particularly as the volume of available experimental data increases. The regression models, such as Gaussian process and neural network, have the potential to markedly improve predictive capabilities, identify complex coupling between experimental parameters, and reduce uncertainty. In the future, it may be possible to significantly reduce the maximum (Q4) absolute errors using *physics-informed* neural network models or Gaussian process regression models [41].

4. Conclusions and Future Work

In this study, linear, ridge, LASSO, elastic-net, support vector, Gaussian process, and neural network regression models were developed to predict two-stage light gas gun (2SLGG) projectile velocities as a function of 10 independent variables (launch parameters): (1) projectile package mass, (2) primary powder mass, (3) secondary powder mass, (4) pump tube fill (working gas) pressure, (5) tankage (backfill) pressure, (6) piston mass, (7) burst disk score depth, (8) piston fit tightness, (9) number of volume reducers in the powder breech, and (10) gunpowder type. The models were fitted to and validated against performance data from 171 experiments (projectile velocities, 1.5-6.8 km/s) conducted using a powder-driven 12.7 mm bore 2SLGG at the Texas A&M University Hypervelocity Impact Lab (TAMU HVIL). Regression model muzzle velocity estimates for all 171 shots were compared to numerical predictions obtained using the physics-based Piston Compression Light Gas Gun Performance (PCLGGP) software package (*i.e.*, “Charters’ code”). In addition, the prediction absolute error distributions from the regression and PCLGGP models were compared to independent values determined using the cutting-edge, physics-based LGGUN code for a different set of experiments involving multiple 2SLGGs.

Except for the basic quadratic model (*i.e.*, linear regression model with quadratic variables), all of the regression models significantly outperformed Charters’ code over the entire range of experiments. Their root mean square error (RMSE) values (0.234-0.263 km/s) were significantly lower than that for the PCLGGP model (0.597 km/s), and their predicted absolute error distributions were clearly superior to that for Charter’s code. The median absolute error (50^{th} percentile) for each of the regression models was at least 62% lower than the PCLGGP value. In general, there was not a significant difference between simple linear regression (RMSE = 0.260 km/s) and the other regression models. Gaussian process regression (RMSE = 0.240 km/s), however, includes an estimate of prediction confidence. The neural network model (RMSE = 0.234 km/s) was the most accurate regression technique considered and is particularly well-suited to scale with large data sets.

Not surprisingly, the absolute errors associated with independent LGGUN predictions were at least 40% lower than those for PCLGGP for all percentiles. Interestingly, the regression model results from the TAMU 2SLGG experiments compared favorably to LGGUN results. Excluding the basic quadratic model, the regression models' root mean square error (RMSE) values (0.234-0.263 km/s) were lower than that for the LGGUN model (0.302 km/s). Similarly, the corresponding 25^{th}-75^{th} percentile absolute errors for the regression models were significantly lower than corresponding LGGUN values. LGGUN, however, predicted a lower maximum (100^{th} percentile) absolute error than any regression model. In general, regression models may not perform as well as physics-based computational models in predicting individual experimental results involving extreme values of independent variables. In the future, this limitation can be remedied, in part, using physics-informed neural network or Gaussian process models [41] that incorporate both prior data and general physics-based knowledge over the domain.

This study demonstrates that straightforward regression models may provide an attractive alternative to more complex deterministic models for 2SLGG launch velocity predictions. Regression models that are amenable to scaling across different 2SLGG platforms can augment physics-based numerical models (particularly as the volume of available experimental data increases) and have the potential to markedly improve predictive capabilities, identify complex coupling between experimental parameters, and reduce uncertainty.

References

[1] M. I. Allende, J. E. Miller, B. A. Davis, E. L. Christiansen, M. D. Lepech, D. J. Loftus, Prediction of micrometeoroid damage to lunar construction materials using numerical modeling of hypervelocity impact events, International Journal of Impact Engineering 138 (2020) 103499.

[2] J. A. Rogers, N. Bass, P. T. Mead, A. Mote, G. D. Lukasik, M. Intardonato, K. Harrison, J. D. Leaverton, K. R. Kota, J. W. Wilkerson, J. N. Reddy, W. D. Kulatilaka, T. E. Lacy, The texas a&m university hypervelocity impact laboratory: A modern aeroballistic range facility, Review of Scientific Instruments 93 (2022) 085106.

[3] J. Rogers, N. Bass, M. Wiest, Z. Wantz, J. Wilkerson, T. Lacy, The pursuit of hypervelocities: A review of two-stage light gas gun aeroballistic ranges, International Journal of Impact Engineering 185 (2024) 104861.

[4] A. C. Charters, B. P. Denardo, V. J. Rossow, Development of a piston-compressor type light-gas gun for the launching of free-flight models at high velocity, Technical Report, 1957.

[5] R. J. Rynearson, J. L. Rand, Optimization of a two stage light gas gun, Technical Report, 1972.

[6] R. Piacesi, D. F. Gates, A. E. Seigel, Computer analysis of two-stage hypervelocity model launchers, Technical Report, 1963.

[7] D. W. Bogdanoff, R. J. Miller, New higher-order Godunov code for modelling performance of two-stage light gas guns, Technical Report, 1995.

[8] D. W. Bogdanoff, LGGUN User's Manual, 2020.

[9] B. Lexow, M. Wickert, K. Thoma, F. Schäfer, M. H. Poelchau, T. Kenkmann, The extra-large light-gas gun of the fraunhofer emi: Applications for impact cratering research, Meteoritics & Planetary Science 48 (2013) 3–7.

[10] B. Lexow, A. Bueckle, M. Wickert, S. Hiermaier, The xllgg–a hypervelovity launcher for impact cratering research., Bridging the Gap III: Impact Cratering In Nature, Experiments, and Modeling 1861 (2015) 1046.

[11] P. Shojaei, M. Trabia, B. O'Toole, R. Jennings, Predicting the projectile velocity of a two-stage gas gun using machine learning, volume 86168, 2022, p. V003T05A014.

[12] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.

[13] W. D. Crozier, W. Hume, High-velocity, light-gas gun, Journal of Applied Physics 28 (1957) 892–894.

[14] H. F. Swift, Light-gas gun technology: a historical perspective, Springer-Verlag Berlin Heidelberg, Germany, 2005, p. 1–36.

[15] D. Chen, F. Hu, G. Nian, T. Yang, Deep residual learning for nonlinear regression, Entropy 22 (2020) 193.

[16] A. J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and computing 14 (2004) 199–222.

[17] C. E. Rasmussen, Evaluation of Gaussian processes and other methods for non-linear regression, Technical Report, 1997.

[18] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural networks 2 (1989) 359–366.

[19] C. A. Micchelli, Y. Xu, H. Zhang, Universal kernels., Journal of Machine Learning Research 7 (2006).

[20] T. N. Canning, C. S. James, A. Seiff, Ballistic range technology(conference on ballistic range techniques and equipment) (1970).

[21] H. F. Swift, D. E. Strange, Sabot discard technology, Physics Applications Inc., Internal Report (1987).

[22] Extruded Rifle Powders Safety Data Sheet, Technical Report, 2019.

[23] Single Base Powders Safety Data Sheet, Technical Report, 2022.

[24] Shooters World Reloading Data, Technical Report, 2019.

[25] G. James, D. Witten, T. Hastie, R. Tibshirani, J. Taylor, An introduction to statistical learning: With applications in python, Springer, 2023.

[26] S. Ruder, An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747 (2016).

[27] Y. Jung, J. Hu, Ak-fold averaging cross-validation procedure, Journal of nonparametric statistics 27 (2015) 167–179.

[28] O. Sagi, L. Rokach, Ensemble learning: A survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (2018) e1249.

[29] J. Wainer, G. Cawley, Nested cross-validation when selecting classifiers is overzealous for most practical applications, Expert Systems with Applications 182 (2021) 115222.

[30] J. Bergstra, D. Yamins, D. Cox, Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures, 2013, p. 115–123.

[31] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society: Series B (Methodological) 58 (1996) 267–288.

[32] A. E. Hoerl, R. W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, Technometrics 12 (1970) 55–67.

[33] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of the royal statistical society: series B (statistical methodology) 67 (2005) 301–320.

[34] 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, E. Duchesnay, Scikit-learn: Machine learning in python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[35] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, 2017, p. 4700–4708.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (2014) 1929–1958.

[37] A. C. Charters, D. K. Sangster, Fortran computer program for interior ballistic analysis of light-gas guns, Informal Manual available from Dr. Charters (1973).

[38] D. W. Bogdanoff, Further Validation of a CFD Code for Calculating the Performance of Two-Stage Light Gas Guns, Technical Report, 2017.

[39] D. W. Bogdanoff, Private communication regarding individual data points plotted in figure 17 in ref. [38], 2023.

[40] S. H. C. DuToit, A. G. W. Steyn, R. H. Stumpf, Graphical exploratory data analysis, Springer Science & Business Media, 2012.

[41] G. Pang, G. E. Karniadakis, Physics-informed learning machines for partial differential equations: Gaussian processes versus neural networks, Emerging Frontiers in Nonlinear Science (2020) 323–343.

Appendix A: Experimental data

Table A.1: Experimental 2SLGG loading parameters used in regression model training and validation, as well as *measured* muzzle velocity for reference.

Projectile Package Mass (g)	Primary Powder Mass (g)	Secondary Powder Mass (g)	Burst Disc Score Depth (in)	Piston Mass (g)	Target Tank Pressure (Torr)	Pump Tube Pressure (psi)	Volume Reducers	Piston Tightness	H4831SC	IMR 4831	Projectile Velocity (m/s)
6.083	1.420	100.2	0.020	364.4	200	140	1	0.50	0	0	4165
3.381	1.401	100.5	0.020	362.6	200	140	1	0.50	0	0	4665
3.387	1.511	100.0	0.020	365.3	200	140	1	0.50	0	0	4938
3.388	1.745	101.0	0.020	367.1	200	279	1	0.50	0	0	3847
3.372	1.751	100.1	0.020	363.8	200	279	1	0.50	0	0	4091
3.389	1.751	130.0	0.020	363.6	200	140	1	0.50	0	0	5615
3.382	1.754	120.1	0.020	363.7	200	141	1	0.50	0	0	5359
3.389	1.749	110.1	0.020	365.4	200	161	1	0.50	0	0	4956
2.315	1.743	100.1	0.020	363.9	200	280	1	0.50	0	0	4486
3.379	1.753	110.7	0.020	437.7	200	160	1	0.50	0	0	5253
3.385	1.751	150.1	0.020	363.8	200	200	1	0.50	0	0	6035
3.379	1.753	150.1	0.020	366.6	250	220	1	0.50	0	0	5612
3.379	1.753	150.5	0.020	365.7	250	200	1	0.50	0	0	5379
6.059	1.751	100.4	0.020	761.7	250	200	1	0.50	0	0	4215
3.387	1.753	100.0	0.020	366.2	200	279	1	0.50	0	0	3488
6.029	1.755	100.0	0.020	767.2	200	200	1	0.50	0	0	4324
6.055	1.751	100.8	0.020	763.7	200	200	1	0.50	0	0	4413
2.163	1.757	120.0	0.020	363.1	200	174	1	0.50	0	0	5946
2.161	1.747	60.2	0.020	363.5	200	249	1	0.50	0	0	2277
3.391	1.750	100.0	0.020	363.5	200	139	1	0.50	0	0	4483
3.397	1.747	110.0	0.020	363.2	197	160	1	0.50	0	0	4290
3.400	1.749	120.1	0.020	363.6	252	140	1	0.50	0	0	5470
3.399	1.747	120.2	0.020	363.6	250	200	1	0.50	0	0	4789
2.141	1.750	150.0	0.020	363.1	200	221	1	0.50	0	0	6527
2.158	1.764	150.0	0.020	362.8	200	220	1	0.50	0	0	6107

Table A.1: Experimental 2SLGG loading parameters used in regression model training and validation, as well as *measured* muzzle velocity for reference.

Projectile Package Mass (g)	Primary Powder Mass (g)	Secondary Powder Mass (g)	Burst Disc Score Depth (in)	Piston Mass (g)	Target Tank Pressure (Torr)	Pump Tube Pressure (psi)	Volume Reducers	Piston Tightness	H4831SC	IMR 4831	Projectile Velocity (m/s)
3.379	1.746	100.0	0.020	363.8	200	271	1	0.50	0	0	3844
3.394	1.757	60.0	0.020	363.5	200	249	1	0.50	0	0	2485
6.036	1.745	60.1	0.020	363.5	200	250	1	0.50	0	0	1900
6.039	1.760	60.1	0.020	363.4	200	250	1	0.50	0	0	2035
2.081	1.751	60.0	0.020	363.5	202	250	1	0.50	0	0	2524
3.400	1.749	150.0	0.020	363.5	201	221	1	0.50	0	0	6142
3.386	1.749	150.1	0.020	363.9	76	221	1	0.50	0	0	6372
6.033	1.748	60.4	0.020	364.1	199	249	1	0.50	0	0	2003
3.399	1.751	150.0	0.020	363.2	75	220	1	0.50	0	0	6342
6.035	1.757	60.2	0.020	364.0	199	249	1	0.50	0	0	2083
3.399	1.751	60.0	0.020	363.5	199	249	1	0.50	0	0	2163
6.055	1.750	60.0	0.020	363.5	199	250	1	0.50	0	0	1829
6.040	1.750	60.2	0.020	363.9	203	250	1	0.50	0	0	1953
6.042	1.750	60.0	0.020	363.9	199	249	1	0.50	0	0	1905
3.401	1.751	150.0	0.020	363.6	76	200	1	0.50	0	0	6537
2.099	1.750	60.0	0.020	363.7	199	251	1	0.50	0	0	2256
3.387	1.750	150.0	0.020	363.9	75	250	1	0.50	0	0	6295
3.382	1.750	145.0	0.020	363.8	75	220	1	0.50	0	0	6253
3.399	1.749	125.0	0.020	363.0	90	220	1	0.50	0	1	5855
2.135	1.748	165.0	0.020	364.0	100	250	1	0.50	0	1	6484
2.145	1.758	125.0	0.020	364.2	85	220	1	0.50	0	1	6162
6.048	1.750	60.0	0.020	363.2	200	250	1	1.00	0	1	2514
3.404	1.750	83.0	0.020	364.1	100	250	1	0.50	0	1	4054
2.086	1.751	50.0	0.020	362.8	200	250	1	0.50	0	1	1877
2.080	1.750	55.0	0.020	362.4	200	250	1	0.50	0	1	1971
6.056	1.751	60.0	0.020	363.8	200	250	1	0.50	0	1	2034
3.406	1.748	65.0	0.020	363.5	200	251	1	0.50	0	1	2626
3.420	1.750	70.0	0.020	363.7	200	250	1	0.50	0	1	3107

Table A.1: Experimental 2SLGG loading parameters used in regression model training and validation, as well as *measured* muzzle velocity for reference.

Projectile Package Mass (g)	Primary Powder Mass (g)	Secondary Powder Mass (g)	Burst Disc Score Depth (in)	Piston Mass (g)	Target Tank Pressure (Torr)	Pump Tube Pressure (psi)	Volume Reducers	Piston Tightness	H4831SC	IMR 4831	Projectile Velocity (m/s)
3.403	1.748	75.0	0.020	363.6	200	250	1	0.50	0	1	3234
3.403	1.750	75.0	0.020	363.3	110	250	1	0.50	0	1	3496
3.407	1.750	70.0	0.020	363.2	200	250	1	0.50	0	1	3107
3.405	1.752	83.0	0.020	363.5	110	250	1	0.50	0	1	3807
6.045	1.756	65.3	0.020	363.1	300	250	1	0.50	0	1	2264
6.055	1.749	65.0	0.020	361.9	495	260	1	0.50	0	1	1823
3.395	1.749	130.0	0.020	363.1	80	220	1	0.50	0	1	5933
3.416	1.750	85.0	0.020	363.1	100	250	1	0.50	0	1	3897
2.814	1.750	65.0	0.020	362.7	300	260	1	0.50	0	1	2434
2.198	1.751	65.0	0.020	362.8	305	251	1	0.50	0	1	2670
3.409	1.751	65.0	0.020	362.7	299	249	1	0.50	0	1	2435
3.403	1.751	60.8	0.020	362.9	193	251	1	0.50	0	1	2397
3.401	1.768	60.0	0.020	363.1	215	255	1	0.50	0	1	1966
6.010	1.746	70.0	0.020	364.6	196	256	1	0.50	0	1	2612
3.400	1.755	85.0	0.020	363.4	100	250	1	0.50	0	1	3580
3.384	1.749	60.8	0.020	363.0	215	253	1	0.50	0	1	2496
3.404	1.773	109.9	0.020	363.3	100	200	1	0.50	0	1	5307
3.366	1.750	90.0	0.020	363.2	100	220	1	0.50	0	1	3917
3.352	1.751	90.0	0.020	363.6	100	220	1	0.50	0	1	4409
3.391	1.751	80.0	0.020	363.4	100	220	1	0.50	0	1	3583
6.005	1.752	60.0	0.020	362.9	205	250	1	0.50	0	1	1816
6.013	1.752	60.0	0.020	363.1	197	250	1	0.50	0	1	1972
6.002	1.756	65.0	0.020	363.2	200	250	2	0.50	0	1	2629
6.048	1.751	65.0	0.020	363.4	199	250	2	0.50	0	1	2624
2.148	1.750	125.0	0.014	363.5	85	220	1	0.50	0	1	6369
2.122	1.752	125.0	0.014	363.6	85	272	1	0.50	0	1	5715
2.035	1.751	125.0	0.014	363.3	85	242	1	0.50	0	1	6018
2.139	1.751	125.0	0.014	363.5	85	242	1	0.50	0	1	6192

Table A.1: Experimental 2SLGG loading parameters used in regression model training and validation, as well as *measured* muzzle velocity for reference.

Projectile Package Mass (g)	Primary Powder Mass (g)	Secondary Powder Mass (g)	Burst Disc Score Depth (in)	Piston Mass (g)	Target Tank Pressure (Torr)	Pump Tube Pressure (psi)	Volume Reducers	Piston Tightness	H4831SC	IMR 4831	Projectile Velocity (m/s)
2.062	1.749	55.0	0.020	363.2	200	250	2	0.50	0	1	2493
6.065	1.750	60.0	0.020	363.5	199	251	2	0.50	0	1	2418
2.057	2.000	125.0	0.014	366.0	85	242	1	0.25	0	1	6274
6.041	1.753	75.0	0.020	366.3	102	250	2	0.75	0	1	3106
6.045	1.750	73.0	0.020	366.0	100	251	2	0.00	0	1	3087
2.138	2.001	125.0	0.014	366.1	90	250	1	0.00	0	1	5527
2.141	2.001	125.0	0.014	366.3	90	250	1	0.50	0	1	6070
2.095	1.749	55.0	0.020	366.1	149	249	2	0.00	0	1	2618
2.090	1.750	55.0	0.020	366.2	150	250	2	0.00	0	1	2372
2.052	2.002	125.0	0.014	365.9	92	250	1	0.25	0	1	6140
2.500	2.001	115.0	0.014	366.2	0	220	1	0.00	0	1	5856
2.043	1.999	110.0	0.014	365.8	90	250	1	0.50	0	1	5172
6.013	1.753	55.0	0.020	365.4	199	249	2	0.00	0	1	2116
6.022	1.749	73.0	0.020	366.3	200	250	2	0.00	0	1	2897
6.012	1.749	67.0	0.020	366.1	200	250	2	0.25	0	1	2749
6.026	1.751	67.0	0.020	356.9	199	249	2	0.50	0	1	2631
2.052	1.999	110.0	0.014	365.7	92	249	1	0.75	0	1	5453
6.058	1.752	73.0	0.020	365.9	198	251	2	0.50	0	1	2858
6.018	1.748	60.0	0.020	366.2	195	249	2	0.50	0	1	2520
6.014	1.750	57.0	0.020	366.0	196	250	2	0.25	0	1	2358
3.396	1.750	64.0	0.020	366.0	99	180	2	0.75	0	1	3207
3.376	1.751	75.0	0.020	366.2	99	251	2	1.00	0	1	3791
3.414	1.753	86.0	0.020	366.2	98	250	2	0.50	0	1	4244
3.367	1.750	100.0	0.020	366.1	85	250	2	0.50	0	1	4839
3.381	1.999	150.0	0.020	366.0	80	250	1	0.25	0	1	6542
2.007	1.750	55.0	0.020	366.0	149	250	2	0.25	0	1	2553
2.018	1.750	55.0	0.020	366.1	146	249	2	0.25	0	1	2486
3.367	2.002	125.0	0.020	366.2	80	230	1	0.25	0	1	5603

Table A.1: Experimental 2SLGG loading parameters used in regression model training and validation, as well as *measured* muzzle velocity for reference.

Projectile Package Mass (g)	Primary Powder Mass (g)	Secondary Powder Mass (g)	Burst Disc Score Depth (in)	Piston Mass (g)	Target Tank Pressure (Torr)	Pump Tube Pressure (psi)	Volume Reducers	Piston Tightness	H4831SC	IMR 4831	Projectile Velocity (m/s)
3.368	2.000	135.0	0.020	365.8	75	230	1	0.00	0	1	6144
3.371	2.001	130.0	0.020	366.1	80	231	1	0.00	0	1	5997
1.995	1.749	68.0	0.020	365.8	150	250	2	0.50	0	1	3057
2.001	1.750	68.0	0.020	365.9	147	249	2	0.75	0	1	3083
3.370	2.000	130.0	0.020	365.5	80	230	1	0.00	0	1	5924
2.056	2.002	110.0	0.014	366.4	90	250	1	0.75	1	0	5155
2.047	2.001	130.0	0.014	366.2	80	250	1	0.75	1	0	6317
2.060	2.002	125.0	0.014	366.1	80	250	1	0.50	1	0	5996
2.058	2.000	125.0	0.014	366.0	80	250	1	1.00	1	0	6351
2.074	1.755	68.0	0.014	366.1	100	250	2	1.00	0	1	3646
2.008	1.750	94.0	0.020	366.1	125	250	2	0.75	0	1	4768
2.001	2.021	130.0	0.014	366.1	100	250	1	1.00	1	0	6313
1.999	1.751	85.0	0.020	366.0	125	250	2	0.50	1	0	3628
3.380	2.001	127.0	0.020	365.6	80	230	1	0.00	0	1	5928
3.412	1.998	125.0	0.020	365.4	80	230	1	0.00	0	1	5747
3.376	2.003	150.0	0.020	366.0	80	250	1	0.50	1	0	6286
3.369	2.003	157.0	0.020	367.1	80	250	1	1.00	1	0	6355
2.003	1.750	88.0	0.020	366.5	125	250	2	0.75	1	0	3863
1.990	1.751	89.0	0.020	366.6	125	250	2	1.00	1	0	4232
2.005	1.751	88.5	0.020	366.8	125	250	2	1.00	1	0	4306
3.395	1.751	65.0	0.020	367.4	99	180	2	1.00	0	1	3449
3.370	2.004	110.0	0.020	365.8	85	240	1	1.00	1	0	5063
3.370	2.000	160.0	0.020	365.8	80	250	1	1.00	1	0	6791
3.415	1.006	160.0	0.020	366.8	80	250	1	1.00	1	0	6700
2.000	1.755	88.0	0.020	365.7	125	250	2	0.00	1	0	3819
2.013	1.751	88.0	0.020	366.0	125	250	2	0.50	1	0	4289
2.012	1.751	88.5	0.020	366.5	125	250	2	0.50	1	0	3867
1.968	1.751	88.0	0.020	366.5	125	250	2	1.00	1	0	4471

Table A.1: Experimental 2SLGG loading parameters used in regression model training and validation, as well as *measured* muzzle velocity for reference.

Projectile Package Mass (g)	Primary Powder Mass (g)	Secondary Powder Mass (g)	Burst Disc Score Depth (in)	Piston Mass (g)	Target Tank Pressure (Torr)	Pump Tube Pressure (psi)	Volume Reducers	Piston Tightness	H4831SC	IMR 4831	Projectile Velocity (m/s)
2.035	1.750	65.0	0.020	366.6	125	250	2	1.00	1	0	3197
2.039	1.755	65.0	0.020	363.5	126	251	2	1.00	1	0	2982
2.029	1.754	65.0	0.020	366.7	125	250	2	1.00	1	0	3086
2.029	1.756	65.0	0.020	365.7	125	250	2	1.00	1	0	3141
2.031	1.756	65.0	0.020	366.9	125	250	2	1.00	1	0	3131
2.029	1.752	65.0	0.020	365.1	125	250	2	1.00	1	0	3028
2.037	1.751	65.0	0.020	366.8	125	250	2	1.00	1	0	2964
2.042	1.748	65.0	0.020	366.5	125	250	2	1.00	1	0	3107
2.045	1.752	65.0	0.020	366.0	125	250	2	1.00	1	0	3167
2.026	1.750	65.0	0.020	366.0	125	250	2	1.00	1	0	3050
1.981	1.750	88.0	0.020	367.7	125	250	2	0.50	1	0	4625
1.991	1.751	88.0	0.020	367.0	125	250	2	0.50	1	0	4312
2.034	1.751	65.0	0.020	366.8	125	250	2	1.00	1	0	3186
1.982	1.751	65.0	0.020	366.0	125	250	2	1.00	1	0	3067
2.043	1.749	65.0	0.020	366.7	125	250	2	1.00	1	0	3106
1.974	1.749	88.1	0.020	366.2	125	250	2	0.50	1	0	3659
1.982	1.750	130.0	0.014	366.0	125	250	1	0.00	1	0	6219
6.063	1.749	55.0	0.020	362.9	250	250	2	0.00	1	0	1785
1.979	1.751	100.0	0.014	366.0	115	250	1	0.00	1	0	4088
2.085	1.750	115.0	0.014	362.9	95	250	1	0.00	1	0	4706
1.988	1.750	75.0	0.014	363.1	110	250	2	0.00	1	0	3123
1.988	1.751	72.0	0.020	363.1	100	250	2	0.00	1	0	2835
1.971	1.749	120.0	0.014	363.1	89	250	1	0.00	1	0	4902
6.059	1.751	55.0	0.020	363.1	275	250	2	0.00	1	0	1558
6.026	1.751	56.0	0.020	363.1	225	250	2	0.00	1	0	1996
1.986	1.750	73.0	0.020	362.8	100	250	2	0.00	1	0	2997
1.986	1.754	128.0	0.014	362.9	83	250	1	0.00	1	0	5485
1.979	1.748	121.0	0.014	362.9	90	250	1	0.00	1	0	4866

Table A.1: Experimental 2SLGG loading parameters used in regression model training and validation, as well as *measured* muzzle velocity for reference.

Projectile Package Mass (g)	Primary Powder Mass (g)	Secondary Powder Mass (g)	Burst Disc Score Depth (in)	Piston Mass (g)	Target Tank Pressure (Torr)	Pump Tube Pressure (psi)	Volume Reducers	Piston Tightness	H4831SC	IMR 4831	Projectile Velocity (m/s)
1.979	1.750	100.0	0.014	362.9	115	250	2	0.00	1	0	4649
1.979	2.002	100.0	0.014	363.0	114	250	2	0.00	1	0	4663
1.972	1.752	121.0	0.014	363.2	95	250	1	0.00	1	0	5899
5.979	1.750	65.0	0.020	362.6	195	251	2	0.00	1	0	1987
5.925	1.752	67.0	0.020	363.4	195	250	2	1.00	1	0	2687
6.030	1.750	75.0	0.020	370.3	105	249	2	0.50	1	0	2619