LA-UR-19-28368 (Accepted Manuscript)

Analysis of NIF scaling using physics informed machine learning

Hsu, Abigail Elaine
Cheng, Baolian
Bradley, Paul Andrew

To be published in: Physics of Plasmas

DOI to publisher's version: 10.1063/1.5130585

Permalink to record: http://permalink.lanl.gov/object/view?what=info:lanl-repo/lareport/LA-UR-19-28368

Analysis of NIF scaling using physics informed machine learning

Abigail Hsu*
The Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11790, USA

Baolian Cheng† and Paul A. Bradley‡
Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Over 120 DT ice layer thermonuclear (TN) ignition experiments in inertial confinement fusion (ICF) were conducted on the National Ignition Facility (NIF) in the last eight years. None of the experiments achieved ignition. In fact, the measured neutron outputs from the experiments were well below what was expected. Although experiments to fine-tune the target designs are the focus of the national ICF program, insightful analysis of the existing data is a pressing need. In highly integrated ignition experiments, it is impossible to vary only one design parameter without perturbing all the other implosion variables. Thus, to determine the nonlinear relationships between the design parameters and performance from the data, a multivariate analysis based on physics models is necessary. To this end, we apply machine learning and deep learning methods to the existing NIF experimental data to uncover the patterns and physics scaling laws in TN ignition. In this study, we focus on the scaling laws between the implosion parameters and neutron yield using different supervised learning methods. Descriptions, comparisons, and contrasts between the methods are presented. Our results show that these models are able to infer a relationship between the observed stagnation conditions and neutron yields. This exploratory study will help build new capabilities to evaluate capsule designs and provide suggestions for new designs.

I. INTRODUCTION

Achieving ignition in the laboratory is one of the "grand challenge" problems in science, and the National ignition Facility (NIF) [1] was constructed to accomplish this task. In the last several years, National Ignition Facility (NIF) has made significant progress in the improvement of hohlraum, implosion symmetry and capsule performance. A record yield of 1.9×10^{16} fusion neutrons [2, 3] was produced in 2017. Since then, no ignition capsule has achieved a higher yield, in spite of considerable effort. Like many other physical systems, NIF ignition capsules are designed using analytical models and computer simulations. There are many model parameters that take into account the various components of the system to determine the energy efficiency, performance, and reliability of the design. NIF ignition capsules are imploded with marginal energy drive and compressed to extreme conditions of high temperatures, pressures and densities. Thus, any perturbation could affect the performance of these capsules. Improving the implosions becomes a multi-dimensional optimization problem, which gives rise to the question of how best to optimize future capsules, hohlraums and drive pulses to obtain higher yields and possibly ignition. Early in 2009, Murari et al. [4] started to use machine learning techniques to investigate issues in magnetic confinement fusion. Baltz et al. [5] developed the "Optometrist Algorithm"' through Fusing human and machine interaction to find the important parameter settings for the plasma fusion experiments. Machine learning tools are designed to handle large data sets that are sensitive to many different parameters, and some have been adapted to Inertial Confinement Fusion (ICF) problems [6–11]. The paper of Gopalaswamy et al. [9] is especially interesting as they use a statistical model and experiments to iterate to a better cryogenic DT capsule design and laser pulse. Hatfield et al. [11] used a Gaussian process-based machine learning algorithm to investigate the ICF designs on NIF. It is hoped that machine learning tools might suggest a better NIF capsule design that can achieve higher yields. Here, we start by examining the ability of machine learning tools to model the existing cryogenic DT ice layer shots performed on the NIF and further understand the burn physics in ICF.

Traditional machine learning methodology has demonstrated successes in object classifications[12, 13]. Supervised learning tasks are straightforward to solve when large quantities of data are available[14, 15]. For systems with small sets of labeled data, physics informed learning will be extremely important to help overcome the challenge of training on small datasets. In inertial confinement fusion, we usually do not have enough experimental measurements to train a machine learning model using experimental data alone. Gaffney et al.[16] developed a Bayesian framework to use experimental data to update the physics through the calibration of ICF simulations data. Humbird et al. [8] used deep jointly-informed neural network to calibrate from post-shot simulations of experiments. However, we could use physics analysis and knowledge to shortcut and train a machine learning model that is a good reflection of reality via building a model relationship between independent inputs and a dependent, measurable output [5, 17, 18].

In this work, we analyze the scaling laws between the implosion parameters and neutron yield by using dif-

ferent supervised machine learning methods including regression, such as k-Nearest Neighbors Regression (k-NNR), Polynomial Regression (PR), Support Vector Regression (SVR), sparse heteroscedastic Gaussian Process (GPz) developed by Almosallam [19], as well as deep learning methods such as Deep Neural Network (DNN) and the Deep Jointly-Informed Neural Network (DJINN) developed by LLNL[20]. We examine the predictive capabilities of these models to determine their ability to predict the outcomes from the trained experimental data. We will compare the predictions from various machine learning models with both the experimental data and theory.

We provide a brief description of ignition theory, the layered DT implosion dataset we use, machine learning regression methods and performance metric comparison in section II. In section III, we resent our results for single and multiple variable input predictions. Conclusions and discussion of our results are summarized in section IV.

II. DATA AND METHODS

A. Physics

A series of fundamental physics studies in fusion science [21–26] and other sources [3] indicate that the neutron yield of ignition capsules scales with a power law of the peak implosion velocity of the capsule, $Y_n \propto V_{imp}^{\Gamma}$, where $\Gamma \simeq 6-18$ depending on the adiabat of the pusher ($\Gamma \simeq 6$ for low adiabat and $\Gamma \simeq 18$ for very high adiabat [21]). In addition, the theory also demonstrated a power-law dependence of the neutron yield on the ion temperature of the hot spot, $Y_n \propto T^{\beta}$ as presented by Cheng et al. (2018)[22].

B. Data

Many ignition experiments have been conducted on the NIF since 2011 including the low foot[27], high-foot[28], high density carbon (HDC)[3] and Bigfoot [29] ignition capsules. Various ablator materials have been used in these experiments, including doped CH, HDC and Be[30]. During this time, significant improvements in hohlraum design and operation were achieved [28, 31–33]. In particular, hohlraum modeling became more predictive with the switch from high density gas fills to low density gas fills, which reduced the backscatter from Laser-Plasma Instabilities (LPI), among other deleterious effects.

The NIF data set of DT ice layer capsule implosions shows a clear performance distinction between capsules in low and high density gas fill holraulms [32]. In this work, we divided the NIF data into two groups for our analysis. The first group (Group I) are the capsules with high gas fill density hohlraums and the second group (Group II) consists of capsules in low gas fill density hohlraums. Group I consists mostly of shots before 2016, while Group II contains the more recent shots between 2016 and 2019.

We are interested in the neutron yield (Y_n) as a function of peak implosion velocity (V_{imp}) and hot spot temperature (T_{ion}) in this initial investigation because the yield is most sensitive to these two quantities [22]. We display the raw data of neutron yield vs. hot spot temperature for NIF capsules in the two groups as shown in Fig. 1. The points in red solid circles (and red solid line) from Group I have a scaling law of $Y \propto T^{3.97}$. The data points in blue solid squares (and blue dashed line) from Group II have a scaling law of $Y \propto T^{7.61}$. The different scaling patterns in the two groups are obvious.

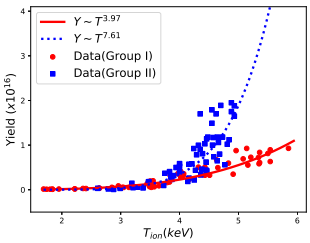

FIG. 1: Yield versus hot spot temperature from Group I (red line and circles) and Group II (blue line and squares) NIF shots.

Similarly, the relationship of raw data of neutron yield versus peak implosion velocity for the NIF capsules from 2011 to 2019 shots in the two groups is presented in Fig. 2. Again, the data points in red solid circles are shots from Group I, which are fit by a power law of $Y \propto V^{16.27}$, shown as the solid red line. The data points in blue solid squares are collected from Group II which are fit by a power law of $Y \propto V^{11.12}$ and shown as the dashed blue line. The separation of two groups is not as obvious as in Fig. 1. This is mainly because of the non-unique relationship between capsule yield and peak implosion velocity. In theory, the neutron yields of capsules scales with peak implosion velocity as $Y = \psi V_{imp}^{\Gamma}$, where coefficient ψ and power index Γ are all functions of the adiabat of the pusher (cold fuel + remaining ablator). A higher adiabat pusher leads to a lower ψ and a higher Γ. Accordingly, a lower adiabat pusher would correspond to a higher ψ and lower Γ. Recent analysis [34] indicates that the pusher adiabat in Group I might be actually higher than the pusher adiabat in Group II because the preheat and implosion quality were better controlled in the later experiments than in Group I. This will be discussed more in subsequent sections.

The correlation matrices between NIF variables including yield, radius ratio of hot spot over ablator

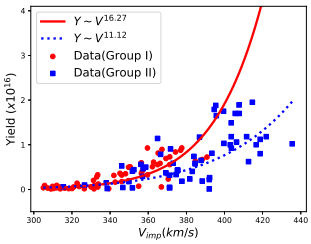

FIG. 2: Yield versus peak implosion velocity from Group I (red line and circles) and Group II (blue line and squares) NIF shots.

($\frac{P_0(hs)}{P_0(abl)}$), down-scattered ratio (DSR), peak implosion velocity (V_{imp}), hot spot temperature (T_{ion}), laser energy (LE), laser energy absorbed (LE_{abs}), and hot spot shape (P_2/P_0) from Groups I and II are shown in Figs. 3 and 4 respectively. Note that the correlation matrices in Figs. 3 and 4 are symmetric about the 45° axis so that the top right color is identical to the bottom left. The strongest correlation is between yield and hot spot T_{ion} in both groups. However, the correlation is weaker for Group II compared to Group I. The second strongest correlated pair is yield versus V_{imp}. Both LE and LE_{abs} are also highly correlated with yield. However, the yield is weakly correlated with the down scattering ratio (DSR) that is proportional to the total areal density of the capsule and hot spot shape metric (P_2/P_0). Particularly, the NIF data in Group I showed a negative correlation of performance to DSR. Such a behavior is consistent with the theory given by Cheng et al [21], where the pusher was shocked or preheated to a higher than designed adiabat, which is counter to simple intuition.

In physical terms, the neutron yield of a capsule directly depends on the areal density of the hot spot, but not the total areal density or DSR of the capsule, although a non-linear relationship exists between the two. In the NIF shots with a higher adiabat, for example the shots of Group I have a relatively higher fraction of the implosion kinetic energy distributed to the pusher and a lower fraction of the implosion energy into the hot spot. Thus, the pushers for capsules of Group I were more compressed (relatively higher DSRs), and the hot spots less compressed. Therefore the neutron yields were lower in this group even though the total areal densities of the capsules were higher. This is why an anti-correlation between the neutron yields and the total areal densities (or DSRs) of the capsules in Group I was observed in Fig. 3. Such an anti-correlation will gradually be reduced as the size and temperature of the hot spots increase.

C. Machine learning regression methods

Machine learning (ML) regression methods predict the output of a given task based on experience from specified inputs and outputs [35]. The collection of the training input is defined as $X = \{x_i\}$ and the observed output is defined as $Y = \{y_i\}$. Given the inputs $X = \{x_i\}_i^n \in R^{n\times d}$, where n is the number of samples in the data set and d is the dimensionality of the input. ML algorithms turn a data set into a model for supervised, unsupervised, classification, regression, etc. In supervised learning, the target label or answer is provided for the training data set. Training and evaluation turn supervised learning algorithms into models to find the set of values that best matches the provided target data by optimizing their parameters. To use numeric data for machine learning regression, it is best to normalize the data to avoid numerical convergence issues arising from large dynamic range. In this work, we normalize the inputs and outputs using min-max normalization method and map the data to the range of 0 and 1. There are dozens of machine learning algorithms for non-linear regression, of which we explored six of these algorithms below and compare their advantages and disadvantages.

All of the constructed models are based on open-source software libraries such as Tensorflow, Scikit-learn, and high-level optimized algorithms such as DJINN, GPz. Tensorflow [36] is massive array manipulation library for designing, building and training networks. It provides a variety of different toolkits and focuses on deep learning algorithms. Tensorflow is customizable and is able to scale up to large numbers of data points. Scikit-learn [37] is a Python module integrating a wide range of machine learning algorithms and the library is focused on modeling data. It has highly-optimized algorithms, but lacks the ability to scale up to extremely large data sets.

1. *K-Nearest Neighbors Regression*

The simplest method is k-nearest neighbors regression (k-NNR) algorithm is a non-parametric method where the input consists of k closest neighbors and the output value is the average of the values of k nearest neighbors [38]. This method does not assume an explicit form of the data, which provides a flexible approach. The optimal value for k depends on the bias-variance trade off. A small value for k provides the most flexible fit and will have a low bias but high variance, while a larger value of k provides a smoother fit. The library "sklearn.neighbors.KNeighborsRegressor" uses $k = 5$ as the default value for n_neighbors. A common suggested rule is to set $k = \sqrt{n}$, where n is the number of samples in the training set. The advantage is that this is a simple technique to implement and no assumptions about the characteristics of the data are needed. However, it easily picks up noisy or irrelevant features. In addition, it

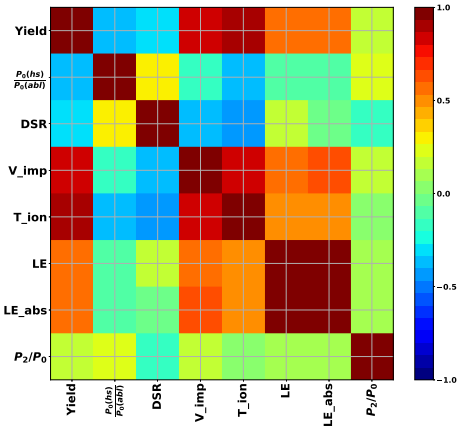

FIG. 3: Variable correlation matrices from Group I NIF shots. Note that the matrix is symmetric about the 45° axis. The yield is most strongly correlated with T_{ion} and V_{imp}.

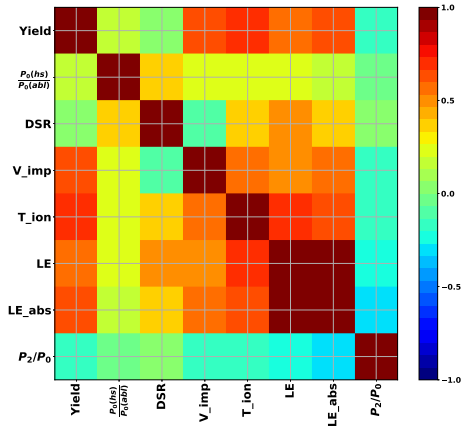

FIG. 4: Variable correlation matrices from Group II NIF shots. Note that the matrix is symmetric about the 45° axis. The yield is most strongly correlated with T_{ion}, LE_{abs}, V_{imp} and LE.

is computationally expensive to find the k nearest neighbors, and to keep track of all the training data, especially when the training dataset is very large.

2. *Polynomial Regression*

Polynomial regression [39] is a form of regression analysis in which the relationship between the independent variable $\{x_i\}$ and dependent variable $\{y_i\}$ is modeled as an n^{th} degree polynomial of $\{x_i\}$. The Polynomial Features routine from the scikit-learn library generates a new feature matrix consisting of all polynomial combinations of the features with degree less than or equal to the specified degree n. Choosing a degree n that is too high or too low can cause overfitting or underfitting respectively. Thus, the theoretical scaling power laws derived earlier will be helpful to guide the choice of n.

3. *Support Vector Regression*

The foundations of support vector regression (SVR) were developed by Vapnik [40]. SVR constructs a hyperplane maximizing the margin which can be seen as regularize the solution by minimizing w, where w is a weight vector that is normal to the hyperplane. For simplicity, we use linear regression (as an example) to introduce the mathematical concepts. The goal is to find a function $f(x_i) = wx_i + b$, where b is the threshold value. The function $f(x_i)$ is fit within a margin of tolerance ϵ from the value $y(x_i)$ for every data point, i.e. $|y(x_i) - f(x_i)| < \epsilon$, where the error function is an epsilon-insensitive loss function which means that errors less than ϵ are ignored. The optimization problem is defined as fellows:

$$\underset{w}{\text{minimize}} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i{}^*)$$

$$\text{subject to} \begin{cases} y_i - w^T x_i - b \leq \epsilon + \xi_i \\ w^T x_i + b - y_i \leq \epsilon + \xi_i \\ \xi_i \geq 0, \xi_i{}^* \geq 0. \end{cases}$$

where C is a parameter that controls the amount of influence of the error and $\|w\|^2$ controls the complexity of the regression function [41]. The kernel functions $\phi(.)$ such as linear, polynomial (poly), radial basis function (rbf), sigmoid from sklearn.svm class which transforms the lower dimensional data into a higher dimensional feature space, that is $\langle \phi(X), \phi(Y) \rangle = K(X, Y)$. The most commonly used non-linear kernels are poly kernel maps to high dimensional d, $K(X, Y) = (X^T Y + 1)^d$ and rbf kernel $K(X, Y) = exp(\frac{-1}{2\sigma}\|X - Y\|^2)$, where σ is a free parameter. The global of minimizing in SVR is always achievable by using convex optimization function.

4. Sparse Heteroscedastic Gaussian Process

Gaussian Process (GP) [42] is a popular and powerful nonparametric, Bayesian supervised machine learning algorithm. The key benefit of GP is rooted in probability and Bayesian inference. Its greatest practical advantage is that it can give a reliable estimate of its own uncertainty. Unlike many supervised ML algorithms that estimate every parameter in a function, the Bayesian approach describes a probability distribution over functions. On a linear function $f(x_i) = wx_i + b$, where w and b are parameters, Bayesian approach specifies a prior distribution, $p(w)$ on the parameter w using Bayes' Rule [42]:

$$\begin{aligned} p(w|y,X) &= \frac{p(y|X,w)\cdot p(w)}{p(y|X)} \\ \text{posterior} &= \frac{\text{likelihood}\cdot\text{prior}}{\text{marginal likelihood}}. \end{aligned} \tag{1}$$

The predictive distribution for unseen testing points, x^*, can be calculated by weighting all possible predictions using the posterior distribution from Eq. (1) as:

$$p(f^*|x^*,y,X) = \int_w p(f^*|x^*,w)\cdot p(w|y,X)\,dw\ .$$

With the GP prior knowledge, it can be easily incorporated into functions through the selection of the mean and covariance functions by summing the label distribution and noise distribution. A class of functions referred to as Mercer kernels [43] guarantee the function maps to a valid covariance matrix, i.e. symmetric and positive semidefinite. The time complexity cost for the GP training is $O(n^3)$ from inverting the $n \times n$ covariance matrix.

A GP-based ML algorithm, Sparse Heteroscedastic Gaussian Process (GPz) [19] is an ML regression algorithm developed to reduce the complexity cost from $O(n^3)$ to $O(nm^2)$ where $m(\leq n)$ is the number of basis functions. It allows for non-stationarity by minimizing the number of required basis functions. Its success was demonstrated in estimating the galactic redshifts of galaxies in astrophysics [44] and the photometric redshift in cosmology experiments [45, 46]. Recently, Hatfield et al. [11] has applied this algorithm to yield predictions of ICF experiments on NIF.

The sparse Gaussian processes is described as basis function models (BFM) with assumptions that the observed target outputs y_i is generated by a linear combination of m non-linear basis functions $\phi(x_i) = [\phi_1(x_i),\ldots,\phi_m(x_i)]^T \in R^m$, and additive noise $\epsilon_i \sim \mathcal{N}(0,\sigma^2)$ described as [47]:

$$y_i = \phi(x_i)^T w + \epsilon_i$$

where w is a length m weights vector of the basis functions. The conditional predictive distribution is constructed as follows:

$$p(y|x) \sim \mathcal{N}(\mu(x),\sigma^2(x))\ ,$$

where μ is the centre mean function and σ is the covariance (kernel) function.

The basis function used in GPz is the radial basis function (RBF), which is parameterized by its centre mean $\mu_j \in R^d$ and precision ${\sigma_j}^{-1} = \Gamma_j^T\Gamma_j$, where $\Gamma_j \in R^{d\times d}$. The general form of the radial basis function is

$$\phi_j(x_i) = exp\left(-\frac{1}{2}(x_i-\mu_j)^T\Gamma_j^T\Gamma_j(x_i-\mu_j)\right)\ .$$

A variety types of covariance structures, such as a bespoke Γ_j variable covariances named "GPVC" for each basis function j are also developed [19].

The GPz regression Matlab implemented source code are accessible at the OxfordML/GPz GitHub repository. The GPz algorithm requires very little tuning with the model specified. here n is the number of observations in the data. In the GPz library, the important parameters are "m", which represents the number of basis functions to use that indicates the model complexity and "method", such as GL, VL, GD, VD, GC and VC are available for the basis function model choice. We used $m = 80$ and the "VC" basis function model for the two groups of the NIF datasets that achieved the best performance.

5. Deep Neural Network Regression

The deep neural network (DNN) [48] is a feed-forward artificial neural network trained with the back propagation algorithm mapped to multi layer perception. The signals always travel to the output layer direction with substantially weighted interconnections. A given node takes the weighted sum of its inputs, and passes it through a linearly/non-linear activation function. The output of the node then becomes the input of another node in the next layer [49], transformed by using the activation function f. The network is initialized randomly, which leads to a lot of variation in the neuron learning behavior. As a result, the earlier hidden layers learn more slowly than later hidden layers. A typical one hidden layer can be mathematically described with weight w and bias b by Eq. (2):

$$z = f(\sum_{i=1}^{n} x_i w_i + b). \tag{2}$$

The most commonly used activation function is a sigmoid function, which maps to values between 0 and 1

[50], defined as:

$$f(x) = sigmoid(x) = \frac{1}{(1+e^{-x})} \tag{3}$$

The use of back propagation with gradient descent is the attraction behind the efficiency of the deep learning model. It is an efficient algorithm computing the gradient of the loss function with respect to the parameter θ. The non-linear optimization method gradient descent adjust weights to reduce value of the error function until it converges. The standard processes are :
1. build multiple hidden layers with an appropriate activation function take in the training input data X
2. compute the estimates of output $f(X;\theta)$
3. measure the difference between the true target Y and the estimates via the loss function
4. minimize the difference by computing the gradient of the loss function with respect to the parameter θ via backpropagation
5. predict the output for test data set using the best parameters tuned through the training data set

The main advantage of neural network is its ability to outperform nearly all traditional machine learning algorithms in multiple domains. The well-known disadvantage of neural networks is the "black box" nature of the algorithm, which means that we do not know how and why a certain output is generated. Neural networks usually require much more data to train them than traditional machine learning algorithms.

6. *Deep Jointly-Informed Neural Network Regression*

Some researchers have investigated the possibility of mapping decision tree (DT) and random forests into neural networks. Humbird et al. [20] proposed a deep jointly-informed neural networks (DJINN) algorithm maps ensemble of DTs trained on the data into a collection of initialized NNs that would then be trained by backpropagation, which was developed at LLNL. The authors presented compelling evidence suggesting that DJINN represents a robust algorithm that can generate accurate NNs for a variety of datasets. The first step of the DJINN algorithm is constructing the ensemble of decision trees. The representation for the DT model is a binary tree. The second step is mapping from trees to neural networks as a guide for the network architecture and weight initialization, and then neural networks are initialized layer by layer. The third step is tuning the weights in neural networks by backpropagation. DJINN uses rectified linear unit activation function at each hidden layer and an Adam optimizer to minimize the cost function.

The regression source code is accessible at the LLNL/DJINN github directory and there are three main tuning parameters for this model: the number of trees represents number of neural nets in the ensemble, max depth of the tree is used to optimize for each data set, and dropout probability for Bayesian approximation.

D. Performance Metrics Comparison

Performance metrics measure the regression error to evaluate frameworks in various fields. In machine learning regression, performance metrics are used to compare the trained model predictions with the observed values from the test data set [51]. In addition, performance metrics are used to evaluate the capability of discriminating among different model results. A vast variety of performance metrics have been described in the literature and we focus on four metrics on point estimates without incorporating the quality of uncertainty estimates.

The most commonly used metric in research is the mean square error (MSE), which is the average of the square of the errors of the predictions. The error in this case means the difference between the observed values y_i and the predicted ones $f(x_i)$, that is MSE $= \frac{1}{N}\sum_{i=1}^{N}(f(x_i)-y_i)^2$. The larger the number is, the larger the error is and the worse the model is. While MSE is commonly used, one very bad prediction will magnify the error, and may overestimate the badness of the model.

Mean Absolute Error (MAE) is calculated as an average of absolute differences between the target values and the predictions via the algorithm, that is MAE $= \frac{1}{N}\sum_{i=1}^{N}|f(x_i)-y_i|$. This metric is not sensitive to outliers as MSE.

The R-Squared (R^2) is the proportion of the variance of the depend output y_i from prediction $f(x_i)$. It evaluates how well a model performs and closely related to MSE, but it has the advantage of being scale-free. The value of R^2 is generally between $-\infty$ and 1 and is defined as follows:

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

where $\bar{y}$ is the mean of the observed value y_i. A value of R^2 close to 1 indicates the error is close to zero, that is a model predictions are very close to the observations.

Finally, there is the Explained variance score (ExVar), defined by

$$ExVar = 1 - \frac{Var(Y-f(X))}{Var(Y)} ,$$

which is equal to R^2 when the mean error is zero.

III. RESULTS

A. Single Input Regression Predictions

A comparative assessment on neutron yield and hot spot temperature of the NIF ignition capsules are performed using the six machine learning regression methods. Predictions are obtained and compared to each other, along with the observed experimental yield data. All the supervised methods take the hot spot T_{ion} as input and conduct predictions on the two groups of data. We divide each group data into two sets with an 80-20 percent random split. The training set contains 80% of the data. We use this set of the data to train and tune the parameters for the algorithm. We then use the tuned parameters to predict the testing set, which is the remaining 20% of the data. Because of the small available experimental dataset, the data does not split into validation dataset which evaluates a model fitting on the training dataset in an unbiased manner, while tuning the hyperparameters for the model. Since the testing set is real data not used for fine tuning, the predictive capability of the trained algorithm can then be directly tested against the experimental data.

Our strategy of training and predicting using experimental data differs from other researchers who use thousands of simulations to train the algorithm. In our view, NIF ignition capsules are operating in a marginal energy regime, all of the perturbations are in the sensitive range to affect the performance of the capsules so that most of the pre-shot simulations are not predictive. Thus, the simulations trained algorithm will also have low predictive capability for new designs. Our methodology that uses experimental data only to train the algorithm first and test the algorithm with real data second and eventually predict new designs (the subject of future work), will give more confidence on future designs using the trained and tested methods.

The training data predictions using different methods are shown in Figs. 5 and 6 for the two groups respectively. All of the methods give a reasonable and consistent predictions on the two training data groups. The solid black line represents the mean from the GPz model. The 95% confidence interval (CI), i.e. plus or minus two standard deviations from the mean, is displayed by the grey shaded area. Results suggest that the high uncertainty area may be improved by adding more data points especially for Group II.

TABLE I: Yield prediction statistics for test dataset taking in T_{ion} for NIF shots from Group I.

Models	MAE	MSE	R^2	ExVar
DJINN	0.07	0.01	0.91	0.92
DNN	0.07	0.01	0.87	0.89
k-NNR	0.1	0.02	0.75	0.85
SVR	0.07	0.01	0.88	0.91
PR	0.08	0.01	0.81	0.88
GPz	0.09	0.02	0.78	0.82

TABLE II: Yield prediction statistics for test dataset taking in T_{ion} for NIF shots from Group II.

Models	MAE	MSE	R^2	ExVar
DJINN	0.09	0.01	0.87	0.87
DNN	0.08	0.01	0.87	0.87
k-NNR	0.17	0.06	0.43	0.49
SVR	0.11	0.02	0.76	0.85
PR	0.09	0.02	0.83	0.87
GPz	0.09	0.01	0.79	0.89

Predictions of the six trained algorithms for the test data are plotted against the observed experiment yields in Figs. 7 and 8. The solid line represents predictions equal to experimental values, which indicates the best predictions. The closer the points from a given method are to the solid line, the better the prediction is. The DJINN method performs best for the two groups on average, (where the diamond shapes are generally closer to the solid straight line) as shown in Figs. 7 and 8 respectively. Note that the GPz model consistently gives a better yield prediction than DJINN for relatively lower values and the very highest ones in both groups. We do not yet have a good interpretation for this behavior. Further study and comparison would be needed.

We notice that the machine learning predictions from all methods except the GPz algorithm for the high yield capsules in training data Group II are lower than the actual measured yields, as displayed in the upper-right corner of Figs. 6 and 8. This happens to be a generic feature in machine learning that often ends up underestimating the high values, and overestimating the low values, as the machine learning algorithm is drawn to the middle where most of the data is. The highest data point can't be overestimated because the algorithm has never seen anything higher in it's training set. These under-predicted values are from the high-yield HDC ignition capsule family, which includes the record yield (1.9×10^{16} neutrons) capsule N170827 shot [2] and its variants. The inability of all machine learning methods (except for the GPz) to predict these data reflects the reality in NIF experiments that capsules performance is quite sensitive to various

design perturbations. These HDC shots had a relatively high peak implosion velocity and thin shell, which put these capsules close to the velocity cliff and increased the risk of shell burn-through, allowing excess mix between pusher and cold fuel. More discussions will be given in later section of the paper.

In machine learning science, there is no one algorithm works best for every problem, which is especially relevant for supervised learning methods. The performance metric comparison of different approaches are shown in Tables I and II. The statistics of mean square error (MSE) of different methods are relatively small and have less variation, which are due to the normalized data having values between 0 and 1. All methods perform well on shots in Group I, as shown in Table I, while DJINN outperforms the other methods by having the highest R^2 and ExVar and lowest MSE snd MAE. All the methods are less predictive for the low gas density hohlraum Group II data set, although DNN and DJINN both perform reasonably well, as shown in Fig. 8 and Table II. The GPz method does not outperform DJINN on the testing dataset for most of the metrics, but has a relatively higher ExVar and lower R^2 than DJINN in Group II. The apparent difference between the R^2 and the ExVar for GPz in Group II indicates a non zero mean error in the GPz methodology. Therefore, a few more different metrics seem necessary to illustrate the model goodness on ICF.

It is worthwhile to point out that physics guided machine learning and physics based judgement when preparing the data sets (before using machine learning) are important in this analysis. Similar to the finding in several other fields [5, 18] that combining machine learning and human judgement gives the best results [17]. If the experimental data had not been divided into two groups according to the gas fill density of the hohlraums, (that is, if we performed training and testing on the combined data) the best predictions were obtained by the DNN method ($R^2 = 0.69$), which is a worse performance, as shown in Fig. 9 compared to the separated groups shown in Tables I and II.

The choice of domain variables can also affect the predictive capability of a method. Therefore, we have also studied how the prediction of different methods is affected by different choice of the domain variables, such as taking peak implosion velocity as an single input to predict yield.

Our results show that all the methods — including DJINN and DNN — perform poorly in predicting Y_n for the training and test data sets from both Group I and II using V_{imp} as an input. As an example, In Fig. 10, we plot the yield predictions for Group I using peak implosion velocity using DNN method. The predictive power is poor with a low R^2 of 0.62. The yield predictions for Group II (not shown) is even worse than for Group I. A similar pattern for yield predictions using hot spot temperature is also observed. These observations can be explained by the correlation matrix displayed in Figs. 3 and 4, where the correlation between yield and T_{ion} is stronger in Group I than in Group II and is much stronger than the correlation between yield and V_{imp} in both data groups.

The reason that implosion velocity in the present NIF data is a poor predictor of yield and the predictive power decreases with hot spot T_{ion} is that other yield degradation mechanisms become important as the velocity increases and the measured temperature depends both on compression of the gas and on non-radial flows in the gas. On the NIF, several campaigns increased the implosion velocity to the point that the yield dropped dramatically. This yield drop is typically attributed to ablator burn-through, resulting in a lack of confinement and increased mix. The measurement of T_{ion} comes from Doppler broadening of the 14 MeV DT neutron peak. However, this broadening can result either from higher ion temperature (and greater burn rates) or residual flows in the gas of the hot spot (such as from non-radial flows induced by time dependent low-mode asymmetries). Significant hot spot material flows can appear to give a high burn temperature, but result in degraded yields.

Quantifying the uncertainties of the training data and the predictions of the different methodologies is crucial. Machine learning is traditionally very good at interpolation, but poor at extrapolation. Recently researchers have performed a systematic analysis on the uncertainty of some NIF experimental data [52]. Due to limited scope of this article, we hope to quantify the uncertainties in both interpolation and extrapolation associated with each machine learning method in a future paper.

B. Multiple Inputs Regression Predictions

In order to explore the predictive capability of the machine learning methods, we look into multiple inputs predictions, especially using capsule input parameters. We take four input variables including hot spot T_{ion}, V_{imp}, LE, and LE_{abs} to predict the yield in both groups. We see that the yield predictions in both groups using these multiple inputs are worse than using the single hot spot T_{ion} variable, especially in the high yield region, as demonstrated in Figs. 11 and 12. Such a trend is expected because high peak implosion velocity could result in less remaining ablator mass, shell-burn-through and lower neutron yields. In addition, many other physical factors played roles during the capsule implosion between the peak implosion velocity time and the stagnation. Thus, the training data in the high velocity regime is pretty much scattered so that the learning in that part of parameter space is not well trained.

Implosion symmetry and hot spot shape play important roles in the thermonuclear performance of the NIF

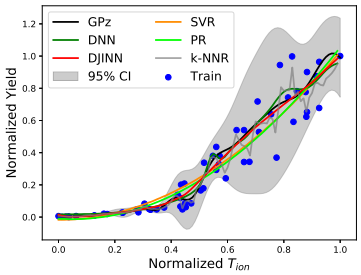

FIG. 5: Yield predictions of various methods for training dataset taking hot spot T_{ion} as input for NIF shots from Group I.

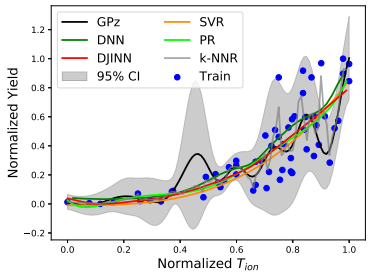

FIG. 6: Yield predictions of various methods for training dataset taking hot spot T_{ion} as input for NIF shots from Group II.

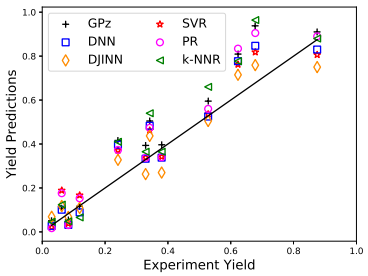

FIG. 7: Yield predictions of various methods for test dataset, taking hot spot T_{ion} as input for NIF shots from Group I.

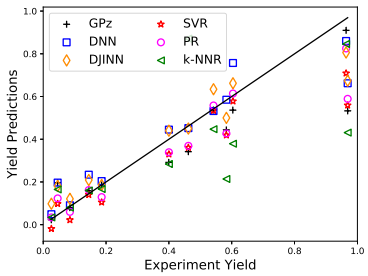

FIG. 8: Yield predictions of various methods for test dataset, taking hot spot T_{ion} as input for NIF shots from Group II.

ignition capsules [25, 52, 53]. The hot spot shape is often characterized by low mode Legendre Polynomials, with a common example being the ratio of the second order to zero order Legendre polynomials (P_2/P_0). If P_2 is negative, it represents an oblate hot-spot shape and if P_2 is positive, it indicates the hot spot is prolate. A spherical hot spot shape has a P_2 value of zero. If we include the effects of hot spot shape by adding the ratio P_2/P_0 as a second input variable (along with the hot spot temperature T_{ion}) to be trained, it will not improve the ability to predict the yield, as demonstrated in Figs. 13 and 14. For example, the performance metric R^2 for the DJINN method is reduced slightly from 0.87 to 0.74 compared with the single input regression performance metric for Group II.

In general, the predictive capability of a machine learning algorithm should not get worse by adding extra information to the algorithm if the information is essential and the system is stable. However, for the NIF capsules, at the current laser power, the implosion kinetic energy of the ignition capsules is not adequate enough to produce a robust implosion that is sustainable to unexpected perturbations. In fact, the thermonuclear burn in the capsules is sensitive to many factors, such as peak implosion velocity, pulse shape, pusher adiabat, and implosion symmetry. Equally weighting these factors resulted in reduced predictive power. Therefore physically selecting and weighting meaningful extra information for inputs is important to improve the predictive capability of the algorithm for new designs.

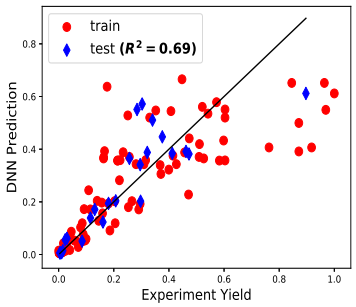

FIG. 9: Yield prediction taking in hot spot T_{ion} in NIF shots from both Groups I and II using DNN.

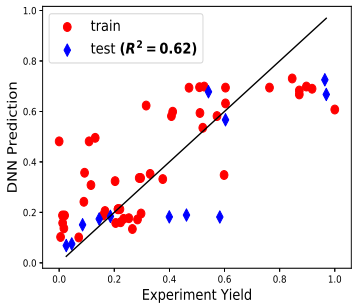

FIG. 10: Yield prediction taking in V_{imp} in NIF shots from Group I using DNN.

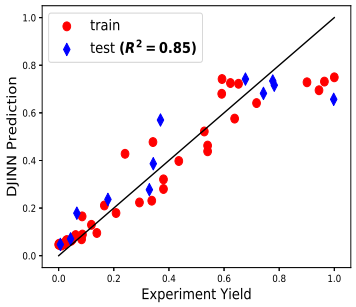

FIG. 11: Yield prediction taking in hot spot T_{ion}, V_{imp}, LE, and LE_{abs} in NIF shots from Group I.

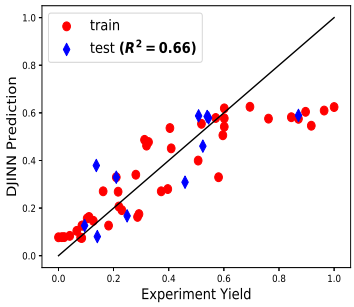

FIG. 12: Yield prediction taking in hot spot T_{ion}, V_{imp}, LE, and LE_{abs} in NIF shots from Group II.

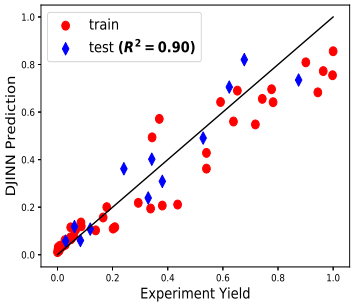

FIG. 13: Yield prediction taking in hot spot T_{ion} and shapes in NIF shots from Group I.

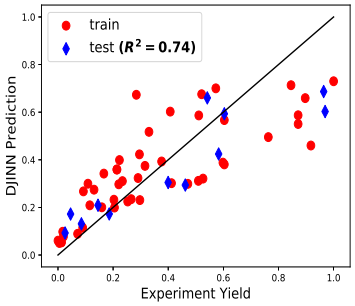

FIG. 14: Yield prediction taking in hot spot T_{ion} and shapes in NIF shots from Group II.

IV. CONCLUSIONS AND DISCUSSIONS

The main purpose of this research is to assess the predictive capability of different machine learning methods on implosion parameters and neutron yields of ignition capsules. In this study, we have done an extensive search for the best parameters to gain the best predictions. Our study demonstrates a remarkable agreement among machine learning methods, simple physics theory and the NIF experiments on physics scaling laws. Our results show that in systems with limited data points, physics

guided machine learning can provide useful predictions to the system. For NIF ignition capsules, the deep learning method can predict the yield reasonably well from hot spot temperatures. The predictions are slightly better for shots from 2011 to end of 2015 (Group I) than for shots from 2016 to 2019 (Group II). The reason might be that more design changes (ablator material and laser pulse adiabat to name two examples) were made in Group II and our machine learning inputs have not captured these design changes.

We observe that the yields prediction power by taking in peak implosion velocity is weaker than the prediction from the hot spot temperature. This weakness could be caused by uncertainties in inferring the velocity. The multiple inputs regression does not add yield prediction power. Our study also show that adding the low mode asymmetry parameters P_2/P_0 of the hot spot in inputs does not increase the predictive capability. The lack of a strong correlation between yield and $P_2/P0$ is also an evidence that NIF is not close to achieving ignition[25].

Theoretically speaking, for a given adiabat, the hot spot pressure should be a strong function of the peak implosion velocity. However, our machine learning study trained by the NIF data indicates that in reality, in the energy regime (marginal) of NIF operations, the peak implosion velocity is not yet solely dominating the performance of these capsules. Many other perturbations such as pusher adiabat also have sizeable impacts on capsule performance. The pusher adiabat, is associated with many design parameters including preheat, pulse shape, instabilities, thickness of ablator, and implosion symmetries. Experimental data shows that with uncontrolled adiabat, the increased performance predicted from increased peak implosion velocity may have well been eliminated by the decreased performance caused by the increased adiabat.

Here we would like to point out that for a given implosion kinetic energy ($M_pV_{imp}^2/2$), increasing peak implosion velocity is at the cost of decreasing the pusher mass M_p, so does the the remaining ablator mass. High resolution simulations [54] and recent Be experiments [55, 56] have shown that large interfacial instabilities would arise in case of low remaining ablator mass, then the shell would be burnt through and the pusher would be pushed into a higher adiabat, which affects the energy partition between pusher and hot spot and results in a low hot spot energy, low hot spot areal density, and low neutron yield. That is why we observe a yield drop at high peak implosion velocity in the NIF shots in Group II. Therefore, in future designs, purely pursuing high implosion velocity without controlling adiabat may not lead to the desired performance improvements.

In this paper, we demonstrate how to use physics guided supervised ML methods to identify meaningful nonlinear correlations and interactive patterns in the data as a complement to traditional regression analysis. From this study, we observe the following: 1) yield and hot spot temperatures of the capsules are strongly correlated. However, temperature is not always a good predictor of yield if there are strong residual kinetic motions in the capsules as shown in Fig. 11 and 12; 2) the correlation between yield and peak implosion velocity is weaker than expected, which is likely due to the existence of a velocity "cliff" where the yield drops off dramatically when there is too little mass remaining in the ablator, causing the shell to break up and increases the mix of material into the hot fuel; 3) deep neural network algorithms, such as DNN and DJINN, outperform machine learning regression methods on different domain variables and data sets, which can be explained by the goodness of the power law fitting in Fig. 1 compared with the power law fitting in Fig. 2; 4) GPz does not outperform overall, but gives the notable best predictions for relatively low and high yield capsules in both groups. 5) other regression methods (k-NNR, SVR) are either not robust or stable in accuracy; and 6) physics guidance learning improves the predictions when a system has a limited quantity of data. In addition, physics informed machine learning is one of the keys for construction and learning from data-driven models, since the "black box" use of ML often leads to false discoveries in scientific applications. For example, if we had not divided the experimental data into two groups, we would be inconsistent in our predictions for the total data set as presented in Fig. 9. This shows the unique advantage of using fundamental physical principles combined with data science to inform the search for a physically meaningful and accurate ML model.

V. FUTURE WORK

In physical point of view, the pusher adiabat is an essential parameter in the energy partition between the hot spot and the pusher shell of the capsules during implosion. Taking into account the effects of the pusher adiabat on implosion and the TN burn of the capsule is very important. Thus adding the pusher adiabat as an additional input parameter in machine learning method would be an important improvement of the current work. However, issue is the pusher adiabat is not measured directly; therefore, calibrated simulations or a "proxy" measurement, such as a physically understood DSR and mass remaining will be needed to create such a plot.

Finally, the current NIF DT ice layer experimental dataset is still very small, so further increasing the amount of experimental data or calibrated simulation data would improve our prediction results. Also exploring the accuracy of the DT ion temperature affected by the mass flows in the hot spot due to low-mode asymmetries and the compressibility of the capsule versus preheat and high mode mix would increase the predictive capability of the methods. Furthermore, quantifying the uncer-

tainties of predictions of the machine learning methods associated with limited training data is another important and necessary task in future studies.

VI. ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable and constructive suggestions, which improved the paper significantly. We are grateful to Dr. O. L. Landen and Dr. C. Cerjan for the complete NIF data set and valuable comments. We thank Dr. J. Glimm, Dr. B. Spears, Dr. L. Peterson, Dr. J. Kline, and Dr. R. Nora for valuable discussions. This work was conducted under the auspices of the U.S. Department of Energy by the Los Alamos National Laboratory under Contract No. 89233218CNA000001.

* abigail.hsu@stonybrook.edu
† bcheng@lanl.gov
‡ pbradley@lanl.gov

[1] G. H. Miller, E. I. Moses, and C. R. Wuest, Nuclear Fusion **44**, S228 (2004).
[2] S. Le Pape, L. F. Berzak Hopkins, L. Divol, A. Pak, E. L. Dewald, S. Bhandarkar, L. R. Bennedetti, T. Bunn, J. Biener, J. Crippen, D. Casey, D. Edgell, D. N. Fittinghoff, M. Gatu-Johnson, C. Goyon, S. Haan, R. Hatarik, M. Havre, D. D. Ho, N. Izumi, J. Jaquez, S. F. Khan, G. A. Kyrala, T. Ma, A. J. Mackinnon, A. G. MacPhee, B. J. MacGowan, N. B. Meezan, J. Milovich, M. Millot, P. Michel, S. R. Nagel, A. Nikroo, P. Patel, J. Ralph, J. S. Ross, N. G. Rice, D. Strozzi, M. Stadermann, P. legov, C. Yeamans, C. Weber, C. Wild, D. Callahan, and O. A. Hurricane, Phys. Rev. Lett. **120**, 245003 (2018).
[3] L. Berzak Hopkins, S. Le Pape, L. D. nd A. Pak, E. L. Dewald, D. D. Ho, N. Bhandarkar, S. Bhandarkar, L. R. Bennedetti, T. Bunn, J. Biener, J. Crippen, D. Casey, D. Clark, D. Edgell, D. N. Fittinghoff, M. Gatu-Johnson, C. Goyon, S. Haan, R. Hatarik, M. Havre, D. Hinkel, H. Huang, N. Izumi, J. Jaquez, O. Joens, S. F. Khan, A. Kritcher, C. Kong, G. A. Kyrala, O. Landen, T. Ma, A. G. MacPhee, B. J. MacGowan, A. J. M. M. Marinak, J. Milovich, M. Millot, P. Michel, A. Moore, S. R. Nagel, A. Nikroo, P. Patel, J. Ralph, H. Robey, J. S. Ross, N. G. Rice, S. Sepke, V. A. Smalyuk, P. Sterne, D. Strozzi, M. Stadermann, P. Volegov, C. Weber, C. Wild, C. Yeamans, D. Callahan, O. A. Hurricane, R. P. J. Town, and M. J. Edwards, Plasma Phys. Control. Fusion **61**, 014023 (2019).
[4] A. Murari, J. C. Vega, D. Mazon, N. F. Martin, G. Rattá, and G. Vagliasindi, "Machine learning methods for data driven theory in the physical sciences with applications to confinement regime identification in nuclear fusion," (2009), preprint.
[5] E. A. Baltz, E. Trask, M. Binderbauer, M. Dikovsky, H. Gota, R. Mendoza, J. C. Platt, and P. F. Riley, *Scientific Reports*, **7**, 6425 (2017).
[6] R. Nora, J. E. F. J. L. Peterson, B. K. Spears, and S. Brandon, Statistical Analysis and Data Mining: The ASA Data Science Journal **10**, 230 (2017).
[7] J. L. Peterson, K. D. Humbird, J. E. Field, S. T. Brandon, S. H. Langer, R. C. Nora, and B. K. Spears, Phys. Plasmas **24**, 032702 (2017).
[8] K. D. Humbird, J. L. Peterson, and R. G. McClarren, "Transfer learning to model inertial confinement fusion experiments," (2018), arXiv:1812.06055 [cs.LG].
[9] V. Gopalaswamy, R. B. J. P. Knauer, N. luciani, D. Patel, K. M. Woo, A. Bose, I. V. Igumenshchev, E. M. Campbell, K. S. Anderson, K. A. Bauer, M. J. Bonino, D. Cao, A. R. Christopherson, G. W. Collins, T. J. B. Collins, J. R. Davies, J. A. Delettrez, D. H. Edgell, R. Epstein, C. J. Forrest, D. H. Froula, V. Y. Glebov, V. N. Goncharov, D. R. Harding, S. X. Hu, D. W. Jacobs-Perkins, R. T. Janezic, J. H. Kelly, O. M. Mannion, A. Maximov, F. J. Marshall, D. T. Michel, S. Miller, S. F. B. Morse, J. Palastro, J. Peebles, P. B. Radha, S. P.Rregan, S. Sampat, T. C. Sangster, A. B. Sefkow, W. Seka, R. C. Shah, W. T. Shmyada, A. Shvydky, C. Stoeckl, A. A. Solodov, W. Theobald, J. D. Zuegel, M. Gatu Johnson, R. D. Petrasso, C. K. li, and J. A. Frenje, Nature **565**, 581 (2019).
[10] P. W. Hatfield, S. J. Rose, and R. H. H. Scott, Physics of Plasmas **26**, 062706 (2019).
[11] P. Hatfield, S. Rose, R. Scott, I. Almosallam, S. Roberts, and M. Jarvis, IEEE Transactions on Plasma Science , 1 (2019).
[12] S. M. Weiss and I. Kapouleas, *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'89, 781 (1989).
[13] S. Tong and D. Koller, J. Mach. Learn. Res. **2**, 45 (2002).
[14] A. Watkins, J. Timmis, and L. Boggess, Genetic Programming and Evolvable Machines **5**, 291 (2004).
[15] D. Turnbull, G. Lanckriet, E. Pampalk, and M. Goto (2007) pp. 51–54.
[16] J. A. Gaffney, S. T. Brandon, K. D. Humbird, M. K. G. Kruse, R. C. Nora, J. L. Peterson, and B. K. Spears, Physics of Plasmas **26**, 082704 (2019).
[17] B. K. Spears, J. Brase, P. T. Bremer, B. Chen, J. Field, J.Gaffney, M. Kruse, S. Langer, K. Lewis, R. Nora, L. Peterson, J. J. Thiagarajan, B. E. Van, and K. Humbird, Physics of Plasmas **25**, 080901 (2018).
[18] M. R. Beck, C. Scarlata, L. F. Fortson, C. J. Lintott, B. D. Simmons, M. A. Galloway, K. W. Willett, H. Dickinson, K. L. Masters, P. J. Marshall, and D. Wright, Monthly Notices of the Royal Astronomical Society **476**, 5516 (2018).
[19] I. A. Almosallam, S. N. Lindsay, M. J. Jarvis, and S. J. Roberts, Monthly Notices of the Royal Astronomical Society **455**, 2387 (2015).
[20] K. D. Humbird, J. L. Peterson, and R. G. Mcclarren, IEEE Transactions on Neural Networks and Learning Systems **30**, 1286 (2019).
[21] B. Cheng, T. J. Kwan, Y. Wang, and S. H. Batha, Phys. Rev. E **88**, 041101 (2013).
[22] B. Cheng, T. J. Kwan, Y. Wang, S. Yi, S. H. Batha, and F. J. Wysocki, Plasma Physics and Controlled Fusion **60** (2018).

[23] B. Cheng, T. J. Kwan, Y. Wang, and S. H. Batha, Physics of Plasmas **21** (2014).
[24] B. Cheng, T. J. Kwan, Y. Wang, S. Yi, S. H. Batha, and F. J. Wysocki, Physics of Plasmas **23** (2016).
[25] B. Cheng, T. J. Kwan, S. Yi, O. L. Landen, Y. Wang, C. J. Cerjan, S. H. Batha, and F. J. Wysocki, Phys. Rev. E **98**, 023203 (2018).
[26] B. Cheng, T. J. Kwan, Y. Wang, F. E. Merrill, C. J. Cerjan, and S. H. Batha, Physics of Plasmas **22** (2015).
[27] M. J. Edwards, P. K. Patel, J. D. Lindl, L. J. Atherton, S. H. Glenzer, S. W. Haan, J. D. Kilkenny, O. L. Landen, E. I. Moses, A. Nikroo, R. Petrasso, T. C. Sangster, P. T. Springer, S. Batha, R. Benedetti, L. Bernstein, R. Betti, D. L. Bleuel, T. R. Boehly, D. K. Bradley, J. A. Caggiano, D. A. Callahan, P. M. Celliers, C. J. Cerjan, K. C. Chen, D. S. Clark, G. W. Collins, E. L. Dewald, L. Divol, S. Dixit, T. Doeppner, D. H. Edgell, J. E. Fair, M. Farrell, R. J. Fortner, J. Frenje, M. G. Gatu Johnson, E. Giraldez, V. Y. Glebov, G. Grim, B. A. Hammel, A. V. Hamza, D. R. Harding, S. P. Hatchett, N. Hein, H. W. Herrmann, D. Hicks, D. E. Hinkel, M. Hoppe, W. W. Hsing, N. Izumi, B. Jacoby, O. S. Jones, D. Kalantar, R. Kauffman, J. L. Kline, J. P. Knauer, J. A. Koch, B. J. Kozioziemski, G. Kyrala, K. N. LaFortune, S. Le Pape, R. J. Leeper, R. Lerche, T. Ma, B. J. MacGowan, A. J. MacKinnon, A. Macphee, E. R. Mapoles, M. M. Marinak, M. Mauldin, P. W. McKenty, M. Meezan, P. A. Michel, J. Milovich, J. D. Moody, M. Moran, D. H. Munro, C. L. Olson, K. Opachich, A. E. Pak, T. Parham, H. S. Park, J. E. Ralph, S. P. Regan, B. Remington, H. Rinderknecht, H. F. Robey, M. Rosen, S. Ross, J. D. Salmonson, J. Sater, D. H. Schneider, F. H. Séguin, S. M. Sepke, D. A. Shaughnessy, V. A. Smalyuk, B. K. Spears, C. Stoeckl, W. Stoeffl, L. Suter, C. A. Thomas, R. Tommasini, R. P. Town, S. V. Weber, P. J. Wegner, K. Widman, M. Wilke, D. C. Wilson, C. B. Yeamans, and A. Zylstra, Phys. Plasmas **20**, 070501 (2013).
[28] O. A. Hurricane, D. A. Callahan, D. T. Casey, P. M. Celliers, C. J. Cerjan, E. L. Dewald, T. R. Dittrichand D. E. Doppner, D. E. Hinkel, L. F. Berzak Hopkins, J. Kline, S. Le Pape, T. Ma, A. G. MacPhee, J. L. Milovich, A. Pak, H. S. Park, P. K. Patel, B. A. Remington, J. D. Salmonson, P. T. Springer, and R. Tommasini, Nature **506**, 343 (2014).
[29] D. T. Casey, C. A. Thomas, K. L. Baker, B. Spears, M. Hohenberger, S. F. Khan, R. C. Nora, C. R. Weber, D. T. Woods, O. Hurricane, D. A. Callahan, R. L. Berger, J. L. Milovich, P. K. Patel, T. Ma, A. Pak, L. R. Benedetti, M. Millot, C. Jarrott, O. L. Landen, R. M. Bionta, B. J. MacGowan, D. J. Strozzi, M. Stadermann, J. Biener, A. Nikroo, C. S. Goyon, N. Izumi, S. R. Nagel, B. Bachmann, P. L. Volegov, D. N. Fittinghoff, G. P. Grim, C. B. Yeamans, M. Gatu Johnson, J. A. Frenje, N. Rice, C. Kong, J. Crippen, J. Jaquez, K. Kangas, and C. Wild, Phys. Plasmas **25**, 056308 (2018).
[30] D. S. Clark, A. L. Kritcher, S. A. Yi, A. B. Zylstra, S. W. Haan, and C. R. Weber, Phys. Plasmas **25**, 032703 (2018).
[31] D. T. Casey, V. A. Smalyuk, K. S. Raman, J. L. Peterson, L. Berzak Hopkins, D. A. Callahan, D. S. Clark, E. L. Dewald, T. R. Dittrich, S. W. Haan, D. E. Hinkel, D. Hoover, O. A. Hurricane, J. J. Kroll, O. L. Landen, A. S. Moore, A. Nikroo, H. S. Park, B. A. Remington, H. F. Robey, J. R. Rygg, J. D. Salmonson, R. Tommasini, and K. Widmann, Phys. Rev. E **90**, 011102 (2014).
[32] O. A. Hurricane, P. T. Springer, P. K. Patel, D. A. Callahan, K. Baker, D. T. Casey, L. DivoL, T. Doppner, D. E. Hinkel, M. Hohenberger, L. F. Berzak Hopkins, C. Jarrott, A. Kritcher, S. Le Pape, S. Maclaren, L. Masse, A. Pak, J. Ralph, C. Thomas, P. Volegov, and A. Zylstra, Physics of Plasmas **26**, 052704 (2019).
[33] V. A. Smalyuk, H. F. Robey, C. L. Alday, P. Amendt, C. Aracne-Ruddle, J. R. Bigelow, T. Bunn, D. T. Casey, K. C. Chen, D. S. Clark, J. P. Cortez, J. Crippen, S. Diaz, M. Farrell, S. Felker, J. Field, J. Jaquez, S. Johnson, S. W. Haan, B. A. Hammel, A. V. Hamza, M. O. Havre, C. Heinbockel, W. W. Hsing, K. Kangas, J. J. Kroll, S. O. Kucheyev, O. L. Landen, X. Lepro-Chavez, A. G. MacPhee, D. A. Martinez, J. Milovich, A. Nikroo, L. A. Pickworth, N. Rice, M. Stadermann, D. Steich, and C. R. Weber, Physics of Plasmas **25**, 072705 (2018).
[34] B. Cheng, 11th International Conference on Inertial Fusion Science and Applications (IFSA2019), Japan (2019).
[35] T. Mitchell, *Machine Learning*, McGraw-Hill International Editions (McGraw-Hill, 1997).
[36] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, , 265 (2016).
[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondeland P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, J. Mach. Learn. Res. **12**, 2825 (2011).
[38] O. Kramer, "K-nearest neighbors," in *Dimensionality Reduction with Unsupervised Nearest Neighbors* (Springer Berlin Heidelberg, 2013) pp. 13–23.
[39] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*, 1st ed. (Wiley, 2010).
[40] V. N. Vapnik, *The Nature of Statistical Learning Theory* (Springer New York, 2000).
[41] G. Bonaccorso, *Machine Learning Algorithms: A Reference Guide to Popular Algorithms for Data Science and Machine Learning* (Packt Publishing, Birmingham UK, 2017).
[42] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (MIT Press, Cambridge, MA, 2006).
[43] J. Mercer and A. R. Forsyth, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character **209**, 415 (1909).
[44] I. A. Almosallam, M. J. Jarvis, and S. J. Roberts, Monthly Notices of the Royal Astronomical Society **462**, 726 (2016).
[45] K. J. Duncan, M. J. I. Brown, W. L. Williams, P. N. Best, V. Buat, D. Burgarella, M. J. Jarvis, K. Małek, S. J. Oliver, H. J. A. Röttgering, and D. J. B. Smith, Monthly Notices of the Royal Astronomical Society **473**, 2655 (2017).
[46] Z. Gomes, M. J. Jarvis, I. A. Almosallam, and S. J. Roberts, Monthly Notices of the Royal Astronomical Society **475**, 331 (2017).

[47] I. A. Almosallam, Machine Learning Research Group (MLRG) (2017).
[48] P. Mehta, M. Bukov, C. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, Physics Reports **810**, 1 (2019).
[49] M. A. Nielsen, *Neural Networks and Deep Learning* (Determination Press, 2018).
[50] B. Karlik and A. Vehbi Olgac, International Journal of Artificial Intelligence And Expert Systems **1**, 111 (2011).
[51] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, PLOS ONE **13**, 1 (2018).
[52] D. Osthus, S. A. Vander Wiel, N. M. Hoffman, and F. J. Wysocki, SIAM/ASA J. Uncertainty Quantification **7**, 604 (2019).
[53] K. M. Woo, R. Betti, D. Shvarts, A. Bose, D. Patel, R. Yan, P. Chang, O. M. Mannion, R. Epstein, J. A. Delettrez, M. Charissis, K. S. Anderson, P. B. Radha, A. Shvydky, I. V. Igumenshchev, V. Gopalaswamy, A. R. Christopherson, J. Sanz, and H. Aluie, Physics of Plasmas **25**, 052704 (2018).
[54] D. S. Clark, C. R. Weber, J. L. Milovich, A. E. Pak, D. T. Casey, B. A. Hammel, D. D. Ho, O. S. Jones, J. M. Koning, A. L. Kritcher, M. M. Marinak, L. P. Masse, D. H. Munro, M. V. Patel, P. K. Patel, H. F. Robey, C. R. Schroeder, S. M. Sepke, and M. J. Edwards, Physics of Plasmas **26**, 050601 (2019).
[55] D. S. Clark, A. L. Kritcher, J. L. Milovich, J. D. Salmonson, C. R. Weber, S. W. Haan, B. A. Hammel, D. E. Hinkel, M. M. Marinak, M. V. Patel, and S. M. Sepke, Plasma Physics and Controlled Fusion **59**, 055006 (2017).
[56] A. B. Zylstra, S. MacLaren, S. A. Yi, J. Kline, D. Callahan, O. Hurricane, B. Bachmann, G. Kyrala, L. Masse, P. Patel, J. E. Ralph, J. Salmonson, P. Volegov, and C. Wilde, Physics of Plasmas **26**, 052707 (2019).