



**HAL**  
open science

## Selecting Machine Learning Models to Support the Design of Al/CuO Nanothermites

Yasser Sami, Nicolas Richard, David Gauchard, Alain Estève, Carole Rossi

► **To cite this version:**

Yasser Sami, Nicolas Richard, David Gauchard, Alain Estève, Carole Rossi. Selecting Machine Learning Models to Support the Design of Al/CuO Nanothermites. *Journal of Physical Chemistry A*, 2022, 126 (7), pp.1245-1254. 10.1021/acs.jpca.1c09520 . hal-03574270

**HAL Id: hal-03574270**

**<https://laas.hal.science/hal-03574270>**

Submitted on 15 Feb 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Selecting Machine Learning Models to Support the Design of Al/CuO Nanothermites

Yasser Sami<sup>1</sup>, Nicolas Richard<sup>2</sup>, David Gauchard<sup>1</sup>, Alain Estève<sup>1</sup>, Carole Rossi<sup>\*1</sup>

<sup>1</sup> LAAS-CNRS, University of Toulouse, 7 Avenue du colonel Roche, 31400 Toulouse, France

<sup>2</sup> CEA-DAM, DIF, 91297 Arpajon, France

\*e-mail : [rossi@laas.fr](mailto:rossi@laas.fr)

**Keywords:** Machine Learning, Nanoenergetics, Nanothermite, Simulation optimization, Surrogate models, Al/CuO, Multilayer Perceptron

## Abstract

Novel properties associated with nanothermites have attracted great interest for several applications, including lead-free primers and igniters. However, the prediction of quantitative structure-energetic performance relationships are still challenging. This study investigates machine learning methods as tools to surrogate complex physical models to design novel nanothermites with optimized burning rates chosen for energetic performance. The study focuses on Al/CuO nanolaminates, for which nine supervised regressors commonly used in ML applied to materials science are investigated. For each, a ML model is built using a database containing a set of 2,700 Al/CuO nanolaminate systems, specifically generated for this study. We demonstrate the superiority of the multilayer perceptron algorithm to surrogate conventional physical-based models and predict the Al/CuO nanolaminate microstructure-burn rate relationship with good efficiency: the burn rate is estimated with less than 1% error (0.07 m.s<sup>-1</sup>), which is very good for designing nanoengineered energetic materials, knowing that it typically varies from approximately 8-20 m.s<sup>-1</sup>. In addition, the optimization of the Al/CuO nanolaminate structure for burn rate maximization through machine learning takes a few

milliseconds, against the several days to achieve this task using the physical model, and months experimentally.

## 1. Introduction

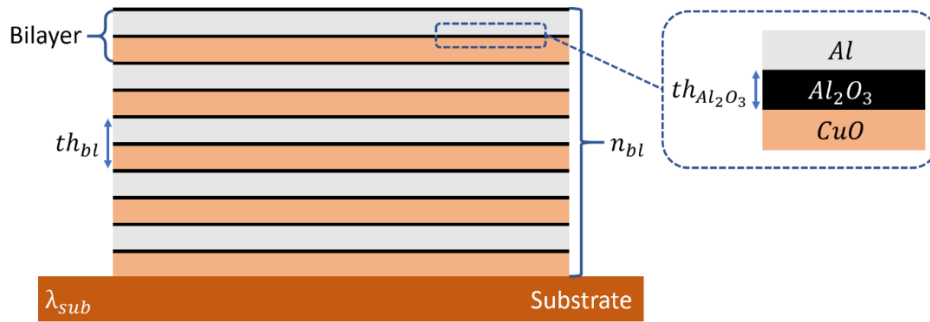
Energetic materials, which include propellants, explosives, and pyrotechnics, are controllable chemical energy sources widely used by industries for civil and military applications. Through the 20<sup>th</sup> century, when developing a new energetic compound, an optimal tradeoff in energy content, safety, and cost was sought. This was achieved through the formulation of high-energy metastable molecules, combining known chemical compounds and/or synthesis parameters. Since the 1990s, progress in nanomanufacturing<sup>1,2</sup> has enabled a new class of energetic materials commonly referred to as nanoenergetic materials, or simply nanoenergetics, including nanothermites, composed of nanosized metals (Al, Mg, Ti) and metal oxides (CuO, Fe<sub>2</sub>O<sub>3</sub>, Bi<sub>2</sub>O<sub>3</sub>, MoO<sub>3</sub>, etc.). Highly complex reactant geometries are now feasible (nanofolios, nanowires, core-shell nanomaterials, and 3D porous nanostructures), and local variations of the microstructure can be achieved through specific technological processing variations<sup>3,4</sup>, offering the perspective to finely tune the energetic performances by tailoring the microstructure. A high amount of experimental data<sup>3-11</sup> was created in research labs, but the consistency, quality, and reproducibility of the energetic properties of these nanomaterials and consequently the collection of relevant datasets still represents an open challenge inhibiting the full development of military and industrial applications. In addition, the development of physical models able to predict the combustion of nanothermites materials requires the implementation of always more complex physics based on computation fluid dynamics scheme computing more than 7 chemical reactions between a dozen of species interacting in both condensed and gas phase. This means that the cost of calculations is extremely high and cannot be used for systematic design purpose. Therefore, it is crucial to develop alternative methods for accelerating the deployment of these novel nanoenergetic materials in applications. This study seeks to

demonstrate that supervised machine learning (ML) approaches, which have been widely adopted in the fields of biology<sup>12</sup>, chemistry<sup>13</sup>, and materials science<sup>14-17</sup>, can surrogate physical models to guide the design of novel nanostructures with optimized properties. The sole study considering machine learning in the field of energetic materials science was published recently by Kang et al.<sup>18</sup>. The authors combined ML methods with thermochemical data from existing ICSD and PubChem databases to screen new candidates for CHNO energetic materials. They succeeded in training a satisfactory surrogate ML model to design new stable molecular CHNO energetic materials with high enthalpy values. In the present study, we apply an alternative strategy to use supervised ML to specific nanothermite materials, i.e., Al-based thermites, for which no database does exist. We focus on Al/CuO nanolaminates alternating Al and CuO nanolayers deposited on a substrate<sup>19</sup>, as they represent one interesting nanoengineered energetic system with direct and immediate applications for tunable initiation of secondary energetic materials<sup>20-24</sup>, joining, brazing, and sealing<sup>25-28</sup>. In addition, Al/CuO nanolaminates have been widely studied both experimentally and theoretically<sup>29-40</sup> so that sufficient data and accurate physical models do exist to generate the necessary database to conduct this study. With the database generated, we build, train, and evaluate several standard supervised ML algorithms, namely, linear regression, support vector regression, kernel ridge regression, Gaussian process regression, decision tree, AdaBoost, random forest, K-nearest neighbors, and multilayer perceptron, for the optimization of Al/CuO nanolaminate energetic performance. To characterize the energetic performance, the burn rate is used as the target property. We demonstrate the efficiency of neural networks and, to a lesser extent, kernel-based methods to surrogate conventional physical-based models to predict the Al/CuO nanolaminate microstructure-burn rate relationship with good efficiency: the burn rate is estimated with less than 1% accuracy ( $0.7 \text{ m}\cdot\text{s}^{-1}$ ), which is very good for designing nanoengineered energetic materials knowing that it typically varies from approximately  $8\text{-}20 \text{ m}\cdot\text{s}^{-1}$ .

## 1. Materials and Methods

### 2.1. Energetic system description

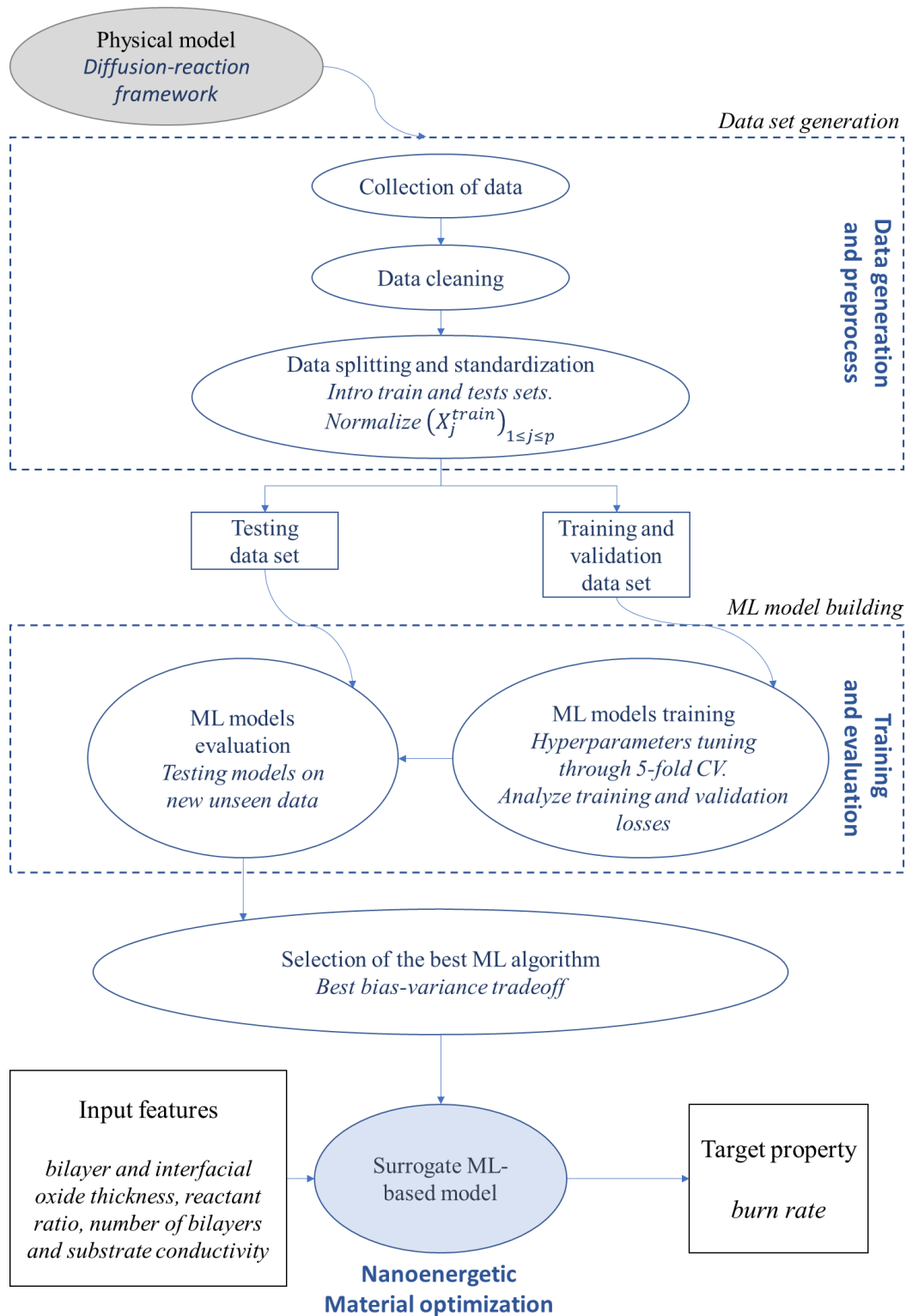
The Al/CuO energetic system (**Figure 1**) is characterized by 5 input features (structural parameters): the bilayer (Al/CuO) thickness, denoted as  $th_{bl}$ , ranging from 25 to 1000 nm; the interfacial oxide thickness,  $th_{Al_2O_3}$ , ranging from 2 to 10 nm; the Al/CuO reactant ratio,  $\varphi$ , ranging from 0.5 to 4; the number of bilayers, noted  $n_{bl}$ , ranging from 5 to 30 bilayers; and the substrate conductivity,  $\lambda_{sub}$ , ranging from  $0.01 \text{ W.K}^{-1}.\text{m}^{-1}$  (equivalent to organic substrate such as Kapton <sup>41</sup>) to  $10 \text{ W.K}^{-1}.\text{m}^{-1}$  (equivalent to ceramic wafer). The ranges of the nanolaminate structural parameters are chosen to fit the technological possibilities and by experience in the field: for example,  $th_{bl} = 25 \text{ nm}$  is the thinnest bilayer that can be deposited by magnetron sputtering, and, it is known that no reaction occurs for  $\varphi < 0.5$  or  $> 4$  or when  $th_{Al_2O_3} > 10 \text{ nm}$ . The main physical properties, i.e., the thermal conductivity, density and heat capacity of Al and CuO are set to  $237 \text{ W.K}^{-1}.\text{m}^{-1}$ ,  $2698 \text{ kg.m}^{-3}$ ,  $897 \text{ J.K}^{-1}.\text{kg}^{-1}$  and  $3.09 \text{ W.K}^{-1}.\text{m}^{-1}$ ,  $6313 \text{ kg.m}^{-3}$ ,  $532 \text{ J.K}^{-1}.\text{kg}^{-1}$ , respectively. <sup>39,40</sup>



**Figure 1.** Schematic of the geometrical features of an Al/CuO energetic system composed of an Al/CuO nanolaminate deposited on a substrate

### 2.2 Machine learning methods

As shown in **Figure 2**, the construction of our ML surrogate model followed two main steps, which are further detailed. The first step is database generation using a physical model: it consists of collecting the data and then cleaning, splitting, and standardizing them to be manipulated by ML algorithms. The second step is building the ML models considering several regression algorithms that include several substeps: training, validation, and evaluation.



**Figure 2.** Flowchart showing the different steps used to develop the surrogate ML-based model.

### 2.3 Data generation, cleaning and preprocessing

To generate the database, a set of 2,700 Al/CuO nanolaminate systems was defined to determine the corresponding burn rate,  $v_{prop}$ , systematically calculated using an in-house thermochemical solver specifically developed for the combustion of sputter-deposited Al/CuO multilayers<sup>42,43</sup>. The average computational time for the physical model to calculate the burn rate for a single Al/CuO nanolaminate is between 10-20 minutes; the cumulative time to generate the database took 20 days (run on an Intel® Xeon® Gold 5218 CPU @ 2.30 GHz). However, as this step is easily parallelizable, a massive time saving occurs shrinking CPU time to few hours: around 8 hours on 64 CPUs.

**Table 1** summarizes all features, their ranges of variation, and, importantly, their linear correlations to the target variable,  $v_{prop}$ , representing their impact on the burn rate. As expected, the bilayer and interface thicknesses, stoichiometry, and substrate conductivity exhibit the highest correlations with the burn rate, two orders of magnitude higher than the number of bilayers. The high correlation between the burn rate and substrate conductivity indicates that using a substrate of high thermal conductivity ( $\lambda_{sub} > 5 \text{ W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$ ) can lead to flame quenching, meaning that  $v_{prop}$  will suddenly drop to 0.

**Table 1.** Feature description classified by order of importance according to their correlation to the target property,  $v_{prop}$ .

| Features [ <i>unity</i> ]                   | Values                             | Correlation (feature, $v_{prop}$ )<br>44 |
|---|------------------------------------|--|
| $\lambda_{sub} [W\cdot m^{-1}\cdot K^{-1}]$ | {0.01, 0.1, 1, 5, 10}              | $-6.74 \times 10^{-1}$                   |
| $th_{bl} [nm]$                              | {25, 50, 100, 200, 300, ..., 1000} | $-3.38 \times 10^{-1}$                   |
| $th_{Al_2O_3} [nm]$                         | {2, 5, 10}                         | $-1.85 \times 10^{-1}$                   |
| $\varphi$                                   | {0.5, 1, 2, 3, 4}                  | $+1.17 \times 10^{-1}$                   |
| $n_{bl}$                                    | {5, 15, 30}                        | $+9.30 \times 10^{-3}$                   |

#### 2.4. Data cleaning, splitting and preprocessing



Once data are collected, the next step in the machine learning workflow is the encoding of the data into a set of specific feature variables  $X_{i,p}$  and a target variable  $Y_p$ , which are manipulated by the ML algorithms. The input features are denoted as  $X_{i,p}$ , where  $i$  corresponds to a given virtual nanolaminate system and  $p$  is one of the five input parameters: ( $p=1$ ) substrate conductivity, ( $p=2$ ) bilayer thickness, ( $p=3$ ) oxide interface thickness, and ( $p=4$ ) stoichiometric ratio. Al/CuO mass ratio, and ( $p=5$ ) number of bilayers. The target property to be optimized,  $Y_p$ , is the burn rate ( $v_{prop}$ ). Then, a step of data cleaning is proceeded: data with aberrant target values and unavailable data, i.e., with missing target values are withdrawn from the database. Typically, they correspond to limit cases where calculations have difficulty converging. It typically concerns Al-rich systems, with  $\varphi > 3$ , which leads to poor convergence and eventually quenching of the flame propagation, as observed experimentally<sup>38</sup>. The cleaned dataset (2,700) is then blocked under different groups, which, in turn, are randomly split into training and test sets. The purpose of using a test set is to provide an unbiased evaluation of the model fit to the training set so that the different models can be compared *in fine*. The blocking strategy of clean data follows a phylogenetic structure scheme where samples with the same  $th_{bl}$ ,  $th_{Al_2O_3}$ , and  $\varphi$  values are put into the same group. This allows us to ensure the independence of the test set from the training set, as explained in<sup>45</sup>. Finally, once the cleaning and splitting processes are operated, the training set is standardized, i.e., data are centered and normalized (removing the mean and scaling to unit variance each feature vector  $X_p = (X_{i,p})_{1 \leq i \leq n}$ ).

## 2.5. ML models training and cross validation

In this study, we consider 9 commonly used machine learning regression methods in materials science, namely: linear regression, support vector regression (SVR), kernel ridge regression (KRR), Gaussian process regression (GPR), decision tree, AdaBoost, random forest, K-nearest

neighbors (KNN), and multilayer perceptron (MLP). The details of the algorithms, as well as their advantages and disadvantages, are well debated and known from the literature; therefore, we only give a brief description of their main characteristics in **Table 2**.

Hyperparameter tuning<sup>46-50</sup> using 5-fold cross-validation (CV)<sup>51-53</sup> on the training dataset is performed for each algorithm. Once hyperparameters are determined, a final model is trained on the whole training set and later tested on the test dataset. CV relies on keeping some data, called validation data, hidden from the model during the training process. Therefore, the training set is split into 5 equally sized sets with respect to the blocking strategy used for splitting. Then, the algorithm is trained with all but one of these 5 subsets, which is used for validation. Permutation of the subsets allows multiple scoring with the training and validation stages. The purpose of using a validation set is to provide an unbiased evaluation of the model's fit on the training set used during hyperparameter tuning. Thus, group splitting is important to prevent, as for training-testing split, from nonindependence of the training and validation sets. The metric used for this hyperparameter tuning step is the mean absolute error (MAE) expressed by **Eq. (1)**, where  $Y_i$  and  $\tilde{Y}_i$  are true and predicted target values, respectively. Finally, we used a *randomizedsearchcv* scheme to optimize hyperparameters, which is particularly suitable for small datasets<sup>54,55</sup> (Details are given in **Table S2** and **Scheme S2**).

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \tilde{Y}_i| \quad (1)$$

**Table 2.** Description of the main ML algorithms used in this work with their main characteristics, advantages, and drawbacks.

| Algorithms                         | Main characteristics                | Advantages  | Drawbacks                                      |
|------------------------------------|-------------------------------------|---|--|
| Linear Regression                  | “Naïve” e.g., simple model          | Simple equation describing the target property    | Cannot describe nonlinear behaviors            |
| <b><i>Kernel-based methods</i></b> |                                     |   |  |
| Support Vector Regression (SVR)    | Nonlinear kernel-based methods with | Nonlinear equation describing the target property | Depends on kernel choice and parameter tuning. |

|                                   |  |  |   |
|-----------------------------------|--|--|---|
|                                   | epsilon-insensitive loss   |  | Optimization is not closed-form   |
| Kernel Ridge Regression (KRR)     | Very similar to SVR but with fitting in closed-form thanks to ridge loss | Computationally more efficient than SVR in some cases  | Computes predictions slower than SVR<br>Hyperparameter optimization scales    |
| Gaussian Process Regression (GPR) | Kernel-based Bayesian approach   | Equivalent to KRR but without kernel's hyperparameters tuning. Computes the target property distribution for each data | Long computation time due to variance generation of predictive distribution   |
| <b><i>Ensemble methods</i></b>    |  |  |   |
| Decision Tree                     | Ensemble approach  | Few parameters to tune.<br>Explicit ensembles  | Ensembles interpretability questionable.<br>Overfitting, sensible to outliers |
| AdaBoost                          | Ensemble boosting approach   | Fast to learn, no parameter tuning.<br>Simple classifier due to feature selection                                      | Sensible to outliers  |
| Random Forest                     | Relevant ensemble approach   | Very accurate and more robust than classic decision tree (no overfitting/no sensibility to outliers)                   | Longer training time  |
| <b><i>Neural Networks</i></b>     |  |  |   |
| Multilayer Perceptron (MLP)       | Neural network approach  | Robust feature engineering process   | Can easily build overfitted model<br>Many and complex parameters to tune      |

## 2.6. Evaluation of ML Models

The accuracy and performance of the different algorithms are characterized using the MAE metric to compute two major losses: the training loss and the test loss. The higher the loss is, the lower the accuracy is. Importantly, these losses enable us to quantify both bias and variance errors of the model to find the best bias-variance tradeoff. Bias characterizes the error induced by false hypotheses in the learning algorithm; thus, high bias may lead to neglecting important relations between features and target variables and cause underfitting. Variance characterizes the error caused by sensitivity to small fluctuations on the training data, where high variance can cause the algorithm to fit to the inherent noise present in data: this process is called overfitting. Three other metrics were also adopted to quantify the relevance of the learning

algorithms: the  $R^2$  score and the training and inference computational times. The  $R^2$  score, expressed by **Eq. (2)**, normalizes the mean square error by the target variance and thus offers a good criterion to assess the model accuracy when typical error tolerance is unknown.

$$R^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \tilde{Y}_i)^2}{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \quad (2)$$

where,  $Y_i$ ,  $\tilde{Y}_i$  and  $\hat{Y}_i$  are true, predicted and average target values

## 2. Results

### 3.1. Comparison of the different ML methods

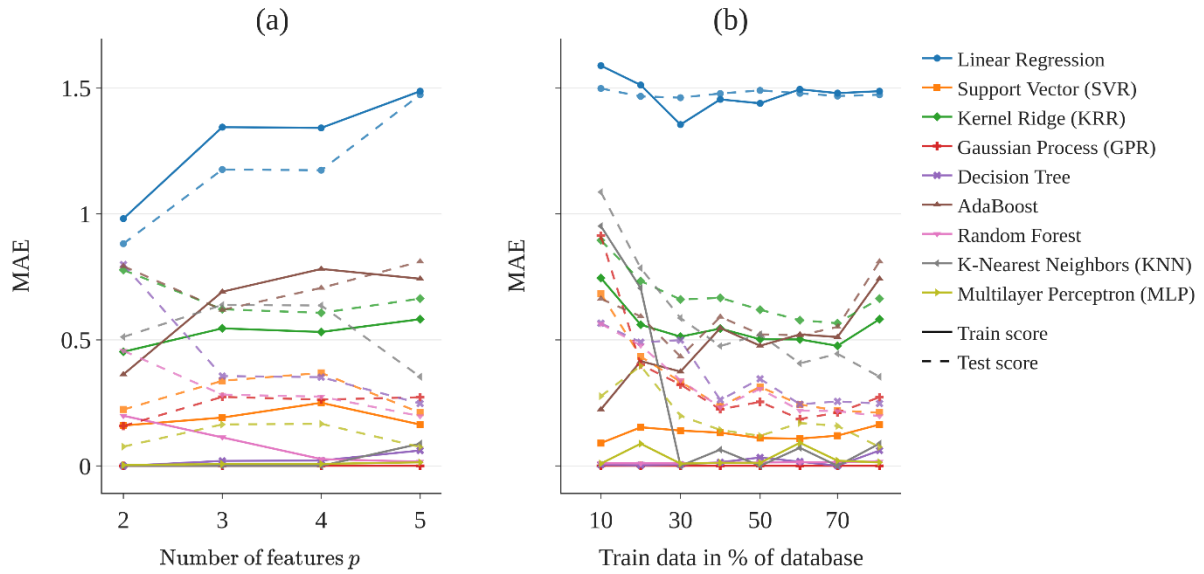
This first section is dedicated to evaluating the learning efficiency and accuracy of the different ML regression methods presented in **Table 2**. **Figure 3** shows how the number of input features (**Figure 3a**) and the size of the training dataset used to generate the database (**Figure 3b**) affect learning in terms of the MAE loss. The size of the training dataset is given as a percentage of the total dataset (2,700 simulations). The test set size remains constant at 20% of the total dataset. Note that each training process here includes the reoptimization (tuning) of the hyperparameters as described in Section 2.4. To plot **Figure 3a**, the structural features are added incrementally, one by one, from the most correlated to the less correlated one, as determined in **Table 1**, i.e.,  $\lambda_{sub}$ ,  $th_{bl}$ ,  $th_{Al_2O_3}$ ,  $\varphi$ ,  $n_{bl}$ . At each step, a data split process is executed with a ratio of 80% training set and 20% test set samples, corresponding to a specific number of simulations, as reported in **Table 3**.

The regression methods achieve different levels of accuracy, and **Figure 3a** shows two different behaviors; one for 2-feature datasets and one for beyond 3-feature datasets. For a 2-feature dataset, all supervised ML models except SVR and MLP have relatively high test losses than training losses, indicating a very bad bias-variance tradeoff. For example, decision trees and random forests have test losses of  $0.80 \text{ m.s}^{-1}$  and  $0.46 \text{ m.s}^{-1}$ , respectively, versus a training loss of 0. Beyond 3 selected features, except for KNN, the difference between the training and

test losses becomes small ( $< 0.3 \text{ m.s}^{-1}$ ): a good bias-variance tradeoff is thus achieved. SVR and MLP offer the best tradeoffs with a training-testing loss difference smaller than  $0.15 \text{ m.s}^{-1}$  and test losses stable at approximately  $0.25 \text{ m.s}^{-1}$  while adding features. This indicates that SVR and MLP feature a stable accuracy over datasets with an increasing number of features. Therefore, they are more robust than other regression algorithms to fit the nanolaminate burn rate. In contrast, linear regression features the lowest accuracy ( $\text{MAE} \sim 1 \text{ m.s}^{-1}$ ) due to its high bias error. Its simplistic linear model clearly does not offer enough flexibility when confronted with complex featurization, as it is marked by high training and test losses.

**Figure 3b** shows that for training set sizes ranging from 10 to 50% of the total dataset, all algorithms, except linear regression, display a strong learning rate as the training set size increases. Thus, increasing the training set sizes has a positive effect on the model's efficiency in terms of their capability to predict the burn rate target property.

Over large training set sizes, i.e., when at least 50% of the total dataset (1,350 data points) is used for learning, most regression methods, except KNN, present a good bias-variance tradeoff, and the best losses are obtained for SVR, GPR, decision tree, random forest, and MLP. Looking into more details, the best tradeoffs are found for SVR and MLP, where the training losses are  $0.13$  and  $0.03 \text{ m.s}^{-1}$ , and test losses are  $0.24$  and  $0.12 \text{ m.s}^{-1}$ , respectively, which corresponds to the training versus test losses having the smallest difference among all models. Note that all models exhibit average test losses higher or equivalent than SVR and MLP. Then, SVR and MLP not only have the best bias-variance tradeoff but also perform the highest scores, especially MLP. This indicates that using SVR and MLP algorithms ensures accurate predictions of the burn rate target property.



**Figure 3.** Variation of training and test losses for each regression method: (a) with increasing number of features at 80%/20% training-testing split, (b) with increasing size of training set at 20% test set size and in a 5-feature database.

**Table 3.** Number of data considered used for the training as a function of the percentage of the total dataset and the number of considered features  $p$ . Note that the total dataset comprises 2,700 simulation points of the target property,  $v_{prop}$ .

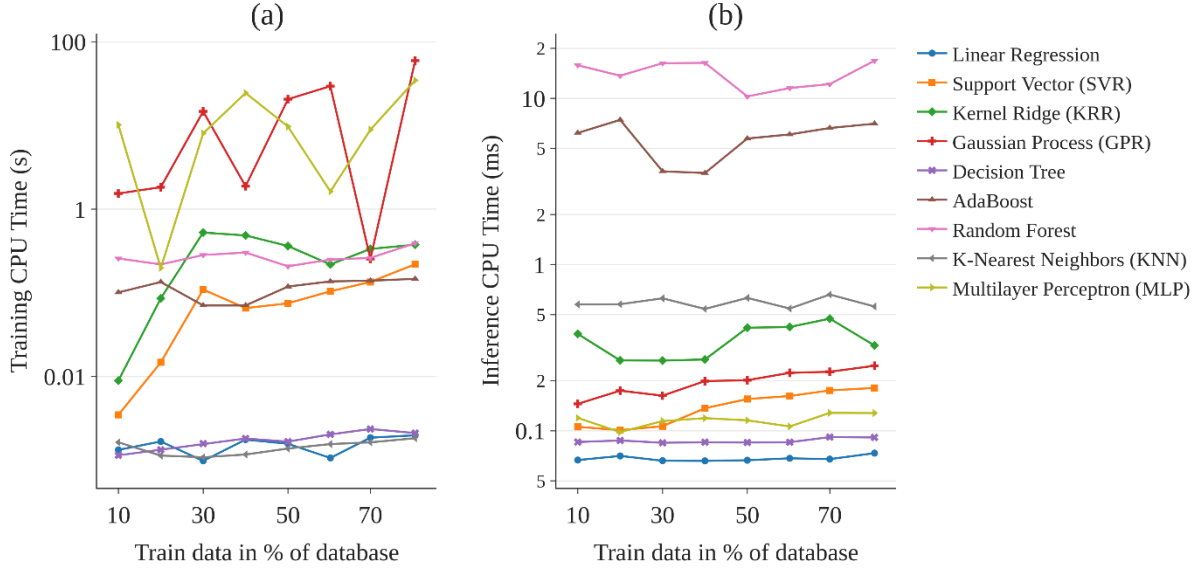
| Number of features $p$ | Set at 20% | Set at 50% | Set at 80% |
|------------------------|------------|------------|------------|
| 2                      | 12         | 30         | 48         |
| 3                      | 36         | 90         | 144        |
| 4                      | 180        | 450        | 720        |
| 5                      | 540        | 1,350      | 2,160      |

To further evaluate the relative efficiency of all supervised ML models, **Figure 4** plots the computational time, in terms of the central processing unit (CPU) time, spent for the learning process (training time, **Figure 4a**) and for the  $v_{prop}$  inference process (inference time, **Figure 4b**) to predict the target property as a function of training set sizes and for each algorithm. In **Figure 4a**, the linear regression, decision tree, AdaBoost, random forest, and KNN algorithms feature a very slow training time rise with the training set size: the variation is  $< 0.01$  s over the full-size range. The training times of the SVR, KRR, GPR, and MLP methods increase more

rapidly with the size of the training set, ranging from 10 to 80%, but still remain reasonable: less than 1 s for SVR and KRR and in the range of a hundred seconds for GPR and MLP algorithms.

In **Figure 4b**, as expected, the inference times remain very short and negligible compared to the training times for all algorithms: from 0.06 to 15 ms for training set sizes ranging from 10 to 80%. Because their evaluation functions contain as many terms as the number of training data, the SVR, GPR, and KRR algorithms are more affected by the increasing number of training data. Additionally, random forest has the highest inference time (13 ms), as it has a complexity almost equal to that of a decision tree multiplied by  $p \times n_{trees}$ , where  $p$  is the number of features and  $n_{trees}$  is the number of built trees (hundreds of estimators). This is, in fact, applicable if we compare random forest inference time (13 ms) to decision tree inference time (0.08 ms).

However, regardless of the selected ML algorithm, the inference times are all very small compared to the computational time when using the physical model, which required ~10–20 minutes to carry out a single simulation. This clearly demonstrates reason for the interest in substituting the physical-based simulator with a ML approach, supposing that the target property can be predicted with enough accuracy. In our case, enough accuracy means the prediction of the burn rate with a MAE test loss  $< 0.5 \text{ m}\cdot\text{s}^{-1}$ , which is achieved here for several ML methods, whose best are MLP and, to a lesser extent, SVR.

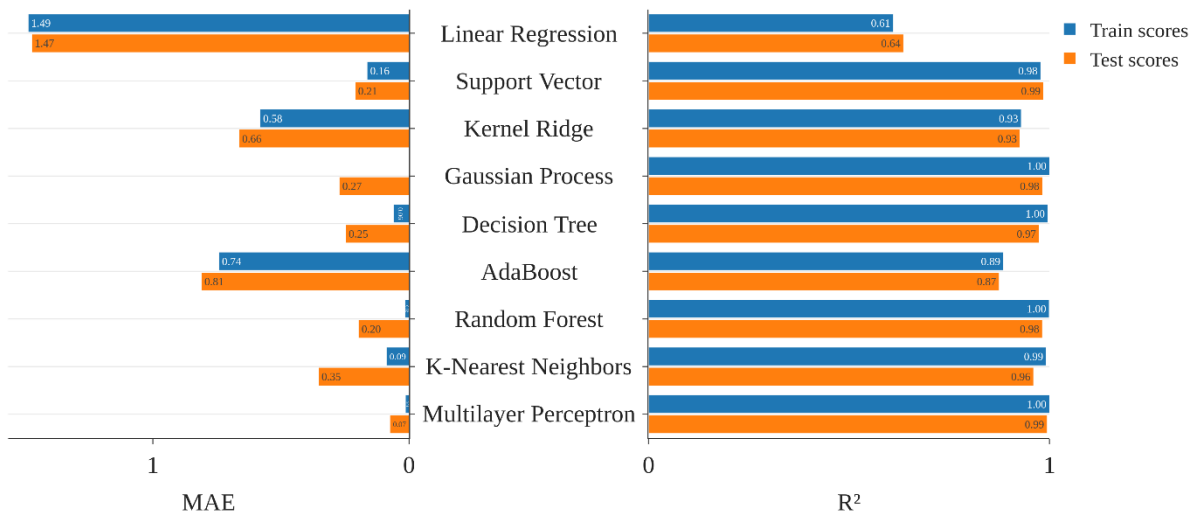


**Figure 4.** Variation of the CPU times required for the learning process (a) and  $v_{prop}$  inference process (b) with the size of the training set. All simulations were performed on an Intel® Xeon® Gold 5218 CPU @ 2.30 GHz, from which the CPU time was systematically recorded.

The training dataset is a 5-feature database.

As a last experiment, we partition the dataset generated with 5-input features into 80% of the data for the model training phase and 20% for the testing phase. We then quantify the relative accuracy of the learning algorithms. This benchmark is still realized by comparing training and test losses, which are summarized in **Figure 5**, plotting both MAE loss and  $R^2$  score for the 9 selected supervised methods. MLP and SVR give the highest accuracy ( $MAE < 0.2 \text{ m.s}^{-1}$ ) and the highest  $R^2$  scores (0.985 for SVR and 0.994 for MLP). Linear regression and AdaBoost algorithms offer the lowest accuracy ( $MAE \sim 1.4 \text{ m.s}^{-1}$  and  $0.87 \text{ m.s}^{-1}$ , respectively) and worst  $R^2$  score (0.636 and 0.875, respectively), which means that compared to burn rate variance, the mean square error is too high.





**Figure 5.** MAE losses and  $R^2$  scores of each regression method using a 5-feature dataset with 80% training set size.

To summarize, a minimum of 1,350 data points must be used for the training step, i.e., 50% of the dataset, considering a 5-feature database, to obtain sufficient accuracy in predicting the target property,  $v_{prop}$ . Under these training conditions, MLP outperforms other algorithms, as it demonstrates the best accuracy (MAE  $\sim 0.07$  m.s $^{-1}$ ). However, its learning step is time-consuming ( $\sim 50$ – $100$  s). Consequently, SVR offers a good compromise thanks to its short training and inference times: learning is performed 2 to 100 times faster, and inferring times are 2 times faster (see **Figure 4**) than MLP while offering a correct accuracy (MAE  $\sim 0.21$  m.s $^{-1}$ ).

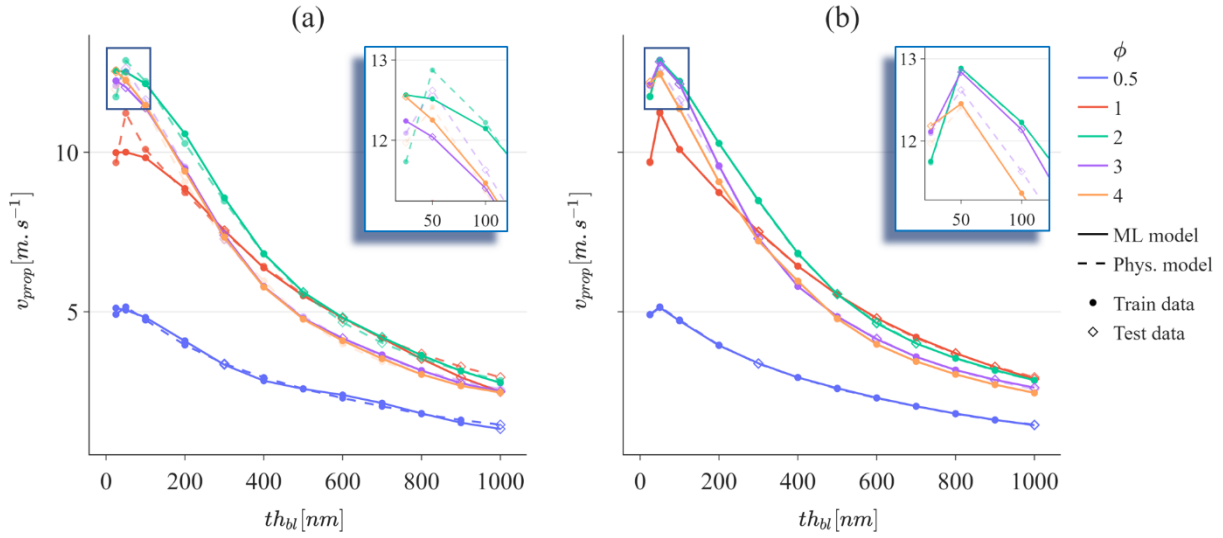
The GPR method offers good performances with MAE test losses of  $0.25$  m.s $^{-1}$  predicting the target properties but is the most resource-consuming: it requires 400 Mo of memory for a training set size of 80% (2,160 simulations), which is 40 times more than the resources consumed by SVR or MLP models. Additionally, it has a very time-consuming learning phase ( $\sim 100$  s), which escalates quickly for big datasets, reaching 1 h for 10k data points. Decision tree and random forest also offer good performances with MAE test losses equivalent to SVR,

i.e., 0.25 and 0.20 m.s<sup>-1</sup>, respectively. However, these algorithms are less stable (worse bias-variance tradeoff) and thus less robust over the different training sets compared with SVR or MLP algorithms (see **Figure 3a**).

We, therefore, choose both SVR and MLP algorithms to build two different surrogate ML models to estimate the burn rate of Al/CuO nanolaminates, varying the Al/CuO bilayer thickness ( $th_{bl}$ ), for different Al/CuO reactant ratios, which are the most interesting parameters for engineering applications.

### 3.2. Evaluation of the accuracy of the surrogate ML models

The number of bilayers, the interfacial oxide thickness, the substrate thickness, and conductivity are set to:  $n_{bl} = 15$ ,  $th_{Al_2O_3} = 2$  nm,  $th_{sub} = 25$   $\mu$ m, and  $\lambda_{sub} = 0.1$  W.K<sup>-1</sup>.m<sup>-1</sup>. The developed ML models were trained using a 5-feature dataset with an 80% training set size; the MAE for  $v_{prop}$  prediction was equal to 0.21 m.s<sup>-1</sup> for SVR and 0.07 m.s<sup>-1</sup> for MLP (R<sup>2</sup> scores were 0.985 and 0.994, respectively). Estimations of  $v_{prop}$  by the surrogate models as a function of  $th_{bl}$ , are plotted in **Figure 6** (solid lines). To assess the accuracy of the surrogate model,  $v_{prop}$  is also calculated from the physical model (dashed lines). As regression is conducted in a high dimension space, only two specific dimensions are visualized: bilayer thickness and stoichiometry. They were chosen for their relevancy regarding nanolaminate engineering.



**Figure 6.** Burn rate ( $v_{prop}$ ) as a function of bilayer thickness and for five Al/CuO

stoichiometric ratios. Solid lines show predictions of the burn rate using the surrogate SVR (a) and MLP (b) ML based models, while dashed lines show estimations using the physical-based model. Filled circles and empty diamonds correspond to train data and test data, respectively.

SVR and MLP surrogate ML algorithms are trained using a 5-feature dataset with an 80% training set size.

The effect of bilayer thickness on the burn rate shows the same global behavior as has been previously reported in several experimental works<sup>31,42</sup>. From **Figure 6**, we clearly distinguish the first regime for thick bilayers ( $th_{bl} > 150$  nm) with a burn rate decreasing slowly below 2  $m.s^{-1}$  with the thickening of  $th_{bl}$ . Decreasing  $th_{bl}$  down to the nanoscale, the burn rate increases continuously up to a maximum depending on the stoichiometry. Below this maximum, the physical model generally provides a bilayer thickness value of 25 nm, showing an abrupt burn rate drop. For the ultrathin bilayer, i.e., below 25 nm, no reaction leading to a steady-state burn rate is engaged. This results from the lack of sufficient chemical energy with respect to the total heat capacity of the films, in which the interfacial layer (alumina) is no longer negligible compared to the amount of reactive (Al/CuO) material.

This sudden change in velocity behavior causes some difficulties for the SVR surrogate model due to the lack of data points in the ultrathin bilayer range, i.e., below the maximum  $v_{prop}$ . Despite this fact, the SVR surrogate model is largely satisfactory in the range of technological and applicative interests ( $th_{bl}$  ranges from 100–1000 nm) and succeeds in catching the inflection of the burn rate for  $th_{bl} < 50$  nm (see insert at the top right corner in **Figure 6a**) but not precisely: it cannot predict the saddle point and sharp decrease in burn rate. In contrast, the MLP surrogate model not only predicts the burn rate with better accuracy than SVR through the  $th_{bl}$  range of 50–1000 nm and over the different stoichiometric ratios but also succeeds in catching the nonlinear behavior for  $th_{bl} < 50$  nm with a very satisfying accuracy; see the insert at the top right corner in **Figure 6b**. This illustrates the excellent accuracy of MLP algorithms, as discussed in Section 3.1.

In summary, both SVR and MLP surrogate models provide a reasonable estimations of the burn rate ( $v_{prop}$ ) with a significant ( $\times 10^6$ ) speed up in the CPU time without considering data generation process. For a more complete comparison we include data generation CPU time, which is around 8 hours considering parallelization. Moreover, this step is done only once: if 10 optimizations need a total CPU time of 7 days using physical model, then using a surrogate model shortens global process to 8 hours, including data generation (optimization using ML surrogate model takes only few seconds, negligible compared to hours). We observe that MLP largely supplants SVR in predicting combustion in tedious regimes where nonlinearity dominates.

### 3.3. Optimizing a nanolaminate using ML surrogate models

In this last section, the SVR and MLP surrogate models are used to maximize the burn rate value, varying and automatically optimizing the Al/CuO nanolaminate feature parameters. The following technological constraints were also imposed: the substrate should be a 50  $\mu\text{m}$  thick glass wafer ( $th_{sub} = 50 \mu\text{m}$ ,  $\lambda_{sub} = 1.4 \text{ W} \cdot \text{K}^{-1} \cdot \text{m}^{-1}$ ) and the bilayer thickness should

not be thinner than 25 nm. The optimization tool used to conduct this study is a standard BGFS-type algorithm, namely, L-BGFS-B<sup>56</sup>. Maximizing  $v_{prop}$  using the physical model, with a convergence criterion equal to  $10^{-5}$  m.s<sup>-1</sup>, takes 402 iterations over a total duration of 16 hours. For a calculated burn rate of 12.61 m.s<sup>-1</sup>, one optimized parameter combination output is  $th_{bl} = 37$  nm,  $th_{Al_2O_3} = 2$  nm,  $\varphi = 3.08$ , and  $n_{bl} = 15$ . The optimization of the nanolaminate to reach the fastest  $v_{prop}$  using the ML surrogate models (details in **Figure S3**) also requires a similar number of iterations as in the case of the physical model (330 iterations for SVR and 468 for MLP) but is significantly faster: both methods achieve this task within approximately 0.1 s, representing a speedup of  $\times 10^6$ ; burn rates of 13.02 and 12.77 m.s<sup>-1</sup> are obtained for SVR and MLP, respectively. The optimized parameter combination output is also very close to the physical model results, especially for MLP (see **Table 4** for a summary). As expected, optimal structures in both cases, using physical or surrogate models, are obtained when the interfacial oxide thickness  $th_{Al_2O_3}$  is minimized at 2 nm and when the bilayer thickness  $th_{bl}$  and Al/CuO reactant ratio  $\varphi$  are at optimal values. Minimizing  $th_{Al_2O_3}$  shortens the diffusion distance of oxygen atoms, thus improving the reactivity of the nanolaminates, but the Al/CuO reactant ratio  $\varphi$  also has a great effect on the burn rate. Indeed, the combustion of Al/CuO nanolaminates is purely governed by a thermal conduction process (**Supporting Information S1**); thus, the burn rate is directly affected by the material thermal diffusivity and heat of reaction. The thermal diffusivities of an Al/CuO nanolaminate are estimated to be ranked as  $\varphi 3$  ( $2.4 \times 10^{-5}$  m<sup>2</sup>/s) >  $\varphi 2$  ( $2 \times 10^{-5}$  m<sup>2</sup>/s) >  $\varphi 1$  ( $1.5 \times 10^{-5}$  m<sup>2</sup>/s), as the thermal diffusivity of aluminum ( $\sim 1 \times 10^{-4}$  m<sup>2</sup>/s) is more than 10 times higher than that of CuO ( $\sim 1 \times 10^{-5}$  m<sup>2</sup>/s)<sup>38</sup>. Increasing the thermal diffusivity enhances flame propagation. However, increasing  $\varphi$  also lowers the heat of reaction, which has a direct negative effect on the flame temperature, and thus on the burn rate. The heat of reactions is estimated to be ranked as  $\varphi 1$  (3.9 kJ.kg<sup>-1</sup>) >  $\varphi 2$  (3.3 kJ.kg<sup>-1</sup>) >  $\varphi 3$  (2.7 kJ.kg<sup>-1</sup>). This

simple reasoning explains why a compromise exists, which was found for  $\varphi = 2.82$  with the MLP surrogate model and 3.08 with the physical model.

Finally, comparing statistical and more physical approaches, it can be argued that data generation for learning purposes is a time-consuming process, such as the 20 days to build the whole database in our case. Nevertheless, this duration is only a few days more than a basic automated optimization using the physical model, while the database is generated only once. This expensive CPU time is therefore largely compensated by the very short inference time (order of ms) of the downstream surrogate model. Note also that the procedure to generate the database is massively parallelizable, as all calculations are independent. Furthermore, the physical model used in this study can be considered computationally efficient, as the physics inside is restricted to basic diffusion/reaction mechanisms. The use of more complex methods involving mass transport and exchanges through fluid dynamics considerations will require much more computational resources, giving important perspectives for ML methods to accompany the conventional physical approaches in the T-CAD toolboxes.

**Table 4.** Comparison of optimization results using the physical model vs. ML surrogate model. Data Generation CPU time is considered in case of parallelization on 64 CPUs.

|                                      |                  | Physical model | SVR surrogate<br>model | MLP surrogate<br>model |
|--------------------------------------|------------------|----------------|------------------------|------------------------|
| Algorithm                            |                  |                | L-BGFS-B               |                        |
| Number of iterations                 |                  | 402            | 330                    | 468                    |
| CPU time                             | Data generation  | -              | 8 hours                | 8 hours                |
|                                      | Optimization     | 16 hours       | 0.1 s                  | 0.1 s                  |
|                                      | 10 optimizations | 7 days         | 1 s                    | 1 s                    |
| Best $v_{prop}$ [ $m \cdot s^{-1}$ ] |                  | 12.61          | 13.02                  | 12.77                  |
| $th_{bl}$ [ $nm$ ]                   |                  | 37             | 25                     | 40                     |

|                     |      |      |      |
|---------------------|------|------|------|
| $th_{Al_2O_3}$ [nm] | 2    | 2    | 2    |
| $\varphi$           | 3.08 | 1.79 | 2.82 |
| $n_{bl}$            | 15   | 9    | 13   |

#### 4. Conclusion

This paper investigates different surrogate machine learning algorithms for the optimization of Al/CuO nanolaminate energetic performance in terms of the burn rate. Nine regressors commonly used in machine learning and commonly applied to materials science were investigated: Linear Regression, Support Vector Regression, Kernel Ridge Regression, Gaussian Process Regression, Decision Tree, AdaBoost, Random Forest, K-Nearest Neighbors, Multilayer Perceptron. For each, an ML model is built using at least 1,250 Al/CuO nanolaminate systems generated using an in-house physical model, in the absence of sufficient and reliable experimental data. Their accuracy and performance in terms of CPU time for learning and latency were compared and analyzed as a function of the size of train data set and number of features. Kernel, ensemble-based learning methods, as well as neural networks, perform much better than linear regression or KNN, which does not suit the nonlinear behavior of the combustion process. Then, we demonstrated the efficiency of neural networks to surrogate conventional physical based models to predict Al/CuO nanolaminate microstructure-burn rate relationship: the burn rate is estimated with less than 1% accuracy ( $0.07 \text{ m}\cdot\text{s}^{-1}$ ) which is very good for designing purpose of nanoengineered energetic materials knowing that it varies typically between 8 - 20  $\text{m}\cdot\text{s}^{-1}$ . Results also show that optimization of the nanolaminate, in term of the maximization of the burn rate, can be achieved within 0.1 s, against 16 hours using the physical model. Note, that the inference time remains always short (few milliseconds). Thus, for more CPU time consuming physical models (> 1 hour per simulation), our approach offers an even better time saving. This study clearly demonstrates that the knowledge gap in

nanoenergetic materials can effectively be addressed by combining ML approaches and current physical models, to generate dataset and then extract, at low computation cost, quantitative relationships between the nanostructure and combustion performances. A long-term goal of this study is to enable surrogating more sophisticated physical models, with higher complexity (more input parameters) and longer computational time. To do so, potential enhancing of data sampling algorithm is necessary and will help saving computation time on data generation step. For now, it has to be noted that using the trained ML models, especially MLP, offers a very high accuracy. Compared to linear regression or interpolation (KNN) it clearly enables surrogating physical model (for optimization purpose) with a very high precision which is not possible otherwise.

#### **Author contributions**

Y. S. performed all the physical simulations, generated the data base, built the ML models and carried out the scientific study. D. G. assisted Y. S. in solving computational issues and in optimizing the computational resources. C. R. and A. E. supervised this research. And N. R., A. E. and C. R. provided support for manuscript preparation.

#### **Funding sources**

C.R. received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 832889 - PyroSafe).

#### **Competing financial interests**

The authors declare no competing financial interests.

#### **Supporting Information**

In parallel of this article we present a description of the physical model theory. We also introduce ranges used for hyperparameters tuning of each ML model. Lastly, plots of optimization trials using physical model, SVR and MLP are presented to demonstrate the similarity of optimization trials and result thanks to their (surrogate models) accuracy.



## **Acknowledgments**

The authors thank Emilian Tichtchenko from LAAS-CNRS for his scientific support for physical model understanding. The authors also thank Salim Aboudou from Qube-RT (Paris) and Mohamed Kerroumi from Saint-Gobain (Paris) for their constructive discussions for ML models building steps. The authors acknowledge support the European Research Council (H2020 Excellent Science) Researcher Award (grant 832889 –PyroSafe).

## **References**

- (1) Chryssolouris, G.; Stavropoulos, P.; Tsoukantas, G.; Salonitis, K.; Stournaras, A. Nanomanufacturing Processes: A Critical Review. *IJMPT* **2004**, *21* (4), 331. <https://doi.org/10.1504/IJMPT.2004.004946>.
- (2) Imboden, M.; Bishop, D. Top-down Nanomanufacturing. *Physics Today* **2014**, *67* (12), 45–50. <https://doi.org/10.1063/PT.3.2621>.
- (3) van der Heijden, A. E. D. M. Developments and Challenges in the Manufacturing, Characterization and Scale-up of Energetic Nanomaterials – A Review. *Chemical Engineering Journal* **2018**, *350*, 939–948. <https://doi.org/10.1016/j.cej.2018.06.051>.
- (4) Hastings, D. L.; Dreizin, E. L. Reactive Structural Materials: Preparation and Characterization. *Advanced Engineering Materials* **2018**, *20* (3), 1700631. <https://doi.org/10.1002/adem.201700631>.
- (5) Kabra, S.; Gharde, S.; Gore, P. M.; Jain, S.; Khire, V. H.; Kandasubramanian, B. Recent Trends in Nanothermites: Fabrication, Characteristics and Applications. *Nano Ex.* **2020**, *1* (3), 032001. <https://doi.org/10.1088/2632-959X/abbce7>.
- (6) He, W.; Liu, P.-J.; He, G.-Q.; Gozin, M.; Yan, Q.-L. Highly Reactive Metastable Intermixed Composites (MICs): Preparation and Characterization. *Advanced Materials* **2018**, *30* (41), 1706293. <https://doi.org/10.1002/adma.201706293>.
- (7) Xu, F.; Biswas, P.; Nava, G.; Schwan, J.; Kline, D. J.; Rehwoldt, M. C.; Mangolini, L.; Zachariah, M. R. Tuning the Reactivity and Energy Release Rate of I2O5 Based Ternary Thermite Systems. *Combustion and Flame* **2021**, *228*, 210–217. <https://doi.org/10.1016/j.combustflame.2020.12.047>.
- (8) Xiao, L.; Zhao, L.; Ke, X.; Zhang, T.; Hao, G.; Hu, Y.; Zhang, G.; Guo, H.; Jiang, W. Energetic Metastable Al/CuO/PVDF/RDX Microspheres with Enhanced Combustion Performance. *Chemical Engineering Science* **2021**, *231*, 116302. <https://doi.org/10.1016/j.ces.2020.116302>.
- (9) Chen, Y.; Ren, W.; Zheng, Z.; Wu, G.; Hu, B.; Chen, J.; Wang, J.; Yu, C.; Ma, K.; Zhou, X.; Zhang, W. Reactivity Adjustment from the Contact Extent between CuO and Al Phases in Nanothermites. *Chemical Engineering Journal* **2020**, *402*, 126288. <https://doi.org/10.1016/j.cej.2020.126288>.
- (10) Muravyev, N. V.; Monogarov, K. A.; Schaller, U.; Fomenkov, I. V.; Pivkina, A. N. Progress in Additive Manufacturing of Energetic Materials: Creating the Reactive Microstructures with High Potential of Applications. *Propellants, Explosives, Pyrotechnics* **2019**, *44* (8), 941–969. <https://doi.org/10.1002/prep.201900060>.
- (11) Rossi, C.; Estève, A.; Vashishta, P. Nanoscale Energetic Materials. *Journal of Physics and Chemistry of Solids* **2010**, *2* (71), 57–58. <https://doi.org/10.1016/j.jpics.2009.10.015>.
- (12) Cheng, J.; Tegge, A. N.; Baldi, P. Machine Learning Methods for Protein Structure Prediction. *IEEE Reviews in Biomedical Engineering* **2008**, *1*, 41–49. <https://doi.org/10.1109/RBME.2008.2008239>.
- (13) Perkins, R.; Fang, H.; Tong, W.; Welsh, W. J. Quantitative Structure-Activity Relationship Methods: Perspectives on Drug Discovery and Toxicology. *Environmental Toxicology and Chemistry* **2003**, *22* (8), 1666–1679. <https://doi.org/10.1897/01-171>.
- (14) Schmidt, J.; Marques, M. R. G.; Botti, S.; Marques, M. A. L. Recent Advances and Applications of Machine Learning in Solid-State Materials Science. *npj Comput Mater* **2019**, *5* (1), 1–36. <https://doi.org/10.1038/s41524-019-0221-0>.
- (15) Rajan, K. Materials Informatics: The Materials “Gene” and Big Data. *Annual Review of Materials Research* **2015**, *45* (1), 153–169. <https://doi.org/10.1146/annurev-matsci-070214-021132>.
- (16) Correa-Baena, J.-P.; Hippalgaonkar, K.; van Duren, J.; Jaffer, S.; Chandrasekhar, V. R.; Stevanovic, V.; Wadia, C.; Guha, S.; Buonassisi, T. Accelerating Materials

- Development via Automation, Machine Learning, and High-Performance Computing. *Joule* **2018**, 2 (8), 1410–1420. <https://doi.org/10.1016/j.joule.2018.05.009>.
- (17) Liu, Y.; Zhao, T.; Ju, W.; Shi, S. Materials Discovery and Design Using Machine Learning. *Journal of Materiomics* **2017**, 3 (3), 159–177. <https://doi.org/10.1016/j.jmat.2017.08.002>.
- (18) Kang, P.; Liu, Z.; Abou-Rachid, H.; Guo, H. Machine-Learning Assisted Screening of Energetic Materials. *J. Phys. Chem. A* **2020**, 124 (26), 5341–5351. <https://doi.org/10.1021/acs.jpca.0c02647>.
- (19) Salvagnac, L.; Assie-Souleille, S.; Rossi, C. Layered Al/CuO Thin Films for Tunable Ignition and Actuators. *Nanomaterials* **2020**, 10 (10), 2009. <https://doi.org/10.3390/nano10102009>.
- (20) Taton, G.; Lagrange, D.; Conedera, V.; Renaud, L.; Rossi, C. Micro-Chip Initiator Realized by Integrating Al/CuO Multilayer Nanothermite on Polymeric Membrane. *J. Micromech. Microeng.* **2013**, 23 (10), 105009. <https://doi.org/10.1088/0960-1317/23/10/105009>.
- (21) Zhang, Q.; Zhu, P.; Wang, K.; Xu, C.; Yang, Z.; Shen, R.; Zheng, G. An Exploding Foil Overpressure Actuator for Multipoint Synchronous Initiation and Mach Reflection Generation. *Sensors and Actuators A: Physical* **2020**, 314, 112248. <https://doi.org/10.1016/j.sna.2020.112248>.
- (22) Glavier, L.; Nicollet, A.; Jouot, F.; Martin, B.; Barberon, J.; Renaud, L.; Rossi, C. Nanothermite/RDX-Based Miniature Device for Impact Ignition of High Explosives. *Propellants, Explosives, Pyrotechnics* **2017**, 42 (3), 308–317. <https://doi.org/10.1002/prop.201600154>.
- (23) Xu, J.; Tai, Y.; Shen, Y.; Dai, J.; Xu, W.; Ye, Y.; Shen, R.; Hu, Y. Characteristics of Energetic Semiconductor Bridge Initiator Based on Different Stoichiometric Ratios of Al/MoO<sub>3</sub> Reactive Multilayer Films under Capacitor Discharge Conditions. *Sensors and Actuators A: Physical* **2019**, 296, 241–248. <https://doi.org/10.1016/j.sna.2019.07.015>.
- (24) Zhu, P.; Hou, G.; Wang, H.; Xu, C.; Zhao, S.; Shen, R. Design, Preparation, and Performance of a Planar Ignitor Inserted With PyroMEMS Safe and Arm Device. *Journal of Microelectromechanical Systems* **2018**, 27 (6), 1186–1192. <https://doi.org/10.1109/JMEMS.2018.2875707>.
- (25) Nicollet, A.; Salvagnac, L.; Baijot, V.; Estève, A.; Rossi, C. Fast Circuit Breaker Based on Integration of Al/CuO Nanothermites. *Sensors and Actuators A: Physical* **2018**, 273, 249–255. <https://doi.org/10.1016/j.sna.2018.02.044>.
- (26) Guo, W.; Chang, S.; Cao, J.; Wu, L.; Shen, R.; Ye, Y. Precisely Controlled Reactive Multilayer Films with Excellent Energy Release Property for Laser-Induced Ignition. *Nanoscale Research Letters* **2019**, 14 (1), 301. <https://doi.org/10.1186/s11671-019-3124-6>.
- (27) Kinsey, A. H.; Slusarski, K.; Sosa, S.; Weihs, T. P. Gas Suppression via Copper Interlayers in Magnetron Sputtered Al–Cu<sub>2</sub>O Multilayers. *ACS Appl. Mater. Interfaces* **2017**, 9 (26), 22026–22036. <https://doi.org/10.1021/acsami.7b03071>.
- (28) Ahn, J. Y.; Kim, S. B.; Kim, J. H.; Jang, N. S.; Kim, D. H.; Lee, H. W.; Kim, J. M.; Kim, S. H. A Micro-Chip Initiator with Controlled Combustion Reactivity Realized by Integrating Al/CuO Nanothermite Composites on a Microhotplate Platform. *J. Micromech. Microeng.* **2015**, 26 (1), 015002. <https://doi.org/10.1088/0960-1317/26/1/015002>.
- (29) Yarrington, C. D.; Abere, M. J.; Adams, D. P.; Hobbs, M. L. Reactive Nanolaminar Pulsed-Laser Ignition Mechanism: Modeling and Experimental Evidence of Diffusion

- Limited Reactions. *Journal of Applied Physics* **2017**, *121* (13), 134301. <https://doi.org/10.1063/1.4979578>.
- (30) Adams, D. P. Reactive Multilayers Fabricated by Vapor Deposition: A Critical Review. *Thin Solid Films* **2015**, *576*, 98–128. <https://doi.org/10.1016/j.tsf.2014.09.042>.
- (31) Bahrami, M.; Taton, G.; Conédéra, V.; Salvagnac, L.; Tenailleau, C.; Alphonse, P.; Rossi, C. Magnetron Sputtered Al-CuO Nanolaminates: Effect of Stoichiometry and Layers Thickness on Energy Release and Burning Rate. *Propellants, Explosives, Pyrotechnics* **2014**, *39* (3), 365–373. <https://doi.org/10.1002/prop.201300080>.
- (32) Mily, E. J.; Oni, A.; LeBeau, J. M.; Liu, Y.; Brown-Shaklee, H. J.; Ihlefeld, J. F.; Maria, J.-P. The Role of Terminal Oxide Structure and Properties in Nanothermite Reactions. *Thin Solid Films* **2014**, *562*, 405–410. <https://doi.org/10.1016/j.tsf.2014.05.005>.
- (33) Egan, G. C.; Mily, E. J.; Maria, J.-P.; Zachariah, M. R. Probing the Reaction Dynamics of Thermite Nanolaminates. *J. Phys. Chem. C* **2015**, *119* (35), 20401–20408. <https://doi.org/10.1021/acs.jpcc.5b04117>.
- (34) Marín, L.; Nanayakkara, C. E.; Veyan, J.-F.; Warot-Fonrose, B.; Joulie, S.; Estève, A.; Tenailleau, C.; Chabal, Y. J.; Rossi, C. Enhancing the Reactivity of Al/CuO Nanolaminates by Cu Incorporation at the Interfaces. *ACS Appl. Mater. Interfaces* **2015**, *7* (22), 11713–11718. <https://doi.org/10.1021/acsami.5b02653>.
- (35) Lanthony, C.; Guiltat, M.; Ducéré, J. M.; Verdier, A.; Hémercyck, A.; Djafari-Rouhani, M.; Rossi, C.; Chabal, Y. J.; Estève, A. Elementary Surface Chemistry during CuO/Al Nanolaminate-Thermite Synthesis: Copper and Oxygen Deposition on Aluminum (111) Surfaces. *ACS Appl. Mater. Interfaces* **2014**, *6* (17), 15086–15097. <https://doi.org/10.1021/am503126k>.
- (36) Abdallah, I.; Zapata, J.; Lahiner, G.; Warot-Fonrose, B.; Cure, J.; Chabal, Y.; Esteve, A.; Rossi, C. Structure and Chemical Characterization at the Atomic Level of Reactions in Al/CuO Multilayers. *ACS Appl. Energy Mater.* **2018**, *1* (4), 1762–1770. <https://doi.org/10.1021/acsaem.8b00296>.
- (37) Zapata, J.; Nicollet, A.; Julien, B.; Lahiner, G.; Esteve, A.; Rossi, C. Self-Propagating Combustion of Sputter-Deposited Al/CuO Nanolaminates. *Combustion and Flame* **2019**, *205*, 389–396. <https://doi.org/10.1016/j.combustflame.2019.04.031>.
- (38) Wang, H.; Julien, B.; Kline, D. J.; Alibay, Z.; Rehwoldt, M. C.; Rossi, C.; Zachariah, M. R. Probing the Reaction Zone of Nanolaminates at  $\sim\mu\text{s}$  Time and  $\sim\mu\text{m}$  Spatial Resolution. *J. Phys. Chem. C* **2020**, *124* (25), 13679–13687. <https://doi.org/10.1021/acs.jpcc.0c01647>.
- (39) Tichtchenko, E.; Estève, A.; Rossi, C. Modeling the Self-Propagation Reaction in Heterogeneous and Dense Media: Application to Al/CuO Thermite. *Combustion and Flame* **2021**, *228*, 173–183. <https://doi.org/10.1016/j.combustflame.2021.01.040>.
- (40) Rossi, C. Engineering of Al/CuO Reactive Multilayer Thin Films for Tunable Initiation and Actuation. *Prop., Explos., Pyrotech.* **2019**, *44* (1), 94–108. <https://doi.org/10.1002/prop.201800045>.
- (41) Nicollet, A.; Lahiner, G.; Belisario, A.; Souleille, S.; Djafari-Rouhani, M.; Estève, A.; Rossi, C. Investigation of Al/CuO Multilayered Thermite Ignition. *Journal of Applied Physics* **2017**, *121* (3), 034503. <https://doi.org/10.1063/1.4974288>.
- (42) Lahiner, G.; Nicollet, A.; Zapata, J.; Marín, L.; Richard, N.; Rouhani, M. D.; Rossi, C.; Estève, A. A Diffusion–Reaction Scheme for Modeling Ignition and Self-Propagating Reactions in Al/CuO Multilayered Thin Films. *Journal of Applied Physics* **2017**, *122* (15), 155105. <https://doi.org/10.1063/1.5000312>.
- (43) Lahiner, G.; Zapata, J.; Cure, J.; Richard, N.; Djafari-Rouhani, M.; Estève, A.; Rossi, C. A Redox Reaction Model for Self-Heating and Aging Prediction of Al/CuO

- Multilayers. *Combustion Theory and Modelling* **2019**, 23 (4), 700–715. <https://doi.org/10.1080/13647830.2019.1584336>.
- (44) Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson Correlation Coefficient. In *Noise Reduction in Speech Processing*; Springer Topics in Signal Processing; Springer Berlin Heidelberg: Berlin, Heidelberg, 2009; Vol. 2, pp 1–4. [https://doi.org/10.1007/978-3-642-00296-0\\_5](https://doi.org/10.1007/978-3-642-00296-0_5).
- (45) Roberts, D. R.; Bahn, V.; Ciuti, S.; Boyce, M. S.; Elith, J.; Guillera-Arroita, G.; Hauenstein, S.; Lahoz-Monfort, J. J.; Schröder, B.; Thuiller, W.; et al.; Cross-Validation Strategies for Data with Temporal, Spatial, Hierarchical, or Phylogenetic Structure. *Ecography* **2017**, 40 (8), 913–929. <https://doi.org/10.1111/ecog.02881>.
- (46) Hutter, F.; Lücke, J.; Schmidt-Thieme, L. Beyond Manual Tuning of Hyperparameters. *Künstl Intell* **2015**, 29 (4), 329–337. <https://doi.org/10.1007/s13218-015-0381-0>.
- (47) Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-Generation Hyperparameter Optimization Framework. *arXiv:1907.10902 [cs, stat]* **2019**.
- (48) Duan, K.; Keerthi, S. S.; Poo, A. N. Evaluation of Simple Performance Measures for Tuning SVM Hyperparameters. *Neurocomputing* **2003**, 51, 41–59. [https://doi.org/10.1016/S0925-2312\(02\)00601-X](https://doi.org/10.1016/S0925-2312(02)00601-X).
- (49) Probst, P.; Wright, M. N.; Boulesteix, A. Hyperparameters and Tuning Strategies for Random Forest. *WIREs Data Mining Knowl Discov* **2019**, 9 (3). <https://doi.org/10.1002/widm.1301>.
- (50) Schilling, N.; Wistuba, M.; Drumond, L.; Schmidt-Thieme, L. Hyperparameter Optimization with Factorized Multilayer Perceptrons. In *Machine Learning and Knowledge Discovery in Databases*; Appice, A., Rodrigues, P. P., Santos Costa, V., Gama, J., Jorge, A., Soares, C., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2015; Vol. 9285, pp 87–103. [https://doi.org/10.1007/978-3-319-23525-7\\_6](https://doi.org/10.1007/978-3-319-23525-7_6).
- (51) Berrar, D. Cross-Validation. In *Encyclopedia of Bioinformatics and Computational Biology*; Elsevier, 2019; pp 542–545. <https://doi.org/10.1016/B978-0-12-809633-8.20349-X>.
- (52) Browne, M. W. Cross-Validation Methods. *Journal of Mathematical Psychology* **2000**, 44 (1), 108–132. <https://doi.org/10.1006/jmps.1999.1279>.
- (53) Picard, R. R.; Cook, R. D. Cross-Validation of Regression Models. *Journal of the American Statistical Association* **1984**, 79 (387), 575–583. <https://doi.org/10.1080/01621459.1984.10478083>.
- (54) Bergstra, J.; Bengio, Y. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research* **2012**, 13 (10), 281–305
- (55) Bardenet, R.; Brendel, M.; Kégl, B.; Sebag, M. Collaborative Hyperparameter Tuning. 9. *Proceedings of the 30th International Conference on Machine Learning*, PMLR **2016**, 28(2),199-207.
- (56) Byrd, R. H.; Lu, P.; Nocedal, J.; Zhu, C. A Limited Memory Algorithm for Bound Constrained Optimization, *SIAM Journal on Scientific Computing*, **1995**, 16(5), 1190–1208. <https://doi.org/10.1137/0916069>

For Table of Contents Only

