Undergraduate Final Thesis

Mechanical Engineering

Applying machine learning methods to predict the yield stress in high entropy alloys

June 1, 2019

Author: Pau Cutrina Vilalta

Advisor: Xin-Cindy Wang

Universitat Politècnica de Catalunya
Escola d'Enginyeria Barcelona Est

University of Colorado at Colorado Springs

Abstract

This study examines the efficacy of machine learning (ML) methods on predicting the yield stress in high entropy alloys due to fluctuation in the stacking fault energy. This work is motivated by the high computation cost of the physics based models such as dislocation dynamics simulations and molecular dynamics. We show that our ML model can serve as a surrogate model given sufficient training data.

Contents

1 Introduction **4**

2 Methods **6**

2.1 Data set 6

2.1.1 Input type 1: Statistical descriptors of the SFE fluctuation 7

2.1.2 Input type 2: Estimated statistical descriptors of the SFE fluctuation 7

2.1.3 Input type 3: Pixels 8

2.1.4 Output: Yield stress 8

2.2 Machine learning models 10

2.2.1 Supervised 10

2.2.2 Unsupervised 13

2.3 Metrics 14

2.3.1 Evaluation Metrics 14

2.3.2 Cross-validation 16

2.3.3 Learning curve 17

3 Results **19**

3.1 Data analysis 19

3.2 Input processing analysis 22

3.3 Extrapolation 27

3.3.1 Region size 27

3.3.2 Stacking fault energy 28

3.4 Interpolation 29

3.4.1 Stacking fault energy 30

4 Discussion and concluding remarks **31**

5 Acknowledgment **33**

6 Bibliography **34**

7 Apendix A **41**

8 Additional Information **42**

List of Figures

1 Yield stress increase versus SFE region size as a function of mean SFE . . . 6
2 Statistical attributes . . . 7
3 Estimated statistical attributes . . . 8
4 Image of the stacking energy distribution for a 5.82nm region size . . . 8
5 Cross-validation example [32] . . . 16
6 Variance and bias study in learning curves of Deng et al. [43] . 17
7 Graphical description abut the difference between bias and variance [44] . . . 18
8 Variance analysis [46] . . . 18
9 PCA with three input types scaled . . . 20
10 Unsupervised results . . . 21
11 Yield stress known vs predicted . . . 24
12 Train size analysis . . . 25
13 Learning curves analysis . . . 25
14 Leave one out . . . 26
15 Region size extrapolation . . . 27
16 Stacking faul energy extrapolation . . . 29

1 Introduction

The rapid development in technology has spurred the need for novel materials and alloys with improved mechanical properties [1]. Therefore, there has been active research in finding novel alloy designs [2], [3]. A major challenge in studying complex alloy systems is that the phase diagrams are often not available and the data acquisition (simulation and theoretical) are expensive experimentally and computationally [4].

High entropy alloys (HEA) has shown promise as a new material that has a lot of favorable mechanical and thermal properties. It shows a high yield stress [5], [6] independence between ductility and temperature [7], excellent specific strength, superior mechanical performance at high temperatures, exceptional ductility and fracture toughness at cryogenic temperatures, superparamagnetism, and superconductivity [8]. Also, their high hardness, wear resistance, high-temperature softening resistance and anti-corrosion make the HEAs a perfect candidate for hard-facing [9] as well as many other structural uses in the transportation and energy industries [10] [13]. HEAs are generally classified as an alloys that are composed of five or more alloy elements. The crystal structure of HEAs can be complex with heterogeneous phases [12]. Regardless, they are becoming more popular materials in many applications [11].

A stacking fault is a planar defect in the crystal structure of a closed-packed system [14], [15]. It plays a critical role in the deformation properties of face centered cubic (fcc) metals and alloys. The stacking fault energy also has a strong influence in the formation of partial dislocations, the ability of a dislocation to cross slip, and the formation of twin boundaries [16]. A material with low stacking fault energy is more prone to formation of partial dislocations, cross slip and twin boundaries. [17].

In this report, we focused on a specific type of HEA: Ni-Co-Fe-Cr-Mn. A key parameter that dictates the plastic deformation in materials is the Stacking Fault Energy (SFE) of the material [18]. Traditionally, researchers assume that HEAs have a uniform SFE. However, due to the heterogeneous composition of HEAs, the SFE distribution is not necessarily uniform. The effect of the fluctuation in the SFE on the yield strength of HEAs has been studied in the publication of Sun et al. [19], however, the computational cost to obtain the

yield strength of HEas with fluctuating SFE is high.

We applied machine learning (ML) methods to predict the yield stress of HEAs with varying SFE. The main goal is to learn the relationship between the SFE fluctuations and the yield stress using the results from a physics model, the phase field dislocation model (PFDM).

The inputs to the ML methods are discussed in Section 2.1 and the different ML methods are discussed in 2.2. Then, relevant data properties are reviewed in Section 3.1 using different unsupervised models. In addition, the contribution of each feature in the models prediction is studied. With the aim to find the optimal model structure, the different input types will be deeply analyzed and discussed in the Section 3.2. Different tests are going to be presented such as test size analysis, learning curves and leave one out test.

Finally, two tests will examine the generalization of the surrogate model. The first one uses the extrapolation to predict one entire region size (Section 3.3.1) as well as to predict one entire stacking fault energy (Section 3.3.2). The second one uses the interpolation to predict values inside one stacking fault energy (Section 3.4.1). In the end, the Section 4 discusses the whole project.

Keywords: High entropy alloys, machine learning, dislocation dynamics, yield stress prediction

2 Methods

2.1 Data set

For our study, we used the results of 83 PFDM simulations of the Ni-Co-Fe-Cr-Mn family of high entropy alloys with different distribution of stacking fault energies in the material.

Yifei et al. [19] showed that a heterogeneous distribution of stacking fault energies can result in a difference of around 10 % in the yield stress of HEAs. Figure 1 shows the increase in yield stress compared to HEAs with a homogeneous stacking fault energy throughout the domain. All of the stacking fault energy distributions have the same standard deviation except the calculations with a mean stacking fault energy of 35MPa.

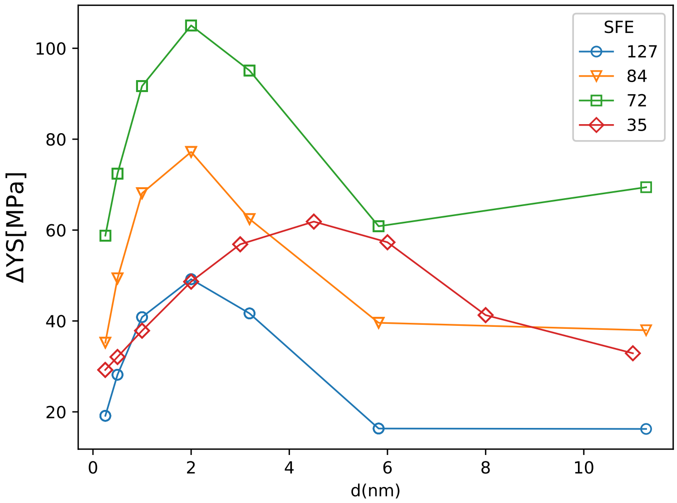


Figure 1: Yield stress increase versus SFE region size as a function of mean SFE

We assessed the efficacy of three different type of inputs to the ML methods in predicting the yield stress.

2.1.1 Input type 1: Statistical descriptors of the SFE fluctuation

We used the following three statistical descriptors to characterize the distribution of the SFE.

- **Mean Stacking fault energy (SFE) in domain**: in all of the PFDM calculations, the SFE is assumed to be uniformly distributed. The mean stacking energy is denoted by gamma (γ). The stacking fault energy of each region is chosen from a uniform distribution with 4 different mean stacking fault energies $\gamma \in [35.0, 72.0, 84.7, 127.1]\frac{\text{mJ}}{\text{m}}$
- **Standard deviation of stacking fault energy distribution**: is denoted by sigma (σ). We used standard deviations $\sigma \in [12, 39]\frac{\text{mJ}}{\text{m}}$.
- **Region size**: The regions of constant stacking fault energy are generated using Voronoi tessellation with mean size between 0.25 nm and 11.0 nm.

Figure 2 is an example of the distribution of the stacking fault energy in a sample material.

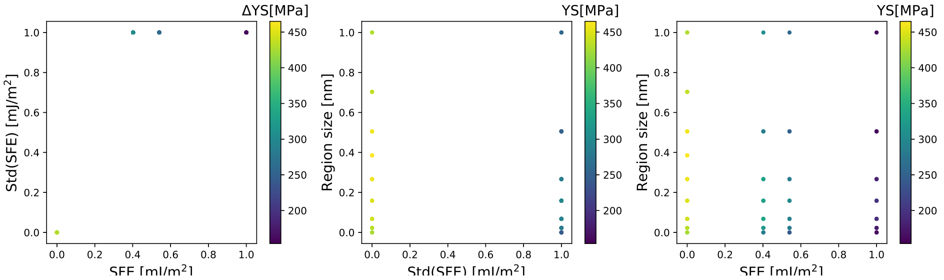


Figure 2: Statistical attributes

2.1.2 Input type 2: Estimated statistical descriptors of the SFE fluctuation

The estimated statistical descriptors are similar to the previous input type, except that the mean and standard deviation of the stacking fault energy are estimated from the pixel values from images such as in 3. Furthermore, the mean region size of same stacking fault energies is replaced by the number of pixels that have equal stacking fault energy up to machine precision.

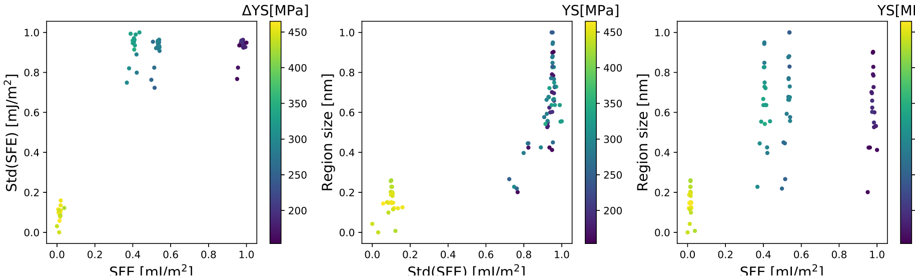


Figure 3: Estimated statistical attributes

2.1.3 Input type 3: Pixels

In this case, the data is a vector of size $256^2 = 65536 \times 1$ for each PFDM simulation. Each pixel represent the value of the stacking fault energy in that point.

Figure 4 illustrates the distribution of the stacking fault energies.

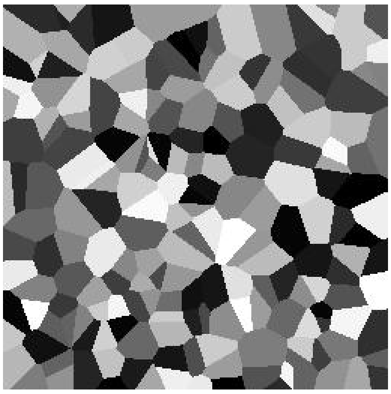


Figure 4: Image of the stacking energy distribution for a 5.82nm region size

2.1.4 Output: Yield stress

The target to predict will be the yield stress. This value goes from 1 460MPa.

In summary, the inputs with statistical descriptors will have only three degrees of freedom. For the input using pixels, the input dimension will be 256x256, which is significantly higher than other two input types. The input and out dimensions are summarized in the Table 1.

Input Type	Samples	Input dimension	Output dimension
Estimated statistical	83	3	1
Statistical	83	3	1
Pixels	83	65536	1

Table 1: Data structure

2.2 Machine learning models

The understanding of the data is fundamental to selecting the correct machine learning model. Their kernels, complexity and size will be sensitive to the data set.

To optimize the hyper-parameters of the models, *GridSearchCV Sklearn's* function has been used. Their iterative process through all the possible configurations, gives the best structure of the model for our data [20].

In the Sections 2.2.1 and 2.2.2 will be discussed. In the Section 7, we included more details on the hyperparameters of these models.

2.2.1 Supervised

The supervised learning is a forward-backward process based on learning the mapping function from the input to labeled output data, thus is called as a supervised process. The goal is to make predictions as close as possible to the labeled outputs.

Here are all the models used in our study. In Section 3, we selected models with the best results.

- **K-Neighbors Regressor:**
  This model uses the similarity of the attributes to predict the value of the tested data. This similarity is computed as distances in the features space through the euclidean distance (Equation 1) where K is the number of neighbors. Finally, the mean of the neighbor's output (Equation 2) gives the predicted value [60].

$$d = \sqrt{\sum_{i=1}^{k} (\hat{x}_i - x_i)^2} \tag{1}$$

$$\hat{y} = \frac{1}{k} \sum_{i=1}^{k} y_i \tag{2}$$

- **Bayesian Ridge Regressor:**
  This probabilistic model is an approach to linear regression with the difference that it estimates a probability distribution rather than a single value. Thus, to obtain the prediction it is necessary to get the Gaussian distribution as is showed in the Equation 3. Where β is the weight matrix and X the predictor matrix and $\beta^T X$ is the mean. θ^2 is the variance and I is the identity matrix [61].

$$\hat{y} \sim N(\beta^T X, \sigma^2 I) \tag{3}$$

- **Decision Tree Regressor:**
  This framework rot the training data as branches where each break-point is a decision node. This decisions are result of probabilistic interactions between the attributes. At the end, a "tree" structure is achieved with one single prediction. Thus, is easy to know the route followed by the predictor avoiding the "black box" problem (unknown of the ML decisions to do predictions). This model is deeply developed by Quinlan et al. [21].

- **Gradient Boosting Regressor:**
  This model learns from the mistakes of the previous predictors reducing the apparition in the future. It is necessary to create a simple model and then increase their complexity depending on the errors of the data too difficult to fit. Each time we create a predictor it is joined in a network. At the end, this model (network) will be evaluated with new data activating the more useful predictors. Thus, it is possible to avoid many iterations reducing significantly the computational cost. For more information see the publications of Yoav et al. [22] and Kearns et al. [23].

- **Support Vector Regressor:**
  The idea of SVR is based on the computation of a linear regression function in a high dimensional feature space where the input data are mapped via a nonlinear function [24]. It tries to fit the error within a certain threshold individualizing the hyper-plane which maximizes the margin. This model has important advantages in high dimensionality space because it does not depend on the dimensionality of the input space [25],[26].

- **Kernel Ridge Regressor:**
  The only difference between this model and the support vector regressor is the loss function used, in this case the squared error loss. It is already known that the efficiency of this model in medium-sized data sets is higher than the SVR with less computational time [27]. More information can be found in the publication of Vladimir et al. [28].

- **Gaussian Process Regressor:**
  It is a model based on stochastic processes who find the functions' distribution using the mean and covariance function. The observed data can identify the posterior distribution over the possible functions. Thus, is possible to know all the functions that match with our data. For more information see the book of Rasmussen et al. [29]. In our case, a radial basis function (RBF) kernel is used, it is obtained using the Equation 4 [37].

$$k(x, y) = exp(-\gamma \left\| x - y \right\|^2) \tag{4}$$

- **Multi-layer Perceptron Regressor:**
  This is a machine learning framework relying on mimicking the learning pattern of natural biological neural networks. This model minimizes the error function between the real values and the predicted ones. This back-propagation process adjusts the weights of the network [31].

- **Ada-Boost Regressor:**
  This model works by repeatedly running a given weak learning machine on different distribution of training data and combining their outputs. At each iteration the distributions of training data depends upon the performance of the machine in the previous iterations [48]. This model overcomes the problem of needing large number of training data and gives a simplified and effective solution to boosting algorithms.

- **Linear discriminant analysis (LDA)**:
  This model is widely used for both feature extraction and dimension reduction. It projects the data onto a lower-dimensional vector space achieving a maximum discrimination [36]. Despite being a supervised learning model, it will be used with the same goal as the unsupervised ones, analyzing the data distribution.

2.2.2 Unsupervised

In contrast with the supervised learning, the unsupervised do not need to label, classify or categorize outputs to train. Otherwise, the model tries to find patters in the data to do predictions.

In this study, the unsupervised learning will be used as data analysis as well as preprocessing. Thereby, these models are not going to do predictions.

- **Principal component analysis (PCA)**:
  This is a multivariate technique that represents the data into a new orthogonal variables called principal components. This components are the axes in which the variability of the data is greater [49].

- **T-distributed stochastic neighbor embedding (tSNE)**:
  This probabilistic approach place the data in a low-dimensional space in a way that preserves neighbor identities. The neighbors distance and population density will classify the samples [33]. The perplexity will define the number of neighbors to take into account for each point.

- **Multidimensional scaling (MDS)**:
  This technique is a dimensional reduction, usually Euclidean, of the original space where each point represents one different sample. The distance between those points correspond to the original dissimilarities [34].

- **Independent Component Analysis (ICA)**:
  This method searches the linear transformation that minimizes the statistical dependence between its components. It is similar to the PCA technique with the difference that, in this case, can be imposed an independence more than second order [35].

2.3 Metrics

2.3.1 Evaluation Metrics

- **Coefficient of determination (R^2)**:
  Also called multiple correlation coefficient, the R^2 is a measure of success of predicting the dependent variable front the independent variables [30]. In this case, this metric will be computed as the square of the correlation between predicted values and the actual values through the Equation 5.

$$R^2(y,\hat{y}) = 1 - \left(\frac{\sum_{i=0}^{n_{samples}-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{samples}-1}(y_i - \overline{y})^2}\right) \tag{5}$$

  where $\overline{y}$ is:

$$\overline{y} = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} y_i \tag{6}$$

  Other evaluation techniques have value range between 0 and 1, in this case, it can be negative. This can arise when the model is actually worse than just fitting a horizontal line. [37]

- **Mean absolute percentage error (MAPE)**:
  This regression quality measure is widely used as a loss function in regression problems of machine learning, it is popular for its very intuitive interpretation in terms of relative error. [38]

$$MAPE = \frac{100}{n} \sum_{n}^{t=1} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{7}$$

- **Mean Absolute Error (MAE)**:
  This metric is a good indicator of average model performance. It gives the same weight to all errors [39] therefore the MAE is a more natural

measure of average error compared to RMSE [40].

$$MAE(y,\hat{y}) = \frac{1}{n_{samples}} \sum_{i=o}^{n_{samples}-1} |y_i - \hat{y}_i| \tag{8}$$

- **Root Mean Square Error (RMSE)**:
  It is appropriate to represent model performance when the error distribution is expected to be Gaussian. It penalizes variance as it gives errors with larger absolute values more weight than errors with smaller absolute values [39].

$$RMSE(y,\hat{y}) = \sqrt{\frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2} \tag{9}$$

2.3.2 Cross-validation

The cross-validation is a method to evaluate how well the proposed model will generalize to an independent data set. Therefore, it is based in split different chunks to check the independence of the training samples to assure the model's accuracy. The number of chunks are specified with the parameter *K*. If the model is not stable, an increasing of variance or bias will be obtained [41]. Both cases are harmful for the prediction's accuracy. In this particular study, it is going to be used to evaluate machine learning models on a limited data sample.

Because of the cross-validation split chunks (folds) and the main parameter is the *K*, the procedure is called k-fold cross-validation. Nevertheless, there is a specific type where the model is fitted with *K*-1 observations and predict the remaining observation left out named leave-one-out [42]. Both methods are going to be used in the results, though, the tables presented in the report shows the mean of the five k-fold cross-validation (*K*=5) results. The goal is to give objective results taking different test outcomes, not only the best one. In the Figure 5 is possible to see graphically one example with 10 folds using cross-validation.

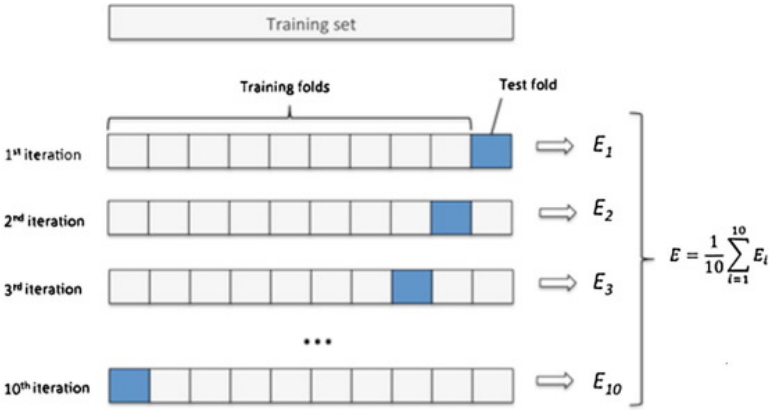


Figure 5: Cross-validation example [32]

2.3.3 Learning curve

The learning curve is a representation of the predictions' accuracy depending on the number of training samples. Their goal is to visualize the evolution of the bias and variance. Thus, may be possible to choose the correct strategy to reduce it. The main goal of supervised learning is to create a model that describes the relationship between the target and the given attribute. For each training set a different model will be learned. The amount of the model variation for each training set is called variance. Another quantity related to the learning curve is called the bias. The bias is a measure of the complexity of the ML model. The less complex models will have a higher bias.

In the Figure 6 is possible to see that the optimal model can be found in a tradeoff position between the model bias and the model variance. Out of that, as complex as the model is, the bias tend to decrease but at the same time the variance increase. Thereby, two situations arise out of the optimal point. The first, called under-fitting, arises when despite have a high prediction error and bias, the variance of the model is still small. On the other hand, high variance and low bias is achieved and is called over-fitted. To understand graphically the consequences of this performance see the Figure 6, 7.

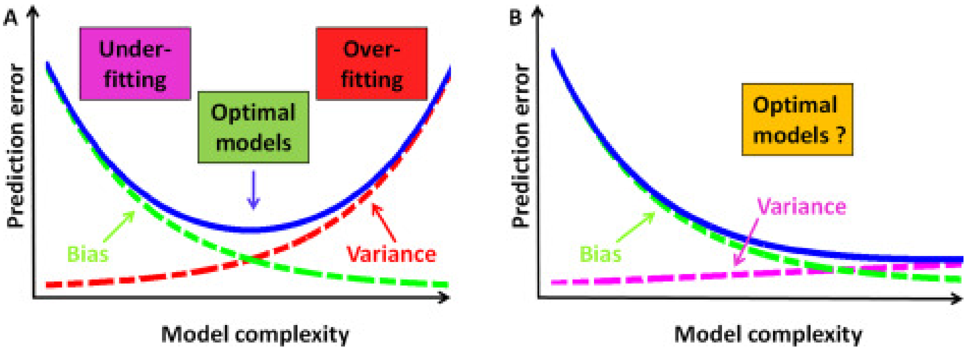


Figure 6: Variance and bias study in learning curves of Deng et al. [43]

In our study, due to the small data set available, the trend will increase the complexity to capture all the information given by the samples. Consequently, it is possible that the model describes noise instead to build relationship between the observations and the predictions [45]. Also, the model can learn very well the training samples but have a poor performance to do predictions. This usually comes up when the model just memorized the observed data.

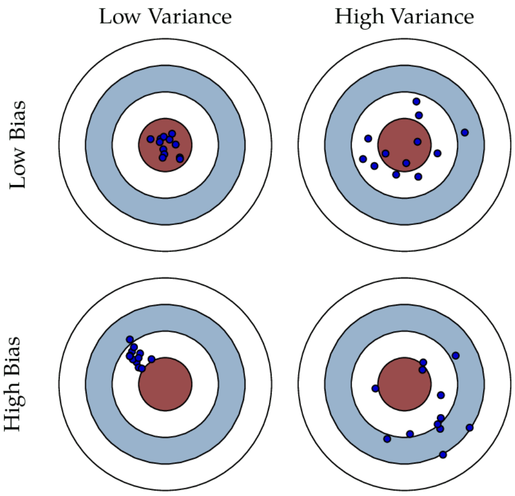


Figure 7: Graphical description abut the difference between bias and variance [44]

This problem is commonly known as "over-fitting".

In the Figure 8, the gap between the training error and the prediction error tells us where we are at the bias-variance trade-off. A small gap indicates more variance and high training error high bias.

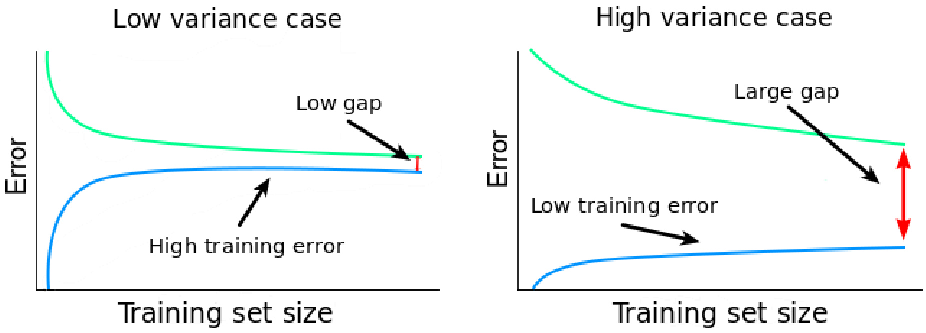


Figure 8: Variance analysis [46]

3 Results

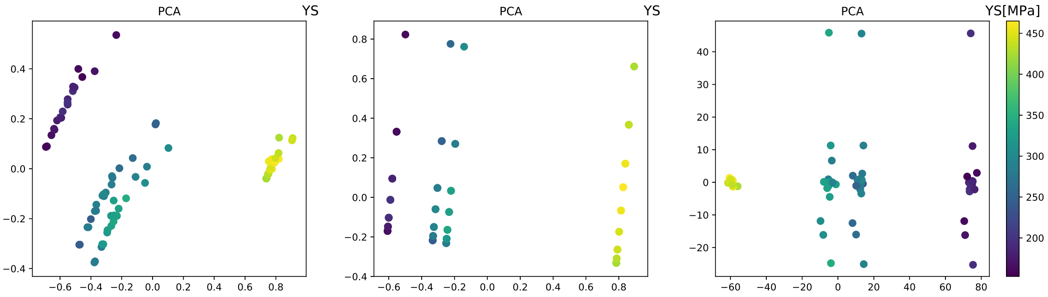
3.1 Data analysis

The next study addresses the question whether unsupervised learning methods would help further reduce the dimensionality of the inputs. The known "curse of dimensionality problem, " hinders the useful information of a data set and leads to computational instability. Therefore, the selection of the features is a challenging task [59]. Thus, we plotted the first few leading components from each unsupervised learning method color-coded by the increase in yield stress. Is important to highlight that there are components, consequently, they has not a direct association with the real features.

As has mentioned in the Section 2 , one of the most interesting unsupervised learning methods to study the data is the PCA. For this reason, is going to be analyzed with the aim to find some other patterns. The following Figure 9 show the values of the first two principle components using PCA for the three input types: pixels, statistical and estimated statistical.

The most remarkable differences between the methods are the slope of the data distribution. In the estimated statistical plot is possible to see that the new feature subspace where the data set is projected, has a small trend to incline itself to the left. It means two things, in one hand it indicates a large negative covariance confirming that they are negative related . On the other hand, it indicated that despite the first feature has the biggest variance there is still an important influence of the second one. As far as the other two are concerned, their performance is very similar (almost a vertical line) but the difference is in the points distribution. The pixels has a nucleus and some points far of it but the statistical data are focused in one single point for each energy.

Other methods has been studied to compare their performance with the PCA. The plots in Figure 10 show how well the leading two principle of components from the ICA, tSNE, LDA and MDS can separate the data as a function of the increase in yield stress. Anyways, using pixels as input, the middle stacking fault energies (84Mpa and 72Mpa) are very close each other. Consequently, the matrix decomposition techniques shows that the variance of these two energies is small. Also, it is important to see that using



(a) Estim. statistical (b) Statistical (c) Pixels

Figure 9: PCA with three input types scaled

the unsupervised models, a high clustering is achieved for both statistical and estimated statistical . This performance gives an intuition about how close is the data depending of the components. Thus, being focused in one single point seems that one feature will be a very high influence for the classification of the data.

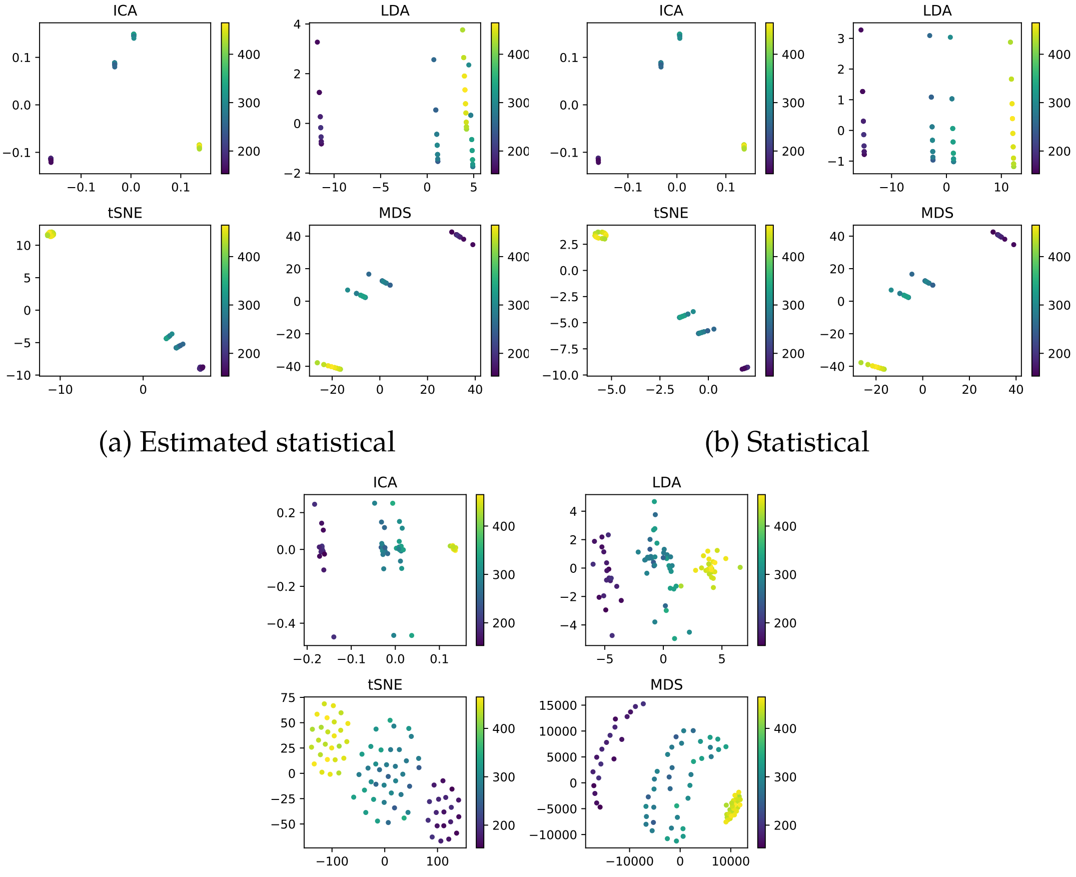
In the Section 3.3.2 the differences between the prediction of each stacking fault energy will be further discussed.

With the aim to know how much of the total, 3 or 65536 dimensional variance is contained within each component, the explained variance ratio (EVR) is studied. It is a percentage of variance explained by each of the selected components. In the Equation 10 where *Var* is the variance, is possible to see how it is calculate [37].

$$EVR(y, \hat{y}) = 1 - \frac{Var(y - \hat{y})}{Var(y)} \tag{10}$$

Table 2 lists the explained variance ratio of the principle components of PCA for the different input types. We see that the first principle component captures information regarding the mean stacking fault energy, and the third principle component captures the mean region size. We can conclude from the PCA plots that the standard deviation of the stacking fault energy has negligible influence on the change in yield stress.

This table also shows that scaling the data the EVR is decreased in both



(a) Estimated statistical

(b) Statistical

(c) Pixels

Figure 10: Unsupervised results

estimated statistical and statistical. This is because the magnitude of the data are normalized and consequently, the features are homogenized.

			Component		
Metric	Input	Type	First	Second	Third
Explained variance ratio [%]	Estimated statistical	Original	99.99	7.88e-03	2.53e-04
		PCA	99.99	7.88e-03	2.53e-04
		Scaled	86.33	10.45	3.20
	Statistical	Original	95.69	3.64	0.66
		PCA	95.69	3.64	0.66
		Scaled	76.15	16.75	7.08
	Pixels	Original	52.83	2.63	2.57
		PCA	52.83	2.63	2.57
		Scaled	52.84	2.62	2.55

Table 2: Explained variance ratio

3.2 Input processing analysis

This numerical study focuses on answering the question which input and preprocessing technique would be the best to describe the change in yield stress.

Table 3 shows the result of learning with the three types of inputs. We used several different measures of model accuracy and they are defined in the Section 2.3.1 of the report. We see that all three input types demonstrate comparable accuracy. At the same time, this test has repeated scaling the results as well as using the first three principle components from PCA to train machine learning models and see if there is any improvement with PCA principle components and dimension reduction. Table 3 summarizes the accuracy of the best performing machine learning model for each input type. *CT* is the computational time needed for the test.

As was discussed in the Section 3.1, using the firsts principle components of the pixels is possible capture 59% of the yield stress. Furthermore, the Table 4 shows that with this three components is possible achieve similar accuracy than with all of them. Thus, is possible conclude that the dimensionality could be reduced without worsen the results. In the statistical case, is possible to see this same performance using only the first component. Finally, the estimated statistical needs at least two components to make accurate predictions.

One common method to evaluate the machine learning models predictions

Input type	Model	Metric	Original	Scaled	PCA
Estimated statistical	GBR	R2	0.9843	0.9845	0.9851
		MAE [MPa]	8.69	8.55	8.92
		RMSE [MPa]	12.03	11.93	13.11
		MAPE [%]	3.14	3.09	3.18
		CT [ms]	86	61	72
Statistical	GBR	R2	0.9950	0.9951	0.9946
		MAE [MPa]	5.38	5.38	5.72
		RMSE [MPa]	6.76	6.76	7.05
		MAPE [%]	1.95	1.95	2.08
		CT [ms]	62	60	60
Pixels	GPR	R2	0.9693	0.9693	0.9693
		MAE [MPa]	14.32	14.33	14.32
		RMSE [MPa]	17.22	17.22	17.22
		MAPE [%]	5.07	5.07	5.07
		CT [ms]	30122	16035	287

Table 3: Preprocessing

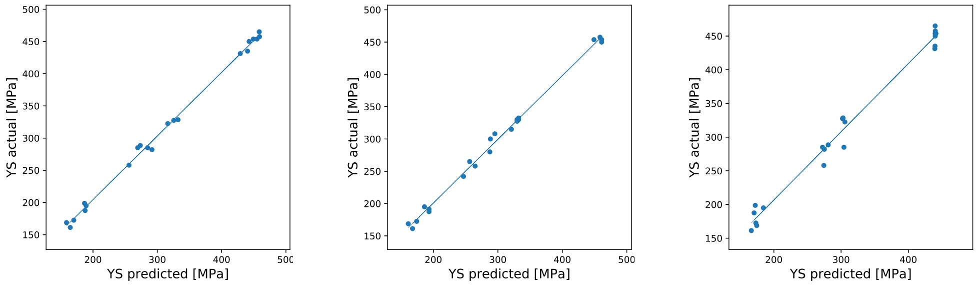
Input type	Model	#Comp	R^2	MAE[MPa]	RMSE[MPa]	MAPE[%]
Estimated statistical	GBR	1	0.7192	40.57	57.13	15.96
	GBR	2	0.9849	9.21	13.20	3.32
	GBR	3	0.9851	8.92	13.11	3.18
Statistical	GBR	1	0.9934	6.37	7.74	2.34
	GBR	2	0.9933	6.32	7.80	2.31
	GBR	3	0.9946	5.72	7.05	2.08
pixels	GPR	1	0.9745	13.55	17.20	4.65
	GPR	5	0.9717	16.60	18.12	5.83
	GPR	83	0.9693	14.32	17.22	5.07

Table 4: Components analysis

is scattering the predicted and observed values. In the publication done by Gervasio et al. [51] is demonstrated analytically that this model evaluation should be done placing the observed values in the y-axis and the predicted values in the x-axis.

This plot shows the effect of the model and compares it against the null model (vertical line). The way to analyze their performance is through the slope of the mean line as well as the dispersion of the values. The most accurate result is the one where all the points are in the line and has a slope of 45 degrees (absolutely diagonal).

In the Figure 11 is possible to see that in all the cases, the bias of the results is really small and the slope of the mean line is diagonal. Thus, in all these cases the accuracy is really high and difficult to differentiate the best model at first sight.



(a) Estimated statistical (b) Statistical (c) pixels

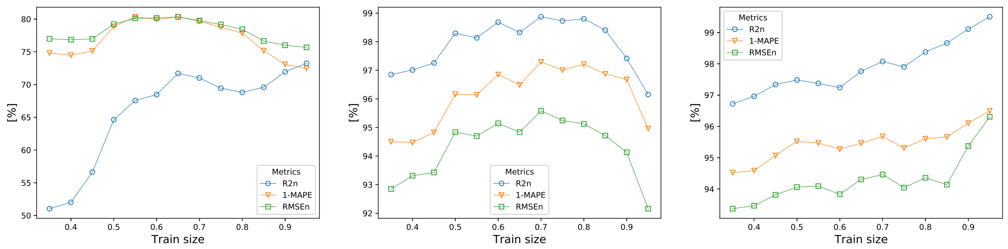
Figure 11: Yield stress known vs predicted

With the aim to find the influence of the optimal training data size, two different test has been conducted. The fist one iterate through different sizes and the second one analyze the learning curves of each input to see the bias and variance in each case.

The amount of data needed depend on the problem complexity as well as the model used. For this reason, in this case has been iterate the training size from 30% to 95% with the aim to find the optimal value.

As was commented in the Section 2, with the small data set available (only 83 pictures), the model need to be complex enough to get the parameters to predict exactly the test values. In this case is possible to see that using around 70% of the samples as training the model proposed has enough information to do accurate predictions. To be sure that the model is not in the critical zones of *over-fitting* or *under-fitting*, the learning curves of each inputs are going to be analyzed.

In Section 2.3.3, we introduced the concept of learning curves. Next we will show the learning curves for the three input types. All the test uses 5 fold

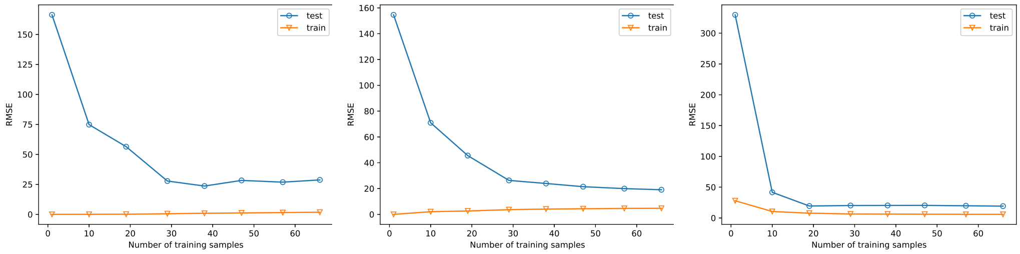


(a) Estimated statistical (b) Statistical (c) pixels

Figure 12: Train size analysis

cross-validation. For the estimated statistical and the statistical the GBR model is used and for the pixels, the GPR. The training size goes from 1 sample to 66 samples.

Firstly, in the Figure 13 is possible to see that with low training data a high variance (gap between the curves) is achieved. Despite have a small training error, the testing one is high. With more data, the testing error is exponentially decreased and the training one increasing a little bit. This happens until a point where both curves remain stable showing that the use of more data is not useful to improve the results.

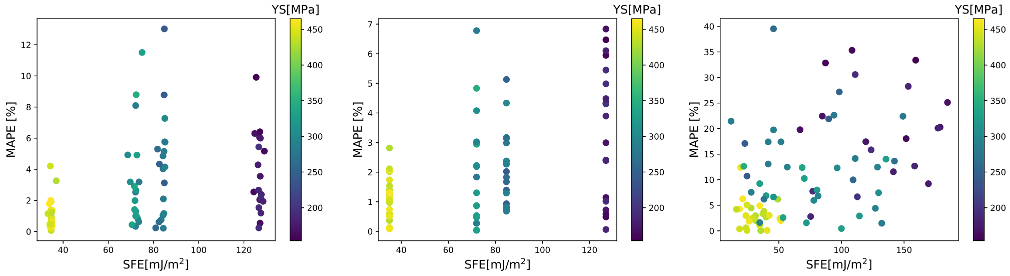


(a) Estimated statistical (b) Statistical (c) Pixels

Figure 13: Learning curves analysis

The last method to see the differences between the three types of inputs is the leave one out technique. In the Section 2.3.2 was introduced but now the results are going to be discussed.

The Figure 14 shows the error of each sample predicted depending of their stacking fault energy. Thus, is possible to see that the performance of the statistical data is highly better than the other inputs. Because in both cases statistical and estimated statistical are used the mean of the stacking fault energy, their distribution is homogeneous. On the other hand, the pixels has a continuous distribution and that is why the error increases substantially.



(a) Estimated statistical (b) Statistical (c) Pixels

Figure 14: Leave one out

3.3 Extrapolation

3.3.1 Region size

In this section one region size will be used as test and the rest as training to evaluate if the models are capable to do this extrapolation. Figure 15 shows the model accuracy when we leave out each set of data. We see that the machine learning model performs poorer when we explicitly exclude data from a given region size or a mean stacking fault energy. Anyways this difference is not meaningful.

It is interesting to see that this is the first simulation where GBR and GPR do not gives the best performance. Anyways, for each prediction a small predicting error is achieved.

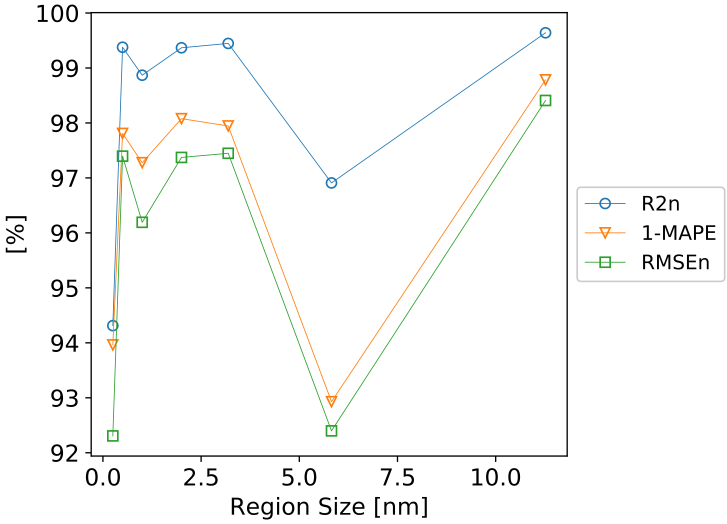


Figure 15: Region size extrapolation

The grain size extrapolations are in good agreement with the extended dislocations performance where bend to remain in zones with lower stacking fault energy [19]. Thereby, the accuracy is substantially decreased in the 5.8nm region size.

Despite have less samples, the prediction of the 11nm region size and be in

Region Size[nm]	Model	R2	MAE[MPA]	RMSE[MPA]	MAPE[%]
0.25	KRR	0.9431	17.59	22.80	6.03
0.5	BRR	0.9890	8.13	9.67	2.57
1	GPR	0.9886	7.67	9.66	2.72
2	GBR	0.9937	5.84	7.33	1.92
3.19	KNR	0.9941	5.86	7.32	2.08
4.5	GPR	-0.5221	3.23	3.77	0.69
5.82	BRR	0.9619	25.91	30.04	11.01
8	BRR	-0.6887	2.93	4.59	0.67
11.27	KNR	0.9970	4.20	5.68	1.40

Table 5: Extrapolation region size

boundaries of the dataset, has a really small error. Conversely, the 0.25nm is predicted with more error.

3.3.2 Stacking fault energy

Our final study assess whether the machine learning model can extrapolate the change in yield stress if we have not trained the modeling using a given mean stacking fault energy or a mean patch size.

To study this, we explicitly remove from our data a pre-defined mean patch size or a mean stacking fault energy. Then we use the removed data as test inputs to see if the machine learning model can predict its change in yield stress.

We see that using the change in yield stress, the machine learning model doesn't seem to perform as good since the yield stress is of order of 400MPa-100MPa, so a few MPa error do not seems like a lot.

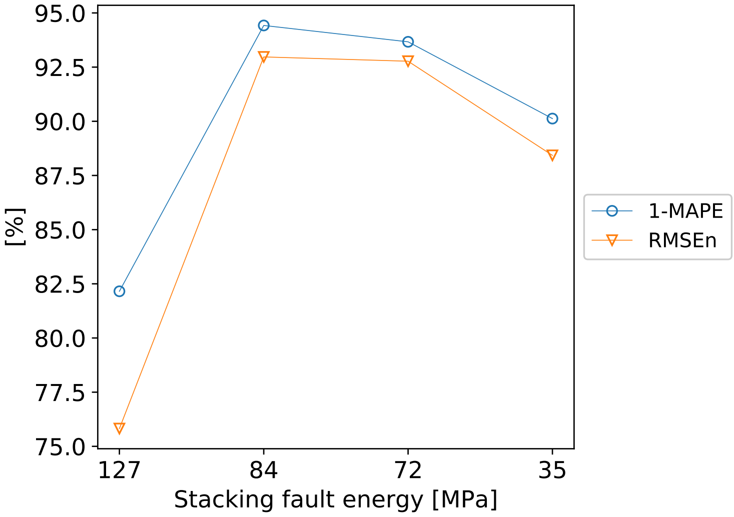


Figure 16: Stacking faul energy extrapolation

SFE[MJ/m]	Model	R2	MAE[MPA]	RMSE[MPA]	MAPE[%]
127	KNR	-5.79	32.35	34.93	17.84
84.7	KNR	-0.2234	14.40	19.91	5.57
72	SVR	-0.2409	20.11	22.06	6.33
35	KNR	-13.30	44.23	46.23	9.87

Table 6: Prediction error when we leave out data from each mean stacking fault energy

3.4 Interpolation

In this section will be discussed the performance of the machine learning models interpolation values into one single stacking fault energy. This test has not been done with the different region sizes because the small amount of data (3-12 samples) for each one was not enough to do accurate predictions.

3.4.1 Stacking fault energy

To do the stacking fault energies interpolation, every single energy is split and then using K-fold=5 the samples are predicted.

In the Table 7 is possible to see that despite the small amount of data to do the interpolation, is possible to do accurate predictions with less than 5% of error. As we have seen i the Table 6, due to be the more farther energy, the 127 SFE has the higher error doubling the other ones.

Model	SFE	R2	MAE[MPA]	RMSE[MPA]	MAPE[%]
GBR	127	0.7587	7.34	8.55	4.11
GBR	84	0.6939	6.74	8.00	2.44
GBR	72	0.5886	6.67	8.80	2.13
GBR	35	0.6937	4.69	5.51	1.05

Table 7: Interpolation

4 Discussion and concluding remarks

The surrogate models presented here proved the efficiency of machine learning methods in learning the relationship between yield stress and the stacking fault energy in high entropy alloys.

There has been a lot of interest in using ML methods as a surrogate model for physics based models. Previous studies in the materials analysis field used machine learning models, chemical environments [58], microstructure optimization [56], the upscaling in random microstructures[57]. Our study has been the first to apply ML to study the correlation between the yield stress and the stacking fault energy in alloys in high entropy alloys.

PFDM is computationally intensive, the proposed ML model is a sophisticated pattern recognition method programmed to learn to the results of PFDM. It allows time cycle reduction and efficient utilization of resources. As it has been proved in the Section 3.1 this model helps use to handle multi-dimensional data which have various variety of data types in a dynamic environment.

Using unsupervised learning has been proved the capability of different unsupervised techniques to separate the data as a function of the increase in yield stress (Section 2.2.2). Also, we can conclude from the PCA plots that the standard deviation of the stacking fault energy for estimated statistical and statistical inputs has negligible influence on the change in yield stress. A dimensional reduction is also possible with the pixels input.

In section 3.2, we demonstrated that estimated statistical and not estimated inputs are much more efficient since they can achieve the same accuracy with significantly fewer degrees of freedom. In addition, the estimated statistical input is amenable for predicting outcomes of PFDM simulations since the mean and standard deviation of the stacking fault energies, and the mean region size are readily available information. The statistical input is more suitable for predicting experimental measurements of stacking fault energy landscape in a high entropy alloy, since we can easily compute the estimated statistical mean and standard deviation and number of pixels with approximately equal stacking fault energy from an image data. Due the large discrepancy in the size of pictures compared to estimated statistical and statistical data, the computational cost of building a surrogate model using the pixels is significantly higher than the other ones.

As for the further research and improvements, an obvious step would be to improve the accuracy of the stacking fault energies extrapolation optimizing the hyperparameters or introducing new machine learning models.

5 Acknowledgment

Pau C.V. thanks Sr. Pete J. Balsells, the University of Colorado and the Generalitat de Catalunya for the Balsells research fellowship and its the financial support, this work would not have been possible without them. Pau is especially indebted to the University of Colorado at Colorado Springs who have been supportive of his career goals and who worked actively to provide him the tools to pursue those goals.

Pau C.V. thanks the project collaborators, Marisol Koslowski, Katerine Saleme and Somayyeh Sheikholeslami for supplying the computational data and their insight and expertise that greatly assisted the research.

Nobody has been more important to Pau in the pursuit of this project than the members of his family. He would like to thank his parents, whose love, guidance and unending inspiration are with him in whatever he pursue. They are the ultimate role models.

6 Bibliography

References

[1] Pehlke, Robert & Jeyarajan, A & Wada, H. (1982). Summary of Thermal Properties for Casting Alloys and Mold Materials. NASA STI/Recon Technical Report N. 83. 36293.

[2] Yeh, Jien-Wei & Chen, S.-K & J. Lin, SU & Gan, Jon-Yiew & Chin, Tsung-Shune & Shun, T.-T & Tsau, C.-H & Chang, SY. (2004). Nanostructured High-Entropy Alloys with Multiple Principal Elements: Novel Alloy Design Concepts and Outcomes. Advanced Engineering Materials. 6. 299 - 303. 10.1002/adem.200300567.

[3] Clemens, Helmut & Mayer, Svea. (2013). Design, Processing, Microstructure, Properties, and Applications of Advanced Intermetallic TiAl Alloys. Advanced Engineering Materials. 15. 10.1002/adem.201200231.

[4] Sheikh, Saad. (2016). Alloy Design and Optimization of Mechanical Properties of High-Entropy Alloys.

[5] Wu, Yidong & Si, Jiajia & Lin, De-Ye & Wang, Tan & Yi Wang, William & Wang, Yandong & Liu, ZiKui & Hui, Xidong. (2018). Phase stability and mechanical properties of AlHfNbTiZr high-entropy alloys. Materials Science and Engineering: A. 724. 10.1016/j.msea.2018.03.071.

[6] Senkov, O., Wilks, G., Scott, J. and Miracle, D. (2011). Mechanical properties of Nb25Mo25Ta25W25 and V20Nb20Mo20Ta20W20 refractory high entropy alloys. Intermetallics, 19(5), pp.698-706.

[7] Gali, Aravind & George, E.P.. (2013). Tensile properties of high- and medium-entropy alloys. Intermetallics. 39. 74–78. 10.1016/j.intermet.2013.03.018.

[8] Yifan, Ye & Wang, Qing & Lu, Jiatian & Liu, C.T. & Yang, Yancong. (2015). High-entropy alloy: challenges and prospects. Materials Today. 19. 10.1016/j.mattod.2015.11.026.

[9] Wu, Wei-Hong & Yang, Chih-Chao & Yeh, Jien-Wei. (2006). Industrial development of high-entropy alloys. European Journal of Control - EUR J CONTROL. 31. 737-747. 10.3166/acsm.31.737-747.

[10] Miracle, Dan & D. Miller, Jonathan & Senkov, Oleg & Woodward, C & D. Uchic, Michael & Tiley, J. (2013). Exploration and Development of High Entropy Alloys for Structural Applications. Entropy. 16. 10.3390/e16010494.

[11] Yeh, Jien-Wei. (2013). Alloy Design Strategies and Future Trends in High-Entropy Alloys. JOM. 65. 10.1007/s11837-013-0761-6.

[12] Gludovatz, Bernd & Hohenwarter, Anton & Catoor, Dhiraj & H. Chang, Edwin & P. George, Easo & Ritchie, Robert. (2014). A Fracture-Resistant High-Entropy Alloy for Cryogenic Applications.. Science. 345. 1153-1158. 10.1126/science.1254581.

[13] Yeh, Jien-Wei. (2006). Recent progress in high-entropy alloys. European Journal of Control - EUR J CONTROL. 31. 633-648. 10.3166/acsm.31.633-648.

[14] J. P. Hirth and J. Lothe, Theory of dislocations (Krieger Pub. Co., Malabar, FL, 1992), 2nd edn.

[15] D. Hull and D. J. Bacon, Introduction to dislocations (Butterworth-Heinemann, Oxford Oxfordshire ; Boston, 2001), 4th edn.

[16] Borovikov, Valery & I. Mendelev, Mikhail & H. King, Alexander & LeSar, Richard. (2015). Effect of Stacking Fault Energy on Mechanism of Plastic Deformation in Nanotwinned FCC Metals. Modelling and Simulation in Materials Science and Engineering. 23. 10.1088/0965-0393/23/5/055003.

[17] Schoeck, Gunther. (1995). The core energy of dislocations. Acta Metallurgica et Materialia. 43. 3679-3684. 10.1016/0956-7151(95)90151-5.

[18] Rao, Satish & Woodward, C & Parthasarathy, Triplicane & Senkov, Oleg. (2017). Atomistic Simulations of Dislocation Behavior in a Model FCC Multicomponent Concentrated Solid Solution Alloy. Acta Materialia. 134. 10.1016/j.actamat.2017.05.071.

[19] Sun, Pei-Ling & H. Zhao, Y & Cooley, Jason & Kassner, M.E. & Horita,

Z & Langdon, T.G. & Lavernia, Enrique & Zhu, Yuntian. (2009). Effect of stacking fault energy on strength and ductility of nanostructured alloys: An evaluation with minimum solution hardening. Materials Science and Engineering A-structural Materials Properties Microstructure and Processing - MATER SCI ENG A-STRUCT MATER. 525. 83-86. 10.1016/j.msea.2009.06.030.

[20] Bergstra, James & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. The Journal of Machine Learning Research. 13. 281-305.

[21] Quinlan, Ross. (1986). Induction of Decision Trees. Machine Learning. 1. 81-106. 10.1007/BF00116251.

[22] Freund, Yoav & E Schapire, Robert. (1999). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. 55. 119-139. 10.1006/jcss.1997.1504.

[23] Kearns, M. (1988). Thoughts on Hypothesis Boosting. Machine Learning class project.

[24] Basak, Debasish & Pal, Srimanta & Chandra Patranabis, Dipak. (2007). Support Vector Regression. Neural Information Processing – Letters and Reviews. 11.

[25] Drucker, Harris & Burges, Christopher & Kaufman, Linda & Smola, Alexander & Vapnik, V. (1997). Support vector regression machines. Adv Neural Inform Process Syst. 28. 779-784.

[26] Smola, Alexander & Burges, Chris & Drucker, Harris & Golowich, Steve & Van Hemmen, Leo & Müller, Klaus-Robert & Scholkopf, Bernhard & Vapnik, Vladimir. (2003). Regression Estimation with Support Vector Learning Machines.

[27] Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Methods. Cambridge University Press, Cambridge (2000)

[28] Vovk, Vladimir. (2013). Kernel Ridge Regression. 10.1007/978-3-642-41136-6-11.

[29] E. Rasmussen, C & K. I. Williams, C. (2006). Gaussian Process for Ma-

chine Learning.

[30] Nagelkerke, N.J.D.. (1991). A More General Definition of the Coefficient of Determination. Biometrika. 78. 10.1093/biomet/78.3.691.

[31] Ramchoun, Hassan & Amine, Mohammed & Janati Idrissi, Mohammed Amine & Ghanou, Youssef & Ettaouil, Mohamed. (2016). Multilayer Perceptron: Architecture Optimization and Training. International Journal of Interactive Multimedia and Artificial Inteligence. 4. 26-30. 10.9781/ijimai.2016.415.

[32] Bunker, Rory & Thabtah, Fadi. (2017). A Machine Learning Framework for Sport Result Prediction. Applied Computing and Informatics. 15. 10.1016/j.aci.2017.09.005.

[33] Hinton, G. E., & Roweis, S. T. (2003). Stochastic neighbor embedding. In Advances in neural information processing systems (pp. 857-864).

[34] Borg, I., & Groenen, P. (2003). Modern multidimensional scaling: Theory and applications. Journal of Educational Measurement, 40(3), 277-280.

[35] Comon, P. (1994). Independent component analysis, a new concept?. Signal processing, 36(3), 287-314.

[36] Ye, J., Janardan, R., & Li, Q. (2005). Two-dimensional linear discriminant analysis. In Advances in neural information processing systems (pp. 1569-1576).

[37] Buitinck, Lars & Louppe, Gilles & Blondel, Mathieu & Pedregosa, Fabian & Mueller, Andreas & Grisel, Olivier & Niculae, Vlad & Prettenhofer, Peter & Gramfort, Alexandre & Grobler, Jaques & Layton, Robert & Vanderplas, Jake & Joly, Arnaud & Holt, Brian & Varoquaux, Gael. (2013). API design for machine learning software: Experiences from the scikit-learn project. API Design for Machine Learning Software: Experiences from the Scikit-learn Project.

[38] De Myttenaere, A., Golden, B., Le Grand, B., & Rossi, F. (2016). Mean absolute percentage error for regression models. Neurocomputing, 192, 38-48.

[39] Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or

mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. Geoscientific model development, 7(3), 1247-1250.

[40] Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate research, 30(1), 79-82.

[41] Kohavi, Ron. (2001). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. 14.

[42] Lachenbruch, P. & Mickey, M. (1968) "Estimation of error rates in discriminant analysis". Technometrics, vol 10, pp 167-178.

[43] Deng, B. C., Yun, Y. H., Liang, Y. Z., Cao, D. S., Xu, Q. S., Yi, L. Z., & Huang, X. (2015). A new strategy to prevent over-fitting in partial least squares models based on model population analysis. Analytica chimica acta, 880, 32-41.

[44] Fortmann-Roe, S. (2012). Understanding the Bias-Variance Tradeoff. [online] Available at: http://scott.fortmann-roe.com/docs/BiasVariance.html

[45] Anzai, Y. (2012). Pattern recognition and machine learning. Elsevier.

[46] Olteanu, A. (2018). Tutorial: Learning Curves for Machine Learning in Python. [online] Dataquest. Available at: https://www.dataquest.io/blog/learning-curves-machine-learning/.

[47] C. Gao, Michael & Qiao, Junwei. (2018). High-Entropy Alloys (HEAs). Metals. 8. 108. 10.3390/met8020108.

[48] D. P. Solomatine and D. L. Shrestha, "AdaBoost.RT: a boosting algorithm for regression problems," 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541), Budapest, 2004, pp. 1163-1168 vol.2. doi: 10.1109/IJCNN.2004.1380102

[49] Abdi, H., & Williams, L. J. (2010). Principal component analysis. Wiley interdisciplinary reviews: computational statistics, 2(4), 433-459.

[50] Pierce, D.T. & JimÉNEZ, J & Bentley, J & Raabe, D & Wittig, J.E.. (2015). The Influence of Stacking Fault Energy on the Microstructural and Strain-

hardening Evolution of Fe-Mn-Al-Si Steels During Tensile Deformation. Acta Materialia. 100. 178-190. 10.1016/j.actamat.2015.08.030.

[51] Piñeiro, Gervasio & Perelman, Susana & Guerschman, Juan & Paruelo, José. (2008). How to Evaluate Models: Observed vs. Predicted or Predicted vs. Observed? Ecological Modelling. 216. 316-322. 10.1016/j.ecolmodel.2008.05.006.

[52] Okamoto, Norihiko & Fujimoto, Shu & Kambara, Yuki & Kawamura, Marino & M. T. Chen, Zhenghao & Matsunoshita, Hirotaka & Tanaka, Katsushi & Inui, Haruyuki & P. George, Easo. (2016). Size effect, critical resolved shear stress, stacking fault energy, and solid solution strengthening in the CrMnFeCoNi high-entropy alloy. Scientific Reports. 6. 10.1038/srep35863.

[53] Cordero, Zachary & E. Knight, Braden & Schuh, Christopher. (2016). Six decades of the Hall–Petch effect – a survey of grain-size strengthening studies on pure metals. International Materials Reviews. 61. 1-18. 10.1080/09506608.2016.1191808.

[54] El-Danaf, Ehab & Soliman, Mahmoud & A. Al-Mutlaq, Ayman. (2015). Correlation of Grain Size, Stacking Fault Energy, and Texture in Cu-Al Alloys Deformed under Simulated Rolling Conditions. Advances in Materials Science and Engineering. 2015. 1-12. 10.1155/2015/953130.

[55] Zhao, Y.H. & Zhu, Yuntian & Liao, Xiaozhou & Horita, Zenji & G. Langdon, Terence. (2007). Influence of stacking fault energy on the minimum grain size achieved in severe plastic deformation. Materials Science and Engineering: A. 463. 22-26. 10.1016/j.msea.2006.08.119.

[56] Liu, Ruoqian & Kumar, Abhishek & Chen, Zhengzhang & Agrawal, Ankit & Sundararaghavan, Veera & Choudhary, Alok. (2015). A predictive machine learning approach for microstructure optimization and materials design. Scientific reports. 5. 11551. 10.1038/srep11551.

[57] Koutsourelakis, P. S. (2007). Stochastic upscaling in solid mechanics: An excercise in machine learning. Journal of Computational Physics, 226(1), 301-325.

[58] Bartók, A. P., De, S., Poelking, C., Bernstein, N., Kermode, J. R., Csányi,

G., & Ceriotti, M. (2017). Machine learning unifies the modeling of materials and molecules. Science advances, 3(12), e1701816.

[59] Kumar, Mukesh & Rath, Santanu. (2016). Feature Selection and Classification of Microarray Data Using Machine Learning Techniques. 10.1016/B978-0-12-804203-8.00015-8.

[60] Zhang Z. Introduction to machine learning: k-nearest neighbors. Ann Transl Med. 2016;4(11):218. doi:10.21037/atm.2016.03.37

[61] Brown, P. J., & Zidek, J. V. (1980). Adaptive multivariate ridge regression. The Annals of Statistics, 8(1), 64-74.

7 Apendix A

- Supervised learning models:
  KNeighborsRegressor($n_{n}eighbors$=2, weights=’distance’)
  BayesianRidge($alpha_{1}$=1e-07, $lambda_{1}$=1e-05, $n_{i}ter$=100, tol=0.001)
  DecisionTreeRegressor(criterion=’mse’, $max_{d}epth$=7)
  GradientBoostingRegressor($learning_{r}ate$=0.1, loss=’ls’, $max_{d}epth$=3, $n_{e}stimators$=80)
  KernelRidge(alpha=0.001, degree=2, gamma=1, kernel=’rbf’)
  GaussianProcessRegressor(alpha=0.01, kernel=Matern($length_{s}cale$=1, nu=1.5))
  SVR(C=1000, degree=3, kernel=’rbf’, gamma=1)
  MLPRegressor(activation=’relu’, $hidden_{l}ayer_{s}izes$=(100,), $learning_{r}ate$=’constant’, $max_{i}ter$=10000, solver=’lbf gs’)
  AdaBoostRegressor($learning_{r}ate$=1, loss=’square’, $n_{e}stimators$=40)

8 Additional Information

Supplementary information and the full code accompanies this report at:
https://github.com/paucutrina/HighentropyalloyML

How to cite this report:
Cutrina P., Wang X. et al. Applying machine learning methods to predict the yield stress in high entropy alloys. (2019).

Competing financial interests:
The authors declare no competing financial interests.