Machine Learning Prediction of Stacking Fault Energy in Steel Alloys Based on Chemical Composition

Ikponmwosa J. Iyinbor
University of Southern California
iyinbor@usc.edu

Ken-ichi Nomura
University of Southern California
knomura@usc.edu

Paulo S. Branicio
University of Southern California
branicio@usc.edu

ABSTRACT

Stacking fault energy (SFE) is a critical parameter in the design of steels with desirable mechanical properties such as strength, ductility, and strain-hardening rate. SFE influences secondary deformation mechanisms like Transformation Induced Plasticity (TRIP) and Twinning Induced Plasticity (TWIP). This work involves creating a machine learning model to classify steel alloys into low, medium, or high SFE categories, aiding in the prediction of secondary deformation behaviors. Data from literature containing experimental and theoretical SFE values for various steel alloy compositions were compiled and preprocessed, resulting in a dataset of 374 observations. Using this dataset, several machine learning models, including Feedforward Neural Network (FFNN), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting Regressor (GBR), and CatBoost Regressor (CAT), and Adaptive Boost Regressor (ADB) were trained and evaluated for SFE prediction accuracy. Two models, SVM and RF, emerged as the top-performing models. To enhance accuracy and reduce misclassification, threshold probabilities were applied, allowing fuzzy classification when model uncertainty was high. Validation against literature data showed strong agreement between predictions and reported SFE values. This study provides valuable insights into predicting SFE and guiding the development of austenitic steel alloys with tailored properties.

KEYWORDS

Supervised Learning, Stacking Fault Energy, Random Forest, Gradient Boosting, Adaptive Boost, Support Vector Machine, Neural Network

1 INTRODUCTION

A stacking fault is an interruption in the normal stacking sequence of atomic planes, and the energy required to create this interruption is known as the stacking fault energy (SFE). The SFE has great importance in designing steels with superior combination of mechanical strength, ductility and strain hardening rate. SFE plays an important role in activating secondary deformation mechanisms such as Transformation-induced and twinning-induced plasticity (TRIP, TWIP) observed in austenitic steels. Experimental work has shown that these deformation mechanisms are primarily a function of their SFE, such that with increasing SFE from low to high, the deformation mechanisms change from TRIP (< 20 mJ/m^2) to TWIP (between 20 mJ/m^2 – 45 mJ/m^2) and then to MBIP (> 45 mJ/m^2), which stands for Microband-Induced Plasticity as shown in figure 1 [2–5, 10, 11].

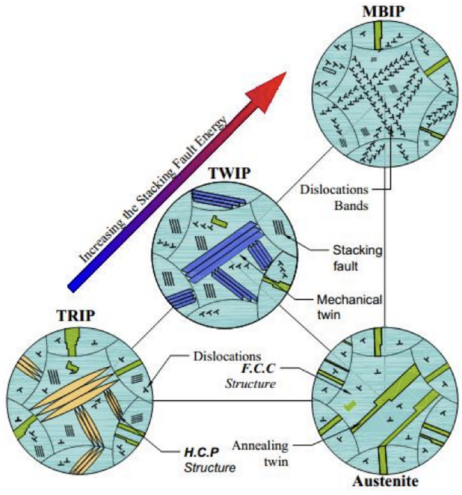

Figure 1: Schematic representation of secondary deformation mechanisms (TRIP, TWIP, and MBIP) in austenitic steels
Note: Reprinted with permission from [2].

Due to this relationship between SFE and secondary deformation mechanisms, knowing the SFE of a steel alloy is crucial in designing advanced structural materials [3, 5]. Several techniques, such as measuring the geometry of dislocations and the diffraction profile, have been used to compute SFE values [7, 9]. However, measuring exact SFE values can be challenging because there are different methods to calculate it, and each has its own sources of error, such as material constants used in the calculation or other physical and material interactions (strain in material) to name a few [3]. Regardless of these uncertainties in calculating exact SFE values, it is possible to classify austenitic steel alloys into different SFE regimes (low SFE (< 20 mJ/m^2), medium SFE (between 20 mJ/m^2 – 45 mJ/m^2), and High SFE (> 45 mJ/m^2). While exact SFE values can vary due to experimental errors or other factors, knowing the general SFE regime is enough to determine the secondary deformation mechanisms in austenitic steels [3, 4].

This project involves creating a machine learning model to predict the SFE regime for different steel alloy compositions. The training data for this model comes from a database of information from various research studies in literature that cover a range of steel

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright ©JOCSE, a supported publication of the Shodor Education Foundation Inc.

© 2026 Journal of Computational Science Education
https://doi.org/10.22369/issn.2153-4136/17/1/1

alloy compositions, including both experimental and theoretical SFE calculations [3].

2 DATA PREPROCESSING AND VISUALIZATION

2.1 Data Collection and Classification

The dataset used in this study was obtained from a database curated by extensively gathering information from literature on various steel alloy compositions [9]. The data included results from experimental and theoretical Stacking Fault Energy (SFE) calculations based on these compositions. The focus was on elements commonly used in the design of austenitic steels, ensuring that our dataset represented a comprehensive range of these alloying elements. The complete list of elements considered for data collection included: C, N, P, S, V, Ni, Nb, Al, Ti, Fe, Hf, Mo, Mn, Co, Si, Cr, and Cu.

The SFE data came from various temperature ranges and was collected through different experimental techniques like Transmission Electron Microscopy (TEM), X-ray Diffraction (XRD), Neutron Diffraction, and thermodynamic modeling. This resulted in a dataset with about 500 data points, each representing a unique combination of material composition, temperature, and method used to obtain the SFE.

To prepare the data for analysis, we applied a systematic approach to data preprocessing. Initially, a subset of 426 observations was chosen, including only data collected at room temperature (300 K), because most of the data in the complete dataset were taken at this temperature, so it made sense to exclude temperature as a variable when training our models. Next, the subset was further refined by removing theoretical modeling data and keeping only the experimental measurements since these represent truth values for SFE. This reduced the subset to 387 observations. Data related to nickel-based alloys was removed, leaving only measurements from ferrous-based alloys, resulting in 379 observations. Lastly, another step involved filtering to ensure there were enough data points from each alloying element used in austenitic steels. Elements such as P, S, V, Ti, Hf, Nb, Co, and Cu had too few data points, so they were excluded from the training dataset. The final list of elements used as predictor variables in the training input data included C, N, Ni, Al, Fe, Mo, Mn, Si, and Cr. After all filtering steps, the final training dataset contained 374 observations. This final subset was then used for further analysis, providing a balanced and reliable data set for the study.

As mentioned in the introduction, understanding the general SFE regime is sufficient to predict deformation mechanisms. Therefore, as seen in Figure 2, each steel alloy was divided into three SFE categories: Low, Medium, and High, corresponding to the secondary deformation mechanisms TRIP, TWIP, and dislocation glide, respectively. The final training dataset contains 77 observations in the Low SFE class, 209 in the Medium SFE class, and 88 in the High SFE class.

2.2 Data Visualization and Dimensionality Reduction

Initially, the relationship between two elements, Ni and Cr, and their impact on SFE is visualized, as shown in Figure 3. The 3D plot

```
#determining class based on SFE value
SFEclass = np.zeros(SFEdata.shape[0])

SFEclass[SFEdata[:,10] <= 20] = 1
SFEclass[(SFEdata[:,10] > 20) & (SFEdata[:,10] <= 45)] = 2
SFEclass[(SFEdata[:,10] > 45)] = 3

#check number of entries for each class
total_low_sfe = (SFEclass == 1).sum()  # Calculate the total number of observations in the low SFE class
total_medium_sfe = (SFEclass == 2).sum()  # Calculate the total number of observations in the low SFE class
total_high_sfe = (SFEclass == 3).sum()  # Calculate the total number of observations in the low SFE class

print(f"There are a total of {total_low_sfe} observations in the low SFE class.")
print(f"There are a total of {total_medium_sfe} observations in the medium SFE class.")
print(f"There are a total of {total_high_sfe} observations in the high SFE class.")

#check if only the needed classes there
print ((SFEclass == 1).sum() + (SFEclass == 2).sum() + (SFEclass == 3).sum())
print (SFEclass.shape)
print (SFEdata.shape)

There are a total of 77 observations in the low SFE class.
There are a total of 209 observations in the medium SFE class.
There are a total of 88 observations in the high SFE class.
374
(374,)
(374, 14)
```

Figure 2: Classification of steel alloys into Low, Medium, or High SFE categories based on their respective SFE values.

reveals a non-linear correlation between SFE and these elements, indicating that the effect of increasing or decreasing either element is not straightforward. This suggests that the influence of their weight percentages (wt%) on SFE cannot be summarized into a simple rule, especially given the limited dataset used for training.

Next, the analysis explores how variations in wt% Cr and another alloying element impact SFE. Figure 4 also shows that no clear trend emerges from the interaction between Cr and other alloying elements, further confirming the complex, non-linear relationship between alloy composition and SFE. The data visualization in Figure 4 provides a limited view, as it examines only a few changing parameters at a time. To gain deeper insights into the relationship between SFE and alloy compositions, it's necessary to visualize the impact of varying all parameters simultaneously. However, with nine predictor variables (alloying elements) and SFE values resulting in a 10-dimensional dataset, spotting meaningful patterns becomes challenging. Dimensionality reduction techniques, such as Principal Component Analysis (PCA), are required to address this high dimensionality while retaining as much crucial information as possible.

PCA is used to compress the 10-dimensional dataset into a 2-D or 3-D solely for visual representation. To achieve this, two separate PCA transformations are applied to the normalized dataset. The first transformation reduces the data to 2 principal components, while the second transformation reduces it to 3 principal components. The 3-D PCA retains about 84.53% of the total variance in the data. The amount of variance explained by each principal component indicates that PC1 (52.33%) is the direction in the data that captures more than half the total variance in the dataset, while PC2 (24.18%) and PC3 (8.02%) explains the rest of the total variance. Additionally, the PCA loading scores for the three principal components are reported in Table 1, where features with higher absolute values contributes to loading scores defining the PC direction. For example, for the PC1 direction, the Mo, Cr, and Ni features are the three strongest positive contributors for this direction, while the Mn, Fe and C features have the three strongest negative influence.

Since the PCA performed in this project is solely for data visualization to identify patterns, this level of variance is sufficient for looking at the high-dimensional data in 2D and 3D.

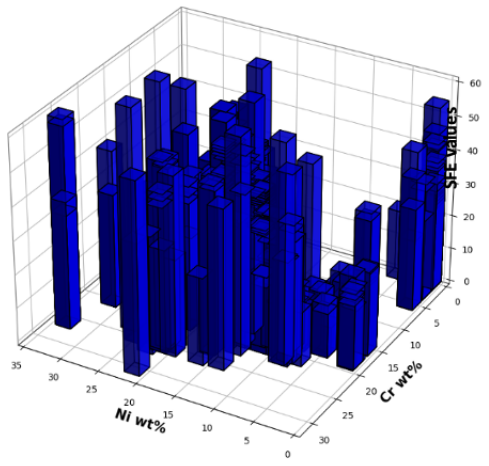

Figure 3: 3D bar plot showing the relationship between wt% Ni and wt% Cr on SFE.

Figure 5 presents the 3-D PCA transformation. This visualization doesn't reveal a clear pattern, as the SFE classes seem mixed without distinct clusters. To improve clarity, a 2-D visualization is generated, shown in Figure 6. This 2-D PCA plot begins to show a pattern, indicating regions where alloy compositions with low and high SFE are absent. Figures 6a and 6b demonstrate that when PC1 (Principal Component 1) is less than -0.2, most of the observations fall within the low and medium SFE classes, with only two outliers in the high SFE class. Similarly, when PC1 is greater than 0.1, most observations belong to the medium and high SFE classes, with two outliers in the high SFE class. These outliers could be due to uncertainties in SFE calculations. Despite these uncertainties, the PCA transformation provides a useful tool for visualizing high-dimensional data, offering insights into the relationships between alloy compositions and SFE values.

3 MACHINE LEARNING MODELS

In this study, seven machine learning models are used to classify the SFE regime (Low, Medium, and High) of steel alloys based on their chemical composition. The workflow involved systematic model development, hyperparameter tuning, cross-validation, and performance comparison. The dataset was split into 80% training and 20% test subsets using random sampling. Models were trained on the training set, and their performance was evaluated on the test set. A 5-fold cross-validation was utilized for hyperparameter tuning and performance validation across all models. A fixed random seed ensured reproducibility of data splits and shuffling procedures. Hyperparameter tuning was conducted using grid search, which systematically explored predefined combinations of parameters to identify those that maximized classification performance. For each model, the optimal hyperparameters and corresponding validation metrics are reported.

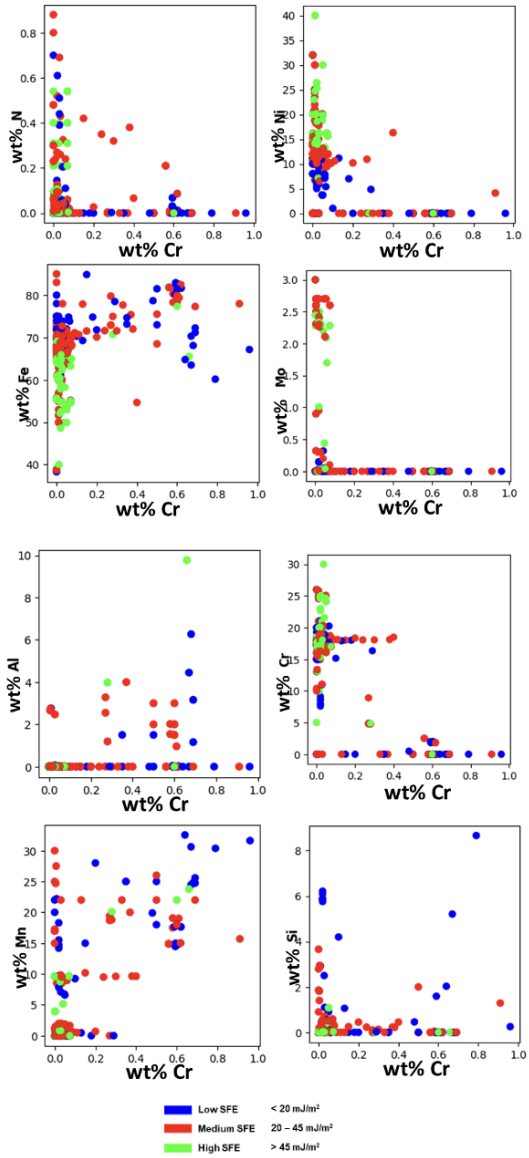

Figure 4: Variation of SFE with wt% Cr for different alloying elements.

3.1 Model Evaluation Metrics

Accuracy was utilized as the primary metric for evaluating model performance. Accuracy is defined as the ratio of correctly predicted instances to the total number of predictions. This provides a straightforward and interpretable measure of overall model correctness and is widely used for initial benchmarking of classification algorithms. It was computed for both the training and test datasets to assess the model's generalization performance and identify signs of potential overfitting or underfitting. While accuracy provides a high-level view of performance, it does not reveal class-specific prediction behavior. For example, in multi-class classification tasks

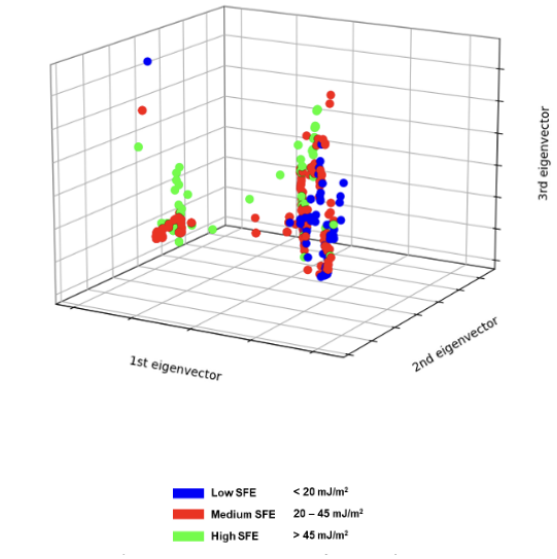

Figure 5: 3D PCA of SFE dataset.

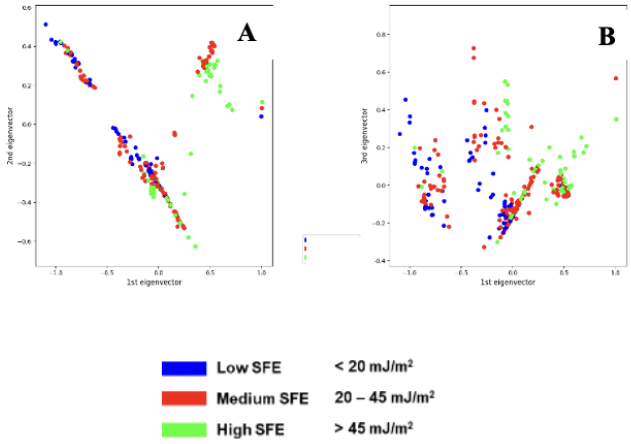

Figure 6: PCA of SFE dataset in 2D with top view (A) and front view (B).

such as predicting Low, Medium, or High SFE regimes, high accuracy can sometimes mask poor performance on minority classes. This limitation is particularly relevant when class distributions are imbalanced or when certain misclassification errors carry greater scientific or practical implications. In this context, further analysis using confusion matrices was conducted to diagnose class-specific prediction errors. The confusion matrix offers insight into how frequently the model misclassifies samples between different SFE categories, which is particularly important in understanding the reliability of predictions across the full range of SFE regimes. This class-level diagnostic can help to identify whether misclassification patterns were random or systematic, and whether specific boundaries were more prone to error. Altogether, accuracy and confusion matrix analysis support a more nuanced view of overall and class-specific model performance, allowing for a more informed comparison between different machine learning models.

3.2 Feed Forward Neural Network (FFNN)

FFNN is one of the simplest classes of artificial neural networks developed, whereby information flows in the forward direction from the input nodes through several hidden nodes, and to a final layer of output nodes. The neurons are interconnected by weights, which form probability-weighted associations between input and output layers through backpropagation. FFNNs can capture complex, non-linear relationships and elemental feature interactions intrinsic to SFE classification without extra feature engineering. Moreover, they possess a strong universal function-approximation capability.

Data preprocessing for building the FFNN model involves data standardization to ensure all features have a mean of 0 and a standard deviation of 1, leading to more stable training. The model architecture consisted of the following layers:

- Input layer: A dense layer with 64 nodes and ReLU activation, with the number of input dimensions matching the feature count (9 input variables) in the training dataset.
- Hidden layer: Another dense layer with 32 nodes, also using ReLU activation.
- Output layer: A dense layer with 3 nodes, employing softmax activation. 3 nodes were chosen because this is a multiclassification task with three possible outcomes (Low, Medium, High).

The model was compiled using the specified optimizer (from hyperparameter search), a sparse categorical cross-entropy, and using accuracy as a metric for model performance. The hyperparameter selection involved an optimization loop (grid search), iterating over different combinations of:

- Optimizers: Adam and RMSprop.
- Batch sizes: 16 and 32.
- Epochs: 10 and 20.

A stratified 5-fold cross-validation was used to ensure each fold's class distribution closely matched that of the whole dataset. This is useful in classification tasks since there's an imbalance in the distribution of SFE classes. Dropout layers were not added due to the small dataset as this could lead to underfitting, specifically when considering the limited amount of data points in each SFE class.

For each combination of hyperparameters, the dataset is divided into 5 stratified folds, and then a model is created, trained on the training folds, and evaluated on the validation folds. The best model hyperparameters are updated if the current combination yields higher accuracy. The best model, if found, is evaluated on the test dataset to determine its performance on unseen data. The hyperparameter selection iteration selected the rmsprop optimizer, batch size of 16, and 20 epochs. The FFNN model was evaluated to assess its performance on both the training and test datasets. The model achieved an accuracy score of 84% on the training set and 73% on the test set, suggesting some overfitting, where it performs less effectively on the unseen test data. A confusion matrix was generated to gain deeper insight into the model's predictive capabilities.

Table 1: PCA loading scores for the three principal components

PC	C	N	Ni	Al	Fe	Mo	Mn	Si	Cr
1	-0.32609088	-0.01520053	0.30024152	0.0758473	-0.20951403	0.61996516	-0.45110012	0.05278928	0.40373872
2	0.26705977	-0.08920191	-0.24654701	0.06424933	0.20177098	0.77477577	0.2646061	0.00906337	-0.3804223
3	0.02332149	0.62694074	0.02404889	0.04225843	-0.53491664	0.11385471	0.51021697	0.12927422	0.16704913

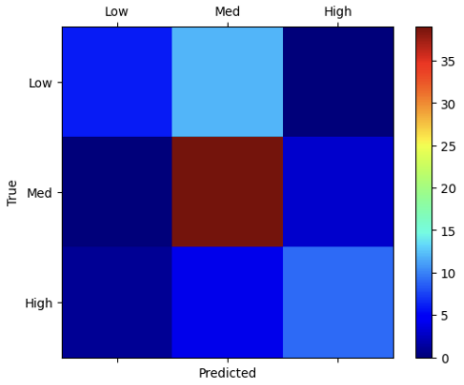

Figure 7: FFNN confusion matrix of validation set.

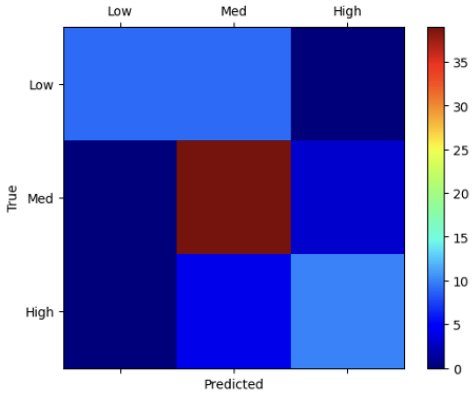

Figure 8: KNN confusion matrix of validation set.

As seen in the confusion matrix in Figure 7, the most amount of misclassification occurs between low and medium classes, with a total of 11 misclassifications here. Some misclassifications also occur between medium and high classes (4 misclassifications). Overall, the model performs the best in predicting the medium class, and this can be attributed to there being more data points for the medium SFE class in the training dataset.

3.3 K-Nearest Neighbors (KNN)

The KNN algorithm is a simple, non-parametric, instance-based learning method used for classification and regression tasks. In classification, KNN assigns a label to a new data point based on the majority class among its k closest neighbors in the feature space, where closeness is typically measured using Euclidean distance. Because KNN relies directly on the training data for making predictions, it does not require an explicit training phase. KNN is particularly useful for classifying SFE regimes because it captures local patterns in feature space without assuming an underlying functional relationship between composition and SFE class by majority vote. This approach allows the classification to reflect compositional similarity to known examples, which can be advantageous in complex alloy systems where SFE trends are not strictly linear. However, the effectiveness of KNN in this application depends on the appropriate choice of k and the feature scaling method, as the algorithm is sensitive to variations in feature magnitude and density of training points in the vicinity of decision boundaries.

A similar approach to data preprocessing for the FFNN was also implemented in the KNN. The KNN is a simple algorithm that does not have a traditional network architecture like other neural networks. The hyperparameter selection involved an optimization loop (grid search), iterating over different combinations of:

- Number of k neighbors: 3, 5, 7
- Weight function: uniform and distance

Following this iteration, the optimal model was found to be 7 number of k neighbors and a distance weight function. The model achieved an accuracy score of 96% on the training set and 78% on the test set. Similar to the FFNN, the most misclassifications occurred between the low and medium classes, with a total of 9, as shown in Figure 8.

3.4 Gradient Boosting Regressor (GBR), CatBoost Regressor (CAT), and Adaptive Boost Regressor (ADB)

GBR, CAT, and ADB regressor methods build predictive models by combining multiple weak learners (typically decision trees) in a sequential manner to improve overall accuracy. The GBR operates by iteratively minimizing a loss function using gradient descent, allowing it to capture complex nonlinear relationships between alloying elements and SFE. The CAT, an optimized variant of GBR, introduces efficient handling of categorical variables and robust regularization, which helps reduce overfitting and improves generalization, important for datasets with moderate size and subtle trends like in this case. ADB, on the other hand, emphasizes correcting errors made by previous learners by re-weighting samples, making it effective in refining predictions where initial models underperform. By applying these boosting techniques, the analysis aimed to leverage their high variance-reduction capacity and ability to model intricate feature interactions, which are critical for understanding how small compositional changes influence SFE across different alloy systems.

The hyperparameters tuned for training the GBR were the number of boosting iterations (100, 200, and 300), learning rate (0.01,

0.1, and 0.2), maximum depth (3, 4, and 5), which represent the depth of individual decision trees within the ensemble, and then the minimum number of samples required to split an internal node (2, 5, and 10). In the CAT model, the hyperparameters tested were the number of boosting iterations (500, 1000, and 1500), learning rate (0.01, 0.05, 0.1), and depth (4, 6, and 8). The ADB model was tested with a number of estimators (50, 100, and 200), learning rates (0.01, 0.1, and 0.2), and max depths (3,4, and 5).

The hyperparameters selected by the grid search are listed below:
GBR:max depth – 3; learning rate – 0.1; min samples split – 2; number of iterations – 100.
CAT: max depth - 4; Iterations – 500; Learning rate – 0.05.
ADB: max depth - 5; Iterations – 200; Learning rate – 0.2.

A stratified 5-fold cross-validation was used to train the GBR and ADB models. However, a simple 5-fold was used in training the CAT model because it does not support stratified KFold directly. All three models performed well on the training dataset by obtaining scores of 92% for the GBR, 92% for the CAT model, and 99% for the ADB. However, the CAT and GBR models performed poorly on the unseen test data by obtaining scores of 60% and 61% respectively. The ADB model performed best on the unseen test data, obtaining a score of 82%.

3.5 Support Vector Machines (SVM)

SVMs are powerful supervised learning algorithms that identify optimal hyperplanes to separate data classes (in classification) or fit a function within a margin of tolerance (in regression). For classification, Support Vector Classifiers (SVCs) aim to maximize the margin between compositional feature clusters corresponding to Low, Medium, and High SFE regimes, making them well-suited for cases with clear interclass boundaries but limited training data. The hyperparameters tuned to optimize the SVM include:

- C range: a range of 10 values logarithmically spaced between 2^{-2} to 2^{10}.
- Gamma range: This influences the decision boundary's flexibility, and a range of 5 values from 2^{-9} to 2^{1} was selected.
- Kernel: The kernel options tested included 'linear,' 'poly,' and 'radial basis function (RBF).'

A stratified 5-fold cross-validation was used on the training set to select the hyperparameters. The iteration process selected a poly kernel, a C value of 64, and a gamma value of 2.0. The SVM model achieved an accuracy score of 90% on the training set and 84% on the test set. The SVM yielded one of the best-performing models, likely due to its robustness in handling high-dimensional feature spaces and non-linear decision boundaries. Given the compositional complexity of steel alloys, SVM's ability to construct optimal separating hyperplanes using kernel functions allowed it to distinguish between SFE classes more effectively than the previous models, which relied on linear separability.

3.6 Random Forest (RF)

Random Forest (RF) is an ensemble learning method that constructs a collection of decision trees during training and outputs either the mode of their classifications (for classification tasks) or the average prediction (for regression tasks). Each tree is trained on a bootstrapped subset of the data, and splits are determined using a

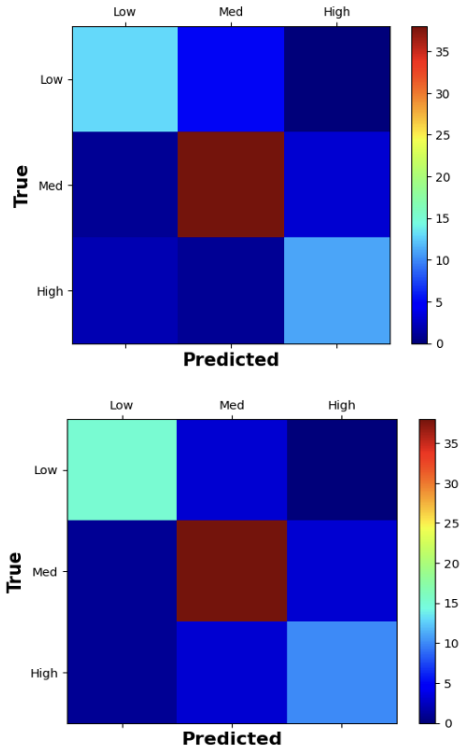

Figure 9: SVM (top) and RF (bottom) confusion matrix with validation set.

random subset of features, which decorrelates individual trees and enhances model generalization. In the context of this SFE analysis, RF was well-suited for capturing complex, nonlinear relationships between alloying elements and stacking fault energy regimes. The model's ability to capture implicit feature interactions and rank feature importance made it particularly effective for multivariate compositional inputs, where some elements can exert a pronounced yet not readily interpretable influence on SFE.

The hyperparameter iteration method was employed to test two key hyperparameters, encompassing the number of trees in the random forest (ranging from 10 to 50) and the maximum number of features considered for node splitting in each tree (ranging from 3 to 5). This resulted in 15 potential sets of hyperparameters to be tested, from which a combination of 40 trees and a maximum feature count of 4 was selected. To determine the optimal hyperparameters, a stratified 5-fold cross-validation was utilized. Impressively, a score of 99% on the training dataset and 85% on the test dataset was achieved by the RF model. The high performance observed for RF in this study likely stems from its ensemble nature, which reduced variance without increasing bias, enabling accurate classification even in the presence of noise or overlapping feature distributions.

Overall, in this study, the confusion matrix derived from the SVM and RF showed the lowest number of misclassifications, as seen in Figure 9, with a total of 5 for the SVM and 3 for the RF. A summary of all model scores is provided in Table 2. The results show that the RF model performed best in both the training and test

Table 2: Summary of model performance across all machine learning models

Model	Model Performance	
	Training data (%)	Testing data (%)
FFNN	84	73
KNN	96	78
SVM	90	84
RF	99	85
GBR	92	61
CAT	92	60
ADB	99	82

data among all the models investigated in this study. Similar better performance is also seen in the confusion matrix for the RF model as well. However, for the next stage of analysis to train the SFE classification model, the RF and SVM models are chosen for further analysis since they both performed best on the unseen test data and had the lowest number of misclassifications in the confusion matrix. Further analysis using the two top-performing models was done to investigate whether one model performed better than the other for the SFE classification tasks.

4 IMPROVING CONFIDENCE IN PREDICTIONS FOR SVM AND RF MODELS

Two machine learning models, SVM and RF, were selected based on their performance on the unseen test data for further analysis and development of a model for the SFE classification task. To reduce misclassifications, there is a requirement for huge amounts of evidence for the particular class the model predicts. To achieve this, a threshold probability is set such that when the model is unsure of an SFE class, it indicates it as fuzzy zones between two classes (low and medium or medium and high). Different threshold probabilities were tested, and after analysis, thresholds of 0.55 for the SVM model and 0.66 for the RF model were selected. These values were chosen because they provide an optimal balance between predictive accuracy and the proportion of fuzzy outputs. As the threshold probability increases, the number of fuzzy outputs also rises. For the SVM model, a threshold of 0.55 was selected, resulting in approximately 10% of the input being classified as fuzzy, with a predictive accuracy of around 88%. Similarly, the RF model's threshold of 0.66 classifies about 10% of the input as fuzzy, achieving a predictive accuracy of about 90%. Tables 3 and 4 provide a summary of the threshold probabilities and their outputs for the SVM and RF models, respectively.

Finally, the model now outputs five SFE classes instead of the original three (low, medium, and high). It introduces two additional classes: Fuzzy-LowMedium and Fuzzy-MediumHigh. For example, the Fuzzy-LowMedium class indicates an SFE value that could potentially fall between the low and medium categories.

5 FEATURE IMPORTANCE ANALYSIS

Austenitic stainless steels are primarily Fe-Cr-Ni alloys with low carbon content. These alloys may also include small amounts of other elements, such as Mn, N, Si, Mo, Al, Ti, and Nb, to impact

Table 3: Threshold probability based on the SVM model

Threshold probability	Predictive accuracy (%)	Fuzzy outputs (%)
0.5	86.49	0
0.55	87.84	9.46
0.66	90.54	22.97
0.7	91.89	38.37
0.8	91.89	51.35
0.9	94.59	82.43

Table 4: Threshold probability based on the RF model

Threshold probability	Predictive accuracy (%)	Fuzzy outputs (%)
0.5	85.14	0
0.6	85.14	1.35
0.66	89.19	9.46
0.7	90.54	14.86
0.8	90.54	25.68
0.9	95.95	50.0

specific properties that can alter the stacking fault energy (SFE) in a non-linear manner. Literature reports indicate that Ni and Fe have a moderate monotonic relationship with SFE [1, 12], while relationships between other elements can be complex. An example of this complex relationship is seen in work done by Vitos et al., where they reported that for Fe-Cr-Ni alloys with 14-16 % Ni, increasing Cr content decreases SFE, whereas for alloys with 17-19 % Cr, increasing Cr content increases SFE [11]. Furthermore, adding Mn to a Fe-Cr-Ni alloy decreases SFE at 0 K, but increases SFE at room temperature when Ni content exceeds 16 %.

This overall non-linearity of SFE values with respect to composition is evident, and therefore, it is important to gain insights into which elements are the key drivers of the model's predictions in this study. To achieve this, a feature importance plot is generated from the RF model, as seen in Fig. 10, which shows the rank of the elements according to how influential they are to the model's predictions. It is seen that Ni has the highest importance in the model prediction, and this finding aligns with what was reported by Wang [12] and Das [4].

SVM models do not inherently provide feature importance in the way the RF models do. Hence, the feature importance plot is only provided for the RF model.

6 VALIDATION OF IMPROVED MODELS

Random austenitic steel alloys, for which SFE calculations have been performed in literature, were used to test the validity of the improved SVM and RF models. Table 5 compares the predictions from the two models with the experimentally or theoretically determined SFE values for different austenitic alloys found in the literature. It is seen that both models perform very well by correctly predicting the SFE class corresponding to the reported experimental values. The RF model classified alloy 5 with a fuzzy zone between medium and high SFE classes. Notably, the reported experimentally

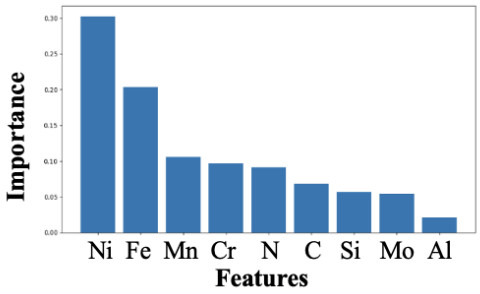

Figure 10: Feature importance plot from RF model.

determined SFE value (45 mJ/m^2 [6]) for this alloy lies directly on the border between the medium (20-45 mJ/m^2) and high SFE (>45 mJ/m^2) class, indicating that the model's prediction reflects the inherent uncertainty near SFE class boundaries rather than returning a misclassification. Similarly, the SVM model also classified alloy 5 as one belonging to a medium SFE class. A comparable case is observed for alloy 2, where the reported SFE value (20 mJ/m^2 [8]) calculated using CALPHAD also lies at the boundary between the low (<20 mJ/m^2) and medium SFE classes, and the two models' predictions differ slightly. Nonetheless, both models are largely in agreement, and neither demonstrates a clear performance advantage over the other. This suggests that using both models in tandem could enhance the reliability of SFE class predictions. An example of using both models in tandem is seen in alloy 3 (45 mJ/m^2 [8]) and alloy 7 (20 mJ/m^2 [13]), which also lie on SFE class boundaries. It is observed that both models agree in their predictions. Therefore, the deformation behavior can be expected to be that observed from the predicted SFE class. Furthermore, for alloys 1, 4, and 6, which are well-defined within SFE classes, it is seen that both models' predictions are accurate and in agreement. Overall, as the SFE dataset grows and more data becomes available for training, each model's accuracy is expected to improve further, potentially reducing the need for a second model as a validation step. In the context of practical applications, both models provide a rapid and efficient screening tool for predicting SFE classes across a wide compositional space, supporting accelerated alloy design. Since the SFE class, rather than the exact SFE value, is more important for predicting secondary deformation behavior, these models offer actionable insights. More importantly, even in compositions where there exists uncertainty in the SFE class, the models do not return outright misclassifications but instead indicate fuzzy zones that can be prioritized for more detailed experimental analysis. Overall, compared to prior approaches that rely solely on thermodynamic calculations or time-consuming experimental methods, both of which can exhibit significant scatter, the machine learning approach presented here provides a robust, reliable, and practical alternative for classifying SFE in austenitic steels.

7 CONCLUSIONS

This study developed machine learning models to classify steel alloys based on their stacking fault energy (SFE) and predict secondary deformation mechanisms. Among the models tested, Support Vector Machines (SVM) and Random Forest (RF) showed the highest accuracy. A fuzzy classification approach, using threshold probabilities, further improved prediction accuracy by accounting for uncertainties near SFE category boundaries. Validation against literature data confirmed strong agreement between the predictions of the models and reported SFE values. These results offer a robust framework for predicting SFE in austenitic steel alloys, aiding in the development of alloys with tailored mechanical properties.

The implementation of machine learning models on platforms like Jupyter Notebook provided an interactive and engaging environment for students. Through hands-on experience with data preprocessing, model training, and evaluation, students gained a deeper understanding of machine learning algorithms and their applications to materials science. The project improved analytical skills and offered practical insights into the potential of machine learning for materials design, enhancing both technical knowledge and problem-solving abilities.

8 STUDENT REFLECTION AND EDUCATIONAL IMPACT

This project provided a meaningful learning experience by applying machine learning to an important materials science problem using an existing SFE dataset. One challenge encountered was understanding the limitations and variability within the dataset, which was compiled from diverse literature sources with varying experimental conditions and methods. Another difficulty was in addressing the non-linear and high-dimensional relationship between alloy composition and SFE by utilizing fuzzy zones to represent regions of uncertainty in predictions. Through this project, the student gained practical skills in data preprocessing, feature selection analysis, dimensionality reduction, data interpretation, implementation, and tuning different machine learning models. For future students undertaking similar projects, we recommend paying close attention to documenting their workflow, including code, parameter choices, and rationale for decisions, to make their work more reproducible and easier to troubleshoot. We also advise students to critically evaluate model output rather than focusing mainly on accuracy, and to make good use of the wealth of available data preprocessing, interpretation, and visualization tools to communicate their results. Furthermore, we advise students to incorporate additional evaluation metrics such as precision, recall, or F1-scores into such complex classification tasks, as this will provide a more complete understanding, especially near class decision boundaries.

ACKNOWLEDGEMENTS

K.N. is partially supported by the NSF grant OAC-2118061.

REFERENCES

[1] Gaurav Arora and Dilpuneet Singh Aidhy. 2020. Machine learning enabled prediction of stacking fault energies in concentrated alloys. *Metals* 10, 8 (2020). https://doi.org/10.3390/met10081072

[2] Jaime A. Castañeda, Oscar A. Zambrano, Germán A. Alcázar, Sara A. Rodríguez, and John J. Coronado. 2021. Stacking Fault Energy Determination in Fe-Mn-Al-C

Table 5: Summary of SVM and RF model prediction compared to SFE values reported in literature

S/N	Alloy composition	Reported SFE (mJ/m^2)	Model Prediction	
			SVM	RF
1	$Fe_{0.03}Mn_{0.02}Si_{0.004}C_{15.6}Ni_{17.5}Cr_{2.5}Mo$	49.6 [14]	H	H
2	$Fe_{16}Cr_{13}Ni$	20 [8]	M	L
3	$Fe_{20}Ni_{25}Cr$	45 [8]	M	M
4	$Fe_{15.9}Cr_{12.5}Ni$	24 [6]	M	M
5	$Fe_{17.8}Cr_{14.1}Ni$	45 [6]	M	Fuzzy M-H
6	$Fe_{18}Cr_{10}Ni_{0.2}N$	23 [13]	M	M
7	$Fe_{18}Cr_{10}Ni_8Mn_{0.4}N$	20 [13]	M	M

Austenitic Steels by X-ray Diffraction. *Metals* 11, 11 (2021). https://doi.org/10.3390/met11111701

[3] N. Chaudhary, A. Abu-Odeh, I. Karaman, and R. Arróyave. 2017. A data-driven machine learning approach to predicting stacking faulting energy in austenitic steels. *Journal of Materials Science* 52, 18 (2017). https://doi.org/10.1007/s10853-017-1252-x

[4] Arpan Das. 2016. Revisiting Stacking Fault Energy of Steels. *Metallurgical Transactions A* 47 (2016). https://doi.org/10.1007/BF02646563

[5] Arpan Das, Soumitra Tarafder, and Pravash Chandra Chakraborti. 2011. Estimation of deformation induced martensite in austenitic stainless steels. *Materials Science and Engineering: A* 529 (2011), 9–20. https://doi.org/10.1016/j.msea.2011.08.039

[6] F. Lecroisey and B. Thomas. 1970. On the variation of the intrinsic stacking fault energy with temperature in Fe-18 Cr-12 Ni alloys. *physica status solidi (a)* 2, 4 (1970). https://doi.org/10.1002/pssa.19700020429

[7] Jun Lu, Lars Hultman, Erik Holmström, Karin H. Antonsson, Mikael Grehk, Wei Li, Levente Vitos, and Ardeshir Golpayegani. 2016. Stacking fault energies in austenitic stainless steels. *Acta Materialia* 111 (2016), 39–46. https://doi.org/10.1016/j.actamat.2016.03.042

[8] A.P. Miodownik. 1978. The calculation of stacking fault energies in Fe-Ni-Cr alloys. *Calphad* 2, 3 (1978), 207–226. https://doi.org/10.1016/0364-5916(78)90010-X

[9] Adriana Estela Pontini and Jorge Daniel Hermida. 1997. X-Ray diffraction measurement of the stacking fault energy reduction induced by hydrogen in an AISI 304 steel. *Scripta Materialia* 37, 11 (1997), 1831–1837. https://doi.org/10.1016/S1359-6462(97)00332-1

[10] Cecil G. Rhodes and Anthony W. Thompson. 1977. The composition dependence of stacking fault energy in austenitic stainless steels. *Metallurgical Transactions A* 8 (1977). https://doi.org/10.1007/BF02646563

[11] L. Vitos, J.-O. Nilsson, and B. Johansson. 2006. Alloying effects on the stacking fault energy in austenitic stainless steels from first-principles theory. *Acta Materialia* 54, 14 (2006), 3821–3826. https://doi.org/10.1016/j.actamat.2006.04.013

[12] Xin Wang and Wei Xiong. 2020. Stacking fault energy prediction for austenitic steels: Thermodynamic modeling vs. machine learning. *Science and Technology of Advanced Materials* 21 (2020). https://doi.org/10.1080/14686996.1808433

[13] I.A. Yakubtsov, A. Ariapour, and D.D. Perovic. 1999. Effect of nitrogen on stacking fault energy of f.c.c. iron-based alloys. *Acta Materialia* 47, 4 (1999), 1271–1279. https://doi.org/10.1016/S1359-6454(98)00419-4

[14] Toshio Yonezawa, Ken Suzuki, Suguru Ooki, and Atsushi Hashimoto. 2013. The effect of chemical composition and heat treatment conditions on stacking fault energy for Fe-Cr-Ni austenitic stainless steel. *Metallurgical and Materials Transactions A* 44 (2013). https://doi.org/10.1007/s11661-013-1943-0