CS 229 Final Report: Data-Driven Prediction of Band Gap of Materials

Fariah Hayee and Isha Datye
Department of Electrical Engineering, Stanford University
Rahul Kini
Department of Material Science and Engineering, Stanford University

1. INTRODUCTION

Advancements in material chemistry have remained one of the biggest challenges of the 21^{st} century. Many material discoveries have been the result of purely experimental inference and trial-and-error [Jain et al. 2013]. Accurate prediction of material properties using computational methods like density functional theory (DFT) or Molecular Dynamics (MD) have limitations of being computationally expensive [Ma and Wang-Wang 2016] and are often size-limited.

In recent years, there has been a growing interest in developing machine learning based high-throughput predictive algorithms [Kusne et al. 2015]. Most notably, Gaultois et al. showed that Random Forest ensemble methods with leave one out cross validation (LOOCV) can accurately predict the electrical conductivities and Seebeck coefficients of thermoelectric materials that are different from those commonly found in literature [Gaultois et al. 2016]. Recent studies have used machine learning algorithms to predict properties of materials such as melting temperature, mechanical properties of alloys, and dielectric properties for various applications [Ward et al. 2016]. Ward et al. implemented a Random Forest method after partitioning the dataset into sets of similar materials (based on range of band gaps, or whether a compound contains a halogen, chalcogen or pnictogen). Though prediction accuracy increased by five times, the model assumes materials belong to these pre-defined partitioned subsets, thereby sacrificing robustness.

Other machine learning models have focused on predicting band gaps for solar cell devices where a precise band gap is necessary for both tandem and single junction performance [Dey 2014]. One such machine learning model implemented a linear least squares regression for feature selection compounded with kernel ridge regression (KRR)–a method capable of handling complex non-linear relationships between features–to predict band gaps of double perovskites [Pilania et al. 2015]. Although the model has reported training and test set r^2 scores of 0.997 and 0.937, respectively, this model can only be applied to a limited chemical space for non-magnetic perovskites.

A model to predict important electronic properties like band gap can help guide the search for materials for perovskite and tandem solar cells, heterojunction optical devices, and many other fields. Machine learning based approaches may be used to find a correlation between certain material properties and attributes, which can help direct new material discovery [Gaultois et al. 2016; Meredig et al. 2014]. With this goal, we apply machine learning techniques to predict band gaps of compound materials. To represent the materials in feature space, we follow a chemical composition based approach similar to [Meredig et al. 2014], where features are derived from atomistic, electronic, and structural properties of the constituent elements. Based on a dataset listing previous DFT-calculated band gaps, we evaluate the performance of several regression techniques including ordinary least squares (OLS) linear regression, Ridge, Lasso, Random Forest, Ada Boosting, and Multilayer Perceptron. We also focus on finding significant features and attempt to explain their correlation to band gap based on the underlying physics and chemistry.

This report presents the summary of our progress in applying machine learning techniques to predict the band gaps of materials. In Sec. 2, we discuss the database and important features. In Sec. 3, various machine learning algorithms that we have used are discussed. In Sec. 4, we discuss feature ranking as well as results obtained from applying the different models to our dataset. Finally, in Sec. 5, we discuss our overall conclusions and ideas for future work.

2. DATABASE AND FEATURE SELECTION

The data set, provided to us by Citrine Informatics, is based on DFT-calculated band gaps and structural information collected from Materials Project [Jain et al. 2013]. The database contains 2067 instances of materials listed with band gaps ranging from 24 meV to 11.5 eV. We use a set of 97 attributes which are derived from chemical composition and statistical average over the constituent elements' properties [Meredig et al. 2014; Ward et al. 2016]. The properties can be grouped as:

(1) **Stoichiometric attributes** depend on the fractions of elements present. They include: stoichiometric average over atomic weight, atomic volume, density, electronegativity, covalent radii, melting temperature, ionization potential, row and column numbers in the periodic table, number of free electrons in the s, p, d, and f orbitals, etc. To calculate the stoichiometric average of property M for compound $A_xB_yC_z$, we use:

$$\mathrm{M}^{\mathrm{avg}}_{\mathrm{A_xB_yC_z}} = \frac{\mathrm{x}}{\mathrm{x+y+z}} * \mathrm{M_A} + \frac{\mathrm{y}}{\mathrm{x+y+z}} * \mathrm{M_B} + \frac{\mathrm{z}}{\mathrm{x+y+z}} * \mathrm{M_C} \quad (1)$$

(2) **Max and Min attributes** include the maximum and minimum of all the attributes listed above. As an example, the maximum and minimum of the number of free electrons in the s, p, d, and f orbitals and row and column numbers are in this set.

(3) **Electronic structure attributes** include sparse attributes which define if one of the constituent elements is in the d or

f block, or a metal or metalloid, etc. They also include the average fraction of s, p, d, or f electrons in the valence shell.

2.1 Hand-crafting Complex Features

Based on our understanding of material properties, we design features which can be used to model electronic properties. A few of them are listed below:

—Mapping electron numbers to *effective free valence electron* numbers: If the element has a d-band electron number N_d, it is mapped to $(10 - N_d), \text{if} N_d > 5$ to express the number of available free electrons. Similarly for p, N_p is mapped to $6 - N_p$ when $N_p > 3$.

—Difference in p and d electron probabilities in the valence shell: We calculate the 'p (d)-character' of the compound by multiplying the maximum p (d)-electron number by the stoichiometric average of the percent of p (d)-electron. Then, the square of the difference is calculated as:

$$\Delta P_{\text{p-d}} = max\left\{0, (\%p * Np_{max} - \%d * Nd_{max})^2\right\} \quad (2)$$

2.2 Preprocessing

Some of our features are sparse and have very low variance. We exclude features which have variance less than a certain threshold. The boolean feature threshold is defined as $\text{Var}\,[X] = p(1-p)$ and we exclude features where p = 0.98. We also use standard scaling to force a mean of zero and scale the data to unity variance, since features with variance orders of magnitude larger than the others may dominate the learning process and inhibit learning from the smaller variance features.

3. METHODS IN MACHINE LEARNING

The methods utilized in this project are ordinary least squares (OLS) linear regression, Ridge, Lasso, Random Forest, Ada Boosting and Multilayer Perceptron. Our project employs packages in Python, mainly scikit-learn [Pedregosa et al. 2011].

3.1 Ordinary Least Squares Regression

Ordinary least squares (OLS) linear regression calculates the fitting parameter θ using Eq. 3, where X is the matrix of training examples and y is the output of band gap values.

$$\theta = \text{argmin}||X\theta - y||_2^2 \quad (3)$$

To find significant features, we use forward selection with 10-fold cross-validation (CV) on the entire $M = 97$ feature set. Our code starts with finding the first feature that minimizes the 10-fold CV mean squared error (MSE). For the subsequent kth iteration, it finds the next best feature minimizing CV MSE from the remaining $M - k$ feature set. This code also allows us to determine the optimal number of features to balance bias-variance trade-off.

3.2 Ridge and Lasso Regression

Ridge and Lasso (Least Absolute Shrinkage and Selection Operator) regression place a penalty on the size of the coefficients θ and attempt to reduce the variance of the estimates. The equation used to calculate the coefficients θ in Ridge, shown in Eq. 4, includes a regularization term using α given by the L2 norm. The θ in Lasso, shown in Eq. 5, includes a regularization term given by the L1 norm. X is the matrix of training examples, y is the output of band gap values, and $n_{samples}$ is the number of training samples.

$$\theta = \text{argmin}||X\theta - y||_2^2 + \alpha||\theta||_2^2 \quad (4)$$

$$\theta = \text{argmin}\frac{1}{2n_{samples}}||X\theta - y||_2^2 + \alpha||\theta||_1 \quad (5)$$

The term α is a tuning parameter that controls the size of the coefficients, so as α increases, the coefficients tend toward zero and become more robust to collinearity. When α = 0, Ridge and Lasso reduce to linear regression. Ridge regression is useful for ill-conditioned matrices, since a minor change in the target variable can lead to a significant change in the coefficients. Lasso forces some coefficients to be set to zero, and thus prefers solutions with fewer parameter values. Bias tends to increase with α, but variance typically decreases [Tibshirani 1996; Pedregosa et al. 2011].

3.3 Random Forest

Random Forest is an ensemble method that fits classifying decision trees on subsets of a dataset and uses averaging over the trees to improve accuracy of the predictions and reduce over-fitting. Each tree is constructed with a random vector Θ_k and training samples drawn with replacement (bootstrapping), forming a classifier $h(x, \Theta_k)$, where x is an input vector. Sampling with replacement decreases the variance of the model without increasing bias. When splitting a node in the tree, a random subset of the features is used to create the best split. The bias of the whole forest is usually higher than the bias of a single tree because of this randomness, but averaging over the trees tends to reduce variance and thus provides a good model. Typically, increasing the number of trees results in higher bias and lower variance but also takes longer to compute [Breiman 2001; Pedregosa et al. 2011].

3.4 Ada Boosting

In Adaptive (Ada) Boosting, weak learners are iteratively trained on weighted versions of the dataset, with incorrectly predicted samples weighted higher than correctly predicted samples for each iteration. Therefore, the weak learners focus more on the more difficult samples that previous weak learners incorrectly predicted. All predictions are combined through a weighted average to yield the final result. This method tends to reduce bias more than variance. Unlike Random Forest which fits separate decision trees to multiple copies of the original data and combines these trees into a single predictive model, Ada Boosting grows each weak classifier sequentially. Boosting learns slowly from the residual error of fitting the model to the data [Schapire 2013; Pedregosa et al. 2011].

3.5 Multilayer Perceptron (MLP)

Multilayer perceptron is a supervised learning method which uses multiple layers of nodes, with each layer fully connected to the next. The first layer is the input layer, consisting of a set of neurons $\{x_i|x_1, x_2, ..., x_m\}$ representing the input features. There can be any number of hidden layers, where each neuron transforms the values of the previous layer by a weighted summation

$w_1x_1 + w_2x_2 + ... + w_mx_m$ followed by a non-linear activation function $g(\cdot) : R \rightarrow R$. The output layer receives values from the last hidden layer and, based upon the type of problem (regression or classification), it can have one or multiple nodes.

MLP learns through backpropagation i.e, learning occurs by changing the connection weights after each piece of data is processed, based on the amount of error in the output compared to the expected result. Let's consider the accumulation of the weighted values of a neuron as $\text{net}_{pj} = \sum_i w_{ij}o_{pi}$,where, w_{ij} is the weight connecting the ith neuron in a given layer, to the jth neuron in the subsequent layer and o_{pi} is the input for the pth data point at ith node. The change in weights can be written as:

$$\Delta_p w_{ij} = \eta\delta_{pj}o_{pi}, \tag{6}$$

where, η is the learning rate and δ_{pj} can have two definitions based on the jth node. If j is an output node,

$$\delta_{pi} = \epsilon_{pj}f_j'(\text{net}_{pj}), \tag{7}$$

where ϵ_{pj} is the error and if unit j is a hidden layer, then

$$\delta_{pi} = f_j'(\text{net}_{pj})\sum_k \delta_{pk}w_{kj}. \tag{8}$$

The hidden layer weights are determined by changing the output layer weights according to the derivative of the activation function, and thus this algorithm represents a backpropagation of the activation function [Murtagh 1991].

4. RESULTS AND DISCUSSION

In this section, we discuss the results of all methods described in the previous section. For each method, we perform 10-fold cross-validation for varying numbers of features. We determined that the optimal number of features that resulted in the lowest CV MSE for all methods was 40 features. After optimizing the parameters of the model using 10-fold CV with 40 features, we randomly split the data into a train and test set with a ratio of 90:10 (1860 train:207 test samples) and report the test MSE for all models in consideration. We compare the training MSE, test MSE, and CV MSE values to determine which model is most effective at predicting material band gaps.

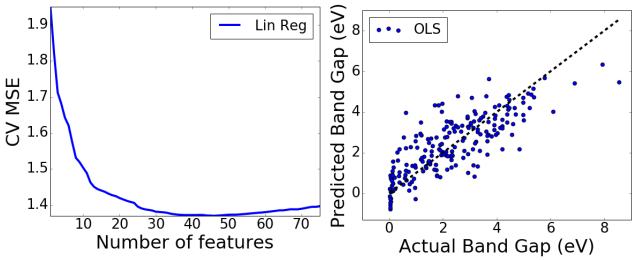
4.1 Feature Ranking

With forward selection, we ranked the features based upon their significance. We implemented OLS linear regression and plotted the 10-fold CV mean squared error (MSE) vs. number of features, shown in Fig. 1(a). From the figure, we determine that the optimal number of features to minimize CV MSE is between 40 and 50, which gives an MSE of $\sim$1.37. Hence we chose 40 features to compare our linear regression model performances in the next section.

From forward selection, we find the five most important features as: electronegativity difference (ΔE.N), fraction of f-electron *max number of f-electron ($f * N_f$), fraction of p-electron, stoichiometric average of covalent radius and squared difference of p and d electron ($\Delta\text{P}_{\text{p-d}}$) defined in 2.1.

Analyzing the fitting parameter (θ values), we find that band gap is proportional to ΔE.N and fraction of p-electron. ΔE.N is related to ionic bond formation and compound molecule size and a larger difference indicates a higher probability of forming ionic bonds. As electronegativity of the compound increases, the transition of the valence shell electrons to conduction band become more difficult and thus band gap increases. Fraction of p-electron indicates the probability of forming a π-bond versus a σ-bond. Having more of a p-characteristic in the valence band thus results in higher band gap. Band gap is also positively correlated to $f * N_f$, which indicates that having any Lanthanide element increases band gap.

On the other hand, band gap is negatively correlated to the stoichiometric average of covalent radius indicating that the larger the compound molecule size, the easier it is for the valence electrons to make a transition to the conduction band. Band gap is also negatively correlated to higher density and higher atomic weight atoms in the compound, which means that, in general, larger atoms reduce the band gap. Having a d or f block metal reduces band gap as it is negatively dependent on 'Is D block max' and maximum number of f-shell valence electrons attributes. Finally, we notice that band gap is positively correlated to $\Delta\text{P}_{\text{p-d}}$ which can be interpreted as the electron density difference between the p-orbital and the d-orbital.

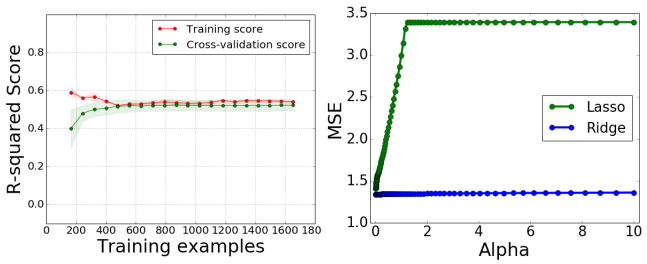


(a) (b)

Fig. 1: (a) Plot of 10-fold CV error vs. number of features using OLS. (b) OLS prediction of band gaps plotted against DFT-calculated band gaps.

4.2 Comparison of Linear Regression Models

We implemented OLS, Ridge, and Lasso linear regression models using 40 features and evaluated the models by randomly splitting the dataset into train and test sets with a ratio of 90:10. The plot of predicted band gap vs. actual band gap for the test data with OLS is shown in Fig. 1(b). The training data had an MSE of 1.34 with an r^2 score of 0.61, while the test data had an MSE of 1.05 and an r^2 score of 0.63 (shown in Table I). The learning curve in Fig. 2(a) shows that both the training and the test scores converge to low values and do not change with increasing the number of samples beyond 600, indicating high bias in the OLS model. Due to the high bias in OLS, the regularization in Ridge and Lasso fail to improve the fitting. As shown in Fig. 2(b), increasing alpha from 0 to 10 increases the training MSE when using Ridge and Lasso. Therefore, Ridge and Lasso regression do not yield better results than OLS. Analyzing the fitting for OLS, we see that linear regression under-predicts high band gaps and over-predicts very small band gaps.



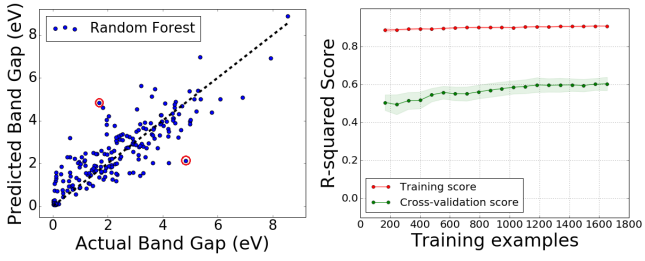
(a) (b)

Fig. 2: (a) Learning curve for OLS using 40 features. (b) The training MSE for Lasso and Ridge regression for varying values of α.

4.3 Random Forest and Ada Boosting

As OLS has very high bias in this model, we chose Random Forest for this problem. For 40 features and with optimized parameters, the CV MSE is 1.13 (see Table I) for this method. The criterion used for training this method was MSE, and the most important parameters used in this method are number of trees, maximum depth of trees, and number of features used in each split. The optimal number of trees and depth of trees were $\sim$300 and $\sim$25, respectively, since any number higher or lower than these either resulted in increased or the same MSE. At each split, the total number of features were used as it resulted in the lowest CV MSE.

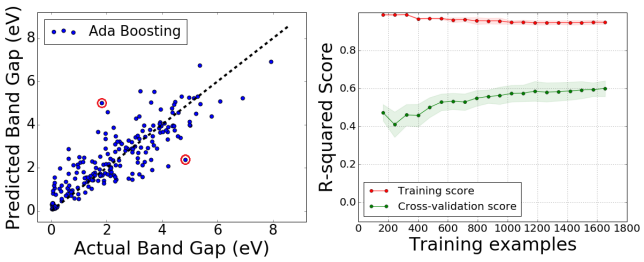
Fig. 3(a) shows the predicted band gap vs. actual band gap for the test dataset using Random Forest, with an MSE of 0.76 and r^2 score of 0.73. The training dataset had an MSE of 0.22 and r^2 score of 0.93 (see Table I). The test MSE was much larger than the train MSE, suggesting that this model over-fits the data and has high variance. This is further demonstrated by the learning curve for Random Forest, shown in Fig. 3(b). If we had more training examples, we might be able to increase the CV score. We tried changing the parameters of the model to reduce over-fitting, but reducing the number of trees, the depth of trees, or the number of features used in each split typically increased the CV MSE.



(a) (b)

Fig. 3: (a) Random Forest prediction of band gaps plotted against actual band gaps. (b) Learning curve for Random Forest using 40 features.

We also performed CV of Ada Boosting using 40 features and obtained an optimal CV MSE of $\sim$1.14. The main parameters we tweaked in Ada Boosting were the estimator (weak learner), the number of estimators used, and the learning rate. The estimator we used in the Ada Boosting method was Decision Tree Regressor. The inputs of the decision tree are similar to those of Random Forest. For this method, we selected MSE as the criterion, the maximum number of features (40) for each split, and a maximum tree depth of $\sim$40. We ran Ada Boosting using $\sim$90 estimators, meaning that the method iterated over the weak learners 90 times. The learning rate is the factor by which the contribution of each regressor is reduced. After experimenting with different values of the learning rate, we chose 0.9, since this gave us the best bias-variance tradeoff.



(a) (b)

Fig. 4: (a) Ada Boosting prediction of band gaps plotted against actual band gaps. (b) Learning curve for Ada Boosting using 40 features.

Fig. 4(a) shows the plot of predicted band gap vs. actual band gap for Ada Boosting. The training samples had an MSE of 0.02 with an r^2 score of 0.99, and the test samples had an MSE of 0.72 with an r^2 score of 0.74. Since the model was fitting the training dataset almost perfectly but the test MSE was much higher, Ada Boosting was over-fitting our dataset. The learning curve, shown in Fig. 4(b) shows a decrease in training score and an increase in CV score as the number of training examples increases. Therefore, to reduce variance further, a larger number of training examples is necessary.

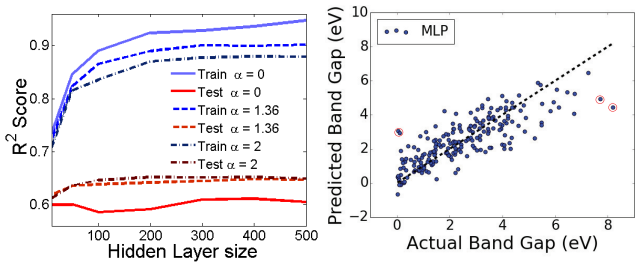
As with Random Forest, we experimented with the parameters of the decision tree estimator, the number of estimators used, and the learning rate, but this typically resulted in higher CV MSE and therefore a worse model. Ada Boosting performed very similarly to Random Forest; they had very similar CV MSE, test MSE, and test r^2 scores, but the training MSE and r^2 scores were quite different. Therefore, we conclude that the variance of Ada Boosting is larger than that of Random Forest.

To determine why Random Forest and Ada Boosting incorrectly predict the band gaps of certain materials, we look at the outliers circled in red in Fig. 3(a) and Fig. 4(a). For example, both Random Forest and Ada Boosting incorrectly predicted the band gaps of the materials $SbH(OF3)_2$ and $Ti(AlBr_4)_2$ by more than 2.3 eV. When examining the attributes of these compounds, we observe that the electronegativity difference (ΔE.N) and covalent radius do not follow the expected trends as discussed earlier. $SbH(OF3)_2$ has a band gap of 1.83 eV, ΔE.N of 1.93, and covalent radius of 64.4, while $Ti(AlBr_4)_2$ has a band gap of 4.84 eV, ΔE.N of 1.42, and covalent radius of 124. We expect a positive correlation between band gap and ΔE.N and a negative correlation between band gap and covalent radius, but we see the opposite trend with these materials. These trends may cause our models to incorrectly predict the band gaps of these materials as well as others.

4.4 Multilayer Perceptron (MLP)

As electronic properties like band gap are derived from atomistic properties through a very complex interdependence, we employ a supervised neural network using MLPregressor in scikit learn to model the band gap. We optimize the number of hidden layers, learning rate, regularization parameter and activation potential through a grid-search algorithm while using stochastic gradient descent (sgd) as the solver. As a means of optimization, for 40 features extracted by principle component analysis (PCA) from the whole feature set, in Fig. 5(a) we plot the mean training and test scores (from the grid-search algorithm) for 10-fold cross-validation with varying hidden layer numbers for different regularization (α) values. Figure 5(a) shows that the training score increases with more number of layers, however the test score does not improve much after α reaches 1.36. To balance between bias and variance, we chose the hidden layer number as 200 and α as 1.36.

In Fig. 5(b), we have plotted the predicted band gaps vs. DFT-calculated band gaps. For the randomly split train and test sets used to evaluate the other models, the train and test scores for MLP are 0.76 and 0.70, respectively, and the MSE is 0.83 for both the train and test sets, suggesting that this model has lower variance than the ensemble methods. The optimal CV MSE was 1.13. The MLP fails



(a) (b)

Fig. 5: Neural Networks (a) Dependence of train and test scores on regularization parameter α and hidden layer size (b) Prediction of band gaps plotted against DFT-calculated band gaps using MLP.

at some outliers as shown in Fig. 5(b). Running the code multiple times and observing the highest error points as marked in Fig. 5, we notice that this model often overestimates very small band gaps (< 1 eV) when the atomic volume of the molecule is exceptionally big. In this case, the model overestimates $Ba_3ZnIn_2O_9$ and $BaNbO_3$ band gaps to $\sim$3 eV where the actual band gaps are 3 and 10 meV, respectively. Both of the compounds have exceptionally high atomic volume because of In and Nb, compared to other small band gap materials. On the other hand, the model underestimates high band gaps when the compound contains f-block materials but has high band gap.

We note that the MSE from taking an average of all the band gaps in our dataset is 3.34. The MSE for the test set we used when testing all methods is 2.81. Therefore, all of our models perform much better than random guessing.

Method	Training Data		Test Data		CV MSE
	MSE	Score (r^2)	MSE	Score (r^2)	
OLS	1.34	0.61	1.05	0.63	1.37
RF	0.22	0.93	0.76	0.73	1.13
Ada	0.02	0.99	0.72	0.74	1.14
MLP	0.83	0.76	0.83	0.70	1.13

Table I. : Summary of Results: CV MSE as well as MSE and r^2 score for different machine learning methods using the same test and train datasets.

5. CONCLUSION AND FUTURE WORK

We have presented fitting models to predict material band gaps using OLS, Ridge, Lasso, Random Forest, Ada Boosting, and Multilayer Perceptron. Using the forward search method, we have determined the optimal number and combination of features to balance bias and variance of the models as well as the most significant features, which we correlated with underlying physical understanding. The features with low variance were removed, and all features were normalized to have a mean zero and unity variance. As OLS has high bias, regularization algorithms like Ridge and Lasso show no improvement over OLS. To reduce error due to bias, ensemble methods like Random Forest and Ada Boosting, as well as MLP are compared. MLP, Ada Boosting, and Random Forest have very similar CV and test MSE values and test scores. However, Ada Boosting and Random Forest over-fit the data and have high variance. MLP performs comparably to Ada Boosting and Random Forest but has lower variance. All models presented perform much better than random guessing.

Since band gap is a complex function of atomic, structural, and electronic properties, we want to train a neural network to partition the dataset into k groups of similar materials and then use each subset for training with regression and ensemble methods. We could also try training a convolutional neural network or a deep learning network on a much larger database, such as Materials Project or Open Quantum Materials Database as our best models have high error due to variance. These additional methods may give lower error and better prediction of material band gaps.

ACKNOWLEDGMENTS

We are grateful to the following people for resources, discussions and suggestions: Bryce Meredig and the Citrine Informatics team who provided us with the material datasets necessary for this project, and the teaching assistants of CS 229 who helped guide the selection of appropriate algorithms for this project.

REFERENCES

L. Breiman. 2001. Random forests. *Machine Learning* 45, 5-32 (2001).

P. Dey. 2014. Informatics-aided bandgap engineering for solar materials. *Computation Materials Science* 83, 185-195 (2014).

M. Gaultois, A. Oliynyk, T. Sparks, G. Mullholand, and B. Meredig. 2016. Perspective: Web-based machine learning models for real-time screening of thermoelectric materials properties. *APL Materials* 4, 053213 (2016).

A. Jain, S. Ong, G. Hautier, W. Chen, W. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. Persson. 2013. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials* 1, 011002 (2013).

A. G. Kusne, D. Keller, A. Anderson, A. Zaban, and I. Takeuchi. 2015. High-throughput determination of structural phase diagram and constituent phases using GRENDEL. *Nanotechnology* 26, 444002 (2015).

J. Ma and L. Wang-Wang. 2016. Using Wannier functions to improve solid band gap predictions in density functional theory. *Nature* 6, 24924 (2016).

B. Meredig, A. Agrawal, S. Kirklin, J. Saal, J. Doak, A. Thompson, K. Zhang, A. Choudhary, and C. Wolverton. 2014. Combinatorial screening for new materials in unconstrained composition space with machine learning. *Physical Review B* 89, 094104 (2014).

F. Murtagh. 1991. Multilayer perceptrons for classification and regression. *Neurocomputing* 2, 183–197 (1991).

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 2825–2830 (2011).

A. Pilania, A. Mannodi, B. Uberuaga, R. Ramprasad, J. Gubernatis, and T. Lookman. 2015. Machine learning bandgaps of double perovskites. *Nature Scientific Reports* 6, 5–32 (2015).

R. Schapire. 2013. Empirical Inference- Explaining Adaboost. *Springer* 37–52 (2013).

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

L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton. 2016. A general-purpose machine learning framework for predicting properties of inorganic materials. *NPF Computational Materials* 2, 16028 (2016).