Investigate Machine Learning Methods for Transparent Conductors Prediction

Yan Sun yas108@eng.ucsd.edu Yiyuan Xing yix016@eng.ucsd.edu
Xufan Xiong x7xiong@eng.ucsd.edu Tianduo Hao t6hao@eng.ucsd.edu

***Abstract*— In this project, we tried to use machine learning method for predicting some important features of transparent conductors, formation energy and bandgap energy, which affect the basic functions of the material. What we have are some basic atom information like element compositions of the 3000 conductors. We tried some different machine learning models, including linear regression, artificial neural network, adaboost and random forest to accomplish the task. After testing our models, we find that among these methods, random forest shows the best result comparing to the others and linear regression could not work well on this data set.**

I. INTRODUCTION

Nowadays the development of innovative materials is one of then most challenges for physical application, which concerns about the development of heath equipment, new energy application and many other fields. In order to optimize the property of materials, it is crucial to get a deep understanding the relationship among properties, composition and internal energy condition. Specifically, transparent conductors are significant compounds that are electrically conductive and low absorption in the visible range, which is a special property of these conductors and could make them applied to sensors, transistors and laser equipment[1].

However, transparency and conductivity is a pair of competing properties. If we have a wider bandgap, more photons with energy less than the bandgap will not be absorbed by the material which means wider range of visible light pass through the material that make it more transparent; Nonetheless, for the conductivity, a wider bandgap make it difficult to activate electrons from valance band to conduction band which lead to a less conductivity of the material. Traditionally, density functional theory (DFT) are always used for calculating the relative properties of materials, but it requires too much computing resources and time to achieve the goal. In this case, we are going to use machine learning models to predict some important features of transparent conductor instead of using density functional theory.

Another formidable problem is only a small portion of compounds is well understood that is able to be considered as transparent conductor. In order to find the optimum composition for transparent conductor, some basic principle should be the basic rule for computational approach. The alloys for transparent conductor should be $(Al_xGa_yIn_z)_{2N}O_{3N}$, where $x+y+z=1$ and N is an integer (usually between 5 and 100). There are infinite possible combinations for the values of x,y and z so the choice of computational method is the pivotal issue for transparent conductor materials design efficiency. There exist one primary computational method for materials science named Density Functional Theory, which is able to get high accuracy result but requires much computing time even for supercomputers. In this way, the data-driven method will be an alternative way to improve the efficiency for the transparent conductor design process.

II. RELATED WORK

In 2016, Joohwi Lee, *et al.* [2] made the prediction of G_0W_0 band gaps for inorganic compounds by some machine learning methods. In the paper, they used regression models, including OLSR and SVR. And found that SVR shows lower RMSE and when a new feature E_g added, the RMSE decrease a lot which help us to add new feature when the bandgap energy is not well predicted.

In 2015, Felix Faber, *et al.* [3] used DFT method to predict the formation energies of crystals. In their result, we found that although they use a traditional method for prediction, PCA was applied to dealing with the data set which helped for a better and faster prediction of formation energy.

In 2004, Shujiang Yang, *et al.* [4] made the prediction of bandgap for conjugated polymers. During their work, we could find that when calculating the bandgap energy, they used linear regression for calculating the features of number of atoms which is similar to our project, but we did linear regression on a larger data set which may cause more unexpected consequences.

H. K. D. H. Bhadeshia, *et al.* [5] introduced neural network to solve some difficult problem in materials science in 1999. In this article, he try to predict the tensile strength of metal, and some other properties of metal by training some neural network model based on already existed experimental datas.

In 2013, Sajeev, R. et al. [6] adopted several discrimitive classifiers to predict whether new molecules are semiconductor molecules or not. Among the models they used, including Naive Bayes, Random Forest, Support Vector Machines and Decision trees, Random Forest gained the highest accuracy.

III. DATA AND FEATURES

A. Data set

In this project, we used a data set which from a competition on Kaggle called Nomad2018 Predicting Transparent Conductors [7] that predicting the key properties of novel transparent conductors through some features. In the training data set, it contains about 3000 conductors which each one has 12 features that deciding the predicting result of two

extra properties of transparent conductors.
The 12 features of each conductors are:

- Space group which show the category of the material;
- Total number of Al, Ga, In and O atoms in the unit cell which influence the basic structure and properties of the material;
- Relative compositions of Al, Ga and In in the material which also affect the basic structure and properties of the material;
- lattice vectors $lv1$, $lv2$, $lv3$ which show the basic unit structure of material;
- lattice angles α, β, γ which also show the basic unit structure of material;
- Coordinate information (x, y, z) of each atom in each data sample which show the general structure of the material;

And the two predicting properties of the conductors are:

- Formation energy which is an important property that affect the stability of the material;
- Bandgap energy which is an important property that affect the optoelectronic properties of the material;

B. K-fold Cross-Validation

For a better prediction result, we didn't just separate the 3000 conductors into a training set and a validation set. Instead, we used K-fold cross-validation method which separate the total data set into K roughly equaled parts and for each $k = 1, 2, 3, ..., K$, use the other $K-1$ parts as training set to fit the model and compute its error in predicting K^{th} part. Then repeat k times so that each part could be a validation set and we could compute a average cross-validation error by summing all the errors and divided by K. K-fold cross validation is a advanced cross-validation method which could set every data be both a training data and a validation data that avoid the situation like some data are extremely different from others, but they are separated as a validation set which lead to a bad training result. However, K-fold cross validation may cost more time for computing because it will train the model K times. In this project, we set our $K = 5, 10$ or 15.

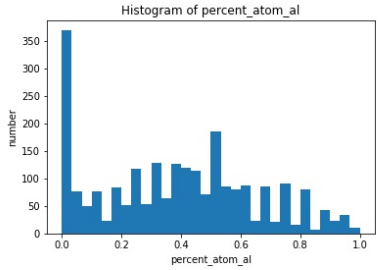
C. Data Preprocessing

1) Feature Selection: For the data of 3000 conductors, we tried to preprocess the data for a better training result. The conductors can be represented by the formula $(Al_xGa_yIn_z)_{2N}O_{3N}$, where x, y, z can vary but limited by the constraint $x + y + z = 1$. Considering the independence of features, we decide to drop the feature of percent component of In to make sure that all the features are linearly independent. After testing the model for the first time, we also tried some other methods for optimization. The first one is that we add a new feature *Volume* to the 3000 conductors by following a general volume equation for lattice regardless of special length or angle as

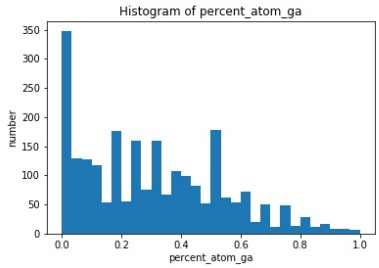
$$V = lv1 * lv2 * lv3 * \sqrt{1 + 2\cos\alpha\cos\beta\cos\gamma - \cos\alpha^2 - \cos\beta^2 - \cos\gamma^2} \quad (1)$$

To get the volume for all the conductors, we have to first transit the lattice angles from radians to degrees to fit the equation and also from the volumes we could get the density of each conductor by divide total number of atoms for each conductor by its volume.

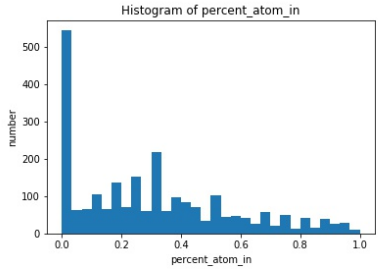
In order to observe the distribution on each feature and thus choosing reasonable models to model the 2 energies of these conductors in the data set, we plot several histograms of numbers of samples with the same values of each feature, and 3 of them are provided in Fig.1.



(a)



(b)



(c)

Fig. 1. Histogram of percent atom

2) PCA: To reduce the dimension of the coordinates, we also used a method called principle component analysis which is also to increase the computing speed. Principle component analysis is a widely using method that could find the principle components which have a higher variance that have a large contribution to the result and drop the uncorrelated components from the large data set to make the dimension reduce to a expected value so that the complexity

of the whole data set will decrease. We could understand the PCA method by the following steps:

- Assume we have a data set with a number of m and n features, we could define a matrix X with $m*n$ to illustrate the whole data.
- Calculate the mean value for each feature:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{2}$$

- Calculate the covariance matrix for the data set:

$$C = \frac{1}{m}\sum_{i=1}^{m}(x_i-\mu)(x_i-\mu)^T \tag{3}$$

- Calculate the eigenvalues and eigenvectors:

$$Cv_i = \lambda_i \tag{4}$$

- Sort the eigenvalues by a decreasing order and choose the first k values as k principle components.
- Project the data on the new feature space constructed by the k eigenvectors corresponding to the k eigenvalues for the principle components.

In this project, we used PCA method for the reduction of the coordinates dimension to two principle directions that could get rid of extra effect on uncorrelated features and increasing the computing speed.

IV. Methods

A. Linear Model

What we used first is the linear regression method. Linear regression is one of simplest and fastest machine learning model. The purpose of linear regression method is to find a linear function that the best fit the data, and in a high dimensional feature space, the linear regression is to find a hyperplane that could best predict the trend of the whole data set. First, we decide a cost function and try to minimize it so that the total error between the hypothesis and the original data can be minimized. Specifically, we use a matrix $X = (x_1, x_2, ..., x_n)^T$ which $x_i \in R^m$ to express the data and $Y = (y_1, y_2, ..., y_n)$ express the labels. Then for a data x_i, the output is

$$f(x_i) = \sum_{j=1}^{m} w_j x_{ij} + w_0 = w^T x_i \tag{5}$$

The w_0 is called bias and we use least square mean as a loss function as

$$J(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 = \frac{1}{n}||y - Xw||^2 \tag{6}$$

This time, we have to use the gradient descent that repeat until the function converge to find the fastest direction to the global or local minimum point. The basic gradient descent can be expressed as

$$w_j = w_j + \alpha\frac{d}{dw_j}J(w) \tag{7}$$

The α is called learning rate that should be determined to decide the step to the extremum and avoid divergence. For the whole data set, we can use a stochastic gradient descent as

$$\begin{aligned} &for\ i\ =\ 1\ to\ n \\ &w_j = w_j + \alpha(y_i - f(x_i))x_{ij} \end{aligned} \tag{8}$$

This is an advanced method comparing to the batch especially for a large data set because w_i only need one data to renew and that could dramatically improve the computing speed.

B. Neural Network Model

Since linear regression is a kind of linear model, which fitting the data with a straight line, it can't extract more complex information from data and underfit the data sometimes. Therefore, in order to develop the model, we use Artificial Neural Network (ANN),as shown in Fig.2 which is a computing system imitated from biological neural network. ANN is a kind of non-linear model, constructed by a input layer, an output layer and some hidden layers.

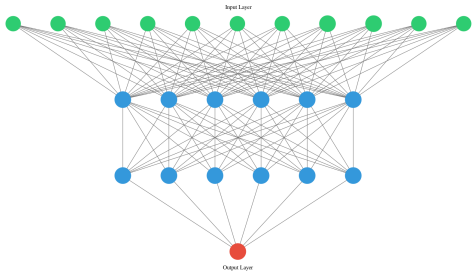


Fig. 2. A simple schematic diagram of neural network

The input is m number points in $n-dimensional$ feature space, such that input matrix is $X = (x_1, x_2, ..., x_n)^T$ where $i = 1, 2, ..., m$ and x_i is a point in n-dimensional feature space. Every hidden layer have a weight matrix $W = (w_1, w_2, ..., w_k)$, where dimensional of W is $k \times n$, k is number of node in current hidden layer and n is number of node in last hidden layer. Then, the output of this layer is $\widehat{X} = WX + b$, where a is the output of last layer and b is bias. And then, in order to avoid gradient vanish problem and accelerate convergence, $ReLU$ activation function which is $a_i = \sigma(x_i) = \max(0, x_i)$ or $a_i = \sigma(x_i) = \max(0.01x_i, x_i)$ is introduced. At last, the output of all points is $\widehat{Y} = (\widehat{y}_1, \widehat{y}_2, ..., \widehat{y}_n)$.

To make the prediction $\widehat{Y}$ much closer to real result Y, the use gradient decent of loss function with regularization to calculate the parameters. In addition, in order to prevent ANN overfitting the data, a $l2$ regularization is introduced to loss function. Hence, the loss function is given by:

$$L\left(Y, \widehat{Y}\right) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y}_i) + \frac{\lambda}{2N}\sum|W|^2 \tag{9}$$

and parameter W and b are calculated by gradient decent is given by:

$$W_{n+1} = W_n - \alpha\frac{\partial L}{\partial W_n}\ \ and\ \ b_{n+1} = b_n - \alpha\frac{\partial L}{\partial b_n} \tag{10}$$

We train 4 layers and 6 layers of neural network separately by introducing xavier initializer, l2 regularization and Adam optimization. The performance of ANN is better than linear regression.

C. Tree Based Model

Regression Tree is a commonly used regression model in many cases, especially when data points distributed relatively non-uniformly. In decision tress, the points along the tree where the data space are split into 2 parts are called nodes, and the corresponding segment connecting parent nodes and child notes are branches. For computational feasibility, recursive binary splitting are adopted, i.e., splitting the feature spaces based on one certain feature X_i into the region $\{X|X_i < s\}$ and $\{X|X_i \leq s\}$. The basic idea of regression tree method is to use the mean of all the values of training observations within one certain region as the prediction of each observation. And each splitting operation applied the greedy algorithms, i.e., choosing the splitting feature X_i and the value s that can obtaining the lowest error. The splitting process will stop only a certain condition is satisfied, such as when each region contains data points fewer than a fixed number. Finally, tree pruning is implemented to make a trade-off between the training accuracy and the number of trees, which is essential to reduce overfitting.

However, overfitting is still hard to prevent when making predictions based on only one decision tree. Therefore, bagging and ensemble methods are proposed to reduce the possibility of overfitting. There are three tree based models are used in this assignment, which are Adaptive Boosting (Adaboost) Regression tree, Gradient Boost Decision Tree and Random Forest. Different methods have specific tricks to optimize the regression prediction result.

1) Adaptive Boosting Regression Tree: Adaboost method is an ensemble learning method which can be applied by combining many other machine learning algorithms to improve their performance[8]. Specifically, the prediction results of weak learners, whether regression or classification, could be assembled into a weighted sum result that represents the final prediction result. Although the performance of each learners is weak, the final combined result could be strong as long as the performance of each one is better than random guessing.

Combined with decision tree, the hardness of each data sample will be collected at each stage. Then the later decision trees will tend to focus on the hard data samples. At the final step, the outputs of all weak learners will be calculated for a weighted sum to get the final output.

2) Gradient Boost Decision Tree: Gradient Boost Decision Tree is another type of boost learning method that could be used for ensemble learning. The gradient boost algorithm will optimize a cost function over cost function space. Specifically, the later weak learner will be trained aimed at the residual error of previous weak learner. The final strong learner is the weighted sum of all the weak learners. The weak learners are always selected as CART trees[9]. The primary parameters needed to be optimized is the number of estimators and maximum depth.

3) Random Forest: Random Forest is an ensemble learning method that generate multiple decision trees during the training procedure and make the output be the mean prediction result of all generated trees[10]. The randomness of random forest can be showed into the 2 following aspects. On the one hand, the training algorithm of random forest applied general technique of bootstrap to the tree learners. Given the training set and labels, the training process randomly select part of the samples with replacement to fit each independent tree. On the other hand, for the training process of each tree, only m features out of the total of p features are randomly chosen as predictors to train the model. Typically, m is approximately set to be $\sqrt{p}$ [11]. The randomness of random forest largely decreases the probability of overfitting of the model. After training expected number of trees, the prediction result of the test data is the average value of all individual trees.

V. Result and Discussion

We listed the Root Mean Squared Logarithmic Error (RMSLE) of the 5 models in Table I. And the RMSLE is given by the following equation:

$$RMSLE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left((\log \widehat{y}_i + 1) - \log(y_i + 1)\right)^2} \qquad (11)$$

Index	Model	Type	Formation Energy RMSLE	Bandgap Energy RMSLE
1	Linear Regression	Linear	0.064418	0.171237
2	Artificial Neural Network	Net	0.054654	0.094746
3	Adaboost Regression Tree	Tree	0.046249	0.123980
4	Gradient Boost Regression Tree	Tree	0.033199	0.092334
5	Random Forest	Tree	0.032237	0.090953

TABLE I
RMSLE OF 5 MODELS

A. Linear Regression

In this project, we first trained a linear model that could fit the whole data set. For the K-fold cross validation we used $k = 10$, $shuffle = True$ and $random\ state = 30$. We also tried ridge regression for regularization and the α we set was 0.1, 1.0, 10.0. However, what we got is a bad result with a high RMSLE error for the bandgap energy compare to the formation energy. It shows that the whole data set are not fit a hyperplane due to the large dimension of the feature space. In this case, we tried to add a new feature volume to the data set which could partially express the effects of lattice vectors and lattice angles. We also tried to use PCA to the lattice vectors. This time, the result error only decrease a little which means the linear regression model could not better predict the formation energy and bandgap energy for the transparent conductors.

B. Artificial Neural Network

Number of layers	Architecture	Formation Energy RMSLE	Bandgap Energy RMSLE
4	Input $\rightarrow$ 1024 $\rightarrow$ 512 $\rightarrow$ 64 $\rightarrow$ 2 $\rightarrow$ *out put*	0.069342	0.104174
6	Input $\rightarrow$ 1024 $\rightarrow$ 512 $\rightarrow$ 256 $\rightarrow$ 128 $\rightarrow$ 64 $\rightarrow$ 2 $\rightarrow$ *out put*	0.054654	0.094746

TABLE II

RESULTS TWO DIFFERENT ANN

Since the model of linear regression shows underfitting, we decided to implement Artificial Neural Network in this part. At first, we trained a shallow neural network, which has 4 hidden layers. The architecture of this ANN is shown in first row of Table II. In this ANN, we used the unpreprocessed data to train the network. As shown in Fig. 3, this ANN completed training through 100 epochs. When it near 15^{th} epochs, the cost function shows thee convergence. At last, when we use the trained ANN to predict a test set, the value of RMSLE shows lower than linear regression, so ANN have a better performance than linear model. However, from the result of RMSLE in Table II, this 4 layers ANN still has a relatively high error of bandgap energy. Hence, we should consider how to reduce the error.

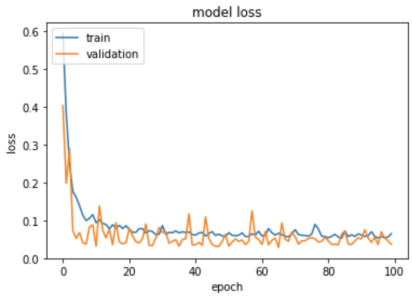


Fig. 3. Relationship between cost function and epoch before optimization

When we used a 4 layers ANN, it may still underfit the data. Hence, we tried to build a deeper ANN to solve the problem. What we built is a pyramidal neural network which has 6 layers and its architecture is shown in the second row of Table II. Besides, in order to extract more features from the data, we calculated the volume and density of materials. After adding more features to describe the materials more accurately, we train this 6 layers ANN. As shown in Fig. 4, when it near 20^{th} epochs, the cost function shows convergence, and the value of cost function is much smaller than previous ANN. As shown in Table II, the error of formation energy and bandgap energy are smaller than previous model. However, compared with the result from Kaggle, we still have a relatively high error. Hence, we will introduce other models to solve the problem in the next step.

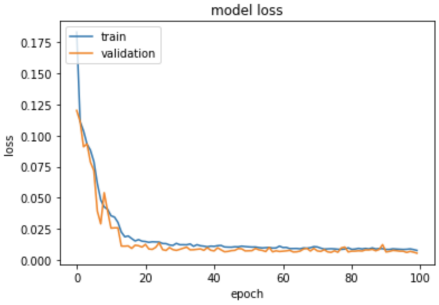


Fig. 4. Relationship between cost function and epoch after optimization

C. AdaBoost and Gradient Boost Regression Tree

Firstly, we simply extract useful features of the data, i.e., drop out 1 feature out of the total 11 features for the independence of each feature, and feed them into the two target models. With try-and-errors method, we eventually set the parameters and gained the corresponding errors as follows:

Methods	Number of estimators	maximum depth	Formation Energy RMSLE	Bandgap Energy RMSLE
AdaBoost	50	3	0.046249	0.129517
Gradient Boost	90	3	0.033199	0.101471

TABLE III

RESULTS OF THE MODEL TRAINED WITH THE ORIGINAL 10 FEATURES

Obviously, the error of the first prediction is much lower than that of the second one. In order to further increase the accuracy of the models, we then applied the atomic coordinates provided in the data set and feed them into the models to predict the Bandgap energy after decreasing the dimensions of them using Principal component analysis (PCA). The reason why we used PCA is the large The final parameters and errors of the optimized models are listed in the following table.

Methods	Number of estimators	maximum depth	Bandgap Energy RMSLE
AdaBoost	50	3	0.123980
Gradient Boost	90	3	0.092334

TABLE IV

RESULTS OF THE MODEL TRAINED WITH MORE FEATURES

The results listed in the table above indicates that for both the two models, there is slightly reduction in errors of bandgap energy, but the value of which is still much higher compared to that of formation energy. Therefore, compared to the atomic coordinates, the general properties of a certain conductor is sufficient for predicting the two target properties. And if higher accuracy is needed for predicting bandgap energy, other related information might be needed.

D. Random Forest

There are 2 hyper-parameters in random forest model, one is the number of trees, the other is the max depth of each tree. Here, we used k-fold cross validation to decide the hyper-parameters. We set k to be 5, and comparing the scores of the

models by setting the number of trees from 100 to 500 and setting the max depth of each tree from 5 to 14. Experimental data shows that under same max depth, the influence of changing the number of trees are slight, while under same number of trees, the influence of changing the max depth is much larger, which is shown in Fig.5. Eventually, when the maximum depth = 9, and the number of trees = 500, the best testing accuracy are gained.

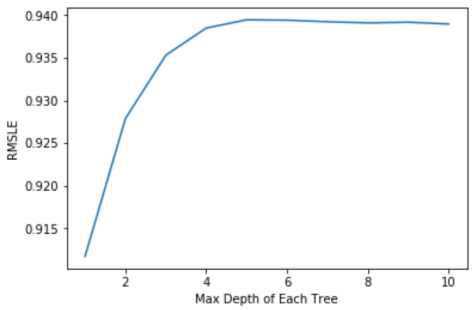


Fig. 5. RMSLE - maximum depth of each tree Curve

Specially, among the 5 models, the tree-based models achieved better performance, which indicates that the tree-based methods are suitable for data sets whose distribution on each features is not uniform. Among the 3 tree-based models, the random forest has the lowest RMSLE and relatively short training time, for it is a method whose training process can be implemented in a parallel way.

VI. Conclusion and Future Work

A. Conclusion

Predicting the target values of this data set we studied on can be regarded as a regression problem with more than 10 features and 2 labels. In this case, simple linear regression model seems too simple to extract the features. Then artificial neural network, with the non-linear activation function, make the error lower than that of linear model. Overall, the performance of the tree-based ensemble models is proved to be better than that of the other two models. There are two possible reasons. The one is that the ensemble methods themselves have high capability of generalization. The other is that the data points is not uniformly distributed on each feature, which means that the mean value of a separated region can be representative. Compared to the two models following serial processes, AdaBoost and Gradient Boost regression trees, random forest can be trained by a parallel process and thus the number of estimators can be much more within the same training time. Besides, the results after adding the atomic coordinates into the feature space indicate that the basic general properties are sufficient to obtained relatively precise predictions.

B. Future Work

Our predictions of the bandgap energy is not as accurate as that of the formation energy, and the RMSLE of predicting bandgap energy kept larger even after a series of optimization methods. Our hypothesis for this problem is that the features currently provided in the data set are more related formation energy of conductors, so more features such as electronic properties of these materials might be needed. Moreover, there may exist models other than we tried in this project more suitable for this problem. Therefore, our future work will focus on the two aspects.

References

[1] D. S. Ginley and J. D. Perkins, "Transparent conductors," *Handbook of Transparent Conductors* , 125 (2010).

[2] J. Lee, A. Seko, K. Shitara, *et al.*, "Prediction model of band gap for inorganic compounds by combination of density functional theory calculations and machine learning techniques," *Physical Review B* **93**(11), 115104 (2016).

[3] F. Faber, A. Lindmaa, O. A. von Lilienfeld, *et al.*, "Crystal structure representations for machine learning models of formation energies," *International Journal of Quantum Chemistry* **115**(16), 1094–1101 (2015).

[4] S. Yang, P. Olishevski, and M. Kertesz, "Bandgap calculations for conjugated polymers," *Synthetic Metals* **141**(1-2), 171–177 (2004).

[5] B. HKDH, "Neural networks in materials science," *ISIJ international* **39**(10), 966–979 (1999).

[6] R. Sajeev, R. S. Athira, M. Nufail, *et al.*, "Computational predictive models for organic semiconductors," *Journal of Computational Electronics* **12**, 790–795 (2013).

[7] "Kaggle: predicting transparent conductors. https://www.kaggle.com/c/nomad2018-predict-transparent-conductors/data."

[8] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences* **55**(1), 119–139 (1997).

[9] L. Breiman, J. Friedman, R. Olshen, *et al.*, "Classification and regression trees. 1984. monterey, ca: Wadsworth & brooks."

[10] T. K. Ho, "Random decision forests," in *Document analysis and recognition, 1995., proceedings of the third international conference on*, **1**, 278–282, IEEE (1995).

[11] G. James, D. Witten, T. Hastie, *et al.*, *An introduction to statistical learning*, vol. 112, Springer (2013).