Prediction of Lattice Thermal Conductivity for Thermoelectric Materials Using Machine Learning

Sudeendra R (MM23S001), Anudeep Sadarla (MM23M005), Atharva Vilas Vyawahare (MM23M006), Sonal Anish Jojo (MM23M021), Syam Sundara Raju Adda (MM23M025)

Abstract: Thermoelectric materials are type of materials that uses thermoelectric effect to convert temperature difference to electric potential. zT is a property that determines the efficiency of a thermoelectric material. Lattice Thermal conductivity is a crucial parameter to determine zT. So, in this project we have created ML model by using various features that predicts the lattice thermal conductivity of a thermoelectric material. The project is divided into two parts. In each part different sets of features were used and ML models were trained according to the features used and results were obtained. We also predicted Seebeck coefficient and electrical conductivity using the same set of features and their results were noted. The trained model is then used to predict lattice thermal conductivity of new compounds using different candidate sets.

Keywords: *Thermoelectric, Lattice Thermal Conductivity, Regression, RMSE &R2*

Introduction

Thermoelectric materials serve as a reliable and alternative source for the production of electricity from waste heat. In addition, the working of thermoelectric devices involves no relative motion between the parts, which increases the efficiency of power conversion. These materials, when subjected to a temperature gradient, develop an electric potential due to the diffusion of majority charge carriers from the hot end to the cold end. Diffusion of the charge carriers should be predominant in such materials in comparison with the thermal conductivity. The efficiency of thermoelectrics is expressed in terms of the figure of merit parameter and is given by the formula below:

$$zT = \frac{S^2}{\rho\kappa}T$$

The figure of merit parameter depends on material properties such as the Seebeck coefficient, electrical resistivity and thermal conductivity and is also dependent on temperature. zT parameter is directly proportional to the Seebeck coefficient and inversely proportional to electrical resistivity and thermal conductivity. Thermal conductivity consists of two components: electronic thermal conductivity and lattice thermal conductivity. Electronic thermal conductivity, electrical resistivity, and Seebeck coefficient are highly correlated; hence, manipulation of these parameters without affecting each other is unfeasible, leaving behind lattice thermal conductivity as an independent and essential parameter to study.

Lattice thermal conductivity, quantized by phonons, can be decreased by doping. Thermoelectric materials mainly include bulk materials made from heavily doped semiconductors as they have low electrical conductivity and also low dimensional systems such as 2D materials. The above-mentioned materials have lesser lattice thermal conductivity because of dopants and reduced dimensions. Measuring the lattice thermal conductivity of such materials is a tedious process, and existing methods of computing lattice thermal conductivity and their drawbacks are discussed below:

1. FIRST PRINCIPLE CALCULATIONS:

First principle calculations include DFT calculation to generate phonon DOS and also to calculate group velocities, harmonic and anharmonic interatomic force constants by solving the Boltzmann transport equation. This method is computationally extremely expensive as it requires multiple large supercell calculations.

2. THERMODYNAMIC/ SEMI-EMPIRICAL MODEL:

Several thermodynamically proposed empirical models prevail to predict the lattice thermal conductivity of various materials. Some of the popular models include the Slack model and the Debye-Callaway model.

$$\kappa_L = A \cdot \frac{(\theta e)^3 M \sqrt[3]{V n_p}}{n^{4/3} T \gamma_a^{\,2}}$$

$$\kappa_{L,tot} = A_1 \frac{M v_s^{\,3}}{T V^{2/3} n^{1/3}} + A_2 \frac{v_s}{V^{2/3}} \left(1 - \frac{1}{n^{2/3}}\right)$$

Although these models provide better insights, they do not convincingly predict the lattice thermal conductivity of various materials.

Hence, the machine learning approach serves as a good alternative in predicting the lattice thermal conductivity of various materials as it bridges the gap between computationally expensive first principle calculations and approximate empirical models.

Model – 1: Data and Features

A total of 113 unique data points of thermoelectric materials consisting of Tellurides, Actinides and Lanthanides, Selenides, Half-Heuslers and other families were used to build the dataset. 18 features were considered as inputs to the model. The features are: AGL Debye temperature acoustic, AGL Debye temperature, AGL Grüneisen parameter, AGL heat capacity C_p (300 K), AGL heat capacity C_v (300 K), AGL vibrational entropy atom (300 K), AGL vibrational free energy atom (300 K), AGL bulk modulus static (300 K), Enthalpy atom, Total energy atom, Electronic band gap, Valence electrons cell (standard), Cell volume relaxed, Mass density, Number of atoms, Weighted Average, Temperature and Weighted Average. All the feature values were extracted from the Aflow database. The output we wished to get was Lattice thermal conductivity using the 18 features. Fig.1 shows the visualization of the input features for model – 1.

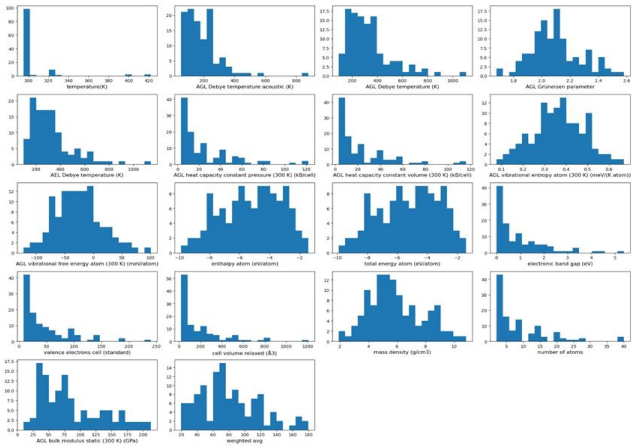


Fig. 1 - Data Visualization of the input features for model - 1

As can be seen from Fig. 1, the temperature is not spread across a wide range but is confined to a very narrow range, whereas the other features have a wide spread of information. This particular issue is going to affect the performance of the model, which will be discussed later in the report, and it is also important to note that not all feature's information spread is identical.

A broad distribution of input feature information within a machine learning model can result in various outcomes, depending on the circumstances:

- Enhanced Generalization: By encompassing a wide range of data aspects, a broad distribution of input features can aid the model in generalizing effectively to unseen data. This reduces the risk of overfitting to the training data and enhances performance on new, unseen instances.
- Resilience to Variability: Diverse input features can bolster the model's resilience to data variability. It enables the model to accommodate different patterns and variations within the dataset, leading to more consistent predictions across diverse scenarios.
- Elevated Computational Complexity: Nonetheless, it is important to acknowledge that a broad distribution of input features may escalate the computational complexity of the model, particularly if many features are noisy or irrelevant. This can prolong training times and increase resource demands.

Now, we'll focus on the algorithms and ML techniques used to build the model - 1, their performances, and the improvements required (if any).

Model - 1: Algorithms, Results and Discussion

The various algorithms and ML techniques used to build the model are: Linear Regression, Ridge & Lasso Regressions and finally Random Forest Regression algorithm.

1. The concept of Linear Regression involves assuming a linear relationship between the input features and the target variable. This model aims to minimize the sum of squared differences between the predicted and actual values. However, it is important to note that Linear Regression can be prone to overfitting if the data is complex or contains noise.
2. Ridge Regression, on the other hand, is an extension of Linear Regression that incorporates a regularization term into the loss function. This regularization term penalizes large coefficients, which helps to reduce overfitting. Ridge Regression is particularly useful when dealing with multicollinearity among the input features.
3. Lasso Regression is similar to Ridge Regression but utilizes L1 regularization instead of L2. The L1 regularization encourages sparsity in the coefficients, leading to feature selection. This makes Lasso Regression beneficial when dealing with high-dimensional data or when there are numerous irrelevant features that need to be eliminated.
4. Random Forest Regression, in contrast, is an ensemble learning method that constructs multiple decision trees during the training process. Each tree is trained on a random subset of the data and features, introducing randomness and reducing overfitting. By combining the predictions from multiple trees, Random Forest Regression offers robustness and improved performance. It is particularly effective in handling non-linear relationships between the features and the target variable.

In conclusion, Linear Regression provides a simple and interpretable model, while Ridge and Lasso Regression incorporate regularization to address overfitting. Random Forest Regression, as an ensemble method, utilizes multiple decision trees to enhance predictive performance and handle complex relationships in the data. Each of these algorithms has its own strengths and is suitable for different types of data and modeling objectives. The parity plots of all the algorithms have been plotted to visualize the output or the performance of the algorithms. The test root mean squared error (RMSE) and the R2 score are also displayed on the plot as the

evaluation metrics to know our model's performance. Fig. 2 to Fig. 5 shows the parity plots of Linear, Ridge, Lasso and Random Forest regression algorithms.

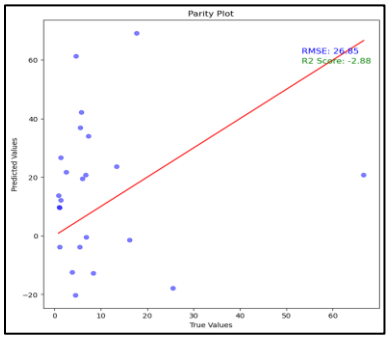


Fig. 2 - Parity plot of Linear Regression

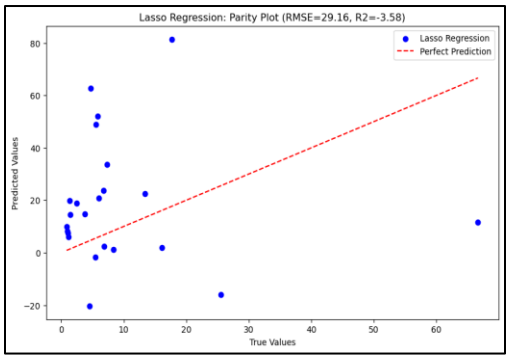


Fig. 3 - Parity plot of Lasso Regression

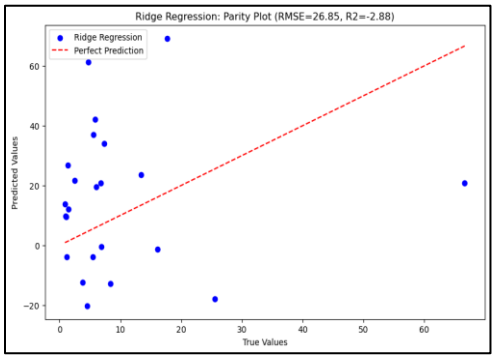


Fig. 4 - Parity plot of Ridge Regression

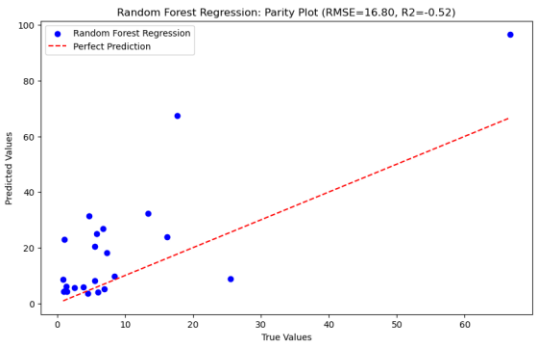


Fig. 5 - Parity plot of Random Forest Regression

As can be seen from the plots, the RMSE values and R2 scores indicate that the algorithms are not able to capture the entire information from the given input features. Fig. 2 to Fig. 4 show the parity plots of linear, lasso and ridge regression algorithms which show a very poor (highly negative) R2 score. We can see that including regularization algorithms, which solve the problem of overfitting, didn't improve our model's performance (in fact they worsened the performance). This shows that the reason for such bad performance in our linear regression algorithm is not due to overfitting but there's some other important issue we overlooked.

This issue could be attributed to the reason that some of our features (if not all) might be dependent on each other and this leads to non-linearity in the feature's information. Considering the non-linear data, our linear regression algorithms like linear, ridge and lasso won't be able to perform well due to the fact that they assume our data to be linear. Coming to the random forest algorithm, as can be seen from Fig. 5, the R2 score is slightly better compared to the previous algorithms but it is still negative. It may have resulted in poor performance (even though it doesn't necessarily assume our data to be linear) because the input data points (or the compounds in our case) are taken from several different families of thermo-electrics. Different families of thermo-electrics have different underlying physics and chemistry, which means we cannot include different families in the same data set for our model.

Fig. 6 shows the feature importance ranking which is obtained using random forest regressor.

1. Examining the feature-importance ranking of input features in a machine learning model offers various benefits. One advantage is gaining insight into the behavior of the model. By understanding which features have the most significant impact on the model's predictions, one can interpret its decisions and uncover the underlying data relationships.
2. Another advantage is the facilitation of feature selection. The ranking of feature importance can assist in identifying the most relevant features for prediction, especially in high-dimensional datasets where not all features contribute equally to the model's performance. Focusing on the most important features can simplify the model, decrease computational complexity, and potentially enhance its ability to generalize to new data.
3. Moreover, feature importance ranking can help in identifying predictive factors related to the target variable. This information is valuable for domain experts who aim to comprehend the variables influencing the outcome of interest, potentially leading to actionable insights or aiding in decision-making processes.
4. Lastly, the detection of redundant or irrelevant features is another advantage of feature importance ranking. Features with low importance scores may indicate redundancy or irrelevance in the dataset. By eliminating these features, the model can be streamlined, noise reduced, and efficiency and interpretability improved.

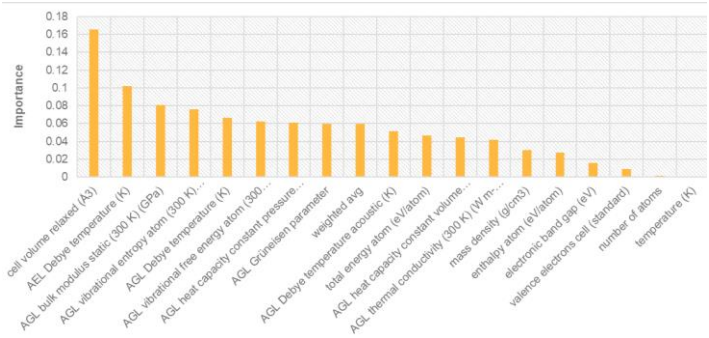


Fig. 6 - Feature importance ranking of all 18 features

As can be seen from the figure, highest importance was given to cell-volume-relaxed but temperature was given the least importance. But from domain knowledge, we know that temperature is the most important factor affecting the lattice thermal conductivity. This shows that our model failed to capture the proper weightage of each feature therefore resulting in a poor performance.

Future Work

Features selected require first principle calculations which results in lesser input data and does not allow it to be used for a completely unknown material. Training data sets consist of thermoelectric with 8 different families. Each family shows unique behaviors of the selected features, with more & sparse data this issue may get resolved. Predicted weight importance of temperature is least amongst all the features, even though all the target properties directly depend on temperature. This could be because the training set mostly consisted of target values at or near the temperature of 300 K.

Model - 2: Data and Features

Compared to the previous model, a new data set of entirely new features and compounds have been taken into account. A total of 5205 data points with 880 unique thermoelectric materials mostly consisting of tellurides & selenides families were taken as the data. (880 unique compounds at different temperatures make up to a total of 5205 data points). Taking the same compounds at different temperatures will enable the model to understand the importance of temperature in predicting our target property, lattice thermal conductivity. Higher level fingerprinting of materials was used. This is done to solve the issue of different underlying physics in different compounds which we faced while training the previous failed model.

All the elemental information was taken from: https://github.com/Kaaiian/CBFV. Fingerprinting was done in this manner: $\sum \boldsymbol{Cj.\ Pj,}$ where, C is stoichiometric coefficient of the element j in the compound & P is elemental property of element j.

A total of 18 features were considered for training this model - 2. These features which are atomic number, atomics weight, mendeleev number, l-quantum number, atomic radius, covalent radius, pauling electronegativity, number of valence electrons, 1st ionization potential, melting point, boiling point, density, specific heat, heat of fusion, heat of vaporization, space group number, thermal conductivity and temperature. Fig. 7 shows the visualization of the input features data for model – 2.

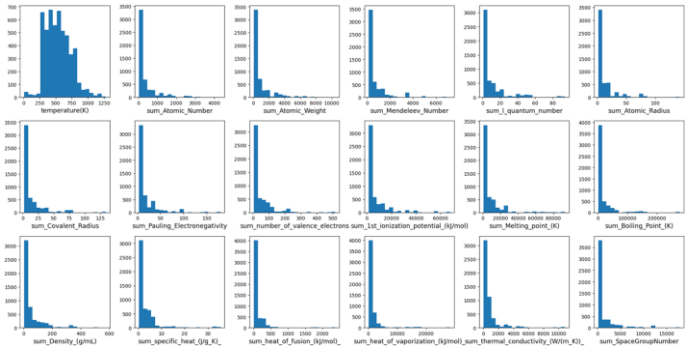


Fig. 7 - Data visualization of the input features for model - 2

In Fig. 7, temperature information is widely spread in a broad range and the information of other features is mostly confined to a narrow range which was expected to happen as we took only 2 different families of thermo-electrics as our primary data set to train our model. (Later in the report we'll talk about the secondary data set which is our candidate data set for making predictions). For training the model - 2, we used 5205 data points with 18 features. The ML techniques and algorithms, their performance and also the scope of improvements for the model are discussed below.

Model - 2: Algorithms, Results and Discussion

(For Lattice thermal Conductivity)

Similar to model - 1, here we used linear algorithms like Linear regression, Ridge regression. Along with these linear algorithms, we also employed the K-nearest neighbor algorithm and ensemble algorithms like Random Forest regression and finally eXtreme Gradient Boosting (XGBoost). In addition to the algorithms whose basic principles were explained previously, here let's discuss the basic principles of KNN and XGBoost techniques:

- K-Nearest Neighbors (KNN) is a simple algorithm that looks at the majority class of its nearest neighbors to classify or predict. It can capture complex patterns in data but can be slow with large datasets and struggles with imbalanced data.
- XGBoost is an efficient implementation of gradient boosting that builds decision trees sequentially to correct errors. It is accurate, handles missing data, and is computationally efficient. However, it needs hyperparameter tuning and can overfit if not regularized properly.

Parity plots of all the algorithms have been plotted to visualize the output or the performance of the algorithms. The test root mean squared error (RMSE) and the R2 score are also displayed on the plot as evaluation metrics to know our model's performance. Starting with the Linear algorithms: Linear and Ridge regression techniques, we can see their parity plots from Fig. 8 and Fig. 9.

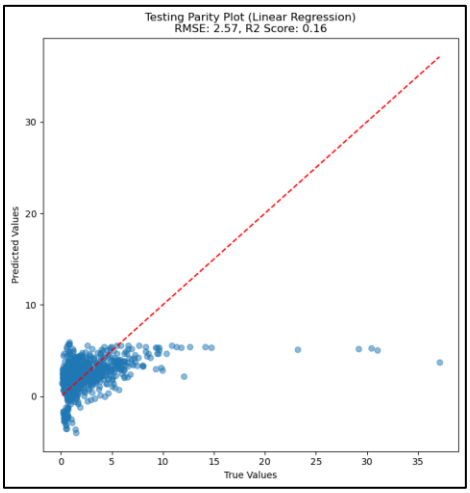


Fig. 8 - Parity plot of Linear

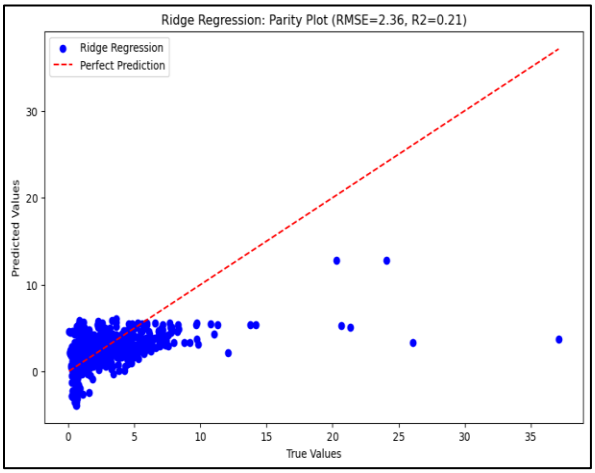


Fig. 9 - Parity plot of Ridge Regression

As can be seen from the plots, we can say that the performance of these algorithms is not very good but the performance is much better compared to the model - 1 case where we saw negative R2 scores for both linear algorithms. This improvement can be attributed to the fact that most of the features (if not all) we chose in our new model are not dependent on each other, i.e., they are linearly independent. Parity plot for the KNN algorithm is shown in Fig. 10.

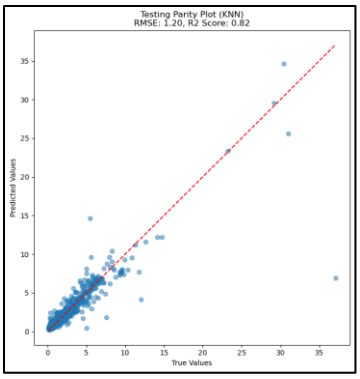
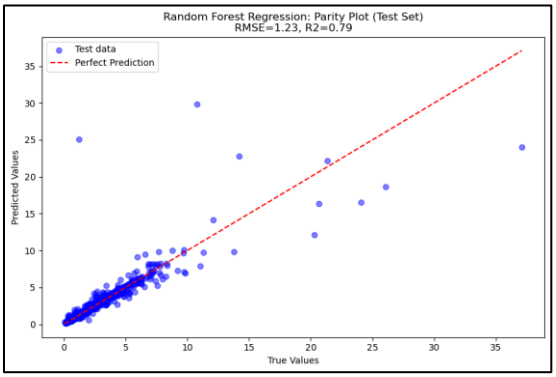


Fig. 10 - Parity plot of KNN algorithm

From the parity plot in Fig. 10, we can see that the R2 score is 0.82 which says that the performance of KNN algorithm is very good compared to the linear algorithms. This could be due to the reason that KNN is a non-parametric algorithm, which offers flexibility in modeling complex relationships by not making any assumptions about the underlying data distribution. It has the ability to capture intricate and nonlinear connections between features and the target variable. In contrast, linear algorithms assume a linear relationship between features and the target, which may not be applicable to all datasets.

Fig. 11 (a) and Fig. 11 (b) show the Random Forest regression algorithm without and with Recursive Feature Elimination (RFE).

(a)



(b)

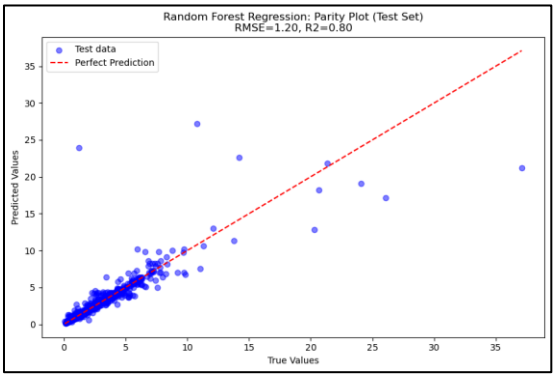
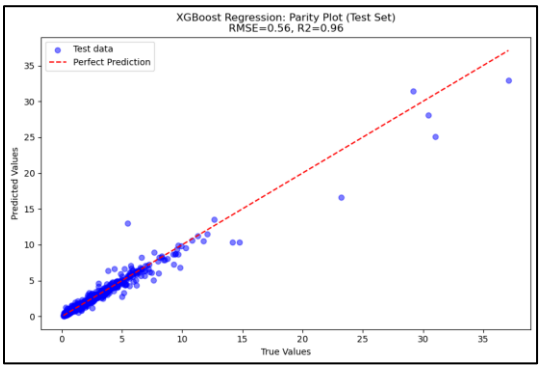


Fig. 11 - (a) Random Forest without RFE
(b) Random Forest with RFE

Recursive Feature Elimination (RFE) is an approach for selecting features in which features are iteratively eliminated from the dataset and the model is trained using the remaining features. Each feature is assigned a weight and ranked according to its significance. This iterative process continues until the desired number of features is achieved. By identifying the most crucial features for the model, RFE enhances its performance by mitigating overfitting and enhancing interpretability. The performance of the random forest regression algorithm is almost identical to that of KNN. The effect of RFE technique on this algorithm is not showing much improvement in the results. The R2 score slightly increased from 0.79 to 0.80 after performing RFE.

The final algorithm used to train our model was XGBoost. Its parity plots without and with performing RFE are shown in Fig. 12 (a) & (b).

(a)



(b)

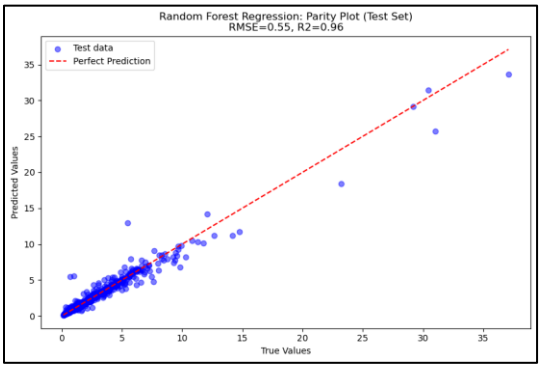


Fig. 12 - (a) XGBoost without RFE
(b) XGBoost with RFE

As can be seen from figures 12 (a) and (b), the performance of the XGBoost algorithm is far superior to all other techniques we employed so far. XGBoost's superior performance can be attributed to its capability to handle complex relationships in data, feature interactions, and noisy data effectively through ensemble learning, feature importance estimation, and regularization techniques. Additionally, its optimized implementation enables efficient training and prediction. Performing RFE did not change the R2 score, i.e., the performance of the algorithm remained the same. This means that our model is generalized with respect to the features and the target value.

Feature importance ranking of lattice thermal conductivity, Fig. 13 displays the feature importance ranking of all the 18 features used to build the model. This is obtained by using the random forest regressor.

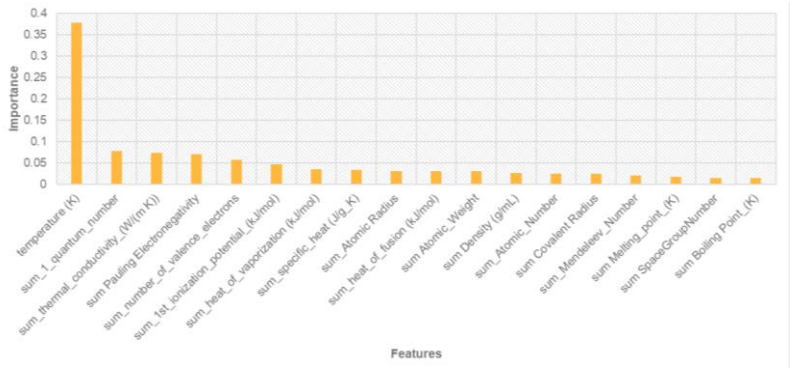


Fig. 13 - Feature importance ranking of all 18 features used for predicting lattice thermal conductivity

As can be seen from the figure, the highest importance was given to temperature, which we were hoping for because of the reason explained in the model-1 case, i.e., the temperature is directly related to lattice thermal conductivity. This is directly reflected in the model's good performance.

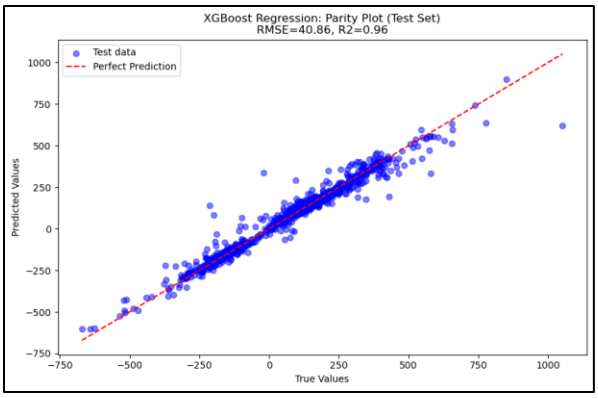
Model - 2: Seebeck Coefficient

Seebeck coefficient is another important parameter in determining the quality of thermoelectric. Due to this reason, we went one step further and tried to train the model for predicting the seebeck coefficient (as well as electrical conductivity, which will be discussed later in the report). It is to be noted that the features used for predicting the other two target properties (seebeck coefficient and electrical conductivity) are the same as the ones used for lattice thermal conductivity.

For Seebeck coefficients, unlike the algorithms used in training the model for lattice thermal conductivity, here we only used Random Forest and XGBoost algorithms as Seebeck coefficient prediction was of secondary interest. Fig. 14 (a) & (b) show the parity plots obtained using the XGBoost regression algorithm without and with RFE.

Just like in the case of the lattice thermal conductivity model, XGBoost performs really well, and the effect of decreasing R2 score after performing RFE could be attributed to the same reason explained before in the lattice thermal conductivity case. Fig. 15 (a) & (b) show the parity plots obtained using the Random Forest regression algorithm without and with RFE.

(a)



(b)

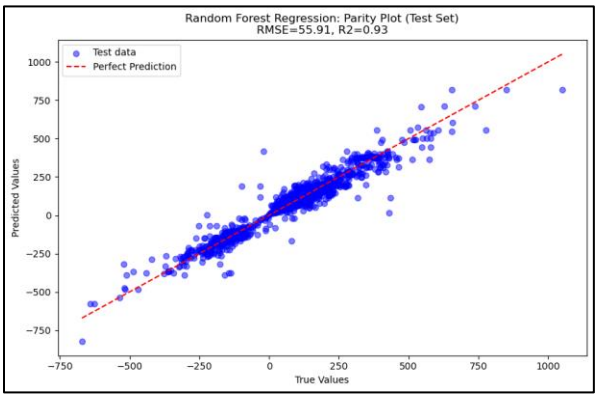
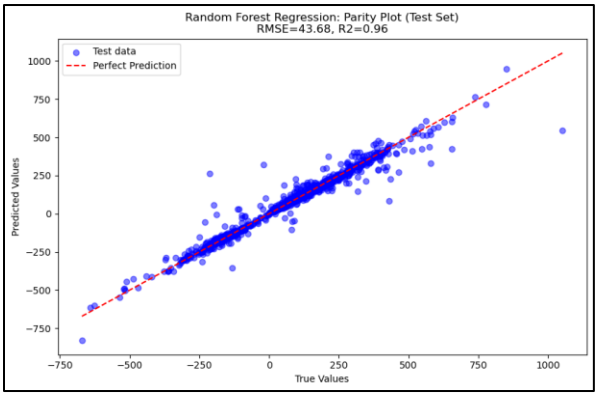


Fig. 14 - (a) XGBoost without RFE
(b) XGBoost with RFE

(a)



(b)

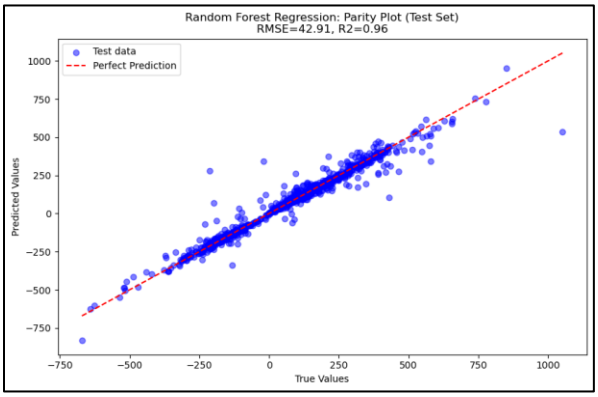


Fig. 15 - (a) Random Forest without RFE
(b) Random Forest with RFE

Random forest works better than XGBoost in predicting the values of Seebeck coefficients compared to the case for lattice thermal conductivity, and again, it can be seen that RFE doesn't change the performance of the model, meaning the model is generalized with respect to the features and target values.

Feature importance ranking of Seebeck coefficient: Fig. 16 displays the feature importance ranking of the Seebeck coefficient model. This is again obtained by using the random forest regressor.

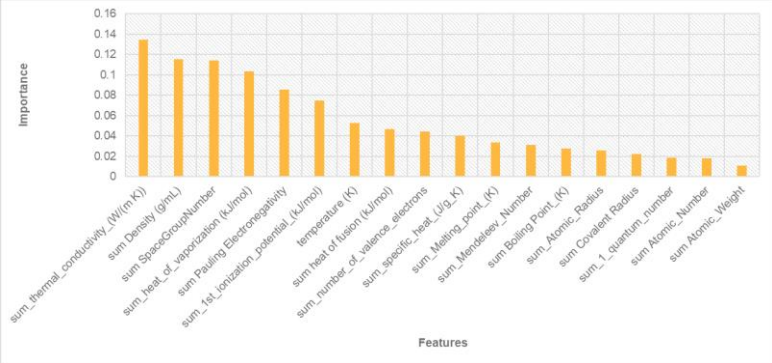


Fig. 16 - Feature importance ranking of the Seebeck coefficient model

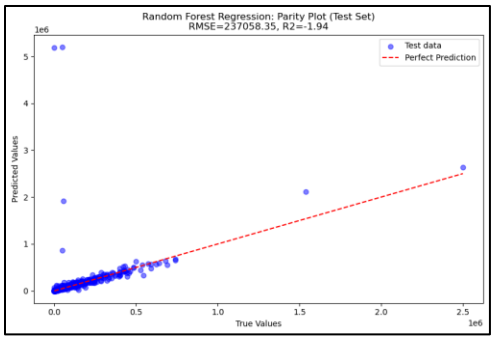
From domain knowledge, we know that the most important factor for predicting the Seebeck coefficient (in other words, the factor on which the Seebeck coefficient. is highly dependent) is the total thermal conductivity. This is exactly what the model also did while performing RFE. This shows how well the model is trained because of the features used.

Model - 2: Electrical Conductivity

As mentioned earlier, we also tried to train a model for predicting electrical conductivity. The same 18 features are used again. For training this particular model, we only employed the Random Forest algorithm. Again, we tried performing RFE on the model to check its performance both with and without RFE. Fig. 17 (a) and (b) show the parity plots obtained using the Random Forest algorithm without and with performing RFE.

As can be seen from the figures, the R2 score is negative with a very high RMSE value for both the with and without RFE cases. The model we trained for predicting the electrical conductivities is way worse than the models for lattice thermal conductivity and Seebeck coefficient predictions, which had R2 scores above 0.90.

(a)



(b)

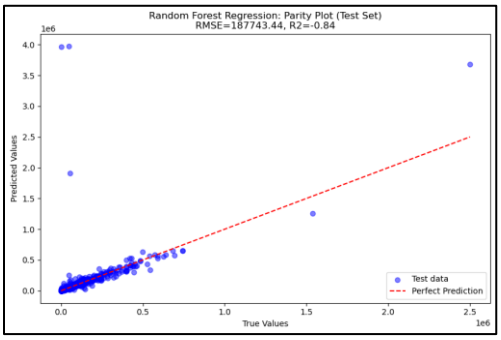


Fig. 17 - (a) Random Forest without RFE
(b) Random Forest with RFE

The reason for this could be the fact that we used the same features again. Actual features necessary for predicting electrical conductivity might be different, and hence, the model is not able to capture the trends in the current features and the corresponding target value. We can validate this reason by looking at the feature importance ranking for the current model for predicting electrical conductivity. Fig. 18 shows the same.

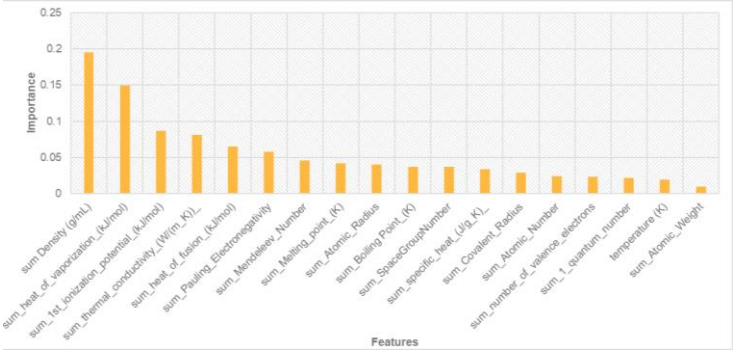


Fig. 18 - Feature importance ranking of the electrical conductivity model

From the figure, it's very clear that the model failed to capture the fact that electrical conductivity mainly depends on temperature. Instead, the model gave importance to density, heat of vaporization, etc.

Model - 2: Predictions (Candidate data set)

A separate data set known as the Candidate set is prepared, containing only the names (or) the compounds list. We use this data set to make predictions for lattice thermal conductivities of the new compounds by fitting the data set to the same model we trained for lattice thermal conductivity. Two distinct types of candidate sets were prepared:

- Candidate Set - 1: Contains compounds belonging to the same thermoelectric family (tellurides). This data has a total of 108 data points with 56 unique compounds at different temperatures.
- Candidate Set - 2: Contains compounds belonging to 8 different thermoelectric families. This data has a total of 44 unique compound data points at different temperatures.

Once we fit the trained model to the candidate data, we can make predictions for the new compounds. Fig. 19 shows the predictions made using the candidate set - 1.

		Lattice Thermal conductivity		
Formula	Temperature(K)	Prediction	TRUE	Error
AgCuTe0.9Se0.1	670	0.4741	0.25	0.2241
Ge0.9Sb0.1Te	300	1.6447	1.42	0.2247
K0.02Pb0.98Te0.75Se0.25	773	1.043	0.8	0.243
AgCuTe0.9I0.1	463	0.4538	0.17	0.2838
Pb0.953Na0.04Ge0.007Te	805	0.9643	0.67	0.2943
Bi0.3Sb1.625In0.075Te3	300	0.9054	0.61	0.2954
(Ag1.9996Te)0.9(PbTe)0.1	550	0.5386	0.23	0.3086
AgInTe2	300	1.7349	1.42	0.3149
AgCuTe	660	0.5623	0.23	0.3323
Ge0.9Sb0.1Te	725	1.5208	1.15	0.3708
Ag2Sb0.02Te0.98	300	0.7496	0.36	0.3896
CuInTe2	875	1.2058	0.81	0.3958
BiTe	500	16.6527	0.76	15.8927
BiTe	300	17.3373	0.72	16.6173
Mn0.98Na0.02Te	900	25.87	0.58	25.29
MnTe	900	26.0979	0.67	25.4279
MnTe	300	27.1478	1.18	25.9678
Mn0.98Na0.02Te	300	26.9963	0.82	26.1763

Fig. 19 - Lattice thermal cond. predictions for candidate set – 1

The true values of the lattice thermal conductivities of these compounds were obtained from the data given in the paper: https://doi.org/10.1039/C4EE03157A. The predicted values, the true values and the error associated with each prediction are given in Fig. 19. The compounds marked in green are the ones having very low errors between the predicted and true values, whereas the ones marked in red have very high errors.

However, out of 108 data points in the candidate set, only around 5-6% of the data had high errors in their predictions. This shows that our model's performance/evaluation metrics were true to their values, and the model performed really well in predicting the target property for entirely new and unseen data. This is because candidate set-1 has compounds belonging to the same telluride family of thermoelectric. This means that the underlying chemistry of all compounds in this data set is the same, so there won't be any additional error introduced while making predictions apart from the error associated with the model itself.

Fig. 20 shows the predictions made using the candidate set - 2.

		Lattice Thermal Conductivity		
Formula	Temperature(K)	Prediction	TRUE	Error Percentage
Yb0.066Co4Sb12	185	4.729199	4.8	1.475020833
Ca0.89Pr0.08Sr0.03MnO3	373	1.7069881	1.8	5.167327778
Sr0.8La0.067Ti0.8Nb0.2O3	310	4.268607	4.6	7.204195652
Bi0.94Y0.06CuSeO	300	0.83466965	0.7	19.23852143
Mg2Si	300	18.8099	6	213.4983333
YB66	300	6.528094	2	226.4047
Sr14MgSb11	300	2.1282022	0.55	286.9458545
Cu1.98Ag0.2Se	900	1.1695403	0.25	367.81612
MnSi1.733	850	9.413545	2	370.67725
Ce0.14Co4Sb12	122	5.657649	1.1	414.3317273
MnSi1.746Te0.03	823	9.456624	1.5	530.4416
Mg2.16Si0.392Sn0.591Sb0.015	300	12.564038	1.7	639.0610588
Mg1.98Li0.02Si0.4Sn0.6	300	15.871912	1.6	891.9945

Fig. 20 - Lattice thermal conductivity predictions for the candidate set – 2

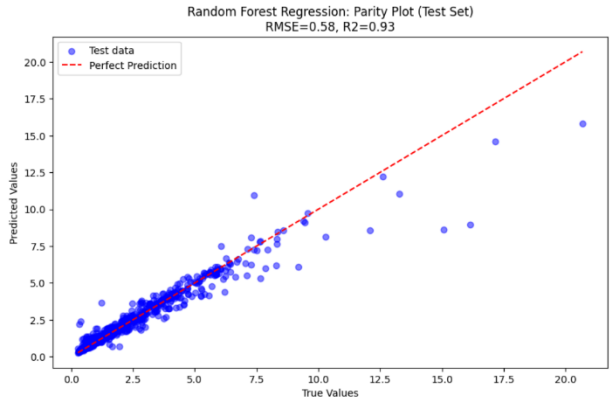
The true values of the lattice thermal conductivities of these compounds were obtained from the data given in the same paper as the one used for the candidate set - 1. The same marking was done for this candidate set as well - green for low error predictions and red for high error predictions. A relatively large number of data points can be seen to have high errors associated with them compared to very few compounds with less error.

This can be understood by the fact that candidate set - 2 was prepared by including compounds from 8 different families of thermoelectrics. This means that the underlying chemistry changes from family to family, but we didn't let the model know about these changes in the chemistry during training, i.e., we didn't show this new trend to the model. Hence, the model made very bad predictions.

Future Work

- A more generalized model approach where the training data consists of more diverse Thermoelectric families.
- Data splitting is to be done in such a way that the model only gets chemical information of some compounds, holding back chemical information of others since, for the same compound, all input features except temperature are the same. To address this, we took a candidate set from the same family.
- Hyperparameter modification is highly recommended; otherwise, models may get overfitted; this modification is made in Random Forest for both Lattice thermal conductivity & Seebeck coefficient. Fig. 21 (a) and (b) show the same
- All the target properties in these models have a significant dependence on the processing route; one hot encoded method can be used to provide the model with experimental information.
- Another approach for the prediction of target properties (except for kL) is to consider arbitrary carrier concentration as an input feature, as it depends on temperature & experimental parameters.
- Better Feature selection for improved predictions of Electrical Conductivities.
- Making use of empirical models to get better predictions.

(a)



(b)

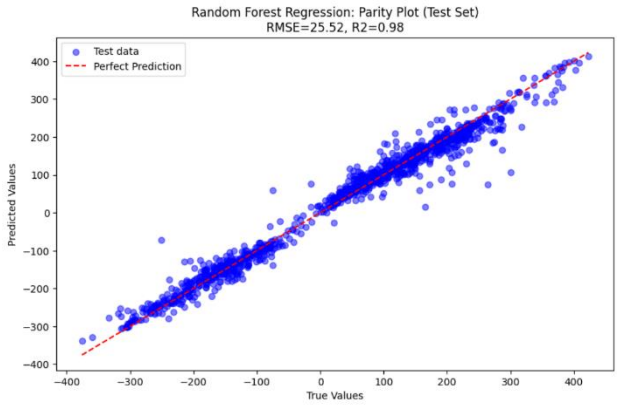


Fig. 21 - Random forest hyperparameter adjustment for **(a)** Lattice thermal conductivity **(b)** Seeback Coefficient

Note: All the figures shown, codes, and the data sets used in the report can be accessed using this link.

Conclusion

We explored two distinct modeling approaches to predict material properties: one using features derived from first principles and the other utilizing simple atomic features. The performance of the first model was limited by data sparsity constraints, leading to suboptimal results. In contrast, the second approach, which employed simple atomic features, performed significantly better on both the training and candidate datasets (from the same family), achieving a strong R^2 score and more accurate predictions.

Moving forward, future developments will focus on incorporating additional descriptors to enhance the electrical conductivity model. Furthermore, expanding our dataset to include more diverse thermoelectric materials will be crucial to improving the robustness and generalizability of our models.

Contribution from the Authors:

Sudeendra R (MM23S001): Conceptualization, Feature Selection (Model-1)

Anudeep Sadarla (MM23M005): Machine Learning Codes (XGBOOST, LR), Fingerprinting.

Atharva Vilas Vyawahare (MM23M006): Planning, Codes (Random Forest), Feature Selection (Model-2).

Sonal Anish Jojo (MM23M021): Fingerprinting, Codes (LASSO, RIDGE).

Syam Sundara Raju Adda (MM23M025): Data extraction, Data mining, Codes (KNN).

References

1. Machine-learning guided discovery of new thermoelectric material – Yuma Iwasaki, Ichiro Takeuchi, Valentin Stanev, Aaron Gilad Kusne, Masahiko Ishida, Akihiro Kirihara, Kazuki Ihara, Ryohto Sawada, Koichi Terashima, Hiroko Someya, Ken-ichi Uchida, Eiji Saitoh & Shinichi Yorozu – https://doi.org/10.1038/s41598-019-39278-z
2. Machine learning models for the lattice thermal conductivity prediction of inorganic materials – Lihua Chen, Huan Tran, Rohit Batra, Chiho Kim, Rampi Ramprasad – https://doi.org/10.1016/j.commatsci.2019.109155
3. Material descriptors for predicting thermoelectric performance – Jun Yan, Prashun Gorai, Brenden Ortiz, Sam Miller, Scott A. Barnett, Thomas Mason, Vladan Stevanović and Eric S. Toberer – https://doi.org/10.1039/C4EE03157A
4. Key properties of inorganic thermoelectric materials – Robert Freer, Dursun Ekren, Tanmoy Ghosh, Kanishka Biswas, Pengfei Qiu, Shun Wan, Lidong Chen, Shen Han, Chenguang Fu, Tiejun Zhu – https://iopscience.iop.org/article/10.1088/2515-7655/ac49dc
5. Machine learning for accelerated prediction of the Seebeck coefficient at arbitrary carrier concentration – H.M. Yuan, S.H. Han, R. Hu, W.Y. Jiao, M.K. Li, H.J. Liu, Y. Fang – https://doi.org/10.1016/j.mtphys.2022.100706