# Benchmarking Machine Learning Models on a Dielectric Constant Database for Bandgap Prediction

Mohammad Hadi Yazdani
Mork Family Department of Chemical Engineering & Materials Science
University of Southern California
yazdanim@usc.edu

Paulo S. Branicio
Mork Family Department of Chemical Engineering & Materials Science
University of Southern California
branicio@usc.edu

Ken-ichi Nomura
Mork Family Department of Chemical Engineering & Materials Science
University of Southern California
knomura@usc.edu

## ABSTRACT

In this study, we investigate the performance of several regression models by utilizing a database of dielectric constants. First, the database is processed using the Matminer Python library to create features, and then divided into training, validation, and testing subsets. We evaluate several models: Linear Regression, Random Forest, Gradient Boosting, XGBoost, Support Vector Regression, and Feedforward Neural Network, with the objective of predicting the bandgap values. The results indicate superior performance of tree-based ensemble models over Linear Regression and Support Vector Regression. Additionally, a Feedforward Neural Network with two hidden layers demonstrates comparable proficiency in capturing the relationship between the features generated by Matminer and the bandgap target values.

## KEYWORDS

Supervised Learning, Linear Regression, Random Forest, Gradient Boosting, XGBoost, Support Vector Machine, Neural Network

## 1 INTRODUCTION

The field of Materials Informatics represent a data-centric methodology aimed at accelerating innovations in materials design and discovery [9]. Currently, an array of open-source software is available for materials scientists and engineers, facilitating the integration of informatics into their research. Notably, Matminer–an open-source Python library designed for materials informatics–has gained popularity due to its extensive suite of tools for data extraction and analysis, robust feature extraction capabilities, and open APIs that provide unrestricted access to online databases of materials data [10].

The Matminer dielectric constant dataset [8] is a comprehensive repository of data encompassing the dielectric properties of over 1,000 inorganic compounds as well as additional attributes such as formation energy, band gap, and melting point. The dielectric constant, also known as the relative permittivity, quantifies the capacity of a material to store electrical energy when subjected to an electric field, making it a crucial parameter for materials in the realms of electronics and energy storage. This dataset is a valuable resource for materials science and engineering researchers focused on developing novel materials for applications such as capacitors, solar cells, and sensors. The dataset was generated using Density Functional Perturbation Theory utilizing the Perdew-Burke-Ernzerhof (PBE) functional.


https://doi.org/10.22369/issn.2153-4136/15/1/2

Density Functional Theory (DFT) is a powerful quantum mechanical theory that accurately describes many material properties at their ground state. Density Functional Perturbation Theory (DFPT) [1] builds upon DFT to incorporate the effects of an external perturbation, such as changes in the electronic structure induced by an external electric field. This extension enables the calculation of a wide range of material properties, such as dielectric constants, phonon frequencies, and piezoelectric coefficients. PBE functional developed by Perdew, Burke, and Ernzerhof, based on the generalized gradient approximation (GGA) [7] is widely used to create reliable materials datasets.

Band gap is a fundamental concept in materials science and solid-state physics that plays a crucial role in determining the electrical and optical properties of a material. It is defined as the energy gap between the top of the valence band and the bottom of the conduction band within a material. Materials with wider band gaps are typically insulators, whereas those with a narrow or nonexistent band gap act as semiconductors or conductors, respectively. Understanding the band gap of a material is essential for designing and optimizing a wide range of electronic and photonic devices, as it determines how the material responds to electrical and optical stimuli.

## 2 METHODS

Linear Regression (LR) is a widely used statistical method for modeling the relationship between a dependent variable and one or more independent variables, specifically capturing their linear correlation. Although LR exhibits a comparatively high model bias, it remains extensively utilized in practical applications ranging from stock price forecasts to the analysis of experimental data, largely due to its strong generalizability and interpretability. Furthermore, LR serves as the cornerstone for numerous sophisticated regression methods, rendering it an indispensable instrument for data analysts and researchers across diverse disciplines.

Random Forest (RF) [2] is a machine learning algorithm widely utilized for both regression and classification tasks. It operates by constructing a set of many decision trees, each generated from a subset of features, thereby ensuring a diverse population of models.

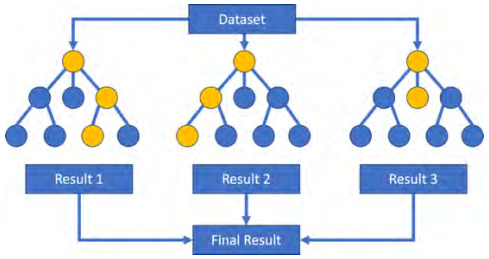


**Figure 1. Random Forest model**

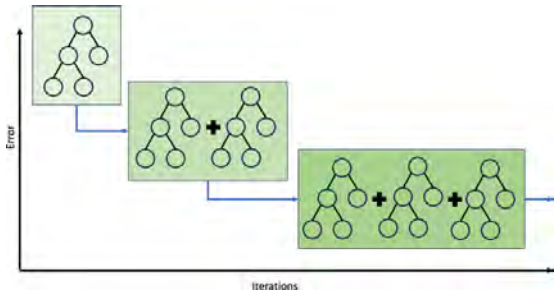


**Figure 2. Gradient Boosting model**

An RF model aggregates the predictions made by each individual tree, using either the mean or mode as its final prediction. RF has been very popular due to its capability of handling high-dimensional datasets with numerous and diverse features. Another significant advantage of RF is its robustness against overfitting, which is a common issue when a single decision tree is trained on a complex dataset.

Gradient Boosting (GB) [5] is a machine learning technique employed for both regression and classification tasks. Similar to RF, the goal of GB is to generate numerous decision trees to cover a large model population. However, GB distinguishes itself by constructing trees sequentially: each new tree is built to correct the errors made by the previous ones. This is achieved by fitting the new tree to the negative gradient of the loss function, which represents the direction in which the model should be adjusted to improve accuracy. This iterative process continues until the model reaches a predefined level of precision. GB's capacity to manage complex datasets and yield highly precise predictions has made it a favored algorithm in diverse domains such as natural language processing, computer vision, and recommendation systems.

XGBoost [3], short for eXtreme Gradieng Boosting, enhances traditional gradient boosting methods through a suite of algorithmic improvements. These enhancements accelerate model training and increase predictive accuracy. XGBoost incorporates several regularization algorithms, such as Shrinkage and Column Subsampling, which help prevent overfitting during tree generation and improve its overall generalization capabilities. Additionally, XGBoost is designed to exploit modern CPU and GPU architectures for computational efficiency. The combination of these enhancements makes XGBoost a highly desired tool for various applications, such as customer behavior prediction in marketing and medical data analysis in healthcare.

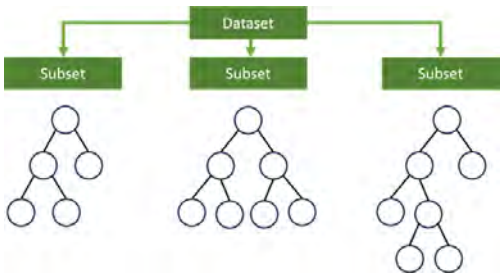


**Figure 3. XGBoost model**

Support Vector Machine (SVM) [4] is a machine learning algorithm designed to optimize the margin between the decision boundary and the nearest data points, known as support vectors, to improve its predictive generalizability on new data. Support Vector Regression (SVR) is a popular regression model based on SVM algorithm that has been implemented in numerous machine learning libraries. SVR utilizes kernel functions, such as the radial basis function (RBF) or a polynomial function, to model non-linear relationships between input features and target variables effectively. These kernel functions facilitate the mapping of input data to a higher-dimensional space, where linear separation is possible. The robustness of SVR makes it suitable for diverse applications across fields such as finance, engineering, and biology.

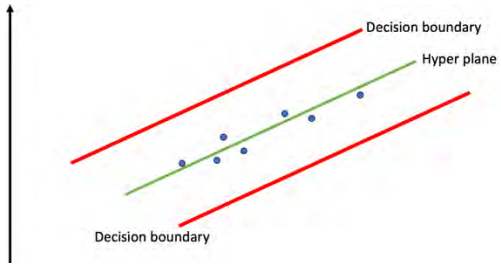


**Figure 4. Support Vector Regression model**

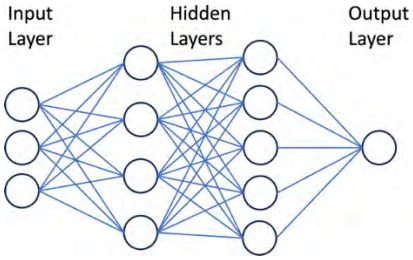


**Figure 5. Feedforward Neural Network model**

A Feedforward Neural Network (FFNN) [6] is a machine learning architecture that comprises multiple layers of nodes or neurons. These layers include an input layer that takes in data, followed by several hidden layers that process the data sequentially, and an output layer that delivers the final prediction. Each neuron in one layer is connected to neurons in the subsequent layer through weights. These weights are iteratively adjusted during the training phase to minimize the discrepancy between the prediction of the network and the actual data. FFNN are renowned their ability to learn complex patterns within datasets and make accurate predictions, making them a popular choice for many machine learning tasks such as image and speech recognition, natural language processing, and financial forecasting.

At first, the dielectric constant dataset from Matminer is used, and the bandgap feature is designated as the target variable. Supplementary input features comprise the chemical formula, the refractive index (denoted as n), the space group (an integer specifying the crystallographic structure of the material), the structure (presented as a pandas Series defining the structure of the material), the number of sites (nsites, representing the number of atoms in the unit cell of the calculation), the volume of the cell, among others.

To enrich the dataset with additional features, specific featurizers from the Matminer library are employed. These featurizers are algorithms designed to extract meaningful information from the raw data, transforming it into quantifiable attributes that can be utilized by machine learning models to improve their predictive performance. The following featurizers are used:

1. *matminer.featurizers.composition.ElementProperty:* This feature extractor calculates elemental properties such as atomic number, atomic mass, atomic radius,

electronegativity, and so on, for a given chemical composition.

2. *matminer.featurizers.structure.DensityFeatures:* This feature extractor calculates various features related to the density of a crystal structure, such as the total volume of the unit cell, the packing fraction, and the Voronoi volume of each atom.
3. *matminer.featurizers.structure.CoulombMatrix:* This feature extractor calculates a matrix of pairwise interactions between atoms in a crystal structure, based on their charges and distances from each other.
4. *matminer.featurizers.composition.OxidationStates:* This feature extractor calculates the most likely oxidation states of each element in a given chemical composition, based on the electronegativity and coordination number of each element.
5. *matminer.featurizers.structure.ElectronicRadialDistribution*: Function: This feature extractor calculates the distribution of electron density around each atom in a crystal structure, as a function of radial distance from the atom.

The dataset comprises 166 potential features and is partitioned into training, validation, and testing sets with proportions of 70%, 15%, and 15%, respectively.

The Scikit-learn library is employed for the construction of the linear regression models. The $R^2$ score and the Root Mean Square Error (RMSE) are utilized as metrics for model evaluation. For the RF model, the ensemble comprises 1,000 trees, as indicated by the number of estimators. The GB model incorporates 100 estimators, adopts a learning rate of 0.2, and a maximum depth of 5, with each new tree intended to enhance the performance of the model incrementally. The XGBoost model is set with 50 estimators. For the SVR model, the RBF kernel is chosen with a gamma of $8\times10^{-7}$ and a margin of error or epsilon of 0.1. The RBF kernel is favored for its efficiency in mapping input features into a higher-dimensional space. The FFNN is designed with two hidden layers, containing 128 and 64 units, respectively, a dropout rate of 0.1, and a mini-batch size of 16. The FFNN is trained using the Adam optimizer over 200 epochs.

## 3 RESULTS AND DISCUSSION

The outcomes of the six models are delineated in Table 1.

**Table 1. Comparative performance of six regression models.**

| Model | Training | | Validation | | Test | |
|---|---|---|---|---|---|---|
| | $R^2$ | RMSE | $R^2$ | RMSE | $R^2$ | RMSE |
| LR | 0.732 | 0.831 | 0.582 | 1.081 | N/A | N/A |
| RF | 0.975 | 0.239 | 0.752 | 0.833 | 0.780 | 0.708 |
| GB | 0.999 | 0.045 | 0.804 | 0.740 | 0.818 | 0.644 |
| XGBoost | 0.999 | 0.029 | 0.783 | 0.778 | 0.818 | 0.643 |
| SVR | 0.943 | 0.384 | 0.535 | 1.140 | 0.405 | 1.164 |
| FFNN | 0.978 | 0.091 | 0.819 | 0.504 | 0.820 | 0.395 |

From the results shown in Table 1 one can note that the LR model is inadequate for capturing the non-linearity inherent in the actual data, as evidenced by its inferior results on the validation set. While the SVR yielded satisfactory outcomes during the training phase, its performance on the validation and testing sets is suboptimal. In contrast, the tree-based ensemble models, i.e., RF, GB, and XGBoost, exhibited superior performance, underlying their capability and efficiency. The FFNN shows demonstrate a comparable performance to the ensemble models displaying robust results on the testing subset.

## 4 CONCLUSIONS

This investigation assessed six regression models applied to a dielectric constant materials dataset, with an emphasis on predicting bandgaps. The results indicate that LR and SVR models yield the least satisfactory results for this application. In contrast, the GB and XGBoost methods as well as the FFNN architecture delivered the most accurate predictions. This demonstrates their superior capacity to learn complex input-output relationships, making them well-suited for tasks requiring high accuracy and the analysis of extensive datasets.

All ML models were implemented on Jupyter Notebook, which substantially increased the student's engagement and comprehension of the algorithms in this project. With the interactive platform and GUI interface, the student could easily evaluate the significance of input parameters in the machine learning algorithm. This project also honed students' analytical skills and hands-on experiences, providing a profound awareness of their potential for materials informatics and real-world engineering applications.

## ACKNOWLEDGEMENTS

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

## REFERENCES

[1] Stefano Baroni, Stefano de Gironcoli, Andrea Dal Corso, and Paolo Giannozzi. 2001. Phonons and related crystal properties from density-functional perturbation theory. *Reviews of Modern Physics* 73, 2, 515.

[2] Leo Breiman. 2001. Random forests. *Machine Learning* 45, 5-32.

[3] Tianqi Chen, and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *KDD'16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794.

[4] Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. 1996. Support vector regression machines. In *Advances in Neural Information Processing Systems* 9. papers.nips.cc/paper_files/paper/1996

[5] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics* 29, 5, 1189-1232.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT press.

[7] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. 1997. Generalized gradient approximation made simple. *Phys. Rev. Letters* 77, 18, 3865.

[8] Ioannis Petousis, David Mrdjenovich, Eric Ballouz, Miao Liu, Donald Winston, Wei Chen, Tanja Graf, Thomas D. Schladt, Kristin A. Persson, and Fritz B. Prinz. 2017. High-throughput screening of inorganic compounds for the discovery of novel dielectric and optical materials. *Scientific Data* 4, 1, 1-12.

[9] Rampi Ramprasad, Rohit Batra, Ghanshyam Pilania, Arun Mannodi-Kanakkithodi, and Chiho Kim. 2017. Machine learning in materials informatics: Recent applications and prospects. *Computational Materials* 3, 1, 54.

[10] Logan Ward, Alexander Dunn, Alireza Faghaninia, Nils E.R. Zimmermann, Saurabh Bajaj, Qi Wang, Joseph Montoya, Jiming Chen, Kyle Bystrom, Maxwell Dylla, Kyle Chard, Mark Asta, Kristin A. Persson, G. Jeffrey Snyder, Ian Foster, and Anubhav Jain. 2018. Matminer: An open source toolkit for materials data mining. *Computational Materials Science* 152, 60-69.