REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

ESTIMATING THE BAND GAP OF MATERIALS WITH MACHINE LEARNING METHODS

Aydın EROL

MASTER OF SCIENCE THESIS

Department of Physics

Physics Program

Supervisor

Assoc. Prof. Dr. Seçkin Dündar GÜNAY

June, 2022

REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

ESTIMATING THE BAND GAP OF MATERIALS WITH MACHINE LEARNING METHODS

A thesis submitted by Aydın EROL in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE is approved by the committee on 23/06/2022 in Department of Physics, Physics Program.

Assoc. Prof. Dr. Seçkin Dündar GÜNAY
Yıldız Technical University
Supervisor

Approved By the Examining Committee

Assoc. Prof. Dr. Seçkin Dündar GÜNAY, Supervisor
Yıldız Technical University ____________________

Prof. Dr. Çetin TAŞSEVEN, Member
Yıldız Technical University ____________________

Asst. Prof. Dr. Utku CANCI, Member
Istanbul Gedik University ____________________

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Estimating the Band Gap of Materials with Machine Learning Methods supervised by my supervisor, Assoc. Prof. Dr. Seçkin Dündar GÜNAY. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Aydın EROL

Dedicated to my family

ACKNOWLEDGEMENTS

Firstly, I would like to offer my gratitude to my supervisor, Assoc. Prof. Dr. Seçkin Dündar GÜNAY, for encouraging and helping me to determine an interdisciplinary subject for my master's thesis, for his valuable feedback and tremendous support towards this thesis, and lastly, for his patience and advisory towards me.

My gratitude extends to all my teachers in the pre-academy, and professors at both Istanbul University and Yıldız Technical University, for guiding me into becoming a better individual and a more knowledgeable researcher.

I also would like to thank Prof. Dr. Özlem ÖZ for her counseling that impacts both my academic career and personal life, which I wish to continue lifelong. I also immensely appreciate both material and spiritual support me and my family received from my one and only cousin, Nil Sidem ERŞEN. I would also thank my friends for their company. Lastly, I would like to thank my family for their discretion and loving support.

Aydın EROL

TABLE OF CONTENTS

LIST OF SYMBOLS

f	Activation function
y	Actual value of target variable
R^2_{adj}	Adjusted coefficient of determination
b_j	Bias for jth layer
R^2	Coefficient of determination
$\hat{y}_i$	Estimated value of individual i
$\hat{y}$	Estimated value of the target
x	Feature vector
x_i	Input feature vector of individual i
K	Kelvin
$\overline{x}$	Mean value of feature values
$\overline{y}$	Mean value of observations
n	Number of features in the test data set
N	Number of neurons from the previous layers
y_j	Output value for jth neural layer
T	Temperature
e_i	Vector of residuals of individual i
w_{ji}	Weight multiplier for ith neuron and jth layer

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
BCC	Base Centered Cubic
DFT	Density Functional Theory
eV	Electron Volt
FCC	Face Centered Cubic
GGA	General Gradient Approximation
HHI	Herfindahl-Hirschman Index
ICSD	Inorganic Crystal Structure Database
KRR	Kernel Ridge Regression
LDA	Local Density Approximation
MAE	Mean Absolute Error
Magpie	Materials Agnostic Platform for Informatics and Exploration
MAST-ML	Materials Simulation Toolkit for Machine Learning
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
OQMD	Open Quantum Materials Database
Pandas	Python Data Analysis Library
PBE	Perdew-Burke-Ernzerhof
PCA	Principal Component Analysis
RBF	Radial Basis Function
RF	Random Forest
RL	Reinforcement Learning
RMSE	Root Mean Squared Error
SSE	Sum of Squared Errors
SSR	Sum of Squared Residuals
STP	Standard Temperature and Pressure
SVM	Support Vector Machine
TSS	Total Sum of Squares
VASP	Vienna Ab Initio Simulation Package
WEKA	Waikato Environment for Knowledge Analysis

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Estimating the Band Gap of Materials with Machine Learning Methods

Aydın EROL

Department of Physics

Master of Science Thesis

Supervisor: Assoc. Prof. Dr. Seçkin Dündar GÜNAY

Methods of machine learning have shown significant progress in the last decade. The number and quality of applications of these methods in various fields of physics are also increasing. Machine learning algorithms are mathematical models that can learn patterns in a data set and estimate the values of the target label afterward. The selection and optimization of a learning algorithm depend on the problem and the structure of the used data set. Processing this data set and selecting features before training a model is also important.

Predicting the band gap of different types of materials with machine learning methods while investigating methods that are used for model optimization is the main objective of this thesis. Knowing the band structures is especially important in environmental technologies, such as solar panels or light-emitting diodes. They consist of semiconductor devices, and like all semiconductor materials, their band gap determines their conductivity. Forecasting material properties in physics is challenging because of the long work hours and computational resources required to dedicate to related experiments or simulations. Machine learning can offer a solution to these problems and increase overall efficiency by decreasing workloads. The proposed random forest model was developed and optimized in Python programming language. A custom feature selector algorithm that utilizes multiple metrics as a feedback tool to select optimal features also improves the performance of the final model. Results show that the optimized random forest model can predict band gaps of the materials in the Citrine and Matminer data sets with less than the mean absolute error value of 0.500 eV and an R^2 score higher than 0.800.

Keywords: Band gap, random forest, machine learning, solar cell, semiconductor

ÖZET

Malzemelerin Bant Aralığının Makine Öğrenmesi Yöntemleriyle Tahmin Edilmesi

Aydın EROL

Fizik Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. Seçkin Dündar GÜNAY

Makine öğrenmesi yöntemleri son on yılda kayda değer ilerleme göstermiştir. Bu yöntemlerin fiziğin çeşitli alanlarındaki uygulamalarının sayısı ve niteliği de ayrıca artmaktadır. Makine öğrenmesi algoritmaları bir veri setindeki örüntüyü öğrenebilen ve sonrasında hedef alınan etiketin değerlerini tahmin edebilen matematiksel modellerdir. Öğrenebilen bir algoritmanın seçilmesi ve en iyi hale getirilmesi, probleme ve kullanılan veri setinin yapısına bağlıdır. Bir modeli eğitmeden önce bu veri setinin işlenmesi ve özelliklerin seçilmesi ayrıca önemlidir.

Makine öğrenmesi yöntemleri kullanılarak materyallerin bant aralığını tahmin ederken modeli en iyileştirmede kullanılan yöntemlerin incelenmesi bu tezin ana amacıdır. Bant yapılarının bilinmesi, güneş panelleri veya ışık yayan diyotlar gibi çevresel teknolojiler için ayrıca önemlidir. Bu gibi teknolojiler yarı-iletken materyallerden oluşur ve tüm yarı-iletkenler için geçerli olduğu üzere materyallerin bant aralıkları bu materyallerin iletkenliğini belirler. Materyal özelliklerinin tahmin edilmesi, ilgili simülasyonlara veya deneylere uzun mesai saatlerinin ve bilgisayar kaynaklarının ayrılmasını gerektirmesi nedeniyle zorlayıcıdır. Makine öğrenmesi bu problemler için gereken iş yükünü azaltarak genel verimin artması için bir çözüm sunabilmektedir. Bu tez çalışmasında önerilen rastgele orman modeli Python programlama dili kullanılarak geliştirilmiş ve en iyi hale getirilmiştir. Birden çok hata ölçüsünü bir geri bildirim aracı olarak kullanan özel yapım özellik seçici algoritma son modelin performansını iyileştirmiştir. Sonuçların gösterdiği üzere Citrine ve Matminer veri setlerindeki materyallerin bant aralıkları, en iyileştirilmiş rastgele orman modeli tarafından 0.500 eV'tan az bir ortalama mutlak hata ve 0.800'den yüksek R^2 puanı ile tahmin edebilmektedir.

Anahtar Kelimeler: Bant aralığı, rastgele orman, makine öğrenmesi, güneş hücresi, yarı-iletken

1
INTRODUCTION

1.1 Literature Review

Machine learning (ML) is a relatively new research field that became popular in the last decade. As the name itself suggests, ML algorithms recognize patterns in the data by learning the relationship between the features. This process is called training, and training is satisfied over a given set of data. The number of research for the applications of ML methods in various science fields is significantly increased in the last decade, parallel to the rise in computational power and many competitions in computer science. On the other hand, applications of these methods in different fields of physics escalating in the last years. ML algorithms act as a predictor, and they estimate a value or values depending on the given data. These algorithms learn from the features in the data to predict values. The result of a learning algorithm tends to diverge from optimal results when there are radical data points. The impact of outliers creates an overdependency on clean and numerous data. There is no skeleton key in ML –yet– and a single algorithm won't yield the best results for every problem. ML models have parameters that optimize the model to compensate for the issues regarding performance. There are various algorithms–such as random forest [1] (RF) and multilayer perceptron [2] (MLP)–which offer practical solutions to different real-world problems. These problems can vary for each discipline–classifying a tumor for medicine [3], making a steering decision according to road curves for self-driving cars [4], [5], estimating the absorbed radiation dose of passengers in a flight [6]–which all represent a prediction problem for ML.

In physics, a solid material has different features such as shear modulus, electronegativity, or band gap. Most of these features have a specific method or a simplified formula to calculate them or a simple algorithm to range their values. In condensed matter and material physics, ML methods can be and are used to predict different experiment results like bulk modulus, elastic modulus, or the band gap of a solid [7], [8], [9].

In material physics, researchers try to generate a material that yields maximum efficiency for the task at hand. This process, however, has unpredictable costs such as time and money due to the methods and computational burden of the whole process. ML methods can learn the relation between the material features and construct a generalizable model applicable to different materials with similar characteristics. Therefore, utilizing ML methods can reduce the time required to produce a material compared to conventional trial & error methods.

Calculating the band gap of a material is a complex problem that doesn't have a simplified formula. One of the methods that are used to estimate materials' band gap is Density functional theory (DFT). DFT is a mathematical model developed in 1964 [10] and consists of different mathematical methods which describe a quantum-mechanical system. Some of the experiments and simulations in the literature utilize DFT to calculate atomic properties or electronic structure successfully, thus making a significant contribution by enlarging the data set for a material [11]. Although DFT is successful when calculating various physical properties, some reports indicate that it may miscalculate the band gap of materials by a certain degree [11], [12], and calculations take time.

DFT can be combined with different approximations such as local density approximation (LDA) [13], generalized gradient approximation (GGA), or Perdew-Burke-Ernzerhof (PBE) approximation [14], which are also successful when estimating material properties. These approximations, however, may underestimate or overestimate the band gap when used separately [12], [15], [16], [17], [18]. In literature, ML and mathematical approximations blended to determine the band gap [16], [19], [20], [21] for various material-based problems, such as lithium-ion batteries [22], light-emitting diodes, or solar cells [23].

Chemical compounds of a solar cell material that may yield high power conversion efficiency can also be estimated using ML methods [24]. According to Raccigulia et. al. [25], failed material production experiments can be predicted and recycled via ML. A support vector machine (SVM) model, a method of ML, was used to predict the necessary conditions for a chemical reaction experiment to achieve success and yield a product. A C4.5 decision tree algorithm from the WEKA library was used to enhance the SVM. The developed model predicted experimental results with an accuracy of 78%, and the experiments using these new conditions met a success rate of 89%.

In the past years, quantum computer technology has been utilized to solve physics problems effectively and to compute complex ML operations, where the resource of a classical computer became insufficient to solve. Reinforcement learning (RL) is a form of ML. Unlike most ML methods, where the algorithm explicitly told how to solve the problem, RL algorithms find the optimum way by traditional trial and error method to yield a solution for the task at hand. RL algorithms converge towards a maximized numerical reward signal after each action in training [26]. Like the ML, RL is also applicable to quantum computers, but the learning process still takes time. According to Saggio et al. [27]–which is just one example of interdisciplinary work between computer science and quantum physics–required learning time can be reduced using the superposition principle in quantum mechanics. This new learning protocol allows the algorithm to switch from classical to quantum epoch to receive active feedback during learning and check whether it performs better or worse. It does not limit the algorithm to evaluate action at the end of the epoch.

1.2 Objective of the Thesis

The main purpose of this thesis is to estimate the band gap values of various materials by utilizing machine learning methods. Investigating the performance of the models is executed by comparing the scores of different metrics. The algorithm libraries that are capable of finding optimal parameters and the methods that manage feature selection will also be in the aspect of this thesis. The second objective of this thesis is to develop a custom feature selector algorithm that maximizes final model performance. The contribution of the findings to the literature will be reported. Another aim of this thesis is to design a program in Python language depending on the codes in Machine Learning Lab Module [7] and Materials Simulation Toolkit for Machine Learning (MAST-ML) [28] in the NanoHub. Codes will also be published to contribute to the growth of open-source programming on cloud platforms, such as Kaggle and NanoHub.

1.3 Hypothesis

The main hypothesis of this thesis is an ML model can be developed to be trained in minutes and predict band gaps in a data set within seconds without discriminating the type of materials – whether it's a semiconductor or an insulator. The second hypothesis of this thesis is that the developed feature selector algorithm depends on specific statistical measures that can list the optimal features and can overperform popular feature selector algorithms in the literature for the band gap prediction task.

2
GENERAL INFORMATION

2.1 Machine Learning

The concept of learning depends on the feedback obtained during a search for a pattern in data. Patterns exist in different parts of life, like this paragraph having an intro and an outro to keep the multiple texts attached to the whole page. Living beings such as humans, dogs, or bugs can learn the pattern of an event by processing neurological signals originating from organic sensors. Depending on the temperature, touching a pan on a stove can be painful or harmless. If the temperature is high, the brain receives a signal indicating pain, which acts as a negative reward. Treating food to pets acts as a feedback instrument and corresponds to a positive reward for the action. Any learning process, including ML, can be induced by a simple algorithm: whether the reward for action is positive, which promotes repetition of the behavior, or the reward multiplier can be 0 or negative, which means no logical reason to continue acting the same. People are good at generating alternative actions based on feedback. As human life starts in the womb, so does data mining in the brain. During the course of life, the human brain receives and processes a continuous flow of data obtained through the senses.

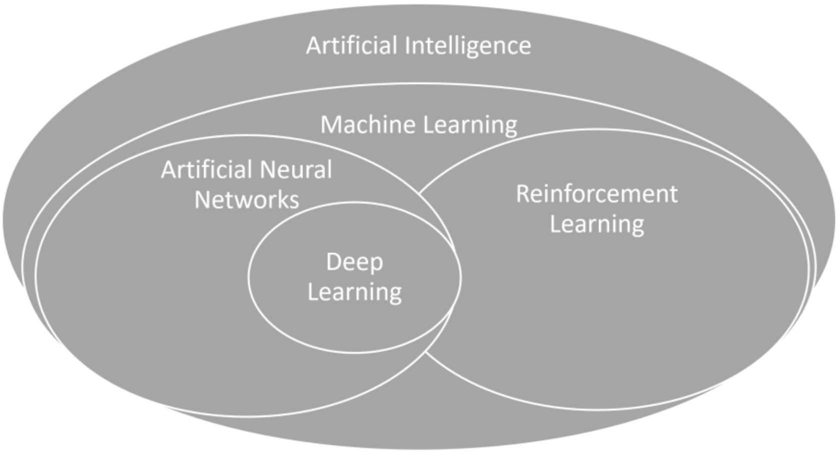

Figure 2.1 Hierarchical description of the relation between computational methods.

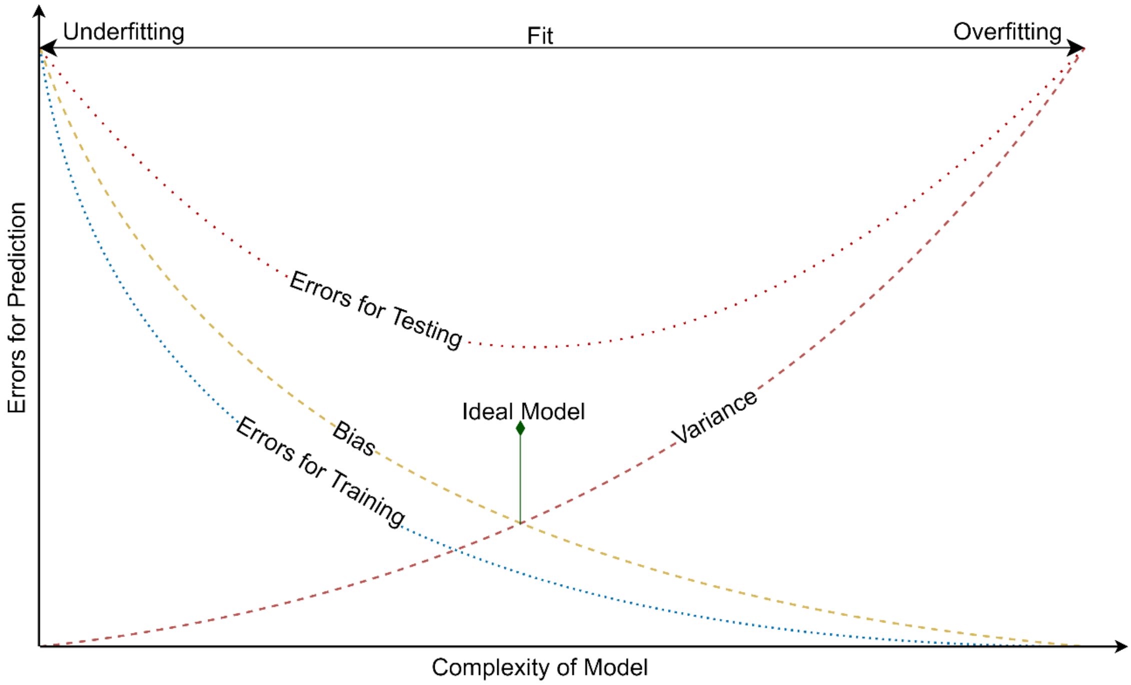

Figure 2.2 Optimum model achieved with balanced learning and tuned hyperparameters.

Mathematical algorithms that can mimic some of the brain's capabilities–learning, decision making, and acting–are called artificial intelligence (AI). ML is a subsection of AI, as illustrated in Figure 2.1. Since humans have various methods to learn and obtain knowledge, so does ML. Learning is essentially a mathematical process where features are weighted or biased depending on their relevance in the pattern to achieve the best fit. Fit is a mathematical line used to represent a model. Overfitting and underfitting are the most common problems of ML that decrease the performance of a model. Both these problems address how a model fails to match the data. Overfit or underfit decreases the accuracy of predictions. Overfit expresses the poor performance on the unseen data even though the model overperforms on the training data. Models that overfit are complex, and their generalizability is compromised. Underfit occurs when the fit is not descriptive enough to match all the data. It refers to the poor model performance both in the training and the testing due to insufficient learning. These problems can occur depending on the variance or bias of the model. Variability of a model prediction can also be referred to as variance. Variance and bias are inversely proportioned. Model bias changes the accuracy of predictions obtained by a model while disregarding or heavily weighting some of the data. An ideal ML model has a balance between its variance and bias, as shown in Figure 2.2. Some algorithms are more robust to underfitting characteristically. Parametric algorithms such as linear regression have a higher bias with a lower variance while algorithms that are flexible enough, such as RFs, yield lower bias and higher variance when compared with the decision tree and linear regression.

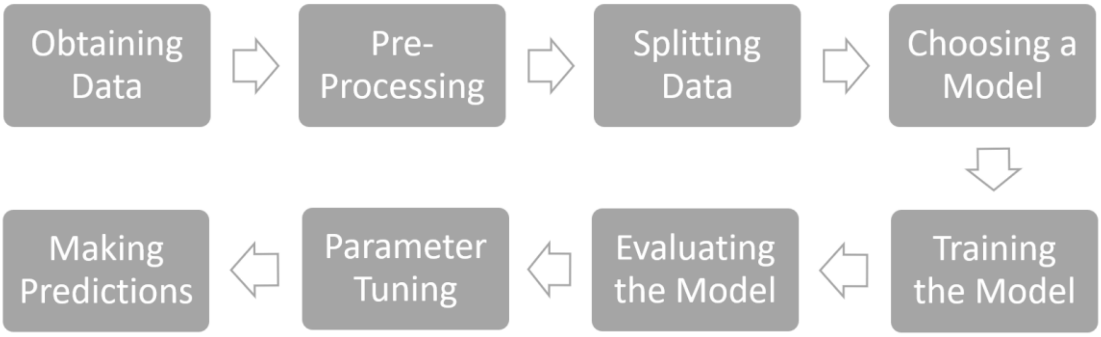

Figure 2.3 Workflow for machine learning model.

Training a machine is a process with many limits, mainly due to a lack of data. The main goal of model training is to create a model that accurately predicts a set of values. Predictions are outputs of a learning algorithm that completes a learning cycle, also called an epoch. Obtaining relevant data is the starting point of ML workflow, as shown in Figure 2.3.

Intelligent creatures can try different actions by instinct. Since machines lack human instincts, they must be enforced to try different settings to avoid getting stuck in an endless loop of a non-resulting action that doesn't yield a satisfactory result. One way of forcing the machine to search for different patterns is using randomization methods. Algorithms create consistent results after fixing the randomization to a specific numeric value, named seed. The importance of seed becomes evident when comparing different results obtained for models by changing their configurations or changing training data which alters model parameters.

ML algorithms that discover patterns to predict values can be addressed in three different groups depending on the data used to train a learning algorithm: reinforcement, unsupervised, and supervised learning [29]. The supervised learning method calculates the mathematical relation and the mapping function using input variables and an output variable. Supervised learning tasks can be examined in two subsections: classification and regression. Classification is a problem of predicting the values of at least two discrete labels, such as looking at a set of pictures and guessing which resembles a bicycle more than a car or motorbike. Regression, on the other hand, is a problem of predicting continuous label values in a data set, such as prices, years, etc. Classification problems consist of more than one regression. There are different algorithms for regression: SVM regression, linear regression, or RF regression.

The data set must be processed and cleansed from missing or irrelevant elements before being fed into a learning algorithm since it affects the model performance significantly. Even small changes in the handling of data may affect the results. Pre-processed data should decrease the model errors and increase the efficiency of the training when fed to any learning algorithm. There are different parts of pre-processing, such as data cleaning, data transformation, or data reduction. Missing values in the data set should either be excluded or filled manually during the data cleaning process. Data transformation describes the process of normalization or scaling values within a specific range. Learning can also be affected by having too much irrelevant data. Reduction in the dimensions of the data set leads to fewer parameters, which consequently reduces the computational burden. Principal Component Analysis (PCA) is an old but widely used ML algorithm for dimensionality reduction problems. Identifying and dropping the highly correlated features also increases model performance.

One other step of data pre-processing is splitting the data. Testing the model with the data that the model has already trained on would be unreliable. A processed data set must be divided into two or three subsets before the training of any ML algorithm: training, validation, and testing data sets. Splitting the data into training and validation sets can be satisfied by using a ratio of 75:25 or alike. Using seed, one can obtain the same data splits. Training and validating multiple algorithms using the same data split is crucial when comparing the results of different models.

Model evaluation represents the generalizability of the final model. ML models are validated by evaluating how the model behaved when tested with an independent set containing unseen data. One of the popular techniques in ML is k-fold cross-validation, and it allows training and testing a model using different subsets created k-times. Predictions obtained using cross-validation data sets can be averaged, contributing to a better balance between variance and bias. Performance evaluation metrics demonstrate the model performance.

Models that use their default configurations may not perform optimally enough during or after training. ML models have hyperparameters that allow customizing the model to improve its performance. This process is named hyperparameter tuning and is one of the steps for obtaining an ML model that performs well. Hyperparameters differ from the learning parameters calculated by the model on the training process. Hyperparameters act as coefficients and are set manually to lead the model during learning. Algorithms may share the same hyperparameters depending on the similarities between them. While tree-based models share the same hyperparameter for the number of trees, neural network-based models share the number of hidden layers. Searching for the best set of hyperparameters in the search space is called hyperparameter tuning. Each dimension in search space volume represents a hyperparameter, and each point represents a model configuration. Hyperparameter tuning aims to find the vector that represents the most optimal point that produces minimum error after the training. Searching for optimal parameters is an exhaustive process to be executed manually. Various optimization algorithms and libraries can optimize models by tuning a dictionary of hyperparameters. Random search and grid search are two popular algorithms for model optimization.

2.2 Regressors

2.2.1 Linear Regression

Linear regression (also known as ordinary least-squares) is an old mathematical analysis method for analyzing the relation between numerical input and output values. Depending on the number of input variables, linear regression is referred to as simple linear regression or multiple linear regression. Linear regression models are implemented in the ML because the idea of linear regression itself is trivial. Many real-life problems can be explained using a linear relation, such as the increase in fuel consumption depending on the throttle position, grades of a student depending on work hours, or density of a semiconductor material depending on volume. Depending on the independent value (or feature) x, linear regression can predict dependent variable y.

$$y=\beta_0+\beta_1 x \tag{2.1}$$

Here the intercept (value of y when x=0) is represented by the β_0, and β_1 stands for the slope coefficient for the line.

$$\beta_0=\bar{y}-\beta_1\bar{x} \tag{2.2}$$

$$\beta_1=\frac{\sum_{i=1}^{n}(x_i-\bar{x})\left(y_i-\bar{y}\right)}{\sum_{i=1}^{n}(x_i-\bar{x})^2} \tag{2.3}$$

Here $\bar{y}$ represents the mean of prediction values y_i, and $\bar{x}$ is the mean value of independent variables.

2.2.2 Kernel Ridge Regression

Being one of the models developed based on linear regression, Kernel ridge regression (KRR) merges the ordinary linear least squares method with the l2-regularization and a kernel–which is a matrix. KRR adds an error term to Equation (2.1). This error term corresponds to a value needed to correct predicted values and penalize errors. Error term also adds bias to the least-squares line, thus reducing variance. KRR is applicable when there are a few features, and these features correlate enough to explain prediction values. The type of kernel or strength of regularization (alpha) can be adjusted when tuning hyperparameters.

2.2.3 Decision Tree

The decision tree is an ML algorithm developed by Ross Quinlan [30], and they construct a tree of rules to predict label values for classification or regression problems. A decision tree starts with a root node. Then, it creates an inductive tree of rules, evaluating the relationship between the features in a data set. Each non-terminal node in the tree is associated with a feature value. The data set is processed starting from the root node. Nodes create rules by discriminating features one by one. Decision nodes connect via branches until leaf nodes. The decision tree algorithm forms multiple nodes for each possible outcome until it reaches a leaf node. Leaves of a decision tree represent a terminal node, which yields outputs of the algorithm and represents class or classes.

2.2.4 Random Forest

An RF algorithm consists of multiple decision trees combined with bootstrap aggregating or bagging for short [31]. As the name implies, the bagging method randomly resamples the data into smaller bags to train trees with a lower variance. Then, bootstrapping distributes the features to most of the trees so that the averaged results aggregate into a single prediction. The predicted class or value is obtained by averaging the values of bootstrapped aggregated decision trees. Bagging improves model performance by decreasing the variance of the model. RF models can be optimized through their hyperparameters, whether to be precise or to be fast. The number of trees (n_estimators) that are generated by the learning algorithm can be specified, or the randomization in the trees (random_state) can become consistent by passing a seed.

2.2.5 Multilayer Perceptron

The concept of a perceptron was firstly introduced by Rudolf Carnap in 1936 [32]. The mathematical theory of a perceptron has been developed by Frank Rosenblatt, inspired by biological (retina) neurons, in 1958 [2]. Both biological and artificial perceptrons (neurons) process the information in three steps: take the input, evaluate information in the neuron, and create an output signal. These steps are called layers: input layer, hidden layer, and output layer. A neural network is constructed after calculating the coefficient before passing the signal coming from the input layer to the output neuron. The correlation between neurons in layers is affected by bias and weight factor assigned to each connection [33]. Weight and bias values are randomly assigned before the training begins and updated as the learning progress. This optimization process is called stochastic gradient descent. The method that updates weights using backpropagation of errors is called backpropagation. Thus, each layer (except the input layer) has a correlation-based mathematical calculation of weight and bias values for previous layers. Bigger weight values carry a significant influence on the change of the output. MLPs have become the fundamentals of artificial neural networks.

$$y_j = f\left(\sum_{i=1}^{N} w_{ji} x_i + b_j\right) \tag{2.4}$$

The output of the j^th neuron layer with N number of neurons, y_j, can be calculated using Equation (2.4). Here x_i is the input vector from previous layer, w_{ji} is the weight factor, b_j is the bias value assigned to the hidden layer, and f is an activation function. Adding bias to each weighted neuron changes the position of model fit by shifting it. The effect of the bias is similar to linear models, but neural network algorithms generally create non-linear models. The output neurons of the j^{th} layer are calculated by multiplying the sum with an activation function (sigmoid, tangent, linear, or non-linear).

Similar to other ML models, neural network models also have hyperparameters that can be used to optimize their performance. A few of these optimizable parameters are optimizer type, the number of hidden layers (hidden_layer_sizes), activation function (activation), learning rate, and momentum coefficient. Neural network models can be trained for a specified number of epochs or stopped earlier to decrease computational time loss when error metrics stop improving. The activation function also determines the shape of the output neuron or neurons.

2.3 Metrics

The performance and generalizability of an ML model are optimized and evaluated through metrics. Classification and regression are similar but different problems of ML. Performance metrics also cluster depending on the type of the problem. Root mean squared error (RMSE), mean absolute error (MAE), mean squared error (MSE), and coefficient of determination (R^2) are just some of the metrics which commonly used to assess regression models. In supervised learning, targets in the data set are actual values and can be used for statistical calculations to evaluate how well the model performs for its predictions. Metrics are composed of statistical methods that calculate the variance between the actual values in the data and the predicted values obtained by the ML model [17], as shown in Figure 2.4. The relative relation between metric scores is illustrated in Figure 2.5.

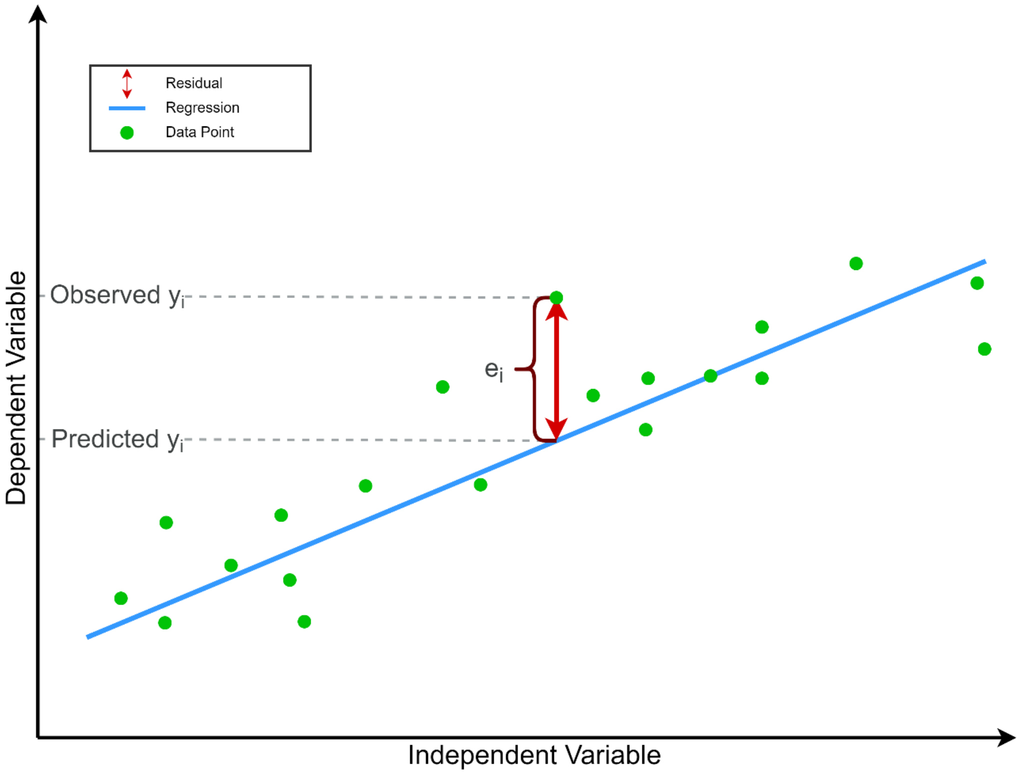

Figure 2.4 Illustration of the fit obtained by a linear regressor model. Metrics evaluate the model performance by calculating the residual e_i.

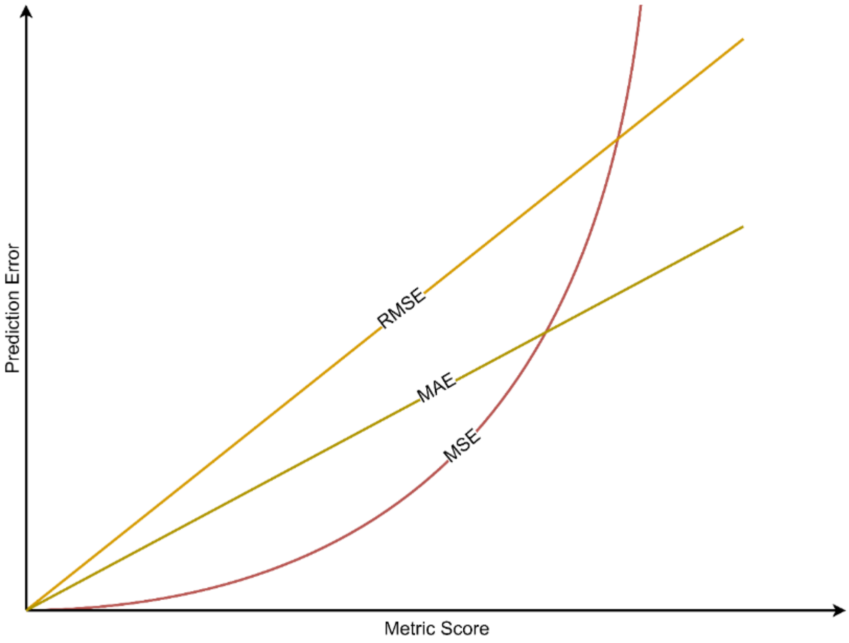

Figure 2.5 The relation between the prediction errors and error scores obtained using ML metrics.

2.3.1 Sum of Squares

Before diving into the mathematics of widely used metrics mentioned above, it would be better to address fundamental statistical concepts first. The error is the difference between the actual and the expected value. In ML terms, the vector of residuals e_i represents the variation between the estimated and the actual value of the target label.

$$e_i = y_i - \hat{y}_i \tag{2.5}$$

Here $\hat{y}_i$ represents the predicted value, and y_i is the actual value corresponding to the predicted variable. One can calculate the sum of errors for the n number of predictions using the following formula:

$$SE = \sum_{i=1}^{n} (y_i - \hat{y}_i) \tag{2.6}$$

The sum of squared errors (SSE) uses the squared difference between actual and predicted values. Penalization of larger errors is a benefit of squaring factor. SSE for prediction-actual value pairs averaged for n number of predictions is given by:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2 \tag{2.7}$$

The sum of squared residuals (SSR) represents the sum of squared deviation between the predicted data $\hat{y}_i$ and the mean of actual values $\bar{y}$:

$$SSR = \sum_{i=1}^{n} (\bar{y} - \hat{y}_i)^2 \tag{2.8}$$

Total sum of the squares (TSS, or SST) is defined as a sum over all squared errors for the set of observations y_i and their mean $\bar{y}$ [34].

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{2.9}$$

TSS can be rewritten as a combination of SSR and SSE:

$$TSS = SSR + SSE \tag{2.10}$$

2.3.2 Mean Absolute Error

MAE represents the magnitude of the errors for continuous variables [35], and optimal prediction should be in the median. MAE is a simple and computationally lightweight error calculation method. It is also a linear metric that scales proportionally to the data. MAE is the sum of (positive) distances between predicted value $\hat{y}_i$ and the observation value y_i averaged over n number of data:

$$\mathrm{MAE}=\frac{1}{n}\sum_{i=1}^{n}\left|y_i\text{-}\hat{y}_i\right| \tag{2.11}$$

2.3.3 Mean Squared Error

MSE is an average of the square of errors. Due to squaring, MSE penalizes large errors more than small ones. Ideal prediction lies in the median value. Squaring also makes MSE vulnerable to outliers or noisy data sets. MSE of a predictor algorithm is calculated by diving the SSE by n number of samples:

$$\mathrm{MSE}=\frac{1}{n}\sum_{i=1}^{n}\left(y_i\text{-}\hat{y}_i\right)^2=\frac{1}{n}\sum_{i=1}^{n}e_i^2=\frac{\mathrm{SSE}}{n} \tag{2.12}$$

2.3.4 Root Mean Squared Error

The squaring factor of MSE shows a sensitivity for outliers. RMSE is calculated over the square root of the MSE to compensate for its downsides. RMSE is more robust to outlier values compared to MAE and MSE. RMSE is also one other metric that scales with the data like MAE, giving an error in units of the target variable. Hence the lower RMSE score may indicate a lower error.

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i\text{-}\hat{y}_i\right)^2}=\sqrt{\mathrm{MSE}} \tag{2.13}$$

2.3.5 R^2

R^2 is one of the widely used regression metrics, also referred to as the coefficient of determination, and ranges between -∞ and 1. R^2 gauges the error in the fit of the regression line. If the R^2 is close to 0, then fitted line is horizontal, which indicates that the model will not be able to predict the target variable. If R^2 is close to 1, then the obtained fit may represent the model flawlessly. The goodness of fit can be evaluated via R^2. R^2 can be calculated using SSE and TSS.

$$R^2=\frac{SSR}{TSS}=1-\frac{SSE}{TSS} \tag{2.14}$$

2.3.6 Adjusted R^2

When the number of terms increases, so does the R^2 score. But this can be misleading because a model with more terms doesn't necessarily yield better results when the model may not learn from new features. Adjusted R^2 (R^2_{adj}) recalculates the R^2 considering the usefulness of an independent variable added to the model later. For example, atomic density has a linear relationship with the atomic mass because both variables are dependent and correlated, meaning any change in the atomic mass will affect density. A linear model can easily predict the density values depending on the atomic mass. Error in predictions using this model will decline even further when another variable that density depends on, such as atomic volume, is introduced to the model. In this case, both R^2 and R^2_{adj} would improve. However, R^2 may still increase when the atomic electronegativity is added to the predictor model, even though the relation between two variables does not influence the training positively. R^2_{adj} only increases after adding useful features that contribute to real improvements. R^2_{adj} is a metric that can be used to estimate model performance depending on the variation of k amount of independent variables for n number of observations.

$$R^2_{adj}=1-\left(\frac{\left(1-R^2\right)(n-1)}{n-k-1}\right) \tag{2.15}$$

2.4 WEKA

Waikato Environment for Knowledge Analysis, WEKA, is a data mining software with ML tools [36]. WEKA is a powerful tool for ML since it can cover ML problems such as classifying, regression, clustering, and association. One can use it to analyze a data set in a short time and with different aspects: which models yield better metrics, the correlation between features, forecasting values, and such. Most of the hyperparameters of ML models can be changed and experimented with using WEKA.

2.5 Band Gap

Atoms in materials consist of discrete energy levels. When the distance between the number of atoms decreases, they form continuous energy levels named bands because atomic orbitals overlap with the orbitals of nearby atoms. Electrons in the outer orbitals of an atom are named valence electrons, and they create the valence band. The next energy band is called the conduction band. The energy gap between these two bands that no electron can occupy due to the Pauli exclusion principle is called the band gap. A material is said to be a conductor when the electrons jump from the valence band into the conduction band. The width of the band gap determines the electrical property of a material. Metals in the periodic table have overlapping bands, which means having no band gap for electrons to overcome. Material is an insulator if the gap is too large, and electrons cannot jump to the conduction band. The allowed energy bands of an insulator are either full or empty, and no electrons can be moved continuously by an electric field without disrupting the electronic structure. Semiconductor materials have a narrow band gap between two partially filled bands. Figure 2.6 shows a simplified illustration of the electronic bands mentioned above.

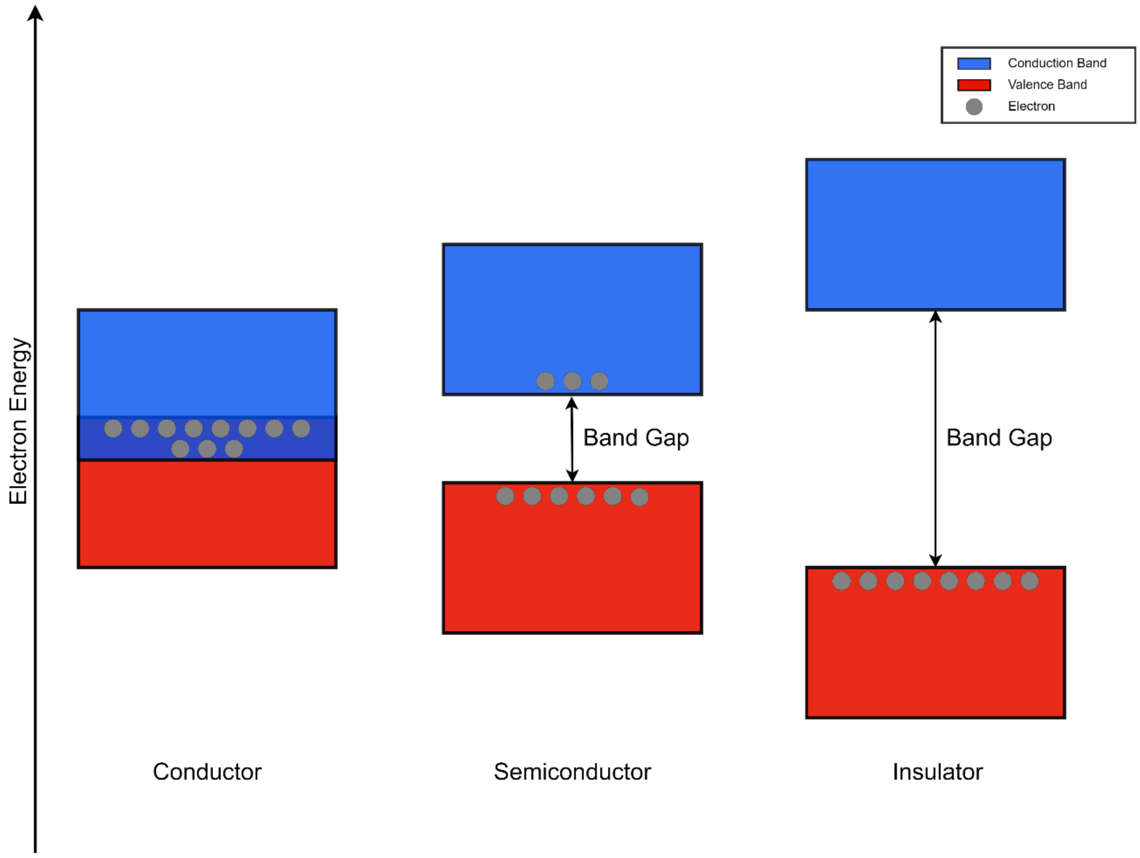

Figure 2.6 Illustration of band gaps for conductor, semiconductor, and insulator materials. Valence and conduction bands of materials at temperatures above absolute zero are filled with the relative number of electrons for each band structure.

2.5.1 Semiconductors

Devices like diodes, transistors, switches, detectors, and photovoltaic cells are based on semiconductors [37]. Materials like Silicon (Si), Germanium (Ge), Gallium (Ga), and Arsenide (As) are a few of the commonly used semiconductors in many daily life technologies like sensors, LEDs, or solar cells. Semiconductors can conduct electricity when an electron receives enough energy to overcome the band gap and move from the valence band into the conduction band. This influence that required for excitation of electron may be optical or thermal. When a negatively charged electron is excited into the conduction band, it leaves an empty valence orbital, a positively charged hole behind because of the absence of an electron. A hole can move in a crystal lattice in the opposite direction of the electron. This process is also reversible, which means that a conduction electron can release energy to de-excite and recombine with a valence hole. Moving electrons and holes are charge carriers that conduct electricity.

A semiconductor material with four valence electrons forms perfect covalent bonds with the other four neighboring atoms and creates a pure crystal lattice without any free electrons. Impurities or doping means intentionally decreasing the impurity of semiconductor material [37] and used to manipulate the material properties, such as band gap. When the impurity atom (donor) donates negatively charged carriers, it creates n-type doping semiconductors, while p-type doping semiconductors have positively charged carriers of acceptor atoms. Acceptors accept electrons from the valence band to create covalent bonds with neighboring atoms.

2.5.2 Solar Cells

Diodes are electronic components that transmit electrical current in one way. As the name implies, a di-ode consists of two electrical conductors (electrodes). Diodes are formed with a junction of p-type and n-type semiconductors. Holes that are concentrated on the p-side tend to diffuse and fill the crystal structure uniformly, while electrons diffuse from the n-side. Negatively charged electrons in the n-type region fill the positively charged holes in the p-type region in the depletion region. This charge transfer leaves positive donor ions on the n-side and negatively ionized acceptors on the p-side of the depletion region, as can be seen in Figure 2.7. The depletion zone acts as a buffer zone for the junction. Free electrons absorb enough energy to overcome the depletion zone to combine with holes.

Photovoltaic (or solar) cells are a type of diode. Solar cells act as a diode in the absence of light, and they convert the radiation sourced from the sun into electrical energy using the photovoltaic effect. Each photon absorbed by the semiconductor creates an electron-hole pair. These charge carriers have an electric field opposite the built-in electric field of the p-n junction, thus creating a voltage barrier. Carriers diffuse into the depletion zone by absorbing photons to overcome barrier potential. The separation of these carriers in

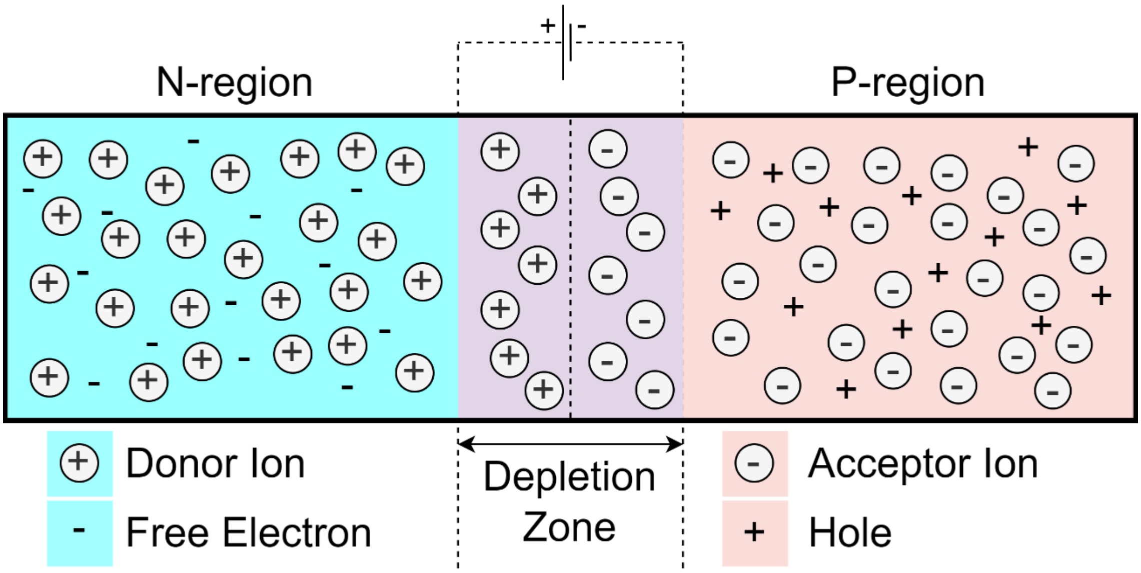

Figure 2.7 Diagram for a semiconductor with a p-n junction.

the depletion zone produces a forward voltage across the junction barrier [37]. Since different elements have different band gaps, so do the material compounds created using these elements. The band gap of solar cells changes with the crystal structure. Modifying the compounds of the material that solar cell is made of also modifies cell efficiency. The theoretical efficiency of a solar cell made from a single p-n junction is expressed as the Shockley-Queisser limit [38], which can be seen in Figure 2.8.

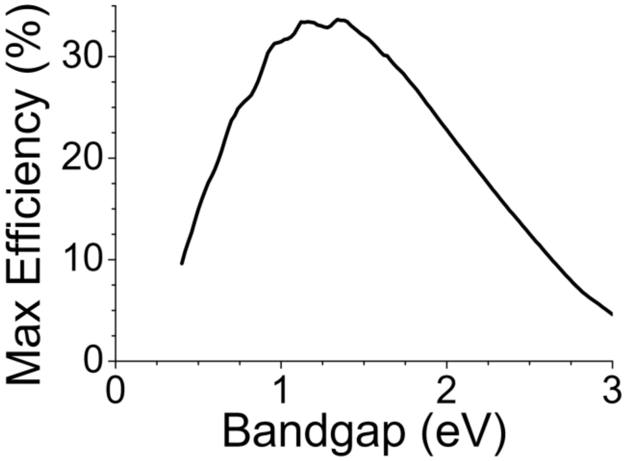

Figure 2.8 Maximum theoretical efficiency of a solar cell made of a single p-n junction known as Shockley-Queisser limit.

3 METHODOLOGY

3.1 Data

The quality of the data set is important for obtaining an ML model that can operate for different data sets. As mentioned in Section 2.1, ML models also require quantified clean data. Obtaining a data set is a prerequisite before advancing into other operations of ML. The number of databases compiled to support material research are increasing in recent years. A few of these publicly available databases are OQMD [39], AFLOW [40], Materials Project [41], and Citrine.

In this thesis, the Citrine data set is used for creating an ML model that can predict material band gaps. Then experimental band gap data set from Matminer [8] is used for additional testing. The band gap data set obtained from Citrine was compiled by Strehlow and Cook in 1973 [42]. This band gap data set covers 723 material samples with four columns that contain different information about materials, such as the material's chemical formula, crystallinity, color, and band gap. Using the compound formulas in the data set, one can generate additional features to improve the training process for the ML models via data augmentation.

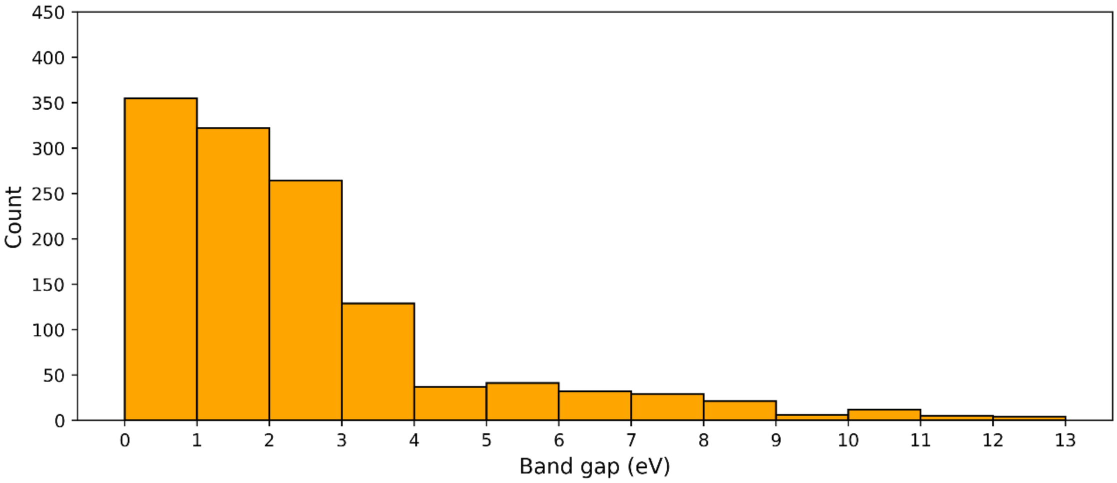

Figure 3.1 Distribution of the material band gaps for cleaned Citrine data set.

3.1.1 Pre-Processing

Before generating additional features, the band gap data set is processed manually via the Python Data Analysis Library (Pandas). Color information had many missing values; hence it's eliminated as the first step. Then, band gap values of 17 compounds had uncertainties, and excluding 17 compounds from the data set is the next step for cleaning data. Among the materials in the Citrine dataset, not all are unique. Averaging the band gaps of reemerged materials in the list is the last step before feature generation. Figure 3.1 visualizes the counts left after cleaning. Figure 3.2 briefly shows the process. The cleaned data set covers 424 materials with a mean band gap of 2.236 eV, a minimum band gap of 0.008 eV, and a maximum band gap of 12.435 eV.

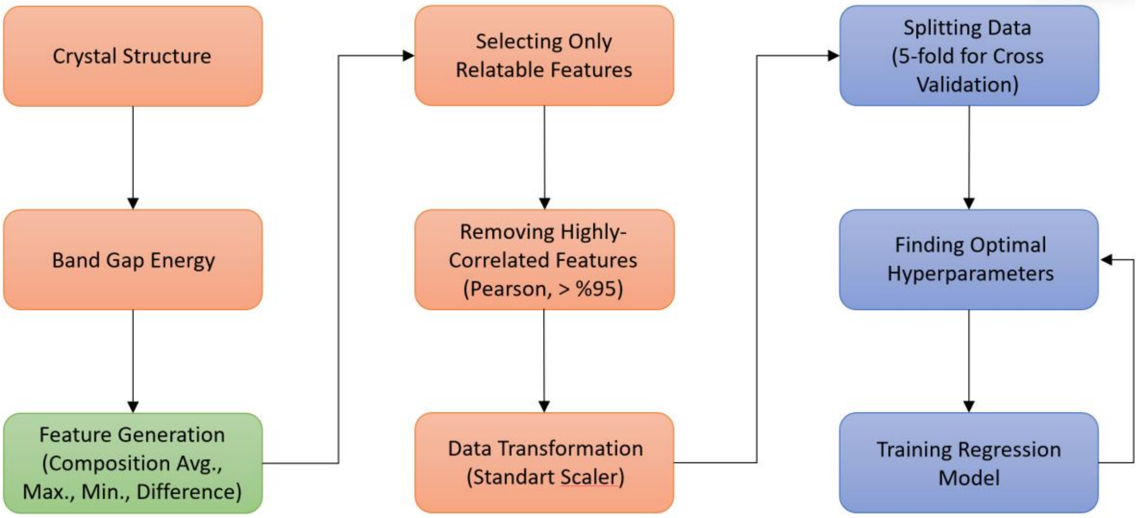
Crystal Structure
Band Gap Energy
Feature Generation (Composition Avg., Max., Min., Difference)
Selecting Only Relatable Features
Removing Highly-Correlated Features (Pearson, > %95)
Data Transformation (Standart Scaler)
Splitting Data (5-fold for Cross Validation)
Finding Optimal Hyperparameters
Training Regression Model

Figure 3.2 Workflow for obtaining the final model using Citrine data set.

3.1.2 Feature Generation

MAST-ML [28] library can generate 87 atomic features in 5 types using Materials Agnostic Platform for Informatics and Exploration (Magpie). Magpie software was developed as a part of the OQMD to predict the properties of materials [43]. These features for chemical compounds are composition average, arithmetic average, maximum value, minimum value, and difference. One downside of the feature generator is being limited to three compounds and cannot generate features for materials that consist of four or more elements in one data set. The list of 87 atomic features that Magpie have and their definitions are listed in Table A.1.

Using the elemental feature generator of MAST-ML, feature data is augmented by generating additional features in four types: composition average, difference, maximum value, and minimum value. The generated data set now has 349 features for 424 materials.

3.1.3 Feature Selection

Feature selection is another major step in ML since features and their data have a significant role in overall model performance. There are different methods to select or eliminate features, such as correlation-based feature selection. Correlation affects the model variance and scales between -1 and 1. Having highly correlated features in training data creates a model that overfits. Selecting only one of the two highly correlated features would improve a model's performance by decreasing the number of outliers. There are also several correlation types. Pearson is used commonly with linear models that need a decrease in outliers. Pearson's correlation is a measure of the linearity between two features. Using the Pearson method, a set of features that are associated with more than %95 are excluded from the manually selected features data set.

Table 3.1 23 selected features used for creating the final models.

	Selected Features	
Valence _difference	ElectronAffinity _difference	ElectronAffinity _max_value
Polarizability _min_value	SecondIonizationEnergy _max_value	AtomicRadii _min_value
HeatFusion _max_value	IsRareEarth _composition_average	GSbandgap _max_value
ThirdIonizationEnergy _composition_average	n_ws^third _composition_average	NfValence _composition_average
GSenergy_pa _max_value	IsMetalloid _composition_average	NValance _composition_average
BCCfermi _min_value	NpUnfilled _composition_average	BCCfermi _composition_average
IonicRadii _max_value	NpValence _composition_average	NValance _min_value
SpaceGroupNumber _difference	NdUnfilled _max_value	

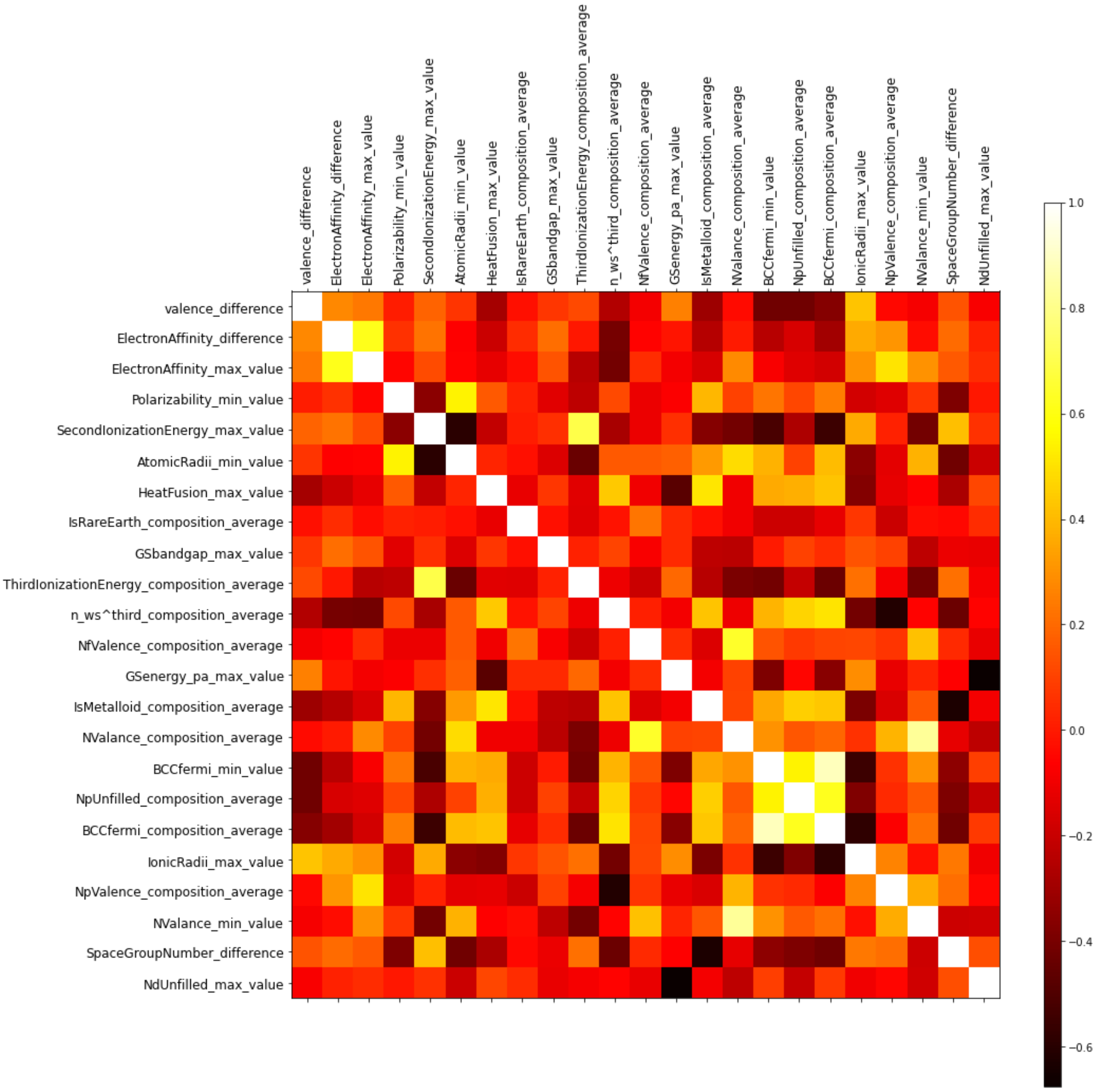

Figure 3.3 Correlation matrix with a color bar visualizes the correlation between 23 selected features shown in Table 3.1.

At first, among the atomic features generated via MAST-ML, only features that increase the R^2_{adj} score of the trained model are tried and selected manually one by one to investigate how the model performs under distinct features. Then, a custom algorithm is developed to automate this exhaustive process. The developed method evaluates the contribution of each feature to the model performance and creates an optimal list of the features using RMSE, MAE, R^2, and R^2_{adj} metrics as feedback. This method also drops highly correlated features and scales the new data set using StandartScaler in each epoch until the feature list takes its final form when metric scores stop improving. Table 3.1 shows the list of the selected features. Three manually selected features that initialize the feature selector algorithm are in the first row. Figure 3.3 is a visual representation of the Pearson correlation matrix calculated for selected features.

3.2 Model

3.2.1 Model Selection

In ML, there is no best model. Selecting a model to work on is a problem of statistical modeling. An inquiry must be executed by evaluating a list of algorithms via methods such as cross-validation to determine which model to use. Metric scores obtained after training and validating the model can be used to conclude the inquiry. The bias and variance are a few measures that express the quality of an estimator, along with MAE and RMSE. In this thesis, model selection is executed based on the metric performance. Tools in WEKA were also used to reduce the time required for this inquiry about eliminating models that may perform poorly on the data set. RF model is selected mainly due to the predictive capabilities of the algorithm for the task subjected to this thesis.

3.2.2 Hyperparameter Tuning

Models trained using their default hyperparameters may yield unsatisfactory performance. Monitoring model performance using the validation set under different validation methods is essential to prevent the issues relating to performance. Poor performance can be determined by assessing metrics obtained right after model validation. As addressed above in Section 2.1 as the bias-variance trade-off, models tend to overfit when validation errors stop decreasing while training continues. There are different methods to improve the performance and metrics of a trained ML model.

Hyperparameter optimization uses only training and validation sets. GridSearchCv algorithm in the Scikit-learn library can cross-validate models while searching for their best parameters Different parameters of models influence performance in different amounts. Searching for its optimal value for every parameter is a computationally expensive process. Only the parameters that affect a model more than others were the aspect of this search. Grid search can be optimized by specifying a range of parameter values. Hyperparameter tuning is executed using 5-fold cross-validation for all models. The search dictionary for model parameters is defined in Table 3.2.

Table 3.2 Hyperparameters of regression models with their default values and range arrays in the parameter search dictionary.

Model	Hyperparameter	Range	Default
RF	n_estimators	50, 100, 150, 250, 500	100
	bootstrap	True, False	True
	criterion	‘mse’,’mae’,’poisson’	‘mse’
MLP	hidden_layer_sizes	(1,):(21,), (70,), (170,)	(100,)
	activation	‘tanh’,’relu’	‘relu’
	max_iter	500, 1000, 1500	500
KRR	kernel	‘linear’,’rbf’	‘linear’
	alpha	[1:1.9,], [1:5]	1

4
RESULTS AND DISCUSSION

4.1 Results – Citrine Data Set

Band gaps of various materials are estimated using ML methods. The performance of some optimized ML models, in terms of metric scores, is declared in Table 4.1. The set of features indicated in Table 3.1 is used for training the models. Both default and the optimized RF model trained using the same set of features performs with an MAE less than 0.5 eV, which proves the success of feature engineering. Figure 4.1 and Figure 4.3 shows the fit line for the RF model consisting of 500 decision trees. The confidence interval in the figures consists of the mean and variance for estimations. An estimated value should be in between the upper and lower bounds of the specified confidence interval, which is 95% in this case. Using RBF as a kernel for KRR, the KRR-version of Figure 4.1 is visualized in Figure 4.2 for comparison sake. The increase in error and divergence from the fit line can be observed when comparing these two figures.

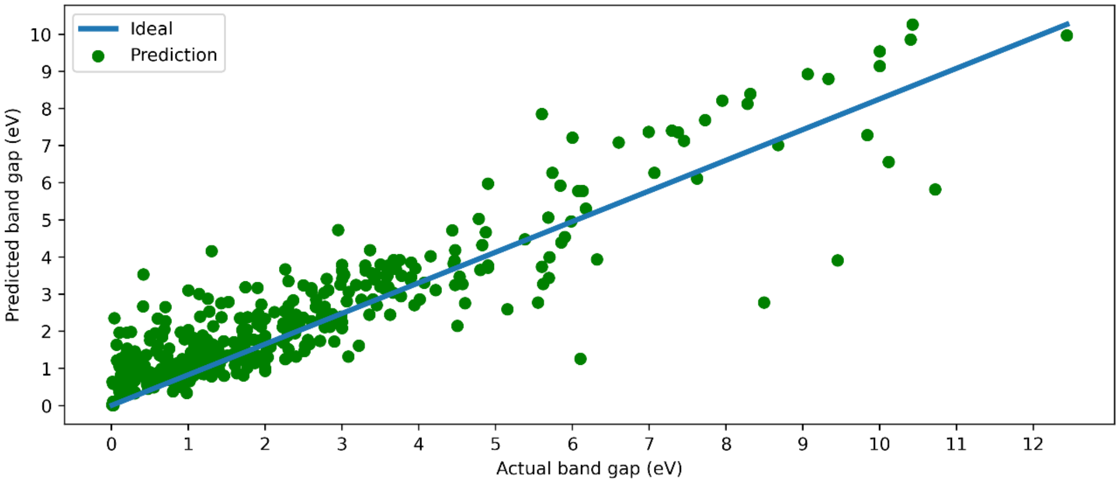

Figure 4.1 Scatter plot with a fit line obtained by RF model.

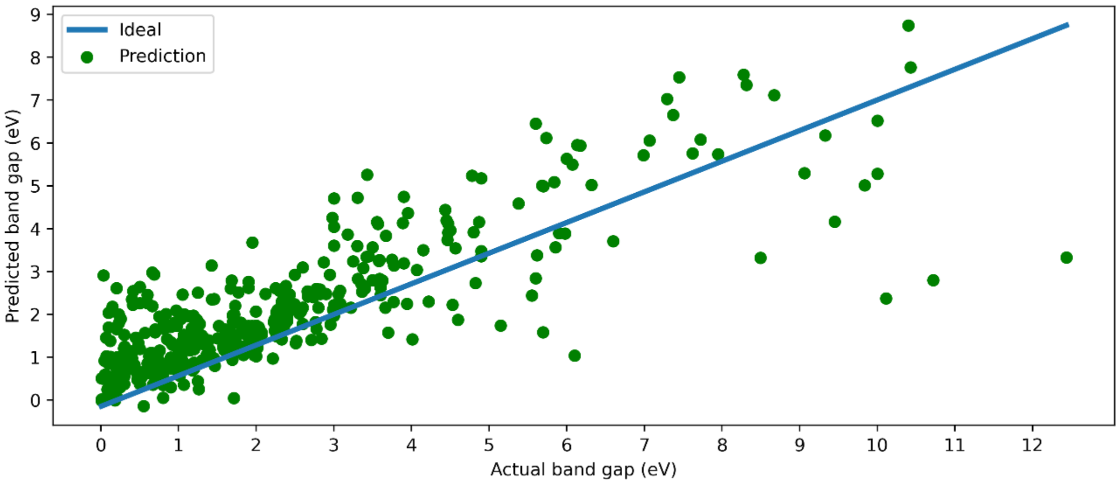

Figure 4.2 Scatter plot with a fit line for KRR model

Table 4.1 Optimal hyperparameters of ML algorithms and their metric scores for 5-fold cross-validation.

Model	**Hyperparameter**		**Metric**			
	Name	Value	MAE	MSE	RMSE	R^2
RF	n_estimators	500	0.476	0.370	0.608	0.868
	bootstrap	True				
	criterion	'mse'				
MLP	hidden_layer_sizes	(170,)	0.595	0.621	0.788	0.778
	activation	'relu'				
	max_iter	500				
KRR	kernel	'rbf'	0.551	0.567	0.753	0.797
	alpha	1.0				

RF model also predicts the band gap of 102 compounds among 424 materials with less than a 10% error rate, which corresponds to an accuracy of ∼24% for the RF model. The number of band gaps predicted with a low error rate (10%) went up to 108 (~26% accuracy) when investigating the effects of using different sets of features, but on the other hand, metric scores worsened. Table 4.2 shows the actual band gaps and their estimated values for 13 materials with less than a 1% error rate.

Table 4.2 Materials in the Citrine data set, actual band gaps, and predicted band gaps by RF model with less than 1% error rate.

Formula	Actual	Predicted	Error	Error (%)
RuSe2	1.000	0.994	0.006	0.595
Ge0.891Si0.109	1.126	1.134	-0.008	0.736
GeSe	1.200	1.206	-0.006	0.488
Eu3P2	1.200	1.201	-0.001	0.072
Sb2Se3	1.213	1.221	-0.008	0.696
Dy2O3	2.235	2.221	0.014	0.626
As2S5	2.330	2.335	-0.005	0.208
BiBr3	2.660	2.654	0.006	0.237
TlBr	3.085	3.057	0.027	0.888
GaN	3.420	3.404	0.016	0.459
CBr4	3.700	3.680	0.020	0.532
NaBr	7.371	7.367	0.004	0.054
LiBr	7.725	7.694	0.031	0.400

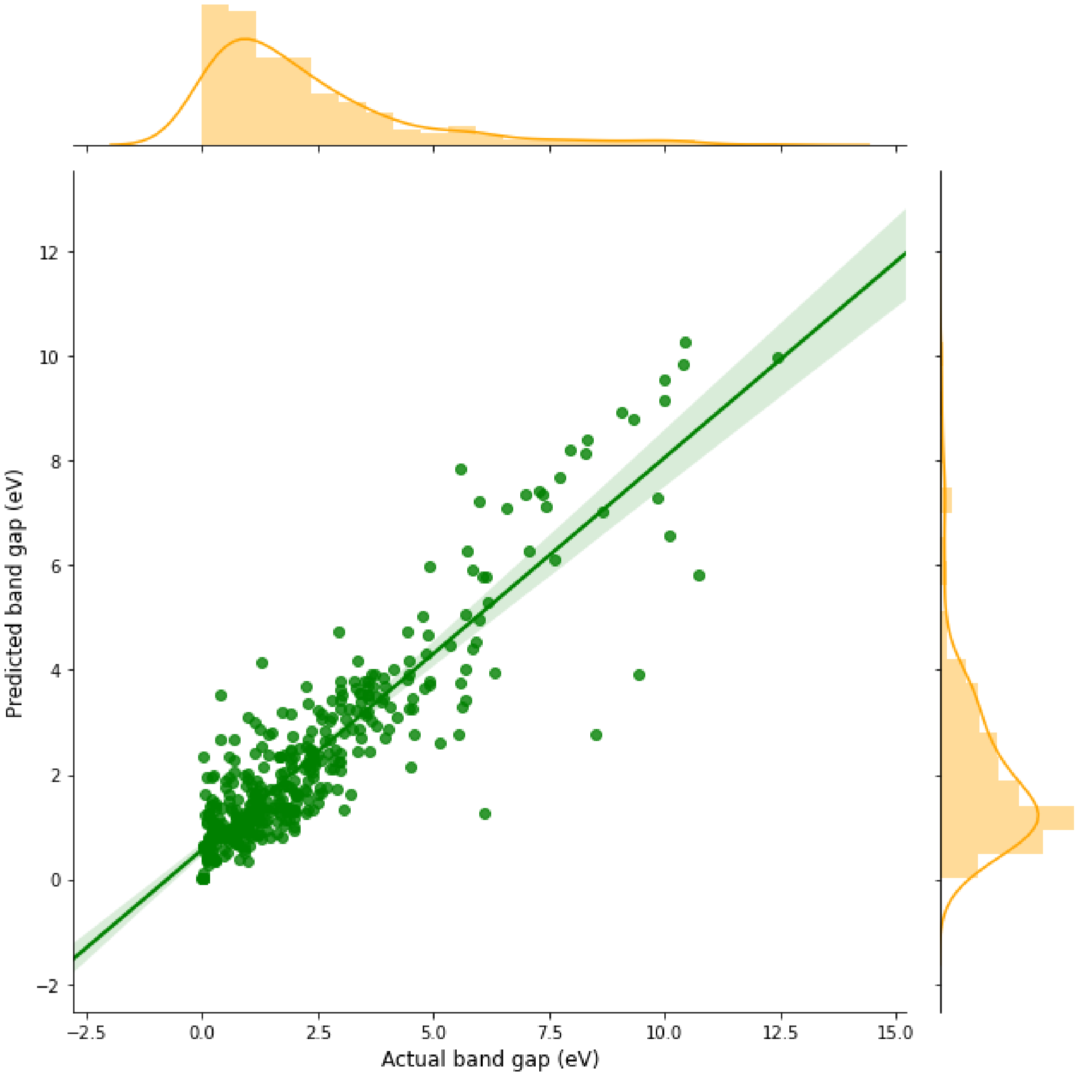

Figure 4.3 Same fit line in Figure 4.1 with 95% confidence interval. Histograms show the distribution of both actual and predicted band gaps.

4.2 Results – Matminer Data Set

The difference between the RF model and other models emerges after testing these models using different features or data. Model performances are also evaluated using pre-processed features generated using the Matminer data set. Due to limitations of MAST-ML, only the materials that consist of 3 elements are included for feature generation, and the rest is discarded. Excluding materials that have a band gap less than 0.008 eV yields several advantages, such as a drastic decrease in prediction errors and improved model performance. After filtering operations, the data set used for model training covers 1480 materials with a maximum band gap of 11.700 eV and has a mean band gap of 1.830 eV.

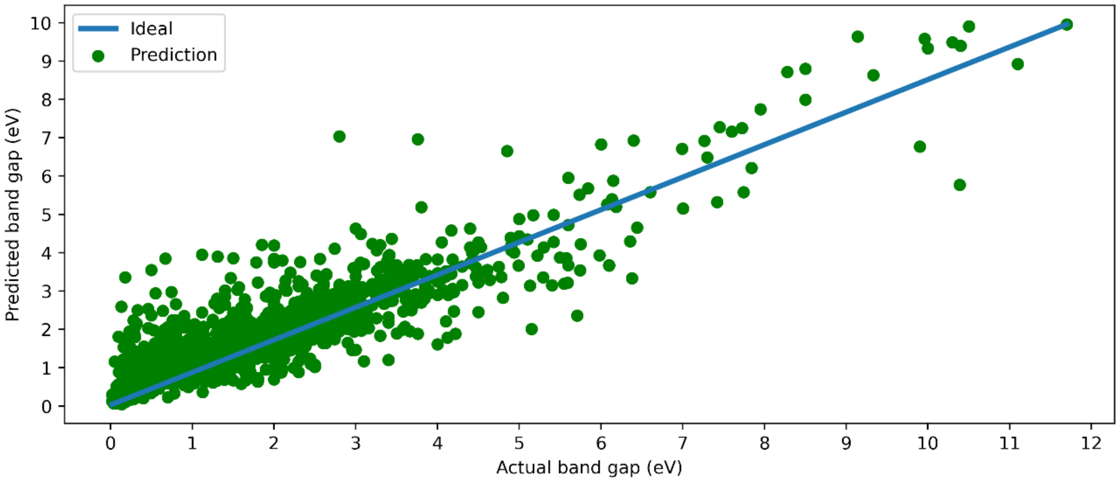

Figure 4.4 Scatter plot with a fit line obtained for RF model.

Table 4.3 Model performance for Matminer data set and metric scores for 5-fold cross-validation.

Model	**Metric**			
	MAE	MSE	RMSE	R^2
RF	0.453	0.599	0.774	0.800
MLP	0.490	0.672	0.820	0.775
KRR	0.586	1.114	1.056	0.627

Results obtained under 5-fold cross-validation are shown in Table 4.3. Figure 4.4 and Figure 4.5 visualizes model fit and predicted band gaps of the Matminer data set. The small difference between RMSE of default (0.771 eV) and optimized RF (0.774 eV) models having the same MAE proves that the optimized RF is applicable to different data sets. Errors for KRR increased parallel to the number of predictions. After comparing both the tables and the results obtained for two different data sets using the same features, one can suggest that RF models, whether optimized or not, are more robust to changes in the number of predictions than linear models. When the MSE of RF and MLP models slightly increased, the MSE of the KRR almost doubled. RF model predicts the band gap of 467 materials (∼31%) with less than 10% error.

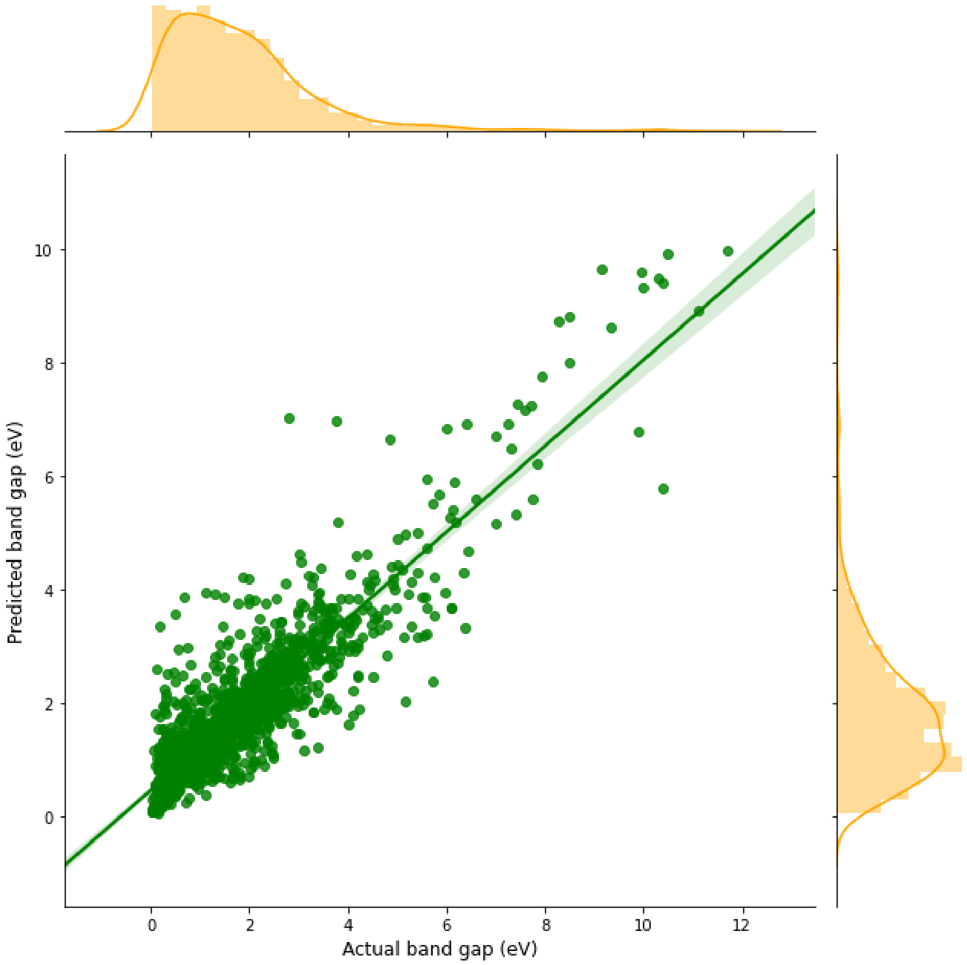

Figure 4.5 Same fit line in Figure 4.4 with 95% confidence interval and distribution histograms for both actual and predicted band gaps.

4.3 Discussion

ML is a game-changer for the forthcoming years. While software such as WEKA produces results with medium accuracy, a well-optimized model can estimate band gaps with little margin of error (results for various ML models obtained using WEKA without selecting any specific features are given in Table B.1, Table B.2, and Table B.3). Model optimization should not be limited to only hyperparameter tuning. Highly correlated features in the data set significantly affect the model performance. There are successful feature selectors, although they didn't yield optimal results for the task subject to this thesis. Selecting and using the 30 best features obtained via algorithms such as SelectKBest or ExtraTreeRegressor yields MAE of 0.559 and 0.582, or RMSE of 0.783 and 0.844 for the RF model, respectively. Evaluating features and the model without using R^2 along with R^2_{adj}

is not viable. One can have high R^2 scores while MAE and RMSE scores are not decreasing. Utilizing R^2_{adj} while selecting features improves the performance in both the feature selection and the model training. All the supplementary materials, such as codes and data files, are publicly accessible on GitHub [44].

4.4 Future Work

This work is open for future improvement. This may come from additional feature engineering, using different functions for pre-processing, such as MinMaxScaler. The whole process can be developed as a software program with certain automation capabilities. The developed feature selector has room for improvement with a seed-based shuffling of features, better traction for metric scores, etc. Obtaining a simple mathematical formula generalizable to many materials and having competence in estimating the band gaps of materials depending on several atomic features is also possible.

REFERENCES

[1] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.

[2] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *American Psychological Association,* vol. 65, no. 6, pp. 386-408, 1958.

[3] B. H. Menze et al., "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)," *IEEE Transactions on Medical Imaging,* vol. 34, no. 10, pp. 1993-2024, 2015.

[4] D. A. Pomerleau, "ALVINN: An Autonomous Land Vehicle in a Neural Network," *NIPS,* 1988.

[5] M. Bojarski, D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao and K. Zieba, "End to End Learning for Self-Driving Cars," *ArXiv,* vol. abs/1604.07316, 206.

[6] L. Susam, H. Yılmaz Alan, A. Yilmaz, A. Erol, C. Inci, F. C. Akinci, B. Akkus, M. Demir, M. Emirhan, O. Faydasicok and E. Güdekli, "Estımation Of Cosmic Radiation Dose Received On Flight By Means Of Different Machine Learning Methods And CARI-7A Calculations," *Romanian Journal of Physics,* vol. 67, 2022.

[7] B. AFFLERBACH, R. Jiang, J. Tappan and D. MORGAN, "Machine Learning Lab Module," 12 05 2021. [Online]. Available: https://nanohub.org/resources/intromllab. [Accessed 02 08 2021].

[8] Y. Zhuo, A. M. Tehrani and J. Brgoch, "Predicting the Band Gaps of Inorganic Solids by Machine Learning," *The Journal of Physical Chemistry Letters,* vol. 9, no. 7, pp. 1668-1673, 2018.

[9] L. Wu, Y. Xiao, M. Ghosh, Q. Zhou and Q. Hao, "Machine Learning Prediction for Bandgaps of Inorganic Materials," *ES Materials & Manufacturing,* vol. 9, pp. 34-39, 2020.

[10] P. Hohenberg and W. Kohn, "Inhomogeneous Electron Gas," *Phys. Rev.,* vol. 136, no. 3B, pp. B864-B871, 1964.

[11] G. Ravanhani Schleder, A. C. Padilha, C. Acosta, M. Costa and A. Fazzio, "From DFT to Machine Learning: recent approaches to Materials Science – a review," *Journal of Physics Materials,* vol. 2, no. 3, p. 032001, 2019.

[12] D. Bagayoko, "Understanding density functional theory (DFT) and completing it in practice," *AIP Advances,* vol. 4, no. 12, p. 127104, 2014.

[13] W. Kohn, "Nobel Lecture: Electronic structure of matter--wave functions and density functionals," *Rev. Mod. Phys.,* vol. 71, no. 5, pp. 1253-1266, 1999.

[14] J. P. Perdew, K. Burke and M. Ernzerhof, "Generalized Gradient Approximation Made Simple," *Phys. Rev. Lett.,* vol. 77, no. 18, pp. 3865-3868, 1996.

[15] C. Sutton, L. Ghiringhelli, T. Yamamoto, Y. Lysogorskiy, L. Blumenthal, T. Hammerschmidt, J. Golebiowski, X. Liu, A. Ziletti and M. Scheffler, "Crowd-sourcing materials-science challenges with the NOMAD 2018 Kaggle competition," *npj Computational Materials,* vol. 5, no. 1, 2019.

[16] J. R. Moreno, J. Flick and A. Georges, "Machine learning band gaps from the electron density," *Physical Review Materials,* vol. 5, no. 8, p. 083802, 2021.

[17] R. W. Godby, M. Schlüter and L. J. Sham, "Accurate Exchange-Correlation Potential for Silicon and Its Discontinuity on Addition of an Electron," *Phys. Rev. Lett.,* vol. 56, no. 22, pp. 2415-2418, 1986.

[18] M. K. Y. Chan and G. Ceder, "Efficient Band Gap Prediction for Solids," *Phys. Rev. Lett.,* vol. 105, no. 19, p. 196403, 2010.

[19] G. Pilania, J. E. Gubernatis and T. Lookman, "Multi-fidelity machine learning models for accurate bandgap predictions of solids," *Computational Materials Science,* vol. 129, pp. 156-163, 2016.

[20] T. Wang, K. Zhang, J. Thé and H. Yu, "Accurate prediction of band gap of materials using stacking machine learning model," *Computational Materials Science,* vol. 201, p. 110899, 2022.

[21] O. Allam, C. Holmes, Z. Greenberg, K. C. Kim and S. S. Jang, "Density Functional Theory – Machine Learning Approach to Analyze the Bandgap of Elemental Halide Perovskites and Ruddlesden-Popper Phases," *ChemPhysChem,* vol. 19, no. 19, pp. 2559-2565, 2018.

[22] S. Siqi, Z. Yan, W. Qu, G. Jian, L. Yue, J. Wangwei, O. Chuying and X. Ruijuan, "Multi-scale computation methods: Their applications in lithium-ion battery research and development," *Chinese Physics B,* vol. 25, no. 1, p. 018212, 2016.

[23] Y. Huang, C. Yu, W. Chen, Y. Liu, C. Li, C. Niu, F. Wang and Y. Jia, "Band gap and band alignment prediction of nitride-based semiconductors using machine learning," *Journal of Materials Chemistry C,* vol. 7, no. 11, pp. 3238-3245, 2019.

[24] H.-J. Feng, K. Wu and Z.-Y. Deng, "Predicting Inorganic Photovoltaic Materials with Efficiencies >26% via Structure-Relevant Machine Learning and Density Functional Calculations," *Cell Reports Physical Science,* vol. 1, p. 100179, 2020.

[25] P. Raccuglia, K. Elbert, P. D. F. Adler, C. Falk, M. B. Wenny, A. Mollo, M. Zeller, S. A. Friedler, J. Schrier and A. J. Norquist, "Machine-learning-assisted materials discovery using failed experiments," *Nature,* vol. 533, no. 7901, pp. 73-76, 2016.

[26] R. Sutton and A. G. Barto, Reinforcement Learning: An Introduction (2nd Edition), MIT Press, 2018.

[27] V. Saggio, B. Asenbeck, A. Hamann, T. Strömberg, P. Schiansky, V. Dunjko, N. Friis, N. Harris, M. Hochberg, D. Englund, S. Wölk, H. Briegel and P. Walther, " Experimental quantum speed-up in reinforcement learning agents," *Nature,* vol. 591, pp. 229-233, 2021.

[28] R. Jacobs, T. Mayeshiba, B. Afflerbach, L. Miles, M. Williams, M. Turner, R. Finkel and D. Morgan, "The Materials Simulation Toolkit for Machine Learning (MAST-ML): An automated open source toolkit to accelerate data-driven materials research," *Computational Materials Science,* vol. 176, p. 109544, 2020.

[29] J. Schmidt, M. R. G. Marques, S. Botti and M. A. L. Marques, "Recent advances and applications of machine learning in solid-state materials science," *npj Computational Materials,* vol. 5, no. 1, 2019.

[30] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning,* vol. 1, no. 1, pp. 81-106, 1986.

[31] L. Breiman, "Bagging predictors," *Machine Learning,* vol. 24, no. 2, pp. 123-140, 1996.

[32] R. Carnap, "Testability and Meaning," *Philosophy of Science,* vol. 3, no. 4, pp. 419-471, 1936.

[33] M. A. Riedmiller, "Advanced supervised learning in multi-layer perceptrons — From backpropagation to adaptive learning algorithms," *Computer Standarts & Interfaces,* vol. 16, pp. 265-278, 1994.

[34] B. S. Everitt and A. Skrondal, Cambridge Dictionary of Statistics, Cambridge, UK: Cambridge University Press, 2010.

[35] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research,* vol. 30, no. 1, pp. 79-82, 2005.

[36] M. A. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations,* vol. 11, no. 1, pp. 10-18, 200.

[37] C. Kittel, Introduction to Solid State Physics, 8th Edition, John Wiley & Sons Inc, 2004.

[38] W. Shockley and H. J. Queisser, "Detailed Balance Limit of Efficiency of p-n Junction Solar Cells," *Journal of Applied Physics,* vol. 32, no. 3, pp. 510-519, 1961.

[39] S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl and C. Wolverton, "The Open Quantum Materials Database (OQMD) : Assessing the accuracy of DFT formation energies.," *npj Computational Materials,* vol. 1, 2015.

[40] S. Curtarolo, W. Setyawan, G. L. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M. J. Mehl, H. T. Stokes, D. O. Demchenko and D. Morgan, "AFLOW: An automatic framework for high-throughput materials discovery," *Computational Materials Science,* vol. 58, pp. 218-226, 2012.

[41] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder and K. A. Persson, "Commentary: The Materials Project: A materials genome approach to accelerating materials innovation," *APL Materials,* vol. 1, no. 1, p. 011002, 2013.

[42] W. H. Strehlow and E. L. Cook, "Compilation of Energy Band Gaps in Elemental and Binary Compound Semiconductors and Insulators," *Journal of Physical and Chemical Reference Data,* vol. 1, no. 2, pp. 163-200, 1973.

[43] L. Ward, A. Agrawal, A. Choudhary and C. Wolverton, "A general-purpose machine learning framework for predicting properties of inorganic materials," *npj Computational Materials,* vol. 2, no. 1, p. 16028, 2019.

[44] aiostarex, "Band Gap Prediction Using ML," 2022. [Online]. Available: https://github.com/aiostarex/Band-Gap-Prediction-Using-ML/.

[45] wolverton, "Magpie," Wolverton Research Group, 2020. [Online]. Available: https://bitbucket.org/wolverton/magpie/src/master/lookup-data/README.txt. [Accessed 27 02 2022].

A
APPENDIX – List of Features

Table A.1 Definitions of the features generated via MAST-ML and Magpie obtained from Magpie source codes [45].

Name of Feature	Definition
AtomicNumber	Atomic number
AtomicRadii	Atomic radii
AtomicVolume	Atomic volume
AtomicWeight	Atomic weight
BCCefflatcnt	Efficient lattice constant for BCC structure
BCCenergy_pa	BCC energy per atom
BCCfermi	BCC fermi energy
BCCmagmom	BCC magnetic moment per atom
BCCvolume_pa	Volume per atom for BCC structure
BCCvolume_padiff	Difference of volume per atom for BCC structure
BoilingT	Boiling temperature
BulkModulus	Bulk modulus
Column	Column of the atom on the periodic table
CovalentRadii	Covalent radii of atoms
CovalentRadius	Covalent radius of each element
Density	The density of the element at STP
ElasticModulus	Elastic modulus
ElectricalConductivity	Electrical conductivity
ElectronAffinity	Electron affinity
Electronegativity	Pauling electronegativity
FirstIonizationEnergy	Energy required to remove the first electron from an element
GSbandgap	DFT band gap energy of T=0K ground state
GSenergy_pa	DFT energy per atom (raw Vienna Ab Initio Simulation Package (VASP) value) of T=0K ground state
GSestBCClatcnt	Estimated BCC lattice parameter based on the DFT volume of the OQMD ground state for each element
GSestFCClatcnt	Estimated FCC lattice parameter based on the DFT volume of the OQMD ground state for each element
GSmagmom	DFT magnetic moment of T=0K ground state
GSvolume_pa	DFT volume per atom of T=0K ground state
Group	Group of atoms according to the periodic table
HHIp	Herfindahl-Hirschman Index (HHI) production values
HHIr	Herfindahl-Hirschman Index (HHI) reserves values
HeatCapacityMass	Heat capacity per mass at STP

Table A.1 Definitions of the features generated via MAST-ML and Magpie obtained from Magpie source codes [45] (continuing).

HeatCapacityMolar	Molar heat capacity at STP
HeatFusion	Enthalpy of fusion for elements at their melting temperatures
HeatVaporization	Vaporization temperature for an element
ICSDVolume	Volume per atom of ICSD phases at STP
IonicRadii	Radii of ion
IonizationEnergy	Energy required to remove the loosely bound electron from an element
IsAlkali	Whether an element is an alkali metal
IsAlkalineEarth	Whether an element is an alkali earth metal
IsBCC	Whether an element has a body-centered cubic structure
IsBoron	Whether an element is a boron
IsCarbon	Whether an element is a carbon
IsChalcogen	Whether an element is a chalcogen
IsDBlock	Whether an element is a d-block metal
IsFBlock	Whether an element is an f-block metal
IsFCC	Whether an element has a face-centered cubic structure
IsHalogen	Whether an element is a halogen
IsHexagonal	Whether an element has a hexagonal structure
IsMetal	Whether an element is a metal
IsMetalloid	Whether an element is a metalloid
IsMonoclinic	Whether an element has a monoclinic structure
IsNonmetal	Whether an element is a nonmetal
IsOrthorhombic	Whether an element has an orthorhombic structure
IsPnictide	Whether an element is a pnictide
IsRareEarth	Whether an element is a rare earth metal
IsRhombohedral	Whether an element is a rhombohedral metal
IsSimpleCubic	Whether an element has a simple cubic structure
IsTetragonal	Whether an element has a tetragonal structure
IsTransitionMetal	Whether an element is a transition metal
MeltingT	Melting temperature of element
MendeleevNumber	Mendeleev Number (position on the periodic table, counting column-wise starting from Hydrogen)
MiracleRadius	Assessed radii of elements in metallic glass structures
NUnfilled	Number of unfilled valence orbitals
NValance	Group of an atom according to periodic table-according to valence electron number
NdUnfilled	Number of unfilled d valence orbitals
NdValence	Number of filled d valence orbitals
NfUnfilled	Number of unfilled f valence orbitals
NfValence	Number of filled f valence orbitals
NpUnfilled	Number of unfilled p valence orbitals

Table A.1 Definitions of the features generated via MAST-ML and Magpie obtained from Magpie source codes [45] (continuing).

NpValence	Number of filled s valence orbitals
NsUnfilled	Number of unfilled s valence orbitals
NsValence	Number of filled s valence orbitals
Number	The atomic number of element
Period	Period of an atom according to periodic table– row on the periodic table
Polarizability	Static average electric dipole polarizability
Row	Row on the periodic table
SecondIonizationEnergy	Energy to remove the second electron from an element
ShearModulus	Shear modulus
SpaceGroupNumber	The space group of T=0K ground state structure
SpecificHeatCapacity	Specific heat capacity at STP
ThermalConductivity	Thermal conductivity
ThermalExpansionCoefficient	Thermal expansion coefficient
ThirdIonizationEnergy	Energy to remove the third electron from an element
n_ws^third	Electron density at the surface of Wigner-Sietz cell (used in Miedema's model)
phi	Adjusted work function (used in the Miedema's model)
valence	Number of valence electrons

B
APPENDIX – WEKA Results

Table B.1 WEKA results obtained for various ML models for 5-fold cross validation.

Model	Metric		
	MAE	RMSE	R^2
RandomForest	0.643	0.9493	0.9098
RandomTree	0.8972	1.3665	0.8051
REPTree	0.9056	1.3351	0.8088
M5P	1.2896	11.8757	0.091
DecisionStump	1.3383	1.7809	0.5945
M5Rules	0.7952	1.2601	0.8285

Table B.2 WEKA results obtained for various ML models for 10-fold cross validation.

Model	Metric		
	MAE	RMSE	R^2
RandomForest	0.6414	0.9449	0.91
RandomTree	0.974	1.4795	0.7739
REPTree	0.8555	1.2183	0.8364
M5P	1.2896	11.8757	0.091
DecisionStump	1.3743	1.8541	0.5475
M5Rules	0.8025	1.2023	0.8425

Table B.3 WEKA results obtained for various ML models for 90:10 train-validation split.

Model	Metric		
	MAE	RMSE	R^2
RandomForest	0.7355	1.0626	0.8825
RandomTree	1.4246	2.3675	0.4759
REPTree	0.8118	1.1499	0.8621
M5P	6.4764	37.5922	-0.1149
DecisionStump	1.5339	1.9836	0.4854
M5Rules	0.7267	1.0618	0.8848

PUBLICATIONS FROM THE THESIS

Conference Proceedings

1. A. Erol, S. D. Gunay (2021), “Predicting Electron Band Gaps with Machine Learning Using Decision Tree and Random Forest Models”, BOOK OF FULL TEXT PROCEEDINGS TURKISH PHYSICAL SOCIETY 37th INTERNATIONAL PHYSICS CONGRESS (TPS37), TPS37, Vol. 03, No. 03, pp 29-33.