Bachelor's Programme in Science and Technology

Machine Learning in Materials Development

Han Le

Aalto University

BACHELOR'S THESIS

Aalto University
BACHELOR'S THESIS 2022

Machine Learning in Materials Development

Han Le

Thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Technology.
Otaniemi, 15 Apr 2022

Supervisor: professor Michael Hummel
Advisor: professor Kari Laasonen

Aalto University
School of Chemical Engineering
Bachelor's Programme in Science and Technology

Author
Han Le

Title
Machine Learning in Materials Development

School School of Chemical Engineering

Degree programme Bachelor's Programme in Science and Technology

Major Chemical Engineering **Code** CHEM10.A

Supervisor professor Michael Hummel

Advisor professor Kari Laasonen

Level Bachelor's thesis **Date** 15.4.2022 **Pages** 42+5 **Language** English

Abstract

Traditionally, materials science is driven by theory, experimentation, or conventional computation, such as density functional theory. However, these methods for materials research and development are evaluated to have disadvantages in terms of cost and performance.

To tackle this problem, experts in materials research and development have experimented with machine learning, a subfield of artificial intelligence. In recent years, machine learning has become popular in materials science and has shown promising results in the applications of materials research and development. For example, machine learning has been utilized for material property prediction, materials discovery, and density functionals computation. On the other hand, there are some knowledge gaps and challenges imposed by machine learning in materials science that need to be identified.

The goal of this thesis is to identify how machine learning can be utilized to accelerate materials development and how developing the machine learning paradigm in materials development is. For this purpose, the thesis focuses on the principles of machine learning, the overview of data for materials science, and the main applications of machine learning for materials development in recent years.

The main findings of this thesis indicate that recent studies have shown some promising results of machine learning applications in materials science, for example, materials property prediction, new materials discovery, and interatomic potentials development. In addition, the development of machine learning algorithms and the database infrastructures for materials science have made the machine learning paradigm more popular in materials development. However, there are still many limitations to machine learning in materials development. Therefore, it is still too early to say that the machine learning paradigm in materials development has become mature. More research efforts are needed to explore further research directions and to tackle the current challenges imposed by machine learning in materials development.

Keywords materials development, materials science, materials research, machine learning

urn https://aaltodoc.aalto.fi

Contents

1. Introduction

As the amount of data for materials science is growing and machine learning (ML) develops, ML has become popular in materials science, especially computational materials science. In recent years, more researchers in materials science are experimenting with ML for their own research [1]. This has led to promising ML applications in materials research, for example, prediction of material properties [2, 3], discovery of new materials focused on component prediction [4] and crystal structure prediction [5, 6], and development of interatomic potentials [7].

According to several publications in the materials science community [8, 9], the paradigm of ML in materials science is expected to accelerate the research and development of materials, because conventional methods in materials science are limited by many factors. For example, the traditional methods are demanding in terms of equipment and resources; therefore, they are time-consuming and are only for a limited number of materials. However, there are still limitations to and knowledge gaps in ML in materials development that need to be addressed and studied further. Several review articles [8, 9, 10] have stated that the main challenges of data-driven materials science are data quality, data quantity, and interpretability of ML models.

The goal of this thesis is to identify how machine learning can be utilized to accelerate materials development and how developing the machine learning paradigm in materials science is. To achieve this objective, the thesis focuses on the basics of machine learning, the data resources for materials science, and the notable machine learning applications in materials development in recent years. This thesis also presents a short demonstration of the redox potential prediction using two machine learning methods to give a clearer example of machine learning computation. Note that this demonstration is not the center of the thesis. It is meant to present

a hands-on experience with a few common Python libraries for machine learning.

The remainder of this thesis is divided into seven sections. Section 2 provides the basics of materials development and a computational method that has a long history in materials science called density functional theory. Section 2 also identifies the main challenges of the traditional methods in materials development. Section 3 provides the principles of machine learning, with a focus on common ML methods in materials science. The importance of chemical descriptors as machine learning features and interpretability of ML models are also introduced in section 3. In section 4, common data sources and an overview of data quality for materials research are presented. This section also describes the role of quantum chemical calculations in data generation for machine learning. Section 5 highlights the recent applications of machine learning in materials development: materials property prediction, new materials discovery, and interatomic potentials development. Section 6 gives a clearer example of a machine learning problem through a short demonstration of utilizing two machine learning methods to predict the molecular redox potential. Finally, Section 7 concludes the main findings of this thesis, addresses the main challenges of machine learning in materials development and the potential solutions, identifies the limitations of this thesis work, and presents the future research directions.

2. Materials development and current challenges

This section includes the formal definition of materials science and explains how it is involved in materials development. Then, the principles of quantum mechanics and a common computational method in materials science called density functional theory (i.e., DFT) are briefly presented. Studies that involved DFT and machine learning are also mentioned in other parts of this thesis.

In addition, this section gives a summary of the main challenges imposed by conventional methods in materials science, which highlight the need for computational methods that are less expensive and more efficient.

2.1 Materials science and materials development

The literature in materials science usually uses the umbrella term *materials science and engineering*, which can be divided into *materials science* and *materials engineering*. Materials science is a field that studies the relationships between structures, properties, synthesis, and processing of materials [11, 12]. Meanwhile, materials engineering chooses suitable materials and modifies them to turn them into useful structures or devices [11]. Generally, the interdisciplinary field of *materials science and engineering* is involved in the development, synthesis, and processing of materials [12]. That is how the term *materials development* is involved in materials science.

There are four notable terms in the basics of materials science and engineering: structure, property, synthesis, and processing. Structure on the subatomic level deals with the electrons of an atom, and structure on the atomic level involves the arrangement of atoms or molecules [11]. The term *property* of a material refers to the response of that material to external stimuli. A considerable number of materials properties can be

used for materials research and development, for example, boiling point, melting point, elasticity, and ductility. Synthesis is defined as the methods of making materials, which generally can be natural or man-made, and processing means how materials are manipulated into useful components [12]. According to [10], in materials science, computational and experimental methods are usually integrated to provide good knowledge of the structures and properties of materials and how they are related to the synthesis and processing procedures.

2.2 Quantum mechanics and density functional theory in materials science

The Schrödinger equation is considered to be the standard for quantum mechanics. This equation explains the quantum behavior of atoms and forms the important relationship between material structure and material property. Its time-independent form is $H\Psi = E\Psi$, where H is the Hamiltonian operator for a molecular system, Ψ is a set of solutions, and E is the potential energy. As stated by Hohenberg and Kohn [13], the Hamiltonian H of any system is uniquely defined by the external potential, which is determined by a set of nuclear charges $\{Z_I\}$ and atomic positions $\{R_I\}$. The resulting potential energy is obtained by optimizing the set of Ψ. In other words, this principle can be summarized in the mapping below:

$$H(\{Z_I, R_I\}) \xrightarrow{\Psi} E \tag{2.1}$$

In quantum mechanics, a principal task is obtaining the *approximate* solution of the Schrödinger equation for the assemblies of atoms [14, 15]. Quantum mechanics underlies the density functional theory (DFT) [16], a standard tool in many branches of materials science. DFT was one of the computational methods that accelerated the computational revolution in materials science [17]. As explained by Keith et al. [18] in their review article, the term *functional* is a function of a function; in other words, functional takes a function as its input. As Parr and Weitao [19] emphasized, the fundamental of DFT is that it computes the quantum mechanical internal energy (E) of a system from an energy expression that has the functionals of electronic density ρ as a parameter. This principle is described in the equation below, where T[ρ] is the kinetic energy and V[ρ] is the potential energy, determined by the electron density ρ.

$$E[\rho] = T[\rho] + V[\rho] \tag{2.2}$$

DFT is widely used in materials modeling and quantum computation of electronic structures [20]. Today, the most common form of DFT is Kohn-Sham (KS-) DFT [16], which is highly accurate. KS-DFT assumes that the non-interacting electrons (of a fictitious system) have the exact ground state density of the real system of interest. Therefore, the energy functional in equation 2.2 can be expanded as below:

$$E[\rho] = T_s[\rho] + V_{eN}[\rho] + V_{ee}[\rho] + E_{XC}[\rho] \tag{2.3}$$

In the equation 2.3, $T_s[\rho]$ is the kinetic energy of non-interacting electrons, $V_{eN}[\rho]$ is the electron-nuclei potential, and $V_{ee}[\rho]$ is the Coulomb (i.e., Hartree or classical) energy. The important one is $V_{eN}[\rho]$, which is the exchange-correlation energy as a functional of density. The big O notation of KS-DFT is $\Omega(n^3)$, with n as the number of electrons [18].

2.3 Main challenges of the conventional methods in materials science

According to Liu et al. [10], experimental methods in materials science are, for example, physical property measurement, microstructure analysis, and synthesis. These experiments had a major role in the traditional discovery and the characterization of materials; however, they are generally time-consuming and demanding in terms of equipment, resources, and experience of the scientist [9]. For instance, if these kinds of conventional methods are used, the process from the discovery of new materials to the deployment can even take approximately 10–20 years [10].

Computational simulation methods are developed to solve the performance challenges imposed by traditional experimental methods. The computational tools such as DFT [16] have let scientists discover the phase and composition space much more efficiently, and they have brought in many applications, for example, materials modeling [1, 18]. However, Liu et al. [10] stated that the limitations to these computational methods do exist. Running the computational simulation programs requires high-performance computing resources. When a new system is studied, one cannot explicitly exploit the previous calculation results.

Moreover, although DFT is a fairly accurate way to compute the quantum

calculations of electronic structure, its approximations for the energy functional have unavoidable bias [18]. In addition, DFT is unable to provide good approximations of the fundamental gaps of semiconductors and insulators, an important quantity in materials research, such as impurity levels in doped semiconductors [21].

3. Basics of machine learning

Generally, machine learning is defined as a set of algorithms that can identify patterns in an example data or a past data, and then apply these patterns to predict the future data, to gain knowledge from data, or to do both [22, 23]. The core of machine learning is statistics theories, which are used in building mathematical models [23].
This section presents the main components of machine learning. According to Jung [24], a machine learning problem can be described as three main parts: data, model (also called a hypothesis space), and loss function. The choice of data, model, and loss function is a design choice; therefore, it is important to select and validate these components properly to optimize the final result.

3.1 Data

Data is a set of individual data points, which can be, for example, numerical values, pictures, videos, or documents [23, 24]. Depending on the applications, data points can represent, for example, the numeric values of certain molecular properties, known crystal structures, or chemical composition, and data representation usually affects the learning process of the model [1, 25]. The number of data points in a dataset, i.e., observations or training examples, can be denoted as n, which is an integer [26]. Generally, two important properties of a data point are *features* (also known as attributes or descriptors) and *labels* (i.e., targets our outputs) [18, 23].

Features are properties of a data point that can be obtained by, for example, computation or measurement. According to Schmidt et al. [9], in materials science, features that are relevant for the prediction of potential energy surfaces (such as atomic position) are the most researched features. The *feature space* is a vector space of all possible feature values of a data

point [24].

Label of a data point is the output to be predicted by the ML model. Labeled data is required in supervised machine learning and not needed in unsupervised machine learning. This difference is clarified in Section 3.2.

A vector of labels can be denoted as y as below. Specifically, y_i is the label of ith data point [26].

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

3.2 Model

A model, also known as a hypothesis space, is the set of predictor functions $h(\mathbf{x})$ that map the vector of features $\mathbf{x}$ of a data point to a predicted label $\hat{y} = h(\mathbf{x})$ [24, 25]. Generally, machine learning methods can be divided into three main categories: supervised learning, unsupervised learning, and reinforcement learning. Regarding the applications of ML in materials science, supervised learning is the most common and mature category [1, 8, 9]. Hence, in this section, supervised learning is emphasized and unsupervised learning is briefly introduced. Because reinforcement learning is not used as commonly as the other two classes [23], it is not the focus of this section.

3.2.1 Supervised learning

According to several sources about machine learning ([23, 26]), in supervised learning, the training data need input values (features) and the corresponding output values (i.e., labels or descriptors). Each input x_i has an associated label y_i, where i = 1, ..., n. Given a labeled data set of input-output pairs (x_i, y_i), the models learn a mapping from the input vector $\mathbf{x}$ to output y. The training dataset is collected from experiments, computations, or observations [25].

Figure 3.1 illustrates the basic workflow of supervised learning, summarized by [9]. In the collected or generated dataset, the values of the label are known. Necessary data cleaning is performed to make sure that the data is consistent and accurate. Then, a machine learning algorithm is

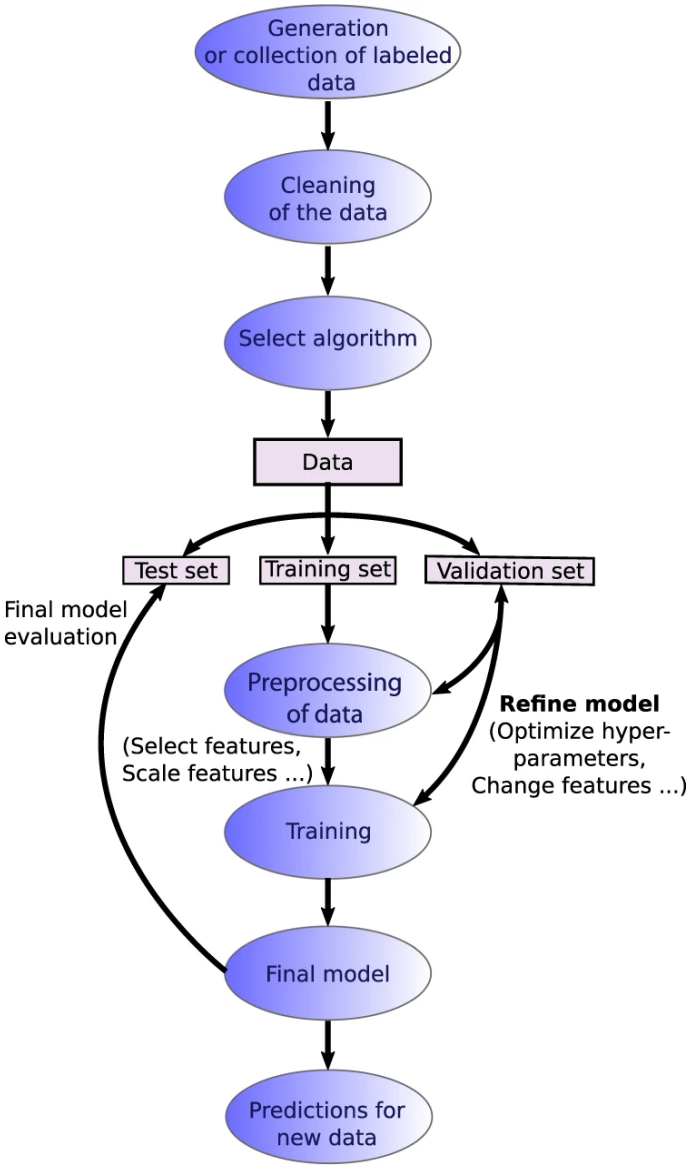

Figure 3.1. The basic workflow of supervised learning, illustrated by Schmidt et al. [9].

chosen. The data are split into a training set (used for training the model), a validation set, and a test set. Although the term *validation set* and *test set* are sometimes used interchangeably, for the sake of consistency in this thesis, the validation set is the data subset used for the modification of model hyperparameters and the test is the data subset used for the final model evaluation.

It is important to process the raw data into certain features that can be used as inputs for the chosen algorithm. Once this step is done, the model is trained by optimizing its performance, which is measured by a loss function. Model training often involves the modification of hyperparameters that control the training process, structure, and properties of the model. The prediction for new data can be obtained from the final model.

One type of supervised learning is classification, which is used to predict

the output values of a given discrete dataset, for example, classifying crystal structures or predicting whether a material is a metal or insulator [1, 18]. Classification learns from inputs $\mathbf{x}$ to outputs y, which belongs to a certain number of classes [23]. Classification can be binary classification (i.e., two classes), multi-class classification (i.e., more than two classes), or multi-label classification (i.e., data points belong to more than one class simultaneously) [23, 24]. Another type of supervised learning is regression, which predicts the output values of a continuous range, for example, polarizability or melting points [1, 25].

Regarding supervised learning methods in materials science, Himanen et al. [8] stated that the notable features of materials are, for example, physical properties and geometrical structures. Some applications of supervised learning are predictions of molecular properties or physical properties, such as formation energies.

3.2.2 Unsupervised learning

Unsupervised machine learning methods are used to discover patterns, knowledge, or underlying structure of the data points that have unlabeled value [23, 24, 18]. In contrast to supervised learning, unsupervised learning does not need data labeling, which might need a human to perform; therefore, labeling can be expensive and time-consuming. In this section, the two main classes of unsupervised learning are presented: clustering and dimensionality reduction.

Clustering groups the input data into subsets or clusters based on a similarity measure [22, 25]. A similarity measure is a metric that evaluates the similarity among data points; therefore, data points in the same subset are more similar to each other than with data points in another subset. A common clustering algorithm is k-means algorithm, which clusters each data point in the dataset to a given number k of different clusters [25, 24].

The second class of unsupervised learning is dimensionality reduction. The main purpose of this approach is to use only a number of highly useful and relevant features, which can save time and computational cost. Therefore, dimensionality reduction can be applied to reduce the high dimensionality of data by projecting the data to a lower-dimensional subspace [23, 24]. A common method of dimensionality reduction is principal components analysis (i.e., PCA) [27].

3.3 Loss function

To choose the best predictor h from the hypothesis space for a specific ML problem, we need a loss function. A principle of ML models is that they learn (or find) a hypothesis (in a given hypothesis space) that results in the minimum loss [24]. There are several ways to denote the loss function in literature, but the main idea is to compute the numeric difference between the true output y and the predicted output $\hat{y}$ [22, 23, 24]. Some common loss functions are mean squared error (function 3.1), Huber loss (function 3.2), and logistic loss.

$$L(\hat{y}, y) = \frac{1}{n} \sum_{i=1}^{n} \left(\hat{y}^{(i)} - y^{(i)}\right)^2 \tag{3.1}$$

$$L(\hat{y}, y) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{if } |y - \hat{y}| \leq \delta \\ \delta(|y - \hat{y}| - \delta/2) & \text{otherwise.} \end{cases} \tag{3.2}$$

The Huber loss is robust to outliers because in the case of $|y - \hat{y}| > \delta$ (where δ is a given parameter), the error is not squared. Thus, these data points have a smaller effect on the total loss over the dataset and hence the resulting fit.

3.4 Common machine learning algorithms in materials science

Generally, some common machine learning algorithms are naive Bayes classifiers, k-nearest neighbors, decision trees, kernel methods, and artificial neural networks [1, 25]. To find a suitable ML model for a specific application, it is important to perform model selection and model validation [28]. In this section, three of the most common algorithms used in materials science are chosen to be reviewed. They are neural networks, kernel ridge regression, and decision trees.

3.4.1 Artificial neural network

Several review articles [9, 18] suggested that artificial neural networks (ANNs) are one of the most popular machine learning methods in materials science due to their performance.

Feed-forward neural network is considered to be the most common class of ANN [29]. Neural networks can have multiple hidden layers and more than one output. The simple feed-forward neural network in figure 3.4 has

two hidden layers and 10 outputs (Y_0 to Y_9). An artificial neuron, which is illustrated as a circle in figure 4.1, represents an arbitrary mathematical function. Although the architecture of ANN, the main points of the ANN algorithm can be summarized and described in figure 3.2.

- The input values X_i in the input layer (the orange circles in figure 3.4) feed into each of the neurons in the hidden layer L_1.
- The hidden layer L_1 computes the activation functions and passes the results to the next hidden layer, i.e., layer L_2
- The final results are obtained from the output layer.

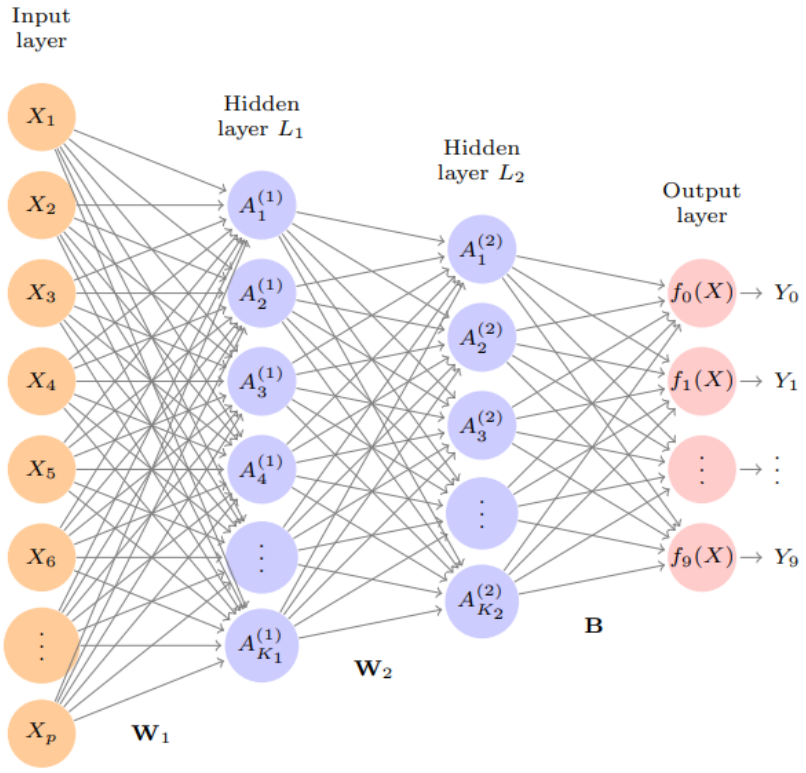

Figure 3.2. A simple illustration of a neural network with two hidden layers by James et al. [26]. Each layer may have hundreds of neurons.

There are several choices for the activation function. For example, common ones are logistic sigmoid function, ReLU (rectified linear unit), and hyperbolic tan function [26]. What these functions return is presented in table 3.1.

Activation function	Formula
Sigmoid	$f(x) = \frac{1}{1+e^{-y}}$
ReLU (rectified linear unit)	$f(x) = max(x, 0)$
Hyperbolic tan	$f(x) = tanh(x)$

Table 3.1. Table of some common activation functions.

Deep neural networks are neural networks with more than two layers. They are the core architecture of deep learning, a sub-field of machine learning that is gaining a lot of research interest in materials science [30]. Some successful applications of neural networks in materials science are material property prediction [3], mapping of materials behavior to materials processes, and development of interatomic potentials [7].

However, a common criticism of ANNs in material property prediction is that they require an input dataset with sufficient representative data to extract some pattern among the data [10]. In addition, more complex neural network architectures such as convolutional neural networks are difficult to interpret; therefore, it is generally not straightforward to understand how they get the final output. Model interpretability is clarified more in Section 3.5.

3.4.2 Kernel methods and kernel ridge regression

Kernel methods are one of the most relevant machine learning models for materials science. According to several ML books [24, 31], applying directly linear machine learning model may not be suitable if the relation between feature vector **x** and label y is non-linear. A solution is to apply kernel methods, a set of methods that transform the input feature vector **x** (i.e., the raw input data) into a higher-dimensional vector **x'**. The main idea of kernel methods is to derive non-linear versions of simple linear models such as logistic regression or linear regression. Kernel methods use kernel functions, which generally can be notated as $k(\mathbf{x}, \mathbf{x'})$.

In several sources [24, 28], it is emphasized that successful application of kernel methods requires valid kernel functions, which can be found from a chosen feature space mapping ϕ that maps the raw feature vector **x** to a new feature vector **x'** = ϕ(**x**), and **x'** belongs to a higher-dimensional feature space. Simple and common kernel functions are listed in the table below, where c, c_1, and c_2 are adjustable scalar parameters.

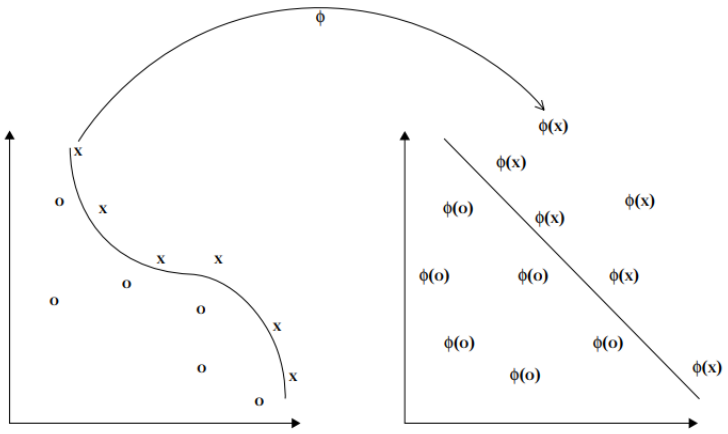

Figure 3.3. A high-level illustration of function ϕ mapping the input data into a higher-dimensional feature space to reveal the linearity among data points [31].

Kernel name	Definition
Linear	$\mathbf{x} \cdot \mathbf{x'} + c$
Polynomial kernel of degree d	$(c_1(\mathbf{x} \cdot \mathbf{x'}) + c_2)^d$

It is mentioned in some review articles ([9, 25]) that kernel ridge regression (KRR) is one of the most widely-used kernel methods in materials science. KRR combines ridge regression with the kernel trick for non-linear problems [23]. In other words, KRR is a non-linear extension of ridge regression. The loss function of KRR combines squared error loss with *l2*-norm regularization. The purpose of regularization is to penalize the models (especially more complex models such as neural networks) and, as a result, to avoid large variance.

KRR is often applied in materials research, for example, for the prediction of materials properties, the development of model Hamiltonians, and the finding of density functionals [25].

3.4.3 Decision tree

Decision trees are a family of non-linear supervised machine learning methods. These trees have graph structures that determine a value or an output. According to [9], the decision tree consists of nodes, which represent certain tests, for example, the root node in figure 3.4 represents the question "is the atomic number > 15?". Decision trees can be a classifier or a regressor. In a regression case, a node assigns a value, and the common function to measure the quality of a tree split is squared error. In a classification case, a node in the tree carries a condition that divides the input data into categories. The splitting conditions are determined by a criterion defined by the user, for example, to minimize the entropy after

the split or to maximize the information gain [32].

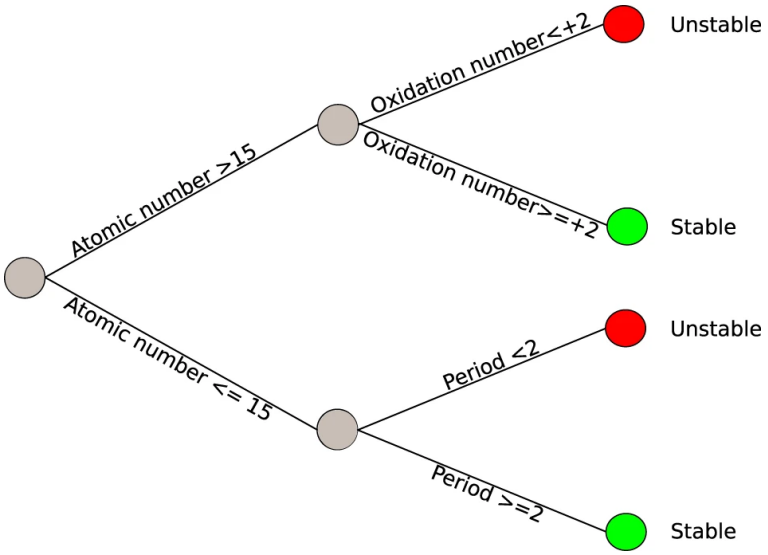

Figure 3.4. A simple illustration of a decision tree classifier that predicts if the material is stable or unstable [9].

According to [33], the decision trees are fairly simple to interpret and fast. However, they have some disadvantages. Too complex trees (for example, trees that have very high tree depth) cannot generalize the data well; thus, it will result in overfitting, in which the model fits the training data well, but fits the unseen data poorly. Therefore, a proposed solution is setting the maximum depth of the tree [33]. Another approach is using decision tree ensemble methods, for example, random forests or extremely randomized trees. Schmidt et al. [9] suggested that the main purpose is to generate many independent decision trees with a randomized training process, instead of training only one decision tree. This randomization can be done by, for example, using a random subset of the features or using a random subset of the training dataset to generate the trees.

3.5 Interpretability of model prediction

Machine learning is often criticized by the scientific community of computational chemistry and materials science for being a "black box", because many ML algorithms and predictions are considered difficult and complex to understand, especially for researchers that are new to machine learning [9, 18]. As the solution, the most common approaches for model interpretability are:

- Build more understandable machine learning models and avoid incom-

prehensible models.

- Extract knowledge from the results of a machine learning model that is hard to understand. This approach is the main focus of this section.

It is highly important to interpret a model prediction (especially when it comes to complex models such as deep neural networks) in order to understand why a machine learning model makes such prediction. One tool for the model interpretability is SHAP [34] (i.e., Shapley additive explanations), a method that uses the Shapley values [35] of game theory to explain the output of machine learning models, such as tree ensemble methods.

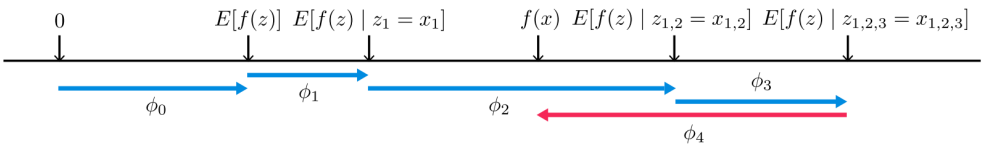

Figure 3.5. The piece-by-piece prediction of the model, from the base value to the final output $f(x)$. When the condition involves a feature i, the SHAP value ϕ_i attributes to that feature the change in the expected prediction [34].

The article [34] by the creators of the SHAP Python package presented the principle of SHAP, which is illustrated in figure 3.5. SHAP values of the features i (i.e., ϕ_i) explain how the machine learning model goes from the base value to the final output $f(x)$, where x is an input. These SHAP values are Shapley values of a conditional expectation function $E[f(z)|z_S] = f_x(z')$, where S is the set of non-zero indices in $z' \in \{0,1\}^n$ (with n as the number of simplified input features). The base value $E[f(z)]$ is what will be predicted if the input features are not known, and the base value can be obtained from, for example, a training dataset of defaults.

Note that figure 3.5 only represents one ordering of computing the SHAP values of features from 1 to 4 (i.e., from ϕ_1 to ϕ_4). If the model is non-linear or the input features are dependent, the order of adding features to the expectation matters.

In materials science, SHAP analysis has been utilized in, for example, a study [36] on the explanation of the hydrogen adsorption of the defective nitrogen-doped carbon nanotubes by DFT and machine learning. This study evaluated that the SHAP analysis could provide a better understanding of feature importance. SHAP analysis is potential to be applied more broadly in materials research publications so that the model output can be explained better.

Another approach to model interpretability is attentive response map, which can be used to indicate the most important parts of the image that affect the decision of the image classification model [37]. As an example, Zilleti et al. [5] used the attentive response map in their research to visualize how their convolutional neural networks can classify over 100,000 crystal structures from diffraction fingerprints. The response maps suggested that their neural networks identified the position of the diffraction peaks and their arrangement as the most important features to classify the crystals. This approach is promising for future research on how the model recognizes the crystal structures from a massive amount of data.

4. Data for materials science

Materials science projects such as the Materials Genome Initiative [38] have been initiated to develop the data-driven approach in materials science, which is the approach of systematically obtaining insights and information from data of materials [8]. Data is one of the most important components in machine learning; therefore, this section focuses on an overview of the current data sources and data quality in materials research. The role of quantum chemical computation in data generation for ML is emphasized.

4.1 Data sources in materials science

In the article by Keith et al. [18], the authors summarized how quantum chemical computation can generate useful data for machine learning applications. Firstly, computation chemistry methods obtain the relevant geometry and total ground state energy of the system through computer software, and then quantum mechanics and statistical mechanics are applied to find the properties, for example, band gap, pressure, and polarizability.

Compared to conventional experiments, computational methods are much more potential to generate high-quality and useful data for materials research in a shorter time period [18]. The property computation for thousands of compounds has been accelerated thanks to high-throughput computational methods [1]. Specifically, DFT, a computational method covered in Section 2, has led to the development of databases that contain properties of known and hypothetical systems such as crystal structures and alloys [1, 9].

Zhang and Ling [39] stated that the limited amount of data is actually an obstacle in computational materials research. According to [9], this

issue is especially concerning when the target (i.e., the label y of data) can only be obtained from expensive experiments, for example, the critical temperature of superconductors. To tackle this challenge, databases for known materials and theoretical materials have been built. This had led to the publicly accessible databases, for example, for the structure and property of solid-state materials, which can be used for machine learning applications. The computational calculation and simulation as well as the high-throughput experimental methods are producing more data for machine learning applications in materials science [9].

Regarding computed structures and properties, there are, for example, Open Quantum Materials Database [40], Novel Materials Discovery Laboratory (NOMAD) [41], Computational Materials Repository [42], and AFLOWlib [43]. Open Quantum Materials Database [40] is a database of thermodynamic and structural properties calculated by DFT for over one million materials, focusing on the inorganic crystal structures. NOMAD is a public database that contains input and output files from calculations using a wide variety of electronic-structure codes [41]. Computational Materials Repository [42] collects, stores, and analyzes the data from electronic-structure codes. Last but not least, AFLOWlib [43] is a database for the construction of materials science electronic structures.

Regarding experimental structures and properties, the public databases are ICSD (Inorganic Crystal Structure Database) and Crystallography Open Database (which contains structures of inorganic, organic, and metal-organic compounds) [1].

4.2 Quality of data for materials research

The performance and result of supervised machine learning (i.e., the most common machine learning class in materials science) largely depend on the quality of the input data [9]. Dataset and data representation are choices made by humans, and if the dataset is not suitable or representative enough, the ML model might be inaccurate and inefficient [1, 18, 24]. Moreover, as illustrated in figure 3.1 in Section 3, data should be cleaned before training the models in order to make data consistent and accurate.

Keith et al. [18] addressed the criticism of the chemistry and materials science community about the data quality for ML applications. According to Himanen et al. [8], it is important to know the quality of datasets on databases for materials science. However, the quality of data for materials

research has remained challenging to evaluate because of the high bias in computational data and the difficulty of controlling experimental errors. In addition, data cleaning tools for materials data have remained a challenge.

Regarding the accuracy of data generated by computational methods, it is affected by unavoidable bias, which is the offset from the experimental ground truth. Approximations of DFT calculation are fairly accurate, but they come with unavoidable systematic errors; therefore, the data generated by DFT have some bias and outliers [18]. In machine learning, if the dataset has a lot of outliers, that will disturb the prediction of the model.

Regarding data generated by experiments, it is difficult to control errors in those data because of several factors, such as material imperfections, interactions between the materials and the environment, and errors from equipment. Therefore, the overall estimation of data quality in materials science is challenging because of the high bias in computational data and the difficulty of controlling errors in experiments. The authors proposed that a possible solution is an error extrapolation scheme, for example, the NOMAD's computational error estimation. Moreover, systematic data collection from both experimental and computational methods will help facilitate bias and variance assessments in the future.

5. Applications of machine learning in materials science

In materials science, there are demands for more efficient and less expensive computational methods that can accurately predict, for example, material properties [10]. As summarized in figure 5.1 created by [10], the applications of machine learning in materials research can be generally divided into three main classes: material property prediction, new materials discovery, and other purposes. Section 6.1 covers the basics of material property prediction and some notable studies. Section 6.2 introduces the main machine learning applications in materials discovery (i.e., crystal structure prediction and composition prediction) and presents some related research publications. Regarding other machine learning applications in materials science, there are studies on the development of interatomic potentials [7], finding density functionals [44], and mapping behavior materials to materials processes. The development of interatomic potentials is chosen to be briefly introduced in part 6.3.

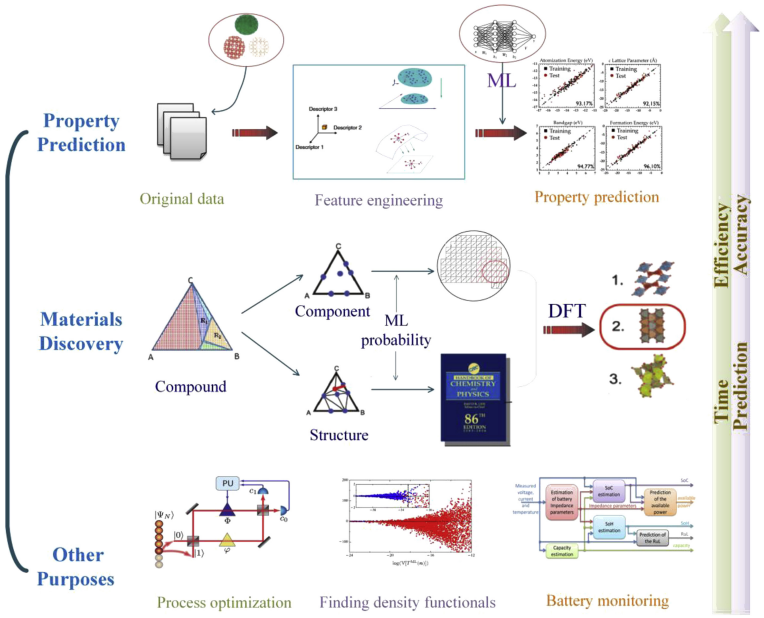

Figure 5.1. the applications of machine learning in materials research can be generally divided into three main classes: material property prediction, new materials discovery, and other purposes such as [10].

5.1 Prediction of material properties

Generally, in machine learning prediction of material properties, the relations (often non-linear) between certain material properties and the factors affecting those properties are obtained from given data [10]. According to [10, 25], data from quantum mechanical computations can be utilized for machine learning to predict certain materials properties. For example, materials can be reduced to numerical fingerprints that represent those materials and act as feature vectors (i.e., *attribute vectors* in figure 5.2). Then, chosen ML models can be trained from the input fingerprints, and the trained models are utilized to predict the properties of materials. The studies on machine learning prediction of material properties can be two main groups: prediction of microscopic property and prediction of macroscopic material performance.

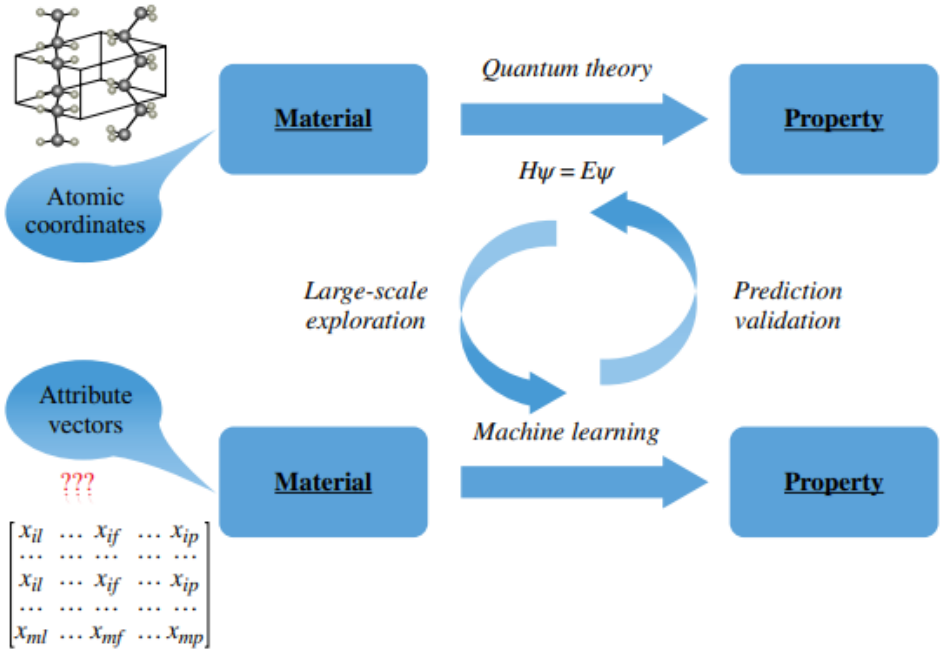

Figure 5.2. The key idea of how machine learning can support quantum mechanical computation [25].

Regarding microscopic property prediction by machine learning, the applications are mainly for, e.g., band energy, and molecular atomization energy [10]. The review article by Schmidt et al. [9] also suggested that there have been many studies using machine learning to predict the band gap, an important electronic property in the design of materials such as conductors and insulators. Band gap is the distance between the electrons' valence band and the conduction band. Band gap is closely related to the HOMO-LUMO gap, which is the energy gap between the highest occupied

molecular orbital and the lowest unoccupied molecular orbital.

As an example, Zhuo et al. [45] published a study in 2018 on the direct prediction of experimental band gaps. The dataset includes over 6000 band gaps measured from past experiments. First, the authors classified materials as either metal or non-metal using support vector classification with a radial basis function (RBF) kernel, because this machine learning model gave the best result among the tested models: an accuracy of around 0.92. Then, the band gap of the non-metals was predicted by support vector regression (i.e., SVR). This SVR model was claimed to predict the band gap at a reduced computational cost and with better performance than DFT calculations. Moreover, this model also has let the authors estimate the band gap of 94095 different compounds.

Regarding macroscopic performance prediction, [10] suggested that artificial neural networks have been widely applied. For example, a study by Xie et al. [3] in 2018 trained the crystal graph convolutional neural network to predict the bulk and shear moduli. The dataset contained 1585 materials. As the result, the test set errors respectively were 0.105log(GPa) and 0.127log(GPa) for bulk and shear moduli. The neural network in this study was also claimed to have a good generalization for the unseen dataset.

5.2 Discovery of materials

Conventional computational methods for materials discovery have to search for new or higher-performance materials through enormous data of compositions and structures [9]. Therefore, ML is a promising candidate to accelerate the material discovery, a process needed to be reduced in terms of time and cost [46]. ML applications in material discovery are mainly for crystal structure prediction and composition prediction.

Crystal structure prediction can reduce some unnecessary experiments related to material structure and reduce the cost of DFT calculation [10]. However, one of the biggest challenges in materials design is to predict the crystal structure of a material that is not yet synthesized [47]. An approach to crystal structure prediction by machine learning was by Fischer et al. [48]. The authors used a database of known crystal structures [49] to predict the probability that a material with a given composition will have a given structure type. As the result, the correct structure was predicted in 90% of the cases in the first five guesses, and 62% when picking the structures according to their frequency in the dataset.

In component prediction, the ideal criterion to evaluate the thermodynamic stability is the energetic distance to the convex hull [9]. In a study conducted by Faber et al. [4], the authors applied kernel ridge regression to compute the formation energies of two million elpasolites crystals (stoichiometry ABC_2D_6) that consists of main-group elements up to bismuth. The training set had 10^4 compositions, and the mean absolute error was 0.1 eV/atom. As the result, 90 unique, new structures were predicted by the model to lie on the convex hull. Models such as the one in this study are promising for the new generation of high-efficiency composition screening [1].

5.3 Development of interatomic potentials

The interatomic potential describes the interaction between a pair of atoms or the interaction of an atom with a group of atoms in a condensed phase. In interatomic potentials, a presupposed potential energy surface (abbreviated as PES) that includes the interactions between atoms is modeled by analytical functions [18]. Interatomic potentials are heavily used for molecular modeling and soft materials modeling, e.g., polymeric materials.

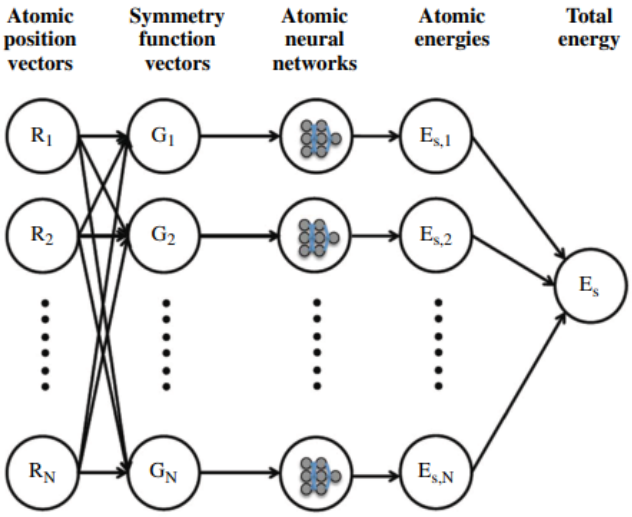

Figure 5.3. A neural network that maps the coordinates of a set of N atoms to the total energy E_s [25].

A perspective of machine learning for the development of interatomic potentials was proposed by Behler [7], which involves a feed-forward neural network, illustrated in figure 5.2. The main idea is that the geometry of the system (i.e., the atomic positions of the system) with N atoms can be used as an input for ANN to predict the total potential energy of the

system. Again, the atomic positions are presented as an input vector of fingerprints. Symmetry functions are essentially a set of functions of the atomic positions, allowing for a transformation of the R_i vectors to the G_i vectors. The atomic neural network was claimed to be capable of predicting $E_{s,i}$ when the G_i vector of an atom in a new configuration is given.

6. Demonstration of redox potential prediction by machine learning

After presenting the basics of machine learning in Section 3, this thesis now gives a simple example of how supervised machine learning methods can be trained to predict chemical properties. In this example, the prediction is for the redox potential (i.e., oxidation / reduction potential). Moreover, this demonstration provides a hands-on experience with a few basic machine learning algorithms and some common Python libraries in machine learning, for example, Scikit-learn [50].

6.1 Problem formulation

The input dataset was generated by DFT computation by the Computational Chemistry research group at Aalto University. There are 1989 data points and no missing values among the data points. Each data point is a known molecule. For this problem, there are nine numerical features, presented in the first nine columns in figure 6.2. They are HOMO, LUMO, and HOMO–LUMO energy gap of molecules A, AH (molecule A bounded by one hydrogen), and AH2 (molecule A bounded by two hydrogen atoms). HOMO–LUMO energy gap is the energy difference between HOMO and LUMO energies (as shown in figure 6.1), also known as the Kohn-Sham gap.

Because the goal is to train the machine learning models to predict the redox potential of molecules, the label (i.e., target or output) is the redox potential column.

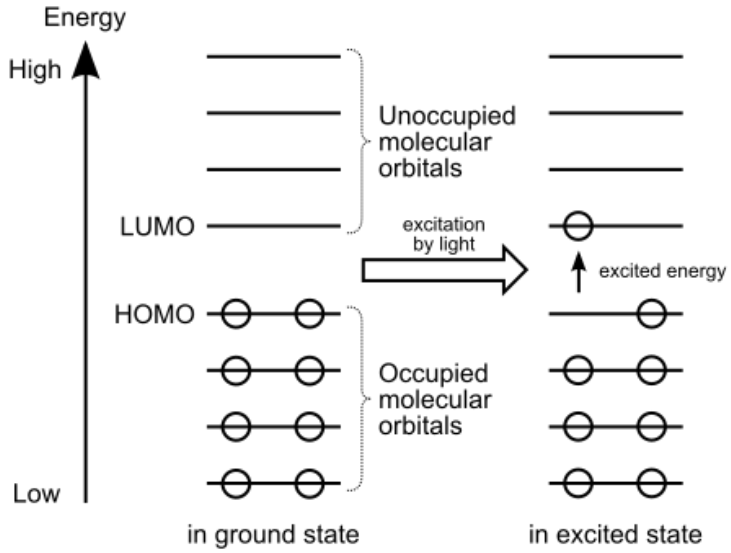

Figure 6.1. Diagram of HOMO (Highest Occupied Molecular Orbital) and LUMO (Lowest Unoccupied Molecular Orbital) [51].

	HomoA	LumoA	GapA	HomoAH	LumoAH	GapAH	HomoAH2	LumoAH2	GapAH2	redoxpot
0	-7.9848	-3.6582	4.3265	-7.5462	-3.5189	4.0272	-6.0675	-0.4076	5.6599	0.9121
1	-6.6740	-3.3543	3.3197	-6.6104	-3.0729	3.5374	-5.8577	-0.2250	5.6327	0.8415
2	-6.3660	-3.3184	3.0476	-6.2117	-3.0280	3.1837	-5.9100	-0.3589	5.5510	0.8053
3	-7.1440	-3.6882	3.4558	-6.8373	-3.5720	3.2653	-5.9666	-0.4971	5.4694	0.9603
4	-7.5521	-3.5521	4.0000	-7.2218	-3.3578	3.8640	-5.9178	-0.1763	5.7415	0.8734
...	...	...	...	...	...	...	...	...	...	...

Figure 6.2. A part of the dataset. In this demonstration, the first nine columns form the feature vector, and the last column (*redoxpot*) on the right acts as the label.

6.2 Methods

6.2.1 Models

Non-linear decision tree regression and kernel ridge regression (KRR) were chosen for this demonstration. The motivations behind this selection are:

- Redox potential (i.e., the label of this problem) is a numerical value; therefore, regression is more reasonable than classification.

- As shown in figure 7.2, because the interaction plots between some features and the label seem to be non-linear, non-linear methods were tested.

- According to some research publications [9, 25], non-linear decision tree-based algorithms and kernel-based methods are some of the most

relevant methods in materials research. Moreover, compared to other methods, algorithms of decision trees and KRR are fairly simple and fast [33, 52].

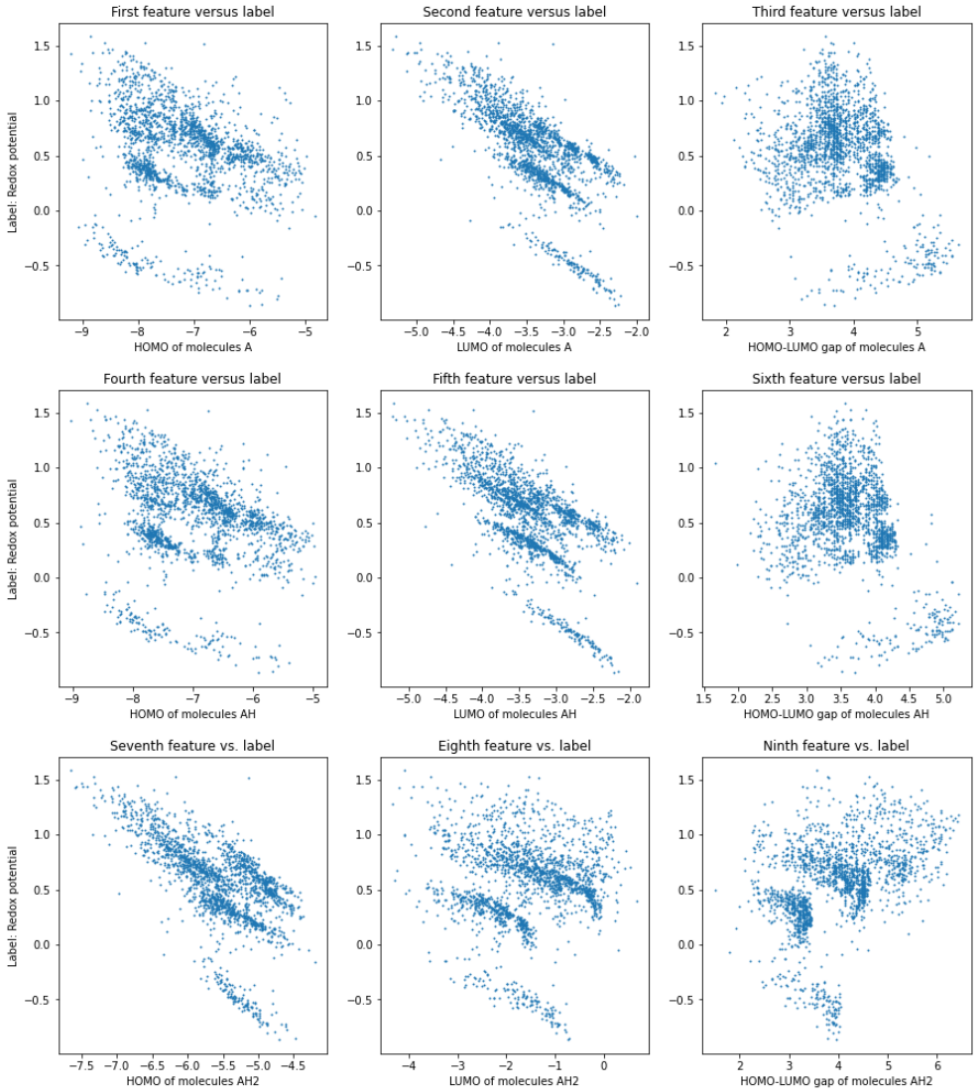

Figure 6.3. Interaction plots between each of nine features and the label.

Computation was done in the Python programming language and used some common Python libraries such as Numpy, Pandas [53], and Scikit-learn [50]. Generally, the Numpy library supports the operations on multi-dimensional arrays and matrices. Pandas is a library for data manipulation and analysis. Scikit-learn is a Python library with many built-in algorithms for supervised and unsupervised machine learning methods.

When using KRR, the valid kernel function and the suitable values for regularization parameter α need to be found [28]. As a solution, the handy class `GridSearchCV` [54] of Scikit-learn was used to search over the parameter values (which are specified by the users) for the best estimator.

This estimator has the best setting of parameters, i.e., the setting that gave the best results from the holdout dataset.

As mentioned in Section 3.4.3, trees that have too high depth may cause overfitting. Therefore, grid search was also used to find the most suitable maximum tree depth for decision tree regression.

6.2.2 Training set, validation set, and test set

Although there is no golden ratio to split the dataset into three different sets (i.e., a training set, a validation set, and a test set), a common practice of splitting the dataset that has several thousand data points is to use 80% of the data for k–fold cross-validation and 20% of the data for the final test set. This practice was applied for the demonstration. In k–fold cross-validation, the training set is divided into k subsets, for instance, five subsets (as in figure 6.4). The model is trained on $k - 1$ of the folds. Then, the resulting model is validated on the remaining of the data by calculating the metrics, for example, accuracy.

The purpose of k–fold cross-validation is to try training and validating the ML models on different subsets of the data, instead of having only one validation set [24]. Thus, using k–fold cross-validation can avoid overfitting. The k–fold cross-validation is especially useful if the dataset is highly diverse or the size of the dataset is small. For this demonstration, the default k value in `GridSearchCV` [54] were used (i.e., k = 5).

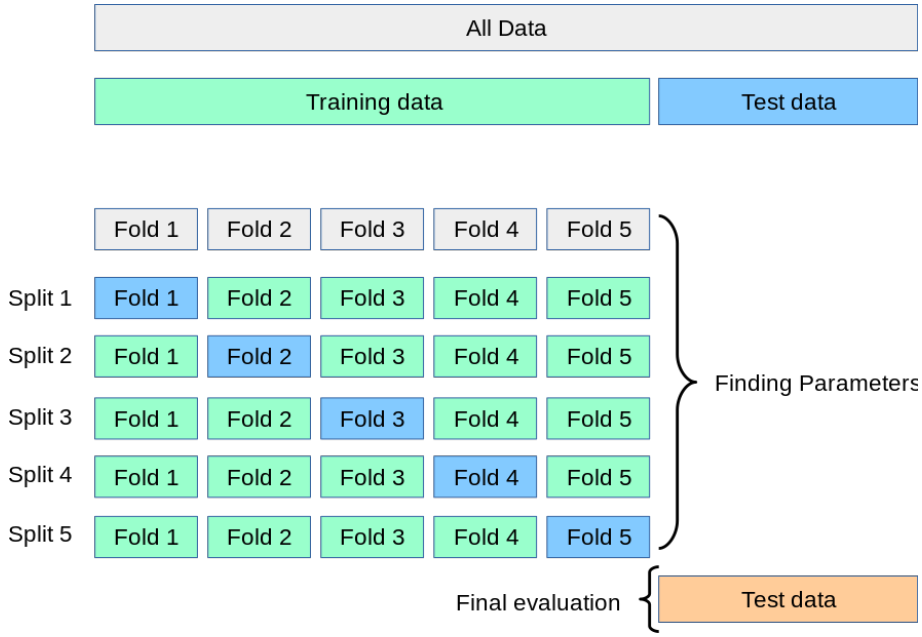

Figure 6.4. The dataset is used for 5–fold cross-validation. The final model evaluation is performed on a separate test dataset.

6.3 Results

6.3.1 Results of kernel ridge regression

`GridSearchCV` [54] was used to find the best regularization strength value (α) (within a range from 0.5 to 3.5) and kernel function for kernel ridge regression. As a result, the best regularization strength obtained was 0.5. The best kernel function was polynomial with the degree of 3, as shown in the first row of the table below. This parameter setting resulted in a mean training score of approximately 0.828 and a mean validation score of approximately 0.814.

Parameters	Mean training score	Average validation score
{'alpha': 0.5, 'degree': 3, 'kernel': 'polynomial'}	0.828137	0.814869
{'alpha': 0.7, 'degree': 3, 'kernel': 'polynomial'}	0.825552	0.813175
{'alpha': 0.8999999999999999, 'degree': 3, 'kernel': 'polynomial'}	0.823417	0.811650
{'alpha': 1.0999999999999999, 'degree': 3, 'kernel': 'polynomial'}	0.821586	0.810271
{'alpha': 1.2999999999999998, 'degree': 3, 'kernel': 'polynomial'}	0.819982	0.809020
{'alpha': 1.4999999999999998, 'degree': 3, 'kernel': 'polynomial'}	0.818553	0.807877
{'alpha': 1.6999999999999997, 'degree': 3, 'kernel': 'polynomial'}	0.817266	0.806829
{'alpha': 1.8999999999999997, 'degree': 3, 'kernel': 'polynomial'}	0.816095	0.805863
{'alpha': 2.0999999999999996, 'degree': 3, 'kernel': 'polynomial'}	0.815022	0.804968
{'alpha': 2.3, 'degree': 3, 'kernel': 'polynomial'}	0.814032	0.804135
{'alpha': 2.4999999999999996, 'degree': 3, 'kernel': 'polynomial'}	0.813114	0.803358
{'alpha': 2.6999999999999993, 'degree': 3, 'kernel': 'polynomial'}	0.812258	0.802629
{'alpha': 2.8999999999999995, 'degree': 3, 'kernel': 'polynomial'}	0.811457	0.801944
{'alpha': 3.0999999999999996, 'degree': 3, 'kernel': 'polynomial'}	0.810704	0.801298
{'alpha': 3.2999999999999994, 'degree': 3, 'kernel': 'polynomial'}	0.809994	0.800688
{'alpha': 0.5, 'degree': 2, 'kernel': 'polynomial'}	0.789167	0.783817
{'alpha': 0.7, 'degree': 2, 'kernel': 'polynomial'}	0.786234	0.781149
{'alpha': 0.8999999999999999, 'degree': 2, 'kernel': 'polynomial'}	0.784027	0.779119
{'alpha': 1.0999999999999999, 'degree': 2, 'kernel': 'polynomial'}	0.782283	0.777505

Figure 6.5. Average training score and validation score obtained from different parameter sets of KRR. The parameter setting with a regularization strength of 0.5 and third-degree polynomial kernel function (i.e., the first row) resulted in the highest scores. For the sake of simplicity, parameter sets that resulted in low scores are not shown in this picture.

Finally, as a final check, a separate test set was used to evaluate the KRR model with the regularization strength of 0.5 and the third-degree polynomial kernel function. The mean squared error and R^2 score (i.e., coefficient of determination) were obtained as metrics. The mean squared error was approximately 0.025 and the R^2 score was approximately 0.87. The best R^2 score is 1.0; thus, the better the model prediction is, the closer

R^2 is to 1.0.

6.3.2 Results of decision tree regression

By using grid search (with 5-fold cross-validation), the most suitable parameters for decision tree regression were searched over a range of maximum tree depth from 2 to 10. The criterion to measure the quality of a tree split was the squared error. Based on the results (which are clarified below), the most suitable maximum tree depth was chosen as six. This parameter setting resulted in a mean training R^2 score of approximately 0.90 and a mean validation R^2 score of approximately 0.794.

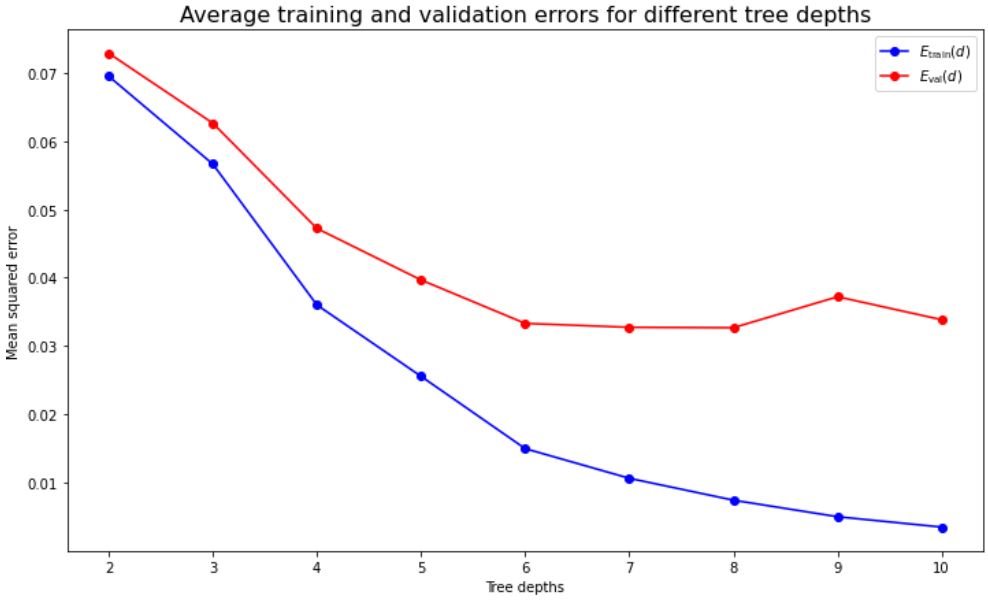

Figure 6.6. Average training and validation error of different maximum tree depths. Blue line represents the result from the training set and the red line represents the result from the validation set.

Mean squared errors and R^2 scores for different maximum tree depths are shown in figure 6.6 and 6.7, respectively. The blue line represents the result from the training dataset and the red line represents the result from the validation dataset.

As the tree grows bigger (i.e., the maximum depth grows from 2 to 6), the training errors and validation errors decrease (shown in figure 6.6) and the R^2 score increases (shown in figure 6.7); therefore, it can be said that the tree predicts the data better. The validation errors and R^2 scores obtained from the maximum tree depth of 6, 7, and 8 are quite similar, as shown by the even red lines. However, when the maximum tree depth becomes 9, the validation errors increase, and the validation R^2 score decreases. This is when the tree starts to overfit the data. Comparing the three maximum depths of 6, 7, and 8, 6 is a reasonable choice to avoid overfitting on new

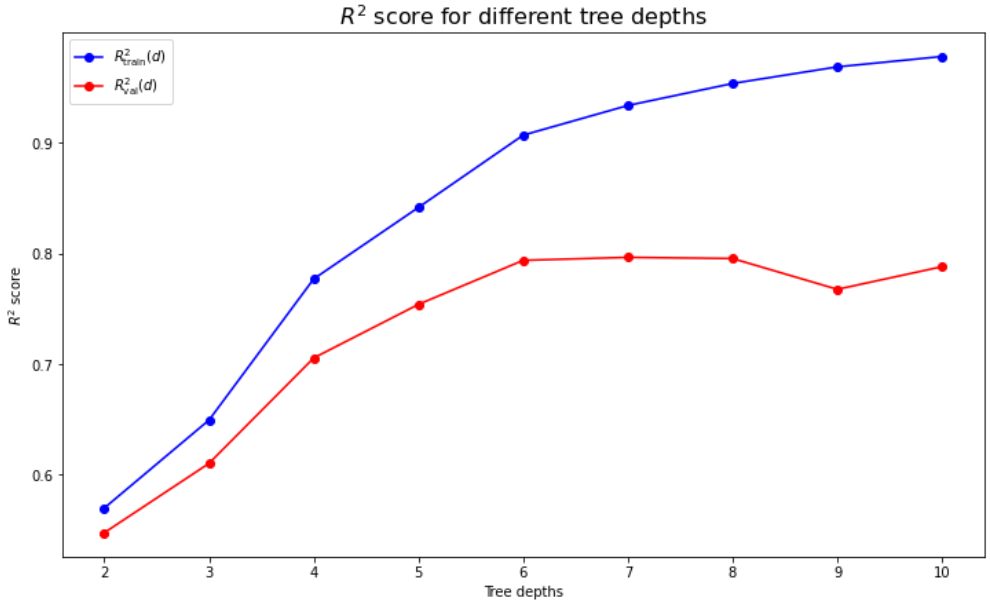

Figure 6.7. R^2 scores were obtained from different maximum tree depths from 2 to 10.

data. Therefore, tho chosen maximum tree depth is 6.

A separate test set was used to evaluate the decision tree regression model with the maximum depth of 6 and squared error as the criterion. The mean squared error obtained from the test set was approximately 0.025 and the R^2 score was approximately 0.844.

6.4 A brief comment for the demonstration

The mean validation R^2 score obtained from KRR was 0.814 and from decision tree regression was 0.794. They are not too different, but the KRR is slightly better.

Although the results seem reasonable, further experiments can be done to improve the performance, for example, testing random forests, as mentioned in Section 3.4.3. According to [55], random forest fits a number of decision trees on different subsets of the dataset and calculates the average to avoid overfitting and enhance the model accuracy.

Moreover, the understanding of model prediction can be improved by experimenting with the SHAP Python library [34], which is covered in Section 3.5. SHAP analysis [34] was not conducted because it is beyond the scope of this simple demonstration. Therefore, as a further experiment, SHAP can be applied to gain a deeper understanding of the model output.

7. Conclusion

This thesis aimed to examine how machine learning can be utilized for materials development and to evaluate how developing the machine learning paradigm in materials development is. To achieve this goal, the thesis focuses on the basics of machine learning, the overview of current data resources for materials science, and some notable machine learning applications in materials development in recent years.

Recent studies have shown some promising results of machine learning applications in materials science, for example, materials property prediction, new materials discovery, and interatomic potentials development. Therefore, machine learning is the potential to accelerate materials research and development. The past years have witnessed the development of various database infrastructures for materials science, as well as the development of machine learning algorithms and publicly accessible machine learning tools, for example, Scikit-learn [50]. As the number of research publications about machine learning in materials development is increasing, it is predicted that machine learning is promising to speed up the future of materials science.

However, according to Schmidt et al. [9], materials science has only utilized machine learning in recent years. It is believed that researchers have only scratched the surface of machine learning applications in materials science. Moreover, there are still many limitations to machine learning in materials development, for example, data quality, data quantity, and model interpretability. Therefore, it is still too early to say that the machine learning paradigm in materials development has become mature.

This section summarizes the main challenges of machine learning in materials development and the potential solutions to these challenges. According to Himanen et al. [8], data quality is one of the significant problems of data infrastructures for materials research. Data quality is

important to increase the acceptance of databases; however, data quality is challenging to quantify. Bias and variance are the indicators of quality. As the solution, systematic data collection and new extrapolation schemes are expected to assess bias and variance in the future.

Another challenge of machine learning for materials development is data quantity. Data generation in materials science is still relatively expensive and slow; therefore, a typical dataset for a specific problem in materials research is often only hundreds or thousands, or even fewer, good-quality data points [1]. The size of the dataset in supervised learning methods (i.e., the currently dominating machine learning class in materials research) is important because the dataset should have a sufficient amount of data so that it can be split appropriately into a training set, a validation set, and ideally a test set [9]. As a solution, Butler et al. [1] suggested that if the size of the dataset is small, meta-learning, a subfield of machine learning, can be used. The basic concept of meta-learning is that the algorithms learn the metadata about their own learning processes and experiments. [56].

Last but not least, a major criticism of ML is model interpretability. As a solution, it is suggested that post hoc analyses such as SHAP [34] and attentive response maps can be used more widely in materials science. These methods can help researchers to gain a better understanding of model interpretability and the trust in complicated yet powerful machine learning models, such as neural networks.

Based on the aims of this thesis, the limitations of this thesis work can be identified. It is possible that the thesis did not identify all of the relevant articles and information about machine learning in materials development. In addition, this thesis only covered the basics of machine learning, provided an overview of materials data, and presented some machine learning applications in materials science. Therefore, more research is needed to gain a deeper knowledge of machine learning in materials development. Specifically, the complex machine learning model such as neural networks, more machine learning applications in materials science, and more aspects of data for materials science can be studied further.

Regarding future research direction, [9] suggested that there will be a difference between methods depending on the quantity of data. For large datasets, e.g., over 10^5 materials, deep neural networks are predicted to be in favor because of their superior performance. For limited datasets, methods such as meta-learning are promising to allow scientists to opti-

mize the results obtained with a small dataset. Moreover, optimization algorithms such as Bayesian optimization [57] is research direction for materials researchers to explore further [9]. As mentioned above, more research efforts are needed to tackle the current challenges imposed by machine learning in materials development.

A. Appendices

A.1 List of features used in the demonstration of redox potential prediction

Nine numerical features of the dataset:

1. HomoA is the Highest Occupied Molecular Orbital (HOMO) of a molecule A. HOMO is the highest-energy molecular orbital that electrons occupy.
2. LumoA is the Lowest Unoccupied Molecular Orbital (LUMO) of a molecule A. LUMO is the lowest-energy molecular orbital that doesn't have any electrons in it.
3. GapA is the energy difference between HOMO and LUMO of molecule A.
4. HomoAH is the HOMO of molecule AH, which is molecule A bounded by one hydrogen atom.
5. LumoAH is the LUMO of molecule AH.
6. GapAH is the energy difference between HomoAH and LumoAH.
7. HomoAH2 is the HOMO of molecule AH2, which is molecule A bounded by two hydrogen atoms.
8. LumoAH2 is the LUMO of molecule AH2.
9. GapAH2 is the energy difference between HomoAH2 and LumoAH2.

Bibliography

[1] K. Butler, D. Davies, H. Cartwright, O. Isayev, and A. Walsh, "Machine learning for molecular and materials science," *Nature*, vol. 559, Jul 2018.

[2] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. Von Lilienfeld, K.-R. Muller, and A. Tkatchenko, "Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space," *The journal of physical chemistry letters*, vol. 6, no. 12, pp. 2326–2331, 2015.

[3] T. Xie and J. C. Grossman, "Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties," *Physical review letters*, vol. 120, no. 14, p. 145301, 2018.

[4] F. A. Faber, A. Lindmaa, O. A. Von Lilienfeld, and R. Armiento, "Machine learning energies of two million elpasolite (a b c 2 d 6) crystals," *Physical review letters*, vol. 117, no. 13, p. 135502, 2016.

[5] A. Ziletti, D. Kumar, M. Scheffler, and L. M. Ghiringhelli, "Insightful classification of crystal structures using deep learning," *Nature communications*, vol. 9, no. 1, pp. 1–10, 2018.

[6] K. Ryan, J. Lengyel, and M. Shatruk, "Crystal structure prediction via deep learning," *Journal of the American Chemical Society*, vol. 140, no. 32, pp. 10158–10168, 2018.

[7] J. Behler, "Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations," *Physical Chemistry Chemical Physics*, vol. 13, no. 40, pp. 17930–17955, 2011.

[8] L. Himanen, A. Geurts, A. S. Foster, and P. Rinke, "Data-driven materials science: Status, challenges, and perspectives," *Advanced Science*, vol. 6, p. 1900808, Sep 2019.

[9] J. Schmidt, M. R. Marques, S. Botti, and M. A. Marques, "Recent advances and applications of machine learning in solid-state materials science," *npj Computational Materials*, vol. 5, no. 1, pp. 1–36, 2019.

[10] Y. Liu, T. Zhao, W. Ju, and S. Shi, "Materials discovery and design using machine learning," *Journal of Materiomics*, vol. 3, no. 3, pp. 159–177, 2017.

[11] W. D. Callister and D. G. Rethwisch, *Materials science and engineering: an introduction*, vol. 9. Wiley New York, 2018.

[12] D. R. Askeland, P. P. Phulé, W. J. Wright, and D. Bhattacharya, *The science and engineering of materials*. Springer, 2003.

[13] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," *Phys. Rev.*, vol. 136, pp. B864–B871, Nov 1964.

[14] W. Koch and M. C. Holthausen, *A chemist's guide to density functional theory*. John Wiley & Sons, 2015.

[15] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. Von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," *Physical review letters*, vol. 108, no. 5, p. 058301, 2012.

[16] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," *Physical review*, vol. 140, no. 4A, p. A1133, 1965.

[17] R. M. Martin, *Electronic structure: basic theory and practical methods*. Cambridge university press, 2020.

[18] J. A. Keith, V. Vassilev-Galindo, B. Cheng, S. Chmiela, M. Gastegger, K.-R. Müller, and A. Tkatchenko, "Combining machine learning and computational chemistry for predictive insights into chemical systems," *Chemical reviews*, vol. 121, no. 16, pp. 9816–9872, 2021.

[19] R. G. Parr and Y. Weitao, "Density functional theory of atoms and molecules," in *Horizons of quantum chemistry*, pp. 5–15, Springer, 1980.

[20] K. Burke, "Perspective on density functional theory," *The Journal of chemical physics*, vol. 136, no. 15, p. 150901, 2012.

[21] M. Giantomassi, M. Stankovski, R. Shaltaf, M. Grüning, F. Bruneval, P. Rinke, and G.-M. Rignanese, "Electronic properties of interfaces and defects from many-body perturbation theory: Recent developments and applications," *physica status solidi (b)*, vol. 248, no. 2, pp. 275–289, 2011.

[22] E. Alpaydin, *Introduction to machine learning*. MIT press, 2014.

[23] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[24] A. Jung, *Machine Learning: The Basics*. Springer Nature, 2021.

[25] T. Mueller, A. G. Kusne, and R. Ramprasad, "Machine learning in materials science: Recent progress and emerging applications," *Reviews in computational chemistry*, vol. 29, pp. 186–273, 2016.

[26] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2021.

[27] I. T. Jolliffe, *Principal component analysis for special types of data*. Springer, 2002.

[28] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4. Springer, 2006.

[29] J. Behler, "Perspective: Machine learning potentials for atomistic simulations," *The Journal of chemical physics*, vol. 145, no. 17, p. 170901, 2016.

[30] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, "Schnet–a deep learning architecture for molecules and materials," *The Journal of Chemical Physics*, vol. 148, no. 24, p. 241722, 2018.

[31] J. Shawe-Taylor, N. Cristianini, *et al.*, *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[32] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[33] Scikit-learn, "Decision Trees." `https://scikit-learn.org/stable/modules/tree.html#tree`.

[34] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*.

[35] L. S. Shapley, *17. A Value for n-Person Games*, pp. 307–318. Princeton University Press, 2016.

[36] R. Kronberg, H. Lappalainen, and K. Laasonen, "Hydrogen adsorption on defective nitrogen-doped carbon nanotubes explained via machine learning augmented dft calculations and game-theoretic feature attributions," *The Journal of Physical Chemistry C*, vol. 125, no. 29, pp. 15918–15933, 2021.

[37] D. Kumar and V. Menkovski, "Understanding anatomy classification through visualization," *CoRR*, vol. abs/1611.06284, 2016.

[38] "Materials Genome Initiative." `https://www.mgi.gov/`, Mar. 2022.

[39] Y. Zhang and C. Ling, "A strategy to apply machine learning to small datasets in materials science," *Npj Computational Materials*, vol. 4, no. 1, pp. 1–8, 2018.

[40] "Open Quantum Materials Database." `https://oqmd.org/`, Apr. 2022.

[41] "Novel Materials Discovery (NOMAD) Laboratory." `https://nomad-coe.eu/`, 2022.

[42] "Computational Materials Repository." `https://cmr.fysik.dtu.dk/`, 2022.

[43] C. E. Calderon, J. J. Plata, C. Toher, C. Oses, O. Levy, M. Fornari, A. Natan, M. J. Mehl, G. Hart, M. B. Nardelli, *et al.*, "The aflow standard for high-throughput materials science calculations," *Computational Materials Science*, vol. 108, pp. 233–238, 2015.

[44] J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke, "Finding density functionals with machine learning," *Physical review letters*, vol. 108, no. 25, p. 253002, 2012.

[45] Y. Zhuo, A. Mansouri 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.

[46] G. B. Olson, "Designing a new material world," *Science*, vol. 288, no. 5468, pp. 993–998, 2000.

[47] S. M. Woodley and R. Catlow, "Crystal structure prediction from first principles," *Nature materials*, vol. 7, no. 12, pp. 937–946, 2008.

[48] C. C. Fischer, K. J. Tibbetts, D. Morgan, and G. Ceder, "Predicting crystal structure by merging data mining with quantum mechanics," *Nature materials*, vol. 5, no. 8, pp. 641–646, 2006.

[49] P. Villars, M. Berndt, K. Brandenburg, K. Cenzual, J. Daams, F. Hulliger, T. Massalski, H. Okamoto, K. Osaki, A. Prince, *et al.*, "The pauling file," *Journal of Alloys and Compounds*, vol. 367, no. 1-2, pp. 293–297, 2004.

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

[51] "HOMO and LUMO." `https://en.wikipedia.org/wiki/HOMO_and_LUMO`, Sep. 2021.

[52] Scikit-learn, "Kernel Ridge Regression." `https://scikit-learn.org/stable/modules/kernel_ridge.html`.

[53] Pandas development team, "pandas-dev/pandas: Pandas," Feb. 2020.

[54] Scikit-learn, "GridSearchCV." `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`, 2021.

[55] Scikit-learn, "Random Forest Regressor." `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor`, 2021.

[56] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *arXiv preprint arXiv:2004.05439*, 2020.

[57] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.