

Article

Integrating NSGA-II and TOPSIS for Stacking Model Optimization in Pursuit of Halide Double Perovskite Screening

Guiqin Liang^{1,2,*}  and Jian Zhang^{3,4,*}

¹ College of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China

² College of Physics and Electronic Information Engineering, Guilin University of Technology, Guilin 541004, China

³ College of Materials Science and Engineering, Guilin University of Electronic Technology, Guilin 541004, China

⁴ Guangxi Key Laboratory of Information Materials, Guangxi Collaborative Innovation Center of Structure and Property for New Energy and Materials, Guilin 541004, China

* Correspondence: 2014054@glut.edu.cn (G.L.); jianzhang@guet.edu.cn (J.Z.)

Abstract

Halide double perovskite materials have been used for various applications; their bandgap (E_g) and heat of formation (ΔH_f) are their key properties. They can be obtained through calculations based on high-throughput density functional theory (DFT), but such calculations are computationally expensive and time-consuming. Machine learning (ML) has proved to be an effective tool for screening potential materials. The prediction accuracy of ML models strongly depends on both input features and ML algorithms. However, there is no unified feature set with which ML models can effectively distinguish halide double perovskite materials. Although it has been proven that stacking ML models can achieve higher prediction accuracy than individual ML models, little attention has been paid to the optimization of stacking models. To solve these problems, we constructed a new feature set obtained from periodic tables for predicting the E_g and ΔH_f of halide double perovskites, and we further proposed a method integrating the nondominated sorting genetic algorithm (NSGA-II) and the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) decision-making tool for stacking model optimization to predict the E_g and ΔH_f of 540 compounds of halide double perovskites. Experimental results from 40 runs of 5-fold cross-validation demonstrate that our proposed new feature set enables ML models to achieve better performance than the original feature set. Moreover, the stacking model optimized by our proposed method yields better predicting performance than that of any individual single model and stacking regression models without optimization, with average improvements of 5.02%, 2.70%, 3.72% and 0.28% in MSE, RMSE, MAE and R^2 , respectively, in E_g prediction, thus providing more effective guidance for screening potential compounds for solar cells from a large quantity of materials.

Keywords: stacking model optimization; TOPSIS; NSGA-II; SHAP; halide double perovskite



Academic Editor: Tomasz Strek

Received: 9 April 2026

Revised: 23 April 2026

Accepted: 9 May 2026

Published: 12 May 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Recently, machine learning (ML) has enabled significant progress in material design and discovery, including hybrid organic–inorganic perovskites [1,2] and inorganic double perovskites [3,4]. Halide double perovskite materials have been applied to various fields, including solar cells, light-emitting diodes, catalysts, batteries, and photodetectors, which

are mainly categorized by their bandgap (E_g) [5,6]. Recent experimental efforts have further advanced perovskite photovoltaic performance through optimized fabrication and device architecture [7,8]. For instance, Barar et al. employed a two-diode model combined with the Lambert W function to extract the key performance parameters of perovskite solar cells [9]. In addition, the stability of compounds can also be taken into consideration and is indicated by the heat of formation (ΔH_f) [10].

Traditionally, E_g and ΔH_f are usually obtained from calculations based on DFT in a traditional way, which is computationally expensive and time-consuming [11,12], making it impossible to apply to a large database. To solve this problem, an ML technique driven by data has been employed to extract potential materials from databases, accelerating the process of material design [5,13–15]. The identification of an accurate input feature set strongly correlates with the performance of ML models, and different feature sets have an influence on the accuracy of ML models to some extent [16–18]. Furthermore, the feature sets mentioned in previous works were constructed randomly by researchers.

Agiorghousis et al. [16] selected only three features from the periodic table—ionization potential (IP), Pauling electronegativity (EN), and atomic radius (AR)—as input features for bandgap prediction. However, the average bandgap errors were as high as 0.457 eV for the training set and 0.514 eV for the testing set based on Random Forest regression. Such prediction accuracy is insufficient for reliably identifying suitable materials based on bandgap values. Although Im et al. [10] predicted the E_g and ΔH_f of halide double perovskite with 32 input features, achieving an average RMSE as low as 0.221 eV for E_g and 0.023 eV/atom for ΔH_f , input features (e.g., distance between cations at A-, B^+ -, and B^{3+} -site, and anions at the X-site) with top feature importance scores need to be obtained from DFT modeling, which requires significant computational cost and enormous computation time [19], making it difficult to apply to screening a large amount of new materials. The desire to easily apply ML models for property prediction based on a new database with high performance accuracy calls for a more proper and convenient input feature set.

Additionally, the selection of an appropriate ML algorithm also has a significant impact on predictive performance. Most previous studies [10,16,20–24] utilized single regression models to predict the properties of perovskite materials, and the prediction accuracy needs to be further improved. Ensemble learning is a common approach for improving overall prediction performance, and it can be implemented by integrating multiple base learners in a certain way to achieve higher prediction accuracy and effectively reduce overfitting risks associated with individual models [25]. Chen et al. [26] proposed an integrated model called R-X-S by combining ridge regression (RR), eXtreme Gradient Boosting (XGBR), and support vector regression (SVR) to predict the melting point of low-melting-point alloys with a low root mean squared error (RMSE) and high correlation coefficient (R) calculated from the simple average results of three models, and it achieved better performance than individual models. Lu et al. [27] developed a weighted voting regressor model to predict the bandgap of hybrid organic–inorganic perovskites (HOIPs) with lower RMSE, which included four sub-models: CATBoost, XGBoost, LightGBM and Gradient Boosting (GBT).

Moreover, ensemble ML models with stacking methods can achieve better prediction results than other ensemble methods (viz. voting, boosting and bagging) and have been widely used in various applications due to their excellent performance [28–30]. It has been found that stacking ensemble learning algorithms can achieve better prediction performance than individual ML models. Meharie et al. [31] proposed a stacking ensemble model with a combination of three models, including linear regression (LR), SVR, and artificial neural networks (ANNs) as base models and GBT as a meta-regressor for predicting the final project cost. Comparison results revealed that the stacking ensemble model outperforms single models. Asante-Okyere et al. [32] developed a stacking model

composed of multivariate adaptive regression spline (MARS), GBT, and a Random Forest Regressor (RFR) as base learners and MARS as a meta-learner for total organic carbon (TOC) prediction; they were able to generate more accurate and reliable TOC predictions than other single models. Wang et al. [33] also proposed a stacking machine learning model with ten individual models as its base models to predict E_g for 3896 inorganic compounds in E-AFLOW, achieving the best performance among other individual models evaluated by RMSE, MAE, MAPE, and R^2 with 5-fold cross-validation, which demonstrates the excellent performance of the stacking approach to E_g prediction.

Chen et al. [34] applied six individual ML models for estimating the changes in mechanical properties in coals under CO_2 , including ANN, SVR, K-nearest neighbors (KNN), RFR and GBT, and then compared their performance with the ensemble stacking model with all six individual ML models as base models and linear regression as its meta-learner. The experimental result proved stacking model can perform better than other ML models. Deng et al. [35] developed a physics-informed machine learning (PIML) framework to model the creep-fatigue interaction behavior of a Ni-based superalloy, which also proves stacking ensemble learning algorithm can perform better, and they clearly pointed out that the assessment and selection of potential base models are critical for the proposed stacking model. However, all of these reported works only indicate that the performances of stacking models can be superior to those of single models used as the base models of the stacking models. None of them have discussed the performances of different stacking models with different combinations of base models and meta-learners, as well as the optimization methods for obtaining the most optimal stacking model. And few studies have been conducted stacking models with optimal methods for predicting properties of halide double perovskite, which has play import role in various fields.

It is meaningful to find out an effective optimization method to obtain the optimal stacking model, which simultaneously considers three indicators during the optimization process for the stacking model, including RMSE, MAE and R^2 . Nondominated sorting genetic algorithm (NSGA-II), proposed by Kalyanmoy Deb et al. [36], has become one of the most famous and widely used multi-objective evolutionary algorithms (MOEAs). Lee et al. [37] propose a computational strategy for perovskite discovery with few computing resources, considering both band gap and effective mass. Etghani et al. [38] employed NSGA-II for optimizing the process of perovskite solar cells with optimum conditions with multiple parameters to achieve PVSCs with high efficiency. However, the optimization results of the NSGA-II algorithm are presented as Pareto-optimal solution sets, which still need further sorting and optimization. TOPSIS is an ideal solution approximation ranking method based on distance, combined with the entropy weight method to achieve objective weight allocation, which has been widely applied in various fields [39–41], including the selection of ML models with optimal performance [42,43].

In this study, we proposed a method integrating TOPSIS with NSGA-II (TOPSIS-NSGA-II) to obtain the best combination of base models and meta-learner for constructing the optimal stacking regression model, which simultaneously considered four regression performance metrics (MSE, RMSE, MAE and R^2) and the number of base models in the stacking models (N_{base}) during the optimization process, achieving more efficient and reasonable decisions in multi-objective optimization for stacking model optimization. And the proposed method was utilized to predict the E_g and ΔH_f for 540 compounds of lead-free double halide perovskite with a feature set obtained from high-throughput calculations based on DFT in previous work [10]. The main contributions of our study are as follows:

- (1) Construction of a new feature set easily obtained from the periodic table as input for ML models and utilization of Shapley Additive exPlanations (SHAP) in feature

selection engineering for predicting E_g and ΔH_f of halide double perovskites were implemented.

- (2) A method integrated NSGA-II and TOPSIS for stacking regression model optimization, simultaneously considering four regression metrics (MSE, MAE, RMSE and R^2) and the number of base models constructed in the stacking models in the Pareto front was proposed.
- (3) The optimal stacking regression model with high predicting accuracy was validated by a new dataset, providing guidance for discovering potential compounds for solar cells from a large quantity of materials.

The remainder of this paper is organized as follows. Section 2 presents the methodology, including feature selection using SHAP (Section 2.1), model evaluation metrics (Section 2.2), stacking model optimization using NSGA-II (Section 2.3), and optimal model selection using TOPSIS (Section 2.4). Section 3 describes the experimental setup, covering the dataset (Section 3.1), input features and feature selection (Section 3.2), and the configuration of stacking ensemble regression models (Section 3.3). Section 4 provides the results and discussion, including performance comparisons between single and stacking models (Section 4.1), stacking models with different base model combinations (Section 4.2), stacking model optimization using NSGA-II and TOPSIS (Section 4.3), and model validation on a completely new test dataset (Section 4.4). Finally, Section 5 concludes the study with a summary of key findings and its limitations. And the optimal stacking model selected from the Pareto solutions by TOPSIS was validated on a new test dataset. Finally, a brief conclusion is given. The overall workflow of the proposed method is shown in Figure 1.

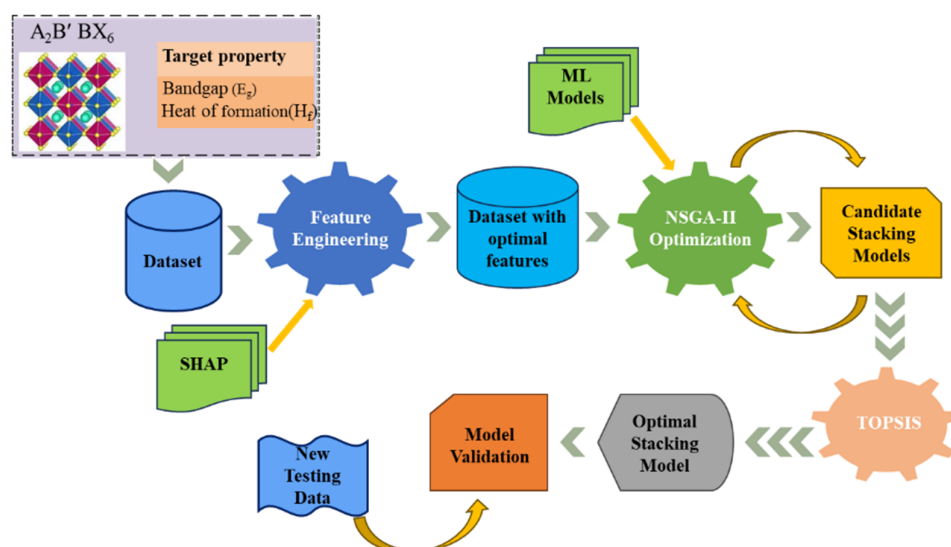


Figure 1. The overall workflow of the proposed method.

2. Methodology

2.1. Feature Selection with Shapley Additive exPlanations (SHAP)

SHAP proposed by Lundberg et al. [44], has been a popular tool used to interpret ML models with Shapley values in material science [45–47], including perovskites [48–50]. Unlike traditional feature selection methods [51–53], the SHAP method can deal with two strongly correlated features [10], and feature importance evaluated by SHAP value can satisfy consistency for each prediction process [4]. And the SHAP value can be used as a reliable property for feature attribution value comparison, making comparisons meaningful [54].

To identify the most predictive features for halide double perovskite properties, we applied the SHAP method following the approach established in our prior work [49]. As

described in [49], SHAP is an additive feature attribution approach that decomposes the output of a ML model into the sum of contributions from each input feature, and it is approximated by a linear explanation model g formulated in Equation (1):

$$g(z') = \varphi_0 + \sum_{i=1}^M \varphi_i z'_i \quad (1)$$

where $z' \in \{0, 1\}^M$, indicates whether a feature is included ($z'_i = 1$) or excluded ($z'_i = 0$) from the model, M is the total number of input features, φ_0 represents the expected value when all inputs are missing, and φ_i is the contribution value of a given feature i to the model presented in Equation (2).

$$\varphi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)] \quad (2)$$

where F is the set of all features, x_S represents the values of the input features in the set S . f is the original prediction model to be explained.

In this study, SHAP values were computed based on the optimal stacking models for E_g and ΔH_f prediction. Features (shown in Figure 2a) were ranked based on their feature importance values measured by their SHAP values, and the selected features were retained for subsequent experiments. The global feature importance rankings and local SHAP are visualized in Figure 3a–d.

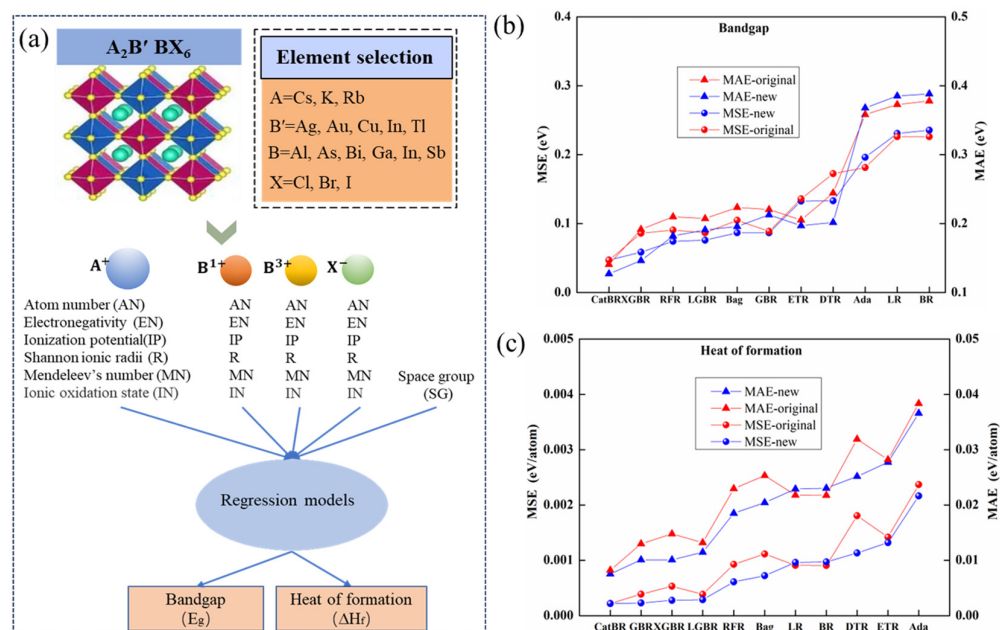


Figure 2. (a) The proposed new feature set for predicting bandgap and heat of formation. (b) Performance comparison between the original feature set in Ref. [10] and the proposed new feature set was measured by MAE and MSE from the averaged values of the testing set under 40 runs of five-fold cross-validation based on different regression models for predicting (b) bandgap and (c) heat of formation, respectively.

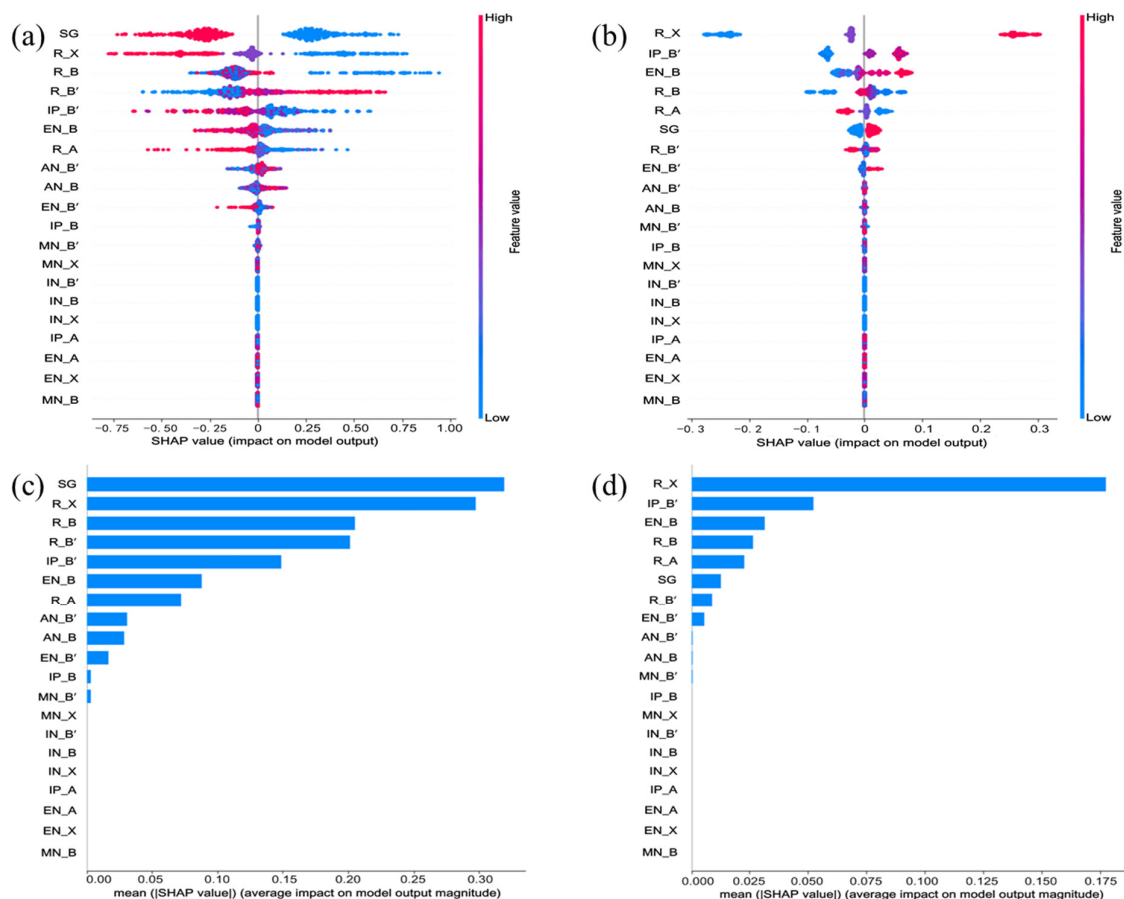


Figure 3. The summarized Sharpley values for the top 20 most important features presented with (a,b) global feature importance and (c,d) local explanation summary based on the XGBR model for bandgap (a,c) and heat of formation (b,d), respectively.

2.2. Model Evaluation

The performance of the regression models was evaluated using four metrics: mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2). These metrics are defined in Equations (3)–(6), respectively.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n \left(\hat{y}_i - y_i \right)^2 \quad (3)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\hat{y}_i - y_i \right)^2} \quad (4)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (5)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n |y_i - \hat{y}_i|^2}{\sum_{i=1}^n |y_i - \bar{y}|^2} \quad (6)$$

where n is the number of samples, $\bar{y}$ is the averaged value of actual values, $\hat{y}_i$ and y_i represent the predicted value and the actual value of the i_{th} sample, respectively. The regression performance of the models was evaluated using MSE, RMSE, MAE, and R^2 . These metrics served as the optimization objectives for the NSGA-II algorithm to identify the optimal meta-learner and combination of base models for constructing an ensemble stacking model with high performance. NSGA-II generates a Pareto front of non-dominated solu-

tions, from which the final solution was selected using the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS).

2.3. NSGA-II for Stacking Model Optimization with

The optimization objectives of NSGA-II are defined with the averaged values of MSE, RMSE, MAE, and R^2 , which are obtained from the stacking models through 40 iterations of five-fold cross-validation, along with the number of base models (N_{base}) contained in each stacking model. Accordingly, the five optimization objectives of NSGA-II are expressed as:

$$\left. \begin{array}{l} \text{Minimize } f_1 = \overline{MSE} = \frac{1}{40} \sum_{i=1}^{40} MSE_i \\ \text{Minimize } f_2 = \overline{RMSE} = \frac{1}{40} \sum_{i=1}^{40} RMSE_i \\ \text{Minimize } f_3 = \overline{MAE} = \frac{1}{40} \sum_{i=1}^{40} MAE_i \\ \text{Maximize } f_4 = \overline{R^2} = \frac{1}{40} \sum_{i=1}^{40} (R^2)_i \\ \text{Minimize } f_5 = N_{base} \end{array} \right\} \quad (7)$$

The output of NSGA-II is a Pareto front, which contains a set of candidate solutions consisting of different stacking models. And it is difficult to directly determine the optimal solution.

2.4. TOPSIS for Optimal Stacking Model Selection

To effectively select the optimal stacking model for the candidate solutions in the Pareto front generated by NSGA-II, the entropy-weighted TOPSIS method was employed [55]. The decision criteria consisted of N_{base} in the stacking ensemble, along with the average values of MSE, RMSE, MAE, and R^2 . Among these, N_{base} , MSE, RMSE, and MAE were designated as cost criteria (lower is better), while R^2 was designated as a benefit criterion (higher is better). The implementation strictly adhered to the standard TOPSIS procedure detailed in [55], and the model with the maximum relative closeness to the ideal solution was selected as the optimal solution for constructing the optimal stacking ensemble model, which defines the optimal meta-learner and base model combination, thereby constructing the final stacking ensemble model.

Assuming that the Pareto front comprises m candidate stacking models, and NSGA-II is formulated with n optimization objectives. As illustrated in [55], the decision matrix is constructed as follows:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \quad (8)$$

where x_{ij} denotes the value of the i -th model with respect to the j -th objective. And the normalized decision matrix $Z = (z_{ij})_{m \times n}$ can be constructed from matrix $X = (x_{ij})_{m \times n}$ using Equation (9):

$$z_{ij} = \frac{x'_{ij}}{\sqrt{\sum_{i=1}^m (x'_{ij})^2}} \quad (9)$$

The weights of the objectives are calculated with the entropy weight method. Specifically, the weight w_j of the j -th objective is derived from the proportion p_{ij} of the i -th model with respect to the j -th objective and the information entropy e_j of the j -th objective. They can be determined by Equation (10) as follows:

$$p_{ij} = \frac{z_{ij}}{\sum_{i=1}^m z_{ij}}, \quad e_j = -\frac{1}{\ln m} \sum_{i=1}^m p_{ij} \ln(p_{ij}), \quad w_j = \frac{1 - e_j}{\sum_{j=1}^n w_j} \quad (10)$$

Subsequently, the weighted normalized decision matrix $V = (v_{ij})_{m \times n}$ is constructed, where $v_{ij} = w_j \cdot z_{ij}$, as presented below:

$$V = Z \cdot W = \begin{bmatrix} w_1 z_{11} & w_2 z_{12} & \cdots & w_n z_{1n} \\ w_1 z_{21} & w_2 z_{22} & \cdots & w_n z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_1 z_{m1} & w_2 z_{m2} & \vdots & w_n z_{mn} \end{bmatrix} \quad (11)$$

The positive ideal solution V^+ and the negative ideal solution V^- were subsequently determined as the best and worst values across all candidate solutions for each objective, respectively. Specifically, $V^+ = (v_1^+, v_2^+, \dots, v_n^+)$, where $v_j^+ = \max(v_{1j}, v_{2j}, \dots, v_{mj})$. And $V^- = (v_1^-, v_2^-, \dots, v_n^-)$, where $v_j^- = \min(v_{1j}, v_{2j}, \dots, v_{mj})$. The Euclidean distances from each solution to the positive and negative ideal solutions, denoted as D_i^+ and D_i^- , respectively, were then calculated using Equations (12) and (13).

$$D_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2} \quad (12)$$

$$D_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2} \quad (13)$$

The closeness degree of each solution is calculated with D_i^+ and D_i^- expressed in Equation (14):

$$C_i = \frac{D_i^-}{D_i^+ + D_i^-}, C_i \in [0, 1] \quad (14)$$

The solution in the Parato front with higher closeness indicates it has better performance. The candidate stacking models are then ranked in descending order of C_i , and the model with the highest C_i is selected as the optimal stacking model.

3. Experimental Data and Model Construction

3.1. Dataset

The dataset for ML models in this study comprised 540 compounds of Pb-free double halide perovskites in the Supporting Information of earlier work [10], which were generated by high-throughput calculations based on DFT with the crystal structure of $A_2B'BX_6$. Each compound contains 32 features named the original feature set, and some of them are obtained from DFT, making it difficult to apply to a new dataset of halide double perovskites. The heat of formation (ΔH_f) values in this dataset are taken directly from [7] and reported in eV/atom—a common unit in DFT calculations (1 eV/atom = 96.485 kJ/mol). To overcome this difficulty, we constructed a new feature set that can be easily collected from accessible existing accessible tables (e.g., periodic table, Shannon ionic radii table and Mendeleev number table), which are shown in Figure 2a.

3.2. Input Features and Feature Selection

3.2.1. Original Feature Set and New Constructed Feature Set

The accuracy of ML models partly depends on the input features [10,17,18,56,57]. Each perovskite compound in the dataset can be described by both elemental and structural features [19]. For halide double halide perovskites with $A_2B'BX_6$ structure, it is known that input features consisting of Shannon ionic radii, atomic number, Mendeleev number, Pauling electronegativity, and ionization potential of A^+ , B'^+ , B^{3+} , X^- site atoms have excellent performance on predicting thermodynamic stability of $A_2B'BX_6$ double halide perovskites [49]. Since the space group (SG) information of the crystal is significant for

bandgap and heat of formation, we continued to use SG from the dataset provided in the previous work [10], which only considers two crystal space groups, including cubic and orthorhombic. All input features used for prediction, named as the new feature set here, are shown in Figure 2a.

It is well known that the non-linear model generally achieves better performance than the linear model, and the tree-based model can provide more accurate predictions than other traditional models [58]. To compare performance of the newly proposed feature set with that of the original feature set, here we employed nine tree-base ML models for predicting bandgap (E_g) and heat of information (ΔH_f) based on original feature set and new proposed feature set, which includes Cat Boosting Regressor (CatBR) [59,60], eXtreme Gradient Boosting Regression (XGBR) [61], Random Forest Regressor (RFR) [62], Bagging Regressor (Bag) [63], Extra-Tree Regressor (ETR) [64], Gradient-Boosting Regressor (GBR) [60], Light Gradient Boosting Machine Regression (LGBR) [65], Decision Tree Regressor (DTR) [66], and AdaBoost (Ada) [67]. Together with another two simpler models, including Linear Regression (LR) and Bayesian Ridge (BR), simpler models have better ability to prevent overfitting [30]. The prediction results for E_g and ΔH_f from eleven individual regression models based on different feature sets can also be presented in Figure 2b,c, respectively.

As shown in Figure 2b,c, most of the regression models with the new feature set can reach better performance for both E_g and ΔH_f , in terms of averaged MAE and averaged MSE on the testing set via five-fold cross-validation repeated 40 times with different random states for dataset splitting. For E_g prediction (Figure 2b), there are notable differences in MSE and MAE between the original feature set and the new feature set. Taking CatBR, the best-performing model, as an example, the MSE and MAE based on the original feature set are 0.0470 eV and 0.1407 eV, respectively, while those based on the new proposed feature set were 0.0468 eV and 0.1272 eV, respectively. For ΔH_f prediction (Figure 2c), although the error gaps between the two feature sets are narrower, the new feature set still yielded improvements: the MAE decreased from 0.008217 eV/atom to 0.007549 eV/atom, and the MSE decreased from 0.000221 eV/atom to 0.000219 eV/atom. These results consistently demonstrate the superiority of the new feature set over the original one. Therefore, all subsequent experiments and discussions in this study are based on the proposed new feature set.

3.2.2. Feature Selection by SHAP

In this study, we employed the SHAP value to evaluate the importance of features in the original feature set and the new feature set, respectively. The top 20 features ranked by SHAP values of the new feature set to predict E_g and ΔH_f are shown in Figure 3.

As shown in Figure 3, the top five features for E_g prediction are space group (SG) of crystal, R_X, R_B, R_B' and IP_B', while those for ΔH_f prediction are R_X, IP_B', EN_B, R_B and R_A. Notably, SG ranks as the most important feature for E_g prediction but drops to sixth place for ΔH_f prediction. In contrast, several features—IN_A, IN_B', IN_B and IN_X exhibit no importance for either target property, as they take constant values across all compounds in the dataset. Furthermore, SG emerges as the most important features for predicting E_g for halide double perovskites. In contrast to predicting ΔH_f , the Shannon ionic radii of halogen anion (R_X) attain the highest importance score, followed by the ionization potential for B³⁺ (IP_B') and electronegativity for B³⁺ (EN_B).

The prediction results, shown in Figure 4, show almost no degradation in performance for either E_g or ΔH_f prediction when using the top 12 features selected by SHAP values compared to using all 25 features. This finding provides strong evidence that SHAP values serve as an effective method for feature selection.

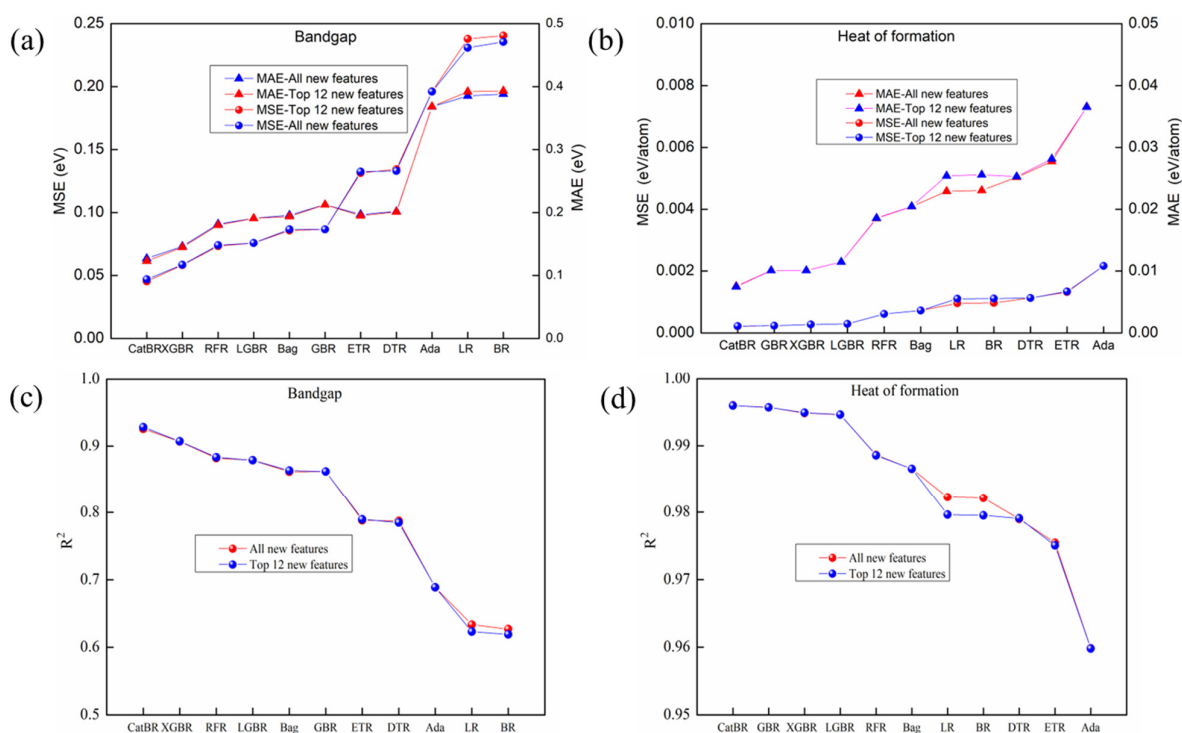


Figure 4. Predicting results of different regression models with the top 12 features selected by SHAP value and all 25 features as input, (a,c) for E_g prediction, (b,d) for ΔH_f prediction.

Furthermore, except for the input features, the abilities of different ML models are different, with a significant influence on the prediction results. That ML model with higher prediction for E_g and ΔH_f can greatly improve the capabilities of screening the most potential stable double halide perovskite from large chemical space for suitable applications. It is necessary to explore more accurate, effective hybrid models to achieve this goal. Since it has been proven that ensemble stacking models can generate better predictive performances, which has been widely used in various applications [28–30,68,69]. Therefore, we further employed an ensemble stacking model for predicting E_g and ΔH_f of halide double perovskites in the following sections.

3.3. Stacking Ensemble Regression Model

Ensemble stacking is an advanced meta-learning algorithm designed to enhance predictive performance by combining multiple heterogeneous regression models, which consists of two layers. In the first layer, a diverse set of base models (e.g., CatBR, XGBR, RFR, LGBR) is trained independently on the original dataset to capture complementary patterns and relationships within the data. The predictions generated by these base models are then used as input features for the meta-learner in the second layer. Except for 11 models mentioned above, including CatBR, XGBR, RFR, LGBR, Bag, GBR, ETR, DTR, Ada, LR, BR, SVR model with the kernel of Radial Basis Function (SVR) usually performs well in the regression case, and it has also been widely used in perovskite material [70–72]. Hence, we also applied SVR in this work. In order to facilitate use and modeling, default parameters are used for both the single model and the combined model [73]. All single regression models used in this study are listed in Table 1. These single models have become candidate models for base models and a meta-learner used to construct a stacking model. The stacking model optimization is used to select the optimal combination of base models and a meta-learner to construct the optimal stacking model for higher prediction accuracy.

The workflow of stacking model optimization with 5-fold cross-validation for predicting E_g and ΔH_f for halide double perovskite is shown in Figure 5.

Table 1. Index for single ML models used as a meta-learner in hybrid stacking models to select the optimal model.

Model ID	Model Name	Note
1	CatBR	Gradient Boosting with categorical feature support
2	XGBR	Extreme Gradient Boosting Regressor
3	RFR	Random Forest Regressor
4	LGBR	Light Gradient Boosting Machine Regression
5	Bag	Bagging Regressor
6	GBR	Gradient Boosting Regressor
7	ETR	Extra-Tree Regressor
8	DTR	Decision Tree Regressor
9	Ada	AdaBoost Regressor
10	LR	Linear regressor
11	BR	Bayesian Ridge
12	SVR	SVR with RBF kernel

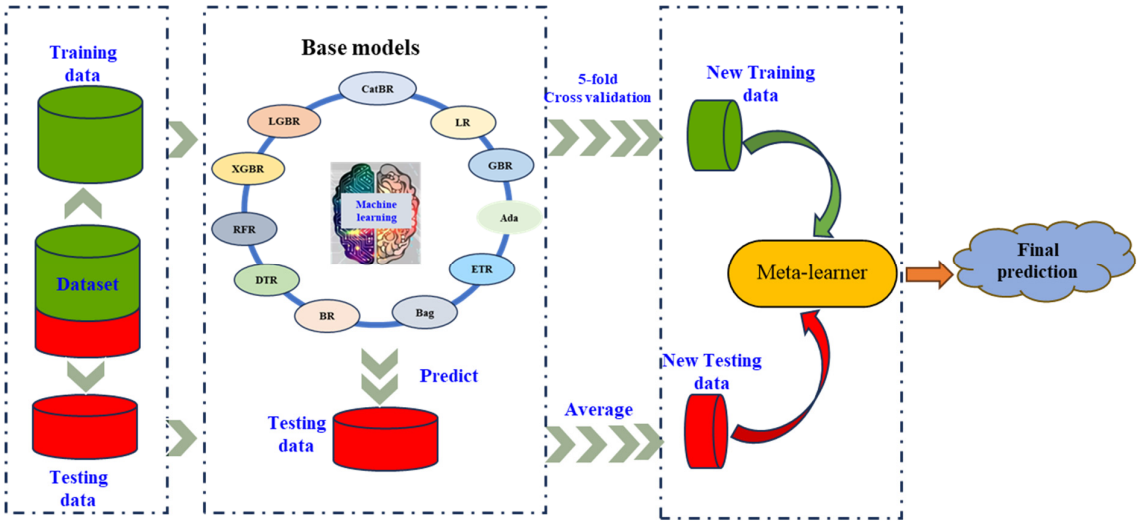


Figure 5. The workflow of the stacking model with 5-fold cross-validation for predicting E_g and ΔH_f for halide double perovskite.

In Figure 5, the dataset with input features optimized by the SHAP value employed in feature engineering was randomly split into two parts, including a training dataset and a testing dataset, both of which consisted of input features (viz. training data and testing data) and outputs (viz. training data label and testing data label). The training data was used to train twelve base models of the stacking model under 5-fold cross-validation to avoid overfitting [30]. The outputs of single models were used to construct new features from new training data as input to train the meta-learner, and the well-trained meta-learner was applied to predict new testing data with new features consisting of the averaged prediction results of the testing data. The final prediction was generated by the trained meta-learner. And performances of different stacking models were measured by four metrics discussed in Equations (3)–(6) in Section 2.2.

4. Results and Discussion

In our study, all single ML models were implemented using the scikit-learn library in Python [74] (version 3.9.5). The test platform was a laptop equipped with Intel (R) Core (TM) i7-1165G7 CPU and 16G RAM. It also needs to be noted that all the results discussed below were measured by the averaged MAE, MSE and R^2 according to 40 runs of 5-fold

cross validation with 80% as training set and the rest of dataset as the testing set based on the 40 different random state, taking top 12 features selected by SHAP value as input, which were showed in Figure 4a for E_g prediction and Figure 4b for ΔH_f prediction.

4.1. Performance Comparisons Between Single Models and Stacking Models

The results of regression models were averaged from 40 runs of 5-fold cross-validation for twelve single models (CatBR, XGBR, RFR, LGBR Bag, GBR, ETR, DTR, Ada, LR, BR and SVR) and twelve stacking models with twelve single models as base models with different meta-learners, which are marked on the horizontal axis shown in Figure 6. CatBR model yields the lowest values of MAE and RMSE among all the single models for both E_g (MAE with 0.1230 eV, RMSE with 0.2091 eV and R^2 with 0.9282) and ΔH_f (MAE with 0.0075 eV/atom, RMSE with 0.0142 eV/atom and R^2 with 0.9957), which is much lower than that reported in [10] calculated by the GBR model with 32 features obtained from DFT, with an averaged RMSE of 0.223 eV for E_g prediction, and an averaged RMSE of 0.021 eV/atom for ΔH_f prediction, achieving an improvement with 6.65% and 47.89%, respectively. However, they are inferior to almost all stacking models except for the stacking model with the LGBR model as the meta-learner, with slightly worse RMSE and R^2 for ΔH_f prediction.

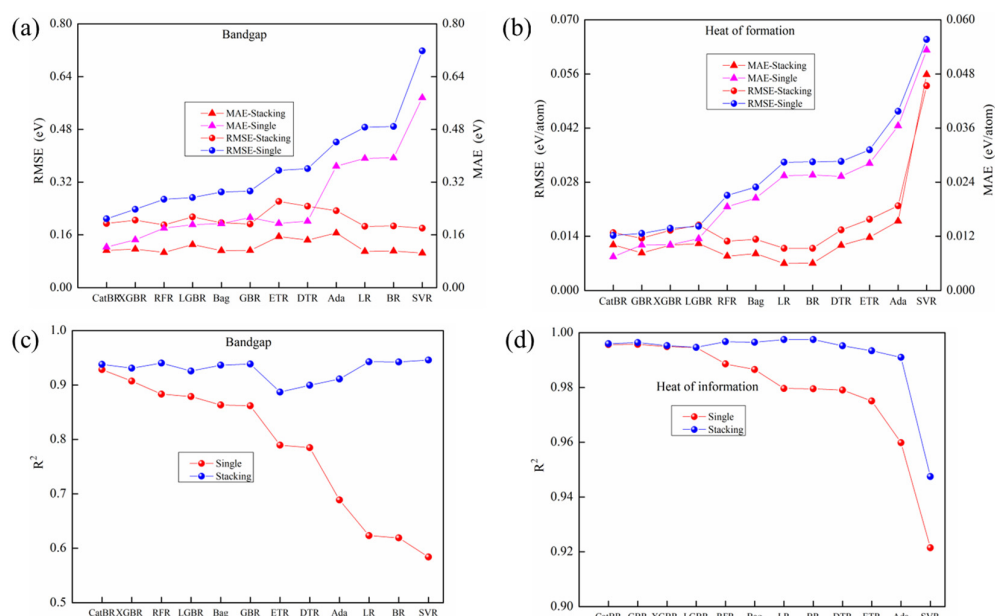


Figure 6. Performance comparison between 11 single models (CatBR, XGBR, RFR, LGBR Bag, GBR, ETR, DTR, Ada, LR and BR) and stacking models with 11 single models as base models with different meta-learners. (a,c) for E_g prediction, (b,d) for ΔH_f prediction.

Furthermore, although the difference between different stacking models lies only in the meta model, there are significant differences in the performance of different stacking models. For E_g prediction, the stacking models with SVR as meta-learner can achieve the best performance in E_g prediction with the smallest MAE (0.1050 eV) and RMSE (0.1803 eV), highest R^2 (0.9459), while the stacking model with the worst performance is the one with ETR as its meta-learner, with the highest MAE (0.1551 eV) and RMSE (0.2614 eV), lowest R^2 (0.8871). Compared to CatBR, the best single model, the improvement percentage in RMSE and MAE of the stacking model with SVR as the meta-learner can reach 13.76% and 14.62%, while the improvement percentage in R^2 is only 1.91%. More details concerning the performance comparison in E_g prediction can be seen in Table 2.

Table 2. Performance comparison between stacking models and single models in E_g prediction.

Stacking Models				Single Models			
Meta-Learner	RMSE (eV)	MAE (eV)	R ²	Model Name	RMSE (eV)	MAE (eV)	R ²
CatBR	0.1944	0.1131	0.9381	CatBR	0.2091	0.1230	0.9282
XGBR	0.2047	0.1173	0.9308	XGBR	0.2375	0.1450	0.9073
RFR	0.1903	0.1070	0.9402	RFR	0.2681	0.1801	0.8833
LGBR	0.2145	0.1307	0.9256	LGBR	0.2730	0.1909	0.8788
Bag	0.1965	0.1124	0.9364	Bag	0.2899	0.1939	0.8632
GBR	0.1932	0.1129	0.9384	GBR	0.2927	0.2126	0.8619
ETR	0.2614	0.1551	0.8871	ETR	0.3557	0.1949	0.7895
DTR	0.2469	0.1442	0.8997	DTR	0.3607	0.2012	0.7848
Ada	0.2334	0.1659	0.9110	Ada	0.4419	0.3684	0.6887
LR	0.1862	0.1102	0.9427	LR	0.4865	0.3920	0.6231
BR	0.1868	0.1108	0.9423	BR	0.4891	0.3933	0.6192
SVR	0.1803	0.1050	0.9459	SVR	0.7182	0.5759	0.5841

Notes: All stacking models consist of the 12 base models detailed in Table 1, with the meta-learner for each model listed in the “Meta-learner” column. The best-performing single model and the best-performing stacking model, along with their respective performance metrics, are highlighted in bold.

For ΔH_f prediction, stacking models with LR as the meta-learner can achieve the best performance with the smallest MAE (0.0060 eV/atom) and RMSE (0.0109 eV/atom) and the highest R² (0.9975). Meanwhile, the worst stacking model is the one with Ada as its meta-learner, with the highest MAE (0.01569 eV/atom) and RMSE (0.02223 eV/atom), lowest R² (0.99074). Compared to the best single model CatBR, the improvement percentage in RMSE and MAE to the stacking model with LR as the meta-learner can reach 23.35% and 19.79%, while the improvement percentage in R² is only 0.17%. More details concerning the performance comparison in ΔH_f prediction can be seen in Table 3.

Table 3. Performance comparison between stacking models and single models in ΔH_f prediction.

Stacking Models				Single Models			
Meta-Learner	RMSE (eV/atom)	MAE (eV/atom)	R ²	Model Name	RMSE (eV/atom)	MAE (eV/atom)	R ²
CatBR	0.01460	0.00985	0.99598	CatBR	0.01422	0.00748	0.99585
GBR	0.01344	0.00830	0.99642	GBR	0.01475	0.01010	0.99572
XGBR	0.01543	0.00973	0.99536	XGBR	0.01608	0.01011	0.99493
LGBR	0.01827	0.01090	0.99370	LGBR	0.01662	0.01148	0.99464
RFR	0.01269	0.00761	0.99675	RFR	0.02460	0.01856	0.98857
Bag	0.01325	0.00807	0.99649	Bag	0.02672	0.02046	0.98654
LR	0.01090	0.00600	0.99750	LR	0.03314	0.02542	0.97962
BR	0.01092	0.00601	0.99749	BR	0.03324	0.02558	0.97950
DTR	0.01694	0.01018	0.99425	DTR	0.03336	0.02529	0.97907
ETR	0.01817	0.01150	0.99348	ETR	0.03639	0.02816	0.97506
Ada	0.02223	0.01569	0.99074	Ada	0.04637	0.03654	0.95982
SVR	0.05297	0.04787	0.94750	SVR	0.06494	0.05328	0.92149

Notes: All stacking models consist of the 12 base models detailed in Table 1, with the meta-learner for each model listed in the “Meta-learner” column. The best-performing single model and the best-performing stacking model, along with their respective performance metrics, are highlighted in bold.

In summary, in both E_g prediction and ΔH_f prediction, the selection of the meta-learner for stacking models has a significant impact on the prediction performances of the stacking model. Although CatBR can achieve the best performance in both E_g and ΔH_f prediction among other single models, when using CatBR as the meta-learner for the stacking model in E_g prediction, its predictive performance is not as good as the stacking models with RFR, BR, LR, and SVR as the meta-models, respectively. This phenomenon also exists in ΔH_f prediction; when CatBR is used as the meta-learner of the stacking model, its predictive performance is not as good as the stacking models with RFR, BR, LR, and Bag as the meta-learners, respectively.

Furthermore, the predictive performance of the best stacking model with 12 single models as its base models is significantly improved compared to the best single model. It is still necessary to explore whether the combination of base models has an impact on predictive performance.

4.2. Stacking Model with Different Combinations of Base Models

It can draw conclusions from Figure 7, which illustrates that the performances of stacking models with the same base model combination but different meta-learners vary significantly. And it clearly points out that the stacking model with twelve single models as base model and SVR as meta-learner can achieve the best performance in E_g prediction, while the stacking model with twelve single models as base model and LR as meta-learner can achieve the best performance in ΔH_f prediction. Furthermore, it is essential to investigate whether stacking models with different combinations of base models exhibit performance differences, providing guidance for subsequent optimization of stacking models.

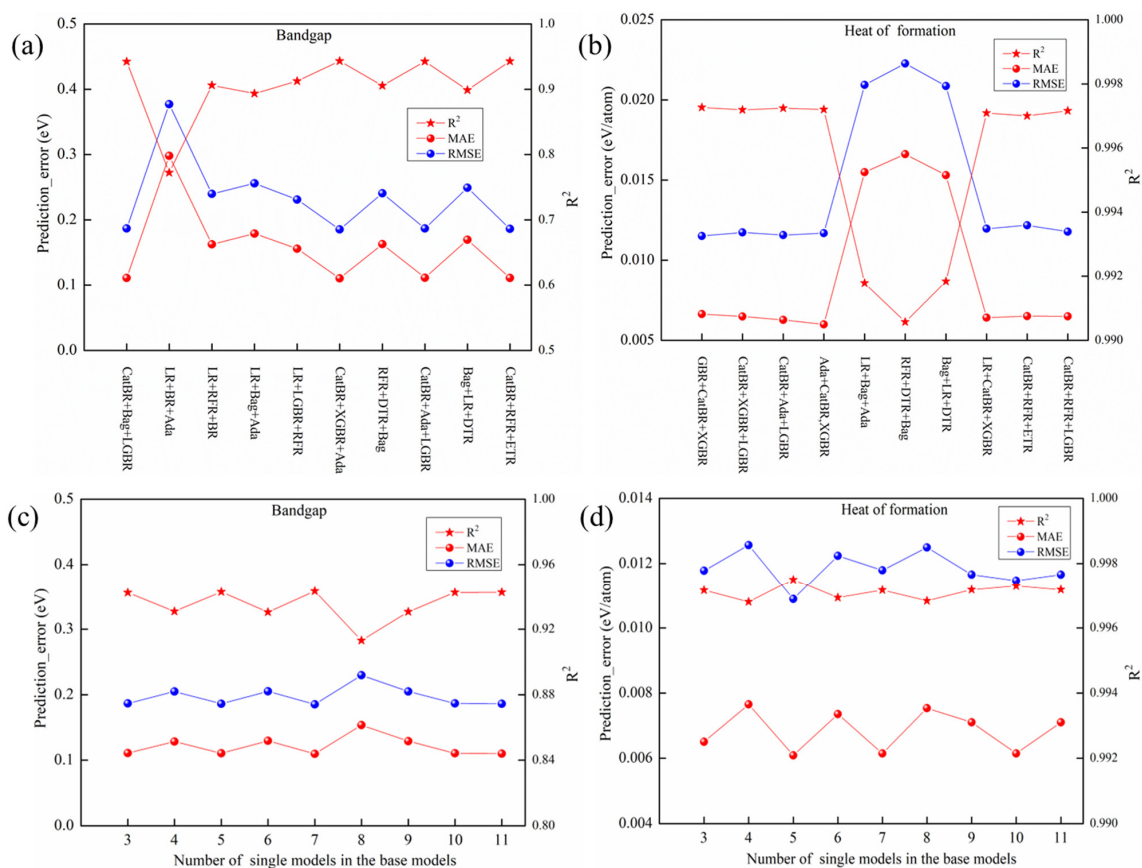


Figure 7. Performances of stacking model with LR as meta-learner and different combinations of base models randomly selected from twelve single models (CatBR, XGBR, RFR, LGBR, Bag, GBR, ETR, DTR, Ada, LR and BR). Base model combinations with three different single models for (a) E_g prediction and (b) ΔH_f prediction. Base model combinations with three to eleven single models for (c) E_g prediction and (d) ΔH_f prediction.

Figure 7a illustrates that the stacking models for E_g prediction with the same number of single models, while different combinations as base models also perform differently, and all of them take SVR as their meta-learner. As the number of single models used to make up the base model of the stacking models increases, the regression performances of the stacking models do not improve accordingly. The performance of the stacking models formed by the same number of base models and meta models varies depending

on the combination of base models. And there are significant differences in the prediction performance of the stacking model with SVR as its meta-learner and the same number of single models, while different combinations as its base models, the value of RMSE can be varied from 0.1788 eV to 0.3694 eV, the value of MAE can be varied from 0.1039 eV to 0.2739 eV, and the value of R^2 can be varied from 0.7807 to 0.9468. The stacking model, which employs only CatBoost, Bagging, and LightGBM (marked as CatBR + Bag + LGBR) as base learners with SVR as meta-learner, achieved slightly superior performance to the full ensemble of twelve base models. Specifically, it reduced the MAE from 0.1050 eV to 0.1039 eV and the RMSE from 0.1803 eV to 0.1788 eV, while increasing R^2 from 0.9459 to 0.9468. Critically, this near-equivalent performance was attained using only three base models instead of twelve. This dramatic reduction in model complexity, coupled with the observed marginal performance gains, underscores the necessity and potential value of systematically optimizing the combination of both base learners and the meta-learner in stacking frameworks.

For ΔH_f prediction (Figure 7b), a contrasting yet insightful trend is observed: a stacking model with only three base models (marked as GBR + CatBR + XGBR) and an LR meta-learner delivers nearly equivalent performance (MAE = 0.00664 eV/atom, RMSE = 0.01151 eV/atom, R^2 = 0.99727) to the all twelve-model ensemble (MAE = 0.00600 eV/atom, RMSE = 0.01090 eV/atom, R^2 = 0.99750), despite a 75% reduction in base models. This demonstrates it needs further research into optimal base-model and meta-model composition strategies.

Furthermore, to systematically investigate the relationship between the number of base learners and the regression performance of the stacking ensemble, we constructed multiple stacking variants by incrementally increasing the count of constituent base models. For the E_g prediction task (Figure 7c), SVR served as the meta-learner, while Linear Regression (LR) was used for the ΔH_f prediction task (Figure 7d). The experimental results clearly demonstrate that predictive performance does not monotonically improve with an increase in the number of base models.

This trend is explicitly illustrated in Figure 7c,d. For the E_g prediction (Figure 7c), as the number of single models used to construct the base models of stacking ensembles increases—through random combinations of the single models listed in Table 1—the predictive performance of the stacking models does not improve correspondingly. Instead, it exhibits irregular fluctuations. Similarly, for the ΔH_f prediction (Figure 7d), increasing the ensemble size leads to fluctuating performance without a clear positive trend. The complete specifications of the base model combinations for each stacking configuration and their associated test set performance (based on 40 runs of 5-fold cross-validation) are provided in Supplementary Tables S1 and S2.

4.3. Stacking Model Optimization with NSGA-II + TOPSIS

Due to the limited combinations of base models considered, it is difficult to identify the optimal stacking model using conventional approaches. Therefore, a more convenient optimization method is required that can incorporate a broad range of combinations of base models and meta-models. In this study, NSGA-II implemented via the DEAP library was employed for stacking model optimization, which simultaneously considers the number of base models (N_{base}) in stacking models and the averaged values of MSE, RMSE, MAE, and R^2 through 40 iterations of five-fold cross-validation, and it yields optimal results in the form of a Pareto front containing multiple non-dominated solutions. Subsequently, TOPSIS was utilized to select the best solution from the Pareto front, and the optimal stacking model was identified based on the closeness degree calculated using Equation (14).

The NSGA-II algorithm is executed with a specific set of parameters to evolve a population of solutions, as shown in Table 4. The population is initialized with 20 individuals, and the algorithm runs for 20 generations to iteratively improve this population. The evolutionary process is guided by a crossover probability of 0.8, and the mutation probability is set to 0.2. The performance of the individual stacking model is evaluated with the averaged MSE, RMSE, MAE and R^2 , which are obtained from 40 runs of 5-fold validation, and they are treated as fitness values during the evaluation process. After the evolutionary process completes, the best-performing individual is selected from the final population, decoded into its constituent base models and meta-model. The NSGA-II algorithm is employed to identify a set of Pareto-optimal solutions, known as the Pareto front, rather than a single optimum.

Table 4. Parameters of NAGA-II for stacking model optimization in E_g and ΔH_f prediction.

Parameter	Value	Description
Population Size	20	Number of candidate stacking model configurations in each generation.
Generations	20	Total iterations of the evolutionary process.
Crossover Probability	0.8	Probability of performing crossover between two parent solutions.
Mutation Probability	0.2	The probability that an individual will undergo mutation.
Number of fitness evaluations	40 runs $\times$ 5-fold CV	Each stacking model evaluated by 40 repeated runs of five-fold cross-validation.
Fitness metrics	MSE, RMSE, MAE, R^2 , N_{base}	Five regression metrics used as multi-objective optimization criteria.

It is well known that the optimization results obtained from the NSGA-II algorithm constitute a set of Pareto-optimal solutions rather than a single optimum. In this study, separate NSGA-II runs were configured with 20 generations to optimize the stacking ensembles for bandgap (E_g) and heat of formation (ΔH_f) prediction, respectively. Each optimization yielded a set of 20 Pareto-optimal solutions, and the consistent convergence behavior observed in the NSGA-II optimizations is illustrated in Figure 8.

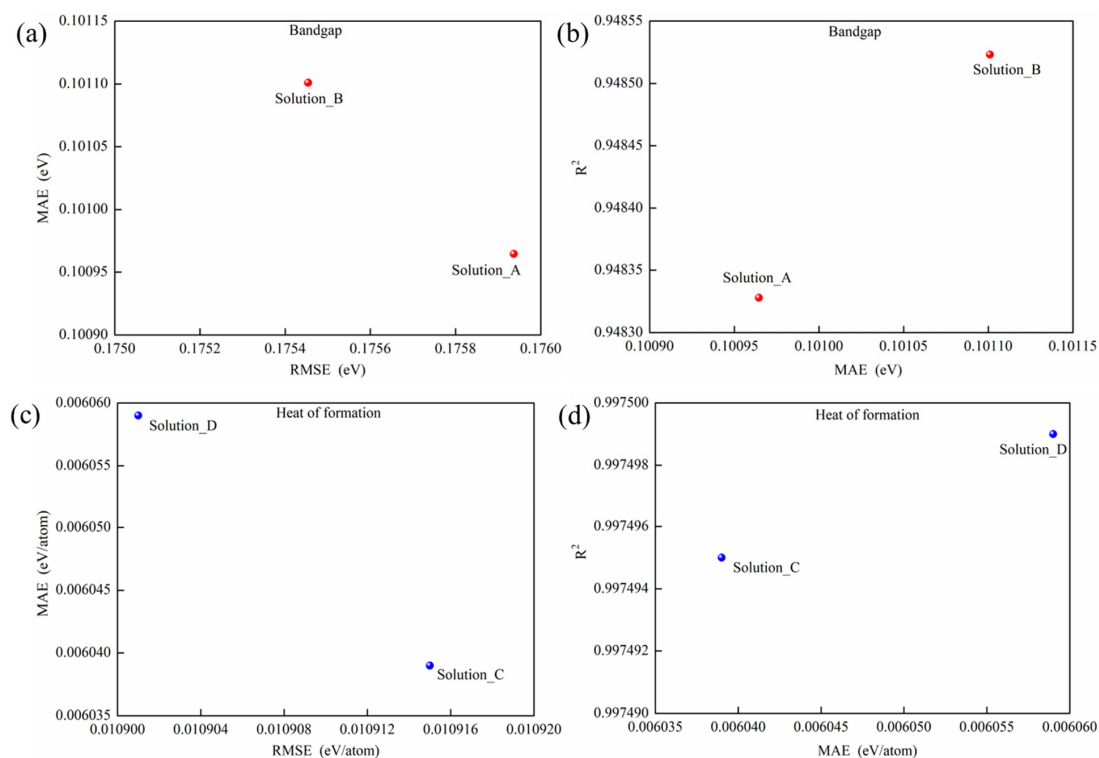


Figure 8. Pareto-optimal solutions from NSGA-II for optimizing the stacking model evaluated with averaged MAE, RMSE and R^2 over 40 runs of 5-fold cross-validation. (a) MAE vs. RMSE and (b) R^2 vs. MAE for predicting E_g ; (c) MAE vs. RMSE and (d) R^2 vs. MAE for predicting ΔH_f .

As shown in Figure 8, the Pareto solutions for both E_g and ΔH_f prediction are heavily concentrated on two stacking models. For E_g prediction, the averaged RMSE of Solution_B is lower than that of Solution_A, whereas the MAE of Solution_A is lower than that of Solution_B (Figure 8a), and the R^2 of Solution_B is higher than that of Solution_A (Figure 8b). For ΔH_f prediction, the averaged RMSE of Solution_D is lower than that of Solution_C, whereas the MAE of Solution_C is lower than that of Solution_D (Figure 8c), and the R^2 of Solution_D is higher than that of Solution_C (Figure 8d). More detailed information on the Pareto solutions is presented in Table 5.

Table 5. Performance of Pareto solutions and the optimal stacking model selected with TOPSIS. (a) Band gap (E_g) prediction. (b) Heat of formation (ΔH_f) prediction.

(a)									
Pareto Solutions	Base_Model_Size	Base_Model	Meta-Learner	MSE (eV ²)	RMSE (eV)	MAE (eV)	R ²	C_I	Optimal Model
Solution_A	4	CatBR, GBR, ETR, Ada	SVR	0.0324	0.1759	0.1010	0.9483	0.3559	N
Solution_B	4	CatBR, RFR, GBR, ETR	SVR	0.0323	0.1755	0.1011	0.9485	0.6441	Y
(b)									
Pareto Solutions	Base_Model_Size	Base_Model	Meta-Learner	MSE (eV ² /atom ²)	RMSE (eV/atom)	MAE (eV/atom)	R ²	C_I	Optimal Model
Solution_C	8	CatBR, RFR, XGBR, Bag, GBR, ETR, Ada, LGBR	BR	0.000136	0.010915	0.006039	0.997495	0.0447	N
Solution_D	6	CatBR, Bag, XGBR, GBR, Ada, LGBR	BR	0.000136	0.010901	0.006059	0.997499	0.9553	Y

As shown in Figure 8 and Table 5, it is difficult to distinguish which solution is the optimal one. Thereby, TOPSIS is employed for selecting the optimal stacking model from the Pareto solutions by calculating the closeness degree (C_I) of each solution, which simultaneously takes five metrics (averaged MSE, RMSE, MAE, R^2 and the number of base models) into consideration, and considers the number of individual models that comprise the base model of the stacking model. And the weights of them can be calculated with Equation (10), respectively. And the results calculated by TOPSIS can also be seen in Table 5.

As summarized in Table 5a, the optimal stacking model selected by TOPSIS for E_g prediction is the Solution_B with a closeness degree of 0.6441, which consists of four base models—CatBR, RFR, GBR, ETR, and employs SVR with a radial basis function (RBF) kernel as the meta-learner. The optimized model achieved the lowest averaged MSE of 0.0323 eV², RMSE of 0.1755 eV, MAE of 0.1011 eV, and R^2 of 0.9485. Compared to the best unoptimized stacking model (comprising 12 base models), the optimized model yielded improvements of 5.02%, 2.70%, 3.72% and 0.28% in MSE, RMSE, MAE and R^2 , respectively. Notably, the number of base models was substantially reduced from 12 to 4 after optimization. A detailed performance comparison between unoptimized and optimized stacking models for E_g prediction is presented in Table 6.

Table 6. Performance comparison of unoptimized and optimized stacking models for E_g prediction.

Metric	Best Stacking Model (Unoptimized)	Optimal Stacking Model (Optimized)	Improvement
MSE (eV ²)	0.0340	0.0323	↓ 5.02%
RMSE (eV)	0.1803	0.1755	↓ 2.70%
MAE (eV)	0.1050	0.1011	↓ 3.72%
R ²	0.9459	0.9485	↑ 0.28%
Base Model Size	12 base models	4 base models	↓ 66.7%
Meta-learner	SVR	SVR	—

Note: ↑ indicates improvement after optimization; ↓ indicates degradation after optimization; — indicates not applicable. The direction of improvement (↑ or ↓) is defined based on the desirability of each metric (e.g., for R^2 , ↑ is better; for RMSE, MSE, MAE and the Base Model Size, ↓ is better).

In terms of ΔH_f prediction in Table 5b, Solution_D was identified as the optimal stacking model by TOPSIS with a closeness degree of 0.9553. This model consists of six base models—CatBR, XGBR, Bag, GBR, Ada and LGBR and uses BR as the meta-learner. Compared to the

unoptimized model, the optimized model achieved averaged values of 0.000136, 0.010901, 0.006059 and 0.997499 for MSE, RMSE, MAE and R^2 , respectively, exhibiting improvements of 0%, 0.06%, 0.8% and 0.0005% in the respective metrics. Furthermore, the number of base models decreased sharply from 12 to 6. Performance comparison of unoptimized and optimized stacking models for ΔH_f prediction is also summarized in Table 7.

Table 7. Performance comparison of unoptimized and optimized stacking models in ΔH_f prediction.

Metric	Best Stacking Model (Unoptimized)	Optimal Stacking Model (Optimized)	Improvement
MSE (eV ² /atom ²)	0.000136	0.000136	0%
RMSE (eV/atom)	0.010907	0.010901	↓ 0.06%
MAE (eV/atom)	0.006011	0.006059	↓ 0.8%
R^2	0.997494	0.997499	↑ 0.0005%
Base Model Size	12 base models	6 base models	↓ 50%
Meta-learner	LR	BR	—

Note: ↑ indicates improvement after optimization; ↓ indicates degradation after optimization; — indicates not applicable. The direction of improvement (↑ or ↓) is defined based on the desirability of each metric (e.g., for R^2 , ↑ is better; for RMSE, MSE, MAE and the Base Model Size, ↓ is better).

In addition, it is necessary to verify whether an overfitting problem exists in the optimal stacking prediction models determined by learning curves [20]. Figure 9 illustrates the learning curves of the optimal stacking models for E_g prediction (Figure 9a) and for ΔH_f prediction (Figure 9b), depicting the relationship between the number of training samples and the proposed model's performance. With the increase in training samples, both the Train-MSE and Cross-validation MSE show a decreasing trend and converge to their minimum values when the training size is approximate to 80%, and the gap between them narrows to near zero, as shown by the MSE_difference curve, confirming the proposed models' strong generalization ability and high predictive accuracy without overfitting.

To further validate the predictive performance of the proposed stacking models, we compared their predicted values with the actual values for E_g and ΔH_f , respectively. The comparison was based on 108 compounds randomly selected from the dataset, using the top 12 features from the new feature set. For E_g prediction, the optimal stacking model employed CatBR, RFR, GBR, and ETR as base models, with SVR (RBF kernel) as the meta-learner. The predicted results are presented in Figure 9c, which are highly consistent with the actual values. In the ΔH_f prediction, the predicted ΔH_f values from the optimal stacking model show excellent agreement with the actual values, as shown in Figure 9d. This model employed CatBR, XGBR, Bag, GBR, Ada and LGBR as base models and LR as the meta-learner. The small deviations observed across all test samples provide compelling evidence of the model's strong generalization capability and high predictive accuracy.

4.4. Comparison of the Optimal Stacking Model Selected by TOPSIS Versus the Optimal Model Identified by Random Search

To further validate the effectiveness of the NSGA-II optimization combined with TOPSIS selection, we compared the optimal stacking model selected by TOPSIS against the optimal model identified by 400 random searches under the same evaluation budget (each model evaluated with 40 runs of 5-fold cross-validation). Table 8 presents the comparison results for both prediction targets, split into two sub-tables for clarity due to the different units (eV for E_g , eV/atom for ΔH_f).

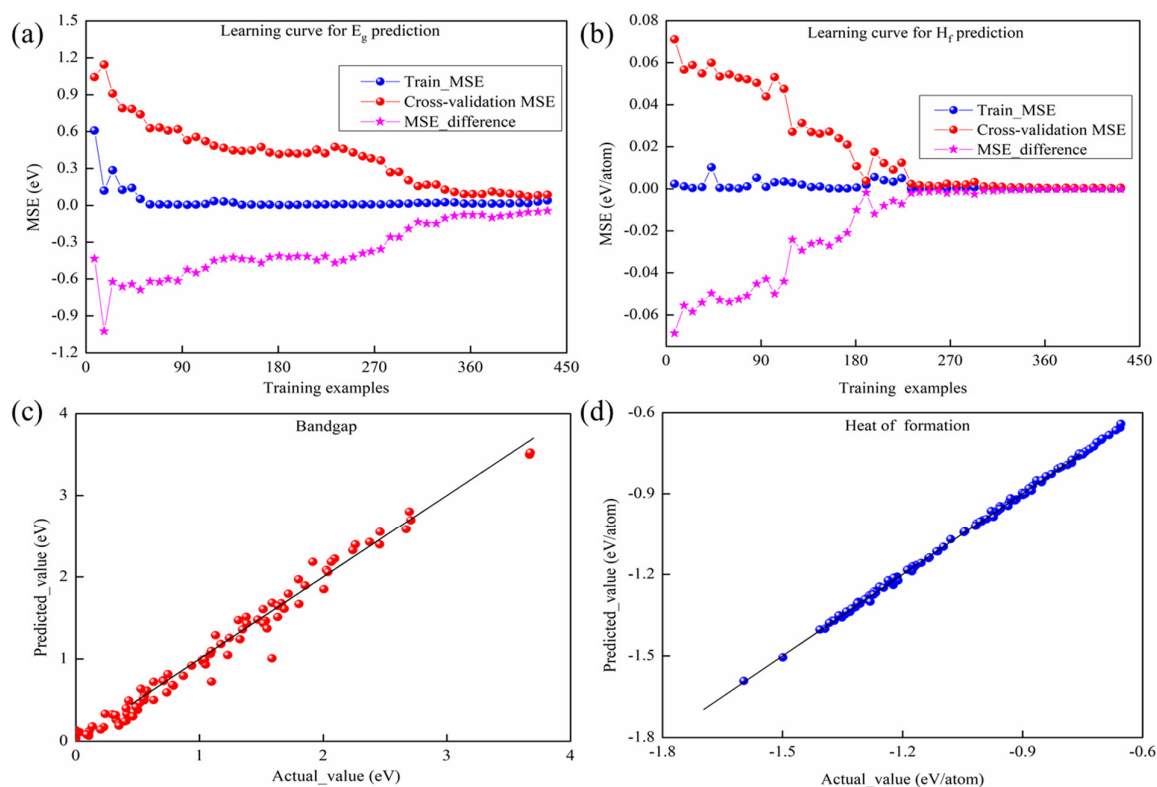


Figure 9. Performances of the optimal stacking models. (a) Learning curves for E_g prediction; (b) Learning curves for ΔH_f prediction; (c) Parity plot of predicted vs. actual E_g values; (d) Parity plot of predicted vs. actual ΔH_f values. In (a,b), Train-MSE, Cross-validation MSE, and their difference (MSE_difference, calculated as Train-MSE minus Cross-validation MSE) are shown. In (c,d), the ideal matchup between predicted and actual values is shown as the 45° dashed line.

Table 8. Performance comparison between the optimal stacking model selected by TOPSIS and the best stacking model identified by 400 random searches. (a) Band gap (E_g) prediction; (b) heat of formation (ΔH_f) prediction.

(a)							
Model	Base_Model_Size	Base_Model	Meta-Learner	MSE (eV ²)	RMSE (eV)	MAE (eV)	R ²
TOPSIS-selected	4	CatBR, RFR, GBR, ETR	SVR	0.0323 (±0.0134)	0.1755 (±0.0375)	0.1011 (±0.0139)	0.9485 (±0.0223)
Optimal from 400 RS	6	CatBR, XGBR, LR, GBR, ETR, Ada	SVR	0.0331 (±0.0135)	0.17834 (±0.0382)	0.1041 (±0.0142)	0.9472 (±0.0225)
Improvement (%)	+33.33%	—	—	+2.47%	+1.59%	+2.91%	+0.14%
(b)							
Model	Base_Model_Size	Base_Model	Meta-Learner	MSE (eV ² /atom ²)	RMSE (eV/atom)	MAE (eV/atom)	R ²
TOPSIS-selected	6	CatBR, Bag, XGBR, GBR, Ada, LGBR	BR	0.000136 (±0.000111)	0.010901 (±0.004134)	0.006059 (±0.000932)	0.997499 (±0.002038)
Optimal from 400 RS	7	CatBR, Bag, GBR, ETR, Ada, LGBM, LR	BR	0.000141 (±0.000113)	0.011119 (±0.004181)	0.006430 (±0.000928)	0.997406 (±0.002081)
Improvement (%)	14.29%	—	—	+3.49%	+1.96%	+5.76%	+0.01%

Note: “Optimal from 400 RS” refers to the stacking model with the lowest RMSE among 400 randomly sampled combinations. Values in parentheses represent standard deviations over 40 independent cross-validation runs. One-tailed paired *t*-tests (TOPSIS-selected vs. Optimal from 400 RS) on RMSE values yield $p = 0.0603$ for E_g and $p = 0.2791$ for ΔH_f , confirming no statistical degradation for either target. Improvement (%) = $(RS - TOPSIS)/RS \times 100\%$ for MSE, RMSE, MAE; for R², improvement = $(TOPSIS - RS)/|RS| \times 100\%$. Conversion: 1 eV/atom = 96.485 kJ/mol.

Table 8a shows the results for E_g prediction. The TOPSIS-selected model achieves an RMSE of 0.1755 eV using only 4 base models (CatBoost, RandomForest, GradientBoosting, and ExtraTree) with SVR as the meta-learner. In contrast, the optimal model identified by random search achieves an RMSE of 0.1783 eV using 6 base models (CatBoost, XGBoost,

GradientBoosting, ExtraTree, AdaBoost, and LinearRegression), also with SVR as the meta-learner. The TOPSIS-selected model yields relative improvements of 1.59% in RMSE, 2.47% in MSE, and 2.91% in MAE, while reducing the base model size by 33.33% (from 6 to 4 models).

Table 8b presents the results for ΔH_f prediction. The TOPSIS-selected model achieves an RMSE of 0.010901 eV/atom using 6 base models (CatBoost, Bagging, XGBoost, GradientBoosting, AdaBoost, and LightGBM) with Bayesian Ridge as the meta-learner. The optimal random search model achieves an RMSE of 0.011119 eV/atom using 7 base models (CatBoost, Bagging, GradientBoosting, ExtraTree, AdaBoost, LightGBM, and LinearRegression) with Bayesian Ridge as the meta-learner. The TOPSIS-selected model demonstrates relative improvements of 1.96% in RMSE, 3.49% in MSE, and 5.76% in MAE, with a 14.29% reduction in base model size (from 7 to 6 models).

To confirm that the proposed NSGA-II+TOPSIS method does not lead to performance degradation, we performed one-tailed paired t-tests comparing its RMSE values against those of the optimal random search model over 40 independent cross-validation runs. For E_g prediction (Table 8a), the test yields a p -value of 0.0603; for ΔH_f prediction (Table 8b), the p -value is 0.2791. Both p -values exceed the conventional significance level of 0.05, statistically confirming no performance degradation for either prediction target. These results demonstrate that NSGA-II with TOPSIS selection effectively identifies more compact stacking ensembles without sacrificing predictive performance.

4.5. Model Validation on Completely New Test Data

The training dataset contains three different A-site cations (viz. Cs, K, and Rb), five B1-site cations (viz. Tl, In, Ag, Au, and Cu) and six B2-site cations (viz. Al, Ga, Bi, In, Sb, and As). It appears on either B1- or B2-sites. The X-site has four different possible choices of F, Cl, Br, and I. And two kinds of the crystal group space were taken into consideration, which consisted of a total of 540 unique $A_2B' BX_6$ compounds in the chemical space. Since all possible compounds in the chemical space are included in the training dataset, we needed to find other data for model validation.

To conform the predicting ability of our proposed hybrid stacking model, we applied it to 24 new compounds in cubic crystal space excluded from the training dataset, six compounds of which contained 3 completely new elements (viz. Na^+ , K^+ , and Rb^+) in B-site never appeared in the training process, and their values used in the comparison for E_g and decomposition enthalpy (ΔH) were calculated by DFT obtained from the work of Zhao et al. [75], which contains 64 compounds, of which 40 already exist in the dataset used for stacking model optimization, and the remaining 24 compounds are used as a new dataset to validate the proposed optimal stacking model. The average predicted results were depicted in Table 9.

There are 2 out of 24 compounds (viz. $Cs_2NaBiBr_6$ with 0.7174 eV, and Cs_2NaBiI_6 with 0.7065 eV) in the classified criteria from Im [10] within the range of [0.3 eV, 0.8 eV] according to the predicted E_g from the proposed model, indicating that they are suitable to be material for solar cells, which is not in line with the results calculated by Zhao [75], where the classified criteria for solar cells are within the range of [0.8 eV, 2 eV]. It needs to be noted that they are all within the averaged MAE of 0.1011 eV obtained by our proposed method.

Table 9. Predicting results of E_g and ΔH_f for 24 new compounds by the proposed hybrid stacking model.

Nos.	Compounds	Predicted E_g (eV)	Classified with Predicted E_g^a	Classified with Calculated E_g^b	Predicted ΔH_f (eV/atom)	Stability Classified with Predicted ΔH_f^a	Stability Classified with Calculated ΔH_f^b
1	Cs2NaSbBr6	1.2198	N	N	−0.7630	Y	Y
2	Cs2KSbBr6	0.9398	N	N	−0.8058	Y	Y
3	Cs2RbSbBr6	1.2994	N	N	−0.8484	Y	Y
4	Cs2NaBiBr6	0.7174	Y	N	−0.8432	Y	Y
5	Cs2KBiBr6	0.9142	N	N	−0.8860	Y	Y
6	Cs2RbBiBr6	1.0474	N	N	−0.9317	Y	Y
7	Cs2NaSbCl6	1.3333	N	N	−1.0036	Y	Y
8	Cs2KSbCl6	1.3671	N	N	−1.0817	Y	Y
9	Cs2RbSbCl6	1.7458	N	N	−1.1066	Y	Y
10	Cs2NaBiCl6	1.1677	N	N	−1.0937	Y	Y
11	Cs2KBiCl6	1.3577	N	N	−1.1813	Y	Y
12	Cs2RbBiCl6	1.9255	N	N	−1.2019	Y	Y
13	Cs2NaSbI6	1.2048	N	N	−0.7630	Y	Y
14	Cs2KSbI6	0.9420	N	N	−0.8058	Y	N
15	Cs2RbSbI6	1.2464	N	N	−0.8484	Y	N
16	Cs2NaBiI6	0.7065	Y	N	−0.8432	Y	Y
17	Cs2KBiI6	0.8835	N	N	−0.8860	Y	N
18	Cs2RbBiI6	0.9172	N	N	−0.9317	Y	N
19	Cs2NaSbF6	1.3905	N	N	−1.1677	Y	Y
20	Cs2NaBiF6	1.3297	N	N	−1.2771	Y	Y
21	Cs2KBiF6	2.0333	N	N	−1.4035	Y	Y
22	Cs2KSbF6	1.9047	N	N	−1.2915	Y	Y
23	Cs2RbSbF6	2.1867	N	N	−1.3060	Y	Y
24	Cs2RbBiF6	2.5132	N	N	−1.4065	Y	Y

Notes: E_g^a and ΔH_f^a denote the classified criteria for solar cells from Ref. [10], E_g^b and ΔH_f^b denote the classified criteria for solar cells from Ref. [75]. Comparisons with DFT-calculated results are made to determine whether the classified results are solar cells (Y) or not (N), based on different criteria for stability and bandgap properties due to different calculation methods in the dataset. The different classified results were highlighted in bold.

In addition, the stability is also another significant property for solar cells, and they can be evaluated to be stable with negative values of ΔH_f [10] or positive decomposition enthalpy [75]. From Table 9, it shows all of them are stable due to their negative values in predicting ΔH_f . 4 out of 24 compounds are classified to be unstable according to the negative decomposition enthalpy (viz. Cs_2KSbI_6 with -3 meV/atom, Cs_2KBiI_6 with -4 meV/atom, $\text{Cs}_2\text{RbSbI}_6$ with -14 meV/atom, and $\text{Cs}_2\text{RbBiI}_6$ with -15 meV/atom), and the calculated values of decomposition enthalpy (ΔH) of Cs_2KSbI_6 and Cs_2KBiI_6 are very close to the stable criteria of positive decomposition enthalpy. These results proved that the proposed model can be an effective method for quickly predicting E_g and ΔH_f for halide double perovskites to identify compounds with elements appearing frequently in the training dataset, obtaining compounds with suitable E_g and ΔH_f for solar cells. However, it may probably not be effective for compounds with elements that are contained in the training dataset, which can lead to a conclusion consistent with that mentioned in [76]. It is worth noting that it took only a few seconds to predict all test compounds with the trained stacking model on the laptop with Intel(R) Core (TM) i7-1165G7 CPU.

The few discrepancies in Table 9 (2 for band gap, 4 for stability) are predominantly observed in compounds containing the new alkali metal ions (Na^+ , K^+ , Rb^+), specifically $\text{Cs}_2\text{NaBiBr}_6$ and $\text{Cs}_2\text{NaBiI}_6$ for band gap and Cs_2KSbI_6 , $\text{Cs}_2\text{RbSbI}_6$, Cs_2KBiI_6 , and $\text{Cs}_2\text{RbBiI}_6$ for stability. These Bi/I-containing systems are particularly sensitive to DFT computational settings. Critically, as noted in Zhao et al. [75], their DFT calculations included spin-orbit coupling (SOC), whereas our training data from Im et al. [10] were computed using the PBE functional without SOC. It is well-established that SOC can reduce the band gaps of Bi/I-containing perovskites by 0.5 – 1.5 eV and alter their formation energies by 0.3 – 0.7 eV/atom. Consequently, the observed discrepancies are more likely attributable to these differences in DFT benchmarks rather than to a fundamental error in our model.

4.6. Generalizability to Other Perovskite Families

To evaluate the generalizability of our proposed feature set and stacking-based classification methodology, we applied them to the expanded double perovskite dataset compiled by Bartel et al. [51], which contains 918 $\text{A}_2\text{B}'\text{BX}_6$ materials including both halide and oxide compositions. This classification task enables a direct benchmark against Bartel's τ descriptor (91% accuracy).

As shown in Table 10, our proposed feature set combined with the optimal stacking configuration (CatBoost Classifier, Random Forest Classifier, Bagging Classifier, AdaBoost Classifier + CatBoost Classifier) identified by our framework—achieves superior performance, with an accuracy of 96.22% ($\pm 1.01\%$), an F1 score of 98.03% ($\pm 0.53\%$), and a precision of 96.89% ($\pm 0.83\%$) under 40 runs of 5-fold cross-validation, surpassing the 91% baseline.

Table 10. Classification performance comparison on the 918 double perovskite dataset.

Method	Accuracy (%)	F1 Score (%)	Precision
Bartel's τ descriptor	91.00	—	—
Our method	96.22 (± 1.01)	98.03 (± 0.53)	96.89 (± 0.83)
Improvement	+5.22%	—	—

Note: Values are reported as mean ($\pm$ standard deviation) over 40 runs of 5-fold cross-validation. Bartel et al. [51] reported only accuracy for their τ descriptor.

These results demonstrate: (1) our feature set generalizes across perovskite families; (2) our feature selection and stacking optimization framework is versatile, effectively handling both regression and classification tasks; and (3) extension to Ruddlesden–Popper phases requires additional layer-specific descriptors (future work).

5. Conclusions

In this study, we aimed to develop a method integrated NSGA-II and TOPSIS for stacking regression model optimization, which simultaneously considered four regression metrics (MSE, MAE, RMSE and R^2) and the number of base models constructed the stacking models as the five optimization targets, and it was utilized for the quantitative determination of E_g and ΔH_f of 540 Pb-free halide double perovskites with new constructed feature set obtained from easily accessible tables. The experimental results have demonstrated that the newly constructed feature set has better performance than the original feature set. It indicated that the SHAP value can be an effective tool for feature selection as well. Furthermore, it also proved that the ensemble stacking model can perform better than any individual ML model in both E_g and ΔH_f prediction with lower MSE and MAE for double halide perovskites with the top 12 features from the new feature set selection by SHAP values. The combination of NSGA-II and TOPSIS has performed well in the ensemble stacking model optimization. The stacking model optimized by our proposed method yields better predicting performance than the stacking regression models without optimization, with improvements in averaged MSE, RMSE, MAE and R^2 with 5.02%, 2.70%, 3.72% and 0.28% in E_g prediction, respectively. And for ΔH_f prediction, as its predictive accuracy was inherently high, despite only a minor performance improvement after optimization, the number of base models for the stacking model dropped sharply from 12 to 6. Finally, the proposed models were validated on completely new test data. The limitations of our study are that the prediction performance of compounds in the testing dataset with elements not contained in the training dataset will be degraded. The training dataset should be extended to contain more elements consisting of compounds in the whole chemistry space to improve the predictive ability. In addition, multi-objective optimization should be introduced to take key properties of halide double perovskite into account simultaneously for screening suitable materials for solar cells more quickly and conveniently. These will be the focus of our research in the future.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/ma19102018/s1>, Table S1: The specific information of the base models of the stacking models and its corresponding performance on the test set with 40 runs of 5-fold cross-validation for E_g prediction. Table S2: The specific information of the base models of the stacking models and its corresponding performance on the test set with 40 runs of 5-fold cross-validation for H_f prediction.

Author Contributions: Conceptualization, G.L. and J.Z.; methodology, G.L.; software, G.L.; validation, G.L. and J.Z.; formal analysis, G.L.; investigation, G.L. and J.Z.; resources, G.L.; data curation, G.L.; writing—original draft preparation, G.L.; writing—review and editing, G.L.; visualization, G.L. and J.Z.; supervision, J.Z.; project administration, J.Z.; funding acquisition, G.L. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The authors thank the financial sponsorship from the National Natural Science Foundation of China [22439001, 22169006], the Guangxi Natural Science Foundation Program [2023AB01273], and the Guangxi Basic Ability Promotion Project for Young and Middle-Aged Teachers [2023KY0266]. The authors are grateful to the Guangxi Collaborative Innovation Center for New Energy and Materials Structure and Properties, Guilin University of Electronic Science and Technology, for its assistance in characterization and measurement.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Shi, Y.; Wen, J.; Wen, C.; Jiang, L.; Wu, B.; Qiu, Y.; Sa, B. Interpretable machine learning insights of power conversion efficiency for hybrid perovskites solar cells. *Sol. Energy* **2025**, *290*, 113373. [\[CrossRef\]](#)
- Cai, Y.; Bai, Z.; Chen, C.; Sun, M.; Wang, Z.; Wang, S.; Zhang, Z.; Xie, J.; Li, D.; Guan, X.; et al. Machine learning prediction of small molecule passivators and their impacts on the passivation and photocatalytic performance of organic-inorganic hybrid perovskite interfaces. *Energy Mater.* **2025**, *5*, 500043. [\[CrossRef\]](#)
- Feng, H.-J.; Wu, K.; Deng, Z.-Y. Predicting Inorganic Photovoltaic Materials with Efficiencies >26% via Structure-Relevant Machine Learning and Density Functional Calculations. *Cell Rep. Phys. Sci.* **2020**, *1*, 100179. [\[CrossRef\]](#)
- Yan, F.; Song, K.; Liu, Y.; Chen, S.; Chen, J. Predictions and mechanism analyses of the fatigue strength of steel based on machine learning. *J. Mater. Sci.* **2020**, *55*, 15334–15349. [\[CrossRef\]](#)
- Maddah, H.A.; Berry, V.; Behura, S.K. Cuboctahedral stability in Titanium halide perovskites via machine learning. *Comput. Mater. Sci.* **2020**, *173*, 109415. [\[CrossRef\]](#)
- Bayon, A.; de la Calle, A.; Ghose, K.K.; Page, A.; McNaughton, R. Experimental, computational and thermodynamic studies in perovskites metal oxides for thermochemical fuel production: A review. *Int. J. Hydrogen Energy* **2020**, *45*, 12653–12679. [\[CrossRef\]](#)
- Li, C.; Chen, Z.; Zhou, C.; Hong, S.; Yang, S. Mixed Antisolvent Strategy Enables Synergistic Crystallization Control and Defect Passivation in Perovskite Solar Cells. *ACS Appl. Mater. Interfaces* **2025**, *17*, 29639–29648. [\[CrossRef\]](#)
- Farias-Basulto, G.; Mehlhop, T.; Otto, N.J.; Bertram, T.; Jäger, K.; Gall, S.; Weinberger, N.; Schlattmann, R.; Lauermann, I.; Klenk, R.; et al. Improving Perovskite/CIGS Tandem Solar Cells for Higher Power Conversion Efficiency through Light Management and Bandgap Engineering. *ACS Appl. Mater. Interfaces* **2025**, *17*, 56250–56255. [\[CrossRef\]](#)
- Bărar, A.; Boscornea, C.; Bălășoiu, M.; Mănănilă-Maximean, D. Two-diode modelling of perovskite solar cells and parameter extraction using the lambert w function. *UPB Sci. Bull. Ser. A Appl. Math. Phys.* **2021**, *83*, 217–225.
- Im, J.; Lee, S.; Ko, T.W.; Kim, H.W.; Hyon, Y.K.; Chang, H. Identifying Pb-free perovskites for solar cells by machine learning. *npj Comput. Mater.* **2019**, *5*, 37. [\[CrossRef\]](#)
- Himanen, L.; Geurts, A.; Foster, A.S.; Rinke, P. Data-Driven Materials Science: Status, Challenges, and Perspectives. *Adv. Sci.* **2019**, *6*, 1900808. [\[CrossRef\]](#)
- Li, Z.; Xu, Q.; Sun, Q.; Hou, Z.; Yin, W.J. Thermodynamic Stability Landscape of Halide Double Perovskites via High-Throughput Computing and Machine Learning. *Adv. Funct. Mater.* **2019**, *29*, 1807280. [\[CrossRef\]](#)
- Brown, K.A.; Brittman, S.; Maccaferri, N.; Jariwala, D.; Celano, U. Machine Learning in Nanoscience: Big Data at Small Scales. *Nano Lett.* **2020**, *20*, 2–10. [\[CrossRef\]](#) [\[PubMed\]](#)
- Ma, X.-Y.; Lyu, H.-Y.; Hao, K.-R.; Zhao, Y.-M.; Qian, X.; Yan, Q.-B.; Su, G. Large family of two-dimensional ferroelectric metals discovered via machine learning. *Sci. Bull.* **2020**, *66*, 233–242. [\[CrossRef\]](#)
- Wang, T.; Zhang, C.; Snoussi, H.; Zhang, G. Machine Learning Approaches for Thermoelectric Materials Research. *Adv. Funct. Mater.* **2020**, *30*, 1906041. [\[CrossRef\]](#)
- Agiorgousis, M.L.; Sun, Y.; Choe, D.; West, D.; Zhang, S. Machine learning augmented discovery of chalcogenide double perovskites for photovoltaics. *Adv. Theory Simul.* **2019**, *2*, 1800173. [\[CrossRef\]](#)
- Wang, J.; Xu, P.; Ji, X.; Li, M.; Lu, W. Feature Selection in Machine Learning for Perovskite Materials Design and Discovery. *Materials* **2023**, *16*, 3134. [\[CrossRef\]](#) [\[PubMed\]](#)
- Kusuma, F.J.; Widiyanto, E.; Wahyono; Santoso, I.; Sholihun; Absor, M.A.U.; Sakti, S.P.; Triyana, K. Multi-Properties prediction of perovskite materials using Machine learning and Meta-Heuristic feature selection. *Sol. Energy* **2025**, *286*, 113189. [\[CrossRef\]](#)
- Park, H.; Mall, R.; Ali, A.; Sanvito, S.; Bensmail, H.; El-Mellouhi, F. Importance of structural deformation features in the prediction of hybrid perovskite bandgaps. *Comput. Mater. Sci.* **2020**, *184*, 109858. [\[CrossRef\]](#)
- Guo, Z.; Lin, B. Machine learning stability and band gap of lead-free halide double perovskite materials for perovskite solar cells. *Sol. Energy* **2021**, *228*, 689–699. [\[CrossRef\]](#)
- Yang, X.; Li, L.; Tao, Q.; Lu, W.; Li, M. Rapid discovery of narrow bandgap oxide double perovskites using machine learning. *Comput. Mater. Sci.* **2021**, *196*, 110528. [\[CrossRef\]](#)
- Yang, Z.; Liu, Y.; Zhang, Y.; Wang, L.; Lin, C.; Lv, Y.; Ma, Y.; Shao, C. Machine Learning Accelerates the Discovery of Light-Absorbing Materials for Double Perovskite Solar Cells. *J. Phys. Chem. C* **2021**, *125*, 22483–22492. [\[CrossRef\]](#)
- Kim, J.; Kim, E.; Min, K. Synthesizable Double Perovskite Oxide Search via Machine Learning and High-Throughput Computational Screening. *Adv. Theory Simul.* **2021**, *4*, 2100263. [\[CrossRef\]](#)
- Gao, Z.; Zhang, H.; Mao, G.; Ren, J.; Chen, Z.; Wu, C.; Gates, I.D.; Yang, W.; Ding, X.; Yao, J. Screening for lead-free inorganic double perovskites with suitable band gaps and high stability using combined machine learning and DFT calculation. *Appl. Surf. Sci.* **2021**, *568*, 150916. [\[CrossRef\]](#)
- Tian, G.; Zhao, P.; Wang, Y.; Zhang, H.; Xing, L.; Cheng, X. Research on Performance Prediction Method of Refractory High-Entropy Alloy Based on Ensemble Learning. *Metals* **2025**, *15*, 371. [\[CrossRef\]](#)

26. Chen, H.; Shang, Z.; Lu, W.; Li, M.; Tan, F. A Property-Driven Stepwise Design Strategy for Multiple Low-Melting Alloys via Machine Learning. *Adv. Eng. Mater.* **2021**, *23*, 2100612. [\[CrossRef\]](#)
27. Lu, T.; Li, H.; Li, M.; Wang, S.; Lu, W. Inverse Design of Hybrid Organic–Inorganic Perovskites with Suitable Bandgaps via Proactive Searching Progress. *ACS Omega* **2022**, *7*, 21583–21594. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Oriola, O. A Stacked Generalization Ensemble Approach for Improved Intrusion Detection. *Int. J. Comput. Sci. Inf. Secur.* **2020**, *18*, 62–67.
29. R., S.; M., S.; Hasan, M.K.; Saeed, R.A.; Alsuhibany, S.A.; Abdel-Khalek, S. An Empirical Model to Predict the Diabetic Positive Using Stacked Ensemble Approach. *Front. Public Health* **2021**, *9*, 792124. [\[CrossRef\]](#)
30. Liu, R.; Liu, Y.; Duan, J.; Hou, F.; Wang, L.; Zhang, X.; Li, G. Ensemble learning directed classification and regression of hydrocarbon fuels. *Fuel* **2022**, *324*, 124520. [\[CrossRef\]](#)
31. Meharie, M.G.; Mengesha, W.J.; Gariy, Z.A.; Mutuku, R.N.N. Application of stacking ensemble machine learning algorithm in predicting the cost of highway construction projects. *Eng. Constr. Archit. Manag.* **2022**, *29*, 2836–2853. [\[CrossRef\]](#)
32. Asante-Okyerere, S.; Marfo, S.A.; Ziggah, Y.Y. Estimating total organic carbon (TOC) of shale rocks from their mineral composition using stacking generalization approach of machine learning. *Upstream Oil Gas Technol.* **2023**, *11*, 100089. [\[CrossRef\]](#)
33. Wang, T.; Zhang, K.; Thé, J.; Yu, H. Accurate prediction of band gap of materials using stacking machine learning model. *Comput. Mater. Sci.* **2022**, *201*, 110899. [\[CrossRef\]](#)
34. Chen, R.; Zhou, S.; Liu, X.; Zhang, W.; Cheng, Q.; Jiang, N.; Zhang, Y.; Liu, D.; Yan, D. Application of Stacking machine learning model with SHapley Additive explanation (SHAP) in estimating the changes of mechanical properties in coals under CO₂. *Fuel* **2026**, *405*, 136461. [\[CrossRef\]](#)
35. Deng, X.; Zhu, S.P.; Zhang, S.; Zhang, X.; Xiong, R.; Dong, Y.; Yan, D. Physics-informed machine learning framework for creep-fatigue life prediction of a Ni-based superalloy using ensemble learning. *Mater. Today Commun.* **2024**, *41*, 110260. [\[CrossRef\]](#)
36. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [\[CrossRef\]](#)
37. Do Lee, B.; Lee, J. Discovery of Pb-free hybrid organic–inorganic 2D perovskites using a stepwise optimization strategy. *npj Comput. Mater.* **2022**, *8*, 83. [\[CrossRef\]](#)
38. Etghani, M.M.; Shojaeefard, M.H.; Khalkhali, A.; Akbari, M. A hybrid method of modified NSGA-II and TOPSIS to optimize performance and emissions of a diesel engine using biodiesel. *Appl. Therm. Eng.* **2013**, *59*, 309–315. [\[CrossRef\]](#)
39. Zhou, Z.; Lin, Z.; Ma, Y.; Niu, J.R.; Liu, J.; Wang, X. Optimal design of colour formulation prediction for cotton fabrics based on NSGA-II and TOPSIS. *Color. Technol.* **2025**, *141*, 63–79. [\[CrossRef\]](#)
40. Kocaturk, A.; Orku, H.H.; Altunkaynak, B. Parameter optimization in biclustering algorithms for large datasets using a combined approach of NSGA-II and TOPSIS. *Int. J. Data Sci. Anal.* **2025**, *20*, 5499–5516. [\[CrossRef\]](#)
41. Sadni, F.E.; Salhi, I.; Belhora, F.; Hajjaji, A. Multi-objective optimization of energy and exergy efficiencies in ORC configurations using NSGA-II and TOPSIS. *Therm. Sci. Eng. Prog.* **2025**, *63*, 103606. [\[CrossRef\]](#)
42. Welchowski, T.; Schmid, M. A framework for parameter estimation and model selection in kernel deep stacking networks. *Artif. Intell. Med.* **2016**, *70*, 31–40. [\[CrossRef\]](#)
43. Tandon, S.; Kumar, V.; Singh, V.B. Empirical Analysis of Machine Learning Classifiers for Architectural Smells Using Critic-Topsis Approach. In Proceedings of the 2024 3rd Edition of IEEE Delhi Section Flagship Conference (DELCON), New Delhi, India, 21–23 November 2024. [\[CrossRef\]](#)
44. Lundberg, S.M.; Erion, G.G.; Lee, S.-I. Consistent individualized feature attribution for tree ensembles. *arXiv* **2018**, arXiv:1802.03888.
45. Lyngdoh, G.A.; Li, H.; Zaki, M.; Krishnan, N.M.A.; Das, S. Elucidating the constitutive relationship of calcium–silicate–hydrate gel using high throughput reactive molecular simulations and machine learning. *Sci. Rep.* **2020**, *10*, 21336. [\[CrossRef\]](#)
46. Kusdhany, M.I.M.; Lyth, S.M. New insights into hydrogen uptake on porous carbon materials via explainable machine learning. *Carbon* **2021**, *179*, 190–201. [\[CrossRef\]](#)
47. Wang, G.; Fearn, T.; Wang, T.; Choy, K.L. Machine-Learning Approach for Predicting the Discharging Capacities of Doped Lithium Nickel–Cobalt–Manganese Cathode Materials in Li-Ion Batteries. *ACS Cent. Sci.* **2021**, *7*, 1551–1560. [\[CrossRef\]](#)
48. Lu, T.; Li, H.; Li, M.; Wang, S.; Lu, W. Predicting Experimental Formability of Hybrid Organic–Inorganic Perovskites via Imbalanced Learning. *J. Phys. Chem. Lett.* **2022**, *13*, 3032–3038. [\[CrossRef\]](#)
49. Liang, G.Q.; Zhang, J. A machine learning model for screening thermodynamic stable lead-free halide double perovskites. *Comput. Mater. Sci.* **2022**, *204*, 111172. [\[CrossRef\]](#)
50. Grojean, C.; Paul, A.; Qian, Z.N.; Strumke, I. Lessons on interpretable machine learning from particle physics. *Nat. Rev. Phys.* **2022**, *4*, 284–286. [\[CrossRef\]](#)
51. Bartel, C.J.; Sutton, C.; Goldsmith, B.R.; Ouyang, R.; Musgrave, C.B.; Ghiringhelli, L.M.; Scheffler, M. New tolerance factor to predict the stability of perovskite oxides and halides. *Sci. Adv.* **2019**, *5*, eaav0693. [\[CrossRef\]](#)
52. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; Routledge: Abingdon, UK, 2017.

53. Huynh-Thu, V.A.; Irrthum, A.; Wehenkel, L.; Geurts, P. Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE* **2010**, *5*, e12776. [\[CrossRef\]](#) [\[PubMed\]](#)
54. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.-I. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2020**, *2*, 56–67. [\[CrossRef\]](#)
55. Li, Y.; Tie, W.; Hu, X.; Duan, B.; Shao, Y.; Meng, Y.; Fang, J. Optimization of Beamline Optical Systems Using NSGA-II and TOPSIS Algorithms. In Proceedings of the 2025 4th International Conference on Advanced Manufacturing Technology and Manufacturing System (ICAMTMS), Wuhu, China, 8–10 August 2025. [\[CrossRef\]](#)
56. Liu, H.; Cheng, J.; Dong, H.; Feng, J.; Pang, B.; Tian, Z.; Ma, S.; Xia, F.; Zhang, C.; Dong, L. Screening stable and metastable ABO₃ perovskites using machine learning and the materials project. *Comput. Mater. Sci.* **2020**, *177*, 109614. [\[CrossRef\]](#)
57. Zhai, X.; Chen, M. Accelerated Design for Perovskite-Oxide-Based Photocatalysts Using Machine Learning Techniques. *Materials* **2024**, *17*, 3026. [\[CrossRef\]](#)
58. Song, K.; Yan, F.; Ding, T.; Gao, L.; Lu, S. A steel property optimization model based on the XGBoost algorithm and improved PSO. *Comput. Mater. Sci.* **2020**, *174*, 109472. [\[CrossRef\]](#)
59. Hancock, J.T.; Khoshgoftaar, T.M. CatBoost for big data: An interdisciplinary review. *J. Big Data* **2020**, *7*, 94. [\[CrossRef\]](#) [\[PubMed\]](#)
60. Degtyarev, V.V.; Naser, M.Z. Boosting machines for predicting shear strength of CFS channels with staggered web perforations. *Structures* **2021**, *34*, 3391–3403. [\[CrossRef\]](#)
61. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
62. Liaw, A.; Wiener, M. Classification and regression by randomForest. *R News* **2002**, *2*, 18–22.
63. Song, Z.; Chen, X.; Meng, F.; Cheng, G.; Wang, C.; Sun, Z.; Yin, W.-J. Machine learning in materials design: Algorithm and application. *Chin. Phys. B* **2020**, *29*, 116103. [\[CrossRef\]](#)
64. Xu, Y.; Zhao, X.; Chen, Y.; Yang, Z. Research on a mixed gas classification algorithm based on extreme random tree. *Appl. Sci.* **2019**, *9*, 1728. [\[CrossRef\]](#)
65. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. LightGBM: A highly efficient gradient boosting decision tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
66. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [\[CrossRef\]](#)
67. Bartlett, P.; Freund, Y.; Lee, W.S.; Schapire, R.E. Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Stat.* **1998**, *26*, 1651–1686. [\[CrossRef\]](#)
68. Blum, L.C.; Raymond, J.L. 970 Million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.* **2009**, *131*, 8732–8733. [\[CrossRef\]](#) [\[PubMed\]](#)
69. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [\[CrossRef\]](#)
70. Zhai, X.; Chen, M.; Lu, W. Accelerated search for perovskite materials with higher Curie temperature based on the machine learning methods. *Comput. Mater. Sci.* **2018**, *151*, 41–48. [\[CrossRef\]](#)
71. Li, C.; Hao, H.; Xu, B.; Shen, Z.; Zhou, E.; Jiang, D.; Liu, H. Improved physics-based structural descriptors of perovskite materials enable higher accuracy of machine learning. *Comput. Mater. Sci.* **2021**, *198*, 110714. [\[CrossRef\]](#)
72. Rajan, A.C.; Mishra, A.; Satsangi, S.; Vaish, R.; Mizuseki, H.; Lee, K.R.; Singh, A.K. Machine-learning-assisted accurate band gap predictions of functionalized mxene. *Chem. Mater.* **2018**, *30*, 4031–4038. [\[CrossRef\]](#)
73. Chou, J.S.; Tsai, C.F.; Pham, A.D.; Lu, Y.H. Machine learning in concrete strength simulations: Multi-nation data analytics. *Constr. Build. Mater.* **2014**, *73*, 771–780. [\[CrossRef\]](#)
74. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
75. Zhao, X.G.; Yang, J.H.; Fu, Y.; Yang, D.; Xu, Q.; Yu, L.; Wei, S.H.; Zhang, L. Design of Lead-Free Inorganic Halide Perovskites for Solar Cells via Cation-Transmutation. *J. Am. Chem. Soc.* **2017**, *139*, 2630–2638. [\[CrossRef\]](#)
76. Li, W.; Jacobs, R.; Morgan, D. Predicting the thermodynamic stability of perovskite oxides using machine learning models. *Comput. Mater. Sci.* **2018**, *150*, 454–463. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.