# Machine Learning in Predicting Lattice Constant of Cubic Perovskite Oxides

*Ujjwal Poudel, Madhu Sudhan Bhusal, Manish Bhurtel, Atish Adhikari and Narayan Prasad Adhikari*

**Journal of Nepal Physical Society**
Volume 8, No 1, 2022
(Special Issue: ICFP 2022)
ISSN: 2392-473X (Print), 2738-9537 (Online)

JNPS, **8** (1), 27-34 (2022)
DOI: http://doi.org/10.3126/jnphyssoc.v8i1.48282

**Published by: Nepal Physical Society**
P.O. Box: 2934
Tri-Chandra Campus
Kathmandu, Nepal
Email: nps.editor@gmail.com

**ICFP 2022**
ISSN:2392-473X (Print)
2738-9537 (Online)

**Research Article ©Nepal Physical Society**
DOI:http://doi.org/10.3126/jnphyssoc.v8i1.48282

# Machine Learning in Predicting Lattice Constant of Cubic Perovskite Oxides

Ujjwal Poudel,[1, a)] Madhu Sudhan Bhusal,[1, b)] Manish Bhurtel,[2] Atish Adhikari,[3] and Narayan Prasad Adhikari[4, c)]

[1)] *Department of Physics, St. Xavier's College, Kathmandu, Nepal.*
[2)] *Institute of Engineering, Tribhuvan University, Kathmandu, Nepal.*
[3)] *Deerhold Ltd, Kathmandu, Nepal.*
[4)] *Central Department of Physics, Tribhuvan University, Kathmandu Nepal*

[a)] *Corresponding author: ujjwal.poudel44@gmail.com*
[b)] *Electronic mail: binalmsb@gmail.com*
[c)] *Electronic mail: npadhikari@gmail.com*

**Abstract.**
A sample of 3,115 data of perovskite oxides in the form of $ABO_3$ (A and B being the cations) was taken for this study of the application of machine learning in predicting the lattice constants (a determining factor in material design). The ANN, DT, RF, KNN, and SVM models were used to predict the lattice constants of perovskites because machine learning techniques have been phenomenal in uncovering crystal structures in the field of material research in recent years. These models used properties like ionic radii, formation energy, and band gap as input features. The $R_2$ score was used to assess the regression model's performance. The Random Forest Regression Model outperforms all other regression models regarding dataset and features.

**Received**: 22 March, 2022; **Revised**: 18 April, 2022; **Accepted**: 28 April, 2022

## INTRODUCTION

The crystal structure of perovskite is similar to that of calcium titanate ($CaTiO_3$). Perovskite has the general formula $ABX_3$, where A and B are alkaline earth (Ca, Ba, Sr, etc.) and transition metal (Fe, Ti, Ni, etc.) cations, respectively, and X is an oxyhalide ion. The B cations are in 6-fold coordination surrounded by an octahedron of anions in an ideal cubic structure, and the A cation is in 12-fold cuboctahedral coordination. In cubic space, they belong to the Pm3m group [1, 2].

Tetragonal, orthorhombic, rhombohedral, monoclinic, and triclinic structures exist in perovskite, depending on structural variations caused by tilting characteristics such as ferroelectricity, charge ordering, spin-dependent transport, high thermopower, and huge magnetoresistance. As a result of their wide range of applications, these characteristics have been frequently exploited in materials research [1]. Several microscopic parameters in material design, such as band gap energy, electron affinity, molecule atomization energy, and lattice constant, are essential in material performance. The lattice mismatch between layers made up of different materials is one of the critical concerns in material design. To solve this problem, predicting the lattice constant from many conceivable combinations of elements is extremely difficult and time-consuming [3].

Researchers are employing various computational tools such as Density Functional Theory (DFT) [4]- [5] as a result of advancements in high-performance computing approaches. DFT is based on quantum mechanics' first principle, which solves the Schrodinger equation by minimizing the system's total energy using a density functional equation known as the Kohn-Sham equation [6]-[7]. The electronic structure of atoms or molecules and the elastic characteristics of compounds are determined separately using the DFT method. Significant computational resources are necessary when utilizing DFT [8] as well as in-depth knowledge of the electrical configuration, bonding energy, charge distribution, and density of the states of the material under study [9].

In recent years, machine learning (ML)-based techniques have offered several advantages over DFT. The physicochemical properties of substances can be determined using different predictive models in machine learning. These models learn from training datasets and predict compound attributes, allowing for high-performance prediction at a low temporal and computational cost.

Different machine learning techniques have been used to estimate the lattice constants of perovskite materials for obtaining lattice constants in $GdFeO_3$ - type $ABO_3$ perovskites, Li et al. employed LR and ANN models [10]. The LR model used elemental ionic radii, whereas the ANN model used other fundamental properties such as the electronegativities of cations A and B and the valence of ion A, including the ionic radii. The ANN model was proven more accurate than the LR model during the experiment. The accuracy error limit was within 2 %, although the prediction accuracy on testing data was lower than on training data. The SVR model outperformed the ANN model on both training and testing data of orthorhombic $ABO_3$ perovskites. [11].

The mean percentage absolute difference (PAD) values obtained for all lattice constants were 0.43 percent, 0.54 percent, and 0.96 percent, indicating that the SVR technique had the best prediction performance. Majid et al. also worked on cubic perovskites, using solely ionic radii as an essential feature to calculate lattice constants. His research used SVR, RF, GRNN, and multiple linear regression (MLR) as machine learning models. According to the study, the SVR model outperformed the others regarding training and testing data sets' accuracy. [2].

In the previous works, it was found that the lattice constant prediction of perovskites was carried out using very few models. This paper uses prediction models such as ANN, DT, RF, KNN, and SVM to calculate the lattice constants of cubic perovskites.

## METHODOLOGY

In the present work, machine learning techniques have been employed to study the lattice constants of the perovskite materials. In predicting the lattice constants, properties such as ionic radii, band gap, and formation energy were used as input parameters. The experimental value of the lattice constant was taken from Wolverton Oxides Data [12]. A total of 3,115 cubic perovskite oxides were taken, where 2,920 compounds were used for training and 195 were used for testing. The testing compounds have been given in the Appendix section. After splitting the datasets, a 16-fold cross-validation was applied for the DT, RF, KNN, SVR, and ANN models. Random Forest Regression provides the maximum value of the $R^2$ score, and the data sets were used to evaluate the performance for the rest of the regression models. Figure 1 shows the block diagram used to carry out all the calculations in this study, including a detailed flow of the methodology.

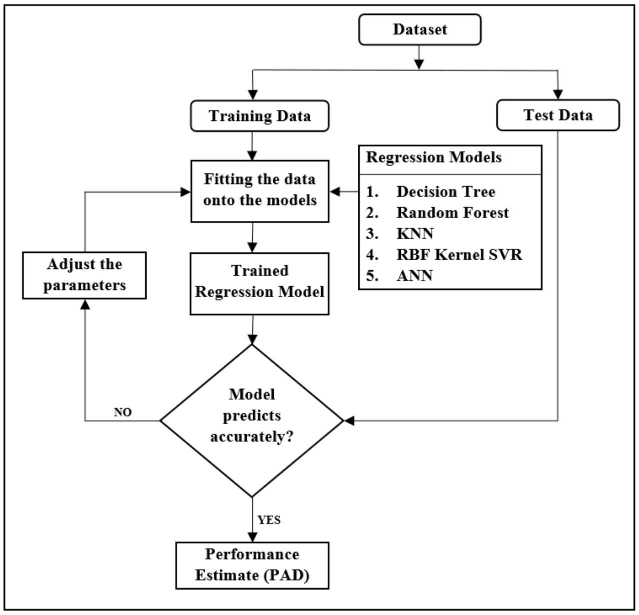


**FIGURE 1.** Block Diagram

The performance evaluation was measured by calculating PAD and $R^2$ values.

Percentage of absolute difference is the performance measure used to evaluate the predictive accuracy and is defined as:

$$PAD\ \% = \frac{Y_{true} - Y_{pred}}{Y_{true}} * 100 \tag{1}$$

where,
$Y_{true}$: true values of the lattice constants
$Y_{pred}$: predicted values of lattice constants

The $R^2$ score is the measure of goodness of the regression model i.e., it represents how much the data converged around the mean value. The $R^2$ score was used to evaluate the performance of the regression models.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2} \tag{2}$$

where
$y_i$ are the actual values
$\hat{y}_i$ are the predicated values
$\bar{y}_i$ are the mean values

# MODEL

## Artificial Neural Network (ANN)

The Artificial Neural Network (ANN) is the formal mathematical description of neurons and the network structure. It suggests that a single neuron can perform the logical function. ANN has three types of layers, namely Input Layer, Output Layer, and Hidden Layer. The input layer receives the data (or signal) from the outside world. The layers of the hidden neurons lie between the input and the output layers, and which cannot be viewed from outside the network [13].

ANN is the group of nodes interconnected to one another which work together to quantify a prediction [14]. They are composed of simple elements called neurons or nodes that receive a set of weighted inputs, process their sum with its activation function $\phi$, and pass the result to nodes further down the graph [15]. With this procedure, a neuron can make simple mathematical decisions.

All the nodes are connected to form a neural network. Usually, this is done in layers—the inputs to a layer of nodes are the outputs from the previous layer. Together, the neurons can analyze complex problems, emulate almost any function including the complicated ones, and provide accurate answers [16].

The goal of this study is to train a network using labeled data so that a set of inputs can be fed into it, and thereby allowing the network to produce the appropriate outputs for the unlabeled data. Since it is a supervised learning algorithm, the input $x_i$ and their corresponding labels $y_i$ are used in pairs. In the first iteration of the training process, the values in the neurons are calculated using the weighted inputs. The values in the output neurons are calculated as well. The predicted and the actual values are compared using the appropriate loss function in order to estimate the loss or error. Afterward, the gradient descent optimization is performed to update the weights and eventually reduce the loss value. The training process is repeated until the optimal weights are achieved [17, 18]. The network and its trained weights form a function (denoted h) that operates on input data. With the trained network, predictions can be made once an unlabeled test input is provided [19]. For the input features $x_i$, the predicted value is shown as [17].

$$\widehat{y} = g\left[\beta_i h\left(\sum_{i=1}^{n} x_i\alpha_i + \phi\right)\right] \tag{3}$$

where,
$x_i$ is the input variables;
$\alpha$ is the weight connecting input node to hidden node;
$\beta$ is the weight connecting hidden node to output node;
$\phi$ is the corresponding bias;
$g()$, $h()$ are the activation functions at hidden and output layers respectively.

The weights are updated in the next phase of back propagation with the following formula [18]

$$W^{(l)} = W^{(l)} - \eta\ \Delta_{W^{(l)}}L \tag{4}$$

where,
$W^{(l)}$ is a set of weights associated with the neurons in layer l;
$\eta$ is the learning rate;
$\Delta W^{(l)}L$ is the Partial Derivative of the quantified Loss with respect to the weights.

Several experiments were performed with the hyperparameters, thereby achieving 80% of the $R^2$ score with the following specifications:

Loss Function: Mean Squared Error; Metrics: Mean Absolute Error; Optimizer: Adam; Epochs: 200.

## K - Nearest Neighbour (KNN)

K-Nearest Neighbor (KNN) is another machine learning algorithm that predicts the targeted numerical values for the corresponding input values using certain similarity metrics (e.g., distance functions). KNN is an effective way of predicting the values of a function in a point with reference to its values in other points [20]. There are different ways of measuring how ‘close’ two points are. The “Euclidean Distance” has been used as the similarity measure. The Euclidean distance between two points is defined as following [21].

$$ED(x,y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|} \tag{5}$$

The value ‘K’ determines the number of similar points to consider while making the prediction (nearest neighbors). The implementation of the KNN regression is to calculate the average of the numerical target of the K nearest neighbors. Regarding the data in this study, the Euclidean Distance was used as the evaluation metric, whereas the five neighbors for the KNN workflow were considered.

## Decision Tree

A decision tree is a supervised machine learning model that is used to predict a target by learning decision rules from features. A decision tree is constructed by recursive partitioning-starting from the root node (known as the first parent), each node can then be split into left and right child nodes. These nodes can be further split, until they themselves become parent nodes of their resulting children nodes.

Starting from the root, the data were split on the feature that results in the highest homogeneity. The Mean Squared Error (MSE) was used to calculate the homogeneity of a numerical sample. The MSE can be calculated as [22].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_{obs,i} - X_{model,i})^2 \qquad (6)$$

where,<br>
$n$ is the number of training samples;<br>
$X_{obs,i}$ is the true value for ith data;<br>
$X_{model,i}$ is the predicted value for $i^{th}$ data.

If the numerical sample is completely homogeneous, the value of MSE is zero. The MSE reduction is based on the decrease in MSE after a dataset is split on an attribute. Constructing a decision tree is all about finding the attribute that returns the highest MSE reduction (i.e., the most homogeneous branches).

There may arise anomalies in the decision trees due to outliers or noise. Slight changes in the values may result in completely different results. Constraints like tree max-depth, max-leaf nodes, etc. were implemented to eradicate the inconsistency and prevent overfitting. Starting from the maximum tree depth of 5, a maximum tree depth of 10 was experimented with and used to achieve the $R^2$ score of 90.7 %.

## Random Forest Method (RF)

The random forests model improves by introducing an additional layer of randomness to bootstrap aggregating, which is used to predict the LC of perovskites compounds [23]. A set of 2,920 training samples were randomly selected as subsets descriptors to grow trees by RF algorithms.

A random forest is a meta-estimator (i.e., it combines the result of multiple predictions) that aggregates many decision trees, with some helpful modifications:

- In each node, the split in the number of features is limited to a certain percentage of the whole. This ensures that the random forest model does not rely too heavily on any individual feature and makes fair use of all potentially predictive features.

- A random sample from the original data is drawn while generating the splits. Randomness is added to avoid overfitting.

The predictions of individual trees are aggregated—through averaging-into a single ensemble random forest model. The ensemble trains the multiple models on training data and uses the prediction of all models to make the final output [24].

In the RF network, a randomly generated tree ($n_{tree}$ = 2920) was provided for better performance. The values of an independently sampled random vector gave the output in this case. After multiple runs on the training data, the best set of data—one that gives the best results—would finally be reported. Similar to the decision tree, a maximum depth of 20 and the minimum samples split of 2 with an experiment for the RF model were used. The quality of the split was measured using the Mean Squared Error.

## Support Vector Machine

Support Vector Regression is used for various applications in materials science [10, 25, 26]. It is a powerful machine learning algorithm that allows the researchers to choose how tolerant they are of errors, both through an acceptable error margin ($\varepsilon$)and through tuning their tolerance of falling outside that acceptable error rate [25].

Theoretical details of SVR models are available in the statistical learning theory [27], but it gives the researchers the flexibility to define the extent of acceptable error in a particular model and find out an appropriate line (or hyperplane in higher dimensions) to fit the data.

In contrast to OLS described in Linear Regression, the objective function of SVR is to minimize the coefficients, more specifically, the l2-norm of the coefficient vector, not the squared error. The error term is instead handled in the constraints, where the absolute error is set less than or equal to a specified margin, called the maximum error ($\varepsilon$). The epsilon can be tuned to gain the desired accuracy for the given model. The new objective function and constraints are as follows [28].

Minimize:

$$\frac{1}{2}\|w\|^2 \tag{7}$$

Subject to:

$$\begin{cases} y_i - < w, x_i > - b \le \varepsilon \\ < w, x_i > + b - y_i \le \varepsilon \end{cases} \tag{8}$$

where,
$w_i$ is the coefficient vector / weights;
$x_i$ is the training sample;
$y_i$ is the predicted values for $x_i$;
$b$ is the intercept / bias.

SVR can also use the kernel functions to transform the data into a higher dimensional feature space and make it possible to perform the linear separation.

In this case, a Radial Basis Function (RBF) kernel was used to support the vector regression. The RBF kernel, along with two feature vectors, $x$ and $x'$ is defined as [29].

$$K(x, x') = exp(-\frac{\|x - x'\|^2}{2\sigma^2}) \tag{9}$$

where,
$\|x - x'\|$ is the squared Euclidean distance between the feature vectors;
$\sigma$ = free parameter.

## RESULTS AND DISCUSSION

Table 1 shows the outcome of the lattice constant prediction for cubic perovskite oxide. One hundred ninety-five compounds have been given in the table, along with their lattice constants a, b, and c, which are used to test the various models. The prediction accuracy is assessed by comparing the experimental and predicted values, using PAD and $R^2$ values from 0 to 1. The smaller the PAD and the higher $R^2$, the more accurate values are with the experimental ones.

### Comparison in terms of PAD

The lattice constants of the compounds were predicted using factors such as ionic radii and electronegativity in previous work. However, to determine the prediction accuracy of the lattice constant of cubic perovskites, characteristics such as ionic radii, band gap, and formation energy were used as input parameters in this study (in the form $ABO_3$). The band gap and lattice constant correlation were determined to be 0.041, which indicates a lesser association. The correlation values of ionic radii and formation energy with the lattice constant were 0.31, 0.80, and 0.29, demonstrating a significant association. For ANN, RF, KNN, DT, and SVR models, mean PAD values of 0.03, 0.01, 0.02, 0.02, and 0.02 were obtained.

### Performance Comparison in terms of Linear Correlation

In terms of linear correlation, the performance of prediction models was also compared. The linear fit graph between the experimental and anticipated values for the testing data set is shown in Figures 1-5.

The observed and predicted lattice constants using KNN are shown in Figure 2. The $R^2$ value was 0.785, indicating that the data averaged around the best fit line.

Figure 3 shows how the SVR with RBF kernel was utilized to forecast the lattice constant values. Most values were mapped around the best fit line with a few outliers. The $R^2$ value was 0.803, showing that the correlation was better than KNN.

Figure 4 depicts the linear correlation concerning ANN in the same way. The ultimate $R^2$ score was 0.809 after several tests with the hyperparameters of the neural network.

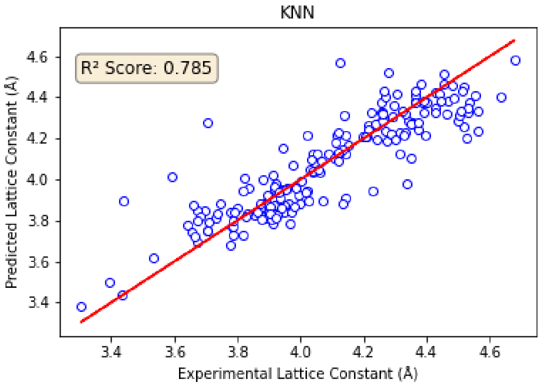
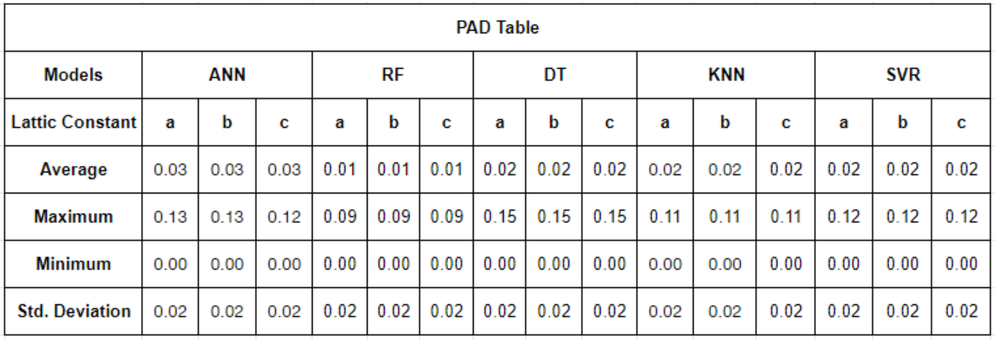


**FIGURE 2.** Experimental Lattice Constant of KNN

A decent correlation with the $R^2$ score of 0.907 was obtained using the decision tree algorithm as shown in figure

**TABLE I.** Summary of Percentage Absolute Difference.



| PAD Table | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Models | ANN | | | RF | | | DT | | | KNN | | | SVR | | |
| Lattic Constant | a | b | c | a | b | c | a | b | c | a | b | c | a | b | c |
| Average | 0.03 | 0.03 | 0.03 | 0.01 | 0.01 | 0.01 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 |
| Maximum | 0.13 | 0.13 | 0.12 | 0.09 | 0.09 | 0.09 | 0.15 | 0.15 | 0.15 | 0.11 | 0.11 | 0.11 | 0.12 | 0.12 | 0.12 |
| Minimum | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Std. Deviation | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 |

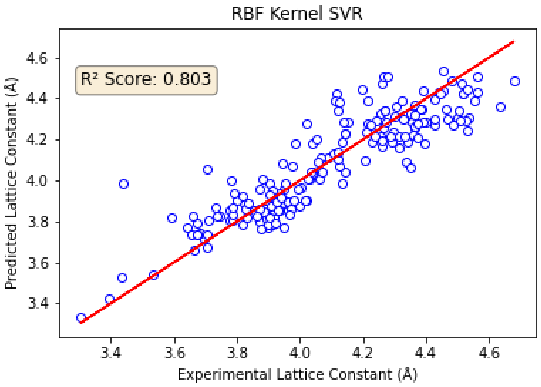


**FIGURE 3.** Experimental Lattice Constant of SVR

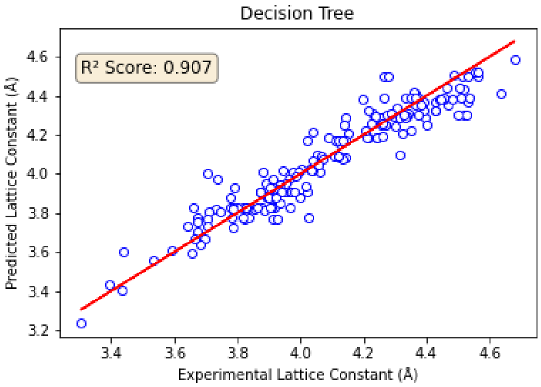


**FIGURE 5.** Experimental Lattice Constant of Decision Tree

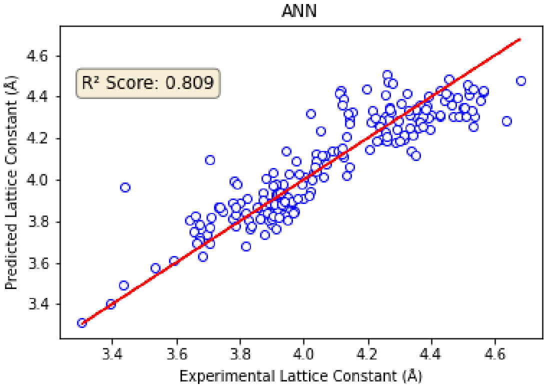


**FIGURE 4.** Experimental Lattice Constant of ANN

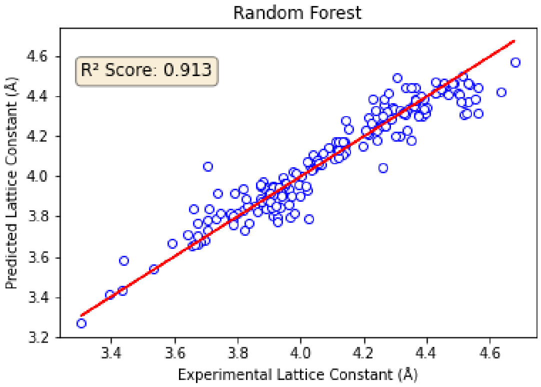


**FIGURE 6.** Experimental Lattice Constant of Random Forest

5. Almost all the data points are centered around the best fit line, showing a minimum error and a decent prediction.

Finally, the best result was obtained using the Random Forest Algorithm that outperformed all the other regression models. Minimum anomalies were observed as shown in figure 6, and the $R^2$ score was obtained to be 0.913.

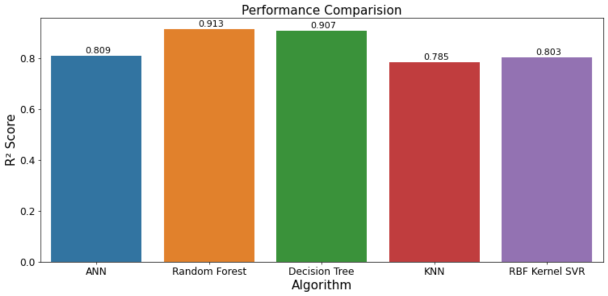


**FIGURE 7.** Performance comparison of different ML models

## CONCLUSIONS

The performance evaluation of various prediction models was compared in this study. A total of 3,115 experimental datasets were employed, with 70 % used to train the algorithm and 30 % used to test model correctness. The empirical and projected lattice constants are compared in Fig. 7 using various regression models. The $R^2$ ratings generated from all of the algorithms are near in value, as evidenced by the height of the bars. The overall results provided in the research, however, show that the Random Forest Regression (RF) model predicted the lattice constants of cubic perovskites with greater accuracy than the other models used. This methodology can be used to estimate the lattice constant of double perovskites, which have also been employed as a material for an active layer in solar cells in the future.

## REFERENCES

1. J. Fowlie, (2019).
2. G. J. A. M. M. A. Majid, A. Khan, “Lattice constant prediction of cubic and monoclinic perovskites using neural networks and support vector regression,” Comput Mater Sci **50**, 363–372 (2010).
3. S. S. Y. Liu, T. Zhao, (2017).
4. G. D. F. G. X. T. Z. W. J. W. H. Y. Xiao, X. D. Jiang, Comput. Mater. Sci. **48**, 768–772 (2010).
5. F. S. G. U. S. Ugur, N. ArIkan, Comput. Mater. Sci. **48**, 866–870 (2010).
6. M. C. K. G.-N. X. G. C. Motta, M. Giantomassi, Comput. Mater. Sci. **48**, 866–870 (2010).
7. M. K. T. S.-A. H. R. Y. A.-D. A. Bouhemadou, R. Khenata, Comput. Mater. Sci. **45**, 474–479 (2009).
8. C. M. W. P. Wu, Y. Z. Zeng, Biomaterials **25**, 1123 – 1130 (2004).
9. F. C. Vallejo, “Serc short report,” Technical University of Denmark (2008).
10. Y. Z. Z. C. M. W. P. W. C. H. Li, Y. H. Thing, “Prediction of lattice constant in perovskites of gdfeo3 structure,” J Phys Chem Solids **64**, 2147–2156 (2003).
11. A. M. A. M. M. J. B. S. G. Javed, A. Khan, “Prediction of lattice constant in perovskites of gdfeo3 structure,” J Phys Chem Solids **64**, 2147–2156 (2003).
12. C. W. A. A. Emery, “High-throughput dft calculations of formation energy, stability and oxygen vacancy formation energy of $abo_3$ perovskites,” Sci. Data **4**, 170153 (2017).
13. A. P. A. P. L. Bertolaccini, P. Solli, “An overview of the use of artificial neural networks in lung cancer research,” Journal of Thoracic Disease **9**, 924 – 931 (2017).
14. M. D. P. T. M. S. Mhatre, F. Siddiqui, “A review paper on artificial neural network: A prediction technique,” International Journal of Scientific & Engineering Research **6**, 161 – 163 (2015).
15. F. Moksony, “Small is beautiful: The use and interpretatio of $r^2$ in social research,” Szociollogiai Szemle , 130 – 138 (1999).
16. A. D. V. Sharma, S. Rai, “A comprehensive study of artificial neural networks,” International Journal of Advanced Research in Computer Science and Software Engineering **2**, 278 – 284 (2012).
17. S. R. N. B. R. S. P. Mittal, S. Chowdhury, “Dual artificial neural network for rainfall-runoff forecasting,” Journal of Water Resource and Protection **4**, 1024 – 1028 (2012).
18. M. P. J. H. Lee, T. Delbruck, “Training deep spiking neural networks using backpropagation,” Frontiers of Neuroscience **10**, 508 (2016).
19. K. T. L. H. Y. Shi, S. L. Hwang, “In-hospital mortality after traumatic brain injury surgery: a nationwide population-based comparison of mortality predictors used in artificial neural network and logistic regression models,” J Neurosurg **118**, 746 – 752 (2013).
20. N. T. E. V. A. Navot, L. Shpigelman, “Nearest neighbor based feature selection for regression and its application to neural activity,” Conference Proceedings: Proceedings of the $18^{th}$ International Conference on Neural Information Processing Systems , 996 – 1002 (2005).
21. O. L. A. S. T. M. B. A. H. S. E. S. V. B. S. P. H. A. A. Alfeilat, A. B. A. Hassanat, “Effects of distance measure choice on k-nearest neighbor classifier performance: A review,” Mary Ann Liebert, Inc. **7**, 221 – 248 (2019).
22. K. A. K. A. G. P. Varshini, “Predictive analytics approaches for software effortestimation: A review,” Indian Journal Of Science And Technology **13**, 2094 – 2013 (2020).
23. N. A. I. M. J. Ali, R. Khan, “Random forests and decision trees,” Journal: International Journal of Computer Science Issues (IJCSI) **9**, 272 – 278 (2012).
24. L. Breiman, “Random forests,” Machine Learning **1**, 5 – 32 (45).
25. J. Y. S. J. J. W. B. T. Chen, T. P. Chang, “Comput, mater. sci,” Machine Learning **44**, 988 – 998 (2009).
26. T. S. C. A. Khan, M. H. Shamsi, “Comput. mater. sci.” Machine Learning **45**, 257 – 265 (2009).
27. V. Vapnik, “The nature of statistical learning theory,” Springer Verlag, NY (1995).
28. B. S. A. J. Smola, “A tutorial on support vector regression,” Springer Verlag, NY **14**, 199 – 222 (2004).
29. B. S. J. P. Vert, K. Tsuda, “Kernel methods in computational biology, a primer on kernel methods,” Springer Verlag, NY **14**, 11 – 54 (2004).
30. P. L. M. Johnsson, “Handbook of magnetism and advanced magnetic materials,” (2007).
31. X. C. L. Z. Y. W. J. Tan, G. Ji, Comput. Mater. Sci. **48**, 796–801 (2010).
32. M. Z. K. D. Z. H. G. S. W. Y. L. Z. S. L.Shang S.L., Saengdeejing A., Comput. Mater. Sci. **48**, 796–801 (2010).
33. H. C. Y. F. Y. D. Y. L. Y. Ouyang, X. Tao, Comput. Mater. Sci. **47**, 297–301 (2009).
34. K. D. S. M. C. K. Bouamama, P. Djemia, Comput. Mater. Sci. **47**, 308–313 (2009).
35. G. P. Berman, Jr. and F. M. Izrailev, Jr., “Stability of nonlinear modes,” Physica D **88**, 445 (1983).
36. E. B. Davies and L. Parns, “Trapped modes in acoustic waveguides,” Q. J. Mech. Appl. Math. **51**, 477–492 (1988).
37. E. Witten, (2001), hep-th/0106109.

38. E. Beutler, “Williams hematology,” (McGraw-Hill, New York, 1994) Chap. 7, pp. 654–662, 5th ed.
39. D. E. Knuth, “Fundamental algorithms,” (Addison-Wesley, Reading, Massachusetts, 1973) Section 1.2, pp. 10–119, 2nd ed., a full INBOOK entry.
40. J. S. Smith and G. W. Johnson, Philos. Trans. R. Soc. London, Ser. B **777**, 1395 (2005).
41. W. J. Smith, T. J. Johnson, and B. G. Miller, “Surface chemistry and preferential crystal orientation on a silicon surface,” J. Appl. Phys. (unpublished).
42. V. K. Smith, K. Johnson, and M. O. Klein, “Surface chemistry and preferential crystal orientation on a silicon surface,” J. Appl. Phys. (submitted).
43. U. Ünderwood, N. Ñet, and P. P̄ot, “Lower bounds for wishful research results,” (1988), talk at Fanstord University (A full UNPUBLISHED entry).
44. M. P. Johnson, K. L. Miller, and K. Smith, personal communication (2007).
45. J. Smith, ed., *AIP Conf. Proc.*, Vol. 841 (2007).
46. W. V. Oz and M. Yannakakis, eds., *Proc. Fifteenth Annual*, All ACM Conferences No. 17, ACM (Academic Press, Boston, 1983) a full PROCEEDINGS entry.
47. Y. Burstyn, “Proceedings of the 5th International Molecular Beam Epitaxy Conference, Santa Fe, NM,” (2004), (unpublished).
48. B. Quinn, ed., *Proceedings of the 2003 Particle Accelerator Conference, Portland, OR, 12-16 May 2005* (Wiley, New York, 2001) albeit the conference was held in 2005, it was the 2003 conference, and the proceedings were published in 2001; go figure.
49. A. G. Agarwal, “Proceedings of the Fifth Low Temperature Conference, Madison, WI, 1999,” Semiconductors **66**, 1238 (2001).
50. R. Smith, “Hummingbirds are our friends,” J. Appl. Phys. (these proceedings) Abstract No. DA-01.
51. J. Smith, Proc. SPIE **124**, 367 (2007), required title is missing.
52. T. Térrific, “An $O(n \log n / \log\log n)$ sorting algorithm,” Wishful Research Result 7 (Fanstord University, Computer Science Department, Fanstord, California, 1988) a full TECHREPORT entry.
53. J. Nelson, TWI Report 666/1999 (Jan. 1999) required institution missing.
54. W. K. Fields, ECE Report No. AL944 (2005) required institution missing.
55. Y. M. Zalkins, e-print arXiv:cond-mat/040426 (2008).
56. J. Nelson, U.S. Patent No. 5,693,000 (12 Dec. 2005).
57. J. K. Nelson, M.S. thesis, New York University (1999).
58. É. Masterly, *Mastering Thesis Writing*, Master’s project, Stanford University, English Department (1988), a full MASTERSTHESIS entry.
59. S. M. Smith, Ph.D. thesis, Massachusetts Institute of Technology (2003).
60. S. R. Kawa and S.-J. Lin, J. Geophys. Res. **108**, 4201 (2003), DOI:10.1029/2002JD002268.
61. F. P. Phony-Baloney, *Fighting Fire with Fire: Festooning French Phrases*, PhD dissertation, Fanstord University, Department of French (1988), a full PHDTHESIS entry.
62. D. E. Knuth, *Seminumerical Algorithms*, 2nd ed., The Art of Computer Programming, Vol. 2 (Addison-Wesley, Reading, Massachusetts, 1981) a full BOOK entry.
63. J. C. Knvth, “The programming of computer art,” Vernier Art Center, Stanford, California (1988), a full BOOKLET entry.
64. R. Ballagh and C. Savage, “Bose-einstein condensation: from atomic physics to quantum fluids, proceedings of the 13th physics summer school,” (World Scientific, Singapore, 2000) cond-mat/0008070.
65. W. Opechowski and R. Guccione, “Introduction to the theory of normal metals,” in *Magnetism*, Vol. IIa, edited by G. T. Rado and H. Suhl (Academic Press, New York) p. 105.
66. W. Opechowski and R. Guccione, “Introduction to the theory of normal metals,” in *Magnetism*, Vol. IIa, edited by G. T. Rado and H. Suhl (Academic Press, New York, 1965) p. 105.
67. J. M. Smith, “Molecular dynamics,” (Academic, New York, 1980).
68. V. E. Zakharov and A. B. Shabat, “Exact theory of two-dimensional self-focusing and one-dimensional self-modulation of waves in nonlinear media,” Zh. Eksp. Teor. Fiz. **61**, 118–134 (1971), [Sov. Phys. JETP **34**, 62 (1972)].
69. E. Beutler, in *Williams Hematology*, Vol. 2, edited by E. Beutler, M. A. Lichtman, B. W. Coller, and T. S. Kipps (McGraw-Hill, New York, 1994) 5th ed., Chap. 7, pp. 654–662.
70. R. Ballagh and C. Savage, “Bose-einstein condensation: from atomic physics to quantum fluids,” in *Proceedings of the 13th Physics Summer School*, edited by C. Savage and M. Das (World Scientific, Singapore, 2000) cond-mat/0008070.
71. W. Opechowski and R. Guccione, “Introduction to the theory of normal metals,” in *Magnetism*, Vol. IIa, edited by G. T. Rado and H. Suhl (Academic Press, New York, 1965) p. 105.
72. J. M. Smith, in *Molecular Dynamics*, edited by C. Brown (Academic, New York, 1980).
73. D. D. Lincoll, “Semigroups of recurrences,” in *High Speed Computer and Algorithm Organization*, Fast Computers No. 23, edited by D. J. Lipcoll, D. H. Lawrie, and A. H. Sameh (Academic Press, New York, 1977) 3rd ed., Part 3, pp. 179–183, a full INCOLLECTION entry.
74. A. V. Oaho, J. D. Ullman, and M. Yannakakis, “On notions of information transfer in VLSI circuits,” in *Proc. Fifteenth Annual ACM*, All ACM Conferences No. 17, edited by W. V. Oz and M. Yannakakis, ACM (Academic Press, Boston, 1983) pp. 133–139, a full INPROCEDINGS entry.
75. L. Manmaker, *The Definitive Computer Manual*, Chips-R-Us, Silicon Valley, silver ed. (1986), a full MANUAL entry.