Scientific Inquiry and Review (SIR)
Volume 10 Issue 2, 2026
ISSN(P): 2521-2427, ISSN(E): 2521-2435
Homepage: https://journals.umt.edu.pk/index.php/SIR

Title:	**Leveraging Grid Search in Tuning Ensemble Models to Predict Boiling Points of Alkanes**
Author (s):	Abdul Rauf Nizami[1], Sayyada Fiza Ali[1], and Muhammad Zia Afzal[2]
Affiliation (s):	[1]University of Central Punjab, Lahore, Pakistan [2]University of the Punjab, Lahore, Pakistan
Academic Editor	Abaid Ur Rehman Virk
DOI:	https://doi.org/10.32350/sir.102.04
History:	Received: March 02, 2026, Revised: April 13, 2026, Accepted: May 21, 2026, Published: June 25, 2026
Citation:	Nizami AR, Ali SF, Afzal MZ. Leveraging grid search in tuning ensemble models to predict boiling points of alkanes. *Sci Inq Rev*. 2026;10(2): 70–91. https://doi.org/10.32350/sir.102.04
Copyright:	© The Authors
Licensing:	This article is open access and is distributed under the terms of Creative Commons Attribution 4.0 International License
Conflict of Interest:	Author(s) declared no conflict of interest

A publication of
The School of Science
University of Management and Technology, Lahore, Pakistan

Leveraging Grid Search in Tuning Ensemble Models to Predict Boiling Points of Alkanes

Abdul Rauf Nizami[1*], Sayyada Fiza Ali[1], and Muhammad Zia Afzal[2]

[1]Department of Mathematics, University of Central Punjab, Lahore, Pakistan
[2]Department of Software Engineering, University of the Punjab, Lahore, Pakistan

ABSTRACT

Predicting the Normal Boiling Points (NBPs) of alkanes is a fundamental problem in physical organic chemistry and chemical engineering. In this article, two-stage GridSearchCV workflow was used to tune Random Forest (RF) and eXtreme Gradient Boosting (XGBoost) regressors in order to predict the NBPs of alkanes containing up to ten carbon atoms. Each alkane was encoded as a unique ten-digit numerical sequence. In Stage I, broad hyperparameter ranges were explored to identify regions of optimal model performance. In Stage II, a refined grid was constructed around the Stage I optima to pinpoint the best hyperparameter combinations with greater precision. Both models were evaluated on held-out test data using Mean Squared Error (MSE) and R^2 metrics. After two-stage tuning, XGBoost showed substantial improvement with R^2=0.987 and an MSE=12.002. While, the RF regressor showed a modest improvement with R^2=0.975 and an MSE=22.538 over Stage I baselines. SHapley Additive exPlanations (SHAP) analysis was subsequently applied to quantify the contribution of individual structural features to each model's predictions. Positional features corresponding to later carbon atoms were identified. The results demonstrated that systematic two-stage hyperparameter optimization *via* GridSearchCV significantly enhanced predictive accuracy and interpretability in chemical property prediction tasks.

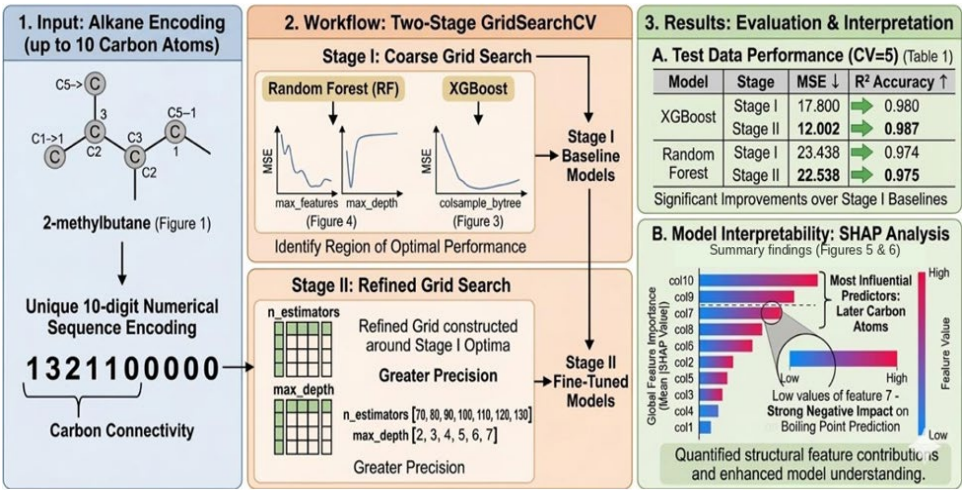
Keywords: alkanes, boiling points, GridSearch, machine learning, random forest, XGBoost

Highlights

- Two-stage grid search improved boiling point prediction for alkanes.
- XGBoost achieved better prediction accuracy than Random Forest (RF).
- SHAP analysis identified important molecular features affecting predictions.

*Corresponding Author: arnizami@ucp.edu.pk

GRAPHICAL ABSTRACT



1. INTRODUCTION

The prediction of Normal Boiling Points (NBPs) of alkanes occupies a central place in physical organic chemistry and chemical engineering [1-3]. Boiling points serve as fundamental thermodynamic descriptors underpinning the design and optimization of separation processes, petroleum refining, and the development of fuel formulations. As experimental determination of boiling points for every possible alkane isomer is expensive and time-consuming, accurate computational models offer a practical and cost-effective alternative. Reliable boiling point predictions contribute directly to Quantitative Structure–Property Relationship (QSPR) modelling [3].

The structure–boiling point relationships can be traced back to the landmark 1947 work of Harold Wiener. The path number, the Wiener index, the first graph-theoretic molecular descriptor was introduced. He demonstrated its correlation with the boiling points of paraffin hydrocarbons [1]. Subsequent decades witnessed an explosion of new molecular descriptors. Among the most studied are the Hosoya index [4], the Gutman index [5], and the Schultz molecular topological index [6]. All of these have been applied to the prediction of alkane boiling points and related thermodynamic quantities. Building on these descriptor-based approaches, backpropagation neural networks was applied [7] to predict the NBPs of acyclic ethers and related compounds. The non-linear models

could outperform contemporary linear QSPR models in both correlation coefficient and standard deviation. Goll further advanced the field by coupling multilinear regression with genetic algorithms and neural networks to predict boiling points for over one hundred organic compounds, achieving a root-mean-square error of 11.72 K with their best neural network model [8]. Espinosa developed neural-network-based QSPR models specifically for aliphatic hydrocarbons, reporting high accuracy across a dataset of 150 alkanes [9]. While, he extended this framework to broader classes of organic compounds using backpropagation architectures [10]. More recently, Liu employed Artificial Intelligence (AI) approaches to predict the NBPs of refrigerants. This further confirmed the broad applicability of machine learning (ML) in thermophysical property prediction [11]. Graph-based deep learning (DL) has also emerged as a powerful paradigm. A graph Convolutional Neural Network (CNN) was trained on over 3,800 organic compounds to predict NBPs [12]. Neural networks were combined with topological indices to model boiling points of refrigerants [13]. Furthermore, Nizami proposed novel graph-theoretic descriptors derived from distance and degree matrices. This demonstrated their competitive performance against high-dimensional representations to predict boiling point in alkanes [14].

Within the ML landscape, ensemble methods — most notably RF [15, 16] and eXtreme Gradient Boosting (XGBoost) [17, 16] — have emerged as particularly powerful tools for regression on structured chemical data. RF, applied to QSAR and cheminformatics modelling, builds a collection of independent decision trees (DTs) through bootstrap aggregation and random feature subsampling. It then averages their predictions to reduce variance and resist overfitting [18]. XGBoost adopts a sequential boosting strategy in which each new tree is trained to correct the residuals of its predecessors, combined with regularization penalties on leaf weights to guard against overfitting. The predictive power of both algorithms depends critically on their hyperparameters: parameters, such as the number of estimators, maximum tree depth, learning rate, subsampling fraction, and column-subsampling fraction interact in complex, non-linear ways that defy intuition-based tuning. GridSearchCV [19], the exhaustive hyperparameter search utility provided by the scikit-learn library, addresses this challenge by systematically evaluating every combination of candidate parameter values over a user-defined grid, selecting the combination that minimizes the cross-validated error. GridSearchCV embeds k-fold cross-validation

within the search. It optimizes hyperparameters and provides an unbiased estimate of generalization performance, making it an indispensable component of rigorous model development. The interpretability of ensemble models has been further enhanced by the TreeSHAP algorithm [20]. This provides consistent, individualized feature attributions for tree-based ensembles and complements the broader SHAP framework [21].

GridSearchCV, in tuning ensemble models, is known in multiple fields. For instance, Chaudhury et al. [22] demonstrated that combining hyperparameter tuning pipelines with GridSearchCV helps in breast cancer detection. This may highlight the method's versatility beyond regression tasks. Alshammari applied GridSearchCV to optimize artificial neural networks (ANNs) architectures for indoor temperature prediction in smart homes. This reported marked gains in predictive accuracy attributable directly to systematic grid-based tuning [23]. In the biomedical domain, optimizing RF classifiers with GridSearchCV has been shown to yield accuracy improvements of several percentage points compared with default-parameter baselines (cf. liver disease prediction literature). In the cheminformatics domain, Chen compared RF, gradient boosting, and related ensemble methods for QSPR tasks. This demonstrated that blended ensemble strategies consistently outperform single-model baselines. Interpretable feature attribution is essential to translate model outputs into chemical insight [24]. The hybrid ensemble learning was combined with QSPR to predict critical properties and boiling points of diverse organic compounds, achieving high predictive accuracy and confirming the practical value of systematic hyperparameter optimization [25]. Taken together, these results established GridSearchCV as a domain-agnostic optimization tool capable of extracting near-optimal performance from complex ensemble learners. The parameter grid was designed in a two-stage manner, *that is*, a coarse initial sweep to locate the region of optimal performance, followed by a refined search to pinpoint the best values.

Although a rich body of literature exists towards prediction purposes, yet a considerable number of questions need satisfactory answers. For instance, '*Is there a numerical way that better represents a molecule?*' and '*Can the prediction accuracy be improved with the same numerical representation?*' The current study addressed these questions. On the one hand, the study vectorized a molecule by recording degrees of carbon atoms from its molecular graph. On the other hand, a two-stage framework was

introduced to boost up the accuracy of ensemble models, RF and XGBoost, to predict the NBPs of alkanes. The framework showed a considerable improvement towards QSPR modeling.

The study is organized as follows: Section 2 covers the general methodology (dataset, molecular graph, molecule encoding); Section 3 and 4 are devoted to architectures, training, and tuning of XGBoost and RF; final reflections and SHAP analysis are covered in Sections 5 and 6; conclusive remarks are given in Section 7.

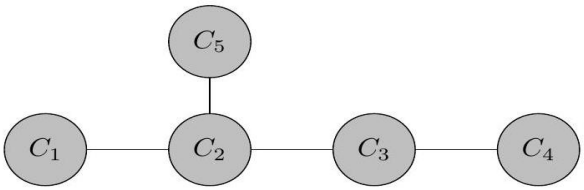
2. METHODOLOGY

2.1. Dataset

The dataset contained 150 laboratory measured NBPs of acyclic alkanes [2, 26] with up to 10 carbon atoms and was obtained from [1]. Alkanes are saturated hydrocarbons made up of carbon and hydrogen atoms, with only single bonds between the carbon atoms. The boiling point, measured in degrees Celsius (℃), is taken as the target property for prediction.

2.2. Encoding Alkanes

The 10-digit alkane encoding was generated using the RDKit library in Python. For each molecule, its SMILES representation was provided as input to RDKit, which assigned a consistent index to each atom in the molecular structure. The numbering started with the carbon atoms in the primary chain, and then moved to those in the branches. The degree of each carbon atom was then determined according to this atom ordering, and the resulting degree values were arranged to form the molecular encoding. For molecules containing fewer than 10 carbon atoms, zeros were added at the end of the sequence so that all molecules have the same vector length. An illustrative example is shown in Figure 1. Since RDKit indexing is unique for each molecule, no two different isomers receive the same code.



$C_1 \to 1,\ C_2 \to 3,\ C_3 \to 2,\ C_4 \to 1,\ C_5 \to 1$

Figure 1. Numerical Encoding of 2-methylbutane as a Ten-digit Sequence Derived from the Ordered Carbon-atom Degrees

3. ML MODELS

A two-stage framework was designed to predict the boiling points of alkanes with two ML models, that is, XGBoost and RF. For both models, 85% data was reserved for training and 15% data for test purposes; to ensure the reproducibility, we set random_state = 42. In Stage I, broad parameter ranges were explored to identify regions of good model performance. In Stage II, a refined grid was constructed around the Stage I optima to pinpoint the best hyperparameter combination with greater precision. In both stages, GridSearchCV, which itself covers the five-fold cross validation, was used to fine tune the models; the GridSearchCV. The final models were evaluated on held-out test data and interpreted using SHAP analysis [21] to quantify the contribution of individual structural features to each prediction.

3.1. XGBoost

XGBoost is a powerful, efficient, and scalable ML library for gradient boosting algorithms. It is widely used for structured and tabular data.

3.1.1. Model Architecture. The model architecture of XGBoost is based on the gradient boosting framework. However, it incorporates a variety of optimizations and enhancements that make it more efficient and powerful than traditional boosting methods. The main components are gradient boosting framework, additive tree model, optimization, and final prediction; see Figure 2.

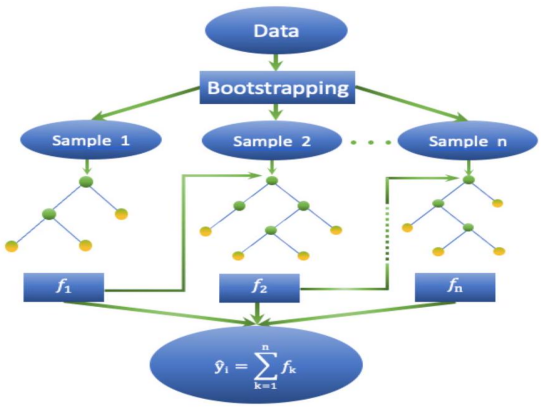


Figure 2. Schematic Architecture of the XGBoost model, Illustrating the Sequential Construction of Decision Trees (DTs) and The Additive Prediction Process

3.1.2. Gradient Boosting Framework. XGBoost is an ensemble algorithm that builds a series of weak learners, usually DTs, in sequence to form a highly accurate predictive model. In each step, a new tree is trained on the residuals of the prior trees, and the model is updated by minimizing the loss function through gradient information and a learning rate. Mathematically, it is represented as

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x)$$

where $F_m(x)$ is the model after m-th iteration, $F_{m-1}(x)$ is the previous model, η is the learning rate (shrinkage), and $h_m(x)$ is the new DT trained on the residuals.

3.1.3. Additive Tree Model. XGBoost builds an additive model of DTs. The overall prediction is the sum of all tree outputs:

$$\hat{y}_i = \sum_{k=1}^{n} f_k(x_i), f_k \in \mathcal{F}$$

where f_k is the prediction of the k-th DT, n is the total number of trees, and $\mathcal{F}$ is the space of DTs. Each tree is trained to minimize the loss function while adding to the previous trees' predictions.

3.1.3.1. Optimizations. Optimization consists of several parts: *Regularized Objective Function*: XGBoost minimizes the following objective function:

$$\mathcal{L} = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{n} \Omega(f_k)$$

Here ℓ is the loss function. The regularization function Ω that penalizes the tree complexity is:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Here T, w_j, γ are the number, weights, and penalty for the leaf nodes, and λ is the $L2$ regularization term. The regularization actually prevents overfitting and improves generalization.

Tree Pruning: Unlike traditional boosting methods, XGBoost uses max

depth pruning rather than greedy splitting. Trees are grown depth-first, and pruning is done after growing the tree to a maximum depth. This ensures computational efficiency.

Weighted Quantile Sketch: For DT splits, XGBoost uses an efficient histogram-based algorithm with a weighted quantile sketch for finding the best split point. This optimization significantly reduces computation time.

Column Subsampling: Similar to RFs, XGBoost allows feature (column) subsampling at each split or tree level. Overfitting is, therefore, reduced with improvement in computation efficiency.

Shrinkage (Learning Rate): A learning rate (η) is applied to scale the contribution of each tree, preventing overfitting.

Sparsity-Aware Split Finding: XGBoost handles missing values internally by learning the optimal path for missing data during training.

Parallelization: XGBoost parallelizes tree construction using multi-threading, enabling faster training.

Final Prediction: For a sample, the final prediction is the sum of outputs from all trees in the ensemble, scaled by the learning rate. For regression, it is

$$\hat{y} = \sum_{k=1}^{n} \eta \cdot f_k(x)$$

3.1.2. Model Parameters. XGBoost has a wide range of hyperparameters that can be tuned to enhance model performance. They can be grouped into three categories – general settings, parameters related to boosting, and those that control regularization.

3.1.2.1. General Parameters. Here is a list of general-purpose parameters used to control the behavior and functionality of the XGBoost model. There are three types of boosting models that can be assigned to booster parameter: gbtree for tree-based models, gblinear for linear models, and dart for dropout trees. The objective parameter can take on any one of the learning tasks, for instance, reg:squarederror for regression or binary:logistic for binary classification. To monitor model performance during training, the eval_metric parameter can be assigned an appropriate evaluation criterion, such as rmse, logloss, and mae. To process multiple

instructions simultaneously, one can assign nthread or n_jobs parameter. They control the number of CPU threads used by the XGBoost model to optimize computational efficiency. One can set different level of logging by adjusting the parameter verbosity: silent ('0') to detailed debugging ('3').

3.1.2.2. Tree-specific Parameters. To gain more control over the trees generated by the XGBoost model during the training loop, the model provides several parameters. Among the most important are: max_depth controls the tree depth (deep trees can capture more detail but are prone to overfitting); min_child_weight allows a node to split further depending on the weight sum (values close to zero make trees deeper to capture complex patterns in the data); gamma which specifies the minimum loss reduction required before allowing a split (values close to zero allow easier splits); subsample (the fraction of observations used per tree); colsample_bytree, colsample_bylevel, and colsample_bynode (fractions of features used at different tree levels); and learning_rate or eta controls the contribution of each tree to the final model (smaller values lead to slower learning and often better generalization).

3.1.2.3. Regularization Parameters. By growing the depth of the trees, learning becomes more conservative and often leads to overfitting problems. To control this behavior, regularization parameters prevent overfitting by penalizing complex trees to extend beyond a certain depth. The lambda parameter uses L2 regularization to the leaf weights to overcome on overfitting. Large values can reduce overfitting. The alpha parameter applies the L1 regularization to reduce the influence of less important features. The gamma parameter and other related parameters that are connected with tree growths have been discussed already in the "tree-specific parameters" paragraph. They can also be used as well to regularize the trees generated during the training session.

3.1.2.4. Dart-specific Parameters. When the booster parameter of XGBoost is set to dart, it introduces dropout methods to overcome the overfitting cases by using probability of dropping trees. The rate_drop parameter is one such parameter which controls the dropout rate, and dropping trees during training reduces the likelihood of relying on specific trees. The skip_drop parameter is the probability of skipping the dropout entirely during a boosting iteration, so it determines how often dropout is applied at all. The sample_type parameter can assume either "uniform" which means all previous trees have equal chance of being dropped, or

"weighted" which leads to the dropout of largely influencing trees. The normalize_type can be assigned either "tree" which normalizes trees on the basis of remaining trees, or "forest" which normalizes based on the entire forest. Thus, for complex dataset, dart parameter and the related parameters can be tuned to reduce overfitting and have a better performance from XGBoost.

3.1.3. Model Training. The split used for training was 85:15. This means that 85% of the data of the set of 150 alkanes was used for training, while the remaining 15% was reserved for validation of the model performance. To tune the model, n_estimator and eta can play a crucial role. The eta parameter acts as the learning rate and scales the contribution of each new tree added during the training rounds, thereby impacting the prediction process. Lower values of eta (e.g., 0.01−0.1) made the learning process slower and required more boosting rounds (trees) to converge. This often leads towards better generalization on validation dataset (unseen data). On the other hand, higher values of eta (e.g., 0.3−0.5) provided a better speed up during training but increased the risk of overfitting. When using a small eta, one has to increase the number of boosting rounds (n_estimator) to allow the model to converge properly.

The parameter subsample allows the fraction of the training data used to fit each individual tree. This randomness to the training process helps improve generalization and prevent overfitting. For instance, a subsample value of 0.85 means that 85% of the training data is randomly selected for each boosting iteration. Using a smaller subset of the data for training can add diversity to the model but may slightly slow down learning. While typical values range between 0.5 and 1.0, setting subsample too low can result in underfitting. As the model may not see enough data to learn effectively.

The min_child_weight parameter depends on the minimum amount of information (Hessian weight) present in a child node that dictates whether the node is allowed to split further into child nodes or not. Its default value is 1. Thus, values close to zero allow the trees to grow deeper to learn more complex patterns at the cost of overfitting problem.

The colsample_bytree parameter controls the fraction of features (columns) that are randomly selected and used for building each tree in the ensemble. This parameter is an important aspect of regularization in

XGBoost, helping to prevent overfitting and improve model robustness. For every tree in the ensemble, XGBoost randomly selects a subset of features, with the size of the subset determined by colsample_bytree. The selected subset of features is then used for splitting at every level of the tree. By default, it uses all the features for each tree.

3.1.3.1. Model Tuning. XGBoost is trained in two stages:

Stage I: In this stage, each parameter is plotted verses a wide range around the default values of other parameters to record the best average value in terms of the minimum MSE; of course, the best values are obtained with GridSearchCV. For instance, colsample_bytree is plotted in against eta, max_depth, n_estimators, and subsample to find its optimum value 0.5. Following the same pattern, the average values of min_child_weight, subsample, n_estimators eta, and max_depth are recorded, respectively 2, 0.8, 106, 3, and 0.0837. Training and (five-fold) test errors using optimal values are 95.33 and 17.8, respectively.

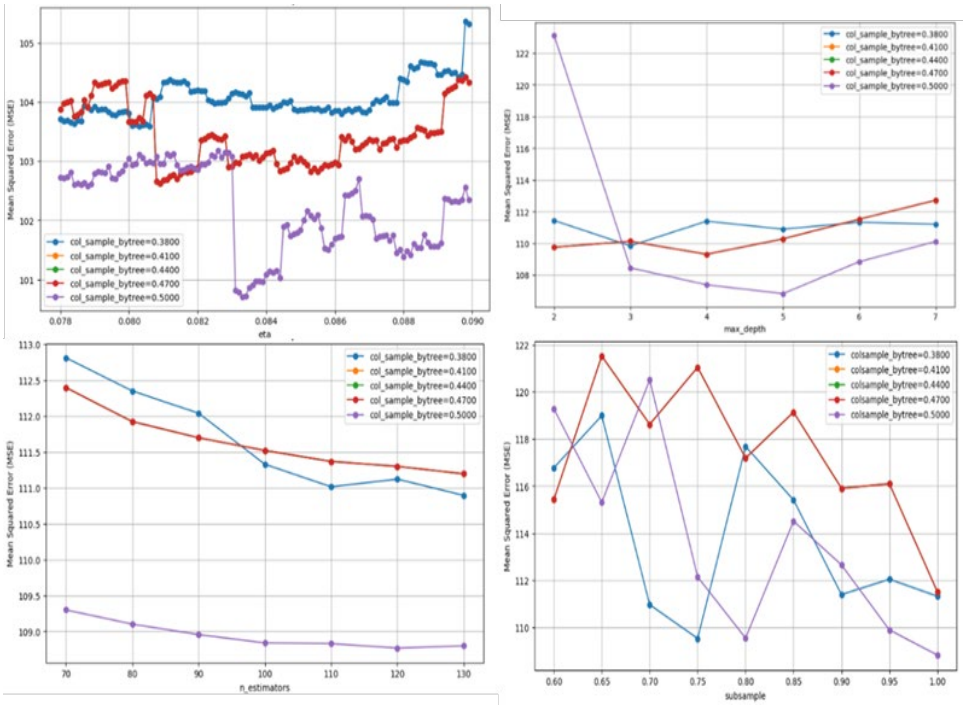


Figure 3. Effect of Colsample_bytree across Plausible Values of Eta, max_depth, Subsample, and n_estimators on the Mean Squared Error of the XGBoost Model

Stage II: At this stage, GridSearchCV is used to further optimize the tuning parameters obtained at Stage I. The optimum values produced are n_estimators=110, min_child_weight=2, colsample_bytree=0.52, subsample=0.75, and eta=0.09. The resulting training error is 89.293 and test error is 12.002 (CV=5), which is a considerable improvement.

3.2. Random Forest (RF) Regressor

This algorithm, known as the RF Regressor, performs regression by generating a collection of DTs and averaging their results to improve prediction accuracy and robustness.

3.2.1. Model Architecture. The structure of a RF Regressor resembles that of XGBoost but it uniquely employs bootstrap aggregation. However, construction of each tree is independent of all trees by selecting a random set of features with replacement from the training dataset.

3.2.2. Model Parameters. Model's performance can be enhanced by tuning the following hyperparameters explained in paragraphs below.

3.2.2.1. n-estimator. The n_estimator parameter controls the number of DTs constructed by RF model to make decisions (predictions). It can be assigned any value in the range of 10 to 1000. Default value is 100. Values close to 10 would make the forest shallow and mostly have the risk of underfitting. Higher values would lower the variance and, hence more reliable predictions. This, however, increases the computational cost.

3.2.2.2. Max Depth. The max_depth parameter places an upper limit on the depth of each tree constructed. It controls the complexity of the model. The value of this parameter can range from 3 to 50. It defaults to 6. Lower values make the training fast but can result in underfitting. Values greater than 20 may produce good results on the training set but may not perform as well on the validation dataset which is termed as overfitting.

3.2.2.3. Max Features. The parameter max_features applies a limit on the selection of features during each split. Smaller values increase the randomness, reduce the correlation among trees, and enhance generalization. Higher values though make the trees stronger in prediction but increase the risk of overfitting. It can be assigned "SQRT" (square root of the total number of features to be considered), "log2", "None" (all features), an integer (these many features), or a float (fraction of features to consider).

3.2.2.4. Min Sample Split. The parameter min_samples_split controls the minimum number of samples required to split an internal node. It can either be an integer or a float (fraction of total number of samples). Larger values make the trees shallow and less computationally complex but have the risk of underfitting. Small values have the reverse effect.

3.2.2.5. Max Sample. The max_sample parameter is used if bootstrap=True, and it controls the fraction or number of samples drawn to train each tree in the ensemble. All the samples are used by default to train each tree in the ensemble.

3.2.2.6. Min Sample Leaf. The min_sample_leaf can also have integer or float values. It determines the minimum count of samples required in a leaf node. This also behaves the same way as other similar parameters, and larger values produce shallow trees.

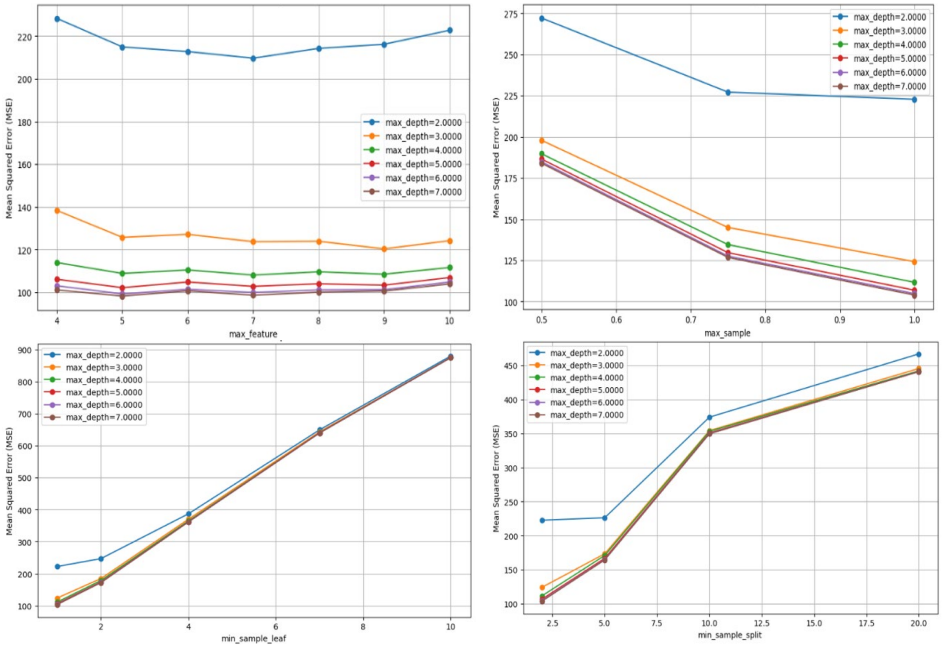


Figure 4. Effect of Max_depth across Plausible Values of Max_features, Max_sample, Min_samples_leaf, and Min_samples_split on the Performance of the RF Regressor

3.2.3. Model Training. The model is now trained for the dataset of 150 isomeric alkanes, reserving 85% data for training and 15% for validation. This is tuned up with n_estimator, max_depth, max_features,

min_samples_split, max_sample, and min_sample_leaf in two stages.

Stage I: Here, the average values of n_estimator, max_depth, max_features, min_samples_split, max_sample, and min_sample_leaf are $102, 6, 5, 2, 1.0$, and 1, respectively; see Figure 4. The training and validation datasets exhibit errors of 98.645 and 23.438, respectively. Moreover, a 5 -fold cross-validation strategy was employed to assess the training data, thereby ensuring the model's performance reliability.

Moreover, it was observed that min_sample_split, max_sample, and min_sample_leaf produced the best result for their default values. With regard to max_depth, the error reduced as its value increased and became optimum at max_depth $= 7$.

Stage II: At this stage, the model of Stage I is improved with GridSearchCV algorithm. The lists for n_estimator, max_depth, max_features, min_samples_split, max_sample, and min_sample_leaf are respectively $[70,80,90, 100,110,120,130]$, $[2,3,4,5,6,7]$, $[4,5,6,7,8,9,10]$, $[2,3,4,5]$, $[0.5,0.75,1.0]$, and $[1,2,3,4]$. The optimum values of the parameters reflected by GridSearchCV are 110, $7, 5, 2, 1.0$, and 1. The minimum cross-validated error on the training dataset is 96.318, obtained with $\mathrm{CV} = 5$. Using these parameters, the model is trained, achieving a minimum validation error of 22.538.

3.3. Predictions

For both models, GridSearchCV determined optimal values of 110 for n_estimators and 7 for max_depth. max_features was set to 5 in the RF Regressor, while eta was set to 0.09 in XGBoost. The MSE and R^2 accuracy for both models (CV=5) on test data are shown in Table 1. (The interested reader can see that the present results are better than the results obtained in previous work [14].

Table 1. Test-set Performance of the Random Forest (RF) and XGBoost Models Before and After Two-stage Hyperparameter Tuning

Model	Stage	MSE	R^2
Random Forest (RF)	I	23.438	0.974
	II	22.538	0.975
XGBoost	I	17.800	0.980
	II	12.002	0.987

4. SHAP ANALYSIS

SHAP is an effective tool for decoding ML models by assessing each feature's impact on specific predictions. SHAP analysis produces detailed information about the importance of each feature in prediction of the respective property. Summary plots or force plots provide all such information about the importance of individual's contribution in a visual way. On the basis of this analysis, less contributors can be eliminated or it may lead towards engineer/focus on the important features and related things. For the RF Regressor, features col10, col7, col9, col8, and col6 have a significant impact. Whereas, features col4 and col1 have a lesser impact on the predictions; refer to Figure 5. In a similar fashion, the contributions of col9, col7, col10, and col6 are more substantial in XGBoost, with col4 and col1 having a smaller effect; see Figure 6. Notably, features 10, 9, and 7 are predominantly influential in both models, and there are certain reasons behind them. First, most of the molecules in the dataset are the isomers of the alkanes with 7, 8, 9, and 10 carbons. Hence, the positions of the feature vectors corresponding to these indexes are nonzero. Second, splitting of nodes at these features give maximum gain with respect to both the ensemble algorithms.

For specific data points, notably low values of feature 7 are positioned markedly to the left. This suggests that the model attributes a strong negative value to these for predicting the boiling point.

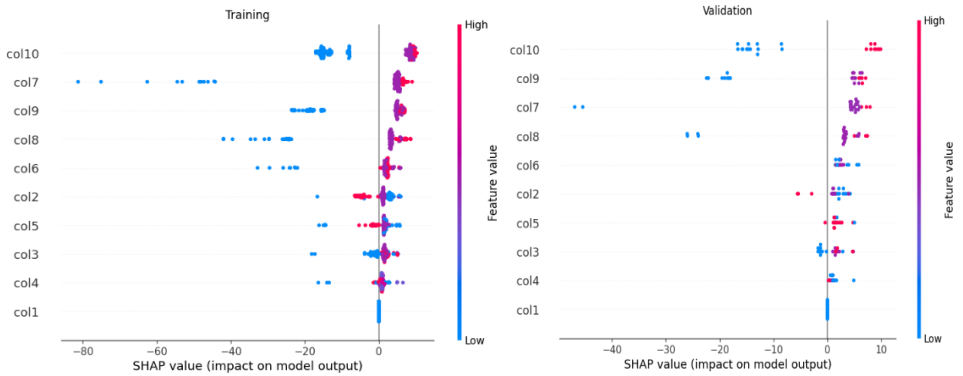


Figure 5. SHAP Feature-importance Summary for the Random Forest (RF) Regressor, Showing the Contribution of the Molecular Encoding Features to Boiling Point Predictions

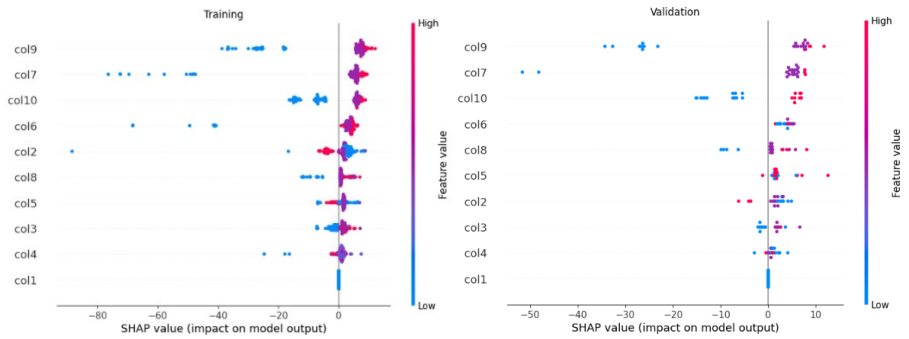


Figure 6. SHAP Feature-importance Summary for the XGBoost Model, Showing the Contribution of the Molecular Encoding Features to Boiling Point Predictions

5. CONCLUSION

The current study employed GridSearchCV to fine-tune the hyperparameters of RF and XGBoost models in order to predict the boiling points of alkanes. Hyperparameter tuning was done in two stages. Initially, the parameters were assigned broader ranges to discover the key parameter values where each model is assuming a better performance. Then, a more refined grid was structured around the values found in the first stage to fine tune the models. This ensured that the best parameters regarding the problem under consideration were identified. Of course, overfitting and related issues were handled appropriately.

SHAP analysis was performed for both models. The main idea here is to just show how this analysis can provide insight into the model's best features. Though indication of which column performed better might not seem extracting a useful information. However, it can be thought as how many of them are really contributing into the prediction at the moment. So, the analysis can be multi-purpose and can be exploited in a variety of ways to derive the ML algorithms to perform in an optimal way. Thus, deeper understanding of models can be enhanced on the information gains from the SHAP analysis.

Overall, the approach used in the study demonstrated the effectiveness of combining systematic hyperparameter optimization with GridSearchCV. It enabled to enhance predictive accuracy and interpretability in chemical

property prediction tasks when combined with SHAP analysis. Future work may be extended to larger datasets along with other chemical property predictions. Furthermore, the accuracy may be with benchmarks.

Author Contribution

Abdul Rauf Nizami: conceptualization, supervision, writing - review & editing. **Sayyada Fiza Ali:** formal analysis, investigation, data curation, writing – original draft. **Muhammad Zia Afzal:** validation, visualization, writing – original draft.

Conflict of Interest

The authors of the manuscript have no financial or non-financial conflict of interest in the subject matter or materials discussed in this manuscript.

Data Availability Statement

Data supporting the findings of this study will be made available by the corresponding author upon request. It is also available at https://doi.org/10.1021/ja01193a005.

Funding Details

No funding has been received for this research.

Generative AI Disclosure Statement

ChatGPT and Claude AI were used to improve the readability and language of the manuscript.

REFERENCES

1. Wiener H. Structural determination of paraffin boiling points. *J Am Chem Soc.* 1947;69(1):17–20. https://doi.org/10.1021/ja01193a005
2. Wessel MD, Jurs PC. Prediction of normal boiling points of hydrocarbons from molecular structure. *J Chem Inf Comput Sci.* 1995;35(1):68–76. https://doi.org/10.1021/ci00023a010
3. Katritzky AR, Lobanov VS, Karelson M. QSPR: The correlation and quantitative prediction of chemical and physical properties from structure. *Chem Soc Rev.* 1995;24(4):279–287. https://doi.org/10.1039/CS9952400279
4. Hosoya H. Topological index. A newly proposed quantity characterizing the topological nature of structural isomers of saturated hydrocarbons. *Bull Chem Soc Jpn.* 1971;44(9):2332–2339. https://doi.org/10.1246/bcsj.44.2332
5. Gutman I. Selected properties of the Schultz molecular topological index. *J Chem Inf Comput Sci.* 1994;34(5):1087–1089. https://doi.org/10.1021/ci00021a009

6. Schultz HP. Topological organic chemistry. 1. Graph theory and topological indices of alkanes. *J Chem Inf Comput Sci.* 1989;29(3):227–228. https://doi.org/10.1021/ci00063a012

7. Cherqaoui D, Villemin D, Mesbah A, Cense JM, Kvasnicka V. Use of a neural network to determine the normal boiling points of acyclic ethers, peroxides, acetals and their sulfur analogues. *J Chem Soc Faraday Trans.* 1994;90(14):2015–2019. https://doi.org/10.1039/FT9949002015

8. Goll ES, Jurs PC. Prediction of the normal boiling points of organic compounds from molecular structures with a computational neural network model. *J Chem Inf Comput Sci.* 1999;39(6):974–983. https://doi.org/10.1021/ci9900711

   Goll and Jurs present one of the earliest systematic benchmarks of neural network models for predicting normal boiling points of structurally diverse organic compounds. Using descriptors derived from molecular topology and geometry, and combining multilinear regression with genetic-algorithm-selected descriptors and neural networks, they achieve a root-mean-square error of 11.72 K on a test set of 91 compounds. This work establishes a quantitative performance baseline against which the ensemble models of the present study can be compared, and it highlights the trajectory from single-architecture neural networks toward the more flexible, regularized tree-based ensembles employed here.

9. Espinosa G, Yaffe D, Cohen Y, Arenas A, Giralt F. Neural network based quantitative structural property relations (QSPRs) for predicting boiling points of aliphatic hydrocarbons. *J Chem Inf Comput Sci.* 2000;40(3):859–879. https://doi.org/10.1021/ci000442u

10. Espinosa G, Arenas A, Giralt F. Prediction of boiling points of organic compounds from molecular descriptors by using backpropagation neural network. In: Carbó-Dorca R, Gironés X, Mezey PG, eds. *Fundamentals of Molecular Similarity.* Kluwer Academic/Plenum Publishers; 2001:1–10. https://doi.org/10.1007/978-1-4757-3273-3_1

11. Liu B, Karimi Nouroddin M. Application of artificial intelligent approach to predict the normal boiling point of refrigerants. *Int J Chem Eng.* 2023;2023:e6809569. https://doi.org/10.1155/2023/6809569

12. Qu C, Kearsley AJ, Schneider BI, Keyrouz W, Allison TC. Graph convolutional neural network applied to the prediction of normal boiling point. *J Mol Graph Model.* 2022;112:e108149. https://doi.org/10.1016/j.jmgm.2022.108149

13. Deng S, Su W, Zhao L. A neural network for predicting normal boiling point of pure refrigerants using molecular groups and a topological index. *Int J Refrig.* 2016;63:63–71. https://doi.org/10.1016/j.ijrefrig.2015.10.025

14. Nizami AR, Ali SF, Afzal MZ, Inc M. Novel descriptors for the prediction of molecular properties. *Open Chem.* 2025;23(1):e20250194. https://doi.org/10.1515/chem-2025-0194
This paper by the present article's corresponding author introduces a family of novel graph-theoretic molecular descriptors derived from weighted distance and degree matrices of molecular graphs. The descriptors are evaluated for their ability to predict boiling points and other physicochemical properties of alkanes, and they are shown to achieve competitive or superior correlation compared with established topological indices. The work provides the theoretical and descriptive foundation for the structural features used in the present study, making it the primary methodological predecessor to the machine learning modelling reported here.

15. Breiman L. Random forests. *Mach Learn.* 2001;45(1):5–32. https://doi.org/10.1023/A:1010933404324

16. Friedman JH. Greedy function approximation: A gradient boosting machine. *Ann Stat.* 2001;29(5):1189–1232. https://doi.org/10.1214/aos/1013203451

17. 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.* Association for Computing Machinery; 2016:785–794. https://doi.org/10.1145/2939672.2939785

18. Svetnik V, Liaw A, Tong C, Culberson JC, Sheridan RP, Feuston BP. Random forest: A classification and regression tool for compound classification and QSAR modeling. *J Chem Inf Comput Sci.* 2003;43(6):1947–1958. https://doi.org/10.1021/ci034160g

19. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine

learning in Python. *J Mach Learn Res.* 2011;12:2825–2830.

20. Lundberg SM, Erion GG, Lee SI. Consistent individualized feature attribution for tree ensembles. *arXiv.* Preprint posted February 12, 2018. arXiv:1802.03888. https://doi.org/10.48550/arXiv.1802.03888

21. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems 30.* 2017:4765–4774.

22. Chaudhury S, Shelke N, Rashid ZM, Sau K. Effect of grid search and hyper parameter tuned pipeline with various classifiers and PCA for breast cancer detection. *Curr Signal Transduct Ther.* 2022;17(3):e150722206811. https://doi.org/10.2174/1574362417666220715105527

    This paper evaluates the combined effect of PCA-based dimensionality reduction and GridSearchCV hyperparameter tuning across multiple classifiers for breast cancer detection. The authors demonstrate consistent and statistically significant performance gains attributable specifically to the grid-search step, isolating its contribution from feature engineering effects. This finding corroborates the methodological choice in the present study of using GridSearchCV as the primary optimisation tool, and it extends the evidence base for grid-search-driven tuning to high-dimensional biomedical datasets with class-imbalance challenges.

23. Alshammari T. Using artificial neural networks with GridSearchCV for predicting indoor temperature in a smart home. *Eng Technol Appl Sci Res.* 2024;14(2):13437–13443. https://doi.org/10.48084/etasr.7008

    This study demonstrates the effectiveness of GridSearchCV in optimising artificial neural network hyperparameters for regression tasks in smart building systems. By systematically searching across network architecture and training parameters, the author achieved significantly lower prediction errors for indoor temperature than baseline models. The work is directly relevant to the present article as a practical application of the same two-stage grid-search strategy applied here to ensemble regressors, and it underscores the domain-agnostic utility of GridSearchCV beyond classification contexts.

24. Chen CH, Tanaka K, Kotera M, Funatsu K. Comparison and

improvement of the predictability and interpretability with ensemble learning models in QSPR applications. *J Cheminform.* 2020;12:e19. https://doi.org/10.1186/s13321-020-0417-9

25. Bounaceur R, Paes F, Privat R, Jaubert JN. AI-powered prediction of critical properties and boiling points: A hybrid ensemble learning and QSPR approach. *J Cheminform.* 2025;17:e132. https://doi.org/10.1186/s13321-025-01062-9
Bounaceur et al. introduce a hybrid framework that integrates QSPR descriptors with stacked ensemble learning to predict normal boiling points and critical thermodynamic properties across chemically diverse datasets. The study systematically benchmarks gradient boosting, Random Forest, and blended meta-learners, demonstrating that ensemble stacking with optimised hyperparameters outperforms single-model and classical QSPR approaches. Its findings are closely aligned with the goals of the present article, providing contemporary evidence that structured hyperparameter optimisation of tree-based ensembles represents the current state of the art in computational thermophysical property prediction.

26. Mukwembi S, Nyabadza F. A new model for predicting boiling points of alkanes. *Sci Rep.* 2021;11:e24261. https://doi.org/10.1038/s41598-021-03541-z