Predicting Glass Transition Temperature of Chemical Compounds Using SMILE Descriptors and Machine Learning Models

Sunny Kaushik
IMT2021007

Rohit Mogli
IMT2021503

***Abstract*—The accurate prediction of the glass transition temperature (Tg) of chemical compounds is crucial for understanding their behavior in various applications. This study explores the application of machine learning models, including Linear Regression, Random Forest, Gradient Boosting, XGBoost, and Extra Trees, to predict Tg values using both functional group counts and SMILES[1] (Simplified Molecular Input Line Entry System) descriptors. We introduced new features, such as the number of sulfur atoms and branching, to enhance prediction accuracy. Our results demonstrate that the Extra Trees model, utilizing the comprehensive feature set, achieves superior performance with the lowest mean absolute error (MAE) and high R^2 score, suggesting its effectiveness in Tg prediction. This research highlights the importance of feature selection and the potential of ensemble methods in improving predictive models for glass transition temperatures.**

I. Introduction

Glass transition temperature (Tg) is a critical property that determines the behavior of materials as they transition from a rigid glassy state to a more flexible state. Accurate prediction of Tg is essential for a wide range of applications, from material science to atmospheric studies. Despite its importance, the experimental determination of Tg for a vast array of chemical compounds remains limited due to resource-intensive processes.

Recent advancements in machine learning provide promising alternatives for Tg prediction. Studies by Alzghoul et al. (2014) and Tao et al. (2019) have demonstrated the potential of support vector machines and random forest models for Tg prediction, albeit on limited datasets. Building on this foundation, our study leverages a larger and more diverse dataset, incorporating innovative features such as SMILES descriptors and branching information to enhance model performance.

The Simplified Molecular Input Line Entry System (SMILES) is a notation that allows a user to represent a chemical structure in a way that can be used by a computer program. SMILES strings encode molecular structures using short ASCII strings, making them compact and easy to manipulate computationally. This representation includes information about atoms, bonds, connectivity, and chirality, enabling the generation of detailed molecular descriptors.

Descriptors derived from SMILES strings capture various structural, topological, and electronic properties of molecules. In this study, we utilized the RDKit library to transform SMILES strings into a comprehensive set of 208-bit descriptors. These descriptors provide a rich source of information about the molecular properties and are instrumental in improving the predictive power of machine learning models.

We employed several machine learning models, including Linear Regression, Random Forest, Gradient Boosting, XGBoost, and Extra Trees, to predict Tg. Additionally, we explored the impact of newly introduced features, such as the number of sulfur atoms and the degree of branching, on model accuracy. Our methodology involved extensive data preprocessing, feature engineering, and model evaluation using robust metrics like mean absolute error (MAE), root mean squared error (RMSE), and the coefficient of determination (R^2).

This paper presents a comprehensive analysis of the models' performance, highlighting the superior accuracy achieved by the Extra Trees model when utilizing the enriched feature set. Our findings underscore the significance of feature selection and ensemble learning techniques in developing reliable predictive models for Tg, paving the way for future research and practical applications in this domain.

II. Dataset

The dataset utilized in this study is sourced from the Bielefeld Molecular Organic Glasses (BIMOG) database[1], which is a comprehensive collection of experimental data on glass transition temperatures. The BIMOG database was established to support research in material science by providing reliable and accessible data on the glass transition temperatures of various organic compounds. The dataset includes a wide range of chemical compounds, each with detailed molecular descriptors that are crucial for accurate Tg prediction.

A. BIMOG Database

The Bielefeld Molecular Organic Glasses (BIMOG) database[1] was formed as part of a collaborative effort to compile extensive experimental data on the glass transition temperatures of organic molecular compounds. This initiative aimed to address the scarcity of available data by systematically collecting and curating Tg measurements from diverse sources. The BIMOG database, established around 2012, has

since become a valuable resource for researchers in material science and related fields. It provides a robust foundation for developing and validating predictive models for Tg.

The BIMOG database includes data on various molecular properties, such as:

- Molecular Weight (M / g/mol)
- Number of Methyl Groups (#CH3)
- Number of Methylene Groups (#CH2)
- Number of Methine Groups (#CH)
- Number of Carbon Atoms (#C)
- Number of Hydroxyl Groups (#OH)
- Number of Ether Linkages (#C-O-C)
- Number of Carbonyl Groups (#O=C)
- Number of Double Bond Equivalents (DBA)
- Number of Nitrogen Atoms (#N)
- Number of Halogen Atoms (#Hal)
- Oxygen to Carbon Ratio (O:C)
- Number of Sulfur Atoms (#S)
- Degree of Branching (#branching)

In addition to these features, the dataset also includes the glass transition temperature (Tg) and melting temperature (Tm) for each compound. These descriptors, derived from both functional group counts and SMILES representations, provide a comprehensive set of inputs for machine learning models.

B. Feature Descriptions

The features in the dataset are detailed as follows:

- **M / g/mol**: Molecular weight of the compound.
- **#CH3**: Count of methyl groups in the compound.
- **#CH2**: Count of methylene groups in the compound.
- **#CH**: Count of methine groups in the compound.
- **#C**: Total number of carbon atoms in the compound.
- **#OH**: Count of hydroxyl groups in the compound.
- **#C-O-C**: Count of ether linkages in the compound.
- **#O=C**: Count of carbonyl groups in the compound.
- **DBA**: Number of double bond equivalents in the compound.
- **#N**: Count of nitrogen atoms in the compound.
- **#Hal**: Count of halogen atoms in the compound.
- **O:C**: Oxygen to carbon ratio in the compound.
- **#S**: Count of sulfur atoms in the compound.
- **#branching**: Degree of branching in the compound.
- **Tg / K**: Glass transition temperature of the compound (in Kelvin).
- **Tm / K**: Melting temperature of the compound (in Kelvin).

III. Methodology

A. Machine Learning Methods

In this section, we will provide a detailed explanation of the machine learning models used in this study. Each model will be described in a separate subsection to provide clarity and depth of understanding.

1) Decision Trees: A decision tree is a flowchart-like structure used for decision-making and predictive modeling. It consists of nodes representing decisions or tests on features, branches representing the outcomes of those decisions, and leaf nodes representing final predictions or outcomes. The key components of a decision tree include:

- **Root Node:** The topmost node in the tree that represents the initial feature test.
- **Decision Nodes:** Intermediate nodes that represent tests on features.
- **Leaf Nodes:** Terminal nodes that provide the final prediction or outcome.

The decision tree algorithm works as follows:

1) **Select Best Feature:** At each node, the algorithm selects the best feature to split the data based on a criterion such as Gini impurity or information gain.
2) **Split Data:** The data is split into subsets based on the selected feature.
3) **Repeat:** This process is repeated recursively for each subset until a stopping condition is met (e.g., maximum depth, minimum samples per leaf).

The Gini impurity for a node t is calculated as:

$$G(t) = 1 - \sum_{i=1}^{n} p_i^2$$

where p_i is the proportion of samples belonging to class i.

2) Random Forest: Random Forest is an ensemble learning method that combines multiple decision trees to improve predictive performance and reduce overfitting. Each tree in the forest is trained on a random subset of the data with replacement (bootstrap sampling), and a random subset of features is considered for each split.

Key steps in the Random Forest algorithm:

1) **Bootstrap Sampling:** Randomly sample the data with replacement to create multiple subsets.
2) **Train Trees:** Train a decision tree on each subset using a random subset of features for splitting.
3) **Aggregate Predictions:** Combine the predictions of all trees (e.g., by averaging for regression or majority voting for classification).

The final prediction $\hat{y}$ for regression is the average of individual tree predictions:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} \hat{y}_t$$

where T is the number of trees and $\hat{y}_t$ is the prediction from tree t.

3) Extra Trees: Extra Trees, or Extremely Randomized Trees, is similar to Random Forest but introduces more randomness in the tree-building process. Instead of selecting the best split based on a criterion, Extra Trees selects random split points for each feature.

Key differences from Random Forest:

- **Random Splits:** Split points are selected randomly rather than based on Gini impurity or information gain.
- **No Bootstrapping:** Extra Trees use the entire dataset to train each tree instead of bootstrap samples.

The algorithm increases model diversity and reduces overfitting by introducing additional randomness.

4) Linear Regression: Linear Regression is a simple and widely used statistical method for modeling the relationship between a dependent variable y and one or more independent variables X. The relationship is modeled as a linear equation:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

where:

- y is the dependent variable.
- $x_1, x_2, \ldots, x_n$ are the independent variables.
- β_0 is the intercept.
- $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients.
- ϵ is the error term.

The objective is to minimize the sum of squared residuals:

$$\min \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

where $\hat{y}_i$ is the predicted value for observation i.

5) Gradient Boosting: Gradient Boosting is an ensemble technique that builds models sequentially, where each new model corrects the errors of the previous models. It combines weak learners (typically decision trees) to form a strong learner.

Key steps in the Gradient Boosting algorithm:

1) **Initialize Model:** Start with an initial model (e.g., the mean of the target values for regression).
2) **Compute Residuals:** Calculate the residuals (errors) from the current model.
3) **Train New Model:** Train a new decision tree to predict the residuals.
4) **Update Model:** Add the new tree to the existing model with a learning rate η:

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

where $F_m(x)$ is the updated model, $F_{m-1}(x)$ is the previous model, and $h_m(x)$ is the new tree.

6) Extreme Gradient Boosting (XGBoost): Extreme Gradient Boosting (XGBoost) is an optimized version of Gradient Boosting that includes several enhancements to improve speed and performance. Key features of XGBoost include:

- **Regularization:** Adds $L1$ and $L2$ regularization terms to the loss function to prevent overfitting.
- **Sparsity Awareness:** Efficient handling of sparse data.
- **Parallel Processing:** Utilizes parallel computing to speed up training.

The objective function in XGBoost includes a regularization term:

$$\mathcal{L}(\Theta) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(h_k)$$

where $\Omega(h_k)$ is the regularization term for tree k.

XGBoost builds trees sequentially, with each tree improving the residual errors of the previous trees, similar to Gradient Boosting but with additional optimizations.

7) Grid Search CV Mechanism: Grid Search with Cross-Validation (Grid Search CV) is a technique used to find the optimal hyperparameters for a machine learning model. It systematically works through multiple combinations of parameter values, cross-validating as it goes to determine which set of parameters provides the best performance.

Key steps in Grid Search CV:

1) **Define Parameter Grid:** Specify a range of values for each hyperparameter you want to optimize. For example, for a Random Forest model, you might vary the number of trees (`n_estimators`), the maximum depth of the trees (`max_depth`), and the number of features considered for splitting (`max_features`).
2) **Cross-Validation:** Split the training data into k subsets (folds). Train the model on $k-1$ folds and validate it on the remaining fold. Repeat this process k times, each time with a different validation fold.
3) **Evaluation:** For each combination of hyperparameters, compute the average performance metric (e.g., accuracy, MAE) over the k folds.
4) **Select Best Parameters:** Choose the hyperparameter combination that results in the best average performance metric.

Formally, the goal is to minimize the cross-validation error:

$$\min_{\theta \in \Theta} \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}(y^{(i)}, \hat{y}^{(i)})$$

where Θ is the set of all hyperparameter combinations, $\mathcal{L}$ is the loss function, $y^{(i)}$ are the true values, and $\hat{y}^{(i)}$ are the predicted values for the i-th fold.

Grid Search CV helps in finding the most effective model configuration, ensuring that the model is neither underfitting nor overfitting the data.

8) Bagging and Boosting: Bagging (Bootstrap Aggregating) and Boosting are two fundamental ensemble learning techniques that improve model performance by combining the predictions of multiple base models.

Bagging: Bagging aims to reduce variance and prevent overfitting by training multiple models on different random subsets of the training data (created using bootstrapping). The final prediction is typically obtained by averaging the predictions (for regression) or voting (for classification).

Key steps in Bagging:

1) **Bootstrap Sampling:** Create multiple random samples of the training data with replacement.
2) **Train Models:** Train a separate model on each bootstrap sample.
3) **Aggregate Predictions:** Combine the predictions of all models. For regression, this is usually done by averaging; for classification, by majority voting.

Mathematically, the final prediction $\hat{y}$ for regression is:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} \hat{y}_t$$

where T is the number of models and $\hat{y}_t$ is the prediction from the t-th model.

Boosting: Boosting focuses on reducing bias and improving model performance by sequentially training models, where each new model attempts to correct the errors of the previous models. The models are trained on weighted versions of the training data, emphasizing the samples that were previously mispredicted.

Key steps in Boosting:

1) **Initialize Model:** Start with an initial base model (e.g., a decision tree).
2) **Compute Residuals:** Calculate the residuals (errors) from the current model.
3) **Train New Model:** Train a new model to predict the residuals.
4) **Update Model:** Add the new model to the existing ensemble with a weight. This process is repeated for a predefined number of iterations.

Mathematically, the updated model $F_m(x)$ is:

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

where η is the learning rate, $h_m(x)$ is the new model, and $F_{m-1}(x)$ is the previous model.

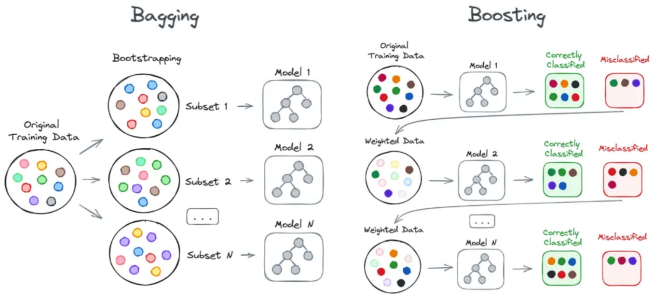

Fig. 1: Bagging and Boosting

Difference between Gradient Boosting and XGBoost:

- **Gradient Boosting:** Combines weak learners, typically decision trees, by optimizing a differentiable loss function in a sequential manner. Each tree attempts to correct the errors of the previous tree.
- **XGBoost:** An enhanced version of Gradient Boosting with additional features such as regularization to prevent overfitting, parallel processing for faster computation, and efficient handling of sparse data.

XGBoost's objective function includes a regularization term to control model complexity:

$$\mathcal{L}(\Theta) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(h_k)$$

where $\Omega(h_k)$ is the regularization term for tree k.

B. SMILE descriptors

In this section, we will discuss the significance of SMILES (Simplified Molecular Input Line Entry System) descriptors and their importance in computing RDKit descriptors. In the previous section, we examined various machine-learning methods. This method can be utilized to predict the glass transition temperature by using descriptors generated from the SMILES representation.

SMILES: SMILES, which stands for Simplified Molecular Input Line Entry System, is a notation that allows a user to represent a chemical structure in a way that can be easily used by computer programs. It uses a line of text to describe the structure of molecules in a manner that is both human-readable and machine-processable

Key rules for the computation of SMILES:

1) **Rule One: Atoms and Bonds**
   SMILES supports all elements in the periodic table. An atom is represented using its respective atomic symbol. Upper case letters refer to non-aromatic atoms; lower case letters refer to aromatic atoms. If the atomic symbol has more than one letter the second letter must be lowercase.
   Bonds are denoted as shown below:
   - - Single bond
   - = Double bond
   - # Triple bond
   - * Aromatic bond
   - . Disconnected structures

   Single bonds are the default and therefore need not be entered. For example, 'CC' would mean that there is a non-aromatic carbon attached to another non-aromatic carbon by a single bond, and the computer would identify the structure as the chemical ethane. It is also assumed that the bond between two lowercase atom symbols is aromatic. A blank terminates the SMILES string.
2) **Rule Two: Simple Chains** By combining atomic symbols and bond symbols simple chain structures can be represented. The structures that are entered using SMILES are hydrogen-suppressed, that is to say, that the molecules are represented without hydrogens. The SMILES software understands the number of possible connections that an atom can have. If enough bonds are not identified by the user through SMILES notation, the system will automatically assume that the other connections are satisfied by hydrogen bonds. **Some Examples:**

CC	CH_3CH_3	Ethane
$C=C$	CH_2CH_2	Ethene
CBr	CH_3Br	Bromomethane
$C\#N$	$C=N$	Hydrocyanic acid
$Na.Cl$	$NaCl$	Sodium chloride

The user can explicitly identify the hydrogen bonds, but if one hydrogen bond is identified in the string,

the SMILES interpreter will assume that the user has identified all hydrogens for that molecule.

HC(H)=C(H)(H) Ethene

Because SMILES allows entry of all elements in the periodic table and also utilizes hydrogen suppression, the user should be aware of chemicals with two letters that could be misinterpreted by the computer. For example, 'Sc' could be interpreted as a sulfur atom connected to an aromatic carbon by a single bond, or it could be the symbol for scandium. The SMILES interpreter gives priority to the interpretation of a single bond connecting a sulfur atom and an aromatic carbon. To identify scandium the user should enter [Sc].

3) **Rule Three: Branches**
A branch from a chain is specified by placing the SMILES symbol(s) for the branch between parenthesis. The string in parentheses is placed directly after the symbol for the atom to which it is connected. If it is connected by a double or triple bond, the bond symbol immediately follows the left parenthesis.
Some Examples:

$CC(O)C$	2-Propanol
$CC(=O)C$	2-Propanone
$CC(CC)C$	2-Methylbutane
$CC(C)CC(=O)$	2-Methylbutanal
$c1c(N(=O)=O)cccc1$	Nitrobenzene
$CC(C)(C)CC$	2,2-Dimethylbutane

4) **Rule Four: Rings**
SMILES allows a user to identify ring structures by using numbers to identify the opening and closing ring atom. For example, in C1CCCCC1, the first carbon has a number '1' which connects by a single bond with the last carbon which also has a number '1'. The resulting structure is cyclohexane. Chemicals that have multiple rings may be identified by using different numbers for each ring. If a double, single, or aromatic bond is used for the ring closure, the bond symbol is placed before the ring closure number.
Some Examples:

C=1CCCCC1	Cyclohexene
c1ccccc1	Benzene
C1OC1CC	Ethyloxirane
1cc2ccccc2cc1	Naphthalene

5) **Rule Five: Charged Atoms**
Charges on an atom can be used to override the knowledge regarding valence that is built into SMILES software. The format for identifying a charged atom consists of the atom followed by brackets which enclose the charge on the atom. The number of charges may be explicitly stated (-1) or not (-).
Some Examples:

CCC(=O)O-1	Ionized form of propanoic acid
c1ccccn+11CC(=O)O	1-Carboxylmethyl pyridinium

Figure 2 shows some more examples of SMILE representation a) Representation is constructed using the fundamental rules of branching, chaining, and atom naming. b) Using the ring rule, the structure is broken in the middle and named accordingly. c) The structure is broken multiple times and named. If an atom has multiple branch breaks, the naming convention will include annotations indicating the number of branch breaks. d) Cis and trans molecules are represented using the symbols '/' and '\', respectively. e) Chiral center molecules are represented using the '@' symbol. Refer to [5] and [6] references for the better understanding.

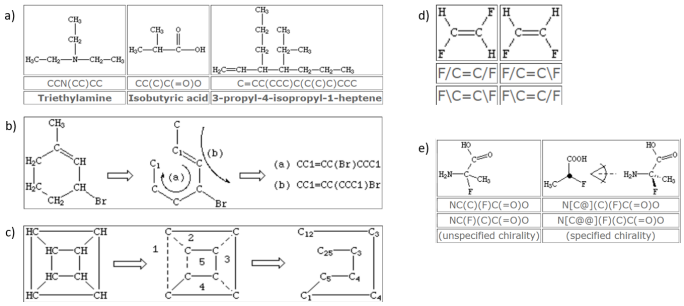

Fig. 2: Examples of SMILES

RDKIT: RDKit is a widely-used open-source software toolkit designed for cheminformatics, which encompasses the storage, retrieval, analysis, and manipulation of chemical information. Developed to facilitate drug discovery and chemical research, RDKit provides a robust set of tools that can handle a variety of tasks in computational chemistry and bioinformatics.

In this study, we utilized RDKit, an open-source cheminformatics toolkit, to process molecular structures, generate molecular descriptors, and apply these descriptors in machine learning (ML) models to predict the glass transition temperature (Tg) of molecules.For implementation refer to [7] of the github link

Methodology:

- **Step 1: Processing SMILES Strings -** SMILES (Simplified Molecular Input Line Entry System) strings are a compact way to represent chemical structures using short ASCII strings. RDKit can convert SMILES strings into molecular objects that can be further analyzed and manipulated
- **Step 2: Generating Molecular Descriptors -** Molecular descriptors are numerical values that describe various aspects of a molecule's structure and properties. RDKit provides tools to calculate a wide array of descriptors(approximately 200 descriptors), which can be used as features in machine learning models.
- **Step 3: Preparing the Dataset -** We prepared a dataset containing SMILES strings of polymers and their corresponding Tg values. The dataset was split into training and testing sets to build and evaluate the ML models.
- **Step 4: Building the Machine Learning Model -** We have tried different machine learning algorithms to model the relationship between molecular descriptors and Tg. As outlined in the methodology section, we have employed all the machine learning algorithms listed in the machine learning methods section.

- **Step 5: Model Evaluation and Interpretation -** The performance of the model was evaluated using the Mean Absolute Error (MAE),Root Mean Squared Error (RMSE), and Coefficient of Determination (R2 Score) metrics which are discussed in detail in the future section. The importance of each descriptor in predicting Tg was analyzed to interpret the model.

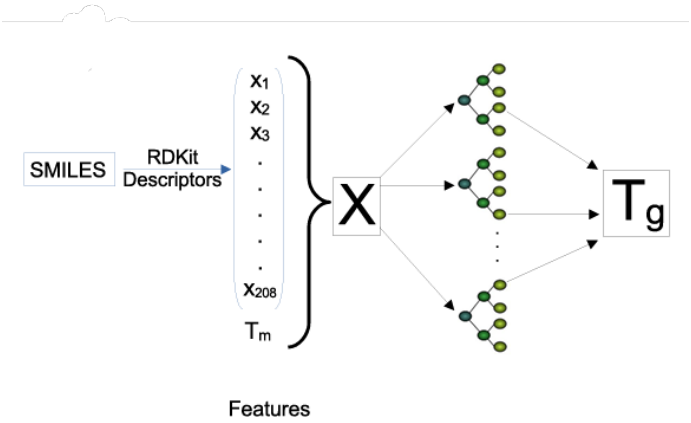

Fig. 3: Schematic representation of the workflow of SMILES mode. It uses molecular descriptors from a SMILES string.

IV. Theoretical Models

The prediction of glass transition temperature (Tg) involves several theoretical models, each based on different principles and assumptions about the nature of the glass transition. Here are some of the most prominent theoretical models:

A. Boyer-Beaman Rule

The Boyer-Beaman rule is an empirical relationship used to estimate the glass transition temperature (T_g) of polymers based on their molecular structure and composition. This rule was introduced by Robert F. Boyer and Ronald G. Beaman in their study on the thermal properties of polymers. The rule provides a simple yet effective way to predict T_g, which is a critical property for understanding the thermal behavior and processing characteristics of polymeric materials. It predicts glass transition temperature (T_g) based on the melting temperature (T_m) of the substance

$$T_g = g \cdot T_m$$

where:

- g is a constant, whose value was found to be approximately 0.7
- T_m is the melting temperature.

The Boyer-Beaman rule provides a straightforward method for estimating the Tg of polymers from their molecular weight, aiding in the design and synthesis of materials with desired thermal properties. However, it is important to note that this rule is empirical and may not account for all factors influencing Tg, such as polymer branching, copolymer composition, or specific intermolecular interactions.

B. Shiraiwa et al

The Shiraiwa et al model is an advanced theoretical approach for predicting the glass transition temperature (Tg) of organic compounds, particularly those found in atmospheric aerosols. Developed by Shiraiwa and colleagues, this model integrates molecular properties with environmental parameters to provide a comprehensive understanding of the Tg of organic mixtures. It is based on the molar mass and atomic oxygen-to-carbon (O/C) ratio of organic compounds:

$$T_g = A + B \cdot M + C \cdot M^2 + D \cdot \left(\frac{O}{C}\right) + E \cdot M \cdot \left(\frac{O}{C}\right)$$

where:

- M is the molar mass
- O/C is the atomic oxygen-to-carbon ratio
- $A = -21.57 \pm 13.47\,[\mathrm{K}]$
- $B = 1.51 \pm 0.14\,[\mathrm{K\ molg^{-1}}]$
- $C = 1.7 \times 10^{-3} \pm 3 \times 10^{-4}\,[\mathrm{K\ mol^2\ g^{-2}}]$
- $D = 131.4 \pm 16.01\,[\mathrm{K}]$
- $E = -0.25 \pm 0.085\,[\mathrm{K\ mol\ g^{-1}}]$

The Shiraiwa et al model is significant in atmospheric chemistry and environmental science, as it helps predict the physical state of organic aerosols, which influences their reactivity, transport, and impact on climate. However, it is important to note that the model assumes ideal mixing and may not fully capture complex interactions in highly diverse mixtures. Therefore, empirical adjustments or additional experimental validation may be necessary for more accurate predictions in specific cases.

V. Evaluation Metrics

To evaluate the performance of the machine learning models used in predicting the glass transition temperature (Tg), we utilized three key metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2 score). Each of these metrics provides unique insights into the accuracy and reliability of the models.

A. Mean Absolute Error (MAE)

Mean Absolute Error (MAE) measures the average magnitude of errors in a set of predictions, without considering their direction. It is the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

where:

- n is the number of observations.
- y_i is the actual value.
- $\hat{y}_i$ is the predicted value.

MAE is a linear score, which means that all the individual differences are weighted equally in the average. It is easy to understand and interpret, making it a popular metric for

regression problems. Lower MAE values indicate better model performance, as they signify smaller average errors.

B. Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is a quadratic scoring rule that also measures the average magnitude of error. However, it gives higher weight to large errors, making it more sensitive to outliers compared to MAE. RMSE is the square root of the average of squared differences between prediction and actual observation.

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

where:

- n is the number of observations.
- y_i is the actual value.
- $\hat{y}_i$ is the predicted value.

RMSE is particularly useful when large errors are particularly undesirable. It penalizes larger errors more than smaller ones due to the squaring of the differences. Lower RMSE values indicate better model performance, as they imply that the predicted values are close to the actual values with fewer large deviations.

C. Coefficient of Determination (R^2 Score)

The Coefficient of Determination, commonly known as the R^2 score, is a statistical measure that explains the proportion of the variance in the dependent variable that is predictable from the independent variables. It provides an indication of goodness-of-fit and typically ranges from 0 to 1.

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

where:

- y_i is the actual value.
- $\hat{y}_i$ is the predicted value.
- $\bar{y}$ is the mean of the actual values.

The R^2 score can be interpreted as the percentage of the response variable variation that is explained by the model. An R^2 score of 1 indicates that the model perfectly explains the variability of the response data, whereas an R^2 score of 0 indicates that the model does not explain any of the variability in the response data. Higher R^2 values indicate better model performance.

VI. Dataset Analysis and the Results

A. Data Analysis

- **Strong Correlation with Molecular Weight:** The feature ”M / g/mol” (molecular weight) shows a strong positive correlation with Tg, with a correlation coefficient of 0.76. This indicates that compounds with higher molecular weight tend to have higher glass transition temperatures.
- **Influence of Functional Groups:**

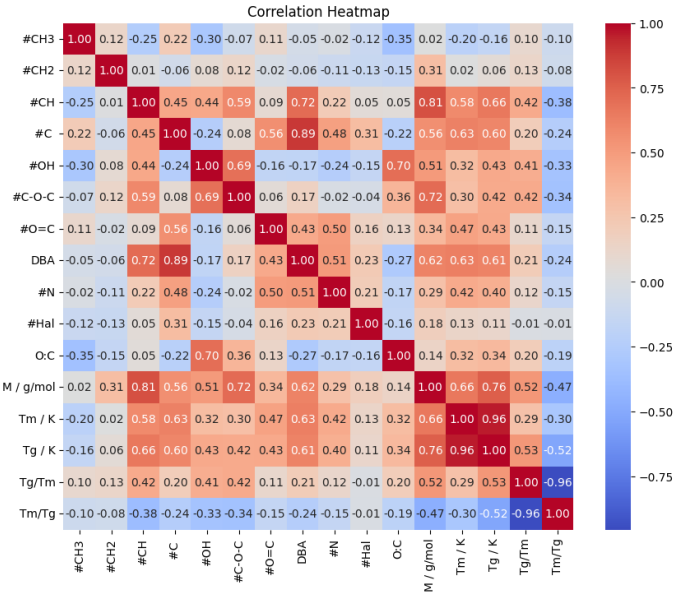

Fig. 4: Correlation Map

  - #OH (number of hydroxyl groups) has a moderate positive correlation with Tg (0.43). Hydroxyl groups likely increase the Tg due to their ability to form hydrogen bonds, which can increase rigidity.
  - #C-O-C (number of ether linkages) also shows a positive correlation with Tg (0.42), suggesting that ether linkages contribute to a higher glass transition temperature.
- **Branching Effect:** The feature ”O:C” (oxygen to carbon ratio) has a moderate positive correlation with Tg (0.34). This suggests that branching, as represented by the oxygen-to-carbon ratio, influences the glass transition temperature, potentially increasing it due to increased molecular interactions and structural complexity.
- **Least Influential Features:** Features such as #CH3 (number of methyl groups), #CH2 (number of methylene groups), #CH (number of methine groups), #C (total number of carbon atoms), #N (number of nitrogen atoms), and #Hal (number of halogen atoms) exhibit low to negligible correlations with Tg (all below 0.1). These features are less important in predicting the glass transition temperature.
- **Boyer-Beaman Relation:** The correlation between Tg and Tm (melting temperature) is very strong, with a coefficient of 0.96. This supports the Boyer-Beaman relation, which states that Tg is approximately 0.7 times Tm ($Tg/Tm \approx 0.7$). The high correlation indicates that Tg strongly depends on Tm, validating this empirical relation for the dataset.
- **Inclusion of Sulfur Atoms:** The feature #S (number of sulfur atoms) shows a moderate positive correlation with Tg (0.34), indicating that the presence of sulfur atoms in a compound can influence its glass transition temperature.

B. Results before Novel Idea Implementation

```
Model Performance for Tg Prediction:
------------------------------------------------------------
Model: Linear Regression
    MAE: 5.7700
    RMSE: 8.5639
    R^2: 0.9927
------------------------------------------------------------
Model: Random Forest
    MAE: 5.2828
    RMSE: 8.7758
    R^2: 0.9924
------------------------------------------------------------
Model: Gradient Boosting
    MAE: 6.0684
    RMSE: 8.3013
    R^2: 0.9932
------------------------------------------------------------
Model: XGBoost
    MAE: 6.4592
    RMSE: 9.8406
    R^2: 0.9904
------------------------------------------------------------
Model: Extra Trees
    MAE: 5.6442
    RMSE: 8.6761
    R^2: 0.9925
------------------------------------------------------------
```

Fig. 5: Results before Novel Idea Implementation

```
Model Performance for Tg Prediction:
------------------------------------------------------------
Model: Linear Regression
    MAE: 5.7370
    RMSE: 8.5415
    R^2: 0.9928
------------------------------------------------------------
Model: Random Forest
    MAE: 5.3246
    RMSE: 9.0742
    R^2: 0.9918
------------------------------------------------------------
Model: Gradient Boosting
    MAE: 6.2035
    RMSE: 8.5041
    R^2: 0.9928
------------------------------------------------------------
Model: XGBoost
    MAE: 6.1789
    RMSE: 8.7936
    R^2: 0.9923
------------------------------------------------------------
Model: Extra Trees
    MAE: 5.8841
    RMSE: 8.4493
    R^2: 0.9929
------------------------------------------------------------
```

Fig. 6: Results after Novel Idea Implementation

Reason for Improvement:

- #S (Sulfur Atoms): Sulfur atoms are significant due to their molecular weight and the impact on the molecular structure's rigidity, influencing Tg.
- #branching (Branching): Branching affects the physical properties and reactivity of molecules by altering their shape and molecular interactions. This structural complexity is crucial in predicting Tg accurately.

1) Result Analysis for XGBoost with Grid Search CV: The optimal hyperparameters for the XGBoost model were identified using Grid Search CV, resulting in significant improvements in the model's performance for predicting the glass transition temperature (Tg). The best results obtained are as follows:

- **Best RMSE:** 7.3711
- **Best R^2 Score:** 0.9946
- **Best MAE:** 5.0727

Interpretation:

- **Best Parameters:** The optimal set of hyperparameters includes:
  - **colsample_bytree = 0.9:** 90% of the features are randomly sampled for each tree, which helps in reducing overfitting by introducing diversity among trees.
  - **learning_rate = 0.2:** A relatively high learning rate ensures faster convergence and effective learning from each boosting iteration.
  - **max_depth = 3:** Limiting the maximum depth of each tree to 3 helps in controlling model complexity and prevents overfitting.
  - **n_estimators = 150:** The model is built using 150 boosting rounds, balancing complexity and performance, ensuring that the model learns adequately without overfitting.
  - **subsample = 0.8:** 80% of the training data is randomly sampled for each tree, promoting robustness and reducing overfitting by ensuring that not all data is used for training each tree.
- **Best RMSE (7.3711):** Indicates the model's average prediction error in terms of root mean squared error, showing that the model's predictions are very close to the actual values, thus ensuring high prediction accuracy.
- **Best R^2 Score (0.9946):** Reflects the proportion of variance in the dependent variable that is predictable from the independent variables. An R^2 score close to 1 signifies excellent model performance and high predictive accuracy.
- **Best MAE (5.0727):** Represents the average absolute error between the predicted and actual values, indicating that the model's predictions are highly accurate on average.

Overall, the tuned XGBoost model with the identified optimal hyperparameters demonstrates superior performance in predicting the glass transition temperature, achieving high accuracy and low error rates. The results highlight the effectiveness of the model and the importance of hyperparameter tuning in improving predictive performance.

C. Results for SMILES Idea Implementation

Reason for Decrease in Model Accuracy:

- #Data Quantity: Insufficient data can lead to overfitting, where the model learns the training data well but fails to generalize to new data.
- #Imbalanced Data: When some classes are underrepresented, the model may become biased towards the majority class, leading to poor performance on the minority classes.

```
Model Performance for Tg Prediction:
------------------------------------------------------------
Model: Random Forest
    MAE: 14.2512
    RMSE: 20.1972
    R^2: 0.9542
------------------------------------------------------------
Model: Gradient Boosting
    MAE: 14.3426
    RMSE: 21.0976
    R^2: 0.9500
------------------------------------------------------------
Model: XGBoost
    MAE: 13.2279
    RMSE: 18.8466
    R^2: 0.9601
------------------------------------------------------------
Model: Extra Trees
    MAE: 12.3550
    RMSE: 18.1562
    R^2: 0.9630
------------------------------------------------------------
```

Fig. 7: Results for SMILES Idea Implementation

- #Increased Complexity: The number of features generated using RDKIT is extensive, which could increase the complexity of the model. It may also include unnecessary features that are not related to predicting the glass transition temperature.

These are the major reasons for the decrease in SMILES model accuracy as compared to the Functional Group model accuracy.

Best Results: The best model is the extra trees which give best results compared to the other model.The best results obtained are as follows:

- **Best RMSE:** 18.1562
- **Best R^2 Score:** 18.1562
- **Best MAE:** 0.9630

VII. Comparative Analysis of Functional Group-Based and SMILES-Based Models

1. Histogram plot of glass transition temperature distribution of unique entries:

Various experimental glass transition temperature (T_g) data exist for the same substance from different sources, methods, and conditions. A past study on predicting T_g using a machine learning (ML) algorithm found that using the median of multiple reported T_g values yielded the best results in terms of root mean squared error (RMSE).This study adopts the same mean-based method to handle duplicate T_g samples.

Molecules may not be uniquely described by feature representations, leading to ambiguities. Ambiguities can occur in the molecular constitution or configuration, causing originally distinct molecules to be represented by the same feature vector. Molecular configuration has minor importance, but molecular constitution can significantly affect T_g, as seen in examples like 2-pentanol vs. 3-pentanol and sucrose vs. trehalose.

The dataset shrinks in unique x-values due to feature representation ambiguities. Multiple y-values can exist for some x-values. Visualized in Figure 2: Initial dataset: 394 entries. SMILES Mode: 368 unique entries (due to lack of stereo configuration resolution by RDKit descriptors). Functional Group Mode: 316 unique entries (less powerful in uniquely defining molecules).

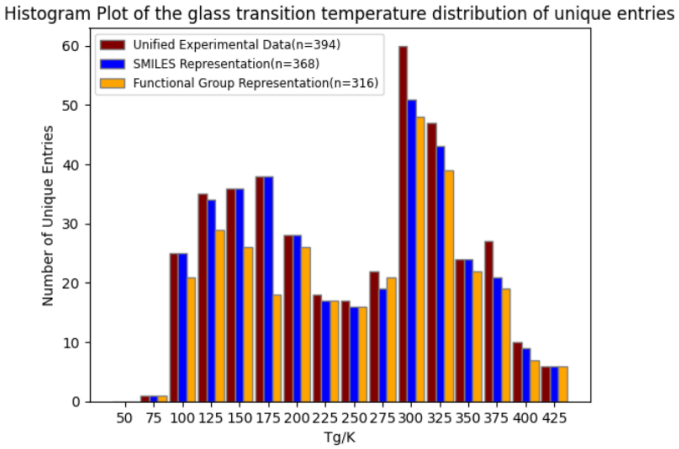

Fig. 8: Histogram plot of glass transition temperature distribution of unique entries

2. Comparison of predicted and experimental values in both mode:

The graphs compare both the Functional Group Model and SMILES Model against the Boyer-Beaman and Shiraiwa theoretical models. The black line represents the accurate line, and the models closer to this line demonstrate better results.

The difference between the experimental and predicted glass transition temperatures ($\Delta T_g = \mathrm{T}_{g,\mathrm{exp}} - T_{g,\mathrm{pred}}$)

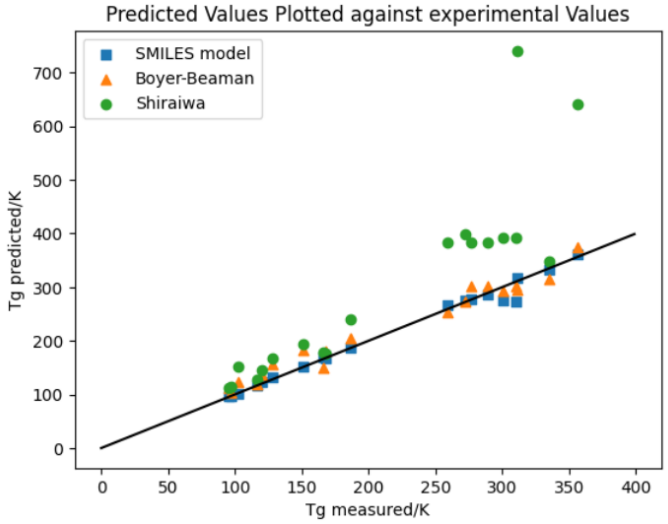

Fig. 9: Comparison of predicted and experimental values in SMILES model

3. Error analysis:

Figure 11 and Figure 12 illustrates the deviation of individual duplicate T_g values from their mean value. Duplicates arise from: Multiple measurements of the same substance and Same feature representation for different substances.Both

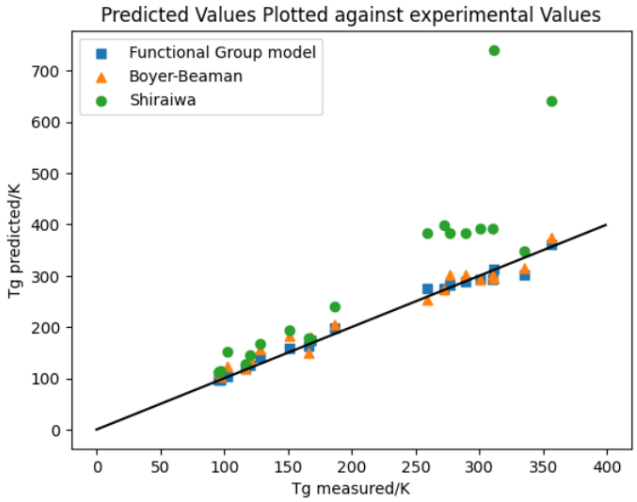

Fig. 10: Comparison of predicted and experimental values in Functional Group model

Functional Group Mode (Figure 12) and SMILES Mode (Figure 11) show a normal distribution centered around zero. Normal distribution likely results from most duplicates being doublets.

The normal distribution of deviations supports the use of the median T_g value for handling duplicate data points.

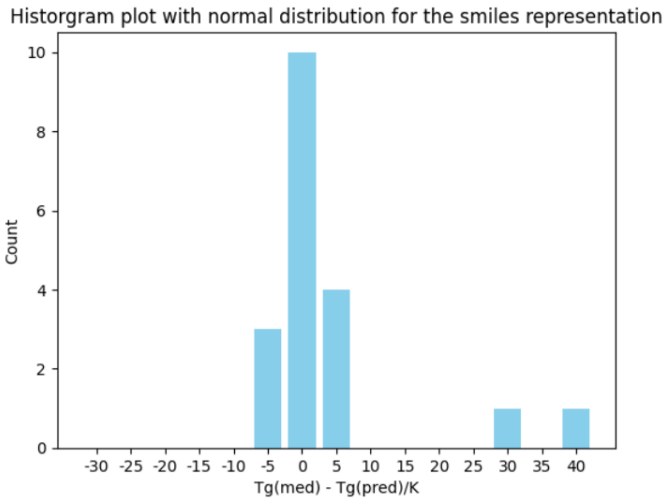

Fig. 11: Error analysis of SMILES model

VIII. Future Work

The study of glass transition temperatures (Tg) for organic compounds using machine learning models has shown promising results. However, several areas can be explored further to enhance the predictive accuracy and applicability of these models. The following points outline the potential directions for future work:

1. Boyer-Beaman Relation: The Boyer-Beaman relation[2][3] of $\frac{Tg}{Tm} = 0.7$ significantly influences the glass transition temperature. Future research can delve deeper into understanding and validating this relationship across a wider range of compounds. Incorporating this empirical relation into machine learning models could potentially improve their accuracy by providing an additional constraint based on known physical properties.

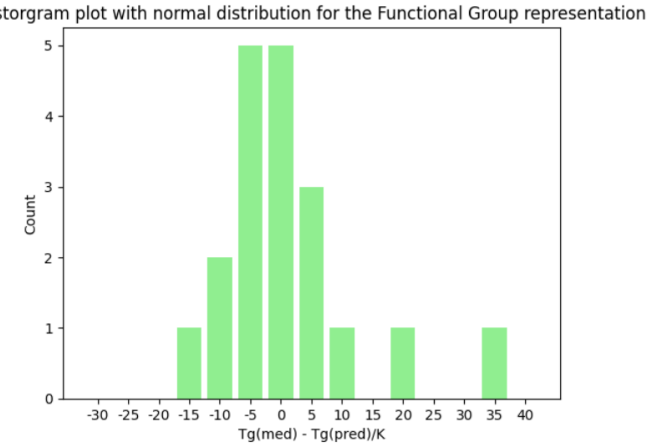

Fig. 12: Error analysis of Functional Group model

2. Model Accuracy: The models used by Armeli et al.[1] have already achieved an accuracy mark of 97%. Building on this foundation, future work can focus on refining these models by exploring more advanced machine learning techniques, such as deep learning and ensemble learning. Hyperparameter optimization and feature engineering can also be leveraged to further improve model performance.

3. Addressing Exceptions: Some exceptions persist in the predictions, which necessitate lab work[4] and can be considered as outliers. Future studies should aim to investigate these exceptions to understand the underlying reasons for their deviation from the model predictions. This could involve experimental validation and exploring additional features or environmental factors that might influence Tg.

4. Experimental Setups: The paper "Structure – Glass transition temperature relationship for non-polymeric molecules: The concept of internal plasticizing effect"[4] published in 2024 suggests that experimental setups can help predict the glass transition temperature more accurately without relying on the melting temperature. Future research should integrate experimental data with machine learning models to validate and enhance predictive accuracy. This approach can bridge the gap between theoretical predictions and practical applications.

5. Inclusion of Polymers and Polygels: While studying the glass transition temperatures of organic compounds, this research did not account for polymers or repetitive chain-like compounds. Future work should extend the scope to include polymers and the polygels, as they constitute a significant portion of materials where Tg is a critical property. Developing models that can accurately predict Tg for both small molecules and polymers would greatly enhance the applicability of these predictive tools.

REFERENCES

[1] Gianluca Armeli, Jan-Hendrik Peters, and Thomas Koop, "Machine-Learning-Based Prediction of the Glass Transition Temperature of Organic Compounds Using Experimental Data," *ACS Omega*, vol. 8, no. 13, pp. 12298–12309, 2023.

[2] Ralph G. Beaman, "Relation between (apparent) second-order transition temperature and melting point," *Journal of Polymer Science*, vol. 9, no. 5, pp. 470–472, 1952.

[3] Raymond F. Boyer, "Relationship of first-to second-order transition temperatures for crystalline high polymers," *Journal of Applied Physics*, vol. 25, no. 7, pp. 825–829, 1954.

[4] Andrzej Nowok, Hubert Hellwig, Kajetan Koperwas, Wioleta Cieślik, Mateusz Dulski, Piotr Kuś, Marian Paluch, and Sebastian Pawlus, "Structure–Glass transition temperature relationship for non-polymeric molecules: The concept of internal plasticizing effect," *Journal of Molecular Liquids*, 2024, p. 124222.

[5] Anderson, E., G.D. Veith, and D. Weininger. 1987. SMILES: A line notation and computerized interpreter for chemical structures. Report No. EPA/600/M-87/021. U.S. Environmental Protection Agency, Environmental Research Laboratory-Duluth, Duluth, MN 55804

[6] Hunter, R.S., F.D. Culver, and A. Fitzgerald. 1987. SMILES User Manual. A Simplified Molecular Input Line Entry System. Includes extended SMILES for defining fragments. Review Draft, Internal Report, Montana State University, Institute for Biological and Chemical Process Control (IPA), Bozeman, MT.

[7] https://github.com/gashawmg/moleculardescriptors/blob/main/Molecular

[8] Weininger, D. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. Journal of Chemical Information and Computer Science 28: 31-36.